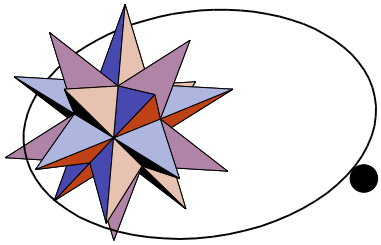
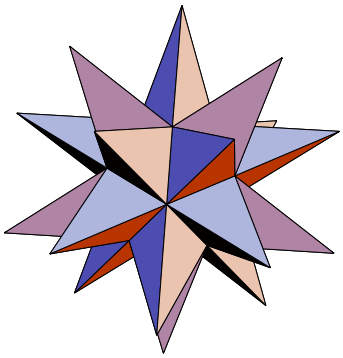
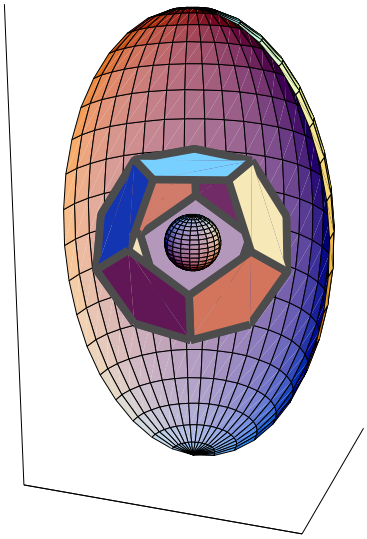




"Kleine Einführung" in *Mathematica*

"Petite Introduction" en *Mathematica*



Scripta bilingua

Ein Kurs zum Selbststudium mit Beispielen und Konzepten nach den Vorbildern "Handbuch Mathematica" von S. Wolfram, "Mathematica: A Practical Approach" von Nancy Blachman und andern Werken.
■ Un cours pour études individuelles avec des exemples et concepts d'après les modèles "Livre de Mathematica" de S. Wolfram, "Mathematica: A Practical Approach" de Nancy Blachman et d'autres oeuvres.

Zusammengestellt von ■ composé par
Rolf Wirz
Ingenieurschule Biel // école d'ingénieurs Bienne // HTA Biel-Bienne
1993/94/95/98/99/2007

Lauffähige Files für Windows Output mit *Mathematica* 5.2
■ Des dossiers - (Files) - exécutables pour Windows ■ Edition sans output

Copyright Rolf Wirz

Konzept ▮ Concept

-. Read_me_Kurs

I. Rundgang in *Mathematica*: Einstieg und Uebersicht
▮ Tour en *Mathematica*: Introduction et vue d'ensemble

(Beispiele nach Wolfram u.a.
▮ Exemples d'après Wolfram et autres)

II. Kurs: Systematische Einführung in die Sprache
▮ Cours: Introduction systématique dans la langue de programmation

(Beispiele nach Blachman u.a., im Print gemischt mit III
▮ Exemples d'après Blachman et aautres, dans le "print" mêlé à III)

III. Uebungen zur systematischen Einführung
▮ Exercices pour l'introduction systématique

(Beispiele nach Blachman u.a., im Print gemischt mit II
▮ Exemples d'après Blachman et autres, dans le "print" mêlé à II)

IV. Uebungen zu speziellen Unterrichtsthemen
▮ Exercices pour des sujets spéciaux

(Diverse Beispiele -> Praktikum ▮ Exemples divers -> Laboratoire)

Inhalt ▮ Contenu

-. Zur Filestruktur (Konzept)
▮ Quant à la structure des fichiers (concept)

I. Rundgang in *Mathematica*: Einstieg und Uebersicht
▮ Tour en *Mathematica*: Introduction et vue d'ensemble

Uebersicht
▮ Vue d'ensemble

1. Numerische Rechnungen
▮ Calculs numériques

2. Graphiken
▮ Graphiques

3 . Algebra und Analysis
▮ Algèbre et analyse

4. Gleichungen lösen
▮ Résoudre des équations

5. Listen
▮ Listes

6. Matrizen
▮ Matrices

7. Transformationsregeln und Definitionen
▮ Règles de transformation et définitions

8. Symbolisches Rechnen
▮ Calculs symboliques

9. Programmierung
▮ Programmation

10. Mathematica Packages
■ Packages en *Mathematica*

11. Datenaustausch mit Mathematica, andere Ausgabeformen
■ Echanges de données avec *Mathematica,*

Namen der Files: ■ Noms des fichiers:

Einst_00.nb: (Uebersicht ■ Vue d'ensemble)
Einst_01.nb: (Numerische Rechnungen ■ Calculs numériques)
etc.
.....
.....
.....
Einst_11.nb: (Datenaustausch mit Mathematica, andere Ausgabeformen
■ Echanges de donnés avec *Mathematica*, d'autres possibilités d'output)

II. Kurs: Systematische Einführung in die Sprache
■ Cours: Introduction systématique dans la langue de programmation

In diesem Inhaltsverzeichnis ist aus Platzgründen pro File nur die oberste Gliederungsebene wiedergegeben.
■ Dans cet index nous n'avons reproduit pour des raisons de place que le niveau supérieur de répartition par fichier.

1. Einstieg in Mathematica
▮ Introduction en *Mathematica*

1.1. Maschinenspezifisches
▮ Problèmes spécifiques selon la machine

1.2. Versionspezifisches
▮ Problèmes spécifiques selon la version

1.3. Front-End versus Kernel
▮ Front-End versus Kernel

1.4. Notation
▮ Notation

1.5. Mathematica interaktiv benutzen
▮ Utiliser *Mathematica* interactivement

1.6. Reservierte Namen
▮ Des noms réservés

1.7. Help-Funktion und Kommando-Vervollständigung
▮ Fonction Help et achèvement des commandes

1.8. Help-Funktion und Kommando-Vervollständigung bei Zeichen
▮ Fonction Help et achèvement des commandes pour des symboles

1.9. Die verschiedenen Klammertypen
▮ Différents types de parenthèses

1.10. Packages
▮ Packages

Files ▮ Fichiers:

Kurs_01.nb, Ueb_01.nb

2. Numerische Probleme
▮ Problèmes numériques

2.1. Arithmetische Operationen
▮ Opérations arithmétiques

2.2. Rationale Zahlen
▮ Nombres rationaux

2.3. Irrationale Zahlen
▮ Nombres irrationaux

2.4. Annäherung durch Dezimalbrüche
▮ Approximation par fractions décimales

2.5. Komplexe Zahlen
▮ Nombres complexes

2.6. Symbole und Zahlen, Listen etc.
▮ Symboles et nombres, listes etc.

2.7. Approximationen
▮ Approximations

2.8. Formatierte Zahlenausgabe
▮ Représentations de nombres formatés

2.9. Wichtige gespeicherte Konstanten
▮ Constantes mises en mémoire importantes

2.10. Zufallszahlen
▮ Nombre probables

2.11. Iteratoren
▮ Itérateurs

2.12. Matrizenrechnung
▮ Calcul des matrices

2.13. Lösen von Gleichungen
▮ Résoudre des équations

2.14. Numerische Integration etc.
▮ Intégration numérique

2.15. Numerische Lösung von Differentialgleichungen
▮ Solution numérique d'équations différentielles

Files ▮ Fichiers:

Kurs_02.nb, Ueb_02.nb

3. Algebraische und symbolische Stärken
▮ Des avantages algébriques et symboliques

3.1. Algebra
▮ Algèbre

3.2. Gleichungen lösen, Lösung weiterverwenden
▮ Résoudre des équations différentielles, réutiliser les solutions

3.3. Vereinfachungen
▮ Simplifications

3.4. Summation
▮ Sommation

3.5. Calculus (Differential- und Integralrechnung)
▮ Calcul (différentiel et intégral)

3.6. Grenzwerte
▮ Valeurs limites

3.7. Potenzreihen
▮ Séries de puissances

3.8. Lösen von Differentialgleichungen
▮ Résoudre des équations différentielles

Files ▮ Fichiers:

Kurs_03.nb, Ueb_03.nb

4. Graphiken
Graphiques

4.1. Zweidimensionale Plots
Plots à deux dimensions

4.2. Options
Options

4.3. Mehrere Graphen in einem Bild
Plusieurs graphiques en un diagramme resp. image

4.4. Parametrisierte Kurven
Courbes paramétrisées

4.5. Weitere Options
D'autres options

4.6. Plots von Flächen im Raum
Plots de surfaces dans l'espace

4.7. Options für Farben, Beleuchtung, ...
Options pour couleurs, éclairage, ...

4.8. Arbeiten mit Daten
Travail avec des données

4.9. Eingebaute Graphikelemente oder Blöcke
Eléments graphiques ou blocs incorporés

4.10. Beschriftung von Graphiken (Labels)
Inscriptions sur graphiques (Labels)

4.11. Graphik-Pakete
Paquets de graphiques

4.12. Animationen
Animations

4. 13. PostScript
PostScript

4. 14. Experimentieren mit Animationen
Expérimenter avec animations

4.15. Probiere eigene Beispiele aus!
▮ Essaie des exemples propres

Files ▮ Fichiers:

Kurs_04.nb, Ueb_04.nb

5. Etwas Umgang mit *Mathematica*
▮ Savoir un peu manier *Mathematica*

5.1. Seiten-Breite setzen
▮ Mettre la largeur d'une page

5.2. Listings von Input und Output
▮ Listings d'input et output

5.3. Zum Editor
▮ Quant à l'éditeur

5.4. Zur On-Line Hilfe
▮ Quant à l'aide on-line

5.5. Ein Blank kann "oder" bedeuten...
▮ Un vide peut signifier "ou" ....

5.6. Symbollisten erzeugen
▮ Créer des listes de symboles

5.7. Verschiedene Möglichkeiten, eine Funktion zu codieren
▮ Différentes possibilités de créer des fonctions

5.8. Output unterdrücken
▮ Empêcher l'output

5.9. Timing - Rechenzeit messen
▮ Mesurer le temps de calculation

5.10 Abbruch und Unterbruch
▮ Rupture et interruption

5.11 Globale Variablen
▮ Variables globales

5.12 Spezielle Formen
▮ Formes spéciales

Files ▮ Fichiers:

Kurs_05.nb, Ueb_05.nb

6. Manipulation von Listen
∎ Manipulation de listes

6.0. Einleitung
∎ Introduction

6.1. Erzeugung von Listen
∎ Création de listes

6.2. Umordnen von Listen
∎ Réarranger des listes

6.3. Erweiterung und Verkürzung von Listen
∎ Elargir et abréger des listes

6.4. Zählung der Elemente einer Liste
∎ Compter les éléments d'une liste

6.5. Zusammenfügen von Listen, Beziehungen zwischen Listen
∎ Réunir des listes, relations entre listes

6.6. Veränderung der Form einer Liste
∎ Changer la forme d'une liste

6.7. Elemente herauspicken
∎ Choisir des éléments

6.8. Auswahl von Daten
∎ Choix de données

6.9. Rechnen mit Listen
∎ Calculer avec des listes

6.10. Auf Listen anwendbare arithmetische Funktionen
∎ Fonctions arithmétiques applicables à des listes

6.11. Listable und Map
∎ Listable et map

6.12. Formatierung von Listen
∎ Formater des listes

Files ∎ Fichiers:

Kurs_06.nb, Ueb_06.nb

7. Probleme zum Thema "Zuordnungen und Regeln"
▮ Problèmes au sujet de "coordinations et règles"

7.1. Zuordnungen
▮ Coordinations

7.2. Löschen der Wertzuweisung
▮ Effacer et assigner des valeurs

7.3. Regeln
▮ Règles

7.4. Gleichheit
▮ Egalité

Files ▮ Fichiers:

Kurs_07.nb, Ueb_07.nb

8. Datentypen
∎ Types de données

8.1. Atomare Typen
∎ Types atomiques

8.2. Andere Typen und Prädikatenfunktionen
∎ Autres types et fonctions de prédicats

8.3. Interne Darstellung in *Mathematica*
∎ Représentation interne en *Mathematica*

8.4. Herauslesen von gewissen Teilen von Ausdrücken
∎ Lire certaines parties d'expressions

8.5. Symbole wie "unendlich" etc.
∎ Symboles tels que "infini" etc.

8.6. Memory: Wie Mathematica speichert
∎ Memory: Mise en mémoire d'après *Mathematica*

Files ∎ Fichiers:

Kurs_08.nb, Ueb_08.nb

9. Funktionen definieren
▮ Définir des fonctions

9.1. Einfache Funktionen
▮ Fonctions simples

9.2. Typenprüfung
▮ Examens de types

9.3. Bedingte Ausführung
▮ Exécution conditionnée

9.4. Vorgegebene Werte
▮ Valeurs données

9.5. Die Abarbeitungshierarchie bei Regeln
▮ Hierarchie d'élaboration pour règles

9.6. Zusammenfügen von Regeln mit "f/: "
▮ Réunir des règles par "f/: "

9.7. Dokumentieren der eigenen Funktionen
▮ Documenter les propres fonctions

9.8. Attribute
▮ Attributs

9.9. Die Art der Abarbeitung eines Ausdrucks
▮ La façon d'élaborer une expression

9.10. Diskrete Funktionen
▮ Fonctions discrètes

Files ▮ Fichiers:

Kurs_09.nb, Ueb_09.nb

10. Lokale Variablen und prozedurales Programmieren
▮ Variables locales et programmation "procédurales"

10.1. Globale Variablen
▮ Variables globales

10.2. Lokale Variablen im Unterschied zu globalen Variablen
▮ Variables locales à la différence des variables globales

10.3. Prozedurale Programmierung
▮ Programmation procédurale

Files ▮ Fichiers:

Kurs_10.nb, Ueb_10.nb

11. Problemsammlung zur Musterentsprechung (Mustererkennen, musterkonformes Abarbeiten)
▮ Collection de problèmes pour "la correspondance de patrons" (Reconnaître des patrons, élaborer selon les patrons

11.1. Mustererkennung bei einer Sequenz
▮ **Reconnaître les patrons d'une séquence**

11.2. Nachbau von Funktionen
▮ **Reconstruire des fonctions**

11.3. "Polymorphe" Definitionen
▮ **Définitions "polymorphes"**

11.4. Benennung von Ausdrücken
▮ **Nommer des expressions**

11.5. Ausdrücke auffinden, die mit Mustern übereinstimmen
▮ **Trouver des expressions qui s'accordent aux patrons**

11.6. Das Attribut "Orderless"
▮ **L'attribut "Orderless"**

11.7. Beispiele mit Mustererkennung
▮ **Exemples avec reconnaissance de patrons**

Files ▮ Fichiers:

Kurs_11.nb, Ueb_11.nb

12. Anonyme Funktionen
▮ Fonctions anonymes

12.1. "Function"
▮ "Function"

12.2. Anwendung auf Datenselektion
▮ Utiliser pour la sélection des données

12.3. Neuauflage einer früher definierten Funktion
▮ Nouvelle édition d'une fonction définie auparavant

12.4. Transformation von Wertepaaren in Regeln
▮ Transformation de paires de valeurs en règles

12.5. Mehrere Argumente
▮ Plusieurs arguments

12.6. Daten filtern
▮ Filtrer des données

Files ▮ Fichiers:

Kurs_12.nb, Ueb_12.nb

13. Fallstricke und "Debugging" (Fehler eliminieren)
▮ Pièges et "Debugging" (éliminer les fautes)

13.1. Fehlermeldungen
▮ Rapport d'erreurs

13.2. Fehler durch vordefinierte Variablen
▮ Erreurs par des variables prédéfinies

13.3. Unvollständige Befehle
▮ Ordres incomplets

13.4. Falsche Klammerung
▮ Parenthèses mal utilisées

13.5. Verwechslung von Zeilen- und Spaltenvektor
▮ Confusion entre vecteurs de lignes et de colonnes

13.6. Weiter mit "Return" statt "Enter"
▮ Continuer par "Return" au lieu de "Enter"

13.7. Probleme mit Wurzeln (Mehrdeutigkeiten)
▮ Problèmes de racines (interprétations différentes, équivoques)

13.8. Abarbeitungsreihenfolge
▮ Suites d'élaborations

13.9. Ordnung von Regeln
▮ Ordre de règles

13.10. "Protect" und Funktionen
▮ "Protect" et fonctions

13.11. Debugging
▮ Debugging

Files ▮ Fichiers:

Kurs_13.nb, Ueb_13.nb

14. Input und Output
▮ Input et output

14.1. Input
▮ Input

14.2. Datenexport
▮ Exportation de données

14.3. Manipulation von Strings
▮ Manipulation de strings

14.4. Eingabe- und Ausgabeform
▮ Formes d'entrées et de sorties

14.5. Uebersetzen von Ausdrücken in andere Programmiersprachen und Manipulation von Sourcecode
▮ Traduire des expressions en d'autres langues de programmation et manipulation du sourcecode

14.6. Formate
▮ Formats

Files ▮ Fichiers:

Kurs_14.nb, Ueb_14.nb

15. Packages
▮ Packages

15.1. Wieso Packages? Wie zugreifen?
▮ Pourquoi des packages? Comment les saisir

15.2. Contexts
▮ Contextes

15.3. Context-Wechsel
▮ Changement de contexte

15.4. Zum Laden von Packages
▮ Pour charger des packages

15.5. Der Package-Stil
▮ Le style-package

15.6. Namenskonflikte
▮ Conflits de noms

Files ▮ Fichiers:

Kurs_15.nb, Ueb_15.nb

III.Uebungen zur systematischen Einführung
■ Exercices pour l'introduction systématique

Files ■ Fichiers:

Ueb_01.nb
Ueb_02.nb
Ueb_03.nb
Ueb_04.nb
Ueb_05.nb
Ueb_06.nb
Ueb_07.nb
Ueb_08.nb
Ueb_09.nb
Ueb_10.nb
Ueb_11.nb
Ueb_12.nb
Ueb_13.nb
Ueb_14.nb
Ueb_15.nb

Dazu existieren diverse Daten- und Programmfiles, die der Ordnung im Directory und der Lesbarkeit wegen alle mit "AAA" oder mit "aaa" beginnen. Man findet sie daher zuoberst.
■ Il existe aussi différents dossiers de données et de programmes qui commencent tous par "AAA" ou par "aaa" pour des raisons d'ordre dans le directoir et pour une meilleure lisibilité. On les trouve tout en haut.

IV. Uebungen zu speziellen Unterrichtsthemen
■ Exercices sur des thèmes spéciaux

Es existieren Uebungen zu diversen Themen. Der Themenkreis wird ständig erweitert. Eine lückenlose Auflistung der Files ist hier nicht möglich. Die Filenamen beginnen jeweils mit "Ueb_"
■ Il existe des exercices sur différents thèmes.Le nombre de thèmes s'élargit constamment. Une liste complète des fichiers n'est pas possible. Les noms des fichiers commencent par "Ueb_"

Files ■ Fichiers:

Ueb_1_Thema.ma
etc.
.....
.....
.....
Ueb_xx_Thema.ma

V. Zur Filestruktur
▮ Quant à la structure des fichiers

Die Folder sind z.B. wie folgt gegliedert:
▮ Les directoires sont répartis p. ex. de la façon suivante:

Eigenes "Root"	-> "work"	-> "Daten"
	-> "MathemDF"	-> "Kurs"
		-> "Uebungen"
		-> "KopieDerDaten"
		(Sicherung gegen Datenverlust)
▮ Propre "Root"	-> "work"	-> "Daten"
	-> "MathemDF"	-> "Kurs"
		-> "Uebungen"
		-> "KopieDerDaten"
		(Mesure de sécurité contre la perte des données)

Die Namen sind selbsterklärend. Die Files in "Kurs" können durch Anklicken der aktionsfähigen Zellen (unbedingt der Reihe nach) abgearbeitet und dabei studiert werden.
▮ Les noms s'expliquent eux-mêmes. Les fichiers courrants peuvent être élaborés en marquant (par la souris) les cellules exécutables (maintenir absolument l'ordre) et en même temps ils peuvent être étudiés.

Anmerkung zum Titel "Kleine Einführung in Mathematica":
▮ Remarque quant au titre "Petite introduction en *Mathematica*":

Verglichen mit einem hohen Berg ist auch ein grosses Haus eine kleine Sache.
"Kleine Einführung" muss noch lange nicht "kurze Einführung" bedeuten ....
▮ En comparaison d'une haute montagne, même une grande maison n'est qu'une petite chose. "Petite introduction" ne signifie pas forcément "courte introduction" ...

VI. Literatur zu den Files
▮ Littérature pour fichiers

S. Wolfram: *Mathematica*
Nancy Blachman: *Mathematica*: A Practical Approach
Nancy Blachman: *Mathematica* griffbereit etc. etc. etc. ---> Bibl.!

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

1. Numerische Rechnungen
▮ Calculs numériques

Taschenrechner
▮ Calculatrice de poche

```
In[1]:= 9 + 17
```

```
Out[1]= 26
```

Exakte Resultate
▮ Résulats exacts

```
In[2]:= 3^101
```

```
Out[2]= 154613256219603399310938338929686381810632256600 3
```

Vorheriges Resultat approximiert
▮ Résultat précédent approximé

```
In[3]:= N[%]
```

```
Out[3]= 1.54613×10^48
```

Numerische Resultate beliebiger Genauigkeit
▮ Résultats numériques de précision quelconque

```
In[4]:= N[Sqrt[10],50]

Out[4]= 3.1622776601683793319988935444327185337195551393252
```

Komplex rechnen
▮ Calculs complexes

```
In[5]:= (4 + 7I)^12

Out[5]= 75311256833 + 4027860144 i
```

Mathematische Standardfunktionen
▮ Fonctions mathématiques standard

```
In[6]:= BesselJ[0,15.2]

Out[6]= -0.0544208
```

Nullstellen
▮ Des zéros

```
In[7]:= FindRoot[BesselJ[0,x],{x,14.5}]

Out[7]= {x → 14.9309}
```

Beliebige Genauigkeit von Resultaten
▮ Précision quelconque de résultats

```
In[8]:= N[Zeta[1/2+13I],36]

Out[8]= 0.443004782505368189197897441332849126 - 0.655483098321168943051369649191335506 i
```

Numerische Integrale
Intégraux numériques

```
In[9]:= NIntegrate[Sin[Sin[x]],{x,0,Pi}]

Out[9]= 1.78649
```

Exakte Rechnungen mit natürlichen Zahlen
▮ Calculs exacts avec les nombres naturels

```
In[10]:= FactorInteger[70987635488]

Out[10]= {{2, 5}, {7, 1}, {11, 1}, {17, 1}, {1694701, 1}}
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[11]:= (* Old Form: Remove["Global`@*"] *)

In[12]:= Remove["Global`*"]
```

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

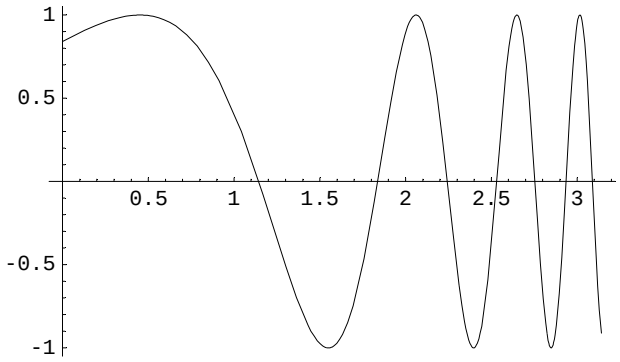
(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

2. Graphiken
▮ Graphiques

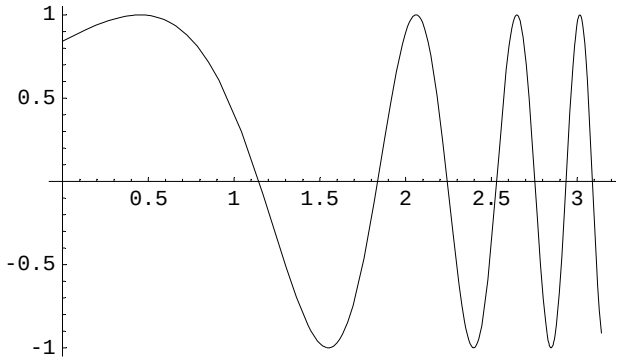
Graph einer Funktion
▮ Graphe d'une fonction

In[1]:= **Plot[Sin[Exp[x]],{x,0,Pi}]**



Out[1]= **-** Graphics **-**

In[2]:= **Plot[Sin[Exp[x]],{x,0,Pi}];**

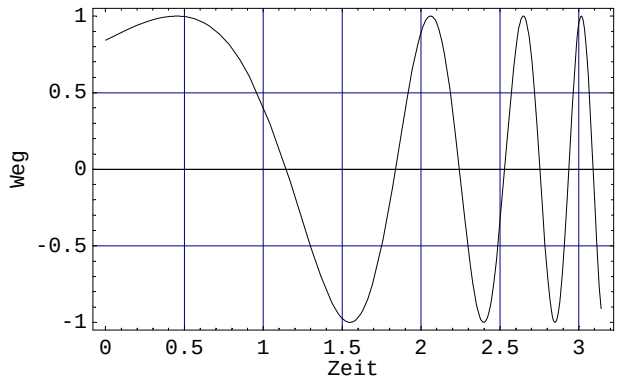


Was war der Unterschied?

Quelle était la différence?

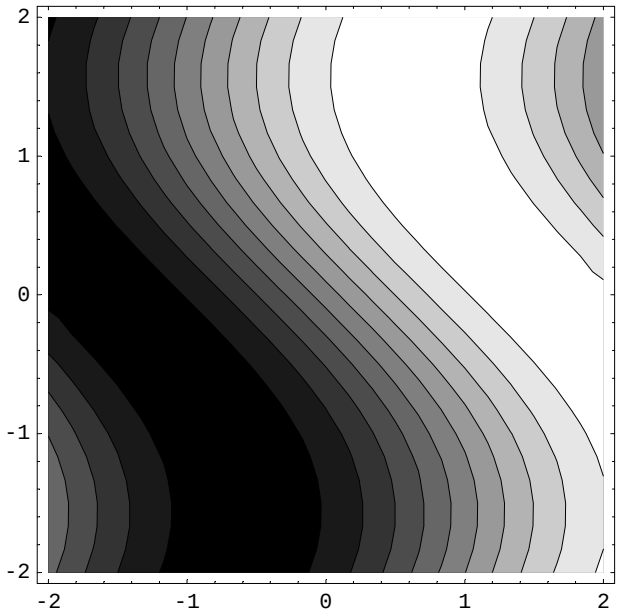
Optionen zum Manipulieren von Plot
Options pour manipuler Plot

In[3]:= **Show[%,Frame->True,FrameLabel->{"Zeit","Weg"},**
GridLines->Automatic];



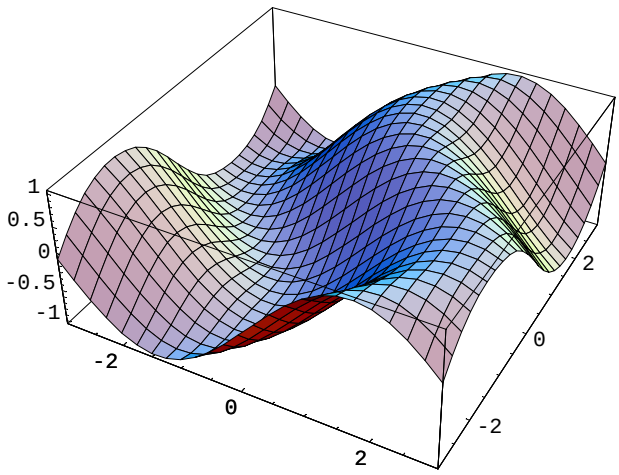
Höhenlinienkarte
▮ Carte des courbes à niveau

In[4]:= **ContourPlot[Sin[x+Sin[y]],{x,-2,2},{y,-2,2}];**



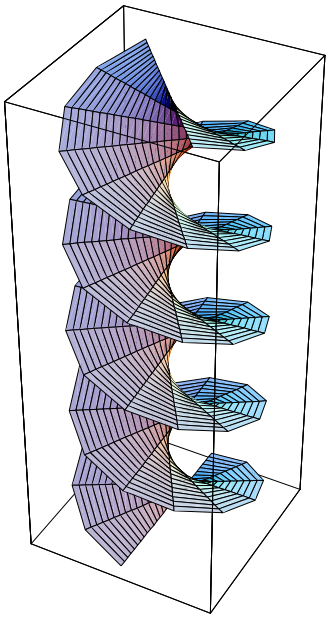
3-dimensionale Graphen
▮ Graphes à 3 dimensions

In[5]:= **Plot3D[Sin[x+Sin[y]],{x,-3,3},{y,-3,3}];**



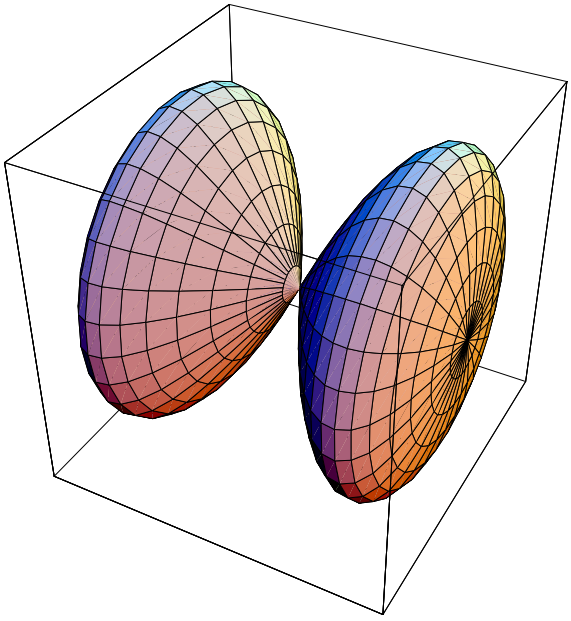
3-dimensionale parametrisierte Flächen
▮ Surfaces paramétrisées à 3 dimensions

```
In[6]:= ParametricPlot3D[{u Sin[t], u Cos[t], t/3},
        {t,0,15},{u,-1,1},Ticks->None];
```



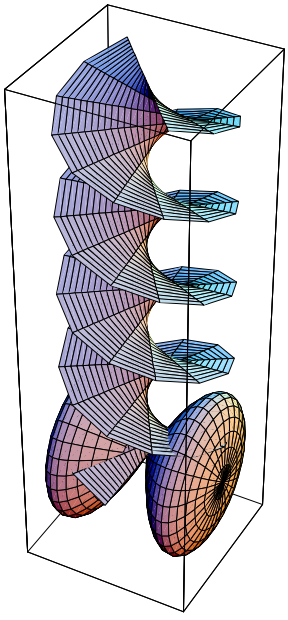
Oder vielleicht so
▮ Ou peut-être ainsi

```
In[7]:= ParametricPlot3D[{Sin[t], Sin[2t] Sin[u], Sin[2t] Cos[u]},
        {t,-Pi/2,Pi/2},{u,0,2 Pi},Ticks->None];
```



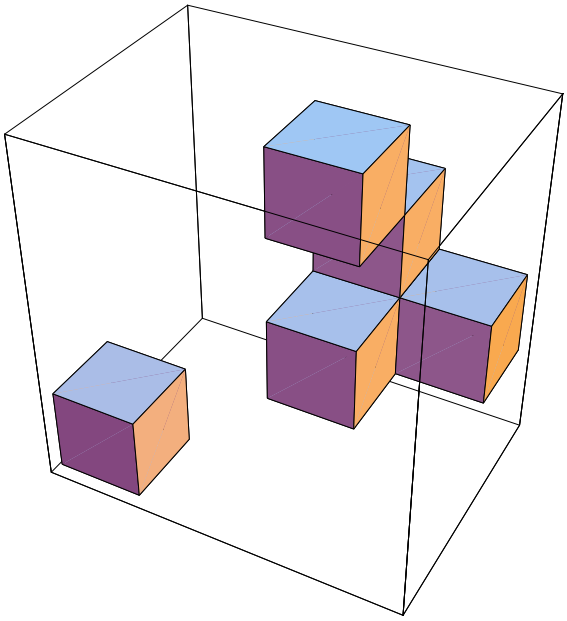
Die beiden letzten Graphen kombiniert
▮ Les deux derniers graphes combinés

```
In[8]:= Show[%,%%];
```



Arbeit mit eingebauten Graphik-Komponenten
▮ Travail avec des composantes de graphiques incorporées

```
In[9]:= Show[Graphics3D[
        {Cuboid[{0,0,0}],Cuboid[{2,2,2}],
         Cuboid[{2,1,3}],Cuboid[{3,2,1}],
         Cuboid[{2,1,1}]} ]]
```

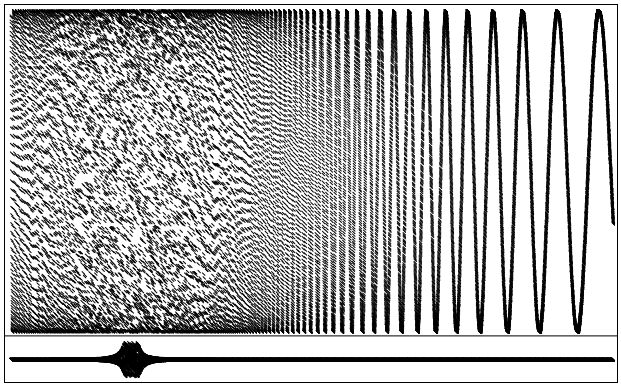


```
Out[9]= - Graphics3D -
```

Musik
Musique

(Das hier dauert etwas lang, bitte warten)
(Cela dure peut-être longtemps, attendre s.v.p.)

In[10]:= **Play[Sin[1000/t],{t,-4,16}]**



Power::infy : Infinite expression $\frac{1}{0.}$ encountered. Mehr…

Out[10]= **-** Sound **-**

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

In[11]:= **(* Old Form: Remove["Global`@*"] *)**

In[12]:= **Remove["Global`*"]**

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

3. Algebra und Analysis
▮ Algèbre et analyse

Ausdrücke umformen
▮ Transformer des expressions

Ausdruck eingeben
▮ Entrer une expression

```
In[1]:= 9(2+x)(x+y)+(x+y)^2
```

$$\text{Out[1]= } 9\,(2+x)\,(x+y)+(x+y)^2$$

Ausdruck hoch 3 ausmultiplizieren
▮ Multiplier une expression puissance 3

```
In[2]:= Expand[%^3]
```

$$\text{Out[2]= } 5832\,x^3+9720\,x^4+5400\,x^5+1000\,x^6+17496\,x^2\,y+30132\,x^3\,y+17280\,x^4\,y+3300\,x^5\,y+17496\,x\,y^2+32076\,x^2\,y^2+19494\,x^3\,y^2+3930\,x^4\,y^2+5832\,y^3+12636\,x\,y^3+8802\,x^2\,y^3+1991\,x^3\,y^3+972\,y^4+1242\,x\,y^4+393\,x^2\,y^4+54\,y^5+33\,x\,y^5+y^6$$

Ausdruck wieder faktorisieren
▮ Réfactoriser une expression

```
In[3]:= Factor[%]
```

$$\text{Out[3]= } (x+y)^3\,(18+10\,x+y)^3$$

Formal integrieren
▮ Intégrer formellement

```
In[4]:= Integrate[x^2 Sin[x]^2,x]
```

$$\text{Out[4]= } \frac{x^3}{6}-\frac{1}{4}\,x\,\text{Cos}[2\,x]-\frac{1}{8}\,(-1+2\,x^2)\,\text{Sin}[2\,x]$$

Resultat wieder differenzieren
▮ Redifférencier le résultat

```
In[5]:= D[%,x]
```

$$\text{Out[5]= } \frac{x^2}{2}-\frac{1}{4}\,\text{Cos}[2\,x]-\frac{1}{4}\,(-1+2\,x^2)\,\text{Cos}[2\,x]$$

Resultat vereinfachen
▮ Simplifier le résultat

```
In[6]:= Simplify[%]
```

$$\text{Out[6]= } x^2\,\text{Sin}[x]^2$$

Resultat als Funktion von x in eine Potenzreihe entwickeln
▮ Développer le résultat comme fonction de x en une série de puissances

```
In[7]:= Series[%,{x,0,14}]
```

$$\text{Out[7]= } x^4-\frac{x^6}{3}+\frac{2\,x^8}{45}-\frac{x^{10}}{315}+\frac{2\,x^{12}}{14175}-\frac{2\,x^{14}}{467775}+O[x]^{15}$$

Symbolisch gegebene Funktion in eine Potenzreihe entwickeln

▮ Développer le résultat comme fonction donnée symboliquement en une série de puissances

```
In[8]:=
        Clear[f];
        Series[ (f[x + h]-f[x - h])/(2h),{h,0,6}]
```

Out[9]= $f'[x] + \frac{1}{6} f^{(3)}[x] h^2 + \frac{1}{120} f^{(5)}[x] h^4 + \frac{f^{(7)}[x] h^6}{5040} + O[h]^7$

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[10]:= (* Old Form: Remove["Global`@*"] *)
```

```
In[11]:= Remove["Global`*"]
```

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

4. Gleichungen lösen
▮ Résoudre des équations

Einfache Gleichung behandeln
▮ Traiter des équations simples

Gleichung eingeben
▮ Entrer une équation

```
In[1]:= x^3 - 7 x^2 + 3 a x == 0
```

$$\text{Out[1]}= \; 3\,a\,x - 7\,x^2 + x^3 == 0$$

Gleichung nach x auflösen
▮ Résoudre une équation d'après x

```
In[2]:= Solve[%,x]
```

$$\text{Out[2]}= \; \left\{\{x \to 0\}, \left\{x \to \frac{1}{2}\left(7 - \sqrt{49 - 12\,a}\right)\right\}, \left\{x \to \frac{1}{2}\left(7 + \sqrt{49 - 12\,a}\right)\right\}\right\}$$

Gleichungssystem lösen nach x und y
▮ Résoudre le système d'équations d'après x et y

```
In[3]:= Solve[{a x + b y == 0, x + y == c},{x,y}]
```

$$\text{Out[3]}= \left\{\left\{x \to -\frac{b\,c}{a-b},\ y \to \frac{a\,c}{a-b}\right\}\right\}$$

Numerische Lösung einer Gleichung höherer Ordnung
▮ Solution numérique d'une équation d'ordre supérieur

```
In[4]:= NSolve[x^5+2x+1==0,x]
```

$$\text{Out[4]}= \{\{x \to -0.701874 - 0.879697\,i\},\ \{x \to -0.701874 + 0.879697\,i\},\ \{x \to -0.486389\},\ \{x \to 0.945068 - 0.854518\,i\},\ \{x \to 0.945068 + 0.854518\,i\}\}$$

Transzendentes Gleichungssystem numerisch lösen
▮ Résoudre numériquement un système d'équations transcendant

```
In[5]:= FindRoot[{Sin[x]==x-y,Cos[y]==x+y},{x,1},{y,0}]
```

$$\text{Out[5]}= \{x \to 0.883401,\ y \to 0.1105\}$$

Differentialgleichung lösen
▮ Résoudre une équation différentielle

```
In[6]:= DSolve[y''[x]-k y[x]==1,y[x],x]
```

$$\text{Out[6]}= \left\{\left\{y[x] \to -\frac{1}{k} + e^{\sqrt{k}\,x}\,C[1] + e^{-\sqrt{k}\,x}\,C[2]\right\}\right\}$$

Differentialgleichung numerisch lösen und Graph ausgeben
▮ Résoudre numériquement une équation différentielle et sortir le graphe

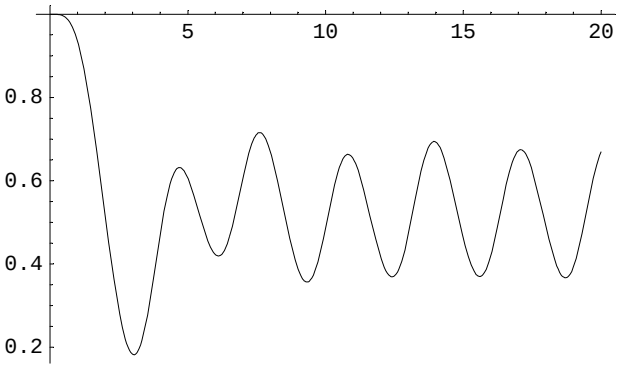
Gleichung lösen
▮ Résoudre équation

```
In[7]:= NDSolve[{y''[x] + Sin[x]^2 y'[x] + y[x] == Cos[x]^2,
                 y[0]==1, y'[0]==0},
                 y, {x,0,20}]
```

```
Out[7]= {{y → InterpolatingFunction[{{0., 20.}}, <>]}}
```

Graph ausgeben
▮ Sortir un graphe

```
In[8]:= Plot[Evaluate[y[x]/.%],{x,0,20}];
```



"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[9]:= (* Old Form: Remove["Global`@*"] *)
```

```
In[10]:= Remove["Global`*"]
```

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

5. Listen
▮ Listes

Beispiel: Liste machen und manipulieren
▮ Exemple: Dresser une liste et la manipuler

Liste von Fakultäten
▮ Liste de factoriaux

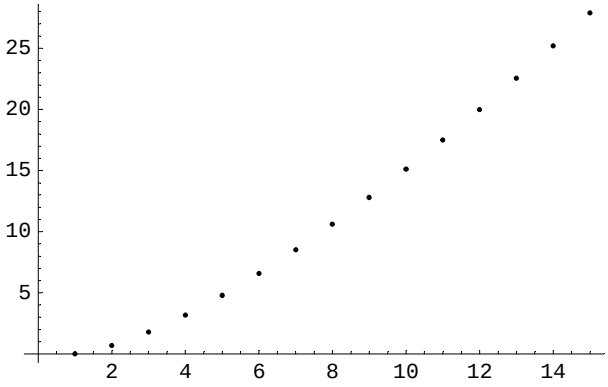
```
In[1]:= Table[n!,{n,1,15}]

Out[1]= {1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800,
         39916800, 479001600, 6227020800, 87178291200, 1307674368000}
```

Logarithmus davon
▮ Logarithme

```
In[2]:= N[Log[%]]

Out[2]= {0., 0.693147, 1.79176, 3.17805, 4.78749, 6.57925, 8.52516,
         10.6046, 12.8018, 15.1044, 17.5023, 19.9872, 22.5522, 25.1912, 27.8993}
```

Liste graphisch darstellen
▮ Faire un graphique de la liste

```
In[3]:= ListPlot[%];
```

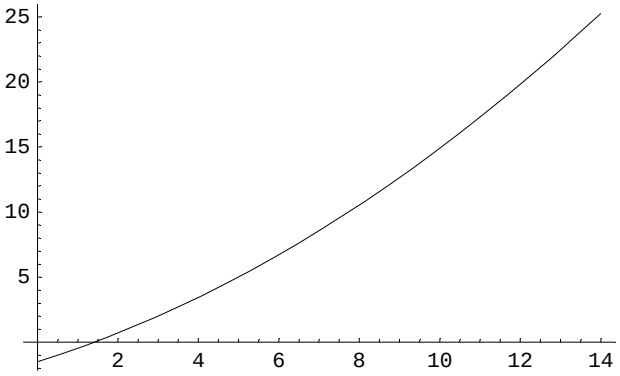


Optimale Parabel durch letzten Graphen legen
▮ Poser une parabole optimale à travers la liste

```
In[4]:= Fit[%%,{1,x,x^2},x]
```

$$\text{Out[4]= } -1.48508 + 0.963131\,x + 0.06766\,x^2$$

```
In[5]:= Plot[%,{x,0,14}];
```



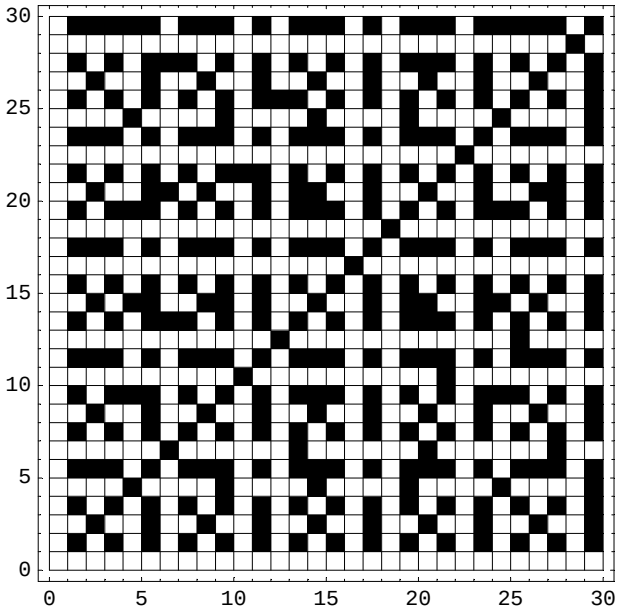
Beispiel: 2-dimensionales Array manipulieren
▮ Exemple: manipuler un array bidimensionnel

Array kreieren: Teilerfremde Paare von andern unterscheiden, Zahlen nicht ausgeben (zu viele)
▮ Créer un array: Distinguer des paires sans facteur commun d'autres paires, ne pas sortir les nombres (il y en a trop)

```
In[6]:= arr = Table[If[GCD[i,j]==1,1,0],{i,30},{j,30}];
```

Dichte-Graph
▮ Graphique de la densité

In[7]:= **ListDensityPlot[arr]**



Out[7]= **-** DensityGraphics **-**

Grösster absoluter Wert in der Fourier-Transformierten von `arr`
▮ Plus grande valeur absolue dans la transformée de Fourier de `arr`

In[8]:= **??Fourier**

```
Fourier[list] finds the discrete Fourier transform of a list of complex numbers.
```
Mehr...

```
Attributes[Fourier] = {Protected}

Options[Fourier] = {FourierParameters → {0, 1}}
```

In[9]:= **Fourier[{1, 1, 1, 1, -1, -1, -1, -1}]**

Out[9]=
```
{0. + 0. ⅈ, 0.707107 + 1.70711 ⅈ, 0. + 0. ⅈ, 0.707107 + 0.292893 ⅈ,
 0. + 0. ⅈ, 0.707107 - 0.292893 ⅈ, 0. + 0. ⅈ, 0.707107 - 1.70711 ⅈ}
```

In[10]:= **Max[Abs[Fourier[{1, 1, 1, 1, -1, -1, -1, -1}]]]**

Out[10]= 1.84776

In[11]:= **Fourier[{1, 1, 1, 1, -1, -1, -1, -1}] // N**

Out[11]=
```
{0. + 0. ⅈ, 0.707107 + 1.70711 ⅈ, 0. + 0. ⅈ, 0.707107 + 0.292893 ⅈ,
 0. + 0. ⅈ, 0.707107 - 0.292893 ⅈ, 0. + 0. ⅈ, 0.707107 - 1.70711 ⅈ}
```

In[12]:= **Max[Abs[N[Fourier[{1, 1, 1, 1, -1, -1, -1, -1}]]]]**

Out[12]= 1.84776

In[13]:= **Max[Abs[N[Fourier[arr]]]]**

Out[13]= 18.5

Frage: Was ist von der Ausgabe des Resultats zu halten?
Question: Que faut-il penser de la sortie du résultat?

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

In[14]:= **(* Old Form: Remove["Global`@*"] *)**

In[15]:= **Remove["Global`*"]**

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

6. Matrizen
▮ Matrices

Beispiel: Matrix eingeben und damit rechnen
▮ Exemple: Entrer une matrice et calculer avec elle

Matrix generieren
▮ Générer une matrice

```
In[1]:= m = Table[1/(i+j+1),{i,3},{j,3}]
```

$$\text{Out[1]}= \left\{\left\{\frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right\}, \left\{\frac{1}{4}, \frac{1}{5}, \frac{1}{6}\right\}, \left\{\frac{1}{5}, \frac{1}{6}, \frac{1}{7}\right\}\right\}$$

Anschauen
▮ Considérer

```
In[2]:= MatrixForm[m]
```

Out[2]//MatrixForm=

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

Inverse von `m`
▮ Inverse de `m`

In[3]:= **Inverse[m]**

Out[3]= {{300, −900, 630}, {−900, 2880, −2100}, {630, −2100, 1575}}

Anschauen
▮ Considérer

In[4]:= **MatrixForm[Inverse[m]]**

Out[4]//MatrixForm=

$$\begin{pmatrix} 300 & -900 & 630 \\ -900 & 2880 & -2100 \\ 630 & -2100 & 1575 \end{pmatrix}$$

Matrix mit Inverser multiplizieren, anschauen (Einheitsmatrix)
▮ Multiplier la matrice avec l'inverse, considérer (matrice unité - unifiée)

In[5]:= **MatrixForm[m.Inverse[m]]**

Out[5]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Neue Matrix mit `m` kreieren
▮ Créer une nouvelle matrice avec `m`

In[6]:= **mNeu = m - x IdentityMatrix[3]**

Out[6]= $\left\{\left\{\frac{1}{3}-x, \frac{1}{4}, \frac{1}{5}\right\}, \left\{\frac{1}{4}, \frac{1}{5}-x, \frac{1}{6}\right\}, \left\{\frac{1}{5}, \frac{1}{6}, \frac{1}{7}-x\right\}\right\}$

Anschauen
▮ Considérer

In[7]:= **MatrixForm[mNeu]**

Out[7]//MatrixForm=

$$\begin{pmatrix} \frac{1}{3}-x & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{4} & \frac{1}{5}-x & \frac{1}{6} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7}-x \end{pmatrix}$$

Determinante von mNeu
▮ Déterminante de mNeu

In[8]:= **Det[mNeu]**

Out[8]= $\frac{1}{378000} - \frac{317\,x}{25200} + \frac{71\,x^2}{105} - x^3$

Eigenwerte von m
▮ Valeurs propres de m

In[9]:= **Eigenvalues[m]**

Out[9]=
```
{Root[-1 + 4755 #1 - 255600 #1^2 + 378000 #1^3 &, 3],
 Root[-1 + 4755 #1 - 255600 #1^2 + 378000 #1^3 &, 2],
 Root[-1 + 4755 #1 - 255600 #1^2 + 378000 #1^3 &, 1]}
```

In[10]:= **N[%]**

Out[10]= {0.657051, 0.0189263, 0.000212737}

Eigenwerte numerisch
▮ Valeurs propres numériques

In[11]:= **N[%] // Chop**

Out[11]= {0.657051, 0.0189263, 0.000212737}

Beispiel: Rechnen mit einer grossen Matrix
▮ Exemple: Calculer avec une grande matrice

Zufällige 100 x 100-Matrix kreieren, nicht ausgeben
▮ Créer une matrice 100 x 100 fortuite, ne pas la sortir

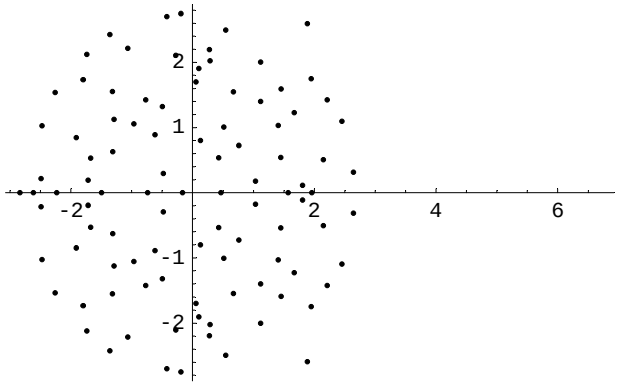
In[12]:= **n = Table[Random[],{100},{100}];**

Davon Eigenwerte (komplex), nicht ausgeben
▮ En calculer les valeurs propres (complexes), ne pas les sortir

In[13]:= **Eigenvalues[%];**

Eigenwerte als Punkte in der komplexen Ebene graphisch darstellen
▮ Représenter graphiquement les valeurs propres comme points dans le plan complexe

In[14]:= **ListPlot[Transpose[{Re[%],Im[%]}]]**



Out[14]= **-** Graphics **-**

Beispiel: Rechnen mit symbolischen Matrizen
▮ Exemple: Calculer avec des matrices symboliques

Matrix geben
▮ Donner une matrice

In[15]:= **s = {{a,b},{-b,2a}}; MatrixForm[s]**

Out[15]//MatrixForm=

$$\begin{pmatrix} a & b \\ -b & 2\,a \end{pmatrix}$$

Davon Eigenvektoren, algebraisch vereinfacht
▮ En donner les vecteurs propres, simplifiés algébraiquement

In[16]:= **Simplify[Eigenvectors[s]]**

Out[16]= $\left\{\left\{\frac{a+\sqrt{a^2-4\,b^2}}{2\,b}, 1\right\}, \left\{\frac{a-\sqrt{a^2-4\,b^2}}{2\,b}, 1\right\}\right\}$

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[17]:= **(* Old Form: Remove["Global`@*"] *)**

In[18]:= **Remove["Global`*"]**

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

7. Transformationsregeln und Definitionen
▮ Règles de transformation et définitions

Ersetzungsregeln
▮ Règles de remplacement

Beispiel: Regel zur Ersetzung von x in einem Ausdruck y durch 1+a
▮ Exemple: Règle pour remplacer x dans une expression y par 1 + a

```
In[1]:= y = 1 + x^2 + 3 x^3
```

$$1 + x^2 + 3\,x^3$$

```
In[2]:= y /. x->1+a
```

$$1 + (1 + a)^2 + 3\,(1 + a)^3$$

Beispiel: In einem Ausdruck f[2] ersetzen durch b
▮ Exemple: Dans une expression remplacer f[2] par b

```
In[3]:= {f[1],f[2],f[3]} /. f[2]->b
```

```
Out[3]= {f[1], b, f[3]}
```

Beispiel: In einem Ausdruck f[n] ersetzen durch n^2 (n^2)
▮ Exemple: Dans une expression remplacer f[n] par n^2 (n^2)

```
In[4]:= {f[1],f[2],f[3]} /. f[n_]->n^2
```

```
Out[4]= {1, 4, 9}
```

Definition einer Funktion
▮ Définition d'une fonction

Definition : f[n] sei n^2(n^2)
▮ Définition : f[n] soit n^2(n^2)

```
In[5]:= f[n_]:= n^2;
        (*Ausgabe auf dem Schirm: ■ Imprimer sur l'écran: *)
        f[n]
```

```
Out[7]= n^2
```

Dann rechnen mit Funktionswerten
▮ Calculer ensuite avec des valeurs fonctionelles

```
In[8]:= f[3] + f[a+b]
```

```
Out[8]= 9 + (a + b)^2
```

Rekursive Definition einer Funktion (Fakultäten)
▮ Définition récursive d'une fonction (factoriaux)

Definition
▮ Définition

```
In[9]:= fac[n_]:= n fac[n-1]
```

Initialisation
▮ Initialisation

```
In[10]:= fac[1]:= 1
```

Abfrage
▮ Interrogation

In[11]:= **?fac**

```
Global`fac

fac[1] := 1

fac[n_] := n fac[n - 1]
```

Anwendung: Berechnung von 25!
▮ Application : Calcul de 25!

In[12]:= **fac[25]**

Out[12]= 15511210043330985984000000

Bsp.: Eingabe der Rechenregeln für den Logarithmus
▮ Ex.: Entrer des règles de calcul pour le logarithme

Definition
▮ Définition

In[13]:= **log[x_ y_]:= log[x] + log[y]**

```
General::spell1 :
 Possible spelling error: new symbol name "log" is similar to existing symbol "Log". Mehr…
```

Anwendung: Berechnung von log[a b c d]
▮ Application: Calcul de log[a b c d]

In[14]:= **log[a b c d]**

Out[14]= log[a] + log[b] + log[c] + log[d]

Bsp.: Ausschaltung der Regel, dass bei einer Definition links vom Gleichheitszeichen kein Operationszeichen zwischen Funktionen stehen darf
▮ Bsp.: Elimination de la règle que dans une définition à gauche du signe égal il est interdit de placer un signe d'opération entre des fonctions

Versuch einer Definition
▮ Essai de définition

```
In[15]:= g[i_] + g[j_] := g[i+j]

         SetDelayed::write : Tag Plus in g[i_] + g[j_] is Protected. Mehr…

Out[15]= $Failed
```

Anders versuchen ● Essayer différamment

Richtige Definition so:
▮ Définition correcte ainsi;

```
In[16]:= g/: g[i_] + g[j_] := g[i+j]
```

Anwendung auf eine Berechnung
▮ Application pour un calcul

```
In[17]:= Clear[y];
         g[x] + g[y] + g[z]

Out[18]= g[x + y + z]
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[19]:= (* Old Form: Remove["Global`@*"] *)

In[20]:= Remove["Global`*"]
```

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

8. Symbolisches Rechnen

Manipulation einer Liste
▮ Manipulation d'une liste

Permutationen
▮ Permutations

```
In[1]:= Permutations[{a,b,c}]

Out[1]= {{a, b, c}, {a, c, b}, {b, a, c}, {b, c, a}, {c, a, b}, {c, b, a}}
```

Aufhebung der Untergruppen
Abolir les sous-groupes

```
In[2]:= Flatten[%]

Out[2]= {a, b, c, a, c, b, b, a, c, b, c, a, c, a, b, c, b, a}
```

Position von b angeben
▮ Donner la position de b

```
In[3]:= Position[%,b]

Out[3]= {{2}, {6}, {7}, {10}, {15}, {17}}
```

Letzte Liste aufmultiplizieren
▮ Multiplier la dernière liste

In[4]:=
```
FoldList[Times,{1},%]
```

Out[4]=
```
{{1}, {2}, {12}, {84}, {840}, {12600}, {214200}}
```

Geschachtelte Listen
▮ Listes emboîtées

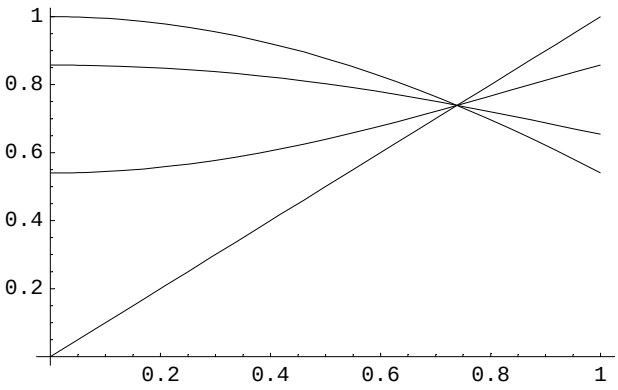
Definition der Liste
▮ Définition de la liste

In[5]:=
```
NestList[Cos,x,3]
```

Out[5]=
```
{x, Cos[x], Cos[Cos[x]], Cos[Cos[Cos[x]]]}
```

Graph dieser Funktionen
▮ Graphiques de ces fonctions

In[6]:=
```
Plot[Evaluate[%],{x,0,1}];
```



Anwenden einer Funktion simultan auf verschiedene Elemente
▮ Applications d'une fonction simultanément à différents éléments

Beispiel
▮ Exemple

In[7]:=
```
Clear[f];
Map[f,{a,b,c,d}]
```

Out[8]=
```
{f[a], f[b], f[c], f[d]}
```

Beispiel mit namenlosen Funktionen (mit "Function")
▮ Exemple avec des fonctions sans nom (avec "Function")

```
In[9]:= Map[Function[x,1+x^2],{a,b,c,d}]
```

Out[9]= $\{1+a^2, 1+b^2, 1+c^2, 1+d^2\}$

Bsp.: Rekursive Berechnung der Fibonacci-Folge, Verfolgung des Rechenwegs
▮ Bsp.: Calcul récursif de la suite de Fibonacci, poursuite de la voie de calcul

Definition
▮ Définition

```
In[10]:= Clear[f];
         f[0]=f[1]=1;
         f[n_]:=f[n-1]+f[n-2]
```

Anwendung: Berechnung f[4]
▮ Application: Calcul de f[4]

```
In[13]:= f[4]
```

Out[13]= 5

Rechenweg verfolgen
▮ Poursuivre la voie de calcul

```
In[14]:= Trace[f[4],f[_]]
```

Out[14]= {f[4], {f[3], {f[2], {f[1]}, {f[0]}}, {f[1]}}, {f[2], {f[1]}, {f[0]}}}

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[15]:= (* Old Form: Remove["Global`@*"] *)
```

```
In[16]:= Remove["Global`*"]
```

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

9. Programmierung
▮ Programmation

Programm zur Erzeugung einer nxn-Hilbert-Matrix
▮ Programme pour la création d'une matrice nxn de Hilbert

Programm
▮ Programme

```
In[1]:= Hilbert[n_]:=Table[1/(i+j-1),{i,n},{j,n}]
```

Aufruf für eine 3x3-Matrix
▮ Appel pour une matrice 3 x 3

```
In[2]:= Hilbert[3]; MatrixForm[%]

Out[2]//MatrixForm=
        Null
```

Programm zur Erzeugung des charakteristischen Polynoms einer Matrix
Programme pour la création du polynome caractéristique d'une matrice

Programm
Programme

```
In[3]:= CharPoly[m_,x_]:=
        Det[m-x IdentityMatrix[Length[m]]]/; MatrixQ[m]
         (* Check auf Matrix bei m *)
```

Wieso /; ?
Pourquoi /; ?

```
In[5]:= ?/;
```

```
patt /; test is a pattern which matches only if the evaluation of test yields True. lhs :>
   rhs /; test represents a rule which applies only if the evaluation of test yields
   True. lhs := rhs /; test is a definition to be used only if test yields True. Mehr...
```

Anwendung
Application

```
In[6]:= m = {{1,2,3},{2,3,4},{3,4,5}};
        Print[MatrixForm[m]];
        CharPoly[m,x]
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}$$

```
Out[8]= 6 x + 9 x^2 - x^3
```

```
In[9]:= m = {1,2,3,2,3,4,3,4,5};
        Print[MatrixForm[m]];
        CharPoly[m,x]
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 3 \\ 4 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$

```
Out[11]= CharPoly[{1, 2, 3, 2, 3, 4, 3, 4, 5}, x]
```

Programm zur Auffindung der nächsten Primzahl

Programme pour trouver le prochain nombre premier

Programm

Programme

```
In[12]:= NextPrime[n_Integer]:=
         Module[{k=n},
                 While[!PrimeQ[k],k++];
                 Return[k]
               ]
```

Anwendung auf n =111

Application pour n = 111

```
In[13]:= NextPrime[111]

Out[13]= 113
```

Programm zur Berechnung statistischer Grössen, ohne Loops zu verwenden (elegant)

▮ Programme pour calculer des valeurs statistiques, sans utiliser des loops (élégant)

Programm
▮ Programme

In[14]:=
```
mean[list_List]    :=Apply[Plus,list] / Length[list];

variance[list_List]:=
Mean[(list-Mean[list])^2] Length[list]/(Length[list]-1);

quantile[list_List,q_]:=
   Part[Sort[list],-Floor[-q Length[list]]]/; 0<q<1;

Alles[list_, q_]:=Module[{},
        Print["Mean = ",mean[list]," = ",
        Mean[list]];
        Print["Variance = ",variance[list]," = ",
        Variance[list]];
        Print["Quantile = ",quantile[list,q]," = ",
        Quantile[list,q]];
                        ]
```

```
General::spell1 :
 Possible spelling error: new symbol name "mean" is similar to existing symbol "Mean". Mehr…

General::spell1 :
 Possible spelling error: new symbol name "variance" is similar to existing symbol "Variance". Mehr…

General::spell1 :
 Possible spelling error: new symbol name "quantile" is similar to existing symbol "Quantile". Mehr…
```

In[19]:=
```
??Module
```

```
Module[{x, y, ... }, expr] specifies that occurrences
   of the symbols x, y, ...  in expr should be treated as local. Module[
   {x = x0, ... }, expr] defines initial values for x, ... . Mehr…

Attributes[Module] = {HoldAll, Protected}
```

Anwendung
▮ Application

In[20]:=
```
l={1,2,2,3,3,3,4,4,4,5,5,6,7,7,8,9}; q=0.1;
Alles[l,q]
```

$$\text{Mean} = \frac{73}{16} = \frac{73}{16}$$

$$\text{Variance} = \frac{1279}{240} = \frac{1279}{240}$$

$$\text{Quantile} = 2 = 2$$

Programm "Zufallsspaziergang" - was macht dieses Programm?

Programme "promenade au hasard" - que fait ce programme?

Programm
Programme

```
In[22]:= zs[n_Integer]:=
           FoldList[Plus,0,Table[Random[]-1/2,{n}]]
```

Anwendung
Application

```
In[23]:=
         zs[100]

Out[23]= {0, 0.397161, 0.774418, 0.552511, 0.608016, 0.882651, 0.783349, 0.931527,
          1.1167, 1.60128, 1.49378, 1.11713, 0.690229, 0.635201, 0.961627, 1.22891,
          1.14827, 0.72646, 0.716078, 0.572787, 0.196796, -0.170223, -0.245881,
          -0.034506, 0.118632, -0.145548, -0.0984632, -0.165181, -0.567548, -0.606363,
          -0.959977, -0.674872, -0.762409, -0.785804, -0.531922, -0.370169, -0.530801,
          -0.999169, -0.571712, -0.177244, 0.242764, 0.696207, 0.634045, 0.671805,
          0.967804, 1.28827, 0.801763, 1.12815, 0.771008, 0.855649, 0.822062, 0.715164,
          0.260393, -0.116152, -0.296126, -0.188128, -0.055363, 0.0914873, 0.157632,
          0.603876, 0.397274, 0.512492, 0.65118, 0.202956, 0.0763466, 0.238122,
          -0.061029, -0.0470129, 0.0303786, 0.371692, 0.0590442, 0.246676, 0.181206,
          -0.0621206, 0.158818, -0.0466518, -0.15735, -0.524132, -0.62322, -0.436688,
          -0.180151, -0.193784, 0.140984, 0.381272, 0.34441, 0.71556, 0.411639,
          0.600152, 0.1899, -0.100726, 0.394504, 0.0690005, 0.081357, -0.0505823,
          0.257296, 0.24416, -0.178013, -0.566626, -0.979687, -1.28735, -1.09883}
```

Programm, das die ersten n Terme in der Kettenbruchentwicklung einer Zahl findet: elegante funktionale Programmierung

Programme qui trouve les n premiers termes d'un nombre

Programm

Programme

In[24]:=
```
?ContinuedFraction
```

```
ContinuedFraction[x, n] generates a list of the first n terms in
   the continued fraction representation of x. ContinuedFraction[x] generates
   a list of all terms that can be obtained given the precision of x. Mehr...
```

In[25]:=
```
ContinuedFraction[h, 30]
```

Out[25]=
```
ContinuedFraction[h, 30]
```

In[26]:=
```
ownContinuedFraction[x_Real,n_Integer]:=
      Floor[NestList[Function[{u},1/(u-Floor[u])],
                     x,n-1]]
```

Anwendung: Goldener Schnitt in Kettenbruchentwicklung

Application: Section d'or en évolution de fractions continues

In[27]:=
```
h=N[(Sqrt[5]-1)/2,100];
Print[h]
ownContinuedFraction[h,30]
```

```
0.6180339887498948482045868343656381177203091798057628621354486227052604628189024497072072041
893911375
```

Out[29]=
```
{0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

Anwendung: e in Kettenbruchentwicklung

Application: e en évolution de fractions continues

In[30]:=
```
e=N[E,100];
Print[e];
ownContinuedFraction[e,30]
```

```
2.7182818284590452353602874713526624977572470936999595749669676277240766303535475945713821785
25166427
```

Out[32]=
```
{2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1,
 10, 1, 1, 12, 1, 1, 14, 1, 1, 16, 1, 1, 18, 1, 1, 20}
```

```
In[33]:= ContinuedFraction[e, 30]

Out[33]= {2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1,
          10, 1, 1, 12, 1, 1, 14, 1, 1, 16, 1, 1, 18, 1, 1, 20}
```

Anwendung: Pi in Kettenbruchentwicklung
▮ Application: Pi en évolution de fractions continues

```
In[34]:=
         p=N[Pi,100];
         Print[p]
         ContinuedFraction[p,30]

        3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534
        42117068

Out[36]= {3, 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14,
          2, 1, 1, 2, 2, 2, 2, 1, 84, 2, 1, 1, 15, 3, 13, 1, 4}
```

Programm mit Mix aus symbolischer Konstruktion und mathematischen Operationen: Ableiten der Funktionen einer Liste nach allen Variablen einer andern Liste

▮ Programme avec mélange de construction symbolique et opérations mathématiques: Dériver les fonctions d'une liste d'après toutes les variables d'une autre liste

Programm
▮ Programme

```
In[37]:= Jac[funs_List,vars_List]:=Outer[D, funs, vars]
```

Anwendung
▮ Application

```
In[38]:= f={x^2+x y-1, Cos[y]-z, a x z};
         v={x,y,z};
         Jac[f,v]

Out[40]= {{2 x + y, x, 0}, {0, -Sin[y], -1}, {a z, 0, a x}}
```

Liste von Regeln für Laplace-Transformationen
▮ Liste de règles pour des transformations de

Laplace

Programm (Regeln)
▮ Programme (règles)

```
In[41]:= Laplace[c_,t_,s_]:=c/s/; FreeQ[c,t];
         Laplace[a_+b_,t_,s_]:=
                     Laplace[a,t,s]+Laplace[b,t,s];
         Laplace[c_ a_,t_,s_]:=
                     c Laplace[a,t,s]/; FreeQ[c,t];
         Laplace[t_^n_.,t_,s_]:=
                     n!/s^(n+1)/; (FreeQ[n,t]&&n>0);
         Laplace[a_. Exp[b_.+c_. t_],t_,s_]:=
                 Laplace[a Exp[b],t,s-c]/; FreeQ[{b,c},t];
```

```
In[46]:= ?_.

         p:v is a pattern object which represents an expression
            of the form p, which, if omitted, should be replaced by v. Mehr...
```

```
In[47]:= ?FreeQ

         FreeQ[expr, form] yields True if no subexpression in expr
            matches form, and yields False otherwise. FreeQ[expr, form, levelspec]
            tests only those parts of expr on levels specified by levelspec. Mehr...
```

Anwendung
▮ Application

```
In[48]:= Laplace[E^t+4t^2-2t+1,t,s]
```

$$\text{Out[48]}= \frac{1}{-1+s}+\frac{8}{s^3}-\frac{2}{s^2}+\frac{1}{s}$$

Programm, das pattern matching benutzt um die Anzahl gleicher Elemente einer Liste auszuzählen - kurz, elegant, effizient!
▮ Programme qui utilise le "pattern matching" pour compter le nombre d'éléments égaux d'une liste - bref, élégant, efficace!

Programm
▮ Programme

```
In[49]:= RunEncode[{rest___Integer,same:(n_Integer)..}]:=
             Append[RunEncode[{rest}],{n,Length[{same}]}];
         RunEncode[{}]:={}
```

Anwendung
▮ Application

```
In[51]:= r={1,2,2,2,3,3,3,3,3,3,4,4,5,6,6,6,7,8,9,9,9,10};
         RunEncode[r]

Out[52]= {{1, 1}, {2, 3}, {3, 7}, {4, 2}, {5, 1}, {6, 3}, {7, 1}, {8, 1}, {9, 3}, {10, 1}}
```

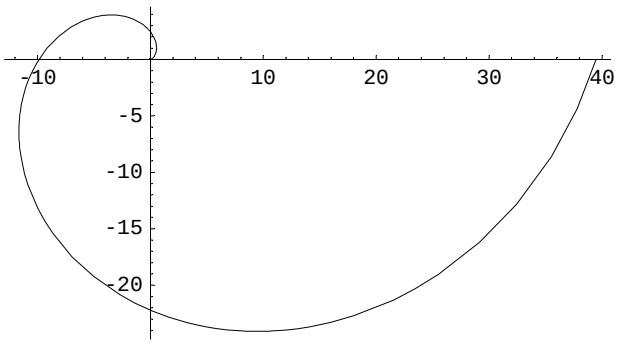
Programm, um Graphik zu erzeugen
▮ Programme pour créer des graphiques

Programm
▮ Programme

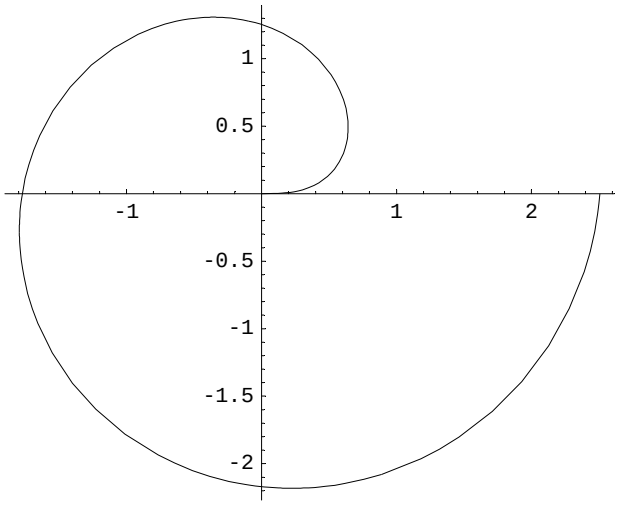
```
In[53]:= PolarPlot[r_,{t_,tmin_,tmax_}]:=
             ParametricPlot[{r Cos[t], r Sin[t]},
                  {t,tmin,tmax},AspectRatio->Automatic]
```

Anwendungen
▮ Applications

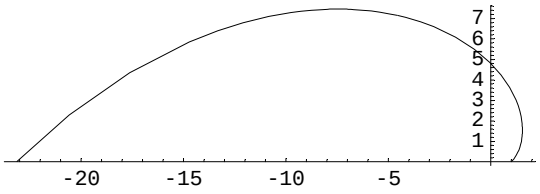
```
In[54]:= PolarPlot[t^2,{t,0,2 Pi}];
```



In[55]:= **PolarPlot[t^(1/2),{t,0,2 Pi}];**



In[56]:= **PolarPlot[E^t,{t,0,Pi}];**



Programm, das die Lösungen einer Polynom-Gleichung als Punkte der komplexen Ebene darstellt

Programme qui représente les solutions d'une équation polynomiale comme points d'un plan complexe

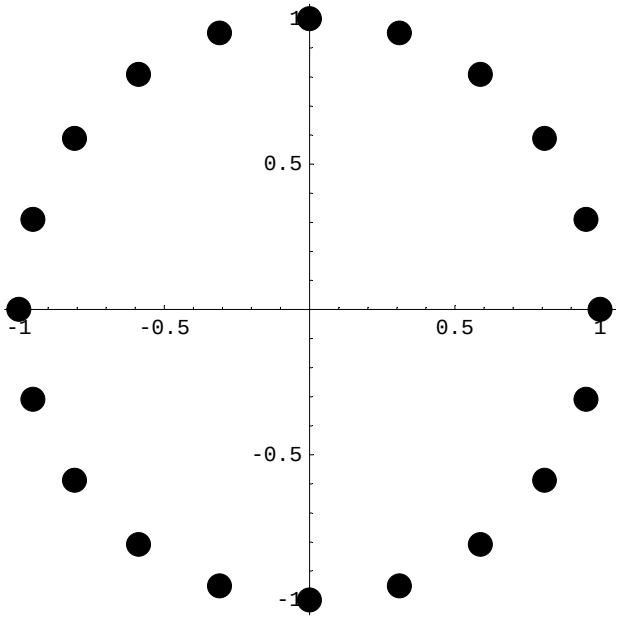
Programm

Programme

In[57]:=
```
RootPlot[poly_,z_]:=
    ListPlot[{Re[z],Im[z]}/. NSolve[poly==0,z],
    AspectRatio->Automatic,
        PlotStyle->{PointSize[0.04]}]/;
                PolynomialQ[poly,z]
```

Anwendung: Einheitswurzeln
▮ Application: Racines unifiées (de l'unité)

In[58]:= **RootPlot[z^20 -1,z];**



Programm, das Zahlen in Matrixform aus einem File liest und als 3D-Graphik darstellt
▮ Programme qui lit des nombres en forme de matrice dans un fichier et les représente en graphique 3D

Programm
▮ Programme

```
In[59]:= myfile[myfile_String]:=
             ReadList[myfile,Number,RecordLists->True]

In[60]:= FilePlot3D[myfile_String]:=
             ListPlot3D[ReadList[myfile,Number,
                                 RecordLists->True]]
```

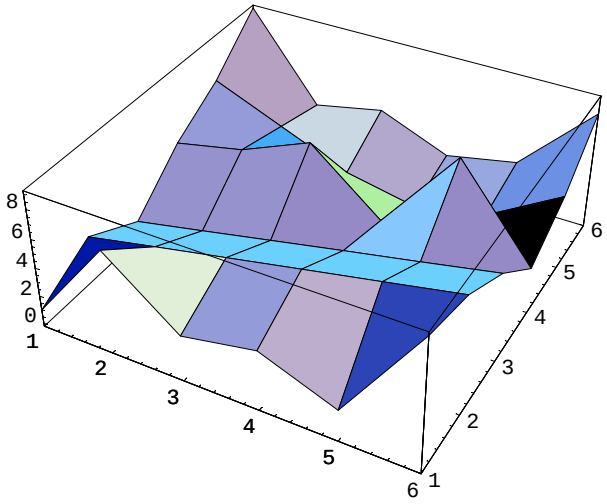
Anwendung
▮ Application

```
In[61]:= myFilePath = "C:\work\MathematicaData\AADATEN"

Out[61]= C:\work\MathematicaData\AADATEN
```

Hallo!!!! >>> Eigenes Directory/ Verzeichnis/ Dossier setzen!!!!!
• Hallo!!!! >>> Remplacer par le propre directoire!!!!!

In[62]:=
```
Print[myfile[myFilePath]];
FilePlot3D[myFilePath]
```

```
{{1, 7, 3, 4, 2, 9}, {3, 4, 5, 6, 7, 8}, {1, 2, 3, 4, 5, 6},
 {4, 5, 7, 3, 9, 3}, {6, 4, 2, 1, 2, 5}, {9, 3, 4, 2, 3, 8}}
```



Out[63]= - SurfaceGraphics -

Anhang
▮ Annexe

In[64]:=
```
?RecordLists
```

```
RecordLists is an option for ReadList which specifies whether objects
   from separate records should be returned in separate sublists. Mehr...
```

In[65]:=
```
myFilePath = "C:\work\MathematicaData\AADATEN"
```

Out[65]= C:\work\MathematicaData\AADATEN

Hallo!!!! >>> Eigenes Directory/ Verzeichnis/ Dossier setzen!!!!!
• Hallo!!!! >>> Remplacer par le propre directoire!!!!!

In[66]:=
```
!! C:\work\MathematicaData\AADATEN
```

```
1 7 3 4 2 9
3 4 5 6 7 8
1 2 3 4 5 6
4 5 7 3 9 3
6 4 2 1 2 5
9 3 4 2 3 8
```

In[67]:=
```
<< C:\work\MathematicaData\AADATEN
```

Out[67]= 5184

In[68]:=
```
Get["C:\work\MathematicaData\AADATEN"]
```

Out[68]= 5184

Was ist jetzt passiert? ● Qu'est-ce qui est arrivé?

```
In[69]:= ??<<
```

```
<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr...

Attributes[Get] = {Protected}

Options[Get] = {CharacterEncoding:→ $CharacterEncoding, Path:→ $Path}
```

Programm, das externe Daten manipuliert - hier: das Files sucht mit gegebenen Teilstrings

▮ Programme qui manipule des données externes - Ici: le fichier cherche par des stings partiels donnés

Programm

▮ Programme

```
In[70]:= ?Select
```

```
Select[list, crit] picks out all elements ei of list for which crit[ei] is True. Select[
   list, crit, n] picks out the first n elements for which crit[ei] is True. Mehr...
```

```
In[71]:= myNewFilePath = "C:\work\MathematicaData\AADATEN"

Out[71]= C:\work\MathematicaData\AADATEN
```

Hallo!!!! >>> Eigenes Directory/ Verzeichnis/ Dossier setzen!!!!!
● Hallo!!!! >>> Remplacer par le propre directoire!!!!!

```
In[72]:= fn=FileNames[]>> C:\work\MathematicaData\fnxxx;
         fnxxx=Flatten[ReadList[myNewFilePath,
                         Expression,RecordLists->True,
                         WordSeparators->{" "}]];
         Print[fnxxx];
         Print[fnxxx[[1]]];
         Print[fnxxx[[3]]];
         Clear[Where];
         Where[s_String_]:=
            Select[fnxxx,(Length[FindList[#,s,1]]>0)&]
```

```
{1512, 20160, 720, 11340, 480, 5184}

1512

720
```

Anwendung: Suche nach *e* im momentanen directory
▮ Application: Cherche *e* dans le directoire momentané

In[79]:= **Where["e"]**

Out[79]= Where[e]

Wo liegt also das Problem hier? ▮ Où est le problème ici?

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[80]:= **(* Old Form: Remove["Global`@*"] *)**

In[81]:= **Remove["Global`*"]**

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

10. *Mathematica* Packages
▮ *Mathematica* Packages

Beispiel: Laplace-Transformationen
▮ Exemple: Transformations de Laplace

Package laden
▮ Charger le Package

```
(* Old versions:
<<Calculus`LaplaceTransform` *)
```

Eine Transformierte rechnen
▮ Calculer une transformée

```
LaplaceTransform[t^n Exp[-c/t],t,s]
```

$$2 \left(\frac{s}{c}\right)^{\frac{1}{2}(-1-n)} \text{BesselK}\left[-1-n, \frac{2\sqrt{s}}{\sqrt{\frac{1}{c}}}\right]$$

Beispiel: Padé-Approximation
Exemple: Approximation de Padé

Package laden (neuer Kern benutzen!)
Charger le Package (employer un nouveau kernel!)

```
<<Calculus`Pade`
```

Eine Approximation rechnen
Calculer une approximation

```
Pade[Exp[Sin[x]],{x, 0, 2, 3}]
```

$$\frac{1 + \frac{8\,x}{15} + \frac{x^2}{4}}{1 - \frac{7\,x}{15} + \frac{13\,x^2}{60} + \frac{x^3}{60}}$$

Beispiel aus der Statistik
Exemple de la statistique

Package laden
Charger le Package

```
<<Statistics`DescriptiveStatistics`
```

Einige Kenngrössen rechnen
Calculer quelques valeurs spécifiques

```
LocationReport[Table[Random[],{1000}]]
```

```
{Mean → 0.496287, HarmonicMean → 0.143913, Median → 0.51231}
```

Beispiel: Permutationen
Exemple: Permutations

Package laden
Charger le Package

```
(* Old versions: <<DiscreteMath`Permutations` *)

<< DiscreteMath`Combinatorica`
```

Zufällige Permutation erzeugen
▮ Créer une permutation par le hazard

```
RandomPermutation[20]
```

```
{12, 16, 5, 20, 11, 1, 17, 9, 8, 3, 10, 18, 4, 13, 7, 6, 14, 19, 2, 15}
```

Zyklische Zerlegung
▮ Décomposition cyclique

```
ToCycles[%]
```

```
{{12, 18, 19, 2, 16, 6, 1}, {5, 11, 10, 3}, {20, 15, 7, 17, 14, 13, 4}, {9, 8}}
```

Beispiel aus der Chemie
▮ Exemple pris de la chimie

Package laden
▮ Charger le Package

```
<<Miscellaneous`ChemicalElements`
```

```
Titanium::shdw : Symbol Titanium appears in multiple contexts
   {Miscellaneous`ChemicalElements`, Graphics`Colors`}; definitions in context
   Miscellaneous`ChemicalElements` may shadow or be shadowed by other definitions. Mehr…

Cobalt::shdw : Symbol Cobalt appears in multiple contexts
   {Miscellaneous`ChemicalElements`, Graphics`Colors`}; definitions in context
   Miscellaneous`ChemicalElements` may shadow or be shadowed by other definitions. Mehr…

Zinc::shdw :
 Symbol Zinc appears in multiple contexts {Miscellaneous`ChemicalElements`, Graphics`Colors`}; definitions
   in context Miscellaneous`ChemicalElements` may shadow or be shadowed by other definitions. Mehr…

Gold::shdw :
 Symbol Gold appears in multiple contexts {Miscellaneous`ChemicalElements`, Graphics`Colors`}; definitions
   in context Miscellaneous`ChemicalElements` may shadow or be shadowed by other definitions. Mehr…
```

Ein Atomgewicht abrufen
▮ Rappeler un poids atomique

```
?AtomicWeight
```

```
AtomicWeight[element] gives the atomic weight of the specified element. Mehr…
```

```
AtomicWeight[Tungsten]
```

```
183.84
```

Noch eines
▮ Encore un

```
AtomicWeight[Plutonium]
```

```
AtomicWeight::unstable : No stable isotope of Plutonium exists.

244
```

Ausschalten der Ausgabe "unstabil"
▮ Eliminer la sortie "instable"

```
Off[AtomicWeight::unstable]
```

Atomnummer
▮ Nombre atomique

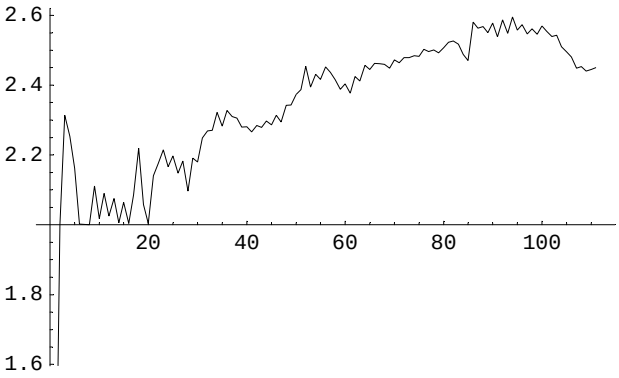
```
AtomicNumber[Plutonium]
```

```
94
```

Verhältnis Gewicht zu Nummer
▮ Corrélation entre poids et nombre

```
ListPlot[AtomicWeight[Elements]/AtomicNumber[
        Elements], PlotJoined->True];
```

```
AtomicWeight::unknown : The atomic weight of Ununbium is not known.

Graphics::gptn : Coordinate Unknown/112 in {112, Unknown/112} is not a floating-point number. Mehr…
```



Beispiel: Sternform eines Platonischen Körpers
Exemple: Forme d'étoile d'un corps platonique

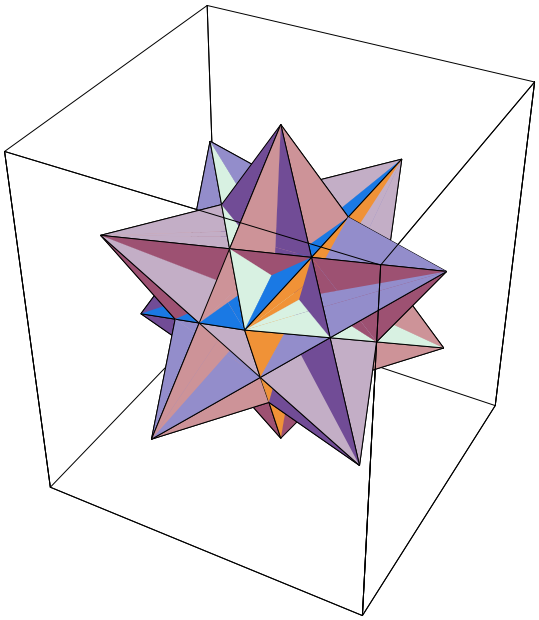
Package laden
Charger le Package

```
<<Graphics`Polyhedra`
```

```
Vertices::shdw :
 Symbol Vertices appears in multiple contexts {Geometry`Polytopes`, DiscreteMath`Combinatorica`};
   definitions in context Geometry`Polytopes` may shadow or be shadowed by other definitions. Mehr…
```

Sternform des Ikosaeders
Forme d'étoile de l'icosaèdre

```
Show[Polyhedron[GreatIcosahedron]];
```



"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

```
(* Old Form: Remove["Global`@*"] *)
```

```
Remove["Global`*"]
```

Rundgang in *Mathematica*

▮ Tour en *Mathematica*

(Nach Ideen aus: Handbuch "*Mathematica*" von S. Wolfram)
▮ (Selon les idées prises dans le manuel "*Mathematica*" de S. Wolfram)

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

11. Datenaustausch mit *Mathematica*, andere Ausgabeformen
Achtung: Einige Befehle sind für UNIX, andere für DOS, andere für Windows...
▮ Echange de données par *Mathematica,* autres formes de sorties
Attention: Quelques ordres sont pour UNIX, d'autres pour DOS, d'autres pour Windows...

Beispiel Ausgabeformen
▮ Exemple de formes de sorties

Beispiel eines betrachteten Ausdrucks, Standardausgabe
▮ Exemple d'une expression considérée, output standardisé

```
In[1]:= (* Old Form: Remove["Global`@*"] *)

In[2]:= (* Remove["Global`*"] *)
```

In[3]:= **c=(a^2+b^2)/(x+y)^5**

Out[3]=
$$\frac{a^2 + b^2}{(x + y)^5}$$

Ausgabe in der Eingabeform
▮ Output en forme d'entrée

In[4]:= **InputForm[c]**

Out[4]//InputForm=
```
(a^2 + b^2)/(x + y)^5
```

Ausgabe in TeX-Form (Profi-Satzsystem)
▮ Output en forme TeX (système de composition professionnel)

In[5]:= **TeXForm[c]**

Out[5]//TeXForm=
```
\frac{a^2+b^2}{(x+y)^5}
```

Ausgabe in Fortran-Form
▮ Output en forme Fortran

In[6]:= **FortranForm[c]**

Out[6]//FortranForm=
```
(a**2 + b**2)/(x + y)**5
```

Ausgabe in C-Form
▮ Output en forme C

In[7]:= **CForm[c]**

Out[7]//CForm=
```
(Power(a,2) + Power(b,2))/Power(x + y,5)
```

Beispiel: Kommunikation mit der Umgebung
Exemple: Communication avec l'entourage

Externe Daten lesen. (Files kreiert mit "A_ReadMeFirst.nb")
(file "C:\WORK\AAAdaten.txt")
Lire de données externes (dossiers). (Fichiers crées avec "A_ReadMeFirst.nb")

In[8]:=
```
$Path
```

Out[8]=
```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Applications,
 ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

In[9]:=
```
(*
 Achtung:
  Der folgende Befehl darf nur einmal laufen gelassen werden. Sonst
   wird der Path auf work zweimal in die Pfadvariable geschrieben,
 was zu Systemabstürzen führen kann!!!!!!!!!!

  Attention: Ne faire marcher l'ordre suivant qu'une fois. Si non le Path
    qui pointe sur work est écrit deux fois dans la variable du Path,
 ce qui peut bloquer le système!!!!!!!!!!!!
*)
```

In[10]:=
```
$Path = Join[$Path, {"C:\Data (Local)\DATEN"}]
```

Out[10]=
```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
   Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel,
 C:\Data (Local)\DATEN}
```

In[11]:=
```
$Path = Join[$Path, {"C:\work\MathematicaData"}]
```

Out[11]=
```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
   Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel,
 C:\Data (Local)\DATEN, C:\work\MathematicaData}
```

In[12]:=
```
$Path
```

Out[12]=
```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
   Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel,
 C:\Data (Local)\DATEN, C:\work\MathematicaData}
```

Versuche:● Essayer:

In[13]:=
```
!cd C : \work\MathematicaData
```

```
Syntax::sntxf : "\"\!\(C :\)\" cannot be followed by \"\!\(\(\(\\
   work\)\)\\MathematicaData\)\".\!\(\"\"\) \!\(\*ButtonBox[\"Mehr…\", ButtonStyle->
   \"RefGuideLinkText\", ButtonFrame->None, ButtonData:>\"General::sntxf\"]\)"
```

Was klemmt? Versuche (Befehl generieren, Syntax beachten!):

● Pourquoi ça na va pas? Essayer (créer les ordres, tenir compte de la syntaxe):

Versuche:● Essayer:

In[14]:=
```
!cd "C:\work\MathematicaData";
ReadList["!dir", String]
```

Out[15]=
```
{ Datentr?ger in Laufwerk C: ist 53_01_02 Software,
  Volumeseriennummer: 847D-377A, ,
  Verzeichnis von C:\Programme\Wolfram Research\Mathematica\5.2, ,
 02.10.2007  22:33    <DIR>          ., 02.10.2007  22:33    <DIR>          ..,
 21.06.2005  02:03                 8 .CreationID,
 02.10.2007  20:59                20 AAADaten,
 02.10.2007  21:07                90 AAAfib,
 02.10.2007  22:31               219 AAKurs15.m,
 25.09.2005  12:06    <DIR>          AddOns,
 25.09.2005  12:04    <DIR>          Configuration,
 25.09.2005  12:04    <DIR>          Documentation,
 20.06.2005  22:13            65'536 math.exe,
 20.06.2005  20:10           110'592 Mathematica.exe,
 20.06.2005  22:13           106'496 MathKernel.exe,
 13.12.2006  21:35               209 myNewFilePath,
 02.10.2007  22:40                66 Statisti.m,
 13.12.2006  18:37               184 String[myNewFilePath],
 25.09.2005  12:06    <DIR>          SystemFiles,
 05.06.2007  14:27    <DIR>          von Hand gespeichert,
 11.05.2006  19:20    <DIR>          Zusaetze,
               10 Datei(en)        283'420 Bytes,
                8 Verzeichnis(se), 53'140'295'680 Bytes frei}
```

```
In[16]:= (* Old Form:
            !cd C:\work\MathematicaData;
          ReadList["!dir",String]
          *)

In[17]:= ReadList["Daten.txt", String]

Out[17]= {kjlkfjlkjlkjdsa, 920389047903,
           jlkdjf lkjf 983u40 u9034, jkjf 4734908 334, 4384983084903, 898908}
```

Was ist passiert? ● Qu'est-ce qui c'est passé?

Aeltere Version oder andere Systeme ● Vieille version ou autres systèmes

```
In[18]:= (* !cd c:\\work;
          ReadList["\!\(\*
          StyleBox[\"C\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\":\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"...\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"...\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\".\",\nEvaluatable->False,\nAspectRatioFixed->True,
              \nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"AAAdaten\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\".\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\"txt\",\nEvaluatable->False,\nAspectRatioFixed->True]\)",String] *)
```

Aeltere Version oder andere Systeme ● Vieille version ou autres systèmes

```
In[19]:= (* ReadList["\!\(\*
          StyleBox[\"C\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\":\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"\\\\\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"work\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"\\\\\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\)\!\(\*
          StyleBox[\"AAAdaten\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\".\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\"txt\",\nEvaluatable->False,\nAspectRatioFixed->True]\)",String] *)
```

Aeltere Version oder andere Systeme ● Vieille version ou autres systèmes

```
In[20]:= (* ReadList["\!\(\*
          StyleBox[\"C\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\":\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\).....\!\(\*
          StyleBox[\"AAAdaten\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\".\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\"txt\",\nEvaluatable->False,\nAspectRatioFixed->True]\)",
            String, RecordLists->True] *)
```

Aeltere Version oder andere Systeme ● Vieille version ou autres systèmes

```
In[21]:= (* !cd C:\work;
          ReadList["\!\(\*
          StyleBox[\"C\",\nEvaluatable->False,\nAspectRatioFixed->True]\)\!\(\*
          StyleBox[\":\",\nEvaluatable->False,\nAspectRatioFixed->
              True,\nFontFamily->\"Courier\",\nFontSize->12]\).....\!\(\*
          StyleBox[\"Adaten\",\nEvaluatable->False,\nAspectRatioFixed->True]\)",String] *)

In[22]:= !cd

In[23]:= data = "a b c d e f 1 2 3 4 5 6 1.1 1.2. 1.3 aa ab ac"

Out[23]= a b c d e f 1 2 3 4 5 6 1.1 1.2. 1.3 aa ab ac

In[24]:= data >> C:\work\MathematicaData\ABdata

In[25]:= ReadList["C:\work\MathematicaData\ABdata", String]

Out[25]= {"a b c d e f 1 2 3 4 5 6 1.1 1.2. 1.3 aa ab ac"}

In[26]:= data = {1, 2, 3, 4, 5, 6, 7, 8, 9}

Out[26]= {1, 2, 3, 4, 5, 6, 7, 8, 9}

In[27]:= data >> C:\work\MathematicaData\ACdata

In[28]:= ReadList["C:\work\MathematicaData\ACdata", String]

Out[28]= {{1, 2, 3, 4, 5, 6, 7, 8, 9}}

In[29]:= ReadList["C:\work\MathematicaData\ACdata", Number]

        Read::readn : Syntax error reading a real number from C:\work\MathematicaData\ACdata. Mehr…

Out[29]= {$Failed}

In[30]:= data = 123456789

Out[30]= 123456789

In[31]:= data >> C:\work\MathematicaData\ADdata

In[32]:= ReadList["C:\work\MathematicaData\ADdata", Number]

Out[32]= {123456789}
```

Externer Befehl wie "square" etc. ausführen und dann Daten einlesen (Funktioniert nicht mit "square" , da Befehl nicht vorhanden)
▮ Exécuter un ordre externe tel que "square" etc. et lire les données ensuite (ne fonctionne pas avec "square", car l'ordre n'est pas disponible)

Die restlichen Befehle sind UNIX-Befehle. "cd": Directory-Wechsel, "ls": Directory lesen.
▮ Les autres ordres sont des ordres UNIX. "cd": Changer le directoire, "ls": Lire le directoire

```
In[33]:= (* ReadList["!square",Number] *)
```

```
In[34]:= (* ReadList["!ls",String] *)

In[35]:= (* ReadList["!cd C:/work;ls",String] *)

In[36]:= (* !cd work;
         ReadList["!dir",String] *)

In[37]:= (* ReadList["!cd C:\work\AAAdaten.txt;dir",String] *)

In[38]:= (* !cd *)
```

Externer Befehl "iterat" in *Mathematica* einbinden (Funktioniert nicht, da Befehl nicht vorhanden)
▮ Relire l'ordre externe "iterat" à *Mathematica* (Ne fonctionne pas parce que l'ordre n'est pas présent)

Aeltere Version oder andere Systeme, Programm fehlt
- Vieille version ou autres systèmes, programme manque

```
In[39]:= (* Install["iterat"] *)

In[40]:= (* getdata["C:\work\AAAdaten.txt",8] *)
```

Nach String "1" im File "AAAdaten" suchen
- Chercher d'après String "l" lans le fichier "AAAdaten"

```
In[41]:= FindList["C:\work\MathematicaData\Daten.TXT","1"]

         Syntax::stresc : Unknown string escape \w. Mehr…

         Syntax::stresc : Unknown string escape \M. Mehr…

         Syntax::stresc : Unknown string escape \D. Mehr…

Out[41]= {}

In[42]:= (* Format false *)

In[43]:= data = "4 5 6 7 8 9 \n 1 2 3 "

Out[43]= 4 5 6 7 8 9
          1 2 3

In[44]:= data >> C:\work\MathematicaData\AEdata.txt

In[45]:= FindList["C:\work\MathematicaData\AEdata.txt", "1"]

Out[45]= {"4 5 6 7 8 9 \n 1 2 3 "}
```

String ersetzen
▮ Remplacer String

```
In[46]:= StringReplace[%,"7"->"a"]

Out[46]= {"4 5 6 a 8 9 \n 1 2 3 "}
```

Suchen in externen Verzeichnissen
▮ Cherger dans des listes externes

```
In[47]:= Directory[]
Out[47]= C:\Programme\Wolfram Research\Mathematica\5.2
```

Das folgende Directory heisst: ● Le directoire suivant s'appelle:
"C:\PROGRAMME\WOLFRAM RESEARCH\MATHEMATICA\4.0\work"

```
In[48]:= CreateDirectory["C:\work\MathematicaData\Hallo"]
Out[48]= C:\work\MathematicaData\Hallo

In[49]:= SetDirectory["C:\work\MathematicaData\Hallo"]
Out[49]= C:\work\MathematicaData\Hallo

In[50]:= SetDirectory["C:\Programme\Wolfram Research\Mathematica\5.2"]
Out[50]= C:\Programme\Wolfram Research\Mathematica\5.2

In[51]:= ?*Directory*
```

System`

```
CopyDirectory     $AddOnsDirectory
CreateDirectory   $BaseDirectory
DeleteDirectory   $HomeDirectory
Directory         $InitialDirectory
DirectoryName     $InstallationDirectory
DirectoryStack    $LaunchDirectory
HomeDirectory     $PreferencesDirectory
NotebookDirectory $RootDirectory
ParentDirectory   $TopDirectory
RenameDirectory   $UserAddOnsDirectory
ResetDirectory    $UserBaseDirectory
SetDirectory
```

In[52]:=
```
?*File*
```

System`

```
ContextToFilename FileNameDialogSettings
ContextToFileName FileNames
CopyFile          FileType
DeleteFile        GetFileName
EndOfFile         IncludeFileExtension
File              RenameFile
FileByteCount     SetFileDate
FileDate          SetFileLoadingContext
FileFormat        ToFileName
FileInformation   $PasswordFile
FileName
```

In[53]:=
```
FileNames[]
```

Out[53]=
```
{AAADaten, AAAfib, AAKurs15.m, AddOns, Configuration, .CreationID, Documentation,
 Mathematica.exe, math.exe, MathKernel.exe, myNewFilePath, Statisti.m,
 String[myNewFilePath], SystemFiles, von Hand gespeichert, Zusaetze}
```

In[54]:=
```
(* Old: !cd C:\\work;
   ReadList["!dir",String]*)
```

In[55]:=
```
! CD "C:\work\MathematicaData";
FileNames[ ]
```

Out[56]=
```
{AAADaten, AAAfib, AAKurs15.m, AddOns, Configuration, .CreationID, Documentation,
 Mathematica.exe, math.exe, MathKernel.exe, myNewFilePath, Statisti.m,
 String[myNewFilePath], SystemFiles, von Hand gespeichert, Zusaetze}
```

In[57]:=
```
!CD C:\work\MathematicaData;
FileNames[  ]
```

Out[58]=
```
{AAADaten, AAAfib, AAKurs15.m, AddOns, Configuration, .CreationID, Documentation,
 Mathematica.exe, math.exe, MathKernel.exe, myNewFilePath, Statisti.m,
 String[myNewFilePath], SystemFiles, von Hand gespeichert, Zusaetze}
```

In[59]:=
```
FileNames[ "*dd*" ]
```

Out[59]=
```
{AddOns}
```

```
In[60]:= SetDirectory["C:\\work\MathematicaData"];
         FileNames[  ]

         Syntax::stresc : Unknown string escape \M. Mehr…

Out[61]= {AAA1TeX.mtex, AAA1TeX.tex, AAA1TeX.txt, AAA2TeX.mtex, AAA2TeX.tex, AAA3TeX.mtex,
           AAA.C, AAADaten, AAA_F, AAAfib, AAAfib1, AAAfib.txt, AAA.MC, AAA.MTX, AAA.mx,
           AA_ASCII, AAA.TEX, AAATeX.mtex, AAATeX.tex, AAA.x, AAA.xls, AABDaten, AACDaten,
           AADATEN, AADATEN0, AADATEN1, AADATEN2, AADATEN3, AADATEN4, AADATEN5, AADATEN.AC,
           AAEDaten.txt, AA_f, AAF, AAFDaten.txt, AAFIB, AAGDaten.txt, AAHDaten, AAKURS15.M,
           AAneuDaten, AAneuDaten1, AAneuDaten2, AASESSIO.MA, ABdata, ACdata, ADdata,
           AEdata.txt, ASession.nb, DATAOUTS, DATAOUTS.MA, DATAREC, DATAREC.MA, DATEN.DOC,
           DATEN.TXT, fib2, fnxxx, Hallo, hallo.jpg, NEUDATEN, OutputBsp.bmp, postsc_1.ma,
           postsc_1.nb, postsc_1.ps, postsc_2.ps, postsc_3.ps, postscri.ma, postscri.nb,
           postscri.ps, PSTEST1.MA, PSTEST1.PS, PSTEST_2.PS, PSTEST_3.PS, PSTEST.MA,
           PSTEST.PS, ReserveIndex.htm, SPSHEET0, SPSHEET1, SPSHEET2, STATISTI.M, TEX}
```

Aeltere Version oder andere Systeme ● Vieille version ou autres systèmes

```
In[62]:= ?FindList

         FindList["file", "text"] gives a list of lines in the file that contain the specified
            string. FindList["file", {"text1", "text2", ... }] gives a list of all lines
            that contain any of the specified strings. FindList[{"file1", ... }, ... ]
            gives a list of lines containing the specified strings in any of the filei.
            FindList[files, text, n] includes only the first n lines found. Mehr...

In[63]:= (* Old: FindList[FileNames[ ],"AddOns"] *)

In[64]:= (* !cd C:\\ WORK; *)

In[65]:= (* !cd C;!cd work; *)

In[66]:= (* !cd C;!cd work;
            FindList["!dir","AAA"] *)

In[67]:= (* FindList["!cd C:\EigeneMathWorkData\Daten;!dir","AAA.x"] *)

In[68]:= (* FileNames[ "*AAA*",{"c:\work"} ] *)

In[69]:= (* FindList["C:\work\*","AAA"] *)
```

Aeltere Version resp. UNIX ● Vieille version resp. UNIX

```
In[70]:= (* FindList["!cd C:\work;ls","A"] *)

In[71]:= (* FindList["!cd C:\work;ls","AA"] *)

In[72]:= (* FindList["!cd C:\work;ls","AAA"] *)

In[73]:= (* FindList["!cd C:\work;ls","AAA_"] *)
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[74]:= (* Old Form: Remove["Global`@*"] *)
```

In[75]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

1. Einstieg in *Mathematica*
▮ Premier contact avec *Mathematica*

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 1 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 1.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

1.1. Maschinenspezifisches
▮ Problèmes spécifiques de la machine

Aufgabe ▮ Problème

Finde heraus, was von *Mathematica* und was von der Maschine abhängt!
▮ Distingue ce qui dépend de *Mathematica* et ce qui dépend de la machine.

Hinweis: Wenn Du einmal nicht weiterkommst, so frage den Kursleiter oder konsultiere die Literatur, z.B. "***Mathematica***" von Wolfram.
▮ **Indication:** Si tu as un problème, adresse-toi au professeur ou consulte la littérature, p.ex. "***Mathematica***" de Wolfram.

1.2. Versionspezifisches
▮ Problèmes spécifiques de la version

Mit welcher *Mathematica*-Version arbeitest Du? Drücke die Enter-Taste in der Zelle mit "$Version".
▮ Avec quelle version de *Mathematica* traveilles-tu? Appuie sur la touche "entrer" dans la cellule avec "$Version".

```
In[1]:= $Version
```

```
Out[1]= 5.2 for Microsoft Windows (June 20, 2005)
```

1.3. Front-End versus Kernel
▮ Front-End versus Kernel

Aufgabe: ▮ Problème:

1. Finde heraus, was "Front-End" und was "Kernel" bedeutet.
▮ Que siginifie "Front-End" et "Kernel"?
2. Was bewirkt die Enter-Taste?
▮ Que produit la touche "Enter"?
3. Was bewirkt "Shift-Return"?
▮ Que produit la touche ""Shift-Return"?
4. Was bewirkt "Command"?
▮ Que produit la touche "Command"?
5. Was bewirken die Icons "Action" - "Interrupt" etc.?
▮ Que produisent les Icons "Action" - "Interrupt" etc.?
6. Wie ist es auf andern Maschinen?
▮ Comment cela se passe-t-il sur d'autres machines?

1.4. Notation
▮ Notation

Aufgabe ▮ Problème

Probiere aus: ▮ Essaie:

```
In[2]:= Apply[Plus, {a,b,c,d}]
```

```
Out[2]= a + b + c + d
```

Wie ist es mit Gross- und Kleinschreibung?
Wie ist es mit den verschiedenen Klammerarten?
Wie ist es mit den "blanks"?

Wie ist es mit der Interpunktion?
Wie ist es mit dem Fond? (Fixe und variable Abstände, z. B. Courier contra Times)?
■
Pose-toi des Questions aux sujets suivants:
Majuscule - minuscule,
parenthèses,
"blanks",
ponctuation,
"fond" (distances foxes et variables, p. ex. Courier contra Times).

1.5. *Mathematica* interaktiv benutzen: ■ Utiliser *Mathematica* interactivement:

Aufgabe ■ Problème

Probiere aus: ■ Essaie:

Potenzieren: ■ Calculer les puissances:

```
In[3]:= 5^10
Out[3]= 9765625
```

Resultat der letzten Zelle verwenden: ■ Utiliser le résultat de la dernière cellule:

```
In[4]:= %^(1/10)
Out[4]= 5

In[5]:= % + a
Out[5]= 5 + a

In[6]:= %3^(1/10)
Out[6]= 5

In[7]:= %%% + b
Out[7]= 5 + b
```

Namen vergeben: ■ Donner des noms:

```
In[8]:= Schifflänge = 3
Out[8]= 3

In[9]:= Schifflänge
Out[9]= 3
```

```
In[10]:= 20 Schifflänge
Out[10]= 60
In[11]:= 2 m + 3 m
Out[11]= 5 m
```

1.6. Reservierte Namen:
▮ Réserve des noms:

Mathematica hat sehr viele Befehle (Funktionen etc.) eingebaut.
Merke: *Mathematica*-Namen beginnen alle mit Grossbuchstaben.
▮ Dans *Mathematica* beaucoup d'ordres (fonctions etc.) sont incorporés.
Retenez: Les noms de *Mathematica* commencent tous par des majuscules.

Aufgabe ▮ Problème:

Probiere aus: ▮ Essaie:

```
In[12]:= Abs[-8]
Out[12]= 8
In[13]:= Cos[Pi]
Out[13]= -1
In[14]:= D[x^2-3x+4,x]
Out[14]= -3 + 2 x
In[15]:= M = {{1,2},{5,1}}; MatrixForm[M]
Out[15]//MatrixForm=
        (1  2)
        (5  1)
In[16]:= Det[M]
Out[16]= -9
In[17]:= GCD[48005, 100098630]
Out[17]= 5
```

1.7. Help-Funktion und Kommando-Vervollständigung: ▮ Fonction Help et complètement du commandement:

Mathematica hat mehr als 700 Funktionen eingebaut, die über das Help-Service abfragbar sind. "*" figuriert dabei als "wild card". Help ist Front End-spezifisch.
Weitere Info: *Mathematica* Quick Reference Guide, *Mathematica* Help Stack, Manual, spez. Literatur.
▮ Dans *Mathematica* sont incorporées plus de 700 fonctions qui sont disponibles par le service Help. "*" y figure comme "wild card". Help est spécifique de Front End. Autres Informations: Quick Reference Guide, *Mathematica* Help Stack, Manual, litt. spéc..

Aufgabe ▮ Problème

Probiere aus: ▮ Essaie:

```
In[18]:= ?Factor*
```

```
System`
Factor          FactorList
FactorComplete  FactorSquareFree
Factorial       FactorSquareFreeList
Factorial2      FactorTerms
FactorInteger   FactorTermsList
```

```
In[19]:= ?FactorInteger
```

```
FactorInteger[n] gives a list of the prime
   factors of the integer n, together with their exponents. Mehr…
```

```
In[20]:= FactorInteger[75]

Out[20]= {{3, 1}, {5, 2}}

In[21]:= FactorInteger[1122445667788]

Out[21]= {{2, 2}, {23, 1}, {12200496389, 1}}

In[22]:= ?Sin
```

```
Sin[z] gives the sine of z. Mehr…
```

In[23]:= **?*Sin***

System`

ArcSin	SingularityDepth
ArcSinh	SingularValueDecomposition
FourierSinTransform	SingularValueList
IncludeSingularTerm	SingularValues
InverseFourierSinTransform	Sinh
Sin	SinhIntegral
SingleLetterItalics	SinIntegral

In[24]:= **?Cos**

Cos[z] gives the cosine of z. Mehr...

In[25]:= **?Tan**

Tan[z] gives the tangent of z. Mehr...

In[26]:= **?Exp***

System`

Exp	ExponentPosition
Expand	ExponentStep
ExpandAll	Export
ExpandDenominator	ExportAutoReplacements
ExpandNumerator	ExportPacket
ExpIntegralE	ExportString
ExpIntegralEi	Expression
Exponent	ExpressionPacket
ExponentFunction	ExpToTrig

In[27]:= **?Log**

Log[z] gives the natural logarithm of z (logarithm to base e). Log[b, z] gives the logarithm to base b. Mehr...

In[28]:= **?Divisors**

Divisors[n] gives a list of the integers that divide n. Mehr...

In[29]:= **Divisors[5000]**

Out[29]= {1, 2, 4, 5, 8, 10, 20, 25, 40, 50, 100, 125, 200, 250, 500, 625, 1000, 1250, 2500, 5000}

In[30]:= **?*Graphic***

System`

ContourGraphics	Graphics3D	GraphicsSpacing
DensityGraphics	GraphicsArray	SoundAndGraphics
FullGraphics	GraphicsData	SurfaceGraphics
Graphics	GraphicsGrouping	

In[31]:= **?*Plot***

System`

```
ArrayPlot       MaxPlotPoints    PlotJoined
ContourPlot     ParametricPlot   PlotLabel
DensityPlot     ParametricPlot3D PlotPoints
ListContourPlot Plot             PlotRange
ListDensityPlot Plot3D           PlotRegion
ListPlot        Plot3Matrix      PlotStyle
ListPlot3D      PlotDivision
```

In[32]:= **??Plot**

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
   xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. Mehr...

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic,
  Compiled → True, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
  DisplayFunction :→ $DisplayFunction, Epilog → {}, FormatType :→ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
  ImageSize → Automatic, MaxBend → 10., PlotDivision → 30., PlotLabel → None,
  PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[33]:= **?@**

System`

```
$
```

Global`

```
a b c d m x
```

Rufe die Information ab zu folgenden Ausdrücken:
▮ Cherche l'information pour les expressions suivantes:

Beispiel: ▮ Exemple:

In[34]:= **?NIntegrate**

```
NIntegrate[f, {x, xmin, xmax}] gives a numerical approximation
   to the integral of f with respect to x from xmin to xmax. Mehr...
```

Integrate
ContourPlot
InterpolatingPolinomial
MapAt
Binomial
Eigenvalues
FindRoot

Integrate
Timing

Kommando vervollständigen
▮ Compléter le commandement

Probiere aus (Achtung, Front-End-aphängig!):
▮ Essaie (attention, peut dépendre du Front-End):

Gehe in die nächste Zelle und drücke "Command k":
▮ Entrer dans la cellule suivante et appuyer "Command k" :

```
In[35]:= Sor
Out[35]= Sor
```

Gehe in die nächste Zelle, markiere mit der Maus "So" und drücke "Command k":
▮ Entrer dans la cellule suivante, marquer à la souris "So" et presser "Command k":

```
In[36]:= Sor
Out[36]= Sor
```

Probiere aus:
▮ Essayer:

```
In[37]:= Sort[{2,5,3,1,8}]
Out[37]= {1, 2, 3, 5, 8}
```

Gehe ins Menu auf ("Action" (old),) "Prepare Input", "Complete Selection":
▮ Entrer dans le menu sur ("Action" (old),) "Prepare Input", "Complete Selection":

```
In[38]:= Sor
Out[38]= Sor
```

Gehe ins Menu auf ("Action" (old),) "Prepare Input", "Make Template":
▮ Entrer dans le menu sur ("Action" (old),) "Prepare Input", "Make Template":

```
In[39]:= Sor
Out[39]= Sor
```

Gehe in die nächste Zelle und drücke "Command I": (Old?)
▮ Entrer dans la cellule suivante et presser "Command I": (Old?)

```
In[40]:= Sor
Out[40]= Sor
```

Gehe in die nächste Zelle und drücke "Command I": (Old)
Markiere "ContourPlot" in der nächste Zelle und drücke "F1".
Markiere die nächste Zelle und drücke "Shift F1".

Entrer dans la cellule suivante et appuier "Command I": (Old)
Marque "ContourPlot" dans la cellule suivante et appuie "F1".
Marque la cellule suivante et appuie "Shift F1".

In[41]:= **ContourPlot**

Out[41]= ContourPlot

1.8. Help-Funktion und Kommando-Vervollständigung bei Zeichen: Fonction Help et complètement de commandement:

Was bedeuten die folgenden Zeichen? Que signifient les signes suivants?
+ - * / ^ ! < <= > >=

In[42]:= **?+**

x + y + z represents a sum of terms. Mehr…

In[43]:= **?-**

x - y is equivalent to x + (-1 * y). Mehr…

In[44]:= **?***

System`

```
Abort
AbortProtect
Above
Abs
AbsoluteDashing
AbsoluteOptions
AbsolutePointSize
AbsoluteThickness
AbsoluteTime
AbsoluteTiming
AccountingForm
Accuracy
AccuracyGoal
Active
ActiveItem
AddOnHelpPath
AddTo
AdjustmentBox
AdjustmentBoxOptions
After
AiryAi
AiryAiPrime
AiryBi
AiryBiPrime
AlgebraicRules
AlgebraicRulesData
Algebraics
Alias
AlignmentMarker
All
AllowInlineCells
AllowScriptLevelChange
Alternatives
AmbientLight
Analytic
AnchoredSearch
And
AnimationCycleOffset
AnimationCycleRepetitions
AnimationDirection
AnimationDisplayTime
Apart
ApartSquareFree
AppellF1
Append
AppendTo
Apply
ArcCos
ArcCosh
ArcCot
ArcCoth
ArcCsc
ArcCsch
ArcSec
ArcSech
ArcSin
ArcSinh
ArcTan
ArcTanh
Arg
ArgumentCountQ
ArithmeticGeometricMean
Array
ArrayDepth
```

Global`

```
a d Schifflänge
b m Sor
c M x
```

In[45]:= **?/**

```
x/y or Divide[x, y] is equivalent to x y^-1.
```
Mehr…

In[46]:= **?^**

```
x^y gives x to the power y.
```
Mehr…

In[47]:= **4!**

Out[47]= 24

In[48]:= **?!**

```
Information::notfound : Symbol ! not found.
```
Mehr…

In[49]:= **?Factorial**

```
n! gives the factorial of n.
```
Mehr…

In[50]:= **?!!**

```
n!! gives the double factorial of n.
```
Mehr…

In[51]:= **?<**

```
x < y yields True if x is determined to be less than y. x1 <
   x2 < x3 yields True if the xi form a strictly increasing sequence.
```
Mehr…

In[52]:= **?<=**

```
x <= y yields True if x is determined to be less than or equal to y.
   x1 <= x2 <= x3 yields True if the xi form a non-decreasing sequence.
```
Mehr…

In[53]:= **?>**

```
x > y yields True if x is determined to be greater than y. x1 >
   x2 > x3 yields True if the xi form a strictly decreasing sequence.
```
Mehr…

In[54]:= **?>=**

```
x >= y yields True if x is determined to be greater than or equal to y.
   x1 >= x2 >= x3 yields True if the xi form a non-increasing sequence.
```
Mehr…

Probiere aus: ■ Essaie:

In[55]:= **4 + 7**

Out[55]= 11

In[56]:= **4/2**

Out[56]= 2

In[57]:= **4/3**

Out[57]= $\frac{4}{3}$

In[58]:= **N[4/3]**

Out[58]= 1.33333

In[59]:= **3 < 6**

Out[59]= True

In[60]:= **?<**

x < y yields True if x is determined to be less than y. x1 < x2 < x3 yields True if the xi form a strictly increasing sequence. Mehr…

In[61]:= **Alias["<"]**

Out[61]= Less

In[62]:= **Alias["="]**

Out[62]= Set

Aufgabe ▮ Problème:

Wende "Alias" auf weitere mathematische Symbole an.
▮ Utilise "Alias" pour d'autres symboles mathématiques.

1.9. Die verschiedenen Klammertypen ▮ Les différents types de parenthèses

Aufgabe ▮ Problème

Beachte im Folgenden, sofern vorhanden, die Klammern. Was passiert jeweils?
▮ Observe dans ce qui suit les parenthèses, quand il y en a.
Que se passe-t-il chaque fois?

Probiere aus: ▮ Essaie:

In[63]:= **1 + 2 + 3**

Out[63]= 6

In[64]:= **1 + (2 + 3)**

Out[64]= 6

In[65]:= **1 / 2 - 5**

Out[65]= $-\frac{9}{2}$

In[66]:= **(1 + 2) * 3**

Out[66]= 9

In[67]:= **(1 + 2) 3**

Out[67]= 9

In[68]:= **1 / (2 - 5)**

Out[68]= $-\frac{1}{3}$

In[69]:= **?Divisors**

```
Divisors[n] gives a list of the integers that divide n. Mehr...
```

Argumente von Funktionen: ■ Arguments de fonctions

In[70]:= **Divisors[100]**

Out[70]= {1, 2, 4, 5, 10, 20, 25, 50, 100}

In[71]:= **?Random**

```
Random[ ] gives a uniformly distributed pseudorandom Real in the range
   0 to 1. Random[type, range] gives a pseudorandom number of the specified
   type, lying in the specified range. Possible types are: Integer, Real and
   Complex. The default range is 0 to 1. You can give the range {min, max}
   explicitly; a range specification of max is equivalent to {0, max}. Mehr...
```

In[72]:= **Random[ ]**

Out[72]= 0.764801

In[73]:= **Random[Integer]**

Out[73]= 0

In[74]:= **Random[Integer,{50,60}]**

Out[74]= 50

Mengen, Listen, Vektoren, Matrizen...:
■ Ensembles, listes, vecteurs, matrices...:

In[75]:= **{x, x^2, x^3}**

Out[75]= $\{x, x^2, x^3\}$

In[76]:= **{{a[1,1],a[1,2]},{a[2,1],a[2,2]}}**

Out[76]= {{a[1, 1], a[1, 2]}, {a[2, 1], a[2, 2]}}

In[77]:= **MatrixForm[{{a[1,1],a[1,2]},{a[2,1],a[2,2]}}]**

Out[77]//MatrixForm=

$$\begin{pmatrix} a[1,1] & a[1,2] \\ a[2,1] & a[2,2] \end{pmatrix}$$

In[78]:= **v={a,b,c}**

Out[78]= {a, b, c}

In[79]:= **v[[1]]**

Out[79]= a

In[80]:= **v[[2]]**

Out[80]= b

In[81]:= **m = {{1,2,3},{4,5,6}}**

Out[81]= {{1, 2, 3}, {4, 5, 6}}

In[82]:= **MatrixForm[m]**

Out[82]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

In[83]:= **m[[2]]**

Out[83]= {4, 5, 6}

In[84]:= **m[[2,1]]**

Out[84]= 4

Kommentare: ▮ Commentaires:

Probiere aus: ▮ Essaie:

In[85]:= **7 + 4 (* Das ist ein Kommentar *)**

Out[85]= 11

1.10. Packages
■ Packages

New in Help, Help Browser
Old:

Gewisse Befehle sind ausgelagert in sogenannten "Packages". (Vgl. in Menu: "Info", Open Function Browser" (old?) --> Help --_ Help.) Packages muss man erst laden, bevor man sie verwenden kann. (NeXT:" ` " ist Alt-Shift-n, vgl. Preferences.)
■ Certains ordres sont stockés à l'extérieur dans les "Packages". (Compare dans le Menu: "Info", Open Function Browser" (old?) --> Help --_ Help.) Il faut d'abort charger Packages, avant de pouvoir les utiliser. (NeXT:" ` " est Alt-Shift-n, voir Preferences.)

Probiere aus ■ Essaie:

```
In[86]:= ?Mean
```

```
Mean[list] gives the statistical mean of the elements in list. Mehr…
```

(Durch den Aufruf ist Mean jetzt definiert. ■ Par l'appel Mean est défini maintenant.)

```
In[87]:= Remove[Mean]
```

```
Remove::rmptc : Symbol Mean is Protected and cannot be removed. Mehr…
```

(Damit Mean neu definiert werden kann, muss das alte erst gelöscht werden. ■ Pour redéfinir Mean, il faut d'abord effacer la vieille version)

```
In[88]:= <<Statistics`DescriptiveStatistics`
```

```
In[89]:= ?Mean
```

```
Mean[list] gives the statistical mean of the elements in list. Mehr…
```

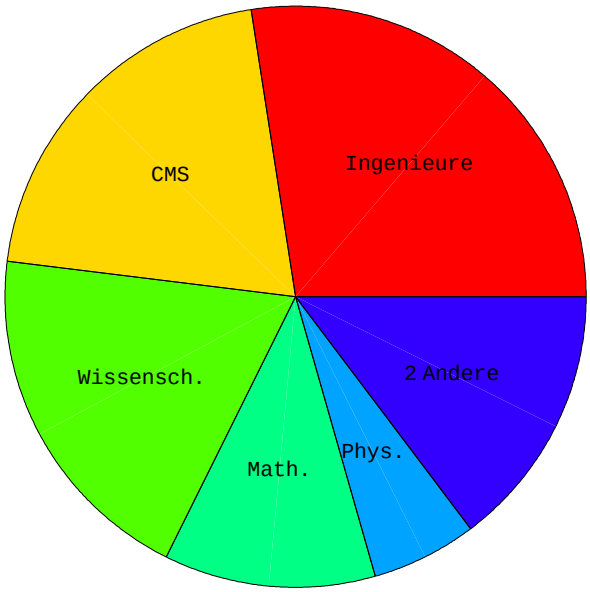
```
In[90]:= Mean[{1,2,3,4,5,6}]
```

$$\text{Out[90]=}\ \frac{7}{2}$$

Eine andere Art, ein Package zu laden:
■ Une autre manière de charger un Package:

```
In[91]:= Needs["Graphics`Graphics`"]
```

```
In[92]:= PieChart[{
                {28,"Ingenieure"},
                {21,"CMS"},
                {20,"Wissensch."},
                {12,"Math."},
                {6,"Phys."},
                {15,2"Andere"}
            }];
```



Aufgabe ▮ Problème

New in Help,Help Browser
Old:
Versuche Dich selbst im Laden von Packages mit Hilfe von Menu: ("Info", Open Function Browser" (old)) --> Help--Help.
▮ Essaie toi-même à charger des Packages à l'aide du menu: ("Info", Open Function Browser" (old)) --> Help--Help.

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[93]:= (* Old Form: Remove["Global`@*"] *)
```

```
In[94]:= Remove["Global`*"]
```

Uebungen ▮ Exercices

1. Einstieg in *Mathematica*

▮ Premier contact avec *Mathematica*

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 1 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 1.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

(a)

Finde alle Kommandos, die mit O beginnen ("Olga").
▮ Trouve tous les commandements qui commencent par O ("Olga").

In[1]:= **?O***

System`

O	Orange
OddQ	Order
Off	OrderedQ
Offset	Ordering
OLEData	Orderless
On	Oscillatory
OneIdentity	Out
OneStepRowReduction	Outer
Open	OutputAutoOverwrite
OpenAppend	OutputForm
OpenFunctionInspectorPacket	OutputFormData
OpenRead	OutputGrouping
OpenSpecialOptions	OutputMathEditExpression
OpenTemporary	OutputNamePacket
OpenWrite	OutputStream
Operate	OutputToOutputForm
OptimizationLevel	OutputToStandardForm
Optional	Over
OptionInspectorSettings	Overflow
OptionQ	Overlaps
Options	Overscript
OptionsPacket	OverscriptBox
OptionValueBox	OverscriptBoxOptions
OptionValueBoxOptions	OwnValues
Or	

(b)

Finde alle Kommandos, die das Teilwort "L i s t" enthalten.

Trouve tous les commandements qui contiennent le mot (partie de mot) " L i s t".

In[2]:= **?*List***

System`

```
BinaryReadList           ListInterpolation
ButtonStyleMenuListing   ListPlay
CoefficientList          ListPlot
CompletionsListPacket    ListPlot3D
ComposeList              ListQ
CounterStyleMenuListing  MessageList
FactorList               NestList
FactorSquareFreeList     NestWhileList
FactorTermsList          PowerModList
FindList                 ReadList
FixedPointList           RecordLists
FoldList                 ReplaceList
List                     SampledSoundList
Listable                 SingularValueList
ListContourPlot          StringReplaceList
ListConvolve             StyleMenuListing
ListCorrelate            TrigFactorList
ListDensityPlot          $MessageList
Listen
```

Aufgabe 2 ▮ Problème 2

Teste alle Funktionen Deines Front-Ends.

▮ Examine toutes les fonctions de ton Front-End

Aufgabe 3 ▮ Problème 3

Verwende "Alias[ ]", um die eingebaute Liste der Entsprechungen zu den Abkürzungen zu generieren.

▮ Utilise "Alias[ ]", por générer la liste incorporée contenant les correspondances des abréviations.

In[3]:= **Alias[ ]**

Out[3]=
```
System`Private`ValueList[" :> String, # :> Slot, % :> Out, & :> Function,
 * :> Times, + :> Plus, - :> Subtract, . :> Dot, / :> Divide, !! :> Factorial2,
 ; :> CompoundExpression, < :> Less, = :> Set, > :> Greater, ? :> PatternTest,
 != :> Unequal, ## :> SlotSequence, { :> List, | :> Alternatives, } :> List,
 ^ :> Power, _ :> Blank, && :> And, ** :> NonCommutativeMultiply, ++ :> Increment,
 *= :> TimesBy, += :> AddTo, -- :> Decrement, -= :> SubtractFrom, -> :> Rule,
 .. :> Repeated, /. :> ReplaceAll, /: :> TagSet, /; :> Condition, /= :> DivideBy,
 /@ :> Map, :: :> MessageName, := :> SetDelayed, :> :> RuleDelayed, << :> Get,
 <= :> LessEqual, <> :> StringJoin, =. :> Unset, == :> Equal, >= :> GreaterEqual,
 >> :> Put, @@ :> Apply, ___ :> BlankNullSequence, ^= :> UpSet, _. :> Optional,
 [[ :> Part, || :> Or, ]] :> Part, ~~ :> StringExpression, __ :> BlankSequence,
 ... :> RepeatedNull, //. :> ReplaceRepeated, //@ :> MapAll, ::= :> Alias,
 ::=. :> UnAlias, =!= :> UnsameQ, === :> SameQ, ^:= :> UpSetDelayed, >>> :> PutAppend]
```

Aufgabe 4 ▮ Problème 4

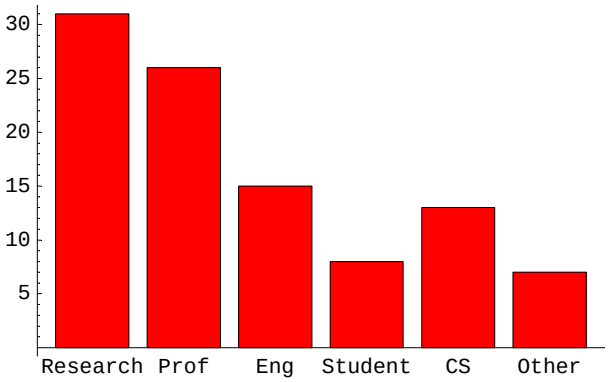
Werte die folgenden Ausdrücke aus:
▮Evalue les expressions suivantes:

```
In[4]:= 3 x + 7 x
Out[4]= 10 x

In[5]:= 2 Meter + 13 Meter
Out[5]= 15 Meter

In[6]:= 120 lbs (1 kg/(2.2 lbs))
Out[6]= 54.5455 kg
```

Aufgabe 5 ▮ Problème 5

Lade ("Graphics.m" (old)) "Graphics.nb" und mache ein Balkendiagramm, das *Mathematica*-Benützer auf Jobs aufgeteilt zeigt.
▮ Charge ("Graphics.m" (old)) "Graphics.nb" et fais un diagramme à barres qui divise les utilisateurs de *Mathematica* selon leurs emplois.

```
In[7]:= << Graphics`Graphics`;
        BarChart[{{31,"Research"},
                  {26,"Prof"},
                  {15,"Eng"},
                  {8,"Student"},
                  {13,"CS"},
                  {7,"Other"}}]
```




```
Out[8]= - Graphics -
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[9]:= (* Old Form: Remove["Global`@*"] *)
```

In[10]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

2. Numerische Probleme
▮ Problèmes numériques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 2 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 2.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Mathematica kennt ganze Zahlen (Integers), Fliesskommazahlen (mit Dezimalpunkt), rationale Zahlen, komplexe Zahlen sowie Zahlen, die durch Symbole gegeben sind (z.B. Pi).
▮ *Mathematica* connaît les nombres entiers (integers), les nombres avec virgule flottante (avec point décimal), les nombres rationaux, les nombres complexes ainsi que les nombres représentés par des symboles (p.ex. Pi).

2.1. Arithmetische Operationen
▮ Opérations arithmétiques

Probiere aus: ▮ Essaie:

```
In[1]:= 53 + 78
Out[1]= 131
```

Multiplikation: ▮ Multiplication:

```
In[2]:= 127*9721
Out[2]= 1234567

In[3]:= 127 9721
Out[3]= 1234567
```

Potenzieren: ■ Calculer les puissances:

```
In[4]:= 34^56

Out[4]= 57918773205287127842044254126179599852840968492056164062843692360166371779746690236416
```

```
In[5]:= %^(1/56)

Out[5]= 34
```

2.2. Rationale Zahlen
■ Nombres rationaux

Mathematica rechnet exakt, wenn nicht anders verlangt!
■ *Mathematica* calcule exactement, sauf si on le veut autrement!

Probiere aus: ■ Essaie:

```
In[6]:= 2/4 + 24/144

Out[6]= 2/3
```

```
In[7]:= 2 + 2/5

Out[7]= 12/5
```

2.3. Irrationale Zahlen
■ Nombres irrationaux

Mathematica rechnet exakt
■ *Mathematica* calcule exactement

Beispiel: Quadratwurzel ■ Exemple: racine carrée

```
In[8]:= Sqrt[17]

Out[8]= Sqrt[17]
```

```
In[9]:= Sqrt[17.0]

Out[9]= 4.12311
```

```
In[10]:= Sqrt[17]

Out[10]= Sqrt[17]
```

In[11]:= **Sqrt[17]//N**

Out[11]= 4.12311

2.4. Annäherung durch Dezimalbrüche
■ Approche par fractions décimales

Falls irgendwo in einem Ausdruck ein Dezimalpunkt vorkommt, gilt der ganze Ausdruck als Fliesskommazahl.
■ S'il y a quelque part dans une expression un point décimal, toute l'expression a la valeur d'un nombre à virgule flottante.

Probiere aus: ■ Essaie:

In[12]:= **1/2 + 2.4/144**

Out[12]= 0.516667

In[13]:= **.5/7 + Pi**

Out[13]= 3.21302

In[14]:= **.5/7 + Pi // N**

Out[14]= 3.21302

2.5. Komplexe Zahlen
■ Nombres complexes

"I" bedeutet die imaginäre Einheit. "Re" erzeugt den Realanteil, "Im" den Imaginäranteil, "Abs" den absoluten Betrag, "Conjugate" das konjugiert Komplexe, "Round" rundet. Schaue, was *Mathematica* tut und überlege wieso das Resultat so aussieht:
■ "I" signifie l'unité imaginaire. "Re" crée la partie réelle, "Im" la partie imaginaire, "Abs" la valeur absolue, "Conjugate" le complexe conjugé, "Round" arrondit. Observe ce que *Mathematica* fait et réfléchis pourquoi le résultat est ainsi:

In[15]:= **(2 + 4I)/(5 + 2I)**

Out[15]= $\frac{18}{29} + \frac{16\,i}{29}$

In[16]:= **Re[3 + 4I]**

Out[16]= 3

In[17]:= **Im[3 + 4I]**

Out[17]= 4

In[18]:= **Conjugate[3 + 4I]**

Out[18]= 3 − 4 i

```
In[19]:= Abs[3 + 4I]
Out[19]= 5

In[20]:= Round[2.7 - 8.6 I]
Out[20]= 3 - 9 i

In[21]:= Cos[3 + 4I] // N
Out[21]= -27.0349 - 3.85115 i

In[22]:= Sin[3 + 4I]
Out[22]= Sin[3 + 4 i]

In[23]:= Exp[3 + 4I] // N
Out[23]= -13.1288 - 15.2008 i

In[24]:= Log[3 + 4I]
Out[24]= Log[3 + 4 i]
```

2.6. Symbole und Zahlen, Listen etc.
■ Symboles et nombres, listes etc.

Für *Mathematica* ist ein Buchstabe oder eine Buchstabenfolge ein Symbol, z.B. eine Variable. Variablen können auch mit "$" beginnen. "7a" jedoch bedeutet "7 mal a". Probiere aus:

■ Pour *Mathematica* une lettre ou une suite de lettres sont un symbole, p.ex. une variable. Les variables peuvent aussi commencer par "$". "7a" par contre signifie "7 fois a". Essaie:

```
In[25]:= 7a
Out[25]= 7 a

In[26]:= a = 5
Out[26]= 5

In[27]:= 7a
Out[27]= 35
```

In "Miscellaneus/Units.nb" sind Einheiten und Funktionen zur Einheitenverwandlung gespeichert:

■ Dans "Miscellaneos/Units.nb" sont mémorisées les unités et les fonctions pour transformer les unités:

```
In[28]:= Needs["Miscellaneous`Units`"]

In[29]:= 120 Pound
Out[29]= 120 Pound

In[30]:= Convert[120 Pound, Kilogram]
Out[30]= 54.431 Kilogram
```

Wichtige transzendente Zahlen sind als Symbole gespeichert.
Probiere aus:
■ Les nombres transcendants importants sont mémorisés comme symboles.
Essaie:

```
In[31]:= E
Out[31]= e

In[32]:= N[E]
Out[32]= 2.71828

In[33]:= I
Out[33]= i

In[34]:= I^2
Out[34]= -1

In[35]:= Degree
Out[35]= °

In[36]:= N[Sin[Pi/4]]
Out[36]= 0.707107

In[37]:= N[Sin[45 Degree]]
Out[37]= 0.707107
```

Mathematica kann weiter mit Listen (Vektoren, Matrizen, Tabellen etc rechnen). Der Themenkreis wird später behandelt.
■ *Mathematica* sait aussi calculer avec des listes (vecteurs, matrices tableaux etc.) Cet ensemble de thèmes sera traité plus tard.

```
In[38]:= ??List
         {e1, e2, ... } is a list of elements. Mehr…
         Attributes[List] = {Locked, Protected}
```

2.7. Approximationen
■ Approximations

Rationale Zahlen sind exakt gespeichert. "N" bedeutet approximieren. Probiere aus:
■ Les nombres rationaux sont sauvés précisément. "N" signifie approximer. Essaie:

```
In[39]:= Sqrt[17]
Out[39]= √17
```

In[40]:= **N[%]**

Out[40]= 4.12311

In[41]:= **17^(1/2)**

Out[41]= $\sqrt{17}$

In[42]:= **% // N**

Out[42]= 4.12311

In[43]:= **17^(0.5)**

Out[43]= 4.12311

In[44]:= **N[Sqrt[17]]**

Out[44]= 4.12311

In[45]:= **Sqrt[17] // N**

Out[45]= 4.12311

In[46]:= **Precision[%]**

Out[46]= MachinePrecision

Wie gross ist also die Präzision? Die Präzision kann "beliebig" vergrðssert werden:
■ Combien la précision est-elle grande? La précision peut être agrandie "autant qu'on veut":

In[47]:= **N[Sqrt[17], 100]**

Out[47]=
```
4.123105625617660549821409855974077025147199225373620434398633573094954346337621
 5358786365081068429 7
```

Wie gross ist hier die Präzision? ■ Combien la précision est-elle grande?

In[48]:= **Pi**

Out[48]= π

In[49]:= **N[Pi]**

Out[49]= 3.14159

In[50]:= **N[Pi,500]**

Out[50]=
```
3.14159265358979323846264338327950288419716939937510582097494459230781640628620
 8986280348253421170679821480865132823066470938446095505822317253594081284811174
 5028410270193852110555964462294895493038196442881097566593344612847564823378678
 3165271201909145648566923460348610454326648213393607260249141273724587006606315
 5881748815209209628292540917153643678925903600113305305488204665213841469519415
 1160943305727036575959195309218611738193261179310511854807446237996274956735188
 5752724891227938183011949 1
```

In[51]:= **N[E,100]**

Out[51]=
```
2.718281828459045235360287471352662497757247093699959574966967627724076630353547
 5945713821785251664 27
```

2.7. 1. Exakt versus approximiert:
▮ Exact versus approximé:

Probiere aus: ▮ Essaie:

```
In[52]:= 3/10 + 1/2
```

$$\text{Out[52]= } \frac{4}{5}$$

```
In[53]:= Precision[%]
```

$$\text{Out[53]= } \infty$$

```
In[54]:= .36 + 1/3
Out[54]= 0.693333

In[55]:= Precision[%]
Out[55]= MachinePrecision
```

2.7.2. Konvertierung approximativer Werte in exakte Werte:
▮ Transformation des valeurs approximatives en valeurs exactes:

Lasse Dir mit der Help-Funktion folgende Funktionen erklären:
▮ Fais-toi expliquer par la fonction Help les fonctions suivantes:

```
In[56]:= ?Rationalize

        Rationalize[x] takes Real numbers in x that are close to rationals, and
           converts them to exact Rational numbers. Rationalize[x, dx] performs the
           conversion whenever the error made is smaller in magnitude than dx. Mehr…

In[57]:= ?Round

        Round[x] gives the integer closest to x. Mehr…

In[58]:= ?Chop

        Chop[expr] replaces approximate real numbers
           in expr that are close to zero by the exact integer 0. Mehr…

In[59]:= ?Floor

        Floor[x] gives the greatest integer less than or equal to x. Mehr…

In[60]:= ?Ceiling

        Ceiling[x] gives the smallest integer greater than or equal to x. Mehr…
```

Probiere aus: ▮ Essaie:

```
In[61]:= Rationalize[3.1416]
```

$$\text{Out[61]= } \frac{3927}{1250}$$

```
In[62]:= Rationalize[3.1415926536]

Out[62]= 3.14159

In[63]:= Rationalize[3.1415926536,0]

Out[63]= 228350988/72686377

In[64]:= Round[2.57432]

Out[64]= 3

In[65]:= Round[2,45892]

         Round::argx : Round called with 2 arguments; 1 argument is expected. Mehr…

Out[65]= Round[2, 45892]
```

Was war los ? ▮ Que se passe-t-il?

```
In[66]:= Round[Sqrt[17]]

Out[66]= 4

In[67]:= Round[N[Sqrt[17]]]

Out[67]= 4

In[68]:= Chop[0.00000000002]

Out[68]= 0

In[69]:= Ceiling[3.4]

Out[69]= 4

In[70]:= Floor[3.4]

Out[70]= 3
```

2.7.3. Arbeiten auf einem selber fixierten Präzisions-Niveau: ▮ Travailler à un niveau de précision fixé par soi-même:

Man kann z.B. selbst eine Funktion definieren, die die auszugebenden Kommastellen festlegt:
▮ On peut définir soi-même par exemple une fonction qui détermine le nombre de places dernière la virgule à émettre:

```
In[71]:= n30[x_]:=N[x,30]
```

Anwendung: ▮ Application:

```
In[72]:= n30[5/7]

Out[72]= 0.714285714285714285714285714286

In[73]:= N[5/7]

Out[73]= 0.714286
```

Es gibt aber auch die globale Variable "$Post", in die sich eine Funktion eingeben lässt. Diese Variable wird auf jeden auszugebenen Ausdruck angewandt:

■ Mais il existe aussi la variable globale "$Post", dans laquelle on peut entrer une fonction. Cette variable est appliquée à chaque expression à émettre:

```
In[74]:= ?$Post

        $Post is a global variable whose value, if set, is applied to every output expression. Mehr...

In[75]:= $Post=n30

Out[75]= n30

In[76]:= Sqrt[3]

Out[76]= 1.732050807568877293527446341 51

In[77]:= Precision[%]

Out[77]= 30.
```

Der Inhalt von $Post soll auch wieder gelöscht werden können. Das geschieht durch Eingabe des Zeichens für "missing value", nämlich "." :

■ Il faut pouvoir aussi effacer le contenu de $Post. On peut faire cela en entrant le signe pour "missing value", c'est-à-dite "." :

```
In[78]:= $Post=.

In[79]:= Sqrt[3]

Out[79]= √3

In[80]:= N[Sqrt[3]]

Out[80]= 1.73205

In[81]:= N[Pi,25]

Out[81]= 3.1415926535897932384626 43

In[82]:= N[%,200]

Out[82]= 3.1415926535897932384626 43
```

Wie genau ist also die übliche Präzision?

■ Combien la précision courante est-elle exacte?

2.8. Formatierte Zahlenausgabe
■ Sortie formatée de nombres

Mathematica bietet Möglichkeiten, die Zahlenausgabe zu formatieren. So lässt sich die Leserlichkeit verbessern. Probiere aus:

Mathematica offre la possibilité de formater la sortie de nombres. $Ainsi on peut améliorer la lisibilité. Essaie:

In[83]:=
```
Options[NumberForm]
```

Out[83]=
```
{DigitBlock → ∞, ExponentFunction → Automatic, ExponentStep → 1,
 NumberFormat → Automatic, NumberMultiplier → ×, NumberPadding → {, },
 NumberPoint → ., NumberSeparator → ,, NumberSigns → {-, }, SignPadding → False}
```

In[84]:=
```
1.23456789 10^20
```

Out[84]= 1.23457×10^{20}

In[85]:=
```
NumberForm[1.23456789 10^20,
        ExponentStep -> 19,
        NumberSigns  -> {"-","+"}
            ]
```

Out[85]//NumberForm=

$+12.3457 \times 10^{19}$

In[86]:=
```
NumberForm[1.23456789 10^20,
        ExponentStep -> 21,
        NumberSigns  -> {"-","+"}
            ]
```

```
NumberForm::sigz :
 In addition to the number of digits requested, one or more zeros will appear as placeholders. Mehr…
```

Out[86]//NumberForm=

+123456789000000000000.

In[87]:=
```
NumberForm[123456789 10^20,
        ExponentStep -> 1000,
        NumberSigns  -> {"-","+"}
            ]
```

Out[87]//NumberForm=

+12345678900000000000000000000

In[88]:=
```
BaseForm[1.4,2]
```

Out[88]//BaseForm=

1.0110011001100110011_2

In[89]:=
```
BaseForm[1.4,3]
```

Out[89]//BaseForm=

1.101210121012_3

In[90]:=
```
BaseForm[1.4,4]
```

Out[90]//BaseForm=

1.121212122_4

In[91]:=
```
BaseForm[1.4,5]
```

Out[91]//BaseForm=

1.2_5

In[92]:=
```
BaseForm[1.4,6]
```

Out[92]//BaseForm=

1.2222222_6

In[93]:= **BaseForm[1.4,7]**

Out[93]//BaseForm=

1.254126_7

In[94]:= **BaseForm[1.4,10]**

Out[94]//BaseForm=

1.4

In[95]:= **BaseForm[1.4,20]**

Out[95]//BaseForm=

1.8_{20}

Was war los ? ■ Que se passe-t-il?

2.9. Wichtige gespeicherte Konstanten ■ Constantes importantes mémorisées

Falls Dir eine der folgenden Konstanten unbekannt ist, so schlage in deinem Formelbuch nach. Probiere aus:
■ Si tu ne connais pas l'une des constantes suivantes, consulte ton manuel de formules. Essaie:

In[96]:= **??Catalan**

Catalan is Catalan's constant, with numerical value approximately equal to 0.915966. Mehr…

Attributes[Catalan] = {Constant, Protected}

In[97]:= **?Degree**

Degree gives the number of radians in one degree. It
has a numerical value of Pi/180. It is also a unit multiplier. Mehr…

In[98]:= **?E**

E is the exponential constant e (base of natural
logarithms), with numerical value approximately equal to 2.71828. Mehr…

In[99]:= **?EulerGamma**

EulerGamma is Euler's constant gamma, with numerical value approximately equal to 0.577216.
Mehr…

In[100]:=

?GoldenRatio

GoldenRatio is the golden ratio (1 + Sqrt[
5])/2, with numerical value approximately equal to 1.61803. Mehr…

In[101]:=

?I

I represents the imaginary unit Sqrt[-1]. Mehr…

In[102]:=

?Pi

Pi is the constant pi, with numerical value approximately equal to 3.14159. Mehr…

In[103]:=
```
Pi
```

Out[103]=
π

In[104]:=
```
NumberForm[N[Pi,45],
        NumberSeparator  -> " ",
        DigitBlock -> 5
            ]
```

Out[104]//NumberForm=
```
3.14159 26535 89793 23846 26433 83279 50288 41971 6940
```

In[105]:=
```
Log[E]
```

Out[105]=
```
1
```

In[106]:=
```
Cos[Pi/3]
```

Out[106]=
$\frac{1}{2}$

In[107]:=
```
N[Sin[45 Degree]]
```

Out[107]=
```
0.707107
```

In[108]:=
```
avogadro = 6.02250 10^23
```

Out[108]=
6.0225×10^{23}

In[109]:=
```
Log[avogadro]
```

Out[109]=
```
54.755
```

2.10. Zufallszahlen
∎ Nombres aléatoires

Mathematica besitzt die Funktion "Random", die gleichverteilte Pseudozufallszahlen generiert. Solche braucht man z.B. in der Statistik, um Testdaten zu erzeugen. Probiere aus:

∎ *Mathematica* possède la fonction "Random", qui génère des nombres pseudo-aléatoires en distribution homogène. On les emploie p.ex. dans la statistique, pour créer les données de test. Essaie:

In[110]:=
```
Random[]
```
Out[110]=
```
0.771439
```

In[111]:=
```
Random[Integer]
```
Out[111]=
```
1
```

In[112]:=
```
Random[Integer, {0, 100}]
```
Out[112]=
```
36
```

In[113]:=
```
Random[Complex, {1 + 3I, 4 + 7I}]
```
Out[113]=
```
1.93288 + 5.78143 i
```

Die Packages "ContinousDistributions.nb" und "DiscreteDistributions.nb" enthalten Definitionen, mit denen sich andere als gleichverteilte Pseudozufallszahlen erzeugen lassen. Probiere aus:

■ Les Packages "ContinousDistributions.nb" et "DiscreteDistributions.nb" contiennent des définitions par lesquelles on peut générer des nombres pseudo-aléatoires autres que distribués homogènement. Essaie:

In[114]:=
```
Needs["Statistics`ContinuousDistributions`"]
```

In[115]:=
```
Random[NormalDistribution[5]]
```
Out[115]=
```
4.77034
```

Der Pseudozufallszahlengenerator benutzt die Zeit seit dem Einschalten der Maschine als Variable zur Berechnung der Zahlen. Mit "SeedRandom" lässt sich der Generator wieder "zurücksetzen". Das kann benutzt werden, um mehrmals dieselben Zahlen zu generieren. Probiere aus:

■ Le générateur des nombres pseudo-aléatoires utilise le laps de temps dès le démarrage de la machine comme variable pour calculer les nombres. Par "SeedRandom" on peut "remettre en arrière" le générateur. On employe cela pour générer plusieurs fois les mêmes nombres. Essaie:

In[116]:=
```
??SeedRandom
```
```
SeedRandom[n] resets the pseudorandom number generator, using the integer n as a seed.
   SeedRandom[ ] resets the generator, using as a seed the time of day. Mehr...

Attributes[SeedRandom] = {Protected}
```

In[117]:=
```
SeedRandom[2]
```

In[118]:=
```
{Random[], Random[]}
```
Out[118]=
```
{0.238705, 0.844529}
```

In[119]:=
```
SeedRandom[2]
```

In[120]:=
```
{Random[], Random[]}
```
Out[120]=
```
{0.238705, 0.844529}
```

2.11. Iteratoren
▮ Itérateurs

Mathematica stellt iterative Funktionen (Iteratoren) zur Verfügung, um die Eingabe wiederholter Rechnungen abzukürzen. Einige sind hier gezeigt. Probiere aus:

▮ *Mathematica* met à disposition des fonctions itératives pour abréger l'entrée de calculs répétés. Voici quelques-uns. Essaie:

In[121]:=
```
??Table
```
```
Table[expr, {imax}] generates a list of imax copies of expr. Table[
   expr, {i, imax}] generates a list of the values of expr when i runs from
   1 to imax. Table[expr, {i, imin, imax}] starts with i = imin. Table[expr,
   {i, imin, imax, di}] uses steps di. Table[expr, {i, imin, imax}, {j, jmin,
   jmax}, ... ] gives a nested list. The list associated with i is outermost. Mehr...

Attributes[Table] = {HoldAll, Protected}
```

In[122]:=
```
Table[x, {5}]
```
Out[122]=
```
{x, x, x, x, x}
```

In[123]:=
```
??Product
```
```
Product[f, {i, imax}] evaluates the product of the expressions f as evaluated
   for each i from 1 to imax. Product[f, {i, imin, imax}] starts with i = imin.
   Product[f, {i, imin, imax, di}] uses steps di. Product[f, {i, imin, imax},
   {j, jmin, jmax}, ... ] evaluates a product over multiple indices. Mehr...

Attributes[Product] = {HoldAll, Protected, ReadProtected}
```

In[124]:=
```
Product[x + k y, {k, 4}]
```
Out[124]=
$$(x + y)\,(x + 2\,y)\,(x + 3\,y)\,(x + 4\,y)$$

In[125]:=

```
??Do
```

```
Do[expr, {imax}] evaluates expr imax times. Do[expr, {i, imax}] evaluates expr
   with the variable i successively taking on the values 1 through imax (in steps
   of 1). Do[expr, {i, imin, imax}] starts with i = imin. Do[expr, {i, imin,
   imax, di}] uses steps di. Do[expr, {i, imin, imax}, {j, jmin, jmax}, ... ]
   evaluates expr looping over different values of j, etc. for each i. Mehr...

Attributes[Do] = {HoldAll, Protected}
```

In[126]:=

```
Do[Print["k: ", k], {k, 7, 9}]
```

```
k: 7

k: 8

k: 9
```

In[127]:=

```
??TableForm
```

```
TableForm[list] prints with the elements of list arranged in an array of rectangular cells.
  Mehr...

Attributes[TableForm] = {Protected, ReadProtected}

Options[TableForm] := {TableAlignments → Automatic, TableDepth → ∞,
  TableDirections → Column, TableHeadings → None, TableSpacing → Automatic}
```

In[128]:=

```
TableForm[Table[{k, 10^(-k), Chop[N[10^(-k)]]}, {k, 7, 14}]]
```

Out[128]//TableForm=

7	$\frac{1}{10000000}$	$1.\times 10^{-7}$
8	$\frac{1}{100000000}$	$1.\times 10^{-8}$
9	$\frac{1}{1000000000}$	$1.\times 10^{-9}$
10	$\frac{1}{10000000000}$	$1.\times 10^{-10}$
11	$\frac{1}{100000000000}$	0
12	$\frac{1}{1000000000000}$	0
13	$\frac{1}{10000000000000}$	0
14	$\frac{1}{100000000000000}$	0

In[129]:=

```
??Sum
```

```
Sum[f, {i, imax}] evaluates the sum of the expressions f as
   evaluated for each i from 1 to imax. Sum[f, {i, imin, imax}] starts with
   i = imin. Sum[f, {i, imin, imax, di}] uses steps di. Sum[f, {i, imin,
   imax}, {j, jmin, jmax}, ... ] evaluates a sum over multiple indices. Mehr...

Attributes[Sum] = {HoldAll, Protected, ReadProtected}
```

In[130]:=

```
Sum[i x^i, {i, 3, 23, 5}]
```

Out[130]=

$$3\,x^3 + 8\,x^8 + 13\,x^{13} + 18\,x^{18} + 23\,x^{23}$$

2.12. Matrizenrechnung
▮ Calcul des matrices

Probiere aus: ▮ Essaie:

In[131]:=
```
??MatrixForm
```

```
MatrixForm[list] prints with the elements of list arranged in a regular array. Mehr...

Attributes[MatrixForm] = {Protected, ReadProtected}

Options[MatrixForm] := {TableAlignments → Automatic, TableDepth → ∞,
  TableDirections → Column, TableHeadings → None, TableSpacing → Automatic}
```

In[132]:=
```
??Inverse
```

```
Inverse[m] gives the inverse of a square matrix m. Mehr...

Attributes[Inverse] = {Protected}

Options[Inverse] = {Method → Automatic, Modulus → 0, ZeroTest → (#1 == 0 &)}
```

In[133]:=
```
??Det
```

```
Det[m] gives the determinant of the square matrix m. Mehr...

Attributes[Det] = {Protected}

Options[Det] = {Modulus → 0}
```

In[134]:=
```
??Eigenvalues
```

```
Eigenvalues[m] gives a list of the eigenvalues of the square
   matrix m. Eigenvalues[{m, a}] gives the generalized eigenvalues of m with
   respect to a. Eigenvalues[m, k] gives the first k eigenvalues of m. Mehr...

Attributes[Eigenvalues] = {Protected}

Options[Eigenvalues] = {Cubics → False, Method → Automatic, Quartics → False}
```

In[135]:=
```
??Eigenvectors
```

```
Eigenvectors[m] gives a list of the eigenvectors of the square
   matrix m. Eigenvectors[{m, a}] gives the generalized eigenvectors of m with
   respect to a. Eigenvectors[m, k] gives the first k eigenvectors of m. Mehr...

Attributes[Eigenvectors] = {Protected}

Options[Eigenvectors] =
 {Cubics → False, Method → Automatic, Quartics → False, ZeroTest → Automatic}
```

In[136]:=
```
??LinearSolve
```

```
LinearSolve[m, b] finds an x which solves the matrix
   equation m.x==b. LinearSolve[m] generates a LinearSolveFunction[ ... ]
   which can be applied repeatedly to different b. Mehr...

Attributes[LinearSolve] = {Protected}

Options[LinearSolve] = {Method → Automatic, Modulus → 0, ZeroTest → Automatic}
```

In[137]:=
```
??NullSpace
```

```
NullSpace[m] gives a list of vectors that forms a basis for the null space of the matrix m.
  Mehr...

Attributes[NullSpace] = {Protected}

Options[NullSpace] =
  {Method → Automatic, Modulus → 0, Tolerance → Automatic, ZeroTest → Automatic}
```

In[138]:=
```
??RowReduce
```

```
RowReduce[m] gives the row-reduced form of the matrix m. Mehr...

Attributes[RowReduce] = {Protected}

Options[RowReduce] =
  {Method → Automatic, Modulus → 0, Tolerance → Automatic, ZeroTest → Automatic}
```

In[139]:=
```
??Transpose
```

```
Transpose[list] transposes the first two levels in list.
   Transpose[list, {n1, n2, ... }] transposes list so that the levels 1,
   2, ... in list correspond to levels n1, n2, ... in the result. Mehr...

Attributes[Transpose] = {Protected}
```

In[140]:=
```
??IdentityMatrix
```

```
IdentityMatrix[n] gives the n by n identity matrix. Mehr...

Attributes[IdentityMatrix] = {Protected}
```

In[141]:=
```
m1 = {{1, 2},{3, 4}}
```

Out[141]=
```
{{1, 2}, {3, 4}}
```

In[142]:=
```
Transpose[m1] // MatrixForm
```

Out[142]//MatrixForm=

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

In[143]:=
```
MatrixForm[m1]
```
Out[143]//MatrixForm=
$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

In[144]:=
```
MatrixForm[{{1, 2},{3, 4}}]
```
Out[144]//MatrixForm=
$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

In[145]:=
```
m2 = {{0, -1},{8, -6}}
```
Out[145]=
```
{{0, -1}, {8, -6}}
```

In[146]:=
```
m3 = {{0, -1},{8, -6}} // MatrixForm
```
Out[146]//MatrixForm=
$$\begin{pmatrix} 0 & -1 \\ 8 & -6 \end{pmatrix}$$

Das Matrixprodukt wird mit einem gewöhnlichen Punkt geschrieben. Probiere aus:

■ Le produit de matrices s'écrit avec un simple point. Essaie:

In[147]:=
```
m1.m2 // MatrixForm
```
Out[147]//MatrixForm=
$$\begin{pmatrix} 16 & -13 \\ 32 & -27 \end{pmatrix}$$

In[148]:=
```
m1.m3
```
Out[148]=
$$\{\{1, 2\}, \{3, 4\}\}.\begin{pmatrix} 0 & -1 \\ 8 & -6 \end{pmatrix}$$

Elementweise Multiplikation. Probiere aus:

■ Multiplication par élément. Essaie:

In[149]:=
```
m1 m2 // MatrixForm
```
Out[149]//MatrixForm=
$$\begin{pmatrix} 0 & -2 \\ 24 & -24 \end{pmatrix}$$

Probiere aus: ■ Essaie:

In[150]:=
```
?NullSpace
```
```
NullSpace[m] gives a list of vectors that forms a basis for the null space of the matrix m.
```
Mehr…

In[151]:=
```
m4 = {{1,2,3},{1,2,3},{0,0,1}}
```

Out[151]=
```
{{1, 2, 3}, {1, 2, 3}, {0, 0, 1}}
```

In[152]:=
```
NullSpace[m4]
```

Out[152]=
```
{{-2, 1, 0}}
```

In[153]:=
```
?RowReduce
```

```
RowReduce[m] gives the row-reduced form of the matrix m. Mehr...
```

In[154]:=
```
RowReduce[m4] // MatrixForm
```

Out[154]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

In[155]:=
```
Eigenvectors[m1]
```

Out[155]=

$$\left\{\left\{-\frac{4}{3}+\frac{1}{6}\left(5+\sqrt{33}\right), 1\right\}, \left\{-\frac{4}{3}+\frac{1}{6}\left(5-\sqrt{33}\right), 1\right\}\right\}$$

In[156]:=
```
Transpose[m1] // MatrixForm
```

Out[156]//MatrixForm=

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

In[157]:=
```
m = Table[Random[], {3}, {3}]
```

Out[157]=
```
{{0.473928, 0.421515, 0.512692},
 {0.0228035, 0.529257, 0.0507012}, {0.74907, 0.632356, 0.0725195}}
```

In[158]:=
```
MatrixForm[m]
```

Out[158]//MatrixForm=

$$\begin{pmatrix} 0.473928 & 0.421515 & 0.512692 \\ 0.0228035 & 0.529257 & 0.0507012 \\ 0.74907 & 0.632356 & 0.0725195 \end{pmatrix}$$

In[159]:=
```
mInv = Inverse[m]
```

Out[159]=
```
{{-0.0355956, -1.65375, 1.40785},
 {-0.204582, 1.96936, 0.0694849}, {2.15159, -0.0904116, -1.35854}}
```

In[160]:=
```
MatrixForm[%]
```

Out[160]//MatrixForm=

$$\begin{pmatrix} -0.0355956 & -1.65375 & 1.40785 \\ -0.204582 & 1.96936 & 0.0694849 \\ 2.15159 & -0.0904116 & -1.35854 \end{pmatrix}$$

In[161]:=
```
m.mInv
```

Out[161]=

$$\{\{1., 6.93889\times10^{-17}, 0.\}, \{-1.38778\times10^{-17}, 1., 1.38778\times10^{-17}\}, \{-2.77556\times10^{-17}, 1.17961\times10^{-16}, 1.\}\}$$

In[162]:=
```
MatrixForm[%]
```

Out[162]//MatrixForm=

$$\begin{pmatrix} 1. & 6.93889\times10^{-17} & 0. \\ -1.38778\times10^{-17} & 1. & 1.38778\times10^{-17} \\ -2.77556\times10^{-17} & 1.17961\times10^{-16} & 1. \end{pmatrix}$$

In[163]:=
```
Chop[%]
```

Out[163]=

$$\{\{1., 0, 0\}, \{0, 1., 0\}, \{0, 0, 1.\}\}$$

In[164]:=
```
MatrixForm[%]
```

Out[164]//MatrixForm=

$$\begin{pmatrix} 1. & 0 & 0 \\ 0 & 1. & 0 \\ 0 & 0 & 1. \end{pmatrix}$$

In[165]:=
```
IdentityMatrix[7] // MatrixForm
```

Out[165]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

2.13. Lösen von Gleichungen
∎ Résoudre des équations

Probiere aus: ∎ Essaie:

In[166]:=
```
??Solve
```

```
Solve[eqns, vars] attempts to solve an equation or set of
   equations for the variables vars. Solve[eqns, vars, elims] attempts to
   solve the equations for vars, eliminating the variables elims. Mehr...

Attributes[Solve] = {Protected}

Options[Solve] = {InverseFunctions → Automatic, MakeRules → False, Method → 3,
  Mode → Generic, Sort → True, VerifySolutions → Automatic, WorkingPrecision → ∞}
```

In[167]:=
```
??NRoots
```

```
NRoots[lhs==rhs, var] gives a list of numerical approximations to the roots of a polynomial
  equation. NRoots[lhs==rhs, var, n] uses n-digit precision in the computations.

Attributes[NRoots] = {Protected}
```

In[168]:=
```
??FindRoot
```

```
FindRoot[lhs==rhs, {x, x0}] searches for a numerical solution to the equation lhs==
   rhs, starting with x=x0. FindRoot[{eqn1, eqn2, ... }, {{x, x0}, {y, y0}, ... }]
   searches for a numerical solution to the simultaneous equations eqni. Mehr...

Attributes[FindRoot] = {HoldAll, Protected}

Options[FindRoot] = {AccuracyGoal → Automatic, Compiled → True, DampingFactor → 1,
  EvaluationMonitor → None, Jacobian → Automatic, MaxIterations → 100, Method → Automatic,
  PrecisionGoal → Automatic, StepMonitor → None, WorkingPrecision → MachinePrecision}
```

In[169]:=
```
Solve[{2x - 3y ==8, 4x + y == 6},{x, y}]
```

Out[169]=
$$\left\{\left\{x \to \frac{13}{7},\ y \to -\frac{10}{7}\right\}\right\}$$

In[170]:=
```
Solve[m1.{{x},{y}}=={{2},{5}}, {x, y}]
```

Out[170]=
$$\left\{\left\{x \to 1,\ y \to \frac{1}{2}\right\}\right\}$$

In[171]:=
```
NRoots[x^4 +3 x^2 + 5x == 7, x]
```

Out[171]=
```
x == -1.63541 || x == 0.390304 - 2.2034 i || x == 0.390304 + 2.2034 i || x == 0.854804
```

FindRoot benutzt die Newton-Methode. Darin wird die Ableitung benutzt:

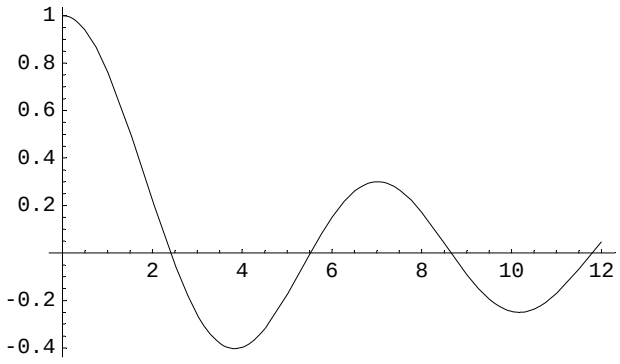
■ FindRoot emploie la méthode de Newton. On y employe la déduction:

In[172]:=
```
FindRoot[Sin[x]/x == 0, {x, 2}]
```

Out[172]=
```
{x → 3.14159}
```

Bessel-Funktionen: ■ Fonctions de Bessel

In[173]:=
```
Plot[BesselJ[0, x], {x, 0, 12}];
```



In[174]:=
```
FindRoot[BesselJ[0, x]/x == 0, {x, 12}]
```

Out[174]=
```
{x → 11.7915}
```

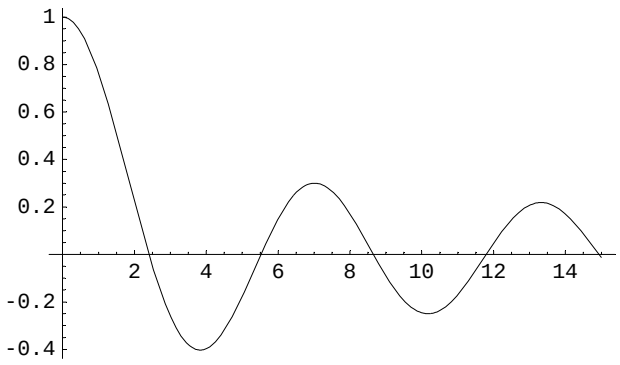
In[175]:=
```
FindRoot[BesselJ[0, x]/x, {x, 12}]
```

Out[175]=
```
{x → 11.7915}
```

In[176]:=
```
FindRoot[BesselJ[0, x]/x, {x, 7}]
```

Out[176]=
```
{x → 14.9309}
```

In[177]:=
```
Plot[BesselJ[0, x], {x, 0, 15}];
```

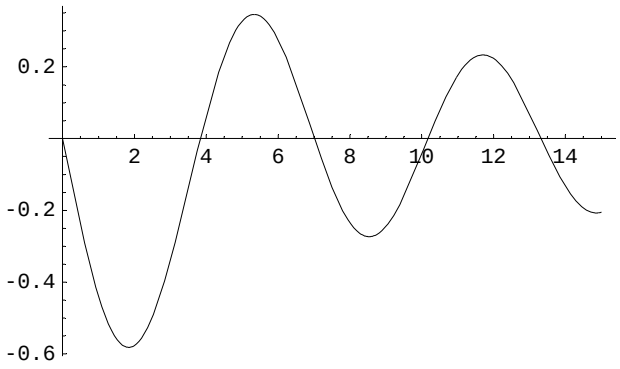


FindRoot benutzt die Newton-Methode. Darin wird die Ableitung benutzt. Hier ein Plot der Ableitung obiger Bessel-Funktion. (Beachte, dass die Ableitung bei x = 7 klein ist.)

FindRoot emploie la méthode de Newton. On y employe la déduction. Voici un plot de la déduction de la fonction de Bessel ci-dessus. (Remarque que la déduction est petite pour x = 7.)

In[178]:=

```
Plot[-BesselJ[1, x], {x, 0, 15}];
```



2.14. Numerische Integration etc.
■ Intégtation numérique etc.

Probiere aus: ■ Essaie:

In[179]:=

```
??NIntegrate
```

```
NIntegrate[f, {x, xmin, xmax}] gives a numerical approximation
   to the integral of f with respect to x from xmin to xmax. Mehr...

Attributes[NIntegrate] = {HoldAll, Protected}

Options[NIntegrate] =
  {AccuracyGoal → ∞, Compiled → True, EvaluationMonitor → None, GaussPoints → Automatic,
   MaxPoints → Automatic, MaxRecursion → 6, Method → Automatic, MinRecursion → 0,
   PrecisionGoal → Automatic, SingularityDepth → 4, WorkingPrecision → MachinePrecision}
```

In[180]:=

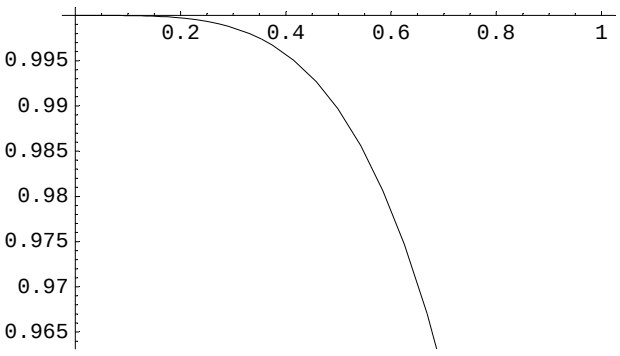
```
NIntegrate[Sin[x^2]/x^2, {x, 0, 1}]
```

Out[180]=

```
0.967577
```

In[181]:=

```
Plot[Sin[x^2]/x^2, {x, 0, 1}];
```



In[182]:=
```
NIntegrate[Sin[Sin[Cos[x^10]]], {x, 0, 1}]
```

Out[182]=
```
0.735169
```

In[183]:=
```
??NSum
```

```
NSum[f, {i, imin, imax}] gives a numerical approximation
   to the sum of the expressions f as evaluated for each i from imin to
   imax. NSum[f, {i, imin, imax, di}] uses a step di in the sum. Mehr...

Attributes[NSum] = {HoldAll, Protected}

Options[NSum] = {AccuracyGoal → ∞, Compiled → True, EvaluationMonitor → None,
  Method → Automatic, NSumExtraTerms → 12, NSumTerms → 15, PrecisionGoal → Automatic,
  VerifyConvergence → True, WorkingPrecision → MachinePrecision, WynnDegree → 1}
```

In[184]:=
```
NSum[(n^2 - n + 1) Log[n]/n^3, {n, 1, 100}]
```

Out[184]=
```
9.87012 + 0. i
```

In[185]:=
```
??NProduct
```

```
NProduct[f, {i, imin, imax}] gives a numerical approximation to the
   product of the expressions f as evaluated for each i from imin to imax.
   NProduct[f, {i, imin, imax, di}] uses a step di in the product. Mehr...

Attributes[NProduct] = {HoldAll, Protected}

Options[NProduct] = {AccuracyGoal → ∞, Compiled → True, EvaluationMonitor → None,
  Method → Automatic, NProductExtraFactors → 15, NProductFactors → 15, PrecisionGoal → Automatic,
  VerifyConvergence → True, WorkingPrecision → MachinePrecision, WynnDegree → 1}
```

In[186]:=
```
NProduct[(1 + 1/n)^n, {n, 1, 100}]
```

Out[186]=
```
2.89824 × 10^42 + 0. i
```

In[187]:=
```
Limit[(1 + 1/n)^n, n->Infinity]
```

Out[187]=
```
e
```

2.15. Numerische Lösung von Differentialgleichungen ▮ Solution numérique d'équations différentielles

Probiere aus: ▮ Essaie:

In[188]:=

```
??NDSolve
```

```
NDSolve[eqns, y, {x, xmin, xmax}] finds a numerical solution to the ordinary differential
   equations eqns for the function y with the independent variable x in the
   range xmin to xmax. NDSolve[eqns, y, {x, xmin, xmax}, {t, tmin, tmax}] finds a
   numerical solution to the partial differential equations eqns. NDSolve[eqns, {y1,
   y2, ... }, {x, xmin, xmax}] finds numerical solutions for the functions yi. Mehr...

Attributes[NDSolve] = {Protected}

Options[NDSolve] = {AccuracyGoal → Automatic, Compiled → True, DependentVariables → Automatic,
  EvaluationMonitor → None, MaxStepFraction → 1/10, MaxSteps → Automatic, MaxStepSize → Automatic,
  Method → Automatic, NormFunction → Automatic, PrecisionGoal → Automatic, SolveDelayed → False,
  StartingStepSize → Automatic, StepMonitor → None, WorkingPrecision → MachinePrecision}
```

In[189]:=

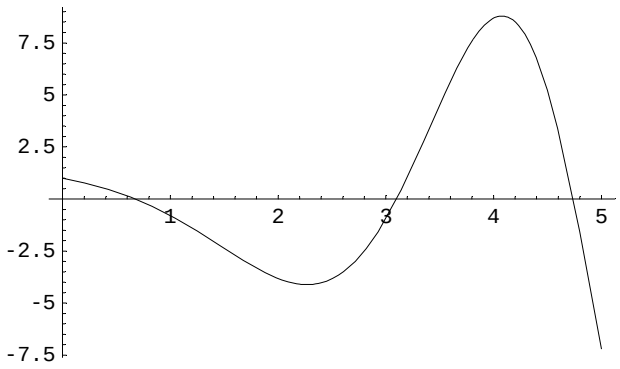
```
NDSolve[{y''[x] - y'[x] + x y[x] == 0, y'[0] == -1, y[0] == 1},
        y, {x, 0, 5}]
```

Out[189]=

```
{{y → InterpolatingFunction[{{0., 5.}}, <>]}}
```

In[190]:=

```
Plot[y[x] /. %, {x, 0, 5}];
```



"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[191]:=

```
(* Old Form: Remove["Global`@*"] *)
```

In[192]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

2. Numerische Probleme
▮ Problèmes numériques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 2 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 2.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Berechne die Anzahl Minuten in einem Jahr mit 365 Tagen.
▮ Calcule le nombre de minutes pour une année de 365 jours.

```
In[1]:= 365 Tag (24 Std/Tag) (60 Min/Std)

Out[1]= 525600 Min
```

Aufgabe 2 ▮ Problème 2

(a)

Berechne Pi auf 770 Stellen. Findest Du 6 sich folgende Ziffern 9?
▮ Calcule Pi sur 770 places. Trouves-tu 6 fois de suite le chiffre 9?

In[2]:= **N[Pi,770]**

Out[2]= 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899\
86280348253421170679821480865132823066470938446095505822317253594081284811174502\
84102701938521105559644622948954930381964428810975665933446128475648233786783165 27\
12019091456485669234603486104543266482133936072602491412737245870066063155881748 8\
15209209628292540917153643678925903600113305305488204665213841469519415116094330 5\
72703657595919530921861173819326117931051185480744623799627495673518857527248912 2\
79381830119491298336733624406566430860213949463952247371907021798609437027705392 1\
71762931767523846748184676694051320005681271452635608277857713427577896091736371 7\
87214684409012249534301465495853710507922796892589235420199561121290219608640344 1\
81598136297747713099605187072113499999984

(b)

Wie nahe ist e hoch (Pi mal Wurzel aus 163) bei einer ganzen Zahl?

■ Combien près d'un nombre entiers se trouve e puissance (Pi fois racine de 163)?

In[3]:= **N[E^(Pi 163^(1/2)),100]-Ceiling[N[E^(Pi 163^(1/2)),100]]**

Out[3]= $-7.49927402801814311120646143662663009137292462589621789352089881392687 0\times 10^{-13}$

In[4]:= **N[E^(Pi 163^(1/2)),100]-Floor[N[E^(Pi 163^(1/2)),100]]**

Out[4]= 0.99999999999925007259719818568887935385633733699086270753741037821064791011860731\
30

Aufgabe 3 ■ Problème 3

Verwende NumberForm um die Zahl 123456789 in 3-er-Blöcken auszugeben.

■ Sers-toi de NumberForm pour sortir le nombre 123456789 en blocs de 3 chiffres.

In[5]:= **NumberForm[123456789, DigitBlock->3]**

Out[5]//NumberForm=

123,456,789

Aufgabe 4 ■ Problème 4

Generiere mit Hilfe von Random zwei Pseudozufallszahlen und addiere diese.

■ Générer à l'aide de Random deux nombres pseudo-aléatoires et fais-en l'addition

In[6]:= **a=Random[];b=Random[];a+b**

Out[6]= 1.42666

Aufgabe 5 ■ Problème 5

Schaue, was *Mathematica* tut, und überlege wieso das Resultat so aussieht:

■ Observe ce que fait *Mathematica* et réfléchis pourquoi le résultat est ainsi:

```
In[7]:= Sqrt[3] + 3
```

Out[7]= $3 + \sqrt{3}$

```
In[8]:= Exp[2 Pi I]
```

```
Out[8]= 1
```

```
In[9]:= 5 > 3
```

```
Out[9]= True
```

```
In[10]:= Pi^E > E ^ Pi
```

```
Out[10]= False
```

```
In[11]:= N[Pi^E > E ^ Pi]
```

```
Out[11]= False
```

Aufgabe 6 ■ Problème 6

(a)

Studiere die folgenden Befehle:

■ Etudie les ordres suivants:

```
In[12]:= ??Table
```

```
Table[expr, {imax}] generates a list of imax copies of expr. Table[
   expr, {i, imax}] generates a list of the values of expr when i runs from
   1 to imax. Table[expr, {i, imin, imax}] starts with i = imin. Table[expr,
   {i, imin, imax, di}] uses steps di. Table[expr, {i, imin, imax}, {j, jmin,
   jmax}, ... ] gives a nested list. The list associated with i is outermost. Mehr...

Attributes[Table] = {HoldAll, Protected}
```

```
In[13]:= ??Sum
```

```
Sum[f, {i, imax}] evaluates the sum of the expressions f as
   evaluated for each i from 1 to imax. Sum[f, {i, imin, imax}] starts with
   i = imin. Sum[f, {i, imin, imax, di}] uses steps di. Sum[f, {i, imin,
   imax}, {j, jmin, jmax}, ... ] evaluates a sum over multiple indices. Mehr...

Attributes[Sum] = {HoldAll, Protected, ReadProtected}
```

```
In[14]:= ??Product
```

```
Product[f, {i, imax}] evaluates the product of the expressions f as evaluated
   for each i from 1 to imax. Product[f, {i, imin, imax}] starts with i = imin.
   Product[f, {i, imin, imax, di}] uses steps di. Product[f, {i, imin, imax},
   {j, jmin, jmax}, ... ] evaluates a product over multiple indices. Mehr...

Attributes[Product] = {HoldAll, Protected, ReadProtected}
```

(b)

Kreiere eine Liste mit 50 Ziffern 9.

- Crée une liste avec 50 fois le chiffre 9.

```
In[15]:= Table[9,{50}]

Out[15]= {9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
          9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9}
```

(c)

Kreiere eine Liste mit den ersten 24 Quadratzahlen.

- Crée une liste avec les 24 premiers nombres carrés.

```
In[16]:= Table[i^2,{i,24}]

Out[16]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144,
          169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576}
```

(d)

Zeige mit Sum, dass 100 * 101/2 die Summe der ersten 100 natürlichen Zahlen ist.

- Montre par Sum que 100 * 101/2 est la somme des 100 premiers nombres naturels.

```
In[17]:= s = Sum[i,{i,100}]

Out[17]= 5050

In[18]:= p = 100 101 / 2

Out[18]= 5050

In[19]:= s == p

Out[19]= True
```

(e)

Berechne 1+1/1!+.....+1/10!

- Calcule 1+1/1!+.....+1/10!

In[20]:= **Sum[1/(n!),{n,0,10}]**

Out[20]= $\frac{9864101}{3628800}$

(f)

Berechne 6! mit "Product".
■ Calcule 6! avec "Product".

In[21]:= **Product[n,{n,1,6}]**

Out[21]= 720

In[22]:= **6!**

Out[22]= 720

Aufgabe 7 ■ Problème 7

Berechne die Inverse der 3*3-Hilbert-Matrix. ((i,j) = 1/(i+j-1).)
■ Calcule l'inverse de la matrice de Hilbert 3*3. ((i,j) = 1/(i+j-1).)

In[23]:= **h = Table[1/(i+j-1),{i,3},{j,3}]; MatrixForm[h]**

Out[23]//MatrixForm=

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

In[24]:= **MatrixForm[Inverse[h]]**

Out[24]//MatrixForm=

$$\begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix}$$

Aufgabe 8 ■ Problème 8

Berechne eine numerische Approximation von Pi durch Integration der Funktion
4/(1+x^2) im Intervall [0,1]
■ Calcule une approximation numérique de Pi en intégrant la fonction 4/(1+x^2) dans l'intervalle [0,1]

In[25]:= **f[x_]:=4/(1+x^2); f[x]**

Out[25]= $\frac{4}{1+x^2}$

In[26]:= **Integrate[f[x],x]**

Out[26]= 4 ArcTan[x]

```
In[27]:= Integrate[f[x],{x,0,1}]

Out[27]= π

In[28]:= NIntegrate[f[x],{x,0,1}]

Out[28]= 3.14159
```

Aufgabe 9 ▮ Problème 9

Finde 5 Wurzeln der folgenden Gleichungen
▮ Trouve 5 racines de l'équation suivante

(a) (NRoots geht für Polynome)
▮ (NRoots joue pour des polynômes)

x^5+5x^4+4x^3+3x^2+2x+1=0

```
In[29]:= NRoots[x^5+5x^4+4x^3+3x^2+2x+1==0,x]

Out[29]= x == -4.19273 || x == -0.564099 - 0.390903 i || x == -0.564099 + 0.390903 i ||
          x == 0.160462 - 0.693272 i || x == 0.160462 + 0.693272 i
```

(b) (FindRoot geht für allgemeine Funktionen, liefert jedoch jeweils nur eine Nullstelle)
▮ (FindRoot va pour des fonctions générales, mais ne livre qu'un zéro)

sin^2(Pi x) - x^2 cos(Pi x) = x

```
In[30]:= Table[FindRoot[Sin[Pi x]^2 - x^2 Cos[Pi x] == x,
         {x,x0}],{x0,-5,5}]

Out[30]= {{x → -2.31035}, {x → -2.31035}, {x → 0.849859}, {x → -1.75741}, {x → -2.47887×10^-17},
          {x → 0.}, {x → 1.}, {x → 0.849859}, {x → 1.42393}, {x → 1.42393}, {x → 2.57929}}
```

(c) (Wie in b)
▮ (Comme dans b)

x tan(x) = 1

```
In[31]:= Table[FindRoot[x Tan[x] == 1,
         {x,x0}],{x0,13}]

Out[31]= {{x → 0.860334}, {x → 3.42562}, {x → 3.42562}, {x → 3.42562},
          {x → 6.4373}, {x → 6.4373}, {x → 6.4373}, {x → 9.52933}, {x → 9.52933},
          {x → 9.52933}, {x → 12.6453}, {x → 12.6453}, {x → 12.6453}}
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[32]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[33]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

3. Algebraische und symbolische Stärken
▮ Forces algébriques et symboliques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 3 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 3.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

3.1. Algebra
▮ Algèbre

Mathematica kann algebraische Ausdrücke manipulieren.
▮ *Mathematica* peut manipuler les expressions algébriques

3.1.1. Studiere: ▮ Etudie:

```
In[1]:= ?Apart

        Apart[expr] rewrites a rational expression as a sum of terms with minimal denominators.
           Apart[expr, var] treats all variables other than var as constants. Mehr...

In[2]:= ?Cancel

        Cancel[expr] cancels out common factors in the numerator and denominator of expr. Mehr...

In[3]:= ?Collect

        Collect[expr, x] collects together terms involving the same powers of objects
           matching x. Collect[expr, {x1, x2, ... }] collects together terms that involve
           the same powers of objects matching x1, x2, ... . Collect[expr, var, h] applies
           h to the expression that forms the coefficient of each term obtained. Mehr...
```

In[4]:= **?Expand**

Expand[expr] expands out products and positive integer powers in expr. Expand[expr, patt] leaves unexpanded any parts of expr that are free of the pattern patt. Mehr...

In[5]:= **?ExpandAll**

ExpandAll[expr] expands out all products and integer powers in any part of expr. ExpandAll[expr, patt] avoids expanding parts of expr that do not contain terms matching the pattern patt. Mehr...

In[6]:= **?ExpandDenominator**

ExpandDenominator[expr] expands out products and powers that appear as denominators in expr. Mehr...

In[7]:= **?ExpandNumerator**

ExpandNumerator[expr] expands out products and powers that appear in the numerator of expr. Mehr...

In[8]:= **?Factor**

Factor[poly] factors a polynomial over the integers. Factor[poly, Modulus->p] factors a polynomial modulo a prime p. Factor[poly, Extension->{a1, a2, ... }] factors a polynomial allowing coefficients that are rational combinations of the algebraic numbers ai. Mehr...

In[9]:= **?Simplify**

Simplify[expr] performs a sequence of algebraic and other transformations on expr, and returns the simplest form it finds. Simplify[expr, assum] does simplification using assumptions. Mehr...

In[10]:= **?Short**

Short[expr] prints as a short form of expr, less than about one line long. Short[expr, n] prints as a form of expr about n lines long. Mehr...

In[11]:= **?Together**

Together[expr] puts terms in a sum over a common denominator, and cancels factors in the result. Mehr...

In[12]:=
```
(* Needs["Algebra`Trigonometry`"] *)
(* Old version *)
```

In[14]:= **?TrigExpand**

TrigExpand[expr] expands out trigonometric functions in expr. Mehr...

In[15]:= **?TrigFactor**

TrigFactor[expr] factors trigonometric functions in expr. Mehr...

In[16]:= **?TrigReduce**

TrigReduce[expr] rewrites products and powers of trigonometric functions in expr in terms of trigonometric functions with combined arguments. Mehr...

In[17]:=
```
(* ?ComplexToTrig *)
(* Old version *)
```

```
In[19]:= (* ?*Trig* *)
         (* Old version *)

In[21]:= (* ?TrigToComplex *)
         (* Old version *)

In[23]:= ?*Complex*
```

System`

```
Complex   ComplexExpand   ComplexityFunction
Complexes ComplexInfinity
```

```
In[24]:= ?ExpToTrig

         ExpToTrig[expr] converts exponentials in expr to trigonometric functions. Mehr...

In[25]:= ?TrigToExp

         TrigToExp[expr] converts trigonometric functions in expr to exponentials. Mehr...

In[26]:= ?Sum

         Sum[f, {i, imax}] evaluates the sum of the expressions f as
            evaluated for each i from 1 to imax. Sum[f, {i, imin, imax}] starts with
            i = imin. Sum[f, {i, imin, imax, di}] uses steps di. Sum[f, {i, imin,
            imax}, {j, jmin, jmax}, ... ] evaluates a sum over multiple indices. Mehr...

In[27]:= Sum[i^2,{i, n}]

Out[27]= 1/6 n (1 + n) (1 + 2 n)

In[28]:= (* Needs["Algebra`GosperSum`"] *)
         (* Old: *)

In[30]:= (* ?GosperSum *)
         (* Old: *)

In[32]:= ?*Sum*
```

System`

```
NSum           NSumTerms Sum
NSumExtraTerms RootSum
```

Was in alten Versionen einmal existiert hat, braucht in neuen nicht mehr zu existieren...

Ce qui existait dans de vieilles versions, n'existe pas forcement dans des versions nouvelles.

```
In[33]:= $Version (*Mathematica-Version*)

Out[33]= 5.2 for Microsoft Windows (June 20, 2005)

In[34]:= ?$Version

         $Version is a string that represents the version of Mathematica you are running. Mehr...

In[35]:= (* Needs["Algebra`SymbolicSum`"] *)
         (* Old: *)
```

In[37]:= **Sum[i^2,{i, n}]**

Out[37]= $\frac{1}{6}\, n\, (1+n)\, (1+2\, n)$

In[38]:= **?Limit**

```
Limit[expr, x->x0] finds the limiting value of expr when x approaches x0. Mehr...
```

In[39]:= **?Integrate**

```
Integrate[f, x] gives the indefinite integral of f with respect to x.
   Integrate[f, {x, xmin, xmax}] gives the definite integral of f with respect
   to x from xmin to xmax. Integrate[f, {x, xmin, xmax}, {y, ymin, ymax}]
   gives a multiple definite integral of f with respect to x and y. Mehr...
```

In[40]:= **?Series**

```
Series[f, {x, x0, n}] generates a power series expansion for f
   about the point x = x0 to order (x - x0)^n. Series[f, {x, x0, nx}, {y, y0,
   ny}] successively finds series expansions with respect to y, then x. Mehr...
```

In[41]:= **?DSolve**

```
DSolve[eqn, y, x] solves a differential equation for the function y, with independent
   variable x. DSolve[{eqn1, eqn2, ... }, {y1, y2, ... }, x] solves a list of differential
   equations. DSolve[eqn, y, {x1, x2, ... }] solves a partial differential equation. Mehr...
```

In[42]:= **??DSolve**

```
DSolve[eqn, y, x] solves a differential equation for the function y, with independent
   variable x. DSolve[{eqn1, eqn2, ... }, {y1, y2, ... }, x] solves a list of differential
   equations. DSolve[eqn, y, {x1, x2, ... }] solves a partial differential equation. Mehr...

Attributes[DSolve] = {Protected}

Options[DSolve] = {GeneratedParameters → C}
```

3.1.2. Probiere: ▮ Essaie:

Ueberlege Dir, was durch die folgenden Operationen bewirkt wird!

▮ Réfléchis à ce qui se produit par les opérations suivantes!

In[43]:= **Expand[(x + 2y + z)^4]**

Out[43]= $x^4 + 8\, x^3\, y + 24\, x^2\, y^2 + 32\, x\, y^3 + 16\, y^4 + 4\, x^3\, z + 24\, x^2\, y\, z + 48\, x\, y^2\, z + 32\, y^3\, z + 6\, x^2\, z^2 + 24\, x\, y\, z^2 + 24\, y^2\, z^2 + 4\, x\, z^3 + 8\, y\, z^3 + z^4$

In[44]:= **Factor[%]**

Out[44]= $(x + 2\, y + z)^4$

In[45]:= **Factor[6x^3 + 35 x^2 y + 58 x y^2 + 21 y^3]**

Out[45]= $(2\, x + y)\, (x + 3\, y)\, (3\, x + 7\, y)$

3.2. Gleichungen lösen
▮ Résoudre des équations

3.2.1. Algebraisch exakte Lösung:
▮ Solution exacte algébriquement:

NRoots rechnet eine Näherung. "||" bedeutet dabei das logische "oder", "or".
▮ NRoots calcule une approximation. "||" signifie le "ou" logique.

```
In[46]:= NRoots[x^3 - 3 x^2 - 17 x + 51 == 0, x]

Out[46]= x == -4.12311 || x == 3. || x == 4.12311
```

"||" ist das logische "oder".
▮ "||" est le "ou" logique.

```
In[47]:= ??||

        e1 || e2 || ...  is the logical OR function. It evaluates its arguments in order, giving
           True immediately if any of them are True, and False if they are all False. Mehr...

        Attributes[Or] = {Flat, HoldAll, OneIdentity, Protected}
```

Gewisse algebraische Gleichungen oder Gleichungssysteme kann man jedoch exakt lösen. Dies geschieht mit "Solve".
▮ On peut résoudre exactement certaines équations algébriques ou certains systèmes d'équations. Cela avec "Solve".

```
In[48]:= Solve[x^3 - 3 x^2 - 17 x + 51 == 0, x]

Out[48]= {{x → 3}, {x → -√17}, {x → √17}}
```

Angenähert: ▮ Approximé

```
In[49]:= % // N

Out[49]= {{x → 3.}, {x → -4.12311}, {x → 4.12311}}
```

Uebersichtlicher dargestellt:
▮ Représenté d'une façon plus claire:

```
In[50]:= ColumnForm[Solve[{  x y + 5x + 6y == 7,
                             x^2 + 5x + 7y == 8}]]

Out[50]= {y → 2/7, x → 1}
         {y → -5 - √37, x → -6 - √37}
         {y → -5 + √37, x → -6 + √37}
```

Der Output besteht aus einer Menge von Ersetzungsregeln.
Wie lässt sich damit weiterrechnen?
▮ L'output est composé d'une quantité de règles de remplacement. Comment continue-t-on de calculer avec cela?

3.2.2. Weiterverwendung der Lösung:
▮ Réutilisation de la solution:

"/." bedeutet "ersetze jedes" resp. "ReplaceAll".
▮ "/." signifie "remplace tout" resp. "Remplace All".

```
In[51]:= ?/.

         expr /. rules applies a rule or list of rules in
            an attempt to transform each subpart of an expression expr. Mehr…

In[52]:= ?ReplaceAll

         expr /. rules applies a rule or list of rules in
            an attempt to transform each subpart of an expression expr. Mehr…

In[53]:= ?Replace

         Replace[expr, rules] applies a rule or list of rules in an
            attempt to transform the entire expression expr. Replace[expr, rules,
            levelspec] applies rules to parts of expr specified by levelspec. Mehr…
```

Studiere die folgenden Anwendungen:
▮ Etudie les applications suivantes:

```
In[54]:= a = x /. x -> 7

Out[54]= 7

In[55]:= a

Out[55]= 7

In[56]:= x

Out[56]= x
```

Weiter: ▮ Ensuite:

```
In[57]:= x + 5y /. {x -> 1, y -> 2/7}

Out[57]= 17/7

In[58]:= x + 5y

Out[58]= x + 5 y
```

Weiter: ▮ Encore:

```
In[59]:= Solve[2 x^5 + 12 x^2 + 18x - 14 == 0]

Out[59]= {{x → Root[-7 + 9 #1 + 6 #1^2 + #1^5 &, 1]},
           {x → Root[-7 + 9 #1 + 6 #1^2 + #1^5 &, 2]}, {x → Root[-7 + 9 #1 + 6 #1^2 + #1^5 &, 3]},
           {x → Root[-7 + 9 #1 + 6 #1^2 + #1^5 &, 4]}, {x → Root[-7 + 9 #1 + 6 #1^2 + #1^5 &, 5]}}
```

Die Lösung hier ist nur symbolisch. Eine allgemeine algebraisch exakte Lösungsformel
für Gleichungen mit Polynomen 5. Grades existiert nicht. Approximativ allerdings geht es:

■ La solution n'est que symbolique. Une forme de solution algébrique générale exacte pour les équations avec polynômes du 5e degré n'existe pas. Mais une approximation est possible (existe):

```
In[60]:= N[%]

Out[60]= {{x → 0.561438}, {x → -1.46299 - 0.832383 i},
          {x → -1.46299 + 0.832383 i}, {x → 1.18228 - 1.73288 i}, {x → 1.18228 + 1.73288 i}}
```

3.3. Vereinfachungen
■ Simplifications

3.3.1. Allgemeines
■ Généralités

Mathematica kann nicht die Frage beantworten, welche Outputform eines Ausdrucks für den Verwender der einfachste ist. Man muss den Ausdruck selbst in die gewünschte Form bringen. Dafür gibt es Hilfsmittel wie z.B. "Simplify":
■ *Mathematica* ne sait pas répondre à la question, laquelle des formes de l'Output d'une expression est la plus simple pour l'utilisateur. Il faut donner soi-même la forme souhaitée à l'expression. Pour cela il y a des moyens tels que p.ex. "Simplyfy".

```
In[61]:= Solve[(a d x^3 + 14 b d x^2 + 14^2 c d x)
                      == 0, x] (14b + 14a)/(-c + b)

Out[61]= {{(98 + 14 b) (x → 0)/(b - c)}, {(98 + 14 b) (x → -b - Sqrt[b^2 - 28 c])/(b - c)},
          {(98 + 14 b) (x → -b + Sqrt[b^2 - 28 c])/(b - c)}}

In[62]:= Simplify[%]

Out[62]= {{14 (7 + b) (x → 0)/(b - c)},
          {14 (7 + b) (x → -b - Sqrt[b^2 - 28 c])/(b - c)}, {14 (7 + b) (x → -b + Sqrt[b^2 - 28 c])/(b - c)}}

In[63]:= ??Simplify

        Simplify[expr] performs a sequence of algebraic and
           other transformations on expr, and returns the simplest form it finds.
           Simplify[expr, assum] does simplification using assumptions. Mehr...

        Attributes[Simplify] = {Protected}

        Options[Simplify] = {Assumptions :> $Assumptions, ComplexityFunction → Automatic,
          TimeConstraint → 300, TransformationFunctions → Automatic, Trig → True}

In[64]:= x/((x+2)(x-2))

Out[64]= x/((-2 + x) (2 + x))

In[65]:= Simplify[%]

Out[65]= x/(-4 + x^2)
```

```
In[66]:= x/Expand[((x+2)(x-2))]
```

$$\text{Out[66]}= \frac{x}{-4+x^2}$$

```
In[67]:= Expand[x/((x+2)(x-2))]
```

$$\text{Out[67]}= \frac{x}{(-2+x)(2+x)}$$

```
In[68]:= x/Simplify[((x+2)(x-2))]
```

$$\text{Out[68]}= \frac{x}{-4+x^2}$$

3.3.2. Andere Vorgehensweisen:
▮ Autres façons de procéder

Beispiel: Partialbruchzerlegung mit "Apart". Probiere aus.:
▮ Exemple: Décomposition de fractions partielles en utilisant "Apart":

```
In[69]:= x/((x+2)(x-2))
```

$$\text{Out[69]}= \frac{x}{(-2+x)(2+x)}$$

```
In[70]:= Apart[%]
```

$$\text{Out[70]}= \frac{1}{2(-2+x)} + \frac{1}{2(2+x)}$$

Retour: ▮ Retour:

```
In[71]:= Together[%]
```

$$\text{Out[71]}= \frac{x}{(-2+x)(2+x)}$$

Man beachte die Reihenfolge der Terme im Output. Zuerst Konstanten, dann Symbole, alphabetisch, Terme nach steigendem Grad etc.:
▮ Il faut tenir compte des termes dans le output. D'abort les constantes, puis les symboles, en ordre alphabétique, termes selon le dégré en augmentant etc.:

```
In[72]:= Expand[(1 + 2y + 3z)^3]
```

$$\text{Out[72]}= 1 + 6y + 12y^2 + 8y^3 + 9z + 36yz + 36y^2z + 27z^2 + 54yz^2 + 27z^3$$

Umordnen: ▮ Réordonner:

```
In[73]:= Collect[%, y]
```

$$\text{Out[73]}= 1 + 8y^3 + 9z + 27z^2 + 27z^3 + y^2(12 + 36z) + y(6 + 36z + 54z^2)$$

3.3.3. Trigonometrische Ausdrücke:
▮ Expressions trigonométriques:

```
In[74]:= Expand[Sin[x]/Cos[x]]

Out[74]= Tan[x]

In[75]:= Expand[(Sin[x]+Cos[x])/Sin[x] (Sin[x]^2 + Cos[x]^2), Trig -> False]

Out[75]= Cos[x]^2 + Cos[x]^2 Cot[x] + Cos[x] Sin[x] + Sin[x]^2

In[76]:= ?Trig

        Trig is an option for algebraic manipulation functions which specifies whether
          trigonometric functions should be treated as rational functions of exponentials.

In[77]:= Expand[(Sin[x]+Cos[x])/Sin[x] (Sin[x]^2 + Cos[x]^2), Trig -> True]

Out[77]= 1 + Cot[x]

In[78]:= (* Old *)
         TrigToComplex[Sin[x]]

Out[79]= TrigToComplex[Sin[x]]

In[80]:= ?TrigToComplex

       Global`TrigToComplex

In[81]:= TrigToExp[Sin[x]]

Out[81]= 1/2 i ℯ^(-i x) - 1/2 i ℯ^(i x)

In[82]:= (* Needs["Algebra`Trigonometry`"] *)
         (* Old *)

In[84]:= ?*Trig*
```

System`

```
ExpToTrig TrigExpand TrigFactorList TrigToExp
Trig      TrigFactor TrigReduce
```

Global`

```
TrigToComplex
```

3.3.4. Lange Resultate:
▮ Résultats longs

Mathematica bietet die Möglichkeit, Resultate abzukürzen:

▮ *Mathematica* offre la possibilité d'abréger les résultats:

In[85]:= **Short[Expand[(x + 2y + 3z)^9]]**

Out[85]//Short=

$x^9 + 18\, x^8\, y + \ll 51 \gg + 118098\, y\, z^8 + 19683\, z^9$

In[86]:= **?Short**

```
Short[expr] prints as a short form of expr, less than about one line
   long. Short[expr, n] prints as a form of expr about n lines long.
```
Mehr…

Mit der globalen Variablen "$PrePrint" lässt sich das Abkürzen der Ausgabe generalisieren:

■ Avec la variable globale "$PrePrint" on peut généralement abréger la sortie:

In[87]:= **$PrePrint = Short**

Out[87]= Short

In[88]:= **Expand[(x + 2y + 3z)^9]**

Out[88]= $x^9 + 18\, x^8\, y + \ll 51 \gg + 118098\, y\, z^8 + 19683\, z^9$

In[89]:= **Expand[(x + 2y + 3z)^11]**

Out[89]= $x^{11} + 22\, x^{10}\, y + \ll 74 \gg + 1299078\, y\, z^{10} + 177147\, z^{11}$

In[90]:= **Expand[(x + 2y + 3z)^5]**

Out[90]= $x^5 + 10\, x^4\, y + 40\, x^3\, y^2 + \ll 16 \gg + 810\, y\, z^4 + 243\, z^5$

Trotzdem kann man das ganze Resultat abrufen mit "Print":

■ En tout cas on peut appeler le résultat entier avec "Print":

In[91]:= **Print[%]**

$$x^5 + 10\, x^4\, y + 40\, x^3\, y^2 + 80\, x^2\, y^3 + 80\, x\, y^4 + 32\, y^5 + 15\, x^4\, z + 120\, x^3\, y\, z + 360\, x^2\, y^2\, z + 480\, x\, y^3\, z + 240\, y^4\, z + 90\, x^3\, z^2 + 540\, x^2\, y\, z^2 + 1080\, x\, y^2\, z^2 + 720\, y^3\, z^2 + 270\, x^2\, z^3 + 1080\, x\, y\, z^3 + 1080\, y^2\, z^3 + 405\, x\, z^4 + 810\, y\, z^4 + 243\, z^5$$

Alter Zustand: ■ Ancien état:

In[92]:= **$PrePrint = .**

In[93]:= **Expand[(x + 2y + 3z)^9]**

Out[93]=

$$x^9 + 18\, x^8\, y + 144\, x^7\, y^2 + 672\, x^6\, y^3 + 2016\, x^5\, y^4 + 4032\, x^4\, y^5 + 5376\, x^3\, y^6 + 4608\, x^2\, y^7 + 2304\, x\, y^8 + 512\, y^9 + 27\, x^8\, z + 432\, x^7\, y\, z + 3024\, x^6\, y^2\, z + 12096\, x^5\, y^3\, z + 30240\, x^4\, y^4\, z + 48384\, x^3\, y^5\, z + 48384\, x^2\, y^6\, z + 27648\, x\, y^7\, z + 6912\, y^8\, z + 324\, x^7\, z^2 + 4536\, x^6\, y\, z^2 + 27216\, x^5\, y^2\, z^2 + 90720\, x^4\, y^3\, z^2 + 181440\, x^3\, y^4\, z^2 + 217728\, x^2\, y^5\, z^2 + 145152\, x\, y^6\, z^2 + 41472\, y^7\, z^2 + 2268\, x^6\, z^3 + 27216\, x^5\, y\, z^3 + 136080\, x^4\, y^2\, z^3 + 362880\, x^3\, y^3\, z^3 + 544320\, x^2\, y^4\, z^3 + 435456\, x\, y^5\, z^3 + 145152\, y^6\, z^3 + 10206\, x^5\, z^4 + 102060\, x^4\, y\, z^4 + 408240\, x^3\, y^2\, z^4 + 816480\, x^2\, y^3\, z^4 + 816480\, x\, y^4\, z^4 + 326592\, y^5\, z^4 + 30618\, x^4\, z^5 + 244944\, x^3\, y\, z^5 + 734832\, x^2\, y^2\, z^5 + 979776\, x\, y^3\, z^5 + 489888\, y^4\, z^5 + 61236\, x^3\, z^6 + 367416\, x^2\, y\, z^6 + 734832\, x\, y^2\, z^6 + 489888\, y^3\, z^6 + 78732\, x^2\, z^7 + 314928\, x\, y\, z^7 + 314928\, y^2\, z^7 + 59049\, x\, z^8 + 118098\, y\, z^8 + 19683\, z^9$$

3.4. Summation
▮ Sommation

Mathematica bietet die Möglichkeit, z.B. Summen mit variablem oberen Summationsindex zu berechnen:

▮ *Mathematica* offre la possibilité p.ex. de calculer des sommes avec un index de sommation supérieur variable:

```
In[94]:= ?*Sum*
```

System`

```
NSum            NSumTerms Sum
NSumExtraTerms RootSum
```

```
In[95]:= ?Sum

         Sum[f, {i, imax}] evaluates the sum of the expressions f as
            evaluated for each i from 1 to imax. Sum[f, {i, imin, imax}] starts with
            i = imin. Sum[f, {i, imin, imax, di}] uses steps di. Sum[f, {i, imin,
            imax}, {j, jmin, jmax}, ... ] evaluates a sum over multiple indices. Mehr...

In[96]:= (* Needs["Algebra`SymbolicSum`"] *)
         (* Old *)

In[98]:= Sum[i^3,{i, n}]

Out[98]= 1/4 n^2 (1 + n)^2
```

Probiere eigene Aufgaben aus:

▮ Essaie des problèmes que tu inventes toi-même:

```
In[99]:=
```

3.5. Calculus (Differential- und Integralrechnung)
▮ Calculus (Calcul différentiel et intégral)

3.5.1. Unbestimmte Integrale
▮ Intégrals indéterminées

Beispiele: ▮ Exemple:

```
In[99]:= Integrate[Cos[x], x]

Out[99]= Sin[x]
```

In[100]:=
```
Integrate[x^4 Cos[x], x]
```
Out[100]=
$$4\,x\,(-6+x^2)\,\text{Cos}[x]+(24-12\,x^2+x^4)\,\text{Sin}[x]$$

In[101]:=
```
1/(2 + 3 x^2)^3
```
Out[101]=
$$\frac{1}{(2+3\,x^2)^3}$$

In[102]:=
```
Integrate[1/(2 + 3 x^2)^3, x]
```
Out[102]=
$$\frac{x}{8\,(2+3\,x^2)^2}+\frac{3\,x}{32\,(2+3\,x^2)}+\frac{1}{32}\sqrt{\frac{3}{2}}\,\text{ArcTan}\left[\sqrt{\frac{3}{2}}\,x\right]$$

Kontrolle: Differenzieren!

■ Contrôle: Différencier!

In[103]:=
```
D[%, x]
```
Out[103]=
$$\frac{3}{64\,\left(1+\frac{3\,x^2}{2}\right)}-\frac{3\,x^2}{2\,(2+3\,x^2)^3}+\frac{1}{8\,(2+3\,x^2)^2}-\frac{9\,x^2}{16\,(2+3\,x^2)^2}+\frac{3}{32\,(2+3\,x^2)}$$

Das Resultat kann durchaus anders aussehen. Es muss vielleicht umgeformt werden.

■ Le résultat peut être différent. Il faut peut-être le transformer.

In[104]:=
```
Simplify[%]
```
Out[104]=
$$\frac{1}{(2+3\,x^2)^3}$$

3.5.2. Bestimmte Integrale
■ Intégrales déterminées

Mathematica kann bestimmte Integrale exakt rechnen. Probiere aus:

■ *Mathematica* sait calculer exactement des intégrales déterminées. Essaie:

In[105]:=
```
?Integrate
```
```
Integrate[f, x] gives the indefinite integral of f with respect to x.
   Integrate[f, {x, xmin, xmax}] gives the definite integral of f with respect
   to x from xmin to xmax. Integrate[f, {x, xmin, xmax}, {y, ymin, ymax}]
   gives a multiple definite integral of f with respect to x and y. Mehr...
```

In[106]:=
```
?*Integ*
```

System`

```
CoshIntegral     Integer         Integrate
CosIntegral      IntegerDigits   LogIntegral
ExpIntegralE     IntegerExponent NIntegrate
ExpIntegralEi    IntegerPart     SinhIntegral
FactorInteger    IntegerQ        SinIntegral
GaussianIntegers Integers
```

In[107]:=
```
Integrate[Exp[x],{x, -1, 1}]
```

Out[107]=

$$-\frac{1}{e}+e$$

In[108]:=
```
N[%]
```

Out[108]=
```
2.3504
```

Numerisch geht es natürlich auch. Probiere aus:

- Il va de soi qu'on peut le faire numériquement. Essaie:

In[109]:=
```
NIntegrate[Exp[x],{x, -1, 1}]
```

Out[109]=
```
2.3504
```

Zu gewissen Funktionen kann keine einfache Stammfunktion gefunden werden:

- Pour certaines fonctions on ne trouve pas de fonction primitive (intégrale) simple:

In[110]:=
```
Integrate[Sin[Sin[x]],{x, 0, 1}]
```

Out[110]=

$$\int_0^1 \text{Sin}[\text{Sin}[x]]\,dx$$

Numerisch geht es natürlich. Probiere aus:

- Numériquement ça va. Essaie:

In[111]:=
```
NIntegrate[Sin[Sin[x]],{x, 0, 1}]
```

Out[111]=
```
0.430606
```

3.5.3. Integrale von Funktionen mit Polen
▮ Intégrales de fonctions avec pôles

Bei Funktionen mit Polen können Fehler auftreten, wenn man einfach eine Stammfunktion bestimmt und die Grenzen einsetzt. Prüfe:

▮ Lors de fonctions avec pôles, des erreurs peuvent appataître, si on détermine simplement une fonction primitive (intégrale) et fixe les limites. Examine:

In[112]:=

```
Integrate[1/(x-Pi/100)^2,{x, -1, 1}]
```

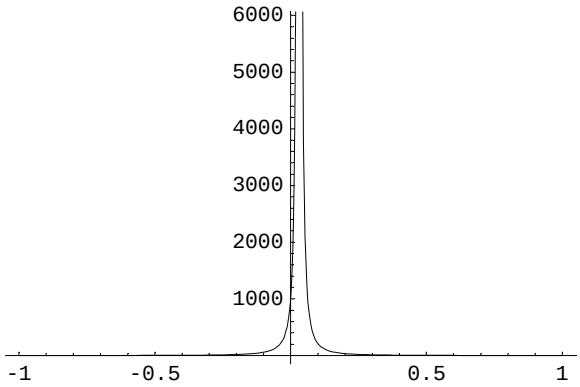
```
Integrate::idiv :
 Integral of FractionBox["1", SuperscriptBox[RowBox[{"(", RowBox[{RowBox[{"-", FractionBox["π", RowBox[
   {"", \(≪ 3 ≫\), ""}]]}], "+", "x"}], ")"}], "2"]] does not converge on {-1, 1}. Mehr…
```

Out[112]=

$$\int_{-1}^{1} \frac{1}{\left(-\frac{\pi}{100}+x\right)^2}\,dx$$

In[113]:=

```
Plot[1/(x-Pi/100)^2,{x, -1, 1}];
```



In[114]:=

```
NIntegrate[1/(x-Pi/100)^2,{x, -1, 1}]
```

```
NIntegrate::ncvb : NIntegrate failed to converge to
   prescribed accuracy after 7 recursive bisections in x near x = 0.0390625`. Mehr…
```

Out[114]=

```
200852.
```

In[115]:=

```
Integrate[1/x^2,{x, -1, 1}]
```

```
Integrate::idiv : Integral of 1/x^2 does not converge on {-1, 1}. Mehr…
```

Out[115]=

$$\int_{-1}^{1} \frac{1}{x^2}\,dx$$

In[116]:=
```
NIntegrate[1/x^2,{x, -1, 1}]
```

```
NIntegrate::slwcon :
 Numerical integration converging too slowly; suspect one of the following: singularity, value of
   the integration being 0, oscillatory integrand, or insufficient WorkingPrecision. If your
   integrand is oscillatory try using the option Method->Oscillatory in NIntegrate. Mehr…

NIntegrate::ncvb : NIntegrate failed to converge to prescribed
   accuracy after 7 recursive bisections in x near x = 4.369993747903698`*^-57. Mehr…
```

Out[116]=
$$2.481310356720057 \times 10^{3498}$$

In[117]:=
```
??NIntegrate
```

```
NIntegrate[f, {x, xmin, xmax}] gives a numerical approximation
   to the integral of f with respect to x from xmin to xmax. Mehr…

Attributes[NIntegrate] = {HoldAll, Protected}

Options[NIntegrate] =
 {AccuracyGoal → ∞, Compiled → True, EvaluationMonitor → None, GaussPoints → Automatic,
  MaxPoints → Automatic, MaxRecursion → 6, Method → Automatic, MinRecursion → 0,
  PrecisionGoal → Automatic, SingularityDepth → 4, WorkingPrecision → MachinePrecision}
```

Es kann z.B. angegeben werden, dass bei 0 eine Singularität (Polstelle) ist. Prüfe:

■ On peut indiquer p.ex. qu'il y a une singularité (pôle) près de 0. Examine:

In[118]:=
```
NIntegrate[1/x^2,{x, -1, 0, 1}]
```

```
NIntegrate::slwcon :
 Numerical integration converging too slowly; suspect one of the following: singularity, value of
   the integration being 0, oscillatory integrand, or insufficient WorkingPrecision. If your
   integrand is oscillatory try using the option Method->Oscillatory in NIntegrate. Mehr…

NIntegrate::ncvb : NIntegrate failed to converge to prescribed
   accuracy after 7 recursive bisections in x near x = 4.369993747903698`*^-57. Mehr…
```

Out[118]=
$$2.481310356720057 \times 10^{3498}$$

3.5.4. Mehrfachintegrale
■ Intégrales multiples

f(x, y) kann über x und y integriert werden. Probiere:

■ f(x, y) peut être intégré sur x et y. Essaie:

In[119]:=
```
Integrate[x^2 Sin[y],x, y]
```

Out[119]=
$$-\frac{1}{3} x^3 \text{Cos}[y]$$

Bei bestimmten Integralen sind die äusseren Grenzen zuerst anzugeben:

■ Pour certains intégrales les limites extérieurs doivent êtres données d'abort.

In[120]:=
```
Integrate[x^2 Sin[y],{x, 0, Pi/3}, {y, 0, x}]
```

Out[120]=
$$\sqrt{3} - \frac{\pi}{3} - \frac{\pi^2}{6\sqrt{3}} + \frac{\pi^3}{81}$$

3.6. Grenzwerte
▮ Valeurs limites

Mit "Limits" lassen sich Grenzwerte berechnen. "Unendlich" ist "Infinity":
▮ Avec "Limits" on peut calculer des valeurs limites. "Infinie" est "Infinity":

In[121]:=
```
?Limit
```
```
Limit[expr, x->x0] finds the limiting value of expr when x approaches x0. Mehr...
```

In[122]:=
```
?Infinity
```
```
Infinity is a symbol that represents a positive infinite quantity. Mehr...
```

In[123]:=
```
Limit[Sin[x]/x, x -> 0]
```

Out[123]=
```
1
```

In[124]:=
```
Limit[1/x, x -> 0]
```

Out[124]=
$$\infty$$

Der Wert des Grenzwerts kann davon abhängen, aus welcher Richtung man sich dem problematischen Punkt nähert. Die Richtung kann angegeben werden:
▮ La valeur de la valeur limite peut dépendre de la direction de laquelle on s'approche du point problématique.

In[125]:=
```
Limit[1/x, x -> 0, Direction -> 1]
```

Out[125]=
$$-\infty$$

In[126]:=
```
Limit[1/x, x -> 0, Direction -> -1]
```

Out[126]=
$$\infty$$

In[127]:=
```
Integrate[E^(-s t) t^n,{t,0,Infinity}]
```

Out[127]=
```
If[Re[n] > -1 && Re[s] > 0, s^(-1-n) Gamma[1 + n],
 Integrate[e^(-s t) t^n, {t, 0, ∞}, Assumptions → Re[s] ≤ 0 || 1 + Re[n] ≤ 0]]
```

In[128]:=
```
Integrate[E^(-s t) Sin[w t],{t,0,Infinity}]
```

Out[128]=
$$\text{If}\left[w \in \text{Reals} \&\& \text{Re}[s] > 0,\ \frac{w}{s^2 + w^2},\ \text{Integrate}[e^{-s\,t}\,\text{Sin}[t\,w],\ \{t,\ 0,\ \infty\},\ \text{Assumptions} \to \text{Re}[s] \le 0 \,||\, w \notin \text{Reals}]\right]$$

3.7. Potenzreihen
∎ Séries de puissances

Beispiele von Potenzreihenentwicklungen:
∎ Exemples de développements de séries de puissances:

In[129]:=
```
Series[f[x],{x,0,9}]
```

Out[129]=
$$f[0] + f'[0]\,x + \frac{1}{2}\,f''[0]\,x^2 + \frac{1}{6}\,f^{(3)}[0]\,x^3 + \frac{1}{24}\,f^{(4)}[0]\,x^4 + \frac{1}{120}\,f^{(5)}[0]\,x^5 + \frac{1}{720}\,f^{(6)}[0]\,x^6 + \frac{f^{(7)}[0]\,x^7}{5040} + \frac{f^{(8)}[0]\,x^8}{40320} + \frac{f^{(9)}[0]\,x^9}{362880} + O[x]^{10}$$

In[130]:=
```
Series[f[x],{x,1,6}]
```

Out[130]=
$$f[1] + f'[1]\,(x-1) + \frac{1}{2}\,f''[1]\,(x-1)^2 + \frac{1}{6}\,f^{(3)}[1]\,(x-1)^3 + \frac{1}{24}\,f^{(4)}[1]\,(x-1)^4 + \frac{1}{120}\,f^{(5)}[1]\,(x-1)^5 + \frac{1}{720}\,f^{(6)}[1]\,(x-1)^6 + O[x-1]^7$$

In[131]:=
```
Series[E^x,{x,0,9}]
```

Out[131]=
$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40320} + \frac{x^9}{362880} + O[x]^{10}$$

In[132]:=
```
Series[Cos[x],{x,0,9}]
```

Out[132]=
$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} + O[x]^{10}$$

Solch ein Output ist noch kein algebraischer Ausdruck, mit dem weitergerechnet werden
kann, denn die angegebene Ordnung der nachfolgenden Glieder ist kein Term. Mit "Normal" kann diese Angabe entfernt werden. Probiere:
Un tel Output n'est pas encore une expression algébrique, avec laquelle on peut continuer de calculer, car l'ordre donné des membres suivants n'est pas un terme. On peut effacer ce problème par "Normal". Essaie:

In[133]:=
```
a = Normal[Series[E^x,{x,0,9}]]
```

Out[133]=
$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40320} + \frac{x^9}{362880}$$

In[134]:=
```
b = Normal[Series[Cos[x],{x,0,9}]]
```

Out[134]=
$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320}$$

In[135]:=
```
b = Normal[Series[Cos[x],{x,0,9}]]
```

Out[135]=
$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320}$$

In[136]:=
```
b /. x->1
```

Out[136]=
$$\frac{4357}{8064}$$

In[137]:=
```
a + b
```

Out[137]=
$$2 + x + \frac{x^3}{6} + \frac{x^4}{12} + \frac{x^5}{120} + \frac{x^7}{5040} + \frac{x^8}{20160} + \frac{x^9}{362880}$$

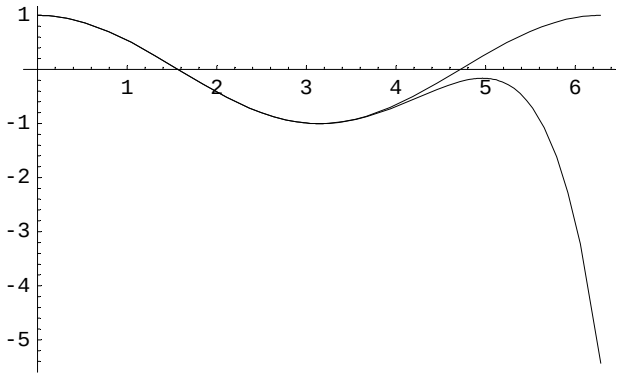
Etwas fürs Auge: ■ Quelque chose pour l'oeil:

In[138]:=
```
Clear[a]
```

In[139]:=
```
prCos[x_]:= Normal[Series[Cos[x],{x,0,11}]];
p = prCos[x]
```

Out[140]=
$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} - \frac{x^{10}}{3628800}$$

In[141]:=
```
Plot[{p, Cos[x]}, {x, 0, 2 Pi}];
```



In[142]:=

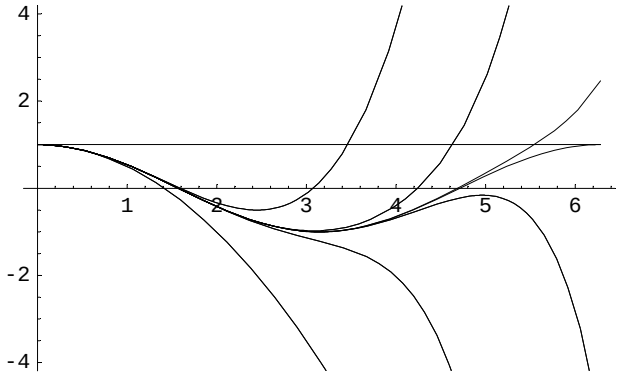
```
a = Append[Evaluate[Table[Normal[Series[Cos[x],{x,0,n}]],{n, 12}]],
Cos[x]]
```

Out[142]=

$$\left\{1, 1-\frac{x^2}{2}, 1-\frac{x^2}{2}, 1-\frac{x^2}{2}+\frac{x^4}{24}, 1-\frac{x^2}{2}+\frac{x^4}{24}, 1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720},\right.$$
$$1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720}, 1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720}+\frac{x^8}{40320}, 1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720}+\frac{x^8}{40320},$$
$$1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720}+\frac{x^8}{40320}-\frac{x^{10}}{3628800}, 1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720}+\frac{x^8}{40320}-\frac{x^{10}}{3628800},$$
$$\left.1-\frac{x^2}{2}+\frac{x^4}{24}-\frac{x^6}{720}+\frac{x^8}{40320}-\frac{x^{10}}{3628800}+\frac{x^{12}}{479001600}, \text{Cos}[x]\right\}$$

In[143]:=

```
Plot[Evaluate[Table[a[[i]],{i,13}]], {x, 0, 2 Pi}];
```



In[144]:=

```
??Append
```

```
Append[expr, elem] gives expr with elem appended. Mehr...

Attributes[Append] = {Protected}
```

3.8. Lösen von Differentialgleichungen
∎ Résoudre des équations différentielles

Mathematica kann gewisse einfache Differentialgleichungen exakt lösen. Probiere:

∎ *Mathematica* peut résoudre avec exactitude certaines équations différentielles. Essaie:

In[145]:=

```
?DSolve
```

```
DSolve[eqn, y, x] solves a differential equation for the function y, with independent
   variable x. DSolve[{eqn1, eqn2, ... }, {y1, y2, ... }, x] solves a list of differential
   equations. DSolve[eqn, y, {x1, x2, ... }] solves a partial differential equation. Mehr...
```

In[146]:=

```
DSolve[y''[x] + y'[x] + y[x] == 0, y[x], x]
```

Out[146]=

$$\left\{\left\{y[x] \to e^{-x/2}\, C[2]\, \text{Cos}\left[\frac{\sqrt{3}\, x}{2}\right] + e^{-x/2}\, C[1]\, \text{Sin}\left[\frac{\sqrt{3}\, x}{2}\right]\right\}\right\}$$

"C[..]" bedeutet eine Konstante. Auch Systeme können behandelt werden. Probiere:
■ "C[..]" signifie une constante. On peut aussi traiter des systèmes. Essaie:

In[147]:=
```
d = DSolve[{y'[x] + y[x] == 0, y[2] == 2}, y[x], x]
```

Out[147]=
$$\{\{y[x] \to 2\, e^{2-x}\}\}$$

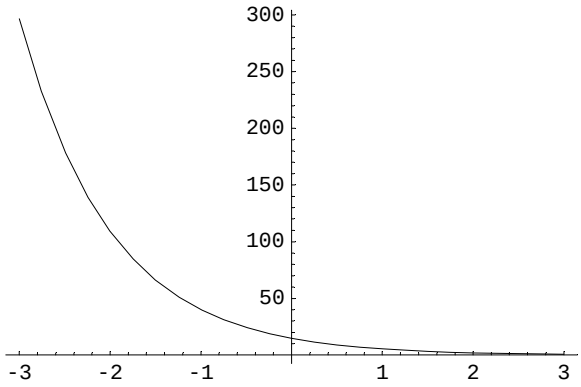
In[148]:=
```
dd =Flatten[d][[1]]
```

Out[148]=
$$y[x] \to 2\, e^{2-x}$$

In[149]:=
```
ddd = dd[[2]]
```

Out[149]=
$$2\, e^{2-x}$$

In[150]:=
```
Plot[ddd, {x, -3, 3}];
```



Aufgabe:Behandle das System {r y'[x] + y[x] == 0, y[2] == 2} für
r = -3, -2, -1, 0, 1, 2, 3 und vergleiche die Graphen.
■ Problème: Traite le système {r y'[x] + y[x] == 0, y[2] == 2} pour
r = -3, -2, -1, 0, 1, 2, 3 et compare les graphes.

"Putzmaschine" einsetzen
■ Employer la "machine de nettoyage"

In[151]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[152]:=
```
Remove["Global`*"]
```

Uebungen ▮ Exercices

3. Algebraische und symbolische Stärken ▮ Forces algébriques et symboliques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 3 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 3.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

(a)

Löse folgende Gleichung: ▮ Résous l'équation suivante:

```
In[1]:= Solve[x^2 + 2x + 1 == 0]
Out[1]= {{x → -1}, {x → -1}}
```

(b)

Löse folgende Gleichung: ▮ Résous l'équation suivante:

```
In[2]:= Solve[ax^2 + bx + c == 0,x]
Out[2]= {{}}
```

(c)

Löse folgende Gleichung: ■ Résous l'équation suivante:

```
In[3]:= Solve[{x + y== 5, 2x + 6y == 23}]
```

$$\text{Out[3]= } \left\{\left\{x \to \frac{7}{4},\ y \to \frac{13}{4}\right\}\right\}$$

Aufgabe 2 ■ Problème 2

(a)

Faktorisiere das Polynom 1-x^n. Entdeckst Du ein Gesetz?
■ Factorise le pôlynome 1-x^n. Découvres-tu une loi?

```
In[4]:= Do[Print[1 - x^n,"==",Factor[1 - x^n]],{n, 2, 9}]
```

$$1 - x^2 == -(-1 + x)\,(1 + x)$$

$$1 - x^3 == -(-1 + x)\,(1 + x + x^2)$$

$$1 - x^4 == -(-1 + x)\,(1 + x)\,(1 + x^2)$$

$$1 - x^5 == -(-1 + x)\,(1 + x + x^2 + x^3 + x^4)$$

$$1 - x^6 == -(-1 + x)\,(1 + x)\,(1 - x + x^2)\,(1 + x + x^2)$$

$$1 - x^7 == -(-1 + x)\,(1 + x + x^2 + x^3 + x^4 + x^5 + x^6)$$

$$1 - x^8 == -(-1 + x)\,(1 + x)\,(1 + x^2)\,(1 + x^4)$$

$$1 - x^9 == -(-1 + x)\,(1 + x + x^2)\,(1 + x^3 + x^6)$$

(b)

Faktorisiere das Polynom 1-x^11
■ Factorise le pôlynome 1-x^11.

```
In[5]:= Factor[1 - x^11]
```

$$\text{Out[5]= } -(-1 + x)\,(1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10})$$

(c)

Faktorisiere das Polynom 1-x^..... (Zahl einsetzen!)
■ Factorise le pôlynome 1-x^..... (Mettez un nombre!)

```
In[6]:= Factor[1 - x^13]
```

$$\text{Out[6]= } -(-1 + x)\,(1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12})$$

Aufgabe 3 ■ Problème 3

(a)

Mache mit "Apart" eine Partialbruchzerlegung:
■ Fais avec "Apart" une décomposition d'une fraction rationelle en éléments simples:

```
In[7]:= Apart[x/(x^2 + 5x + 6)]
```

$$\text{Out[7]}= -\frac{2}{2+x}+\frac{3}{3+x}$$

(a)

Mache mit "Apart" eine Partialbruchzerlegung:
■ Fais avec "Apart" une décomposition d'une fraction rationelle en éléments simples:

```
In[8]:= Apart[(2x + 7)/(x^3 + 3x^2 + 3x + 1)]
```

$$\text{Out[8]}= \frac{5}{(1+x)^3}+\frac{2}{(1+x)^2}$$

Aufgabe 4 ■ Problème 4

(a)

Vereinfachung trigonometrischer Funktionen: Setzt Trig->True:
■ Simplifications de fonctions trigonométriques: Mettez Trig->True:

```
In[9]:= Expand[Cos[x] Cos[y] + Sin[x] Sin[y],Trig->True]
```

```
Out[9]= Cos[x] Cos[y] + Sin[x] Sin[y]
```

```
In[10]:= Factor[Cos[x] Cos[y] + Sin[x] Sin[y],Trig->True]
```

```
Out[10]= Cos[x - y]
```

(b)

Vereinfachung trigonometrischer Funktionen: Setzt Trig->True:
■ Simplifications de fonctions trigonométriques: Mettez Trig->True:

```
In[11]:= Expand[Cos[x] Sin[y] + Sin[x] Cos[y],Trig->True]
```

```
Out[11]= Cos[y] Sin[x] + Cos[x] Sin[y]
```

In[12]:=
```
Expand[Cos[x] Sin[y] + Sin[x] Cos[y],Trig->True]
```

Out[12]=
```
Cos[y] Sin[x] + Cos[x] Sin[y]
```

(c)

Vereinfachung von Exponentialfunktionen: Setzt Trig->True:

■ Simplifications de fonctions exponentielles: Mettez Trig->True:

In[13]:=
```
Expand[E^(-I x) + E^(I x),Trig->True]
```

Out[13]=
$$e^{-i x} + e^{i x}$$

In[14]:=
```
Factor[E^(-I x) + E^(I x),Trig->True]
```

Out[14]=
$$e^{-i x} \left(1 + e^{2 i x}\right)$$

Aufgabe 5 ■ Problème 5

(a)

Generiere mit "Array" eine 2 x 2-Matrix mit den Elementen b[i, j]:

■ Génère avec "Array" une matrice 2 x 2 avec les éléments b[i, j]:

In[15]:=
```
Clear[b]; m = Array[b,{2,2}]
```

Out[15]=
```
{{b[1, 1], b[1, 2]}, {b[2, 1], b[2, 2]}}
```

(b)

Invertiere und Transponiere m:

■ Invertis et transforme m:

In[16]:=
```
Inverse[m]
```

Out[16]=
$$\left\{\left\{\frac{b[2,2]}{-b[1,2]\,b[2,1] + b[1,1]\,b[2,2]}, -\frac{b[1,2]}{-b[1,2]\,b[2,1] + b[1,1]\,b[2,2]}\right\},\right.$$
$$\left.\left\{-\frac{b[2,1]}{-b[1,2]\,b[2,1] + b[1,1]\,b[2,2]}, \frac{b[1,1]}{-b[1,2]\,b[2,1] + b[1,1]\,b[2,2]}\right\}\right\}$$

In[17]:=
```
Transpose[m]
```

Out[17]=
```
{{b[1, 1], b[2, 1]}, {b[1, 2], b[2, 2]}}
```

(c)

Generiere mit "Array" eine 2 x 2-Matrix mit den Elementen c[i, j]:

■ Génère avec "Array" une matrice 2 x 2 avec les éléments c[i, j]:

```
In[18]:= Clear[c]; n = Array[c,{2,2}]

Out[18]= {{c[1, 1], c[1, 2]}, {c[2, 1], c[2, 2]}}

In[19]:= MatrixForm[n]

Out[19]//MatrixForm=
        ( c[1, 1]  c[1, 2] )
        ( c[2, 1]  c[2, 2] )
```

(d)

Berechne m n und m.n . Was ist der Unterschied?

■ Calcule m n et m.n . Quelle est la différence?

```
In[20]:= m n

Out[20]= {{b[1, 1] c[1, 1], b[1, 2] c[1, 2]}, {b[2, 1] c[2, 1], b[2, 2] c[2, 2]}}

In[21]:= m.n

Out[21]= {{b[1, 1] c[1, 1] + b[1, 2] c[2, 1], b[1, 1] c[1, 2] + b[1, 2] c[2, 2]},
          {b[2, 1] c[1, 1] + b[2, 2] c[2, 1], b[2, 1] c[1, 2] + b[2, 2] c[2, 2]}}

In[22]:= MatrixForm[m n]

Out[22]//MatrixForm=
        ( b[1, 1] c[1, 1]  b[1, 2] c[1, 2] )
        ( b[2, 1] c[2, 1]  b[2, 2] c[2, 2] )

In[23]:= MatrixForm[m.n]

Out[23]//MatrixForm=
        ( b[1, 1] c[1, 1] + b[1, 2] c[2, 1]  b[1, 1] c[1, 2] + b[1, 2] c[2, 2] )
        ( b[2, 1] c[1, 1] + b[2, 2] c[2, 1]  b[2, 1] c[1, 2] + b[2, 2] c[2, 2] )
```

Aufgabe 6 ■ Problème 6

(a)

Integration eines Ausdrucks:

■ Intégration d'une expression:

```
In[24]:= w = 1/(x^3 + 1)

Out[24]= 1/(1 + x^3)

In[25]:= u = Integrate[1/(x^3 + 1),x]

Out[25]= ArcTan[(-1+2 x)/Sqrt[3]]/Sqrt[3] + 1/3 Log[1 + x] - 1/6 Log[1 - x + x^2]
```

(b)

Differenziere das Resultat:
■ Différencie le résultat:

```
In[26]:= v = D[%,x]
```

$$\text{Out[26]}= \frac{1}{3\,(1+x)} - \frac{-1+2x}{6\,(1-x+x^2)} + \frac{2}{3\,\left(1+\frac{1}{3}\,(-1+2x)^2\right)}$$

(c)

Hat man nun wieder den ursprünglichen Ausdruck?
■ A-t-on de nouveau l'expression d'origine?

```
In[27]:= Together[v]
```

$$\text{Out[27]}= \frac{1}{(1+x)\,(1-x+x^2)}$$

```
In[28]:= v=ExpandAll[Together[v]]
```

$$\text{Out[28]}= \frac{1}{1+x^3}$$

```
In[29]:= w === v
Out[29]= True
```

Aufgabe 7 ■ Problème 7

(a)

Exakter Wert eines Integrals :
■ Valeur exacte d'une intégrale:

```
In[30]:= Integrate[x y^2,{x,0,1},{y,0,Sqrt[1-x]}]
```

$$\text{Out[30]}= \frac{4}{105}$$

(a)

Exakter Wert eines Integrals :
■ Valeur exacte d'une intégrale:

```
In[31]:= Integrate[y Exp[x^2],{x,0,1},{y,0,Sqrt[x]}]
```

$$\text{Out[31]}= \frac{1}{4}\,(-1+e)$$

Aufgabe 8 ▮ Problème 8

(a) Potenzreihenentwicklung
▮ Développer des séries de puissances

In[32]:= **Series[E^x ,{x, 0, 12}]**

Out[32]= $1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \frac{x^8}{40320} + \frac{x^9}{362880} + \frac{x^{10}}{3628800} + \frac{x^{11}}{39916800} + \frac{x^{12}}{479001600} + O[x]^{13}$

(b) Potenzreihenentwicklung
▮ Développer des séries de puissances

In[33]:= **Series[1/(1 - x),{x, 0, 8}]**

Out[33]= $1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + O[x]^9$

(c) Potenzreihenentwicklung
▮ Développer des séries de puissances

In[34]:= **Series[1/(1 - x)^2, {x, 0, 10}]**

Out[34]= $1 + 2x + 3x^2 + 4x^3 + 5x^4 + 6x^5 + 7x^6 + 8x^7 + 9x^8 + 10x^9 + 11x^{10} + O[x]^{11}$

(d) Potenzreihenentwicklung
▮ Développer des séries de puissances

In[35]:= **Series[f[x], {x, 0, 10}]**

Out[35]= $f[0] + f'[0]\,x + \frac{1}{2} f''[0]\,x^2 + \frac{1}{6} f^{(3)}[0]\,x^3 + \frac{1}{24} f^{(4)}[0]\,x^4 + \frac{1}{120} f^{(5)}[0]\,x^5 + \frac{1}{720} f^{(6)}[0]\,x^6 + \frac{f^{(7)}[0]\,x^7}{5040} + \frac{f^{(8)}[0]\,x^8}{40320} + \frac{f^{(9)}[0]\,x^9}{362880} + \frac{f^{(10)}[0]\,x^{10}}{3628800} + O[x]^{11}$

Aufgabe 9 ▮ Problème 9

(a) Interessante Potenzreihenentwicklung
▮ Développement intéressant de séries de puissances

In[36]:= **Series[1/(1 + x) ,{x, 0, 7}]**

Out[36]= $1 - x + x^2 - x^3 + x^4 - x^5 + x^6 - x^7 + O[x]^8$

(b) Interessante Potenzreihenentwicklung
▮ Développement intéressant de séries de puissances

```
In[37]:= Series[1/(1 - 2x),{x, 0, 8}]
```

$$\text{Out[37]= } 1 + 2\,x + 4\,x^2 + 8\,x^3 + 16\,x^4 + 32\,x^5 + 64\,x^6 + 128\,x^7 + 256\,x^8 + O[x]^9$$

(c) Interessante Potenzreihenentwicklung
▮ Développement intéressant de séries de puissances

```
In[38]:= (1+ 2x + 3x^2) Series[1/(1 - x^3), {x, 0, 8}]
```

$$\text{Out[38]= } 1 + 2\,x + 3\,x^2 + x^3 + 2\,x^4 + 3\,x^5 + x^6 + 2\,x^7 + 3\,x^8 + O[x]^9$$

Aufgabe 10 ▮ Problème 10

(a)

Probleme mit Integration:

▮ Problème d'intégration

```
In[39]:= Integrate[1/(1 + Sin[x]^2 Sqrt[Pi^2 + x]),{x,-2,2}]
```

$$\text{Out[39]= } \int_{-2}^{2} \frac{1}{1 + \sqrt{\pi^2 + x}\,\text{Sin}[x]^2}\,\mathrm{d}x$$

(b)

Näherung durch Integration einer Potenzreihe:

▮ Approximation par l'intégration d'une série de puissances:

```
In[40]:= Clear[s];
         s = Series[1/(1 + Sin[x]^2 Sqrt[Pi^2 + x]),{x,0,8}]
```

$$\text{Out[41]}= 1 - \pi\, x^2 - \frac{x^3}{2\,\pi} + \left(\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2\right) x^4 + \left(1 - \frac{1}{16\,\pi^5} + \frac{1}{6\,\pi}\right) x^5 + \left(\frac{5}{128\,\pi^7} - \frac{1}{24\,\pi^3} + \frac{1}{4\,\pi^2} - \frac{2\,\pi}{45} + \left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3}\right)\pi - \pi\left(\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2\right)\right) x^6 + \left(-\frac{7}{256\,\pi^9} + \frac{1}{48\,\pi^5} - \frac{1}{45\,\pi} + \frac{-\frac{1}{8\,\pi^3} - \frac{\pi}{3}}{2\,\pi} + \left(\frac{1}{16\,\pi^5} - \frac{1}{6\,\pi}\right)\pi - \left(1 - \frac{1}{16\,\pi^5} + \frac{1}{6\,\pi}\right)\pi - \frac{\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2}{2\,\pi}\right) x^7 + \left(\frac{21}{1024\,\pi^{11}} - \frac{5}{384\,\pi^7} + \frac{1}{180\,\pi^3} + \frac{-1 + \frac{1}{16\,\pi^5} - \frac{1}{6\,\pi}}{2\,\pi} + \frac{\frac{1}{16\,\pi^5} - \frac{1}{6\,\pi}}{2\,\pi} + \frac{\pi}{315} + \left(-\frac{5}{128\,\pi^7} + \frac{1}{24\,\pi^3} + \frac{2\,\pi}{45}\right)\pi + \left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3}\right)\left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3} - \pi^2\right) + \pi\left(-\frac{5}{128\,\pi^7} + \frac{1}{24\,\pi^3} - \frac{1}{4\,\pi^2} + \frac{2\,\pi}{45} - \left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3}\right)\pi + \pi\left(\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2\right)\right)\right) x^8 + O[x]^9$$

Verwandlung in einen verwertbaren Ausdruck (O[...] weglassen)

■ Transformation en une expression utilisable (omettre O[...])

```
In[42]:= sn = Normal[s]
```

$$\text{Out[42]}= 1 - \pi\, x^2 - \frac{x^3}{2\,\pi} + \left(\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2\right) x^4 + \left(1 - \frac{1}{16\,\pi^5} + \frac{1}{6\,\pi}\right) x^5 + \left(\frac{5}{128\,\pi^7} - \frac{1}{24\,\pi^3} + \frac{1}{4\,\pi^2} - \frac{2\,\pi}{45} + \left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3}\right)\pi - \pi\left(\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2\right)\right) x^6 + \left(-\frac{7}{256\,\pi^9} + \frac{1}{48\,\pi^5} - \frac{1}{45\,\pi} + \frac{-\frac{1}{8\,\pi^3} - \frac{\pi}{3}}{2\,\pi} + \left(\frac{1}{16\,\pi^5} - \frac{1}{6\,\pi}\right)\pi - \left(1 - \frac{1}{16\,\pi^5} + \frac{1}{6\,\pi}\right)\pi - \frac{\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2}{2\,\pi}\right) x^7 + \left(\frac{21}{1024\,\pi^{11}} - \frac{5}{384\,\pi^7} + \frac{1}{180\,\pi^3} + \frac{-1 + \frac{1}{16\,\pi^5} - \frac{1}{6\,\pi}}{2\,\pi} + \frac{\frac{1}{16\,\pi^5} - \frac{1}{6\,\pi}}{2\,\pi} + \frac{\pi}{315} + \left(-\frac{5}{128\,\pi^7} + \frac{1}{24\,\pi^3} + \frac{2\,\pi}{45}\right)\pi + \left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3}\right)\left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3} - \pi^2\right) + \pi\left(-\frac{5}{128\,\pi^7} + \frac{1}{24\,\pi^3} - \frac{1}{4\,\pi^2} + \frac{2\,\pi}{45} - \left(-\frac{1}{8\,\pi^3} - \frac{\pi}{3}\right)\pi + \pi\left(\frac{1}{8\,\pi^3} + \frac{\pi}{3} + \pi^2\right)\right)\right) x^8$$

Jetzt kann man integrieren:

■ Maintenant on peut intégrer:

```
In[43]:= Integrate[sn,{x,-2,2}]
```

$$\text{Out[43]}= 4 + \frac{7}{3\,\pi^{11}} - \frac{10}{189\,\pi^7} + \frac{2008}{2835\,\pi^3} - \frac{128}{3\,\pi} - \frac{944\,\pi}{405} + \frac{704\,\pi^2}{63} + \frac{4864\,\pi^3}{63} + \frac{1024\,\pi^4}{9}$$

Numerisch: ■ Numériquement:

```
In[44]:= N[%]

Out[44]= 13570.3
```

Aufgabe 11 ▮ Problème 11

(a) Differentialgleichung ▮ Equation différentielle
y'(x) - y(x) tan x = x

```
In[45]:= DSolve[y'[x] - y[x] Tan[x]== x, y[x], x]

Out[45]= {{y[x] → C[1] Sec[x] + Sec[x] (Cos[x] + x Sin[x])}}
```

(b) Differentialgleichung ▮ Equation différentielle
y'(x) + y(x) tan x = sec(x) (secans)

```
In[46]:= DSolve[y'[x] + y[x] Tan[x]== Sec[x], y[x], x]

Out[46]= {{y[x] → C[1] Cos[x] + Sin[x]}}
```

(c) Differentialgleichung ▮ Equation différentielle
y'(x) = y(x)
mit Randwertbedingung
▮ avec condition de valeur au bord
y'(0)=1

```
In[47]:= Clear[y];
         DSolve[{y'[x] == y[x] , y'[0] == 1}, y[x], x]

Out[48]= {{y[x] → ℯ^x}}
```

(d) Differentialgleichung ▮ Equation différentielle
y''''(x) - k^4 y(x) = 0
(Vibration eines Strahls ▮ Vibration d'un rayon)

```
In[49]:= DSolve[y''''[x] - k^4 y[x] == 0, y[x], x]

Out[49]= {{y[x] → ℯ^(-k x) C[2] + ℯ^(k x) C[4] + C[1] Cos[k x] + C[3] Sin[k x]}}
```

(b) Differentialgleichung ▮ Equation différentielle
y'(x) + y(x) tan x = sec(x) (secans)

```
In[50]:= DSolve[y'[x] y[x] + x^2 == x, y[x], x]

Out[50]= {{y[x] → -Sqrt[3 x^2 - 2 x^3 + 6 C[1]]/Sqrt[3]}, {y[x] → Sqrt[3 x^2 - 2 x^3 + 6 C[1]]/Sqrt[3]}}
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[51]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[52]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

4. Graphiken ▮ Graphiques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 4 lesen!
Run mit WIN+*Mathematica* Version 5.2

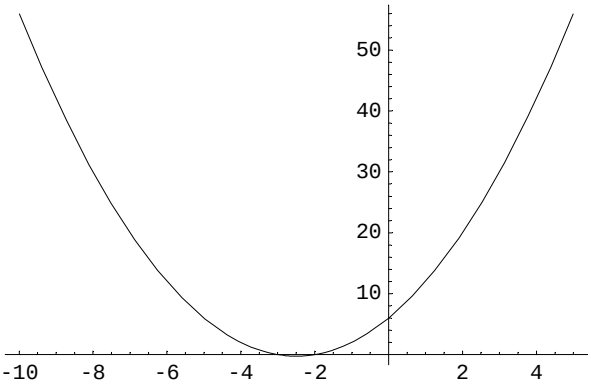
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 4.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

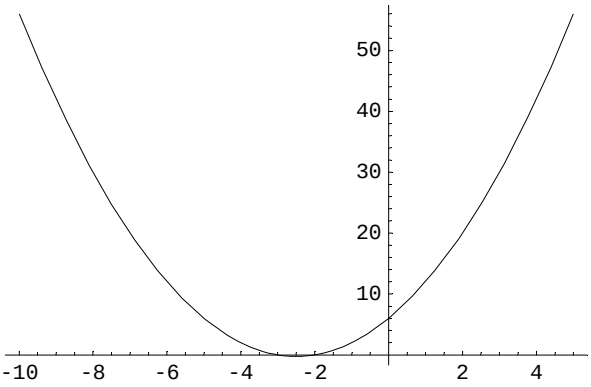
4.1. Zweidimensionale Plots
▮ Plots à deux dimensions

Gewöhnliche Funktion über einem Intervall. ";" unterdrückt den letzten Output:
▮ Fonction commune sur un intervalle. ";" empêche le dernier output.

```
In[1]:= Plot[x^2 + 5x + 6,{x, -10, 5}];
```

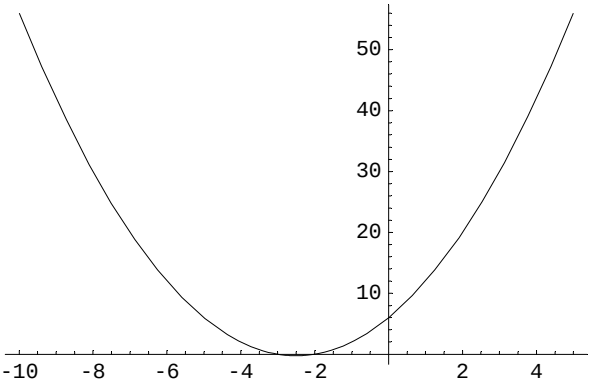


In[2]:= **Plot[x^2 + 5x + 6,{x, -10, 5}]**



Out[2]= **-** Graphics **-**

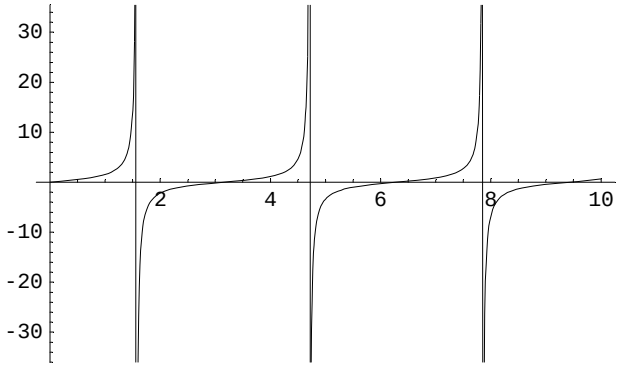
In[3]:= **y = x^2 + 5x + 6;**
Plot[y,{x, -10, 5}];



Pole bieten keine Probleme:

■ Les pôles ne causent pas de problèmes:

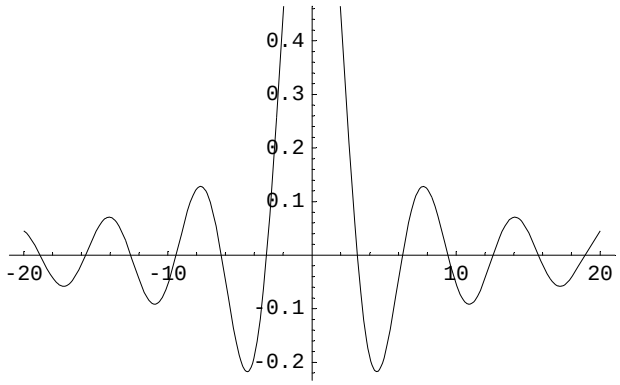
In[5]:= **Plot[Tan[x],{x, 0, 10}];**



Vielleicht wird etwas abgeschnitten:

■ *Mathematica* coupe peut-être quelque chose:

In[6]:=
```
Plot[Sin[x]/x,{x, -20, 20}];
```



4.2. Options
Options

4.2.1. Allgemeines
En général

Ein Plot kann auf diverse Weise durch Angabe der Options verändert werden:

Un plot peut être transformé de différentes façons en indiquant les options:

In[7]:=
```
??Plot
```

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
   xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. Mehr...

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic,
  Compiled → True, DefaultColor → Automatic, DefaultFont :↦ $DefaultFont,
  DisplayFunction :↦ $DisplayFunction, Epilog → {}, FormatType :↦ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
  ImageSize → Automatic, MaxBend → 10., PlotDivision → 30., PlotLabel → None,
  PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :↦ $TextStyle, Ticks → Automatic}
```

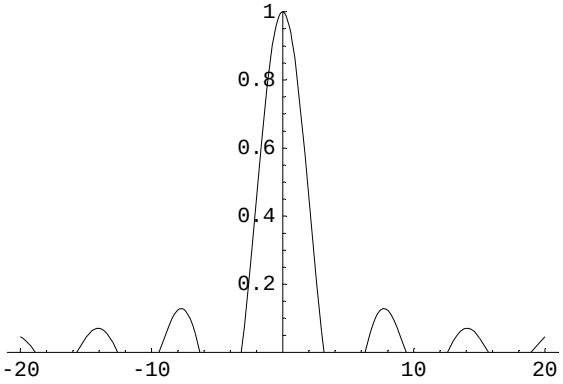
In[8]:=
```
Options[Plot]
```

Out[8]=
$$\left\{\text{AspectRatio} \to \frac{1}{\text{GoldenRatio}}, \text{Axes} \to \text{Automatic}, \text{AxesLabel} \to \text{None},\right.$$
```
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
  ColorOutput → Automatic, Compiled → True, DefaultColor → Automatic,
  DefaultFont :↦ $DefaultFont, DisplayFunction :↦ $DisplayFunction, Epilog → {},
  FormatType :↦ $FormatType, Frame → False, FrameLabel → None, FrameStyle → Automatic,
  FrameTicks → Automatic, GridLines → None, ImageSize → Automatic,
  MaxBend → 10., PlotDivision → 30., PlotLabel → None, PlotPoints → 25,
  PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :↦ $TextStyle, Ticks → Automatic}
```

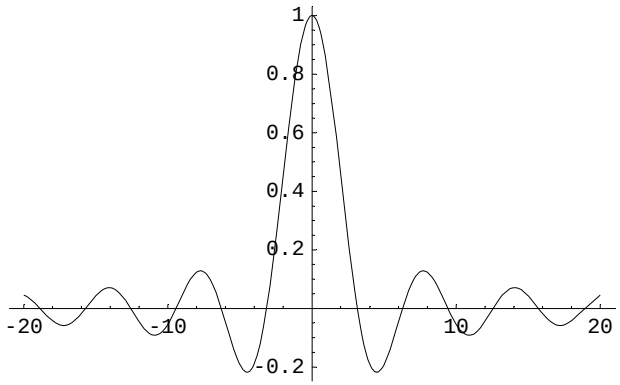
Beispiel mit dem y-Intervall:

Exemple avec l'intervalle y:

In[9]:= **Plot[Sin[x]/x,{x, -20, 20}, PlotRange -> {0, 1}];**



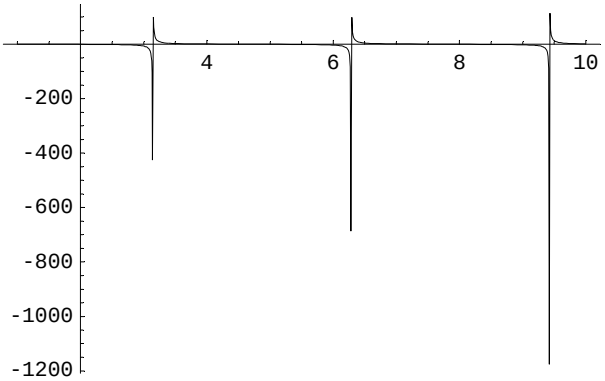
In[10]:= **Plot[Sin[x]/x,{x, -20, 20}, PlotRange -> All];**



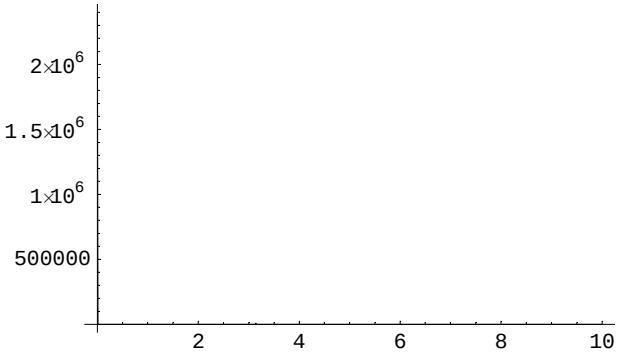
Vielleicht wird trotzdem "etwas" abgeschnitten...:

Peut-être *Mathematica* coupe quand-même "quelque chose"...:

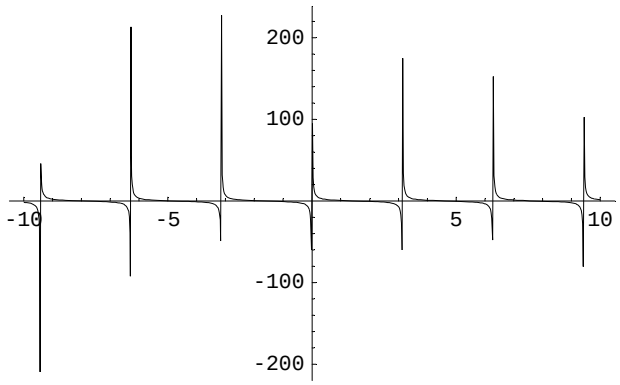
In[11]:= **Plot[Cot[x],{x, 1, 10}, PlotRange -> All];**



In[12]:= **Plot[Cot[x],{x, 0, 10}, PlotRange -> All];**



In[13]:= **Plot[Cot[x],{x, -10, 10}, PlotRange -> All];**



4.2.2. Die Voreinstellung von Mathematica kann abgefragt werden:
▮ La mise au point préalable de *Mathematica* peut être interrogée:

In[14]:= **Options[Plot]**

Out[14]=
$$\left\{\text{AspectRatio} \to \frac{1}{\text{GoldenRatio}}, \text{Axes} \to \text{Automatic}, \text{AxesLabel} \to \text{None},\right.$$
$$\text{AxesOrigin} \to \text{Automatic}, \text{AxesStyle} \to \text{Automatic}, \text{Background} \to \text{Automatic},$$
$$\text{ColorOutput} \to \text{Automatic}, \text{Compiled} \to \text{True}, \text{DefaultColor} \to \text{Automatic},$$
$$\text{DefaultFont} :\to \$\text{DefaultFont}, \text{DisplayFunction} :\to \$\text{DisplayFunction}, \text{Epilog} \to \{\},$$
$$\text{FormatType} :\to \$\text{FormatType}, \text{Frame} \to \text{False}, \text{FrameLabel} \to \text{None}, \text{FrameStyle} \to \text{Automatic},$$
$$\text{FrameTicks} \to \text{Automatic}, \text{GridLines} \to \text{None}, \text{ImageSize} \to \text{Automatic},$$
$$\text{MaxBend} \to 10., \text{PlotDivision} \to 30., \text{PlotLabel} \to \text{None}, \text{PlotPoints} \to 25,$$
$$\text{PlotRange} \to \text{Automatic}, \text{PlotRegion} \to \text{Automatic}, \text{PlotStyle} \to \text{Automatic},$$
$$\left.\text{Prolog} \to \{\}, \text{RotateLabel} \to \text{True}, \text{TextStyle} :\to \$\text{TextStyle}, \text{Ticks} \to \text{Automatic}\right\}$$

In[15]:= **AspectRatio /. Options[Plot]**

Out[15]=
$$\frac{1}{\text{GoldenRatio}}$$

In[16]:= **??AspectRatio**

AspectRatio is an option for Graphics and related
functions which specifies the ratio of height to width for a plot. Mehr...

Attributes[AspectRatio] = {Protected}

```
In[17]:= PlotPoints /. Options[Plot]

Out[17]= 25
```

Der Output kann unterdrückt werden:

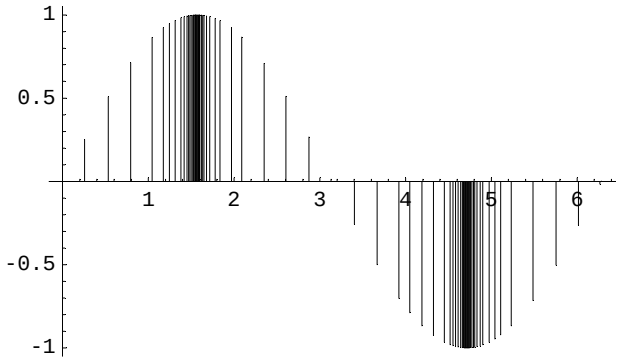
■ On peut réprimer l'output.

```
In[18]:= test = Plot[Sin[x],{x, 0, 2 Pi},
          DisplayFunction -> Identity];
```

Hier eine Zeichnung, die zeigt, in welchen Abständen *Mathematica* die Funktionswerte berechnet:

■ Voici un dessin qui montre à quelles distances *Mathematica* calcule les valeurs des fonctions:

```
In[19]:= Show[Graphics[{Thickness[0.001], Map[Line[{{#[[1]],
             0},#}]&, Nest[First, test, 4]]}],
             Axes->Automatic];
```



Was tun die einzelnen Befehle?

■ Que fait chaque ordre?

```
In[20]:= test

Out[20]= - Graphics -

In[21]:= ??Nest

         Nest[f, expr, n] gives an expression with f applied n times to expr. Mehr…

         Attributes[Nest] = {Protected}
```

In[22]:= **Nest[First, test, 4]**

Out[22]=
```
{{2.61799×10^-7, 2.61799×10^-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
 {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355}, {1.24595, 0.947701},
 {1.31226, 0.966765}, {1.37966, 0.981789}, {1.41517, 0.987915}, {1.45282, 0.993049},
 {1.46855, 0.994777}, {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
 {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
 {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906}, {1.5614, 0.999956},
 {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889}, {1.59373, 0.999737},
 {1.59828, 0.999622}, {1.60249, 0.999498}, {1.6184, 0.998867}, {1.63286, 0.998075},
 {1.64822, 0.997004}, {1.68008, 0.994035}, {1.71525, 0.989585},
 {1.77763, 0.978685}, {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
 {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
 {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
 {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
 {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
 {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
 {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
 {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
 {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
 {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
 {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
 {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
 {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
 {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
 {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
 {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10^-7}}
```

In[23]:= **Nest[First, test, 5]**

Out[23]=
```
{2.61799×10^-7, 2.61799×10^-7}
```

In[24]:= **Nest[First, test, 3]**

Out[24]=
```
Line[{{2.61799×10^-7, 2.61799×10^-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
  {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355}, {1.24595, 0.947701},
  {1.31226, 0.966765}, {1.37966, 0.981789}, {1.41517, 0.987915}, {1.45282, 0.993049},
  {1.46855, 0.994777}, {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
  {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
  {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906}, {1.5614, 0.999956},
  {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889}, {1.59373, 0.999737},
  {1.59828, 0.999622}, {1.60249, 0.999498}, {1.6184, 0.998867}, {1.63286, 0.998075},
  {1.64822, 0.997004}, {1.68008, 0.994035}, {1.71525, 0.989585},
  {1.77763, 0.978685}, {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
  {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
  {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
  {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
  {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
  {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
  {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
  {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
  {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
  {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
  {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
  {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
  {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
  {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
  {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
  {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10^-7}}]
```

In[25]:= **Nest[First, test, 2]**

Out[25]=
```
{Line[{{2.61799×10^-7, 2.61799×10^-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
    {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355},
    {1.24595, 0.947701}, {1.31226, 0.966765}, {1.37966, 0.981789},
    {1.41517, 0.987915}, {1.45282, 0.993049}, {1.46855, 0.994777},
    {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
    {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
    {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906}, {1.5614, 0.999956},
    {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889}, {1.59373, 0.999737},
    {1.59828, 0.999622}, {1.60249, 0.999498}, {1.6184, 0.998867}, {1.63286, 0.998075},
    {1.64822, 0.997004}, {1.68008, 0.994035}, {1.71525, 0.989585},
    {1.77763, 0.978685}, {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
    {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
    {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
    {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
    {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
    {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
    {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
    {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
    {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
    {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
    {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
    {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
    {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
    {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
    {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
    {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10^-7}}]}
```

In[26]:= **Nest[First, test, 1]**

Out[26]=

{{Line[{{2.61799×10^-7, 2.61799×10^-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
{0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355},
{1.24595, 0.947701}, {1.31226, 0.966765}, {1.37966, 0.981789},
{1.41517, 0.987915}, {1.45282, 0.993049}, {1.46855, 0.994777},
{1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
{1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
{1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906},
{1.5614, 0.999956}, {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889},
{1.59373, 0.999737}, {1.59828, 0.999622}, {1.60249, 0.999498},
{1.6184, 0.998867}, {1.63286, 0.998075}, {1.64822, 0.997004},
{1.68008, 0.994035}, {1.71525, 0.989585}, {1.77763, 0.978685},
{1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
{2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
{3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
{3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
{4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
{4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
{4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
{4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
{4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
{4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
{4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
{4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
{4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
{4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
{5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
{6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10^-7}}]}}

In[27]:= **First[test]**

Out[27]=
```
{{Line[{{2.61799×10^-7, 2.61799×10^-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
    {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355},
    {1.24595, 0.947701}, {1.31226, 0.966765}, {1.37966, 0.981789},
    {1.41517, 0.987915}, {1.45282, 0.993049}, {1.46855, 0.994777},
    {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
    {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
    {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906},
    {1.5614, 0.999956}, {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889},
    {1.59373, 0.999737}, {1.59828, 0.999622}, {1.60249, 0.999498},
    {1.6184, 0.998867}, {1.63286, 0.998075}, {1.64822, 0.997004},
    {1.68008, 0.994035}, {1.71525, 0.989585}, {1.77763, 0.978685},
    {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
    {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
    {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
    {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
    {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
    {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
    {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
    {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
    {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
    {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
    {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
    {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
    {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
    {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
    {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
    {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10^-7}}]}}
```

In[28]:= **Map[Line[{{#[[1]],0},#}]&, Nest[First, test, 4]]**

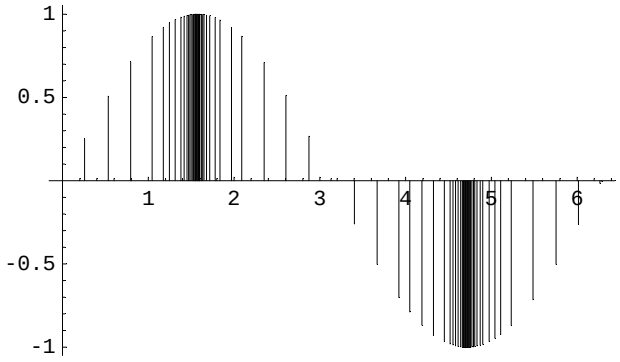
Out[28]=
```
{Line[{{2.61799×10^-7, 0}, {2.61799×10^-7, 2.61799×10^-7}}],
 Line[{{0.25489, 0}, {0.25489, 0.252139}}],
 Line[{{0.532869, 0}, {0.532869, 0.508007}}],
 Line[{{0.793939, 0}, {0.793939, 0.71312}}],
 Line[{{1.04501, 0}, {1.04501, 0.864929}}],
 Line[{{1.17413, 0}, {1.17413, 0.922355}}],
 Line[{{1.24595, 0}, {1.24595, 0.947701}}],
 Line[{{1.31226, 0}, {1.31226, 0.966765}}],
 Line[{{1.37966, 0}, {1.37966, 0.981789}}],
 Line[{{1.41517, 0}, {1.41517, 0.987915}}],
 Line[{{1.45282, 0}, {1.45282, 0.993049}}],
 Line[{{1.46855, 0}, {1.46855, 0.994777}}],
 Line[{{1.48546, 0}, {1.48546, 0.996361}}],
 Line[{{1.50009, 0}, {1.50009, 0.997501}}],
 Line[{{1.51605, 0}, {1.51605, 0.998502}}],
 Line[{{1.52572, 0}, {1.52572, 0.998984}}],
 Line[{{1.53469, 0}, {1.53469, 0.999348}}],
 Line[{{1.54327, 0}, {1.54327, 0.999621}}],
 Line[{{1.54766, 0}, {1.54766, 0.999732}}],
 Line[{{1.55242, 0}, {1.55242, 0.999831}}],
 Line[{{1.5571, 0}, {1.5571, 0.999906}}], Line[{{1.5614, 0}, {1.5614, 0.999956}}],
 Line[{{1.56987, 0}, {1.56987, 1.}}], Line[{{1.57756, 0}, {1.57756, 0.999977}}],
 Line[{{1.58569, 0}, {1.58569, 0.999889}}],
 Line[{{1.59373, 0}, {1.59373, 0.999737}}],
 Line[{{1.59828, 0}, {1.59828, 0.999622}}],
 Line[{{1.60249, 0}, {1.60249, 0.999498}}],
```

```
          Line[{{1.6184, 0}, {1.6184, 0.998867}}], Line[{{1.63286, 0}, {1.63286, 0.998075}}],
          Line[{{1.64822, 0}, {1.64822, 0.997004}}],
          Line[{{1.68008, 0}, {1.68008, 0.994035}}],
          Line[{{1.71525, 0}, {1.71525, 0.989585}}],
          Line[{{1.77763, 0}, {1.77763, 0.978685}}],
          Line[{{1.8362, 0}, {1.8362, 0.964986}}], Line[{{1.96935, 0}, {1.96935, 0.921622}}],
          Line[{{2.09144, 0}, {2.09144, 0.867501}}],
          Line[{{2.35285, 0}, {2.35285, 0.709467}}],
          Line[{{2.60427, 0}, {2.60427, 0.511842}}],
          Line[{{2.87186, 0}, {2.87186, 0.266474}}],
          Line[{{3.12945, 0}, {3.12945, 0.0121375}}],
          Line[{{3.40323, 0}, {3.40323, -0.258662}}],
          Line[{{3.667, 0}, {3.667, -0.501569}}], Line[{{3.92078, 0}, {3.92078, -0.702701}}],
          Line[{{4.04933, 0}, {4.04933, -0.788113}}],
          Line[{{4.19073, 0}, {4.19073, -0.866996}}],
          Line[{{4.32504, 0}, {4.32504, -0.925913}}],
          Line[{{4.45069, 0}, {4.45069, -0.965952}}],
          Line[{{4.51935, 0}, {4.51935, -0.981426}}],
          Line[{{4.55317, 0}, {4.55317, -0.987352}}],
          Line[{{4.58453, 0}, {4.58453, -0.991837}}],
          Line[{{4.61287, 0}, {4.61287, -0.995052}}],
          Line[{{4.64304, 0}, {4.64304, -0.997597}}],
          Line[{{4.66029, 0}, {4.66029, -0.998643}}],
          Line[{{4.66875, 0}, {4.66875, -0.999048}}],
          Line[{{4.67652, 0}, {4.67652, -0.999357}}],
          Line[{{4.68347, 0}, {4.68347, -0.999582}}],
          Line[{{4.69109, 0}, {4.69109, -0.999773}}],
          Line[{{4.69875, 0}, {4.69875, -0.999907}}],
          Line[{{4.70683, 0}, {4.70683, -0.999985}}],
          Line[{{4.71547, 0}, {4.71547, -0.999995}}],
          Line[{{4.72345, 0}, {4.72345, -0.999939}}],
          Line[{{4.72762, 0}, {4.72762, -0.999884}}],
          Line[{{4.73221, 0}, {4.73221, -0.999803}}],
          Line[{{4.74151, 0}, {4.74151, -0.999576}}],
          Line[{{4.74967, 0}, {4.74967, -0.999305}}],
          Line[{{4.75845, 0}, {4.75845, -0.998939}}],
          Line[{{4.77433, 0}, {4.77433, -0.998082}}],
          Line[{{4.78997, 0}, {4.78997, -0.996992}}],
          Line[{{4.80641, 0}, {4.80641, -0.995583}}],
          Line[{{4.83579, 0}, {4.83579, -0.992395}}],
          Line[{{4.87, 0}, {4.87, -0.987606}}], Line[{{4.90178, 0}, {4.90178, -0.982119}}],
          Line[{{4.97386, 0}, {4.97386, -0.96601}}],
          Line[{{5.04347, 0}, {5.04347, -0.945692}}],
          Line[{{5.10909, 0}, {5.10909, -0.92234}}], Line[{{5.2318, 0}, {5.2318, -0.86811}}],
          Line[{{5.48592, 0}, {5.48592, -0.715447}}],
          Line[{{5.75622, 0}, {5.75622, -0.502911}}],
          Line[{{6.01652, 0}, {6.01652, -0.263515}}],
          Line[{{6.26682, 0}, {6.26682, -0.0163632}}],
          Line[{{6.28319, 0}, {6.28319, -2.61799×10^-7}}]]}
```

In[29]:=
```
Graphics[{Thickness[0.001],
                    Map[Line[{{#[[1]],0},#}]&,
                    Nest[First, test, 4]]}]
```

Out[29]= - Graphics -

```
In[30]:= Show[Graphics[{Thickness[0.001],
             Map[Line[{{#[[1]],0},#}]&,
             Nest[First, test, 4]]}], Axes->Automatic];
```

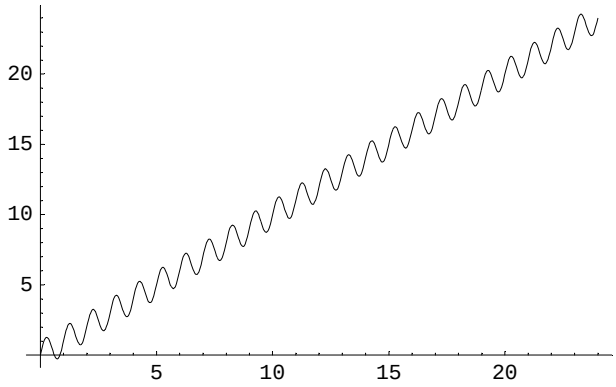


4.2.3. Probleme mit den Plot-Punkten:
▮ Problèmes avec les points-"plot":

Kann das stimmen?

▮ Est-ce correct?

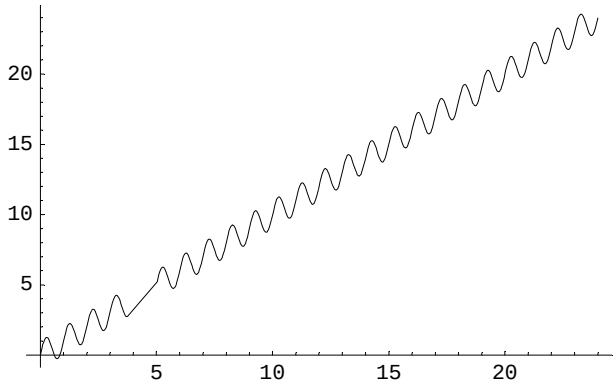
```
In[31]:= Plot[x + Sin[2 Pi x], {x, 0, 24}];
```



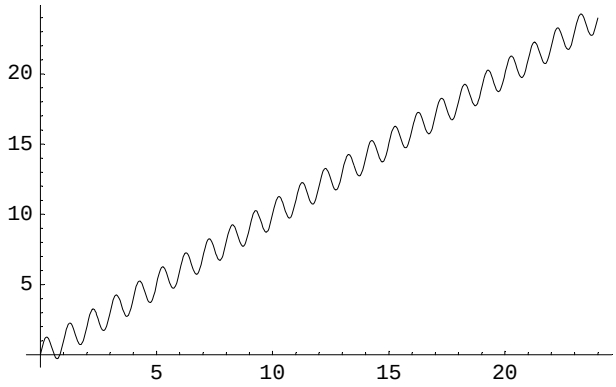
Andere Anzahl Punkte:

▮ Autre nombre ce points:

```
In[32]:= Plot[x + Sin[2 Pi x], {x, 0, 24}, PlotPoints -> 20];
```



In[33]:= **Plot[x + Sin[2 Pi x], {x, 0, 24}, PlotPoints -> 50];**



Welche Punkte rechnet *Mathematica*?

▮ Quels points sont calculés par *Mathematica*?

In[34]:= **Table[{x, x + Sin[2 Pi x]}, {x, 0, 24}]**

Out[34]=
```
{{0, 0}, {1, 1}, {2, 2}, {3, 3}, {4, 4}, {5, 5}, {6, 6}, {7, 7}, {8, 8},
 {9, 9}, {10, 10}, {11, 11}, {12, 12}, {13, 13}, {14, 14}, {15, 15}, {16, 16},
 {17, 17}, {18, 18}, {19, 19}, {20, 20}, {21, 21}, {22, 22}, {23, 23}, {24, 24}}
```

4.2.4. Wechsel des Plot-Stils:
▮ Changement du style "plot":

Liniendicke, Graustufe, Farbe, Linienart, etc. können angepasst werden:

▮ Epaisseur de la ligne, nuance de gris, couleur, type de ligne etc. peuvent être adaptés:

In[35]:= **Options[Plot]**

Out[35]=
```
{AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
 AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
 ColorOutput → Automatic, Compiled → True, DefaultColor → Automatic,
 DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction, Epilog → {},
 FormatType :→ $FormatType, Frame → False, FrameLabel → None, FrameStyle → Automatic,
 FrameTicks → Automatic, GridLines → None, ImageSize → Automatic,
 MaxBend → 10., PlotDivision → 30., PlotLabel → None, PlotPoints → 25,
 PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
 Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[36]:= **??PlotStyle**

```
PlotStyle is an option for Plot and ListPlot
   that specifies the style of lines or points to be plotted. Mehr...

Attributes[PlotStyle] = {Protected}
```

In[37]:= **??Graphics**

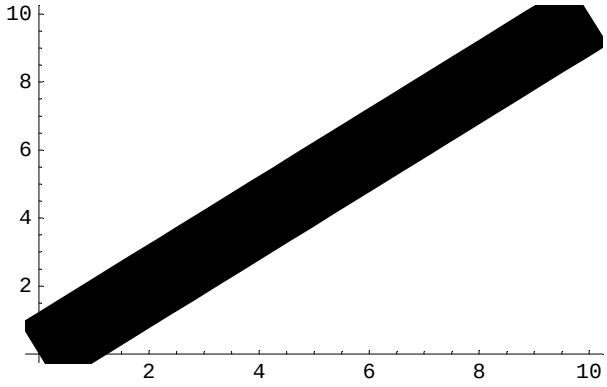
```
Graphics[primitives, options] represents a two-dimensional graphical image. Mehr...

Attributes[Graphics] = {Protected, ReadProtected}

Options[Graphics] = {AspectRatio → 1/GoldenRatio, Axes → False, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
  ColorOutput → Automatic, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
  DisplayFunction :→ $DisplayFunction, Epilog → {}, FormatType :→ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
  ImageSize → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[38]:= **PlotStyle /. Options[Plot]**

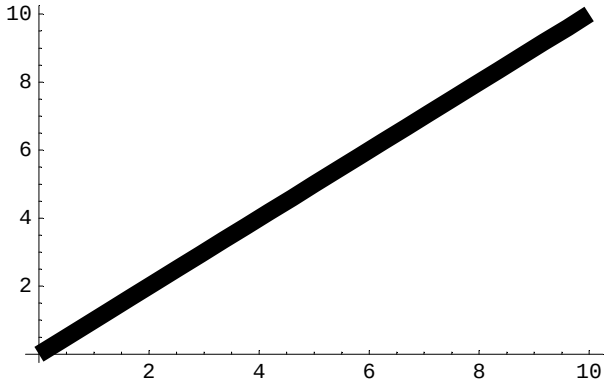
Out[38]= Automatic

Probiere aus: ■ Essaie:

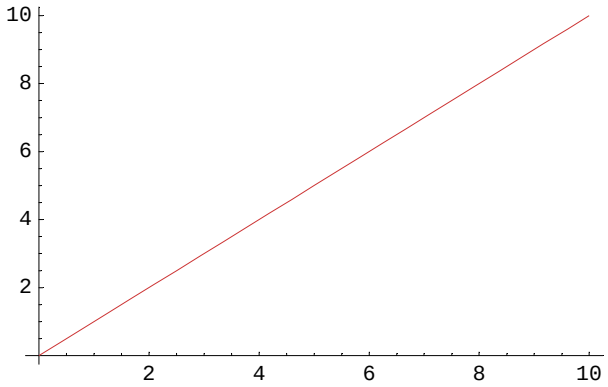
In[39]:= **Plot[x, {x, 0, 10}, PlotStyle -> Thickness[0.125]];**



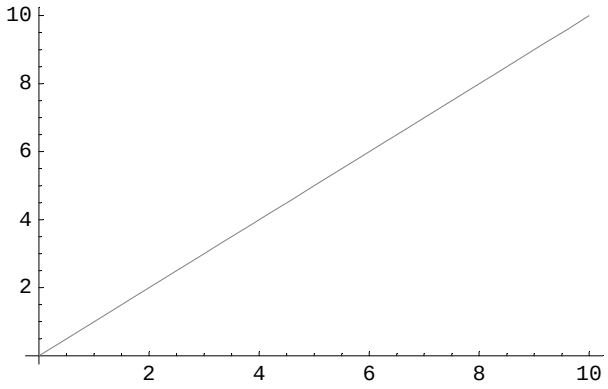
In[40]:= **Plot[x, {x, 0, 10}, PlotStyle -> Thickness[0.03]];**



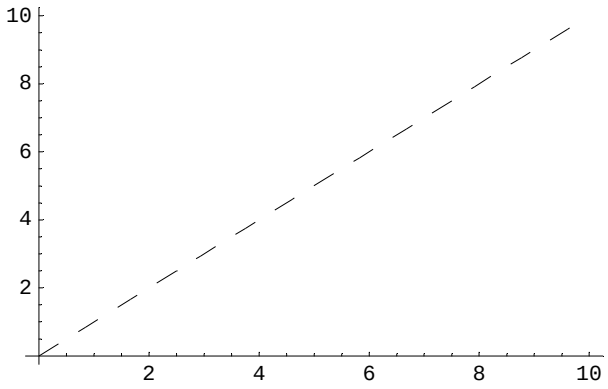
In[41]:=
```
Plot[x, {x, 0, 10}, PlotStyle -> RGBColor[0.8,
                                          0.2, 0.2]];
```



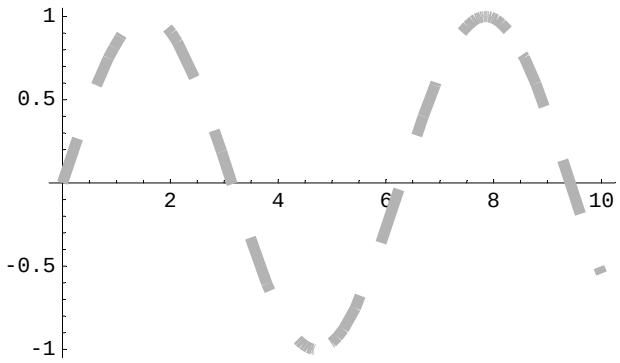
In[42]:=
```
Plot[x, {x, 0, 10}, PlotStyle -> GrayLevel[0.5]];
```



In[43]:=
```
Plot[x, {x, 0, 10}, PlotStyle -> Dashing[{0.04}]];
```



In[44]:=
```
Plot[Sin[x], {x, 0, 10}, PlotStyle -> {{Thickness[
            0.02], GrayLevel[0.7], Dashing[{0.1}]}}];
```

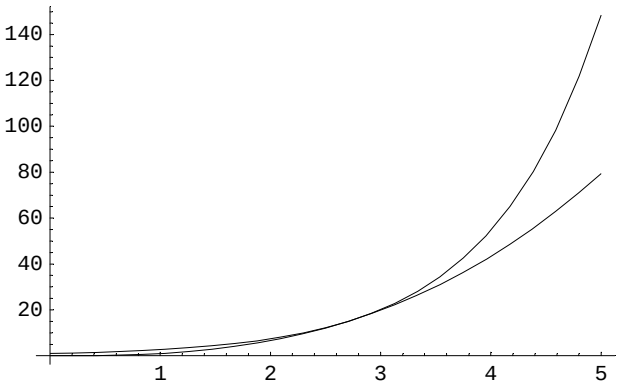


4.3. Mehrere Graphen in einem Bild
▮ Plusieurs graphes en une image

Man kann auch eine Menge von Funktionen auf einmal ausgeben:

▮ On peut aussi sortir un ensemble de fonctions à la fois:

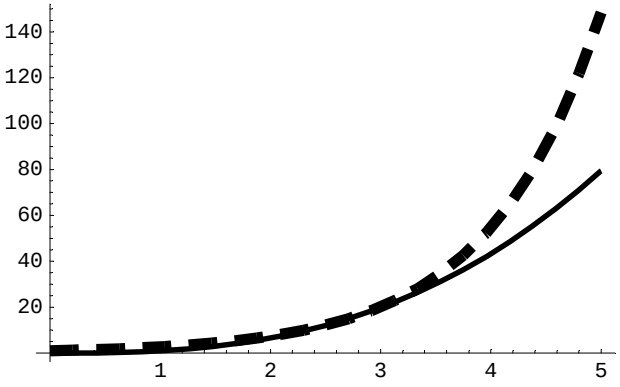
In[45]:=
```
Plot[{E^x, x^E}, {x, 0, 5}];
```



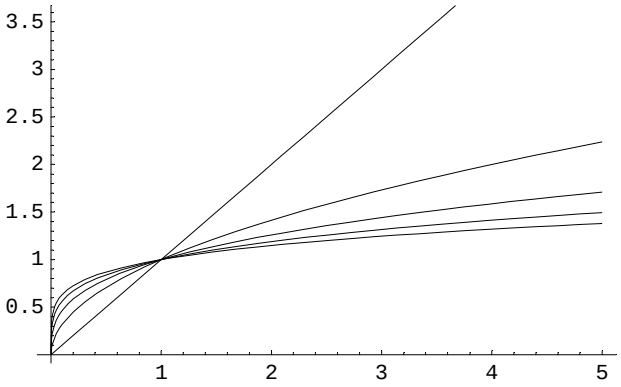
Da es manchmal schwierig ist, die Kurven zu unterscheiden, ist es angebracht, diese verschieden darzustellen:

▮ Comme il est parfois difficile de distinguer les courbes, il est préférable de les représenter de différentes façons:

In[46]:=
```
Plot[{E^x, x^E}, {x, 0, 5}, PlotStyle -> {
            {Thickness[0.02], Dashing[{0.05, 0.03}]},
            {Thickness[0.01]}     }             ];
```



```
In[47]:= Plot[Evaluate[Table[x^(1/n), {n,5}]], {x, 0, 5}];
```




```
In[48]:= Evaluate[Table[x^(1/n), {n,5}]]
```

$$\text{Out[48]= } \left\{x, \sqrt{x}, x^{1/3}, x^{1/4}, x^{1/5}\right\}$$

```
In[49]:= Table[x^(1/n), {n,5}]
```

$$\text{Out[49]= } \left\{x, \sqrt{x}, x^{1/3}, x^{1/4}, x^{1/5}\right\}$$

"Evaluate" steuert die Operationshierarchie. Bevor die Plotpunkte berechnet werden können, muss die Tabelle generiert werden! Versuche Dich nun in eigenen Plots!

■ "Evaluate" dirige la hiérarchie des opérations. Avant de pouvoir calculer les points "plot", il faut générer le tableau! Essaie des plots à toi!

4.4. Parametrisierte Kurven
■ Courbes paramétrisées

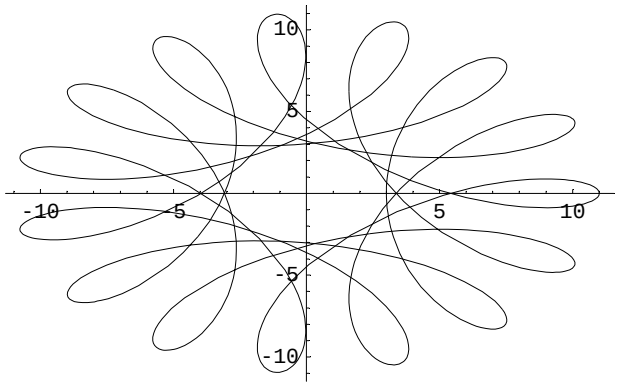
In einem kartesischen Koordinatensystem in der Ebene kann eine Kurve durch einen Vektor gegeben werden, der von einer Variable (Parameter) abhängt. Beispiel:

■ Dans un système de coordonnés cartésien, dans le plan, une courbe peut être donnée par un vecteur qui dépend d'une variable (paramètre). Exemple:

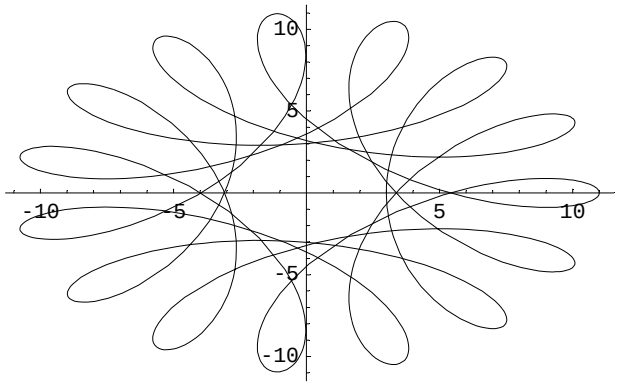
```
In[50]:= v[t_]:= {4 Cos[-11 t/4] + 7 Cos[t], 4 Sin[-11 t/4]
                  + 7 Sin[t]}; v[t]
```

$$\text{Out[50]= } \left\{7\,\text{Cos}[t] + 4\,\text{Cos}\left[\frac{11\,t}{4}\right], 7\,\text{Sin}[t] - 4\,\text{Sin}\left[\frac{11\,t}{4}\right]\right\}$$

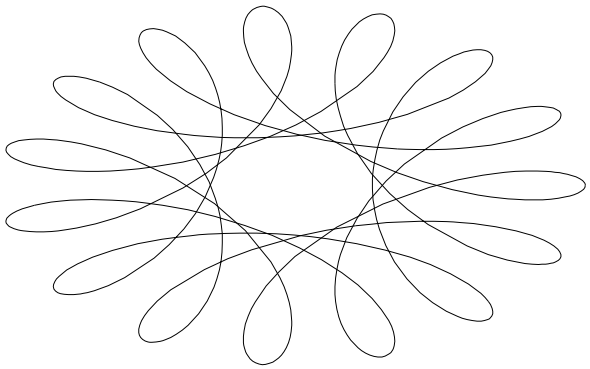
In[51]:=
```
ParametricPlot[v[t], {t, 0, 8 Pi}];
```



In[52]:=
```
ParametricPlot[{4 Cos[-11 t/4] + 7 Cos[t],
                4 Sin[-11 t/4] + 7 Sin[t]},
                {t, 0, 8 Pi}];
```



In[53]:=
```
ParametricPlot[{4 Cos[-11 t/4] + 7 Cos[t],
                4 Sin[-11 t/4] + 7 Sin[t]},
                {t, 0, 8 Pi}, Axes -> None];
```



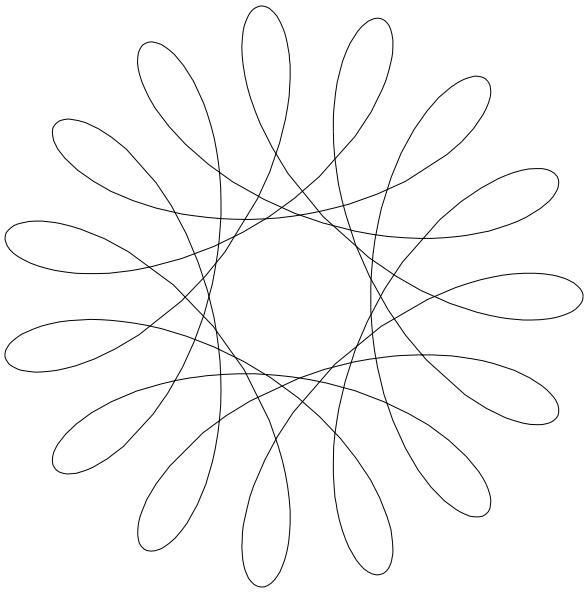
Das Achsenverhältnis:

- Rapport axial:

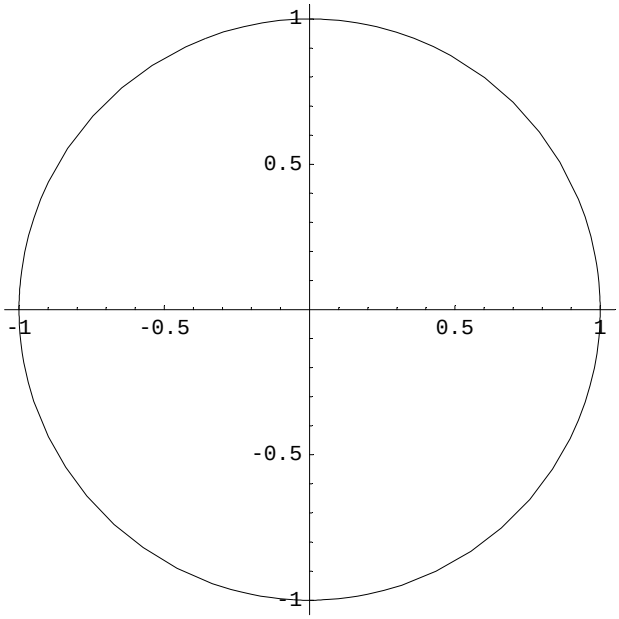
In[54]:=
```
AspectRatio /. Options[Plot]
```

Out[54]=
$$\frac{1}{\text{GoldenRatio}}$$

In[55]:=
```
ParametricPlot[{4 Cos[-11 t/4] + 7 Cos[t],
                4 Sin[-11 t/4] + 7 Sin[t]},
                {t, 0, 8 Pi}, Axes -> None,
                AspectRatio -> Automatic];
```



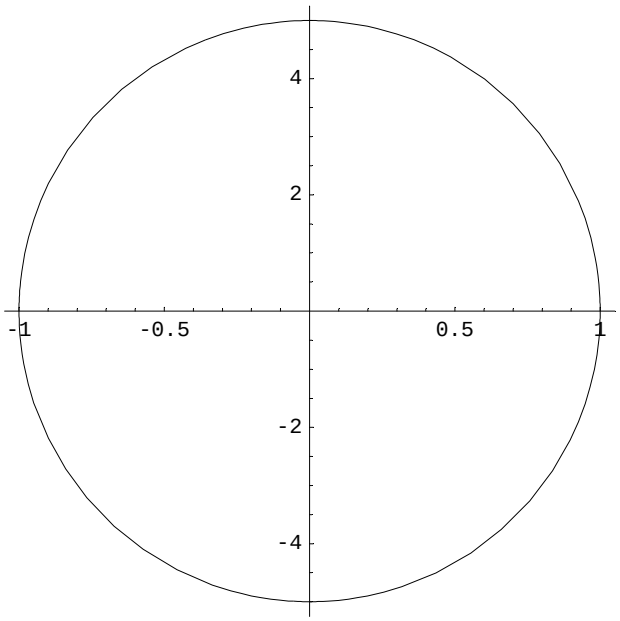
In[56]:=
```
ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2 Pi},
                AspectRatio -> 1];
```



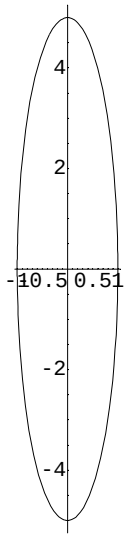
Man beachte den Massstab:

- Tenir compte de l'échelle:

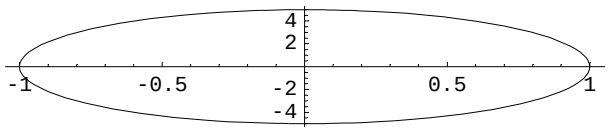
In[57]:=
```
ParametricPlot[{Cos[t], 5 Sin[t]}, {t, 0, 2 Pi},
               AspectRatio -> 1];
```



In[58]:=
```
ParametricPlot[{Cos[t], 5 Sin[t]}, {t, 0, 2 Pi},
               AspectRatio -> 5];
```



In[59]:=
```
ParametricPlot[{Cos[t], 5 Sin[t]}, {t, 0, 2 Pi},
               AspectRatio -> 1/5];
```

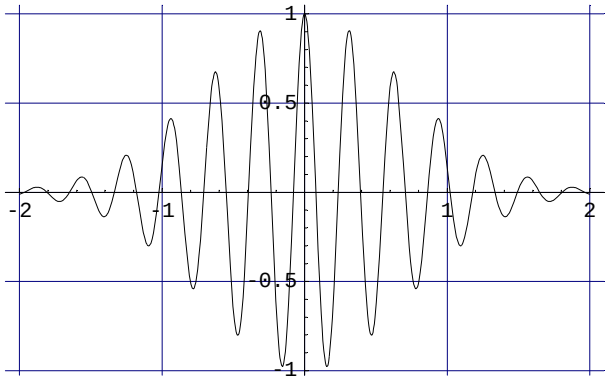


4.5. Weitere Optionen
▮ Autres options

Z.B. ein Gitter:

▮ P. ex. une grille:

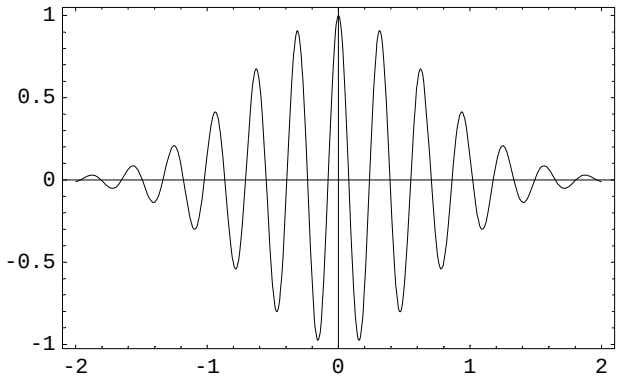
```
In[60]:= Plot[E^-x^2 Cos[20x], {x,-2,2},
                    GridLines -> Automatic];
```



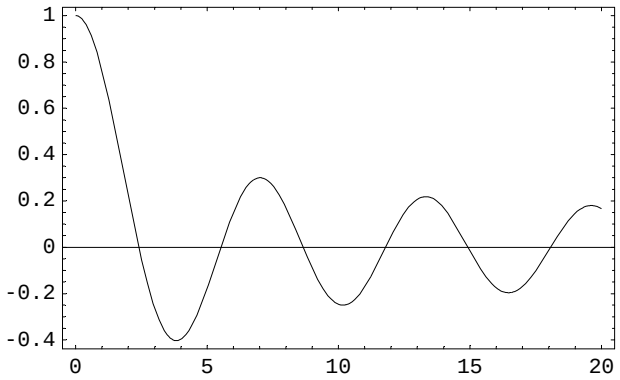
Oder ein Rahmen:

▮ Ou un cadre:

```
In[61]:= Plot[E^-x^2 Cos[20x], {x,-2,2}, Frame -> True];
```




```
In[62]:= Plot[BesselJ[0, x], {x, 0, 20}, Frame -> True];
```

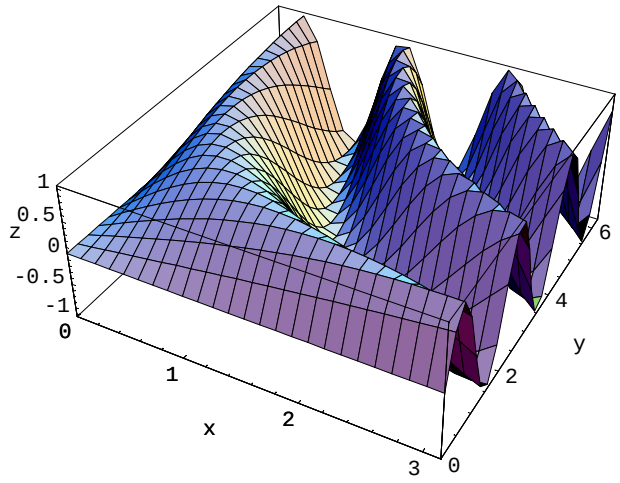


4.6. Plots von Flächen im Raum
▮ Plots de surfaces dans l'espace

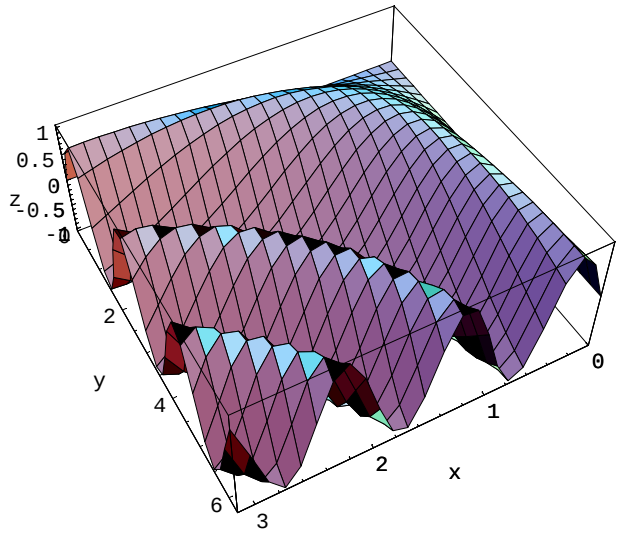
Probiere aus! (Hier werden die Achsen beschriftet):

▮ Essaie! (Ici on met une inscription sur les axes:)

```
In[63]:= Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi},
                    AxesLabel -> {"x", "y", "z"}];
```




```
In[64]:= Show[%, ViewPoint->{1.050, 1.910, 2.440}];
```

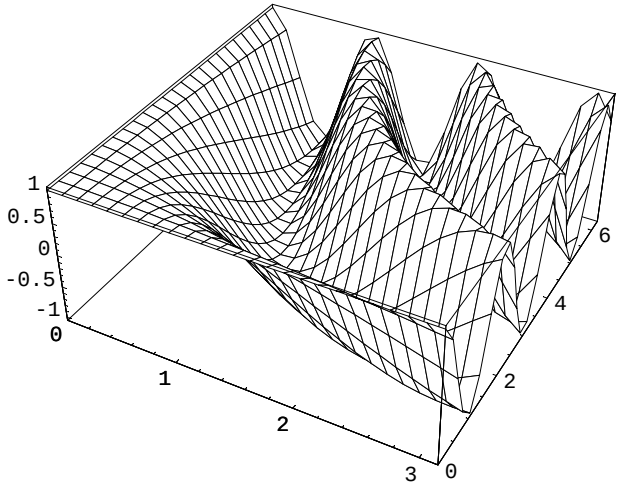


Im Menu unter ("Action", "Prepare Input") neu: "Input", "3D ViewPoint Selector" findest Du eine Hilfe, um den Blickpunkt zu verändern. Du kannst da schieben und dann z.B. die Insert-Taste (Copy-Paste) betätigen, nachdem allerdings der Cursor in einer neuen Zelle parkiert worden ist. Du bekommst dann ein lauffähiges Programm. Probiere die Sache aus!

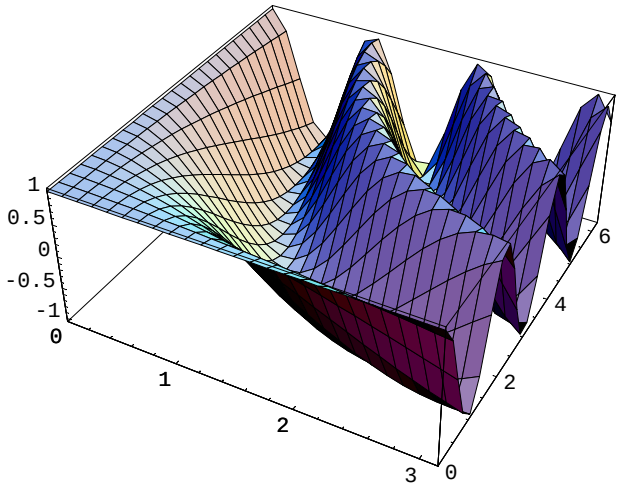
▮ Tu trouves dans le menu sous ("Action", "Prepare Input") nouveau: "Input", "3D ViewPoint Selector" une aide pour changer le point de vue. Là tu peux déplacer et ensuite p.ex. appuyer sur la touche insert, après avoir note bien parqué le curseur dans une nouvelle cellule. Tu obtiendras un programme qui fonctionne. Essaie-le!

In[65]:=
```
Remove["Global`*"]
```

In[66]:=
```
Plot3D[Cos[x y],{x, 0, Pi}, {y, 0, 2 Pi},
Shading → False];
```



In[67]:=
```
Plot3D[Cos[x y],{x, 0, Pi}, {y, 0, 2 Pi},
Shading → True];
```



4.7. Options für Farben, Beleuchtung, ...
▮ Options pour couleurs, éclairage, ...

Probiere aus: ▮ Essaie!

In[68]:= **Options[Plot3D]**

Out[68]=
```
{AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → True,
 AxesEdge → Automatic, AxesLabel → None, AxesStyle → Automatic,
 Background → Automatic, Boxed → True, BoxRatios → {1, 1, 0.4},
 BoxStyle → Automatic, ClipFill → Automatic, ColorFunction → Automatic,
 ColorFunctionScaling → True, ColorOutput → Automatic,
 Compiled → True, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
 DisplayFunction :→ $DisplayFunction, Epilog → {}, FaceGrids → None,
 FormatType :→ $FormatType, HiddenSurface → True, ImageSize → Automatic,
 Lighting → True, LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]},
   {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}},
 Mesh → True, MeshStyle → Automatic, Plot3Matrix → Automatic, PlotLabel → None,
 PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, Prolog → {},
 Shading → True, SphericalRegion → False, TextStyle :→ $TextStyle, Ticks → Automatic,
 ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

In[69]:= **LightSources /. Options[Plot3D]**

Out[69]=
```
{{{1., 0., 1.}, RGBColor[1, 0, 0]},
 {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}}
```

In[70]:= **Lighting /. Options[Plot3D]**

Out[70]= True

Probiere aus: ■ Essaie!

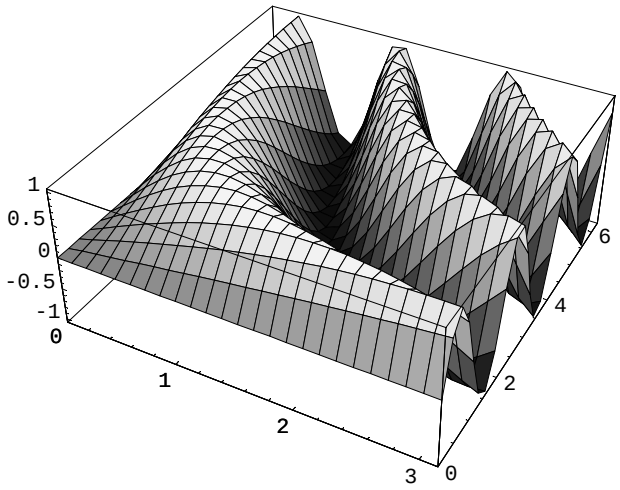
In[71]:= **??SetOptions**

```
SetOptions[s, name1->value1, name2->value2, ... ] sets the specified default
   options for a symbol s. SetOptions[stream, ... ] or SetOptions["name", ... ]
   sets options associated with a particular stream. SetOptions[object, ... ] sets
   options associated with an external object such as a NotebookObject. Mehr...

Attributes[SetOptions] = {Protected}
```

In[72]:= **SetOptions[Plot3D, Lighting -> False]**

Out[72]=
```
{AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → True,
 AxesEdge → Automatic, AxesLabel → None, AxesStyle → Automatic,
 Background → Automatic, Boxed → True, BoxRatios → {1, 1, 0.4},
 BoxStyle → Automatic, ClipFill → Automatic, ColorFunction → Automatic,
 ColorFunctionScaling → True, ColorOutput → Automatic,
 Compiled → True, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
 DisplayFunction :→ $DisplayFunction, Epilog → {}, FaceGrids → None,
 FormatType :→ $FormatType, HiddenSurface → True, ImageSize → Automatic,
 Lighting → False, LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]},
   {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}},
 Mesh → True, MeshStyle → Automatic, Plot3Matrix → Automatic, PlotLabel → None,
 PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, Prolog → {},
 Shading → True, SphericalRegion → False, TextStyle :→ $TextStyle, Ticks → Automatic,
 ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

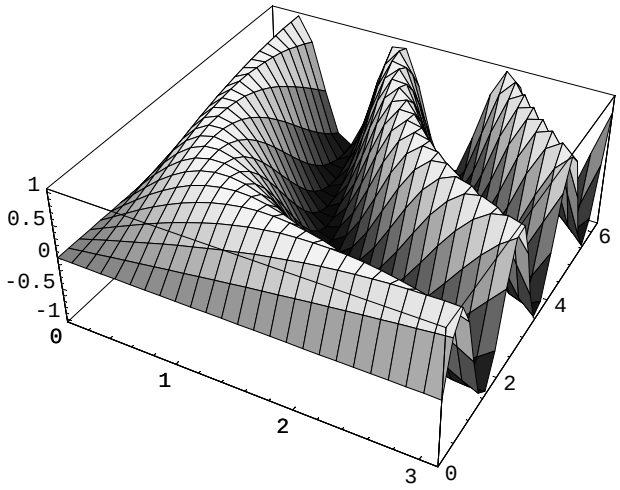
In[73]:=
```
Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi}];
```



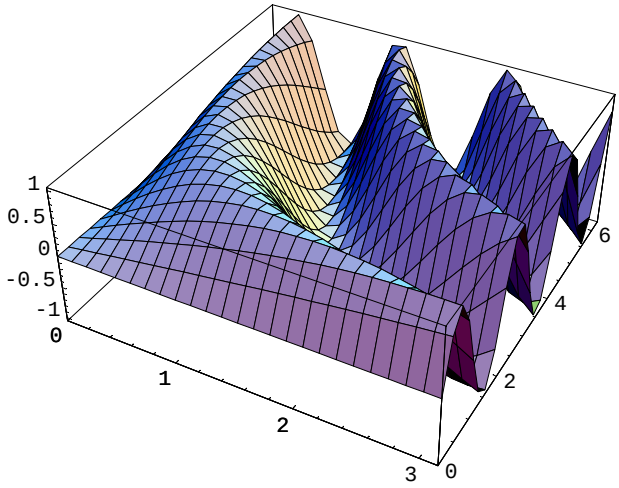
Oder auf einmal:

■ Ou à la fois:

In[74]:=
```
Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi},
Lighting -> False];
```



In[75]:=
```
Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi},
Lighting -> True];
```



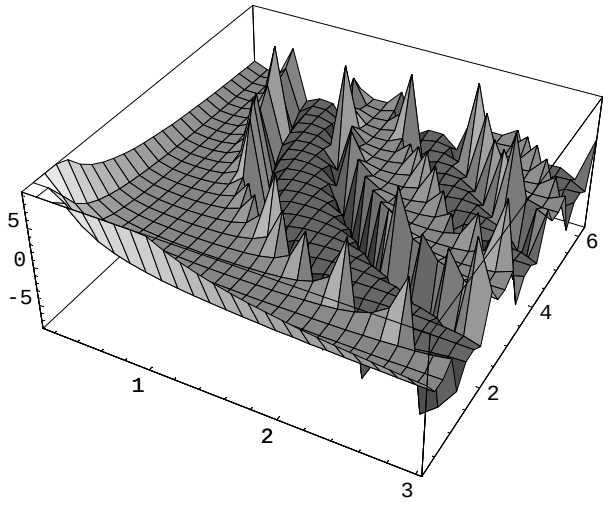
Probiere die Options ein wenig aus!

■ Essaie un peu les options!

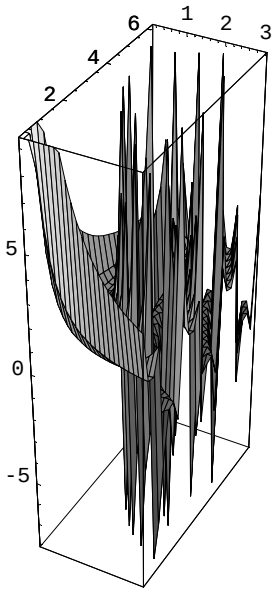
In[76]:=
```
?ClipFill
```

```
ClipFill is an option for Plot3D and related functions
   that specifies how clipped parts of surfaces are to be drawn.
```
Mehr...

In[77]:=
```
Plot3D[1/Sin[x y],{x, 0.1, Pi-0.1}, {y, 0.1, 2 Pi},
ClipFill -> None];
```



In[78]:=
```
Plot3D[1/Sin[x y],{x, 0.1, Pi-0.1}, {y, 0.1, 2 Pi},
 BoxRatios -> {1,2,4}];
```



4.8. Arbeiten mit Daten
▮ Travailler avec des données

4.8.1. Allgemeines
▮ Notions générales

Z.B. mit "ListPlot" und "ListPlot3D"

▮ P.ex. avec "ListPlot" et "ListPlot3D"

In[79]:=
```
??ListPlot
```

```
ListPlot[{y1, y2, ... }] plots points corresponding to a list of values. The x
   coordinates are by default taken to be 1, 2, ... . ListPlot[{{x1, y1}, {x2,
   y2}, ... }] plots a list of points with specified x and y coordinates. Mehr...

Attributes[ListPlot] = {Protected}

Options[ListPlot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic,
  DefaultColor → Automatic, DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction,
  Epilog → {}, FormatType :→ $FormatType, Frame → False, FrameLabel → None, FrameStyle → Automatic,
  FrameTicks → Automatic, GridLines → None, ImageSize → Automatic, PlotJoined → False,
  PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[80]:= **??ListPlot3D**

```
ListPlot3D[array] generates a three-dimensional plot of a surface representing an
   array of height values. ListPlot3D[array, shades] generates a plot with each
   element of the surface shaded according to the specification in shades. Mehr...

Attributes[ListPlot3D] = {Protected}

Options[ListPlot3D] =
 {AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → True, AxesEdge → Automatic,
  AxesLabel → None, AxesStyle → Automatic, Background → Automatic, Boxed → True,
  BoxRatios → {1, 1, 0.4}, BoxStyle → Automatic, ClipFill → Automatic, ColorFunction → Automatic,
  ColorFunctionScaling → True, ColorOutput → Automatic, DefaultColor → Automatic,
  DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction, Epilog → {}, FaceGrids → None,
  FormatType :→ $FormatType, HiddenSurface → True, ImageSize → Automatic, Lighting → True,
  LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]},
    {{0., 1., 1.}, RGBColor[0, 0, 1]}}, Mesh → True, MeshRange → Automatic, MeshStyle → Automatic,
  Plot3Matrix → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,
  Prolog → {}, Shading → True, SphericalRegion → False, TextStyle :→ $TextStyle, Ticks → Automatic,
  ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

Erst Daten generieren:

■ D'abort générer des données:

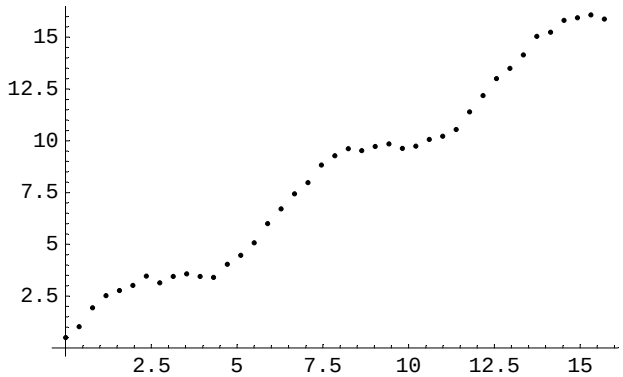
In[81]:=
```
data = N[Table[{x, Sin[x] + x + 0.5 Random[]},
                    {x, 0, 5 Pi, Pi/8}]]
```

Out[81]=
```
{{0., 0.495083}, {0.392699, 1.02267}, {0.785398, 1.93979},
 {1.1781, 2.52274}, {1.5708, 2.77231}, {1.9635, 3.01449}, {2.35619, 3.47185},
 {2.74889, 3.13403}, {3.14159, 3.44368}, {3.53429, 3.5747}, {3.92699, 3.45122},
 {4.31969, 3.40938}, {4.71239, 4.04212}, {5.10509, 4.47738},
 {5.49779, 5.07444}, {5.89049, 6.00353}, {6.28319, 6.71473},
 {6.67588, 7.43788}, {7.06858, 7.98662}, {7.46128, 8.82733}, {7.85398, 9.27882},
 {8.24668, 9.62249}, {8.63938, 9.5258}, {9.03208, 9.72418}, {9.42478, 9.85453},
 {9.81748, 9.63944}, {10.2102, 9.7351}, {10.6029, 10.0676}, {10.9956, 10.2238},
 {11.3883, 10.5419}, {11.781, 11.3973}, {12.1737, 12.1772}, {12.5664, 12.9925},
 {12.9591, 13.4962}, {13.3518, 14.151}, {13.7445, 15.041}, {14.1372, 15.2336},
 {14.5299, 15.812}, {14.9226, 15.9381}, {15.3153, 16.0749}, {15.708, 15.8728}}
```

Dann Daten ploten:

■ Ensuite "ploter" les données:

In[82]:=
```
dataplot = ListPlot[data];
```



Polynom durch diese Punkte als Funktion von x:

Polynôme à travers ces points comme une fonction de x:

In[83]:= **??InterpolatingPolynomial**

```
InterpolatingPolynomial[data, var] gives a polynomial in the variable
   var which provides an exact fit to a list of data. The data can have the
   forms {{x1, f1}, {x2, f2}, ... } or {f1, f2, ... }, where in the second
   case, the xi are taken to have values 1, 2, ... . The fi can be replaced
   by {fi, dfi, ddfi, ... }, specifying derivatives at the points xi. Mehr...

Attributes[InterpolatingPolynomial] = {Protected}

Options[InterpolatingPolynomial] = {Modulus → 0}
```

In[84]:= **InterpolatingPolynomial[data,x]**

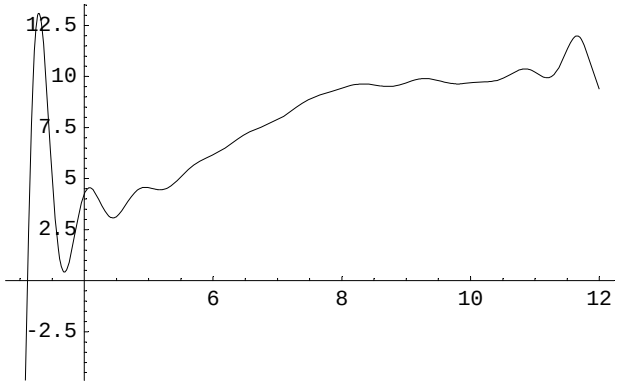
Out[84]=

$$
\begin{aligned}
&0.495083 + \\
&(1.3435 + (1.26296 + (-1.9917 + (1.26931 + (-0.356241 + (-0.0111603 + (-0.0924798 + \\
&(0.271334 + (-0.309284 + (0.230771 + \\
&(-0.130027 + (0.0591378 + (-0.0226376 + \\
&(0.00753882 + (-0.00225315 + (0.00062332 + \\
&(-0.000164065 + (0.0000417999 + \\
&(-0.0000103277 + (2.44697 \times 10^{-6} + \\
&(-5.46525 \times 10^{-7} + (1.13052 \times 10^{-7} + \\
&(-2.1273 \times 10^{-8} + (3.55527 \times 10^{-9} + \\
&(-5.03498 \times 10^{-10} + (5.24806 \times 10^{-11} + \\
&(-1.05591 \times 10^{-12} + (-1.37654 \times 10^{-12} + \\
&(5.01287 \times 10^{-13} + (-1.25382 \times 10^{-13} + \\
&(2.64785 \times 10^{-14} + (-5.02029 \times 10^{-15} + \\
&(8.77875 \times 10^{-16} + (-1.43593 \times 10^{-16} + \\
&(2.21492 \times 10^{-17} + (-3.23784 \times 10^{-18} + \\
&(4.49951 \times 10^{-19} + (-5.956 \times 10^{-20} + \\
&(7.5195 \times 10^{-21} - 9.06216 \times 10^{-22} (-15.3153 + \\
&x)) (-14.9226 + x)) (-14.5299 + x)) \\
&(-14.1372 + x)) (-13.7445 + x)) \\
&(-13.3518 + x)) (-12.9591 + x)) \\
&(-12.5664 + x)) (-12.1737 + x)) (-11.781 + \\
&x)) (-11.3883 + x)) (-10.9956 + x)) \\
&(-10.6029 + x)) (-10.2102 + x)) \\
&(-9.81748 + x)) (-9.42478 + x)) \\
&(-9.03208 + x)) (-8.63938 + x)) \\
&(-8.24668 + x)) (-7.85398 + x)) \\
&(-7.46128 + x)) (-7.06858 + x)) \\
&(-6.67588 + x)) (-6.28319 + x)) \\
&(-5.89049 + x)) (-5.49779 + x)) \\
&(-5.10509 + x)) (-4.71239 + x)) \\
&(-4.31969 + x)) (-3.92699 + x)) \\
&(-3.53429 + x)) (-3.14159 + x)) (-2.74889 + x)) \\
&(-2.35619 + x)) (-1.9635 + x)) (-1.5708 + x)) \\
&(-1.1781 + x)) (-0.785398 + x)) (-0.392699 + x)) (0. + x)
\end{aligned}
$$

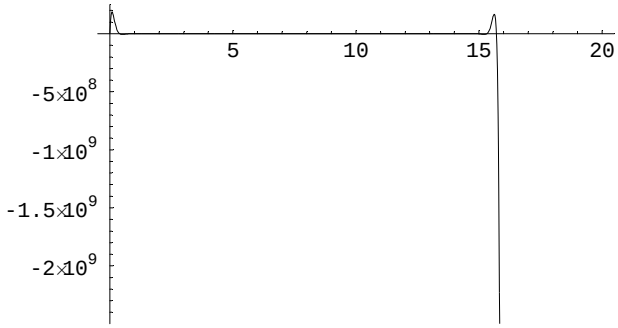
In[85]:= **Chop[InterpolatingPolynomial[data,x]]**

Out[85]=
$$0.495083 + (1.3435 + (1.26296 + (-1.9917 + (1.26931 + (-0.356241 + (-0.0111603 + (-0.0924798 + (0.271334 + (-0.309284 + (0.230771 + (-0.130027 + (0.0591378 + (-0.0226376 + (0.00753882 + (-0.00225315 + (0.00062332 + (-0.000164065 + (0.0000417999 + (-0.0000103277 + (2.44697\times10^{-6} + (-5.46525\times10^{-7} + (1.13052\times10^{-7} + (-2.1273\times10^{-8} + (3.55527\times10^{-9} - 5.03498\times10^{-10} (-9.42478 + x))(-9.03208 + x))(-8.63938 + x))(-8.24668 + x))(-7.85398 + x))(-7.46128 + x))(-7.06858 + x))(-6.67588 + x))(-6.28319 + x))(-5.89049 + x))(-5.49779 + x))(-5.10509 + x))(-4.71239 + x))(-4.31969 + x))(-3.92699 + x))(-3.53429 + x))(-3.14159 + x))(-2.74889 + x))(-2.35619 + x))(-1.9635 + x))(-1.5708 + x))(-1.1781 + x))(-0.785398 + x))(-0.392699 + x))x$$

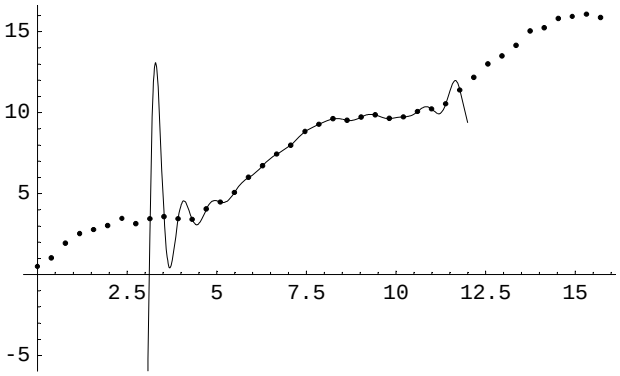
In[86]:=
```
interplot = Plot[Evaluate[
    InterpolatingPolynomial[data,x]],{x,3,12}];
```



In[87]:=
```
Plot[Evaluate[
    InterpolatingPolynomial[data,x]],{x,0,20}];
```



In[88]:= **Show[dataplot, interplot];**



4.8.2. "Regressionskurven"
"Courbes de régression"

Arbeiten mit "Fit":

Travailler avec "Fit":

In[89]:= **??Fit**

```
Fit[data, funs, vars] finds a least-squares fit to a list of data as a linear combination of
   the functions funs of variables vars. The data can have the form {{x1, y1, ... ,
   f1}, {x2, y2, ... , f2}, ... }, where the number of coordinates x, y, ...  is equal
   to the number of variables in the list vars. The data can also be of the form {f1,
   f2, ... }, with a single coordinate assumed to take values 1, 2, ... . The argument
   funs can be any list of functions that depend only on the objects vars. Mehr...

Attributes[Fit] = {Protected}
```

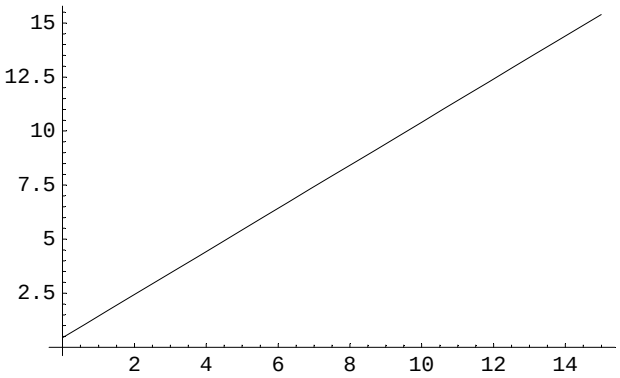
Gerade durch die Punktwolke:

En ligne droite à travers le nuage de points:

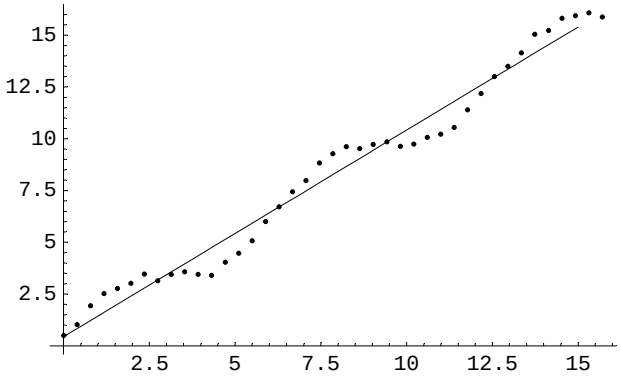
In[90]:= **trent = Fit[data, {1, x}, x]**

Out[90]= 0.446685 + 0.99655 x

In[91]:= **trentplot1 = Plot[trent,{x,0,15}];**



In[92]:= **Show[trentplot1, dataplot];**



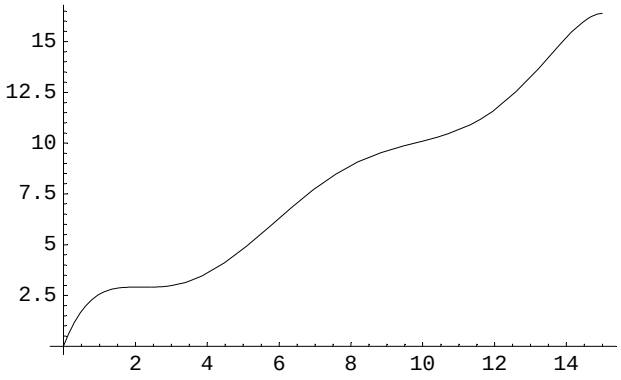
Polynomkurve 6. Grades durch die Punktwolke:

■ Courbe polynomiale de 6e degré à travers le nuage de points:

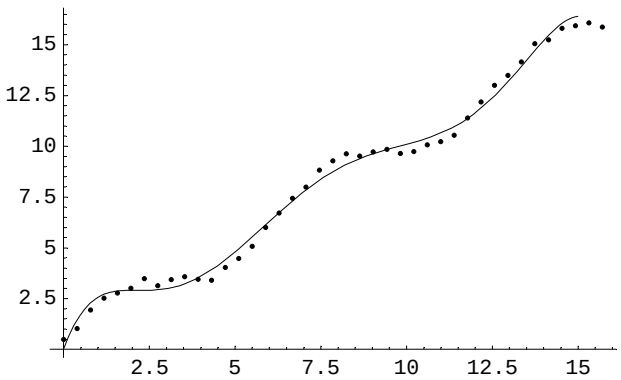
In[93]:=
```
trent6 = Fit[data, {1, x, x^2, x^3, x^4,
                    x^5, x^6}, x]
```

Out[93]= $0.00723834 + 4.68972\,x - 2.80482\,x^2 + 0.758888\,x^3 - 0.0930131\,x^4 + 0.00527162\,x^5 - 0.000112239\,x^6$

In[94]:= **trentplot6 = Plot[trent6,{x,0,15}];**



In[95]:= **Show[trentplot6, dataplot];**



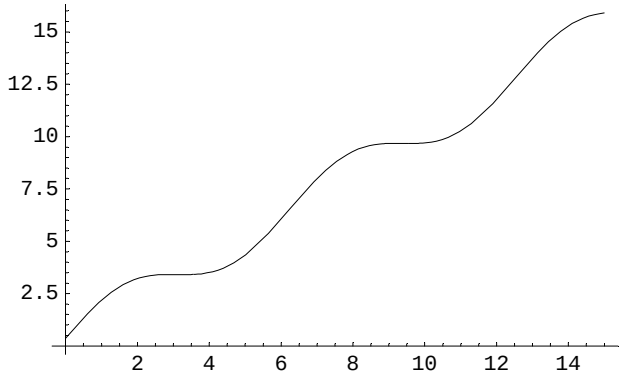
Kurve als Ueberlagerung von Gerade und Cosinus- sowie
Sinusfunktion durch die Punktwolke:

■ Courbe comme superposition d'une droite et d'une fonction cosinus ainsi que sinus à travers le nuage de points:

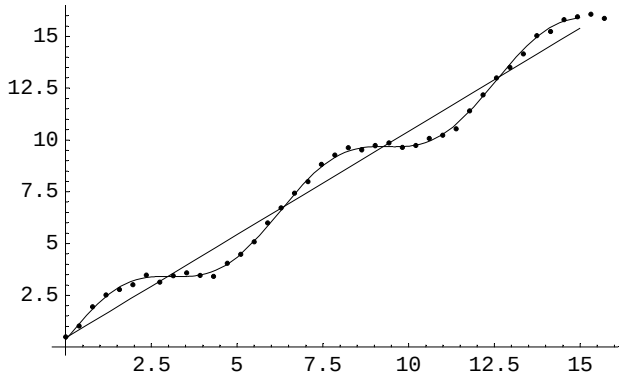
In[96]:= **trentcc = Fit[data, {1, x, Cos[x], Sin[x]}, x]**

Out[96]= 0.317202 + 0.997138 x + 0.0400338 Cos[x] + 1.01829 Sin[x]

In[97]:= **trentplotcc = Plot[trentcc,{x,0,15}];**

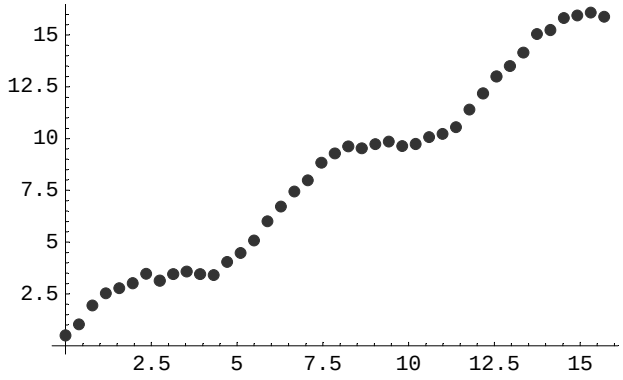


In[98]:= **Show[trentplotcc, trentplot1, dataplot];**



Optionen: ■ Options:

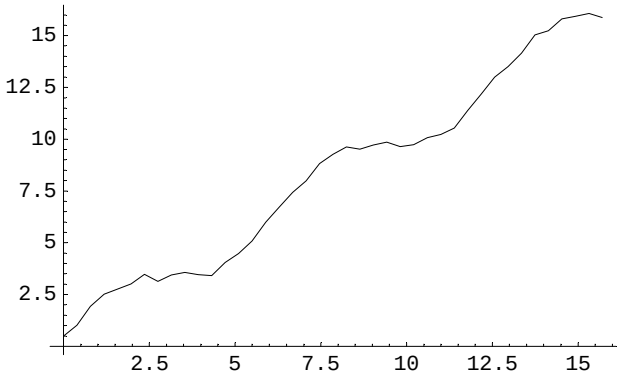
In[99]:= **dataplot1 = ListPlot[data, PlotStyle->{PointSize[0.02], GrayLevel[0.2]}];**



Punkte verbinden:
■ Relier les points:

In[100]:=

```
dataplot2 = ListPlot[data,  PlotJoined->True];
```

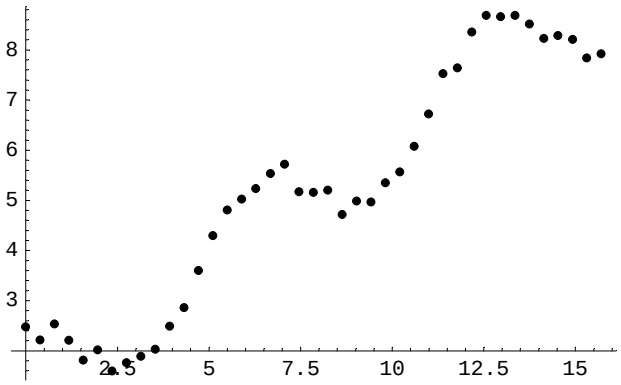


ListPlot kann auf einmal nur eine Datenmenge ploten, jedoch kann man solche Plots überlagern:

■ ListPlot ne peut "ploter" qu'un ensemble de données à la fois, on peut cependant superposer de tels plots:

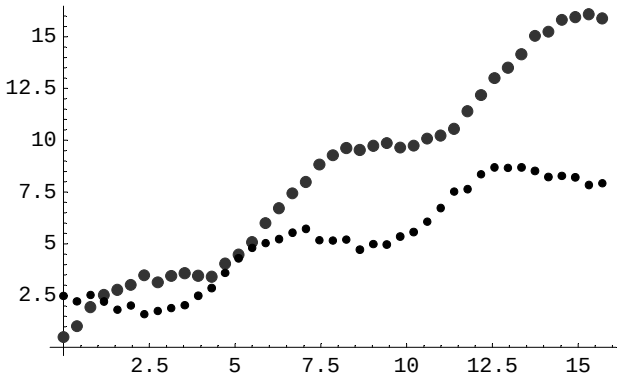
In[101]:=

```
data1 = N[Table[{x, 1 + Cos[x] + 0.5 x +
    0.5 Random[]}, {x, 0, 5 Pi, Pi/8}]];
dataplot3 = ListPlot[data1,
    PlotStyle->{PointSize[0.014]}];
```



In[103]:=

```
Show[dataplot1,dataplot3];
```



4.8.3. Unterdrückung des Graphik-Outputs (Zeit sparen!)
▮ Supprimer la sortie des graphes (économie de temps!)

Studiere: Zuerst Daten generieren (Tabelle mit 3 Listen zu je 5 Elementen, "DisplayFunction"):

▮ Etudie: D'abort générer les données (tableau à 3 listes à 5 éléments, "DisplayFunction"):

```
In[104]:=
myMatrix = Table[Random[], {3}, {5}]
```

```
Out[104]=
{{0.254526, 0.951846, 0.093696, 0.672727, 0.330327},
 {0.749974, 0.304512, 0.166322, 0.601042, 0.417562},
 {0.995414, 0.713963, 0.708045, 0.335777, 0.306895}}
```

```
In[105]:=
??DisplayFunction
```

```
DisplayFunction is an option for graphics and sound functions that specifies a
   function to apply to graphics and sound objects in order to display them. Mehr...

Attributes[DisplayFunction] = {Protected}
```

Graphiken rechnen, jedoch nicht einzeln ausgeben:

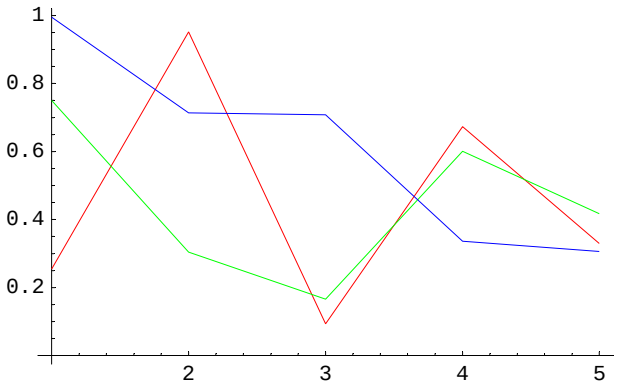
▮ Calculer des graphes, cependant ne pas les sortir:

```
In[106]:=
plot1 = ListPlot[myMatrix[[1]],
    PlotStyle->RGBColor[1,0,0], PlotJoined->True,
    DisplayFunction->Identity];
plot2 = ListPlot[myMatrix[[2]],
    PlotStyle->RGBColor[0,1,0], PlotJoined->True,
    DisplayFunction->Identity];
plot3 = ListPlot[myMatrix[[3]],
    PlotStyle->RGBColor[0,0,1], PlotJoined->True,
    DisplayFunction->Identity];
```

Graphiken ausgeben, alle zusammen in einem Bild:

▮ Sortir les graphes, tous ensemble en une image:

```
In[109]:=
Show[plot1, plot2, plot3,
    DisplayFunction :> $DisplayFunction];
```



Erklärungen: ▮ Explications:

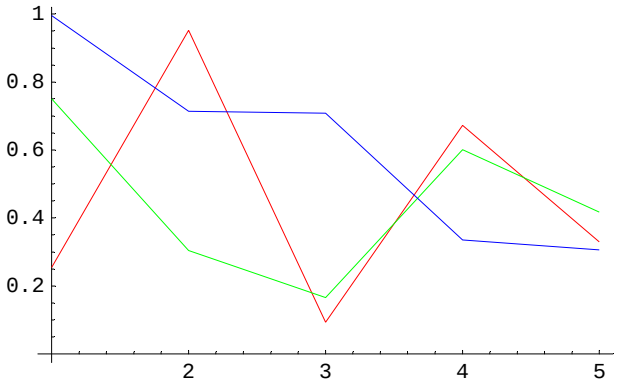
In[110]:=

```
??$DisplayFunction
```

```
$DisplayFunction gives the default setting
   for the option DisplayFunction in graphics functions. Mehr...

$DisplayFunction = Display[$Display, #1] &
```

In[111]:=

```
Show[plot1, plot2, plot3,
    DisplayFunction -> $DisplayFunction];
```



In[112]:=

```
??->
```

```
lhs -> rhs represents a rule that transforms lhs to rhs. Mehr...

Attributes[Rule] = {Protected, SequenceHold}
```

In[113]:=

```
??:>
```

```
lhs :> rhs represents a rule that transforms
   lhs to rhs, evaluating rhs only when the rule is used. Mehr...

Attributes[RuleDelayed] = {HoldRest, Protected, SequenceHold}
```

4.8.4. Plot von 3-dimensionalen Daten
▮ Plot de données à 3 dimensions

Erklärungen: ▮ Explications:

In[114]:=

```
??ListContourPlot
```

```
ListContourPlot[array] generates a contour plot from an array of height values. Mehr…

Attributes[ListContourPlot] = {Protected}

Options[ListContourPlot] =
 {AspectRatio → 1, Axes → False, AxesLabel → None, AxesOrigin → Automatic,
  AxesStyle → Automatic, Background → Automatic, ColorFunction → Automatic,
  ColorFunctionScaling → True, ColorOutput → Automatic, ContourLines → True,
  Contours → 10, ContourShading → True, ContourSmoothing → True, ContourStyle → Automatic,
  DefaultColor → Automatic, DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction,
  Epilog → {}, FormatType :→ $FormatType, Frame → True, FrameLabel → None,
  FrameStyle → Automatic, FrameTicks → Automatic, ImageSize → Automatic,
  MeshRange → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[115]:=

```
??ListDensityPlot
```

```
ListDensityPlot[array] generates a smooth density plot from an array of values. Mehr…

Attributes[ListDensityPlot] = {Protected}

Options[ListDensityPlot] =
 {AspectRatio → 1, Axes → False, AxesLabel → None, AxesOrigin → Automatic,
  AxesStyle → Automatic, Background → Automatic, ColorFunction → Automatic,
  ColorFunctionScaling → True, ColorOutput → Automatic, DefaultColor → Automatic,
  DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction, Epilog → {},
  FormatType :→ $FormatType, Frame → True, FrameLabel → None, FrameStyle → Automatic,
  FrameTicks → Automatic, ImageSize → Automatic, Mesh → True, MeshRange → Automatic,
  MeshStyle → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

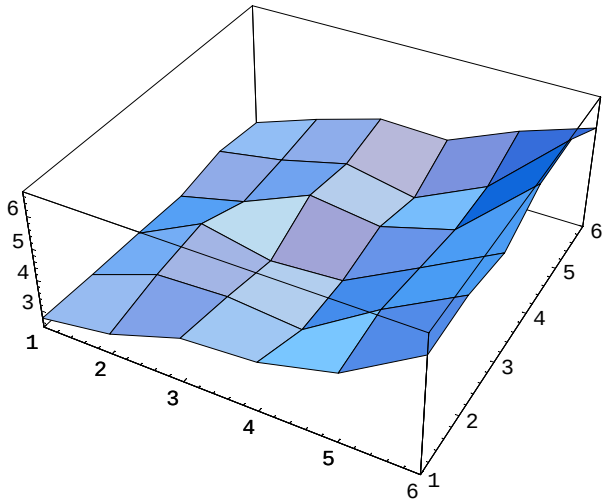
Erst Daten generieren:

- D'abort générer des données:

In[116]:=

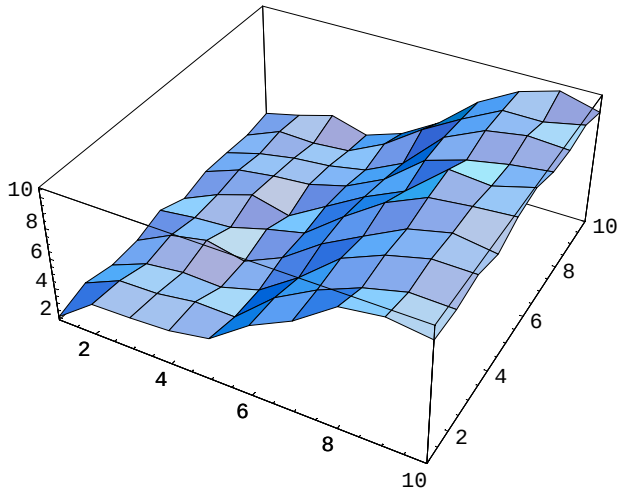
```
array1 = Table[x + Sin[x] + Random[], {6},
    {x, 1, 6}]; Print[array1];
                ListPlot3D[array1];
```

```
{{2.74784, 3.09658, 3.86844, 4.02567, 4.62956, 6.11237},
 {2.66127, 3.47913, 3.59256, 3.38045, 4.90903, 6.19672},
 {2.62019, 3.71623, 3.2591, 3.41482, 4.65347, 5.92647},
 {2.54189, 3.08551, 4.03955, 3.74104, 4.40572, 6.5899},
 {2.83353, 3.21986, 3.77844, 3.33004, 4.44466, 6.63936},
 {2.65899, 3.42631, 4.09326, 4.02472, 4.99064, 5.76146}}
```



In[118]:=

```
array2 = Table[x + Sin[x] + Random[], {10},
    {x, 1, 10}];    ListPlot3D[array2];
```

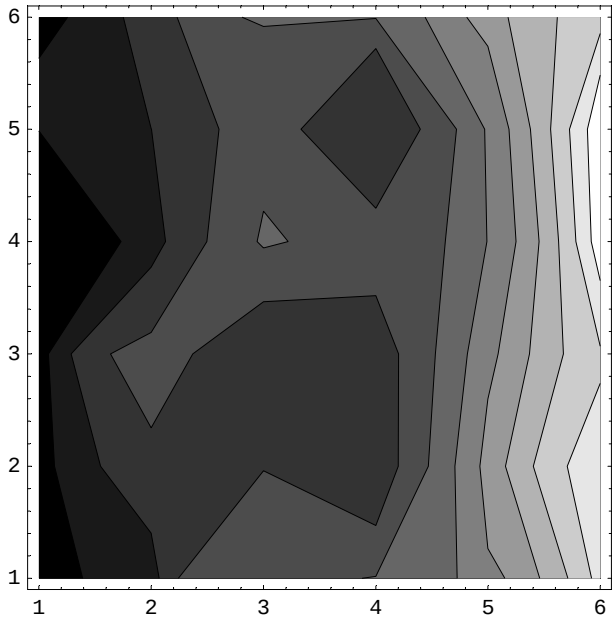


Zur besseren Sichtbarmachung zeichnet *Mathematica* eine Fläche und nicht eine Punktwolke im 3-dimensionalen Raum. ListContourPlot zeichnet davon eine
Höhenlinienkarte:

■ Pour une meilleure perceptibilité, *Mathematica* dessine une surface et non pas un nuage de points dans l'espace à trois dimensions. ListContourPlot dessine une carte indiquant les courbes à niveau:

In[119]:=

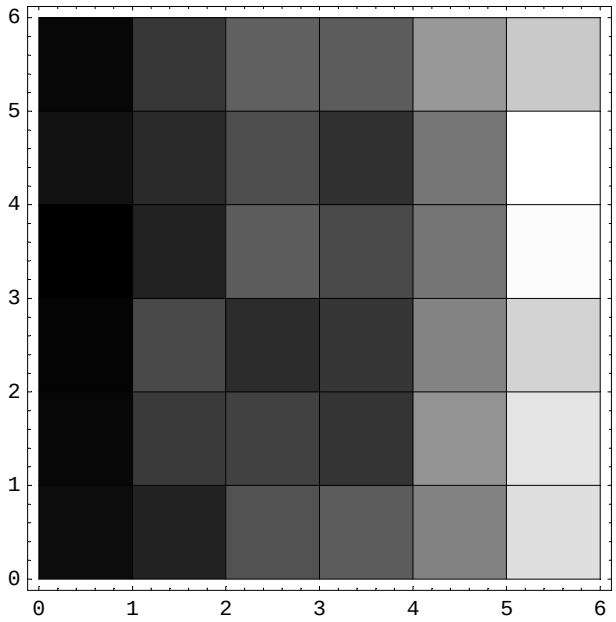
```
ListContourPlot[array1];
```



Eine andere Möglichkeit zur Veranschaulichung bietet ListDensityPlot:
ListDensityPlot offre une autre possibilité d'illustration:

In[120]:=

```
ListDensityPlot[array1];
```



4.9. Eingebaute Graphikelemente oder Blöcke
▮ Eléments ou blocs graphiques incorporés

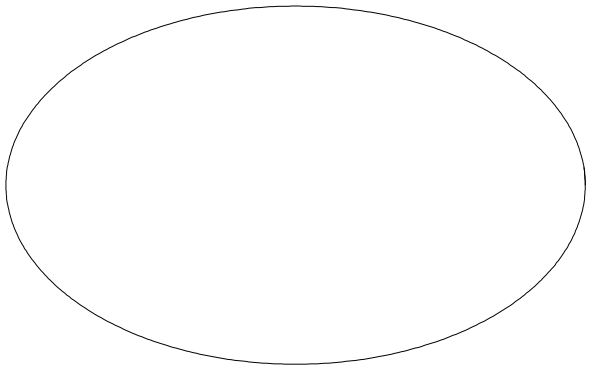
4.9.1. Allgemeines
▮ Notions générales

Mathematica bietet die Möglichkeit, Linien, Punkte, Kreise, Scheiben und Polygone direkt zu zeichnen. Beispiele:

▮ *Mathematica* offre la possibilité de dessigner directement des lignes, points, cercles, disques, polygones. Exemples:

In[121]:=

```
Show[Graphics[{Circle[{0,0},.1]}]];
```



Wieso ist aus dem Kreis eine Ellipse geworden?

▮ Pourquoi le cercle est-il devenu une ellipse?

In[122]:=

```
Options[Graphics][[1]]
```

Out[122]=

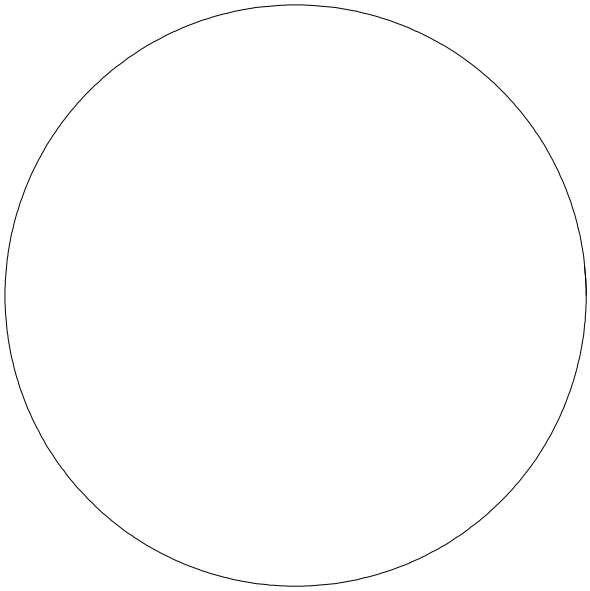
$$\texttt{AspectRatio} \to \frac{1}{\texttt{GoldenRatio}}$$

Man muss also das Achsenverhältnis anpassen!

▮ Il faut donc adapter le rapport des axes:

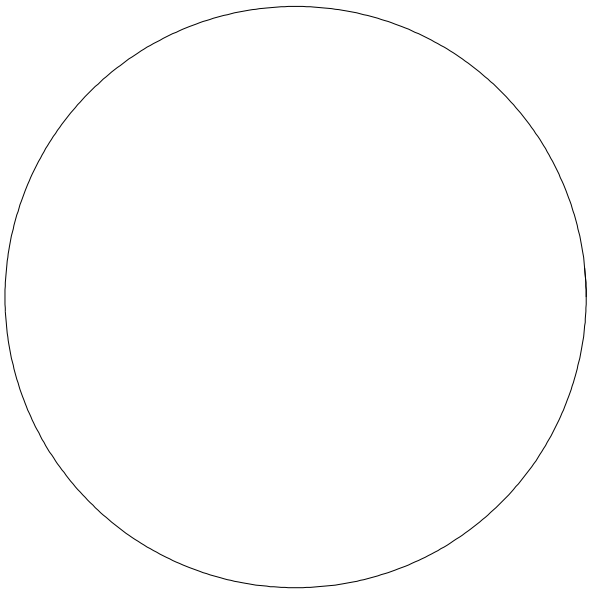
In[123]:=

```
Show[Graphics[{Circle[{0,0},.1]},
    AspectRatio->1]];
```



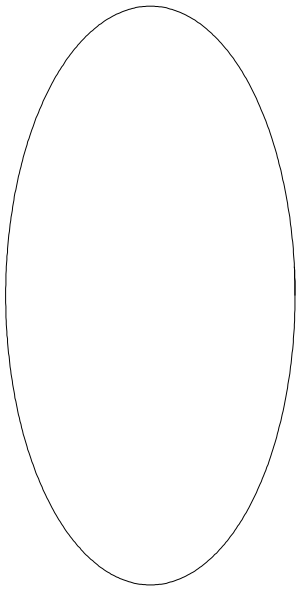
In[124]:=

```
Show[Graphics[{Circle[{0,0},.1]},
    AspectRatio->Automatic]];
```



In[125]:=

```
Show[Graphics[{Circle[{0,0},.1]},
    AspectRatio->2]];
```



Oder: ■ Ou:

In[126]:=

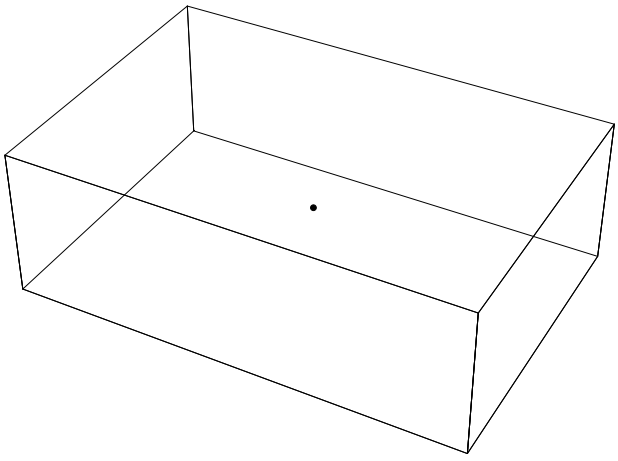
```
Show[Graphics[{
    PointSize[0.05], Point[{0, 0}], Point[{1, 0}],
    Line[{{0, 0},{1, 0}}],
    Circle[{0,0},.1], Circle[{1,0},.1]},
    AspectRatio->Automatic]];
```

Oder ein Punkt im 3-dimensionalen Raum:

■ Ou un point dans l'espace à trois dimensions:

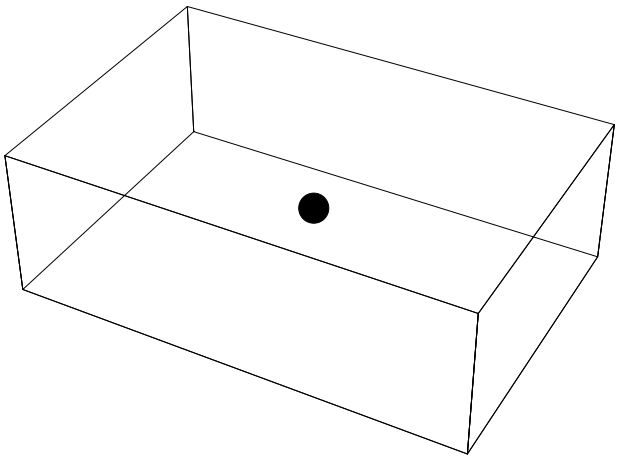
In[127]:=

```
Show[Graphics3D[Point[{3,2,1}]]];
```



In[128]:=

```
Show[Graphics3D[{PointSize[0.05],
        Point[{3,2,1}]}]];
```



Oder erst Liste generieren:

■ Ou générer d'abort une liste:

In[129]:=

```
pts = Table[Random[], {4}, {3}]
```

Out[129]=

```
{{0.65511, 0.860532, 0.213686}, {0.705759, 0.0158655, 0.25467},
 {0.938153, 0.333522, 0.155321}, {0.299344, 0.426078, 0.843024}}
```

Dann Liste manipulieren:

■ Ensuite manipuler la liste:

In[130]:=

```
??Map
```

```
Map[f, expr] or f /@ expr applies f to each element on the first level in expr. Map[
   f, expr, levelspec] applies f to parts of expr specified by levelspec. Mehr...

Attributes[Map] = {Protected}

Options[Map] = {Heads → False}
```

In[131]:=

```
graphicPts = Map[Point, pts]
```

Out[131]=

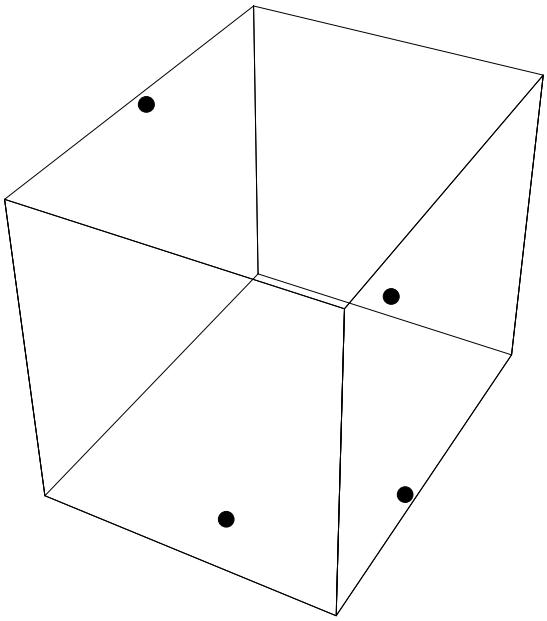
```
{Point[{0.65511, 0.860532, 0.213686}], Point[{0.705759, 0.0158655, 0.25467}],
 Point[{0.938153, 0.333522, 0.155321}], Point[{0.299344, 0.426078, 0.843024}]}
```

Dann davon Plot machen:

■ Ensuite faire un plot:

In[132]:=

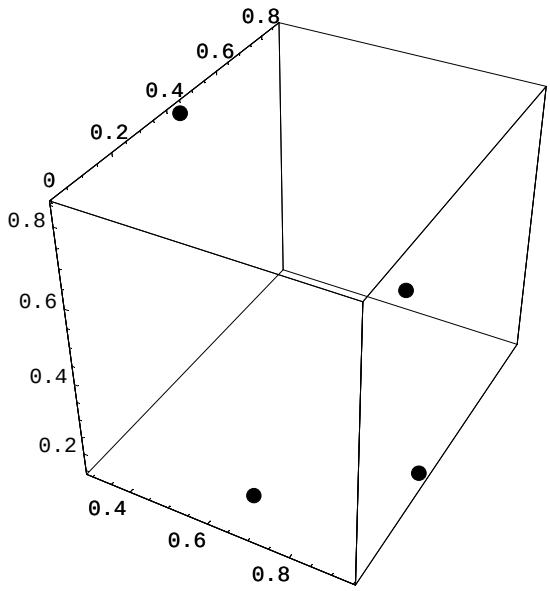
```
Show[Graphics3D[{PointSize[0.03], graphicPts}]];
```



Mit Skala: ■ Avec échelle:

In[133]:=

```
Show[Graphics3D[{PointSize[0.03], graphicPts}],
                Axes->Automatic];
```



4.9.2. Umwandlung von graphischen Objekten: ■ Transformation d'objets graphiques:

Mit Hilfe welcher Graphic-Objekte wird ein Plot ausgeführt?

■ A l'aide de quels objets graphiques exécute-t-on un plot?

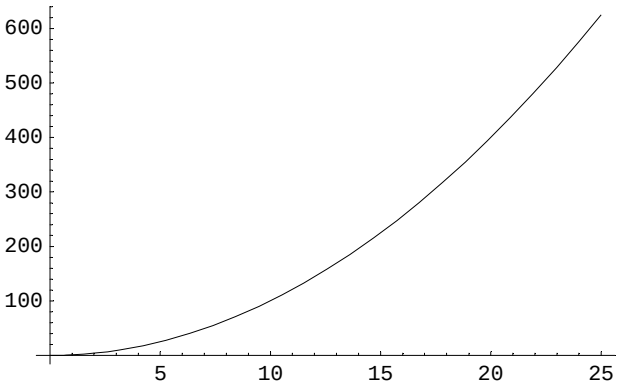
In[134]:=

```
??InputForm
```

```
InputForm[expr] prints as a version of expr suitable for input to Mathematica. Mehr...

Attributes[InputForm] = {Protected}

Options[InputForm] = {NumberMarks :→ $NumberMarks}
```

In[135]:=

```
basicPlot = Plot[x^2, {x, 0, 25}];
```



In[136]:=

```
InputForm[basicPlot]
```

Out[136]//InputForm=

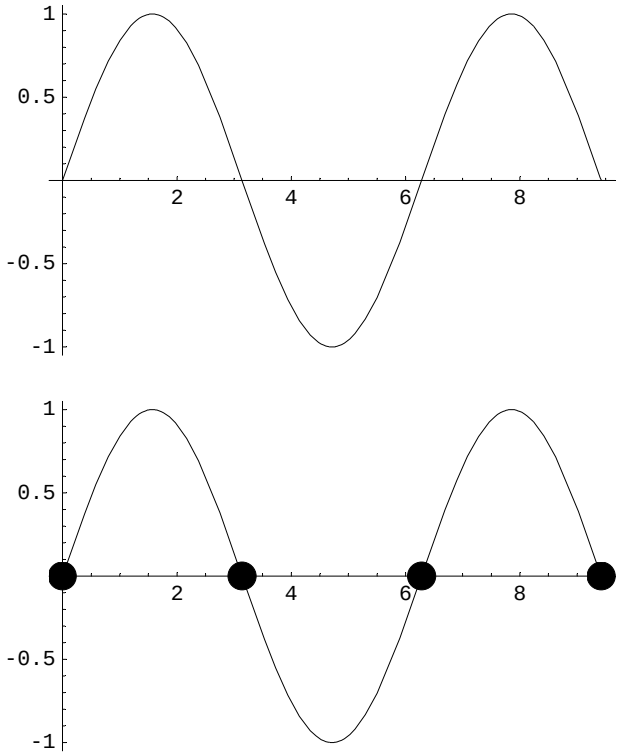
```
Graphics[
 {{Line[{{1.0416666666666667*^-6,
      1.0850694444444446*^-12}, {0.030860679008700317,
      0.0009523815088780364}, {0.05893770458747362,
      0.003473653022040309}, {0.0913469976846706,
      0.008344273986003216}, {0.1219837330957231,
      0.01488003113996861}, {0.17647472934441902,
      0.031143330097185947}, {0.23562300449197435,
      0.05551820024582496}, {0.30076105590546476,
      0.0904572127493701}, {0.36981477027355836,
      0.13676296431248475}, {0.4937042690755045,
      0.24374390530337817}, {0.6215800864064731,
      0.3863618038170786}, {0.7413759154314858,
      0.5496382479818737}, {1.014174789322895,
      1.0285505032981384}, {1.2924370222301953,
      1.6703934564312544}, {1.5470261652831419,
      2.393289956070663}, {2.1202199964843422,
      4.495332833492064}, {2.618584361103532,
      6.856984056215994}, {3.1625718515622814,
      10.001860716294077}, {4.229514120963115,
      17.888789699426393}, {5.256668264192063,
      27.632561239763998}, {6.3482009479157915,
      40.29965527511895}, {7.399945505467634,
      54.759193483890634}, {8.41190193684759,
      70.76009419514024}, {9.488236908722328,
      90.02663963604064}, {10.52478375442518,
      110.77107307741218}, {11.521542473956146,
      132.7459409791755}, {12.582679733981891,
      158.3238292879586}, {13.604028867835751,
      185.06960143690847}, {14.689756542184393,
      215.78894726864917}, {15.735696090361147,
      247.6121314482071}, {16.741847512366014,
      280.2894581273161}, {17.812377474865663,
      317.28079130710165}, {18.843119311193426,
      355.06314537587065}, {19.834073021349305,
      393.39045261621635}, {20.889405271999962,
      436.36725261785983}, {21.904949396478735,
      479.82680806229405}, {22.984872061452286,
      528.3043436813299}, {24.025006600253953,
      577.2009421422459}, {24.999998958333332,
      624.9999479166677}}]}}, {PlotRange -> Automatic,
   AspectRatio -> GoldenRatio^(-1),
   DisplayFunction :> $DisplayFunction,
   ColorOutput -> Automatic, Axes -> Automatic,
   AxesOrigin -> Automatic, PlotLabel -> None,
   AxesLabel -> None, Ticks -> Automatic,
   GridLines -> None, Prolog -> {}, Epilog -> {},
   AxesStyle -> Automatic, Background -> Automatic,
   DefaultColor -> Automatic, DefaultFont :>
    $DefaultFont, RotateLabel -> True, Frame -> False,
   FrameStyle -> Automatic, FrameTicks -> Automatic,
   FrameLabel -> None, PlotRegion -> Automatic,
   ImageSize -> Automatic, TextStyle :> $TextStyle,
   FormatType :> $FormatType}]
```

Trick: Ausgabe mehrerer Plots auf einmal:

Truc: Emission de plusieurs plots à la fois:

In[137]:=

```
Show[
    Plot[Sin[x], {x, 0, 3 Pi}],
    Graphics[{PointSize[0.05],
    Table[Point[{n Pi, 0}], {n, 0, 3}]
    }]
    ];
```

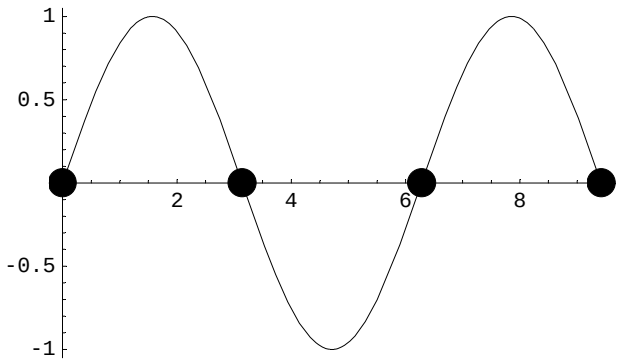


Hier sind 2 Plots gekommen. Damit nur der letzte kommt, muss man mit "DisplayFunction" arbeiten:

Voici 2 plots sortis. Pour faire sortir seulement le dernier plot, il faut travailler avec "DisplayFunction":

In[138]:=

```
Show[
    Plot[Sin[x], {x, 0, 3 Pi},
        DisplayFunction->Identity],
        Graphics[{PointSize[0.05],
        Table[Point[{n Pi, 0}], {n, 0, 3}]
        }], DisplayFunction->$DisplayFunction
        ];
```



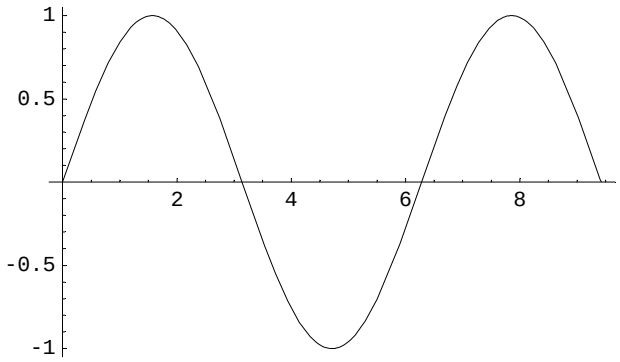
4.10. Beschriftung von Graphiken (Labels)
■ Doter des graphiques d'inscriptions (Labels)

Beschriftung der x-Achse. Zuerst ohne:

■ Faire des inscriptions sur l'axe x. D'abord sans inscriptions:

In[139]:=

```
Plot[Sin[x], {x, 0, 3 Pi}];
```



Dann mit: ■ Ensuite avec inscription:

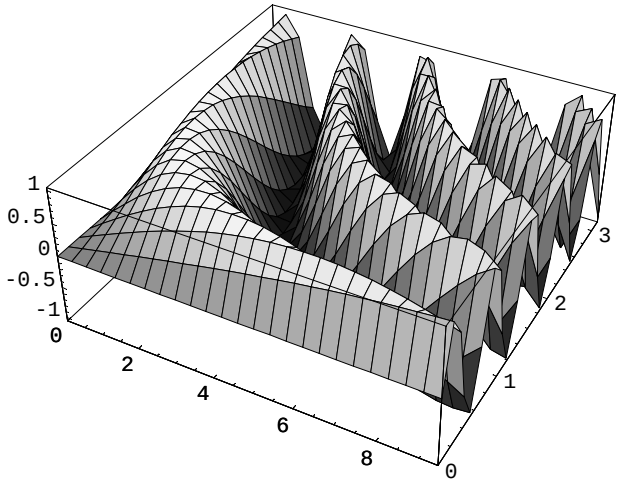
In[140]:=

```
??Ticks
```

```
Ticks is an option for graphics functions that specifies tick marks for axes. Mehr…

Attributes[Ticks] = {Protected}
```

In[141]:=

```
Plot3D[Sin[x y], {x, 0, 3 Pi}, {y, 0, Pi}];
```



Labels: ■ Labels:

In[142]:=

```
??PlotLabel
```

```
PlotLabel is an option for graphics functions that specifies an overall label for a plot.
 Mehr…

Attributes[PlotLabel] = {Protected}
```

In[143]:=

```
??FontForm
```

```
FontForm is an obsolete function, superseded by StyleForm. FontForm[expr, {"font", size}]
   specifies that expr should be printed in the specified font and size. Mehr…

Attributes[FontForm] = {Protected}
```

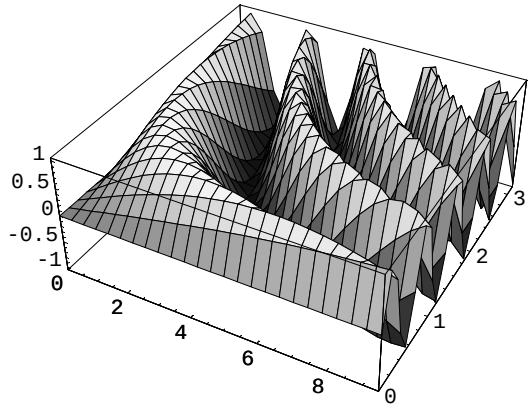
\n: Neue Linie

■ \n: Nouvelle ligne:

In[144]:=

```
Plot3D[Sin[x y], {x, 0, 3 Pi}, {y, 0, Pi},
    PlotLabel -> FontForm["Hallo friend!
    Plot3D[Sin[x y], \n
    {x, 0, 3 Pi}, {y, 0, Pi}]",
    {"Courier-Bold", 15}]];
```

Hallo friend! Plot3D[Sin[x y], {x, 0, 3 Pi}, {y, 0, Pi}]



Oder: ■ Ou:

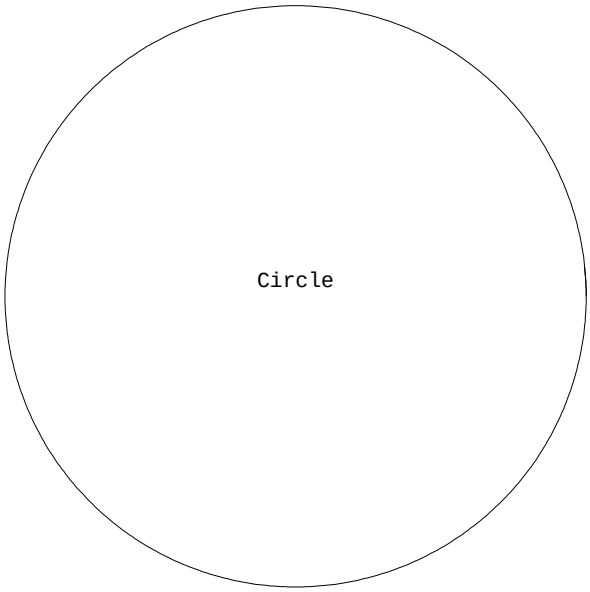
In[145]:=

```
??Text
```

```
Text[expr, coords] is a graphics primitive that represents text corresponding
   to the printed form of expr, centered at the point specified by coords. Mehr…

Attributes[Text] = {Protected}

Options[Text] = {Background → None, FormatType :→ Automatic, TextStyle :→ Automatic}
```

In[146]:=

```
Show[Graphics[{Circle[{0,0},1],
    Text["Circle",{0,0.05}]}],
    AspectRatio -> Automatic];
```



In[147]:=

4.11. Graphik-Pakete
▮ Paquets de graphiques

Zum Beispiel Rotationskörper:
▮ Par exemple corps de révolution:

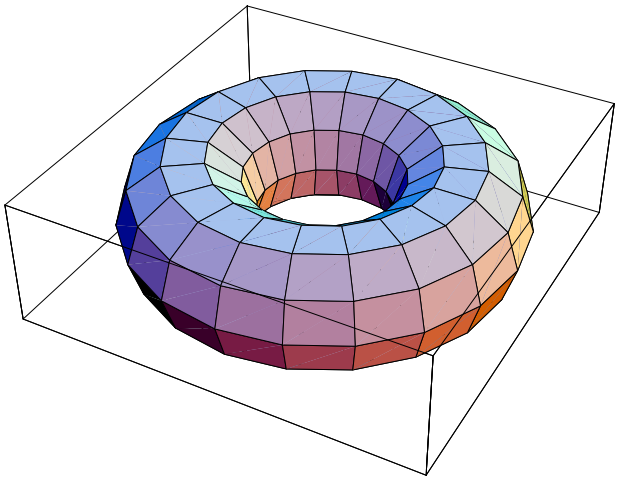
Package laden: ▮ Charger Package:

In[147]:=

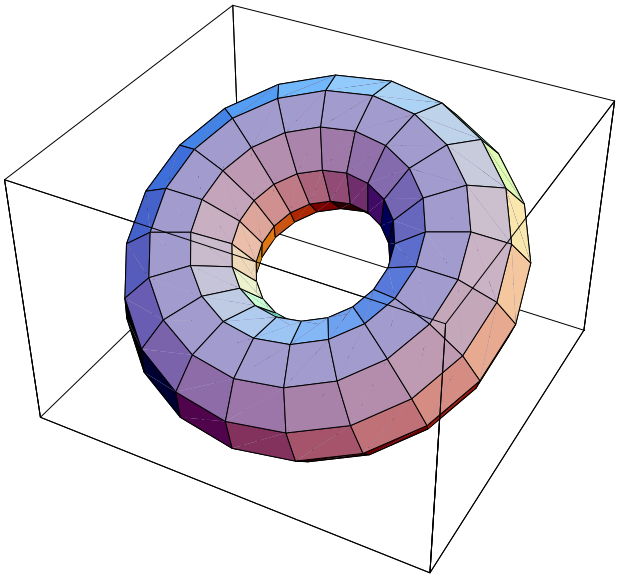
```
<< Graphics/Shapes.m
```

Beispiele: ▮ Exemples:

In[148]:=
```
Show[Graphics3D[Torus[]]];
```



In[149]:=
```
Show[RotateShape[Graphics3D[Torus[]],
        0, 5 Pi/6, 0]];
```



4.12. Animationen
∎ Animations

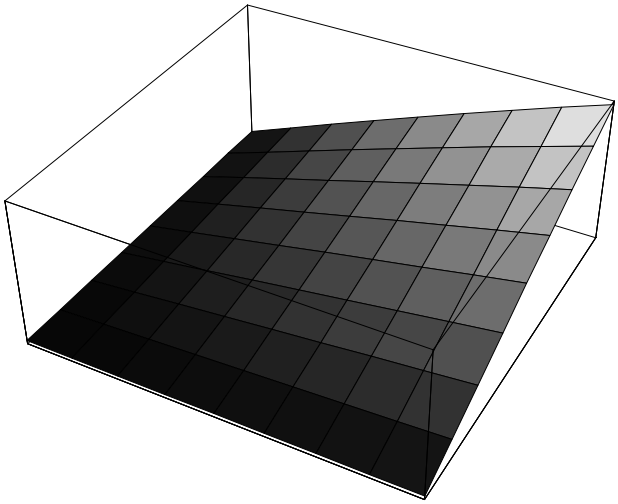
Auf dem Buchdeckel von Nancy Blachmans Buch:
∎ De la couverture du livre de Nancy Blachman:

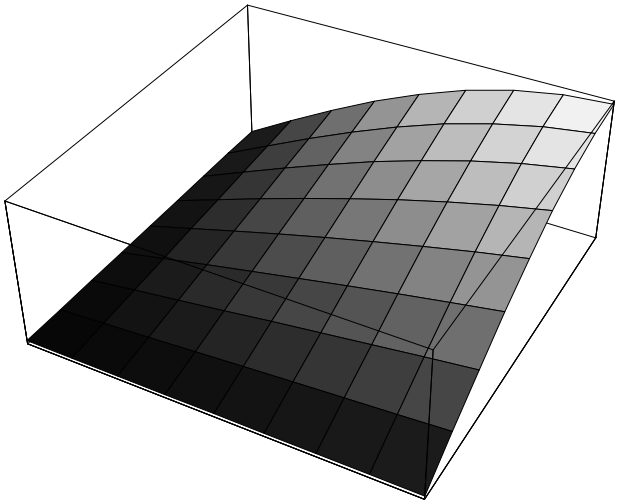
Zuerst Graphiken rechnen (hier nur wenige Bilder zur schnellen und kurzen Demonstration, Bilderzahl bitte im Praktikum selbst erhöhen):

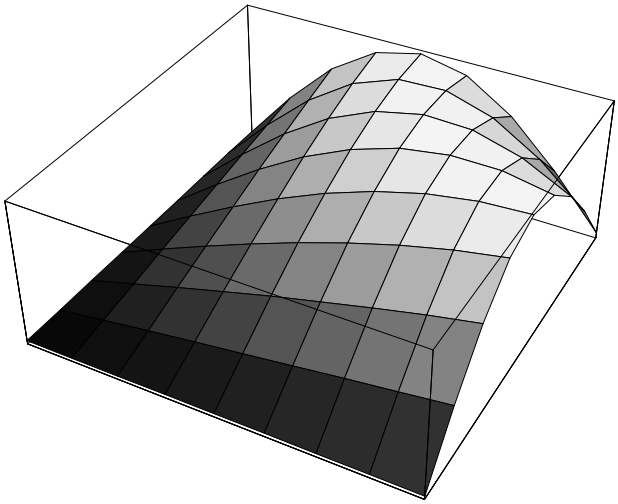
∎ D'abord calculer des graphiques (ici quelques images pour une démonstration rapide et brève, augmenter soi-même le nombre d'images lors du travail pratique):

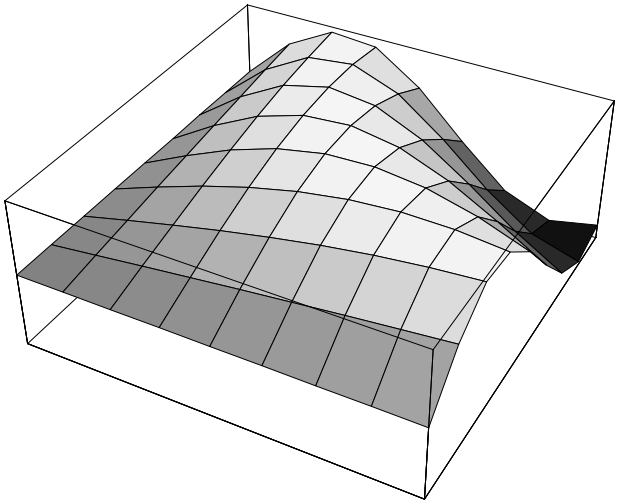
In[150]:=

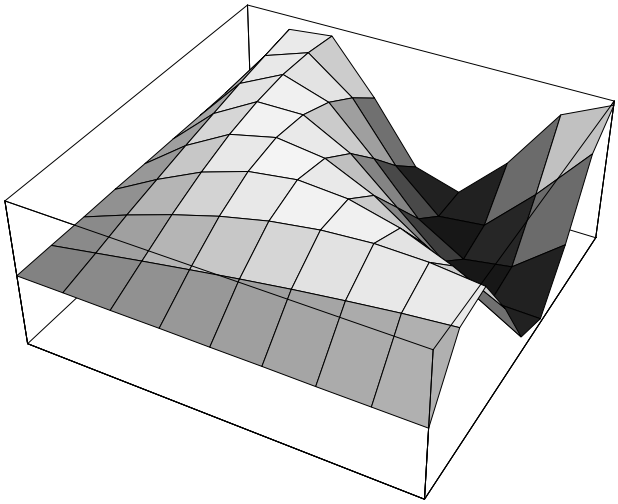
```
cover = Table[Plot3D[Sin[x y], {x, 0, t/2},
        {y, 0, t}, PlotPoints -> 9,
        Ticks -> None],
         {t, 1, 6, 3/4}]
```

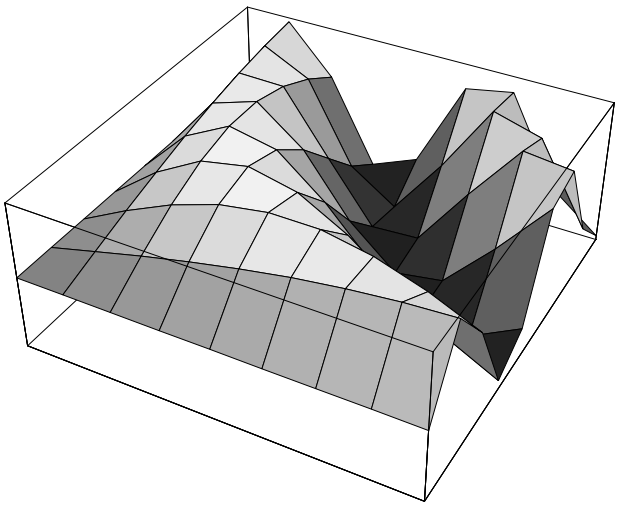


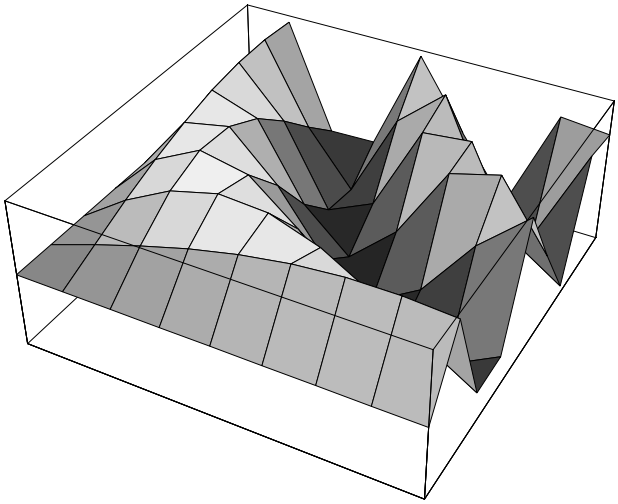












Out[150]=

```
{- SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
 - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -}
```

Vorgehen zum Animieren:
▮ Procédé d'animation:

New: ===>Go to Cell,Animate Selectet Graphics (Ctrl+Y)
Old: ===>
Jede der eben ausgegebenen Graphiken ist in einer eigenen Zelle untergebracht. Alle diese Graphikzellen wiederum bilden eine gemeinsame Zelle (vgl. von links aus 2. senkrechte Linie am rechten Rand). Diese Zelle ist mit der Maus anzuklicken. (Sie wird dann dunkel markiert.) Nun wählst Du mit der Maus im Menu ("Graph") neu "Cell" und dann "Animate Selected Graphics ....y". Du wirst sehen, was geschieht! Um die Sache anzuhalten, brauchst Du nur mit der Maus irgendwohin ausserhalb der eben gewählten Menuzeile zu "klicken".
▮ Chacun des graphiques qui viennent de sortir est placé dans une cellule à lui. Toutes ces cellules de graphiques constituent de nouveau une cellule en commun (compare de gauche la 2e ligne verticale sur le bord de droite). Il faut toucher cette ligne avec la souris. (Elle deviendra foncée.) Maintenant tu choisis avec la souris dans le menu ("Graph") nouveau "Cell" et ensuite "Animate Selected Graphics ....y". Regarde ce qui se passe! Pour arrêter le processus, il te suffit d'activer la souris n'importe où en dehors du menu que tu viens de choisir.

Eine verkleinerte Darstellung:
▮ Une représentation rapetissée:

In[151]:=

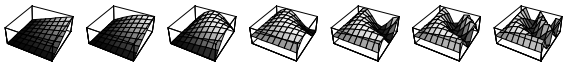
```
??GraphicsArray
```

```
GraphicsArray[{g1, g2, ... }] represents a row of graphics objects. GraphicsArray[{{g11,
   g12, ... }, ... }] represents a two-dimensional array of graphics objects. Mehr...

Attributes[GraphicsArray] = {Protected, ReadProtected}

Options[GraphicsArray] = {AspectRatio → Automatic, Axes → False, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
  ColorOutput → Automatic, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
  DisplayFunction :→ $DisplayFunction, Epilog → {}, FormatType :→ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → None, GraphicsSpacing → 0.1,
  GridLines → None, ImageSize → Automatic, PlotLabel → None, PlotRange → Automatic,
  PlotRegion → Automatic, Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → None}
```

In[152]:=
```
??GraphicsSpacing
```

```
GraphicsSpacing is an option for GraphicsArray
   which specifies the spacing between elements in the array. Mehr...

Attributes[GraphicsSpacing] = {Protected}
```

In[153]:=
```
Show[GraphicsArray[cover],
GraphicsSpacing -> 0.2];
```



4. 13. PostScript

PostScript

Erzeugen, ablegen, einlesen und betrachten von PostScript-Code
Générer, déposer, lire et observer le code PostScript

Einige Erklärungen zu Befehlen:

- Quelques explications quant aux ordres:

In[154]:=
```
??>>
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr...

Attributes[Put] = {Protected}
```

In[155]:=
```
??Display
```

```
Display[channel, graphics] writes graphics or sound to the specified output channel
   in Mathematica PostScript format. Display[channel, graphics, "format"] writes
   graphics or sound in the specified format. Display[channel, expr, "format"]
   writes boxes, cells or notebook expressions in the specified format. Mehr...

Attributes[Display] = {Protected, ReadProtected}

Options[Display] = {CharacterEncoding → {}, ImageOffset → Automatic,
  ImageResolution → Automatic, ImageRotated → False, ImageSize → Automatic}
```

In[156]:=
```
??<<
```

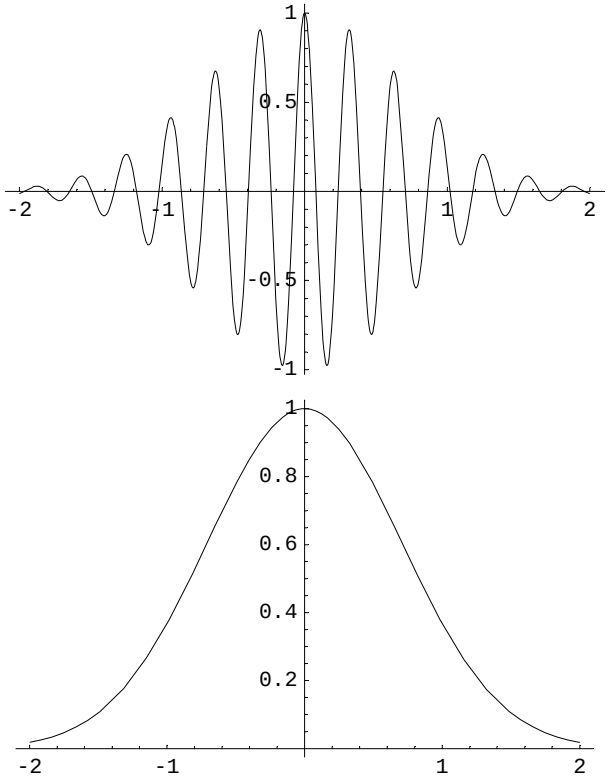
```
<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr...

Attributes[Get] = {Protected}

Options[Get] = {CharacterEncoding :→ $CharacterEncoding, Path :→ $Path}
```

Wir erzeugen eine Graphik:

- Nous générons un graphique:

In[157]:=

```
expCos = Plot[Exp[-x^2] Cos[20 x], {x, -2, 2}];
exp1   = Plot[Exp[-x^2], {x, -2, 2}];
```



Nun schauen wir uns den Inhalt des Files "postscript_test1.ps" auf dem Schirm an. Du siehst sofort, dass hier offenbar Koordinaten von gerechneten Punkten gespeichert worden sind. Arbeite dazu die folgenden Befehle ab:

■ Maintenant nous regardons le contenu des fichiers (files) "postscript_test1.ps" sur l'écran. Tu vois tout de suite qu'ici on a évidemment mémorisé les coordonnés des points calculés. Exécute les ordres suivants:

In[159]:=

```
?exp1
```

```
Global`exp1

exp1 =
 Graphics[{{Line[{{-2., 0.0183157}, {-1.92101, 0.0249653}, {-1.83773, 0.0341409}, {-1.75327,
       0.0462373}, {-1.66076, 0.0634096}, {-1.57856, 0.082756}, {-1.49063, 0.108395},
      {-1.31656, 0.176696}, {-1.14885, 0.267173}, {-0.987513, 0.377124},
      {-0.815873, 0.51394}, {-0.6506, 0.654895}, {-0.491692, 0.785244}, {-0.40333, 0.849867},
      {-0.322484, 0.901229}, {-0.23804, 0.944912}, {-0.159643, 0.974836},
      {-0.121048, 0.985454}, {-0.101284, 0.989794}, {-0.0799043, 0.993636},
      {-0.0676623, 0.995432}, {-0.056399, 0.996824}, {-0.0458428, 0.997901},
      {-0.0342557, 0.998827}, {-0.0292321, 0.999146}, {-0.0238468, 0.999431},
      {-0.0191923, 0.999632}, {-0.0141292, 0.9998}, {-0.0111625, 0.999875},
      {-0.00841775, 0.999929}, {-0.00300714, 0.999991}, {0.00190992, 0.999996},
      {0.00713368, 0.999949}, {0.01191, 0.999858}, {0.0171347, 0.999706}, {0.0225892, 0.99949},
      {0.027664, 0.999235}, {0.0367651, 0.998649}, {0.0464879, 0.997841},
      {0.0687153, 0.995289}, {0.0889119, 0.992126}, {0.128404, 0.983648}, {0.165686, 0.972921},
      {0.249743, 0.939534}, {0.327245, 0.898446}, {0.492737, 0.784436}, {0.651864, 0.653818},
      {0.821291, 0.509401}, {0.984352, 0.379482}, {1.15771, 0.261767}, {1.32471, 0.172932},
      {1.48534, 0.110115}, {1.56813, 0.0855182}, {1.65627, 0.0643623}, {1.74637, 0.0473684},
      {1.83113, 0.0349775}, {1.91563, 0.0254858}, {1.99327, 0.0188149}, {2., 0.0183157}}]}},
   {PlotRange → Automatic, AspectRatio → 1/GoldenRatio, DisplayFunction :→ $DisplayFunction,
    ColorOutput → Automatic,
    Axes → Automatic,
    AxesOrigin → Automatic, PlotLabel → None,
    AxesLabel → None, Ticks → Automatic,
    GridLines → None, Prolog → {},
    Epilog → {}, AxesStyle → Automatic,
    Background → Automatic, DefaultColor → Automatic,
    DefaultFont :→ $DefaultFont, RotateLabel → True,
    Frame → False, FrameStyle → Automatic,
    FrameTicks → Automatic, FrameLabel → None,
    PlotRegion → Automatic, ImageSize → Automatic,
    TextStyle :→ $TextStyle, FormatType :→ $FormatType}]
```

Nun schreiben wir mit ">>" den PostScript-Code in ein File namens "postscript_test.ps" u.s.w. ins eigene Unterverzeichnis: (Hinweis: Der Name des Unterverzeichnisses ist hier "c:\work".)

■ Maintenant nous écrivons avec ">>" le code "PostScript" dans un fichier du nom de "postscript_test.ps" etc. dans notre propre sous-dossier: (Indication: Le nom du dossier qu'on a ici est "c:\work".)

In[160]:=

```
expCos >> C:\work\MathematicaData\postscri.ps;
exp1 >> C:\work\MathematicaData\postsc_1.ps;
```

Führe den Befehl aus und kontrolliere rasch im eigenen Verzeichnis, ob das File jetzt da steht! (Vielleicht vergeht etwas Zeit, bis es kommt.) "Display" tut dasselbe. Statt in ein PostScript-File speichern wir nachher den Code mit "Save" in ein *Mathematica*- File, das wir später wieder einlesen und den *Mathematica*-Code verwenden können. (Ausführen und kontrollieren! Probiere im einzulesenden File erst den PostScript-Code ein wenig zu verändern. Hüte Dich dabei vor Werten, die nicht verarbeitet werden können!):

■ Execute l'ordre et contrôle vite dans le propre dossier, si le fichier y est! (Peut-être tu dois attendre un peu qu'il apparaisse.) "Display" fait la même chose. Au lieu de mémoriser le code dans un fichier PostScript, nous le mémorisons après avec "Save" dans un fichier de *Mathematica*, que nous lisons plus tard en utilisant le code de *Mathematica*. (Faire et contrôler! Essaie de changer d'abord un peu le code PostScript dans le fichier à lire. Evite les valeurs qui ne peuvent pas être élaborées!)

In[162]:=
```
Display["C:\\work\\MathematicaData\\postsc_2.ps",expCos];
Display["C:\\work\\MathematicaData\\postsc_3.ps",exp1];
```

In[164]:=
```
Save[
  "C:\\work\MathematicaData\\postscri.ma",expCos];
Save[
  "C:\\work\\MathematicaData\\postsc_1.ma",exp1]
```

```
Syntax::stresc : Unknown string escape \M. Mehr…
```

In[166]:=
```
Save[
  "C:\\work\\MathematicaData\\postscri.nb",expCos];
Save[
  "C:\\work\\MathematicaData\\postsc_1.nb",exp1]
```

Nun löschen wir den PostScript-Code "expCos" u.s.w.:

Maintenant nous effaçons le code PostScript "expCos" etc.:

In[168]:=
```
??exp1
```

```
Global`exp1

exp1 =
 Graphics[{{Line[{{-2., 0.0183157}, {-1.92101, 0.0249653}, {-1.83773, 0.0341409}, {-1.75327,
       0.0462373}, {-1.66076, 0.0634096}, {-1.57856, 0.082756}, {-1.49063, 0.108395},
      {-1.31656, 0.176696}, {-1.14885, 0.267173}, {-0.987513, 0.377124},
      {-0.815873, 0.51394}, {-0.6506, 0.654895}, {-0.491692, 0.785244}, {-0.40333, 0.849867},
      {-0.322484, 0.901229}, {-0.23804, 0.944912}, {-0.159643, 0.974836},
      {-0.121048, 0.985454}, {-0.101284, 0.989794}, {-0.0799043, 0.993636},
      {-0.0676623, 0.995432}, {-0.056399, 0.996824}, {-0.0458428, 0.997901},
      {-0.0342557, 0.998827}, {-0.0292321, 0.999146}, {-0.0238468, 0.999431},
      {-0.0191923, 0.999632}, {-0.0141292, 0.9998}, {-0.0111625, 0.999875},
      {-0.00841775, 0.999929}, {-0.00300714, 0.999991}, {0.00190992, 0.999996},
      {0.00713368, 0.999949}, {0.01191, 0.999858}, {0.0171347, 0.999706}, {0.0225892, 0.99949},
      {0.027664, 0.999235}, {0.0367651, 0.998649}, {0.0464879, 0.997841},
      {0.0687153, 0.995289}, {0.0889119, 0.992126}, {0.128404, 0.983648}, {0.165686, 0.972921},
      {0.249743, 0.939534}, {0.327245, 0.898446}, {0.492737, 0.784436}, {0.651864, 0.653818},
      {0.821291, 0.509401}, {0.984352, 0.379482}, {1.15771, 0.261767}, {1.32471, 0.172932},
      {1.48534, 0.110115}, {1.56813, 0.0855182}, {1.65627, 0.0643623}, {1.74637, 0.0473684},
      {1.83113, 0.0349775}, {1.91563, 0.0254858}, {1.99327, 0.0188149}, {2., 0.0183157}}]}},
   {PlotRange → Automatic, AspectRatio → 1/GoldenRatio, DisplayFunction :→ $DisplayFunction,
    ColorOutput → Automatic,
    Axes → Automatic,
    AxesOrigin → Automatic, PlotLabel → None,
    AxesLabel → None, Ticks → Automatic,
    GridLines → None, Prolog → {},
    Epilog → {}, AxesStyle → Automatic,
    Background → Automatic, DefaultColor → Automatic,
    DefaultFont :→ $DefaultFont, RotateLabel → True,
    Frame → False, FrameStyle → Automatic,
    FrameTicks → Automatic, FrameLabel → None,
    PlotRegion → Automatic, ImageSize → Automatic,
    TextStyle :→ $TextStyle, FormatType :→ $FormatType}]
```

In[169]:=
```
Remove[expCos, exp1]
```

In[170]:=

```
??exp1
```

```
Information::notfound : Symbol exp1 not found. Mehr…
```

In[171]:=

```
??expCos
```

```
Information::notfound : Symbol expCos not found. Mehr…
```

Das File kann wieder eingelesen werden (Achtung Front-End!):

On peut relire le fichier (Attention Front-End!):

In[172]:=

```
??Get
```

```
<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr…

Attributes[Get] = {Protected}

Options[Get] = {CharacterEncoding:→ $CharacterEncoding, Path:→ $Path}
```

In[173]:=

```
?? !!
```

```
n!! gives the double factorial of n. Mehr…

Attributes[Factorial2] = {Listable, NumericFunction, Protected}
```

In[174]:=

```
?? <<
```

```
<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr…

Attributes[Get] = {Protected}

Options[Get] = {CharacterEncoding:→ $CharacterEncoding, Path:→ $Path}
```

In[175]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb",exp1]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.nb is not encoded. Mehr…
```

Out[175]=

```
StandardForm
```

In[176]:=

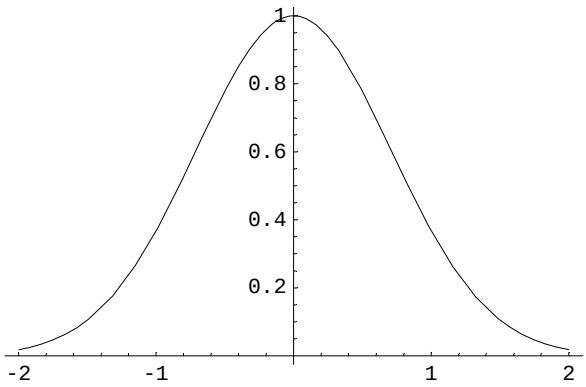
```
Get["C:\work\MathematicaData\postsc_1.nb"]; exp1
```

Out[176]=

```
- Graphics -
```

In[177]:=

```
Get["C:\work\MathematicaData\postsc_1.nb"]; Show[exp1]
```



Out[177]=

```
- Graphics -
```

In[178]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb", exp1]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.nb is not encoded. Mehr…
```

Out[178]=

```
StandardForm
```

In[179]:=

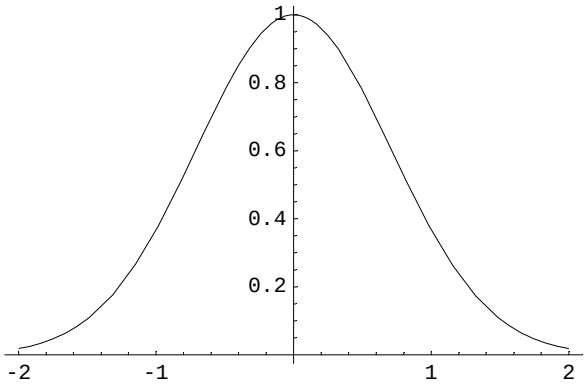
```
?exp1
```

```
Global`exp1

exp1 =
 Graphics[{{Line[{{-2., 0.0183157}, {-1.92101, 0.0249653}, {-1.83773, 0.0341409}, {-1.75327,
       0.0462373}, {-1.66076, 0.0634096}, {-1.57856, 0.082756}, {-1.49063, 0.108395},
      {-1.31656, 0.176696}, {-1.14885, 0.267173}, {-0.987513, 0.377124},
      {-0.815873, 0.51394}, {-0.6506, 0.654895}, {-0.491692, 0.785244}, {-0.40333, 0.849867},
      {-0.322484, 0.901229}, {-0.23804, 0.944912}, {-0.159643, 0.974836},
      {-0.121048, 0.985454}, {-0.101284, 0.989794}, {-0.0799043, 0.993636},
      {-0.0676623, 0.995432}, {-0.056399, 0.996824}, {-0.0458428, 0.997901},
      {-0.0342557, 0.998827}, {-0.0292321, 0.999146}, {-0.0238468, 0.999431},
      {-0.0191923, 0.999632}, {-0.0141292, 0.9998}, {-0.0111625, 0.999875},
      {-0.00841775, 0.999929}, {-0.00300714, 0.999991}, {0.00190992, 0.999996},
      {0.00713368, 0.999949}, {0.01191, 0.999858}, {0.0171347, 0.999706}, {0.0225892, 0.99949},
      {0.027664, 0.999235}, {0.0367651, 0.998649}, {0.0464879, 0.997841},
      {0.0687153, 0.995289}, {0.0889119, 0.992126}, {0.128404, 0.983648}, {0.165686, 0.972921},
      {0.249743, 0.939534}, {0.327245, 0.898446}, {0.492737, 0.784436}, {0.651864, 0.653818},
      {0.821291, 0.509401}, {0.984352, 0.379482}, {1.15771, 0.261767}, {1.32471, 0.172932},
      {1.48534, 0.110115}, {1.56813, 0.0855182}, {1.65627, 0.0643623}, {1.74637, 0.0473684},
      {1.83113, 0.0349775}, {1.91563, 0.0254858}, {1.99327, 0.0188149}, {2., 0.0183157}}]}},
   {PlotRange → Automatic, AspectRatio → 1/GoldenRatio, DisplayFunction :→ $DisplayFunction,
    ColorOutput → Automatic,
    Axes → Automatic,
    AxesOrigin → Automatic, PlotLabel → None,
    AxesLabel → None, Ticks → Automatic,
    GridLines → None, Prolog → {},
    Epilog → {}, AxesStyle → Automatic,
    Background → Automatic, DefaultColor → Automatic,
    DefaultFont :→ $DefaultFont, RotateLabel → True,
    Frame → False, FrameStyle → Automatic,
    FrameTicks → Automatic, FrameLabel → None,
    PlotRegion → Automatic, ImageSize → Automatic,
    TextStyle :→ $TextStyle, FormatType :→ $FormatType}]
```

In[180]:=

```
Show[Graphics[expCos]];
Show[Graphics[exp1]];
```

```
Graphics::gprim : expCos was encountered where a Graphics primitive or directive was expected. Mehr…
```



Der Befehl ist jetzt wieder da! Nun schauen wir uns den Inhalt des Files "postscri.ps" auf dem Schirm an.

Voici de nouveau l'ordre! Maintenant nous regardons le contenu du fichier "postscri.ps" sur l'écran.

In[182]:=

```
!!"C:\\work\\MathematicaData\\postscri.ps"
```

```
General::noopen : Cannot open "C:\\work\\MathematicaData\\postscri.ps". Mehr…
```

In[183]:=

```
!!C:\\work\\MathematicaData\\postscri.ps
```

```
Graphics[{{Line[{{-1.9999998333333333, -0.0122153593511319},
     {-1.9623003192812842, 0.0005082587300581328}, {-1.9210073169479194,
      0.018752002927052425}, {-1.910299579465502, 0.022740269212148626},
     {-1.9001774790199295, 0.025790484842825256}, {-1.8945829997783903,
      0.027103353077953987}, {-1.8916352820184055, 0.02767476304778044},
     {-1.8885103672286825, 0.028184000386560142}, {-1.8829909524966886,
      0.028827453836099467}, {-1.8778952782513714, 0.029115238245558964},
     {-1.8731413317543735, 0.029105227315202174}, {-1.8680274614917154,
      0.028783053185285283}, {-1.863199958792915, 0.028174933726430717},
     {-1.8587831546785645, 0.027355730435410536}, {-1.8536232573178666,
      0.026078553418176047}, {-1.8479909934251308, 0.024292164406973862},
     {-1.8377320337083367, 0.020012423042739088}, {-1.8257366220651903,
      0.013444784972129038}, {-1.8144362944250474, 0.005937356649683411},
     {-1.7933141736895866, -0.010391844332965274}, {-1.768652621938747,
      -0.030023715053337733}, {-1.7578358910677812, -0.03757544820569005},
     {-1.745962817885428, -0.04436555940559883}, {-1.7409954301270045,
      -0.04661226368848812}, {-1.7356703543983865, -0.04856988304671834},
     {-1.7306441655063993, -0.04995350510306155}, {-1.726061421963292,
      -0.05079554189295374}, {-1.7208899413754872, -0.05123896481969352},
     {-1.7179679228759093, -0.05124293728587858}, {-1.7152644782688042,
      -0.05108393336125241}, {-1.7103050520507237, -0.0503777772427448},
```

```
{-1.705036285194253, -0.04903000884182273}, {-1.6998130954536856,
 -0.04707998343149028}, {-1.6941824661510267, -0.04429399736242225},
{-1.6839927695262276, -0.03748701340302931}, {-1.6718751692383962,
 -0.026621373541510396}, {-1.660764800562505, -0.01436785949148811},
{-1.6223373700539767, 0.03698336903382436}, {-1.6122046687997946,
 0.05026582756968046}, {-1.6012204218638049, 0.0631825070687631},
{-1.5915368555668616, 0.07268572843936418}, {-1.58624016644988,
 0.07694697859595899}, {-1.5812846856736067, 0.08024862424865138},
{-1.5754383042802425, 0.0832155003092454}, {-1.572337956846547,
 0.08435492782676386}, {-1.5690072708401817, 0.08522809918894149},
{-1.5659728443536363, 0.08569752092152322}, {-1.5630921326197909,
 0.08584815274454088}, {-1.5576873525951036, 0.08533590009119145},
{-1.5523642925764947, 0.08379516017296415}, {-1.5474289001941306,
 0.08143362486126446}, {-1.5420125470492216, 0.07780448278522988},
{-1.536246079580289, 0.07275700026813073}, {-1.5249155505844803,
 0.059402686355185516}, {-1.5142556242299647, 0.04300384325934713},
{-1.4943432743542384, 0.004472710594773914}, {-1.4532279345574657,
 -0.08515390290044901}, {-1.4432335331173123, -0.10348325385511208},
{-1.432533564692643, -0.11946821817256431}, {-1.42638894081693,
 -0.12655827281836432}, {-1.4207938199321637, -0.13150559927652575},
{-1.41809797650952, -0.13334094680801603}, {-1.4151693384029775,
 -0.13491283836164045}, {-1.4098481479164913, -0.13660415416094238},
{-1.4050416145772238, -0.13679532253598312}, {-1.4005351214442552,
 -0.1357881187725129}, {-1.3955913202634447, -0.13333739136336295},
{-1.3902688897887092, -0.12910990712973433}, {-1.3849829640264526,
 -0.1232818098314748}, {-1.3794251016434265, -0.11542481340098515},
{-1.36949576036685, -0.09713667609456161}, {-1.3516769177229944,
 -0.05213560711740385}, {-1.3322109850418382, 0.010049649308713196},
{-1.311814606942803, 0.0805873886472376}, {-1.301081564243093,
 0.11597117145006212}, {-1.2897332454943118, 0.14947293224243818},
{-1.2797977572941663, 0.17390652324509817}, {-1.2745323012358132,
 0.1845386245184353}, {-1.2688830972260063, 0.19390980702104588},
{-1.2636935060450525, 0.20050558155842912}, {-1.2589579198010443,
 0.20473215653759175}, {-1.2541975830843266, 0.20717275659823553},
{-1.249684335929672, 0.20775197540724452}, {-1.247041828138711,
 0.20728903288692044}, {-1.2441226325567252, 0.20607895510875865},
{-1.2410738117547113, 0.20402551530318114}, {-1.2382092114607313,
 0.20135719496762308}, {-1.2330981005800692, 0.19481588054246313},
{-1.2275838049165742, 0.18521572373110567}, {-1.2214743617585684,
 0.17155776588451163}, {-1.2150369184940277, 0.1538563691722545},
{-1.203523571823205, 0.11438509646840048}, {-1.1616357322503013,
 -0.0838667012692308}, {-1.1410574771688768, -0.18355519085192498},
{-1.1309454699394663, -0.2252446687030288}, {-1.1216144865495554,
 -0.2570070238434482}, {-1.116498155471344, -0.2711457519497164},
{-1.1108655908628202, -0.28370556740513797}, {-1.1081370509617479,
 -0.2885858899753463}, {-1.1052034083602063, -0.29291770319564026},
{-1.1021050224055187, -0.2964329900999418}, {-1.0992038714781798,
 -0.29871236701229176}, {-1.0937451922278991, -0.3002768983062175},
{-1.0885588150162093, -0.29839061424267377}, {-1.0839033652069439,
 -0.2938521069914662}, {-1.0788242866321036, -0.28581302032434724},
{-1.0735924733113682, -0.27417965132220024}, {-1.068666299969677,
 -0.26016454226758307}, {-1.0594607465548975, -0.22631156310060757},
{-1.0388287578280821, -0.11852835113159345}, {-1.0017158165421325,
 0.1380386876955374}, {-0.9804818490676098, 0.27714025981739737},
{-0.9701152472705907, 0.3320839289435045}, {-0.9606680178167547,
 0.37136370846023714}, {-0.9555429871463601, 0.3876711007347637},
{-0.9506903685808147, 0.39957150651606144}, {-0.94556841761877,
 0.4081946077693798}, {-0.9426751414885124, 0.411212892719486},
{-0.9399996360794677, 0.41278513550084933}, {-0.934770753521844,
 0.4124157216304582}, {-0.9317946152243468, 0.41014558532629825},
{-0.929001985645631, 0.4066458604324708}, {-0.9260688269140263,
 0.40153938048657495}, {-0.9233808299909719, 0.39557434627057825},
{-0.9173242395579699, 0.3776634222547061}, {-0.9127976387820719,
 0.3602999772862939}, {-0.9079642254182297, 0.3381144016387083},
{-0.8992602803223766, 0.28915729392454115}, {-0.8795104906864795,
 0.1413842470955593}, {-0.8397190375423287, -0.23005651682605105},
{-0.8187971422559015, -0.4015572594838865}, {-0.8082518687084622,
 -0.4669266707753546}, {-0.8022604864947778, -0.49579074972261117},
{-0.7966578833395991, -0.5167287009438404}, {-0.7935177520634596,
 -0.525757956681514}, {-0.7905381888494231, -0.5324618998910939},
{-0.7878635232886223, -0.5368991395999997}, {-0.7849493165407382,
 -0.5400002249881906}, {-0.7817880476133191, -0.541289292474234},
{-0.7784241259699886, -0.5402608150821243}, {-0.7752499569500357,
```

```
 -0.5370024066246588}, {-0.7723771954811092, -0.5321307725565154},
{-0.7668323840651169, -0.5175679461624914}, {-0.7637114688582315,
 -0.5064022525594425}, {-0.7608453028627378, -0.4942884130331306},
{-0.750001386238209, -0.43286649730882254}, {-0.7398238797968588,
 -0.35436495342595553}, {-0.7303395497421928, -0.2654707763053889},
{-0.7087968591039506, -0.023453258580960444}, {-0.6862069434280346,
 0.2506456308828193}, {-0.6653448287717248, 0.47409810939227504},
{-0.6550941824327547, 0.5599189734032894}, {-0.6498581570937355,
 0.5956349265395152}, {-0.6442239577120495, 0.6271947855867068},
{-0.6412828970965597, 0.6406690475025774}, {-0.6385912090240337,
 0.651124447712424}, {-0.633312731602339, 0.6662557045388081},
{-0.6303669227819514, 0.6715264501165197}, {-0.6271504370779537,
 0.6746302040365503}, {-0.6241122714698554, 0.6749890352056996},
{-0.6212907785054942, 0.673065392685534}, {-0.615848103126219,
 0.6631864529584339}, {-0.610719214050287, 0.6464572445824115},
{-0.6061140961990733, 0.6253747705571909}, {-0.6011310493901006,
 0.5962432752291937}, {-0.58993621358246, 0.5080588617703872},
{-0.5796332258514509, 0.40179883304703257}, {-0.5419911185092119,
 -0.11563789469737708}, {-0.5212657131179157, -0.41140376620744473},
{-0.5016259570844893, -0.6383082637499763}, {-0.49139111028154175,
 -0.7225368945842895}, {-0.4856072258904732, -0.7575352067107914},
{-0.4802543250749207, -0.781149556618428}, {-0.4757632963228615,
 -0.7941758653567549}, {-0.47080452614718676, -0.8011599707142121},
{-0.46562566801066907, -0.800017069509857}, {-0.460762181661922,
 -0.7910321667400466}, {-0.45609740432417917, -0.7752294176464226},
{-0.4511846581763733, -0.7510703522206259}, {-0.4469901738226814,
 -0.7244650488260634}, {-0.4424004989015632, -0.6892291015728184},
{-0.43214488080401514, -0.5887125083849086}, {-0.4226003398105357,
 -0.4709297597147562}, {-0.37797369624595195, 0.25162584454478143},
{-0.35605021441036094, 0.5894204860628905}, {-0.34642322744337106,
 0.7085825762585028}, {-0.34131725061385637, 0.7619387936024965},
{-0.3358302221056187, 0.8107423480186627}, {-0.3304297524471636,
 0.8495136641902389}, {-0.3254936995516747, 0.8764609305371227},
{-0.32053192780411477, 0.8950417017536064}, {-0.3158200401288461,
 0.9045714043821929}, {-0.3105605721446212, 0.9057062735594216},
{-0.3056826236909325, 0.8977335870214513}, {-0.30032936887031186,
 0.8790188868261573}, {-0.2973379125422991, 0.8640692854201357},
{-0.29462830641223864, 0.847791590281556}, {-0.2835658740663681,
 0.7553319633410993}, {-0.27733670569774405, 0.6860096772305007},
{-0.2716113406541768, 0.6123778152682854}, {-0.24994991234572664,
 0.2655800740728669}, {-0.21225773815848598, -0.43057614557528495},
{-0.2014045185385447, -0.607001214670649}, {-0.1911333494049853,
 -0.7490273389929445}, {-0.18191240706908424, -0.8505632497565038},
{-0.1767416129962974, -0.8952652342589389}, {-0.1718920901267761,
 -0.9285918556554922}, {-0.16697957316058187, -0.9535023649777139},
{-0.161562565091676, -0.9703221049620906}, {-0.15628418144238038,
 -0.9757476325207776}, {-0.15331730465130441, -0.9740039752842746},
{-0.15059977644037958, -0.9693770667343177}, {-0.1460165797564408,
 -0.9550405828469034}, {-0.1410081756896768, -0.9301062215571163},
{-0.13577962138178978, -0.893991278427419}, {-0.13091999659907888,
 -0.8515076827970871}, {-0.12123049018585531, -0.7427944269588428},
{-0.11243516664911235, -0.6192856959607796}, {-0.09270736290683174,
 -0.2771819714654778}, {-0.07074128821999481, 0.15456352776450047},
{-0.047508779695364454, 0.5802291446050808}, {-0.03794148475458846,
 0.7245979117086273}, {-0.027673730365563417, 0.8500519974179697},
{-0.021783670915650262, 0.9061559525105312}, {-0.01640722304970641,
 0.9463871901825228}, {-0.01100269176883796, 0.9757675666255191},
{-0.005877094856298342, 0.9930655997505854}, {-0.0012603543457706003,
 0.9996807302191512}, {0.003841377090691665, 0.9970355032139022},
{0.00916971378005925, 0.9831476806414581}, {0.01417765010626921,
 0.9598745223762447}, {0.018835778805047276, 0.9295480348256523},
{0.02311152799549632, 0.8945820542534499}, {0.03275314394312561,
 0.792159363745938}, {0.05323437432389867, 0.4834057057361746},
{0.07482189650603684, 0.07387515592061891}, {0.09497769065648072,
 -0.3199678960821519}, {0.1165347071741706, -0.6795558291669739},
{0.12290188181040415, -0.7637091472760054}, {0.1289196612394384,
 -0.8316141273358323}, {0.13477808256826967, -0.8859263593475684},
{0.14018569726502234, -0.9251000595565696}, {0.14531416564759841,
 -0.9521232602335068}, {0.15074987886152633, -0.9697081023973886},
{0.15620612109614104, -0.9757460076225583}, {0.1593087430762231,
 -0.9739713489533526}, {0.16221476370581453, -0.9688971227041098},
{0.16745001383881453, -0.9515105805200534}, {0.1703767756527555,
 -0.9372399699444143}, {0.17302994423284257, -0.9215400634230125},
```

```
{0.18306876655623647, -0.8393213096217108}, {0.18869925168730767,
 -0.7783995770475729}, {0.194788749285674, -0.7016958861226725},
{0.2058274871176703, -0.5379247908600145}, {0.2472775023612705,
 0.21734939392563712}, {0.2669876956883489, 0.5466263082463674},
{0.27775578978345566, 0.6910376340383966}, {0.28780683415627745,
 0.7955882301276718}, {0.29221298913057425, 0.8311225084438845},
{0.2963862394019538, 0.858646541028629}, {0.30119657032466773,
 0.8827537327474217}, {0.30558165684475946, 0.8974768328574726},
{0.3104357848654642, 0.905610656162231}, {0.31492594941144675,
 0.9054747211033035}, {0.319854075345852, 0.8969036163134283},
{0.3250885838720468, 0.8783015513818669}, {0.32986816478766834,
 0.8529956085057786}, {0.33509542949091004, 0.8165707134217922},
{0.34562891803050866, 0.717360860055138}, {0.3644582901051175,
 0.4686814149708279}, {0.40689090862089167, -0.2373117913483952},
{0.426125604486471, -0.5169094904177725}, {0.43689074850830234,
 -0.6388263696340332}, {0.44197946198477883, -0.6856842565689083},
{0.44667041262385826, -0.7222151789885232}, {0.4510207771170432,
 -0.7501332725563638}, {0.4558052679116797, -0.7740071803631008},
{0.4606148703666571, -0.7906404333586745}, {0.4657059069711432,
 -0.8001009858284969}, {0.4701818412566341, -0.8014806461151873},
{0.47498938854669887, -0.7957813656725395}, {0.4795274484879301,
 -0.7836835448282652}, {0.48367316714241243, -0.7670632421985295},
{0.48910652724614345, -0.7375047588806098}, {0.49421732943212265,
 -0.7020199626620278}, {0.5037632278937437, -0.6174246056732089},
{0.5251478059134861, -0.35894759636214624}, {0.5635945404101352,
 0.19857175803531235}, {0.5742159503756941, 0.33764078266505365},
{0.5854201872696259, 0.4642184562076209}, {0.59498618095222,
 0.5515984138888202}, {0.6003741082859045, 0.591255829543805},
{0.60547052133523, 0.6219780313442358}, {0.6083075285152632,
 0.6361250407609035}, {0.6113889278134459, 0.6490479456023531},
{0.6167770249727986, 0.6654477396485784}, {0.6198296973349456,
 0.6712117051612357}, {0.623067888318758, 0.6745310254743232},
{0.6258324575645496, 0.6750943051423131}, {0.6288793992299165,
 0.6733081258630773}, {0.634303848090634, 0.663966651997967},
{0.6373520249762497, 0.6553223823800816}, {0.6401426151420501,
 0.6453200158057888}, {0.645553737508074, 0.6204158430707607},
{0.650700713891915, 0.5902905186685075}, {0.6939669926869411,
 0.15752673366231526}, {0.7152020721601373, -0.09959206992178356},
{0.7347515180442568, -0.308535470979283}, {0.7439550305288842,
 -0.388503320333675}, {0.7525526748937331, -0.44947412460545044},
{0.761520709343437, -0.4973020699793587}, {0.7662730077468173,
 -0.5157230754805726}, {0.7713706752869288, -0.5299915837104358},
{0.7764655824982817, -0.5385135508046426}, {0.7820111997625422,
 -0.5412697136677055}, {0.7872487221463473, -0.5377047719381315},
{0.7920974807320128, -0.5291826799478381}, {0.7964529521305389,
 -0.5173784076900075}, {0.8012635440655589, -0.49995684200916335},
{0.8110073898497123, -0.45154940718504377}, {0.8216866556673058,
 -0.38077890138613835}, {0.8314843412770514, -0.30276272782875757},
{0.8535723501576126, -0.09933208457858644}, {0.8742799127520443,
 0.09562488950229762}, {0.8935577024893068, 0.25127274525933935},
{0.9034528690271358, 0.3141373454867022}, {0.9140306207806459,
 0.3653613330128868}, {0.9200662975061342, 0.38653503814918144},
{0.9230318897571291, 0.39471010144585883}, {0.9257679017110405,
 0.40093262096077703}, {0.9308868297973646, 0.40915363270998506},
{0.9364394027231373, 0.41303250288556703}, {0.9418327858488762,
 0.4118351189538449}, {0.9468818702625253, 0.4063788296314459},
{0.9524211948594483, 0.3957343604592793}, {0.9555317848759625,
 0.3877026178486205}, {0.9583757626343072, 0.37911768876508517},
{0.9702935393392168, 0.33123670459888427}, {0.9816146322866746,
 0.2704990999329876}, {1.0219337574845162, -0.006447852680366173},
{1.0440580327404785, -0.1494976259339504}, {1.0549091834196738,
 -0.2060834353790531}, {1.0648858169822157, -0.2474436640448388},
{1.0743602626221473, -0.27609822989107596}, {1.0791303978563087,
 -0.2863886585377893}, {1.0843627332104027, -0.2944202724381302},
{1.0896214981734995, -0.29904808873811173}, {1.0925692255875215,
 -0.30013993168987985}, {1.0952550142087834, -0.30020482778189655},
{1.1004751463501758, -0.2978354880883429}, {1.1054168621295604,
 -0.2926351096973051}, {1.1098516796268472, -0.28561328923965135},
{1.1145598420490528, -0.2758356038124899}, {1.1230130242444196,
 -0.25271804093753775}, {1.142140568536079, -0.17872230408925346},
{1.1815509438027387, 0.01708703617472436}, {1.2022395560575547,
 0.10941576698635769}, {1.2126386342258384, 0.1464307316264005},
{1.218596390232883, 0.16405381673229888}, {1.2240060265504031,
```

```
          0.17759683511463492}, {1.2288279588669109, 0.18761035167194878},
         {1.2340514887294487, 0.19620888071038206}, {1.2395697767980838,
          0.20271393807973756}, {1.2447757706178375, 0.20641373354368245},
         {1.2496593073706657, 0.20775038993425438}, {1.2548260055427813,
          0.20695695951905246}, {1.2592371268498836, 0.20453200348774384},
         {1.26408599477186, 0.20007719367288554}, {1.268870350843575,
          0.19392842885484418}, {1.2732388716663274, 0.18687694203521504},
         {1.2830649217771246, 0.1664640675381528}, {1.3008089017854554,
          0.11683131393049553}, {1.3395619533221224, -0.014559309921745341},
         {1.3497807250584448, -0.04656505959061242}, {1.360812454108068,
          -0.07699189854131996}, {1.3705376522004298, -0.09930216737751288},
         {1.3807913421137497, -0.11751844389690592}, {1.3866981275031691,
          -0.1253500384725379}, {1.389824237087754, -0.12868205488181444},
         {1.393168712185386, -0.13161783385082187}, {1.3962490792977973,
          -0.1337452708365822}, {1.399052023099811, -0.1352016057928178},
         {1.4045335760246098, -0.13673969740956485}, {1.4098285490083289,
          -0.13660754156569366}, {1.4128172144948126, -0.13584859261362373},
         {1.4155767598509525, -0.13472094801234416}, {1.4206443460942033,
          -0.1316169197489221}, {1.4260171995048834, -0.12693301271192878},
         {1.436311228356535, -0.1143182202119677}, {1.4459559617815967,
          -0.09877639029056667}, {1.4679201985139687, -0.054223359972494584},
         {1.4887579343823243, -0.00762072873950793}, {1.5115059793294574,
          0.0382345442533714}, {1.5219052911983821, 0.05512766037480968},
         {1.5329927229796834, 0.06937991404491133}, {1.5383799169472308,
          0.07476578756256917}, {1.5434555481172565, 0.07887721208535046},
         {1.5480128760885281, 0.08176009227545963}, {1.5529432631496285,
          0.08401317903186631}, {1.5577999319927756, 0.08535730267782553},
         {1.562334603946491, 0.08583929036751099}, {1.5673100084113765,
          0.08552953552719611}, {1.5726443116134108, 0.08425618533669176},
         {1.5753219956329856, 0.08326380557115205}, {1.5782350884201926,
          0.08192639160190762}, {1.5835319886303225, 0.07883722949145197},
         {1.5934602592130633, 0.07096490215862193}, {1.6128244376616974,
          0.04948420103577662}, {1.6307422583466153, 0.02543404632466773},
         {1.6491132192575328, 0.00029382516263982446}, {1.6684570224553763,
          -0.023063944014359752}, {1.6785864950413831, -0.032992068553395676},
         {1.6896682018240128, -0.04155430222370065}, {1.6945391991512138,
          -0.044491418002653244}, {1.6997435873078401, -0.047049906538513514},
         {1.7046552079069046, -0.04890844329652339}, {1.7091185902618236,
          -0.05012830831614879}, {1.711901482632756, -0.05066418084712502},
         {1.7148481318213324, -0.05104538406562282}, {1.7201314630241844,
          -0.051257364803291655}, {1.7230685323472925, -0.051119227645411644},
         {1.725816724432411, -0.05082882337742177}, {1.7288318026150509,
          -0.050335628120028626}, {1.7320712690896647, -0.049608398055116146},
         {1.7378798445858235, -0.047816864809224556}, {1.743394830049025,
          -0.04557543452017847 5}, {1.7537426239565226, -0.04012069650849713},
         {1.7737875026298968, -0.026116835698228463}, {1.7956411814708972,
          -0.00850661872577543}, {1.8180209034331518, 0.008440018725300728},
         {1.8289944674381577, 0.01538353194694639}, {1.8390785712365771,
          0.020647516279422223}, {1.8482787300585155, 0.024393268036740877},
         {1.8529669537056004, 0.02589139337639477}, {1.8579867443579592,
          0.02718112978745319}, {1.8630295576621545, 0.028148004455478503},
         {1.8677318350704086, 0.02875436483964524}, {1.8706049904237505,
          0.028986380194225708}, {1.873323538248825, 0.0291106936040121},
         {1.8784157905220895, 0.02909979399104083}, {1.8834819593167174,
          0.028783911606270977}, {1.8863644484863291, 0.028473886915831758},
         {1.889067323789397, 0.0281007099146851}, {1.8945097503439552,
          0.027118590924287478}, {1.9003696902891154, 0.02574033225287379 3},
         {1.9109593009628465, 0.022514858084424577}, {1.920657051873991,
          0.018892401701557857}, {1.9665514341784773, -0.0012774939524654253},
         {1.9999998333333333, -0.0122153593511319}}]}},
       {PlotRange -> Automatic, AspectRatio -> GoldenRatio^(-1),
        DisplayFunction :> $DisplayFunction, ColorOutput -> Automatic,
        Axes -> Automatic, AxesOrigin -> Automatic, PlotLabel -> None,
        AxesLabel -> None, Ticks -> Automatic, GridLines -> None, Prolog -> {},
        Epilog -> {}, AxesStyle -> Automatic, Background -> Automatic,
        DefaultColor -> Automatic, DefaultFont :> $DefaultFont,
        RotateLabel -> True, Frame -> False, FrameStyle -> Automatic,
        FrameTicks -> Automatic, FrameLabel -> None, PlotRegion -> Automatic,
        ImageSize -> Automatic, TextStyle :> $TextStyle, FormatType :> $FormatType}]
```

In[184]:=

```
Get["C:\\work\\MathematicaData\\postscri.ps"]
```

Out[184]=

- Graphics **-**

In[185]:=

```
Get["C:\\work\\MathematicaData\\postscri.ps"] // FullForm
```

Out[185]//FullForm=

```
Graphics[List[List[Line[List[List[-1.9999998333333333`, -0.0122153593511319`],
     List[-1.9623003192812842`, 0.0005082587300581328`],
     List[-1.9210073169479194`, 0.018752002927052425`],
     List[-1.910299579465502`, 0.022740269212148626`], List[-1.9001774790199295`,
      0.025790484842825256`], List[-1.8945829997783903`, 0.027103353077953987`],
     List[-1.8916352820184055`, 0.02767476304778044`],
     List[-1.8885103672286825`, 0.028184000386560142`],
     List[-1.8829909524966886`, 0.028827453836099467`],
     List[-1.8778952782513714`, 0.029115238245558964`],
     List[-1.8731413317543735`, 0.029105227315202174`],
     List[-1.8680274614917154`, 0.028783053185285283`],
     List[-1.863199958792915`, 0.028174933726430717`],
     List[-1.8587831546785645`, 0.027355730435410536`],
     List[-1.8536232573178666`, 0.026078553418176047`],
     List[-1.8479909934251308`, 0.024292164406973862`],
     List[-1.8377320337083367`, 0.020012423042739088`],
     List[-1.8257366220651903`, 0.013444784972129038`],
     List[-1.8144362944250474`, 0.005937356649683411`],
     List[-1.7933141736895866`, -0.010391844332965274`],
     List[-1.768652621938747`, -0.030023715053337733`],
     List[-1.7578358910677812`, -0.03757544820569005`],
     List[-1.745962817885428`, -0.04436555940559883`],
     List[-1.7409954301270045`, -0.04661226368848812`],
     List[-1.7356703543983865`, -0.04856988304671834`],
     List[-1.7306441655063993`, -0.04995350510306155`],
     List[-1.726061421963292`, -0.05079554189295374`],
     List[-1.7208899413754872`, -0.05123896481969352`],
     List[-1.7179679228759093`, -0.05124293728587858`],
     List[-1.7152644782688042`, -0.05108393336125241`],
     List[-1.7103050520507237`, -0.0503777772427448`],
     List[-1.705036285194253`, -0.04903000884182273`],
     List[-1.6998130954536856`, -0.04707998343149028`],
     List[-1.6941824661510267`, -0.04429399736242225`],
     List[-1.6839927695262276`, -0.03748701340302931`],
     List[-1.6718751692383962`, -0.026621373541510396`],
     List[-1.660764800562505`, -0.01436785949148811`],
     List[-1.6223373700539767`, 0.03698336903382436`],
     List[-1.6122046687997946`, 0.05026582756968046`],
     List[-1.6012204218638049`, 0.0631825070687631`],
     List[-1.5915368555668616`, 0.07268572843936418`],
     List[-1.58624016644988`, 0.07694697859595899`],
     List[-1.5812846856736067`, 0.08024862424865138`],
     List[-1.5754383042802425`, 0.0832155003092454`],
     List[-1.572337956846547`, 0.08435492782676386`],
     List[-1.5690072708401817`, 0.08522809918894149`],
     List[-1.5659728443536363`, 0.08569752092152322`],
     List[-1.5630921326197909`, 0.08584815274454088`],
     List[-1.5576873525951036`, 0.08533590009119145`],
     List[-1.5523642925764947`, 0.08379516017296415`],
     List[-1.5474289001941306`, 0.08143362486126446`],
     List[-1.5420125470492216`, 0.07780448278522988`],
```

```
List[-1.536246079580289`, 0.07275700026813073`],
List[-1.5249155505844803`, 0.059402686355185516`],
List[-1.5142556242299647`, 0.04300384325934713`],
List[-1.4943432743542384`, 0.004472710594773914`],
List[-1.4532279345574657`, -0.08515390290044901`],
List[-1.4432335331173123`, -0.10348325385511208`],
List[-1.432533564692643`, -0.11946821817256431`],
List[-1.42638894081693`, -0.12655827281836432`],
List[-1.4207938199321637`, -0.13150559927652575`],
List[-1.41809797650952`, -0.13334094680801603`],
List[-1.4151693384029775`, -0.13491283836164045`],
List[-1.4098481479164913`, -0.13660415416094238`],
List[-1.4050416145772238`, -0.13679532253598312`],
List[-1.4005351214442552`, -0.1357881187725129`],
List[-1.3955913202634447`, -0.13333739136336295`],
List[-1.3902688897887092`, -0.12910990712973433`],
List[-1.3849829640264526`, -0.1232818098314748`],
List[-1.3794251016434265`, -0.11542481340098515`],
List[-1.36949576036685`, -0.09713667609456161`],
List[-1.3516769177229944`, -0.05213560711740385`],
List[-1.3322109850418382`, 0.010049649308713196`],
List[-1.311814606942803`, 0.0805873886472376`],
List[-1.301081564243093`, 0.11597117145006212`],
List[-1.2897332454943118`, 0.14947293224243818`],
List[-1.2797977572941663`, 0.17390652324509817`],
List[-1.2745323012358132`, 0.1845386245184353`],
List[-1.2688830972260063`, 0.19390980702104588`],
List[-1.2636935060450525`, 0.20050558155842912`],
List[-1.2589579198010443`, 0.20473215653759175`],
List[-1.2541975830843266`, 0.20717275659823553`],
List[-1.249684335929672`, 0.20775197540724452`],
List[-1.247041828138711`, 0.20728903288692044`],
List[-1.2441226325567252`, 0.20607895510875865`],
List[-1.2410738117547113`, 0.20402551530318114`],
List[-1.2382092114607313`, 0.20135719496762308`],
List[-1.2330981005800692`, 0.19481588054246313`],
List[-1.2275838049165742`, 0.18521572373110567`],
List[-1.2214743617585684`, 0.17155776588451163`],
List[-1.2150369184940277`, 0.1538563691722545`],
List[-1.203523571823205`, 0.11438509646840048`],
List[-1.1616357322503013`, -0.0838667012692308`],
List[-1.1410574771688768`, -0.18355519085192498`],
List[-1.1309454699394663`, -0.2252446687030288`],
List[-1.1216144865495554`, -0.2570070238434482`],
List[-1.116498155471344`, -0.2711457519497164`],
List[-1.1108655908628202`, -0.28370556740513797`],
List[-1.1081370509617479`, -0.2885858899753463`],
List[-1.1052034083602063`, -0.29291770319564026`],
List[-1.1021050224055187`, -0.2964329900999418`],
List[-1.0992038714781798`, -0.29871236701229176`],
List[-1.0937451922278991`, -0.3002768983062175`],
List[-1.0885588150162093`, -0.29839061424267377`],
List[-1.0839033652069439`, -0.2938521069914662`],
List[-1.0788242866321036`, -0.28581302032434724`],
List[-1.0735924733113682`, -0.27417965132220024`],
List[-1.068666299969677`, -0.26016454226758307`],
List[-1.0594607465548975`, -0.22631156310060757`],
List[-1.0388287578280821`, -0.11852835113159345`],
```

```
List[-1.0017158165421325`, 0.1380386876955374`],
List[-0.9804818490676098`, 0.27714025981739737`],
List[-0.9701152472705907`, 0.3320839289435045`],
List[-0.9606680178167547`, 0.37136370846023714`],
List[-0.9555429871463601`, 0.3876711007347637`],
List[-0.9506903685808147`, 0.39957150651606144`],
List[-0.94556841761877`, 0.4081946077693798`],
List[-0.9426751414885124`, 0.411212892719486`],
List[-0.9399996360794677`, 0.41278513550084933`],
List[-0.934770753521844`, 0.4124157216304582`],
List[-0.9317946152243468`, 0.41014558532629825`],
List[-0.929001985645631`, 0.4066458604324708`],
List[-0.9260688269140263`, 0.40153938048657495`],
List[-0.9233808299909719`, 0.39557434627057825`],
List[-0.9173242395579699`, 0.3776634222547061`],
List[-0.9127976387820719`, 0.3602999772862939`],
List[-0.9079642254182297`, 0.3381144016387083`],
List[-0.8992602803223766`, 0.28915729392454115`],
List[-0.8795104906864795`, 0.1413842470955593`],
List[-0.8397190375423287`, -0.23005651682605105`],
List[-0.8187971422559015`, -0.4015572594838865`],
List[-0.8082518687084622`, -0.4669266707753546`],
List[-0.8022604864947778`, -0.49579074972261117`],
List[-0.7966578833395991`, -0.5167287009438404`],
List[-0.7935177520634596`, -0.525757956681514`],
List[-0.7905381888494231`, -0.5324618998910939`],
List[-0.7878635232886223`, -0.5368991395999997`],
List[-0.7849493165407382`, -0.5400002249881906`],
List[-0.7817880476133191`, -0.541289292474234`],
List[-0.7784241259699886`, -0.5402608150821243`],
List[-0.7752499569500357`, -0.5370024066246588`],
List[-0.7723771954811092`, -0.5321307725565154`],
List[-0.7668323840651169`, -0.5175679461624914`],
List[-0.7637114688582315`, -0.5064022525594425`],
List[-0.7608453028627378`, -0.4942884130331306`],
List[-0.750001386238209`, -0.43286649730882254`],
List[-0.7398238797968588`, -0.35436495342595553`],
List[-0.7303395497421928`, -0.2654707763053889`],
List[-0.7087968591039506`, -0.023453258580960444`],
List[-0.6862069434280346`, 0.2506456308828193`],
List[-0.6653448287717248`, 0.47409810939227504`],
List[-0.6550941824327547`, 0.5599189734032894`],
List[-0.6498581570937355`, 0.5956349265395152`],
List[-0.6442239577120495`, 0.6271947855867068`],
List[-0.6412828970965597`, 0.6406690475025774`],
List[-0.6385912090240337`, 0.651124447712424`],
List[-0.633312731602339`, 0.6662557045388081`],
List[-0.6303669227819514`, 0.6715264501165197`],
List[-0.6271504370779537`, 0.6746302040365503`],
List[-0.6241122714698554`, 0.6749890352056996`],
List[-0.6212907785054942`, 0.673065392685534`],
List[-0.615848103126219`, 0.6631864529584339`],
List[-0.610719214050287`, 0.6464572445824115`],
List[-0.6061140961990733`, 0.6253747705571909`],
List[-0.6011310493901006`, 0.5962432752291937`],
List[-0.58993621358246`, 0.5080588617703872`],
List[-0.5796332258514509`, 0.40179883304703257`],
List[-0.5419911185092119`, -0.11563789469737708`],
```

```
List[-0.5212657131179157`, -0.41140376620744473`],
List[-0.5016259570844893`, -0.6383082637499763`],
List[-0.49139111028154175`, -0.7225368945842895`],
List[-0.4856072258904732`, -0.7575352067107914`],
List[-0.4802543250749207`, -0.78114955618428`],
List[-0.4757632963228615`, -0.7941758653567549`],
List[-0.47080452614718676`, -0.8011599707142121`],
List[-0.46562566801066907`, -0.800017069509857`],
List[-0.460762181661922`, -0.7910321667400466`],
List[-0.45609740432417917`, -0.7752294176464226`],
List[-0.4511846581763733`, -0.7510703522206259`],
List[-0.4469901738226814`, -0.7244650488260634`],
List[-0.4424004989015632`, -0.6892291015728184`],
List[-0.43214488080401514`, -0.5887125083849086`],
List[-0.4226003398105357`, -0.4709297597147562`],
List[-0.37797369624595195`, 0.25162584454478143`],
List[-0.35605021441036094`, 0.5894204860628905`],
List[-0.34642322744337106`, 0.7085825762585028`],
List[-0.34131725061385637`, 0.7619387936024965`],
List[-0.3358302221056187`, 0.8107423480186627`],
List[-0.33042975244711636`, 0.8495136641902389`],
List[-0.3254936995516747`, 0.8764609305371227`],
List[-0.32053192780411477`, 0.8950417017536064`],
List[-0.3158200401288461`, 0.9045714043821929`],
List[-0.3105605721446212`, 0.9057062735594216`],
List[-0.3056826236909325`, 0.8977335870214513`],
List[-0.30032936887031186`, 0.8790188868261573`],
List[-0.2973379125422991`, 0.8640692854201357`],
List[-0.29462830641223864`, 0.847791590281556`],
List[-0.2835658740663681`, 0.7553319633410993`],
List[-0.27733670569774405`, 0.6860096772305007`],
List[-0.2716113406541768`, 0.6123778152682854`],
List[-0.24994991234572664`, 0.2655800740728669`],
List[-0.21225773815848598`, -0.43057614557528495`],
List[-0.2014045185385447`, -0.607001214670649`],
List[-0.1911333494049853`, -0.7490273389929445`],
List[-0.18191240706908424`, -0.8505632497565038`],
List[-0.1767416129962974`, -0.8952652342589389`],
List[-0.1718920901267761`, -0.9285918556554922`],
List[-0.16697957316058187`, -0.9535023649777139`],
List[-0.161562565091676`, -0.9703221049620906`],
List[-0.15628418144238038`, -0.9757476325207776`],
List[-0.15331730465130441`, -0.9740039752842746`],
List[-0.15059977644037958`, -0.9693770667343177`],
List[-0.1460165797564408`, -0.9550405828469034`],
List[-0.1410081756896768`, -0.9301062215571163`],
List[-0.13577962138178978`, -0.893991278427419`],
List[-0.13091999659907888`, -0.8515076827970871`],
List[-0.12123049018585531`, -0.7427944269588428`],
List[-0.11243516664911235`, -0.6192856959607796`],
List[-0.09270736290683174`, -0.2771819714654778`],
List[-0.07074128821999481`, 0.15456352776450047`],
List[-0.047508779695364454`, 0.5802291446050808`],
List[-0.03794148475458846`, 0.7245979117086273`],
List[-0.027673730365563417`, 0.8500519974179697`],
List[-0.021783670915650262`, 0.9061559525105312`],
List[-0.01640722304970641`, 0.9463871901825228`],
List[-0.01100269176883796`, 0.9757675666255191`],
```

```
List[-0.005877094856298342`, 0.9930655997505854`],
List[-0.0012603543457706003`, 0.9996807302191512`],
List[0.003841377090691665`, 0.9970355032139022`],
List[0.00916971378005925`, 0.9831476806414581`],
List[0.01417765010626921`, 0.9598745223762447`],
List[0.018835778805047276`, 0.9295480348256523`],
List[0.02311152799549632`, 0.8945820542534499`],
List[0.03275314394312561`, 0.792159363745938`],
List[0.05323437432389867`, 0.4834057057361746`],
List[0.07482189650603684`, 0.07387515592061891`],
List[0.09497769065648072`, -0.3199678960821519`],
List[0.1165347071741706`, -0.6795558291669739`],
List[0.12290188181040415`, -0.7637091472760054`],
List[0.1289196612394384`, -0.8316141273358323`],
List[0.13477808256826967`, -0.8859263593475684`],
List[0.14018569726502234`, -0.9251000595565696`],
List[0.14531416564759841`, -0.9521232602335068`],
List[0.15074987886152633`, -0.9697081023973886`],
List[0.15620612109614104`, -0.9757460076225583`],
List[0.1593087430762231`, -0.9739713489533526`],
List[0.16221476370581453`, -0.9688971227041098`],
List[0.16745001383881453`, -0.9515105805200534`],
List[0.1703767756527555`, -0.9372399699444143`],
List[0.17302994423284257`, -0.9215400634230125`],
List[0.18306876655623647`, -0.8393213096217108`],
List[0.18869925168730767`, -0.7783995770475729`],
List[0.194788749285674`, -0.7016958861226725`],
List[0.2058274871176703`, -0.5379247908600145`],
List[0.2472775023612705`, 0.21734939392563712`],
List[0.2669876956883489`, 0.5466263082463674`],
List[0.27775578978345566`, 0.6910376340383966`],
List[0.28780683415627745`, 0.7955882301276718`],
List[0.29221298913057425`, 0.8311225084438845`],
List[0.2963862394019538`, 0.858646541028629`],
List[0.30119657032466773`, 0.8827537327474217`],
List[0.30558165684475946`, 0.8974768328574726`],
List[0.3104357848654642`, 0.905610656162231`],
List[0.31492594941144675`, 0.9054747211033035`],
List[0.319854075345852`, 0.8969036163134283`],
List[0.3250885838720468`, 0.8783015513818669`],
List[0.32986816478766834`, 0.8529956085057786`],
List[0.33509542949091004`, 0.8165707134217922`],
List[0.34562891803050866`, 0.717360860055138`],
List[0.3644582901051175`, 0.4686814149708279`],
List[0.40689090862089167`, -0.2373117913483952`],
List[0.426125604486471`, -0.5169094904177725`],
List[0.43689074850830234`, -0.6388263696340332`],
List[0.44197946198477883`, -0.6856842565689083`],
List[0.44667041262385826`, -0.7222151789885232`],
List[0.4510207771170432`, -0.7501332725563638`],
List[0.4558052679116797`, -0.7740071803631008`],
List[0.4606148703666571`, -0.7906404333586745`],
List[0.4657059069711432`, -0.8001009858284969`],
List[0.4701818412566341`, -0.8014806461151873`],
List[0.47498938854669887`, -0.7957813656725395`],
List[0.4795274484879301`, -0.7836835448282652`],
List[0.48367316714241243`, -0.7670632421985295`],
List[0.48910652724614345`, -0.7375047588806098`],
```

```
List[0.49421732943212265`, -0.7020199626620278`],
List[0.5037632278937437`, -0.6174246056732089`],
List[0.5251478059134861`, -0.35894759636214624`],
List[0.5635945404101352`, 0.19857175803531235`],
List[0.5742159503756941`, 0.33764078266505365`],
List[0.5854201872696259`, 0.4642184562076209`],
List[0.59498618095222`, 0.5515984138888202`],
List[0.6003741082859045`, 0.591255829543805`],
List[0.60547052133523`, 0.6219780313442358`],
List[0.6083075285152632`, 0.6361250407609035`],
List[0.6113889278134459`, 0.6490479456023531`],
List[0.6167770249727986`, 0.6654477396485784`],
List[0.6198296973349456`, 0.6712117051612357`],
List[0.623067888318758`, 0.6745310254743232`],
List[0.6258324575645496`, 0.6750943051423131`],
List[0.6288793992299165`, 0.6733081258630773`],
List[0.634303848090634`, 0.663966651997967`],
List[0.6373520249762497`, 0.6553223823800816`],
List[0.6401426151420501`, 0.6453200158057888`],
List[0.645553737508074`, 0.6204158430707607`],
List[0.650700713891915`, 0.5902905186685075`],
List[0.6939669926869411`, 0.15752673366231526`],
List[0.7152020721601373`, -0.09959206992178356`],
List[0.7347515180442568`, -0.308535470979283`],
List[0.7439550305288842`, -0.388503320333675`],
List[0.7525526748937331`, -0.44947412460545044`],
List[0.761520709343437`, -0.4973020699793587`],
List[0.7662730077468173`, -0.5157230754805726`],
List[0.7713706752869288`, -0.5299915837104358`],
List[0.7764655824982817`, -0.5385135508046426`],
List[0.7820111997625422`, -0.5412697136677055`],
List[0.7872487221463473`, -0.5377047719381315`],
List[0.7920974807320128`, -0.5291826799478381`],
List[0.7964529521305389`, -0.5173784076900075`],
List[0.8012635440655589`, -0.49995684200916335`],
List[0.8110073898497123`, -0.45154940718504377`],
List[0.8216866556673058`, -0.38077890138613835`],
List[0.8314843412770514`, -0.30276272782875757`],
List[0.8535723501576126`, -0.09933208457858644`],
List[0.8742799127520443`, 0.09562488950229762`],
List[0.8935577024893068`, 0.25127274525933935`],
List[0.9034528690271358`, 0.3141373454867022`],
List[0.9140306207806459`, 0.3653613330128868`],
List[0.9200662975061342`, 0.38653503814918144`],
List[0.9230318897571291`, 0.39471010144585883`],
List[0.9257679017110405`, 0.40093262096077703`],
List[0.9308868297973646`, 0.40915363270998506`],
List[0.9364394027231373`, 0.41303250288556703`],
List[0.9418327858488762`, 0.4118351189538449`],
List[0.9468818702625253`, 0.4063788296314459`],
List[0.9524211948594483`, 0.3957343604592793`],
List[0.9555317848759625`, 0.3877026178486205`],
List[0.9583757626343072`, 0.37911768876508517`],
List[0.9702935393392168`, 0.33123670459888427`],
List[0.9816146322866746`, 0.2704990999329876`],
List[1.0219337574845162`, -0.006447852680366173`],
List[1.0440580327404785`, -0.1494976259339504`],
List[1.0549091834196738`, -0.2060834353790531`],
```

```
List[1.0648858169822157`, -0.2474436640448388`],
List[1.0743602626221473`, -0.27609822989107596`],
List[1.0791303978563087`, -0.2863886585377893`],
List[1.0843627332104027`, -0.2944202724381302`],
List[1.0896214981734995`, -0.29904808873811173`],
List[1.0925692255875215`, -0.30013993168987985`],
List[1.0952550142087834`, -0.30020482778189655`],
List[1.1004751463501758`, -0.2978354880883429`],
List[1.1054168621295604`, -0.2926351096973051`],
List[1.1098516796268472`, -0.28561328923965135`],
List[1.1145598420490528`, -0.2758356038124899`],
List[1.1230130242444196`, -0.25271804093753775`],
List[1.142140568536079`, -0.17872230408925346`],
List[1.1815509438027387`, 0.01708703617472436`],
List[1.2022395560575547`, 0.10941576698635769`],
List[1.2126386342258384`, 0.1464307316264005`],
List[1.218596390232883`, 0.16405381673229888`],
List[1.2240060265504031`, 0.17759683511463492`],
List[1.2288279588669109`, 0.18761035167194878`],
List[1.2340514887294487`, 0.19620888071038206`],
List[1.2395697767980838`, 0.20271393807973756`],
List[1.2447757706178375`, 0.20641373354368245`],
List[1.2496593073706657`, 0.20775038993425438`],
List[1.2548260055427813`, 0.20695695951905246`],
List[1.2592371268498836`, 0.20453200348774384`],
List[1.26408599477186`, 0.20007719367288554`],
List[1.268870350843575`, 0.19392842885484418`],
List[1.2732388716663274`, 0.18687694203521504`],
List[1.2830649217771246`, 0.1664640675381528`],
List[1.3008089017854554`, 0.11683131393049553`],
List[1.3395619533221224`, -0.014559309921745341`],
List[1.3497807250584448`, -0.04656505959061242`],
List[1.360812454108068`, -0.07699189854131996`],
List[1.3705376522004298`, -0.09930216737751288`],
List[1.3807913421137497`, -0.11751844389690592`],
List[1.3866981275031691`, -0.1253500384725379`],
List[1.389824237087754`, -0.12868205488181444`],
List[1.393168712185386`, -0.13161783385082187`],
List[1.3962490792977973`, -0.1337452708365822`],
List[1.399052023099811`, -0.1352016057928178`],
List[1.4045335760246098`, -0.13673969740956485`],
List[1.4098285490083289`, -0.13660754156569366`],
List[1.4128172144948126`, -0.13584859261362373`],
List[1.4155767598509525`, -0.13472094801234416`],
List[1.4206443460942033`, -0.1316169197489221`],
List[1.4260171995048834`, -0.12693301271192878`],
List[1.436311228356535`, -0.1143182202119677`],
List[1.4459559617815967`, -0.09877639029056667`],
List[1.4679201985139687`, -0.054223359972494584`],
List[1.4887579343823243`, -0.00762072873950793`],
List[1.5115059793294574`, 0.0382345442533714`],
List[1.5219052911983821`, 0.05512766037480968`],
List[1.5329927229796834`, 0.06937991404491133`],
List[1.5383799169472308`, 0.07476578756256917`],
List[1.5434555481172565`, 0.07887721208535046`],
List[1.5480128760885281`, 0.08176009227545963`],
List[1.5529432631496285`, 0.08401317903186631`],
List[1.5577999319927756`, 0.08535730267782553`],
```

```
    List[1.562334603946491`, 0.08583929036751099`],
    List[1.5673100084113765`, 0.08552953552719611`],
    List[1.5726443116134108`, 0.08425618533669176`],
    List[1.5753219956329856`, 0.08326380557115205`],
    List[1.5782350884201926`, 0.08192639160190762`],
    List[1.5835319886303225`, 0.07883722949145197`],
    List[1.5934602592130633`, 0.07096490215862193`],
    List[1.6128244376616974`, 0.04948420103577662`],
    List[1.6307422583466153`, 0.02543404632466773`],
    List[1.6491132192575328`, 0.00029382516263982446`],
    List[1.6684570224553763`, -0.023063944014359752`],
    List[1.6785864950413831`, -0.032992068553395676`],
    List[1.6896682018240128`, -0.04155430222370065`],
    List[1.6945391991512138`, -0.044491418002653244`],
    List[1.6997435873078401`, -0.047049906538513514`],
    List[1.7046552079069046`, -0.04890844329652339`],
    List[1.7091185902618236`, -0.05012830831614879`],
    List[1.711901482632756`, -0.05066418084712502`],
    List[1.7148481318213324`, -0.05104538406562282`],
    List[1.7201314630241844`, -0.051257364803291655`],
    List[1.7230685323472925`, -0.051119227645411644`],
    List[1.725816724432411`, -0.05082882337742177`],
    List[1.7288318026150509`, -0.050335628120028626`],
    List[1.7320712690896647`, -0.049608398055116146`],
    List[1.7378798445858235`, -0.047816864809224556`],
    List[1.743394830049025`, -0.045575434520178475`],
    List[1.7537426239565226`, -0.04012069650849713`],
    List[1.7737875026298968`, -0.026116835698228463`],
    List[1.7956411814708972`, -0.00850661872577543`],
    List[1.8180209034331518`, 0.008440018725300728`],
    List[1.8289944674381577`, 0.01538353194694639`],
    List[1.8390785712365771`, 0.020647516279422223`],
    List[1.8482787300585155`, 0.024393268036740877`],
    List[1.8529669537056004`, 0.02589139337639477`],
    List[1.8579867443579592`, 0.02718112978745319`],
    List[1.8630295576621545`, 0.028148004455478503`],
    List[1.8677318350704086`, 0.02875436483964524`],
    List[1.8706049904237505`, 0.028986380194225708`],
    List[1.873323538248825`, 0.0291106936040121`],
    List[1.8784157905220895`, 0.02909979399104083`],
    List[1.8834819593167174`, 0.028783911606270977`],
    List[1.8863644484863291`, 0.028473886915831758`],
    List[1.889067323789397`, 0.0281007099146851`],
    List[1.8945097503439552`, 0.027118590924287478`],
    List[1.9003696902891154`, 0.025740332252873793`],
    List[1.9109593009628465`, 0.022514858084424577`],
    List[1.920657051873991`, 0.018892401701557857`],
    List[1.9665514341784773`, -0.0012774939524654253`],
    List[1.9999998333333333`, -0.0122153593511319`]]]]],
List[Rule[PlotRange, Automatic], Rule[AspectRatio,
  Power[GoldenRatio, -1]],
 RuleDelayed[DisplayFunction, $DisplayFunction],
 Rule[ColorOutput, Automatic],
 Rule[Axes, Automatic],
 Rule[AxesOrigin, Automatic],
 Rule[PlotLabel, None],
 Rule[AxesLabel, None],
 Rule[Ticks, Automatic],
```

```
          Rule[GridLines, None],
          Rule[Prolog, List[]],
          Rule[Epilog, List[]],
          Rule[AxesStyle, Automatic],
          Rule[Background, Automatic],
          Rule[DefaultColor, Automatic],
          RuleDelayed[
           DefaultFont, $DefaultFont],
          Rule[RotateLabel, True],
          Rule[Frame, False],
          Rule[FrameStyle, Automatic],
          Rule[FrameTicks, Automatic],
          Rule[FrameLabel, None],
          Rule[PlotRegion, Automatic],
          Rule[ImageSize, Automatic],
          RuleDelayed[
           TextStyle, $TextStyle],
          RuleDelayed[FormatType, $FormatType]]]
```

In[186]:=

```
Get["C:\\work\\MathematicaData\\postscri.ps"]
```

Out[186]=

```
- Graphics -
```

In[187]:=

```
Get["C:\\work\\MathematicaData\\postscri.ps"] // FullForm
```

Out[187]//FullForm=

```
Graphics[List[List[Line[List[List[-1.9999998333333333`, -0.0122153593511319`],
     List[-1.9623003192812842`, 0.0005082587300581328`],
     List[-1.9210073169479194`, 0.018752002927052425`],
     List[-1.910299579465502`, 0.022740269212148626`], List[-1.9001774790199295`,
      0.025790484842825256`], List[-1.8945829997783903`, 0.027103353077953987`],
     List[-1.8916352820184055`, 0.02767476304778044`],
     List[-1.8885103672286825`, 0.028184000386560142`],
     List[-1.8829909524966886`, 0.028827453836099467`],
     List[-1.8778952782513714`, 0.029115238245558964`],
     List[-1.8731413317543735`, 0.029105227315202174`],
     List[-1.8680274614917154`, 0.028783053185285283`],
     List[-1.863199958792915`, 0.028174933726430717`],
     List[-1.8587831546785645`, 0.027355730435410536`],
     List[-1.8536232573178666`, 0.026078553418176047`],
     List[-1.8479909934251308`, 0.024292164406973862`],
     List[-1.8377320337083367`, 0.020012423042739088`],
     List[-1.8257366220651903`, 0.013444784972129038`],
     List[-1.8144362944250474`, 0.005937356649683411`],
     List[-1.7933141736895866`, -0.010391844332965274`],
     List[-1.768652621938747`, -0.030023715053337733`],
     List[-1.7578358910677812`, -0.03757544820569005`],
     List[-1.745962817885428`, -0.04436555940559883`],
     List[-1.7409954301270045`, -0.04661226368848812`],
     List[-1.7356703543983865`, -0.04856988304671834`],
     List[-1.7306441655063993`, -0.04995350510306155`],
     List[-1.726061421963292`, -0.05079554189295374`],
     List[-1.7208899413754872`, -0.05123896481969352`],
     List[-1.7179679228759093`, -0.05124293728587858`],
     List[-1.7152644782688042`, -0.05108393336125241`],
     List[-1.7103050520507237`, -0.0503777772427448`],
```

```
List[-1.705036285194253`, -0.04903000884182273`],
List[-1.6998130954536856`, -0.04707998343149028`],
List[-1.6941824661510267`, -0.04429399736242225`],
List[-1.6839927695262276`, -0.03748701340302931`],
List[-1.6718751692383962`, -0.026621373541510396`],
List[-1.660764800562505`, -0.01436785949148811`],
List[-1.6223373700539767`, 0.03698336903382436`],
List[-1.6122046687997946`, 0.05026582756968046`],
List[-1.6012204218638049`, 0.0631825070687631`],
List[-1.5915368555668616`, 0.07268572843936418`],
List[-1.58624016644988`, 0.07694697859595899`],
List[-1.5812846856736067`, 0.08024862424865138`],
List[-1.5754383042802425`, 0.0832155003092454`],
List[-1.572337956846547`, 0.08435492782676386`],
List[-1.5690072708401817`, 0.08522809918894149`],
List[-1.5659728443536363`, 0.08569752092152322`],
List[-1.5630921326197909`, 0.08584815274454088`],
List[-1.5576873525951036`, 0.08533590009119145`],
List[-1.5523642925764947`, 0.08379516017296415`],
List[-1.5474289001941306`, 0.08143362486126446`],
List[-1.5420125470492216`, 0.07780448278522988`],
List[-1.536246079580289`, 0.07275700026813073`],
List[-1.5249155505844803`, 0.059402686355185516`],
List[-1.5142556242299647`, 0.04300384325934713`],
List[-1.4943432743542384`, 0.004472710594773914`],
List[-1.4532279345574657`, -0.08515390290044901`],
List[-1.4432335331173123`, -0.10348325385511208`],
List[-1.432533564692643`, -0.11946821817256431`],
List[-1.42638894081693`, -0.12655827281836432`],
List[-1.4207938199321637`, -0.13150559927652575`],
List[-1.41809797650952`, -0.13334094680801603`],
List[-1.4151693384029775`, -0.13491283836164045`],
List[-1.4098481479164913`, -0.13660415416094238`],
List[-1.4050416145772238`, -0.13679532253598312`],
List[-1.4005351214442552`, -0.1357881187725129`],
List[-1.3955913202634447`, -0.13333739136336295`],
List[-1.3902688897887092`, -0.12910990712973433`],
List[-1.3849829640264526`, -0.1232818098314748`],
List[-1.3794251016434265`, -0.11542481340098515`],
List[-1.36949576036685`, -0.09713667609456161`],
List[-1.3516769177229944`, -0.05213560711740385`],
List[-1.3322109850418382`, 0.010049649308713196`],
List[-1.311814606942803`, 0.0805873886472376`],
List[-1.301081564243093`, 0.11597117145006212`],
List[-1.2897332454943118`, 0.14947293224243818`],
List[-1.2797977572941663`, 0.17390652324509817`],
List[-1.2745323012358132`, 0.1845386245184353`],
List[-1.2688830972260063`, 0.19390980702104588`],
List[-1.2636935060450525`, 0.20050558155842912`],
List[-1.2589579198010443`, 0.20473215653759175`],
List[-1.2541975830843266`, 0.20717275659823553`],
List[-1.249684335929672`, 0.20775197540724452`],
List[-1.247041828138711`, 0.20728903288692044`],
List[-1.2441226325567252`, 0.20607895510875865`],
List[-1.2410738117547113`, 0.20402551530318114`],
List[-1.2382092114607313`, 0.20135719496762308`],
List[-1.2330981005800692`, 0.19481588054246313`],
List[-1.2275838049165742`, 0.18521572373110567`],
```

```
List[-1.2214743617585684`, 0.17155776588451163`],
List[-1.2150369184940277`, 0.1538563691722545`],
List[-1.203523571823205`, 0.11438509646840048`],
List[-1.1616357322503013`, -0.0838667012692308`],
List[-1.1410574771688768`, -0.18355519085192498`],
List[-1.1309454699394663`, -0.2252446687030288`],
List[-1.1216144865495554`, -0.2570070238434482`],
List[-1.116498155471344`, -0.2711457519497164`],
List[-1.1108655908628202`, -0.28370556740513797`],
List[-1.1081370509617479`, -0.2885858899753463`],
List[-1.1052034083602063`, -0.29291770319564026`],
List[-1.1021050224055187`, -0.2964329900999418`],
List[-1.0992038714781798`, -0.29871236701229176`],
List[-1.0937451922278991`, -0.3002768983062175`],
List[-1.0885588150162093`, -0.29839061424267377`],
List[-1.0839033652069439`, -0.2938521069914662`],
List[-1.0788242866321036`, -0.28581302032434724`],
List[-1.0735924733113682`, -0.27417965132220024`],
List[-1.068666299969677`, -0.26016454226758307`],
List[-1.0594607465548975`, -0.22631156310060757`],
List[-1.0388287578280821`, -0.11852835113159345`],
List[-1.0017158165421325`, 0.1380386876955374`],
List[-0.9804818490676098`, 0.27714025981739737`],
List[-0.9701152472705907`, 0.3320839289435045`],
List[-0.9606680178167547`, 0.37136370846023714`],
List[-0.9555429871463601`, 0.3876711007347637`],
List[-0.9506903685808147`, 0.39957150651606144`],
List[-0.94556841761877`, 0.4081946077693798`],
List[-0.9426751414885124`, 0.411212892719486`],
List[-0.9399996360794677`, 0.41278513550084933`],
List[-0.934770753521844`, 0.4124157216304582`],
List[-0.9317946152243468`, 0.41014558532629825`],
List[-0.929001985645631`, 0.4066458604324708`],
List[-0.9260688269140263`, 0.40153938048657495`],
List[-0.9233808299909719`, 0.39557434627057825`],
List[-0.9173242395579699`, 0.3776634222547061`],
List[-0.9127976387820719`, 0.3602999772862939`],
List[-0.9079642254182297`, 0.3381144016387083`],
List[-0.8992602803223766`, 0.28915729392454115`],
List[-0.8795104906864795`, 0.1413842470955593`],
List[-0.8397190375423287`, -0.23005651682605105`],
List[-0.8187971422559015`, -0.4015572594838865`],
List[-0.8082518687084622`, -0.4669266707753546`],
List[-0.8022604864947778`, -0.49579074972261117`],
List[-0.7966578833395991`, -0.5167287009438404`],
List[-0.7935177520634596`, -0.525757956681514`],
List[-0.7905381888494231`, -0.5324618998910939`],
List[-0.7878635232886223`, -0.5368991395999997`],
List[-0.7849493165407382`, -0.5400002249881906`],
List[-0.7817880476133191`, -0.541289292474234`],
List[-0.7784241259699886`, -0.5402608150821243`],
List[-0.7752499569500357`, -0.5370024066246588`],
List[-0.7723771954811092`, -0.5321307725565154`],
List[-0.7668323840651169`, -0.5175679461624914`],
List[-0.7637114688582315`, -0.5064022525594425`],
List[-0.7608453028627378`, -0.4942884130331306`],
List[-0.750001386238209`, -0.43286649730882254`],
List[-0.7398238797968588`, -0.35436495342595553`],
```

```
List[-0.7303395497421928`, -0.2654707763053889`],
List[-0.7087968591039506`, -0.023453258580960444`],
List[-0.6862069434280346`, 0.2506456308828193`],
List[-0.6653448287717248`, 0.47409810939227504`],
List[-0.6550941824327547`, 0.5599189734032894`],
List[-0.6498581570937355`, 0.5956349265395152`],
List[-0.6442239577120495`, 0.6271947855867068`],
List[-0.6412828970965597`, 0.6406690475025774`],
List[-0.6385912090240337`, 0.651124447712424`],
List[-0.633312731602339`, 0.6662557045388081`],
List[-0.6303669227819514`, 0.6715264501165197`],
List[-0.6271504370779537`, 0.6746302040365503`],
List[-0.6241122714698554`, 0.6749890352056996`],
List[-0.6212907785054942`, 0.673065392685534`],
List[-0.615848103126219`, 0.6631864529584339`],
List[-0.610719214050287`, 0.6464572445824115`],
List[-0.6061140961990733`, 0.6253747705571909`],
List[-0.6011310493901006`, 0.5962432752291937`],
List[-0.58993621358246`, 0.5080588617703872`],
List[-0.5796332258514509`, 0.40179883304703257`],
List[-0.5419911185092119`, -0.11563789469737708`],
List[-0.5212657131179157`, -0.41140376620744473`],
List[-0.5016259570844893`, -0.6383082637499763`],
List[-0.49139111028154175`, -0.7225368945842895`],
List[-0.4856072258904732`, -0.7575352067107914`],
List[-0.4802543250749207`, -0.78114955618428`],
List[-0.4757632963228615`, -0.7941758653567549`],
List[-0.47080452614718676`, -0.8011599707142121`],
List[-0.46562566801066907`, -0.800017069509857`],
List[-0.460762181661922`, -0.7910321667400466`],
List[-0.4560974043241791 7`, -0.7752294176464226`],
List[-0.4511846581763733`, -0.7510703522206259`],
List[-0.4469901738226814`, -0.7244650488260634`],
List[-0.4424004989015632`, -0.6892291015728184`],
List[-0.43214488080401514`, -0.5887125083849086`],
List[-0.4226003398105357`, -0.4709297597147562`],
List[-0.37797369624595195`, 0.25162584454478143`],
List[-0.35605021441036094`, 0.5894204860628905`],
List[-0.34642322744337106`, 0.7085825762585028`],
List[-0.34131725061385637`, 0.7619387936024965`],
List[-0.3358302221056187`, 0.8107423480186627`],
List[-0.33042975244711636`, 0.8495136641902389`],
List[-0.3254936995516747`, 0.8764609305371227`],
List[-0.32053192780411477`, 0.8950417017536064`],
List[-0.3158200401288461`, 0.9045714043821929`],
List[-0.3105605721446212`, 0.9057062735594216`],
List[-0.3056826236909325`, 0.8977335870214513`],
List[-0.30032936887031186`, 0.8790188868261573`],
List[-0.2973379125422991`, 0.8640692854201357`],
List[-0.29462830641223864`, 0.847791590281556`],
List[-0.2835658740663681`, 0.7553319633410993`],
List[-0.27733670569774405`, 0.6860096772305007`],
List[-0.2716113406541768`, 0.6123778152682854`],
List[-0.24994991234572664`, 0.2655800740728669`],
List[-0.21225773815848598`, -0.43057614557528495`],
List[-0.2014045185385447`, -0.607001214670649`],
List[-0.1911333494049853`, -0.7490273389929445`],
List[-0.18191240706908424`, -0.8505632497565038`],
```

```
List[-0.1767416129962974`, -0.8952652342589389`],
List[-0.1718920901267761`, -0.9285918556554922`],
List[-0.16697957316058187`, -0.9535023649777139`],
List[-0.161562565091676`, -0.9703221049620906`],
List[-0.15628418144238038`, -0.9757476325207776`],
List[-0.15331730465130441`, -0.9740039752842746`],
List[-0.15059977644037958`, -0.9693770667343177`],
List[-0.1460165797564408`, -0.9550405828469034`],
List[-0.1410081756896768`, -0.9301062215571163`],
List[-0.13577962138178978`, -0.893991278427419`],
List[-0.13091999659907888`, -0.8515076827970871`],
List[-0.12123049018585531`, -0.7427944269588428`],
List[-0.11243516664911235`, -0.6192856959607796`],
List[-0.09270736290683174`, -0.2771819714654778`],
List[-0.07074128821999481`, 0.15456352776450047`],
List[-0.047508779695364454`, 0.5802291446050808`],
List[-0.03794148475458846`, 0.7245979117086273`],
List[-0.027673730365563417`, 0.8500519974179697`],
List[-0.021783670915650262`, 0.9061559525105312`],
List[-0.01640722304970641`, 0.9463871901825228`],
List[-0.01100269176883796`, 0.9757675666255191`],
List[-0.005877094856298342`, 0.9930655997505854`],
List[-0.0012603543457706003`, 0.9996807302191512`],
List[0.003841377090691665`, 0.9970355032139022`],
List[0.00916971378005925`, 0.9831476806414581`],
List[0.01417765010626921`, 0.9598745223762447`],
List[0.018835778805047276`, 0.9295480348256523`],
List[0.02311152799549632`, 0.8945820542534499`],
List[0.03275314394312561`, 0.792159363745938`],
List[0.05323437432389867`, 0.4834057057361746`],
List[0.07482189650603684`, 0.07387515592061891`],
List[0.09497769065648072`, -0.3199678960821519`],
List[0.1165347071741706`, -0.6795558291669739`],
List[0.12290188181040415`, -0.7637091472760054`],
List[0.1289196612394384`, -0.8316141273358323`],
List[0.13477808256826967`, -0.8859263593475684`],
List[0.14018569726502234`, -0.9251000595565696`],
List[0.14531416564759841`, -0.9521232602335068`],
List[0.15074987886152633`, -0.9697081023973886`],
List[0.15620612109614104`, -0.9757460076225583`],
List[0.1593087430762231`, -0.9739713489533526`],
List[0.16221476370581453`, -0.9688971227041098`],
List[0.16745001383881453`, -0.9515105805200534`],
List[0.1703767756527555`, -0.9372399699444143`],
List[0.17302994423284257`, -0.9215400634230125`],
List[0.18306876655623647`, -0.8393213096217108`],
List[0.18869925168730767`, -0.7783995770475729`],
List[0.194788749285674`, -0.7016958861226725`],
List[0.2058274871176703`, -0.5379247908600145`],
List[0.2472775023612705`, 0.21734939392563712`],
List[0.2669876956883489`, 0.5466263082463674`],
List[0.27775578978345566`, 0.6910376340383966`],
List[0.28780683415627745`, 0.7955882301276718`],
List[0.29221298913057425`, 0.8311225084438845`],
List[0.2963862394019538`, 0.858646541028629`],
List[0.30119657032466773`, 0.8827537327474217`],
List[0.30558165684475946`, 0.8974768328574726`],
List[0.3104357848654642`, 0.905610656162231`],
```

```
List[0.31492594941144675`, 0.9054747211033035`],
List[0.319854075345852`, 0.8969036163134283`],
List[0.3250885838720468`, 0.8783015513818669`],
List[0.32986816478766834`, 0.8529956085057786`],
List[0.33509542949091004`, 0.8165707134217922`],
List[0.34562891803050866`, 0.717360860055138`],
List[0.3644582901051175`, 0.4686814149708279`],
List[0.40689090862089167`, -0.2373117913483952`],
List[0.426125604486471`, -0.5169094904177725`],
List[0.43689074850830234`, -0.6388263696340332`],
List[0.44197946198477883`, -0.6856842565689083`],
List[0.44667041262385826`, -0.7222151789885232`],
List[0.4510207771170432`, -0.7501332725563638`],
List[0.4558052679116797`, -0.7740071803631008`],
List[0.4606148703666571`, -0.7906404333586745`],
List[0.4657059069711432`, -0.8001009858284969`],
List[0.4701818412566341`, -0.8014806461151873`],
List[0.47498938854669887`, -0.7957813656725395`],
List[0.4795274484879301`, -0.7836835448282652`],
List[0.48367316714241243`, -0.7670632421985295`],
List[0.48910652724614345`, -0.7375047588806098`],
List[0.49421732943212265`, -0.7020199626620278`],
List[0.5037632278937437`, -0.6174246056732089`],
List[0.5251478059134861`, -0.35894759636214624`],
List[0.5635945404101352`, 0.19857175803531235`],
List[0.5742159503756941`, 0.33764078266505365`],
List[0.5854201872696259`, 0.4642184562076209`],
List[0.59498618095222`, 0.5515984138888202`],
List[0.6003741082859045`, 0.591255829543805`],
List[0.60547052133523`, 0.6219780313442358`],
List[0.6083075285152632`, 0.6361250407609035`],
List[0.6113889278134459`, 0.6490479456023531`],
List[0.6167770249727986`, 0.6654477396485784`],
List[0.6198296973349456`, 0.6712117051612357`],
List[0.623067888318758`, 0.6745310254743232`],
List[0.6258324575645496`, 0.6750943051423131`],
List[0.6288793992299165`, 0.6733081258630773`],
List[0.634303848090634`, 0.663966651997967`],
List[0.6373520249762497`, 0.6553223823800816`],
List[0.6401426151420501`, 0.6453200158057888`],
List[0.645553737508074`, 0.6204158430707607`],
List[0.650700713891915`, 0.5902905186685075`],
List[0.6939669926869411`, 0.15752673366231526`],
List[0.7152020721601373`, -0.09959206992178356`],
List[0.7347515180442568`, -0.308535470979283`],
List[0.7439550305288842`, -0.388503320333675`],
List[0.7525526748937331`, -0.44947412460545044`],
List[0.761520709343437`, -0.4973020699793587`],
List[0.7662730077468173`, -0.5157230754805726`],
List[0.7713706752869288`, -0.5299915837104358`],
List[0.7764655824982817`, -0.5385135508046426`],
List[0.7820111997625422`, -0.5412697136677055`],
List[0.7872487221463473`, -0.5377047719381315`],
List[0.7920974807320128`, -0.5291826799478381`],
List[0.7964529521305389`, -0.5173784076900075`],
List[0.8012635440655589`, -0.49995684200916335`],
List[0.8110073898497123`, -0.45154940718504377`],
List[0.8216866556673058`, -0.38077890138613835`],
```

```
List[0.8314843412770514`, -0.30276272782875757`],
List[0.8535723501576126`, -0.09933208457858644`],
List[0.8742799127520443`, 0.09562488950229762`],
List[0.8935577024893068`, 0.25127274525933935`],
List[0.9034528690271358`, 0.3141373454867022`],
List[0.9140306207806459`, 0.3653613330128868`],
List[0.9200662975061342`, 0.38653503814918144`],
List[0.9230318897571291`, 0.39471010144585883`],
List[0.9257679017110405`, 0.40093262096077703`],
List[0.9308868297973646`, 0.40915363270998506`],
List[0.9364394027231373`, 0.41303250288556703`],
List[0.9418327858488762`, 0.4118351189538449`],
List[0.9468818702625253`, 0.4063788296314459`],
List[0.9524211948594483`, 0.3957343604592793`],
List[0.9555317848759625`, 0.3877026178486205`],
List[0.9583757626343072`, 0.37911768876508517`],
List[0.9702935393392168`, 0.33123670459888427`],
List[0.9816146322866746`, 0.2704990999329876`],
List[1.0219337574845162`, -0.006447852680366173`],
List[1.0440580327404785`, -0.1494976259339504`],
List[1.0549091834196738`, -0.2060834353790531`],
List[1.0648858169822157`, -0.2474436640448388`],
List[1.0743602626221473`, -0.27609822989107596`],
List[1.0791303978563087`, -0.2863886585377893`],
List[1.0843627332104027`, -0.2944202724381302`],
List[1.0896214981734995`, -0.29904808873811173`],
List[1.0925692255875215`, -0.30013993168987985`],
List[1.0952550142087834`, -0.30020482778189655`],
List[1.1004751463501758`, -0.2978354880883429`],
List[1.1054168621295604`, -0.2926351096973051`],
List[1.1098516796268472`, -0.28561328923965135`],
List[1.1145598420490528`, -0.2758356038124899`],
List[1.1230130242444196`, -0.25271804093753775`],
List[1.142140568536079`, -0.17872230408925346`],
List[1.1815509438027387`, 0.01708703617472436`],
List[1.2022395560575547`, 0.10941576698635769`],
List[1.2126386342258384`, 0.1464307316264005`],
List[1.218596390232883`, 0.16405381673229888`],
List[1.2240060265504031`, 0.17759683511463492`],
List[1.2288279588669109`, 0.18761035167194878`],
List[1.2340514887294487`, 0.19620888071038206`],
List[1.2395697767980838`, 0.20271393807973756`],
List[1.2447757706178375`, 0.20641373354368245`],
List[1.2496593073706657`, 0.20775038993425438`],
List[1.2548260055427813`, 0.20695695951905246`],
List[1.2592371268498836`, 0.20453200348774384`],
List[1.26408599477186`, 0.20007719367288554`],
List[1.268870350843575`, 0.19392842885484418`],
List[1.2732388716663274`, 0.18687694203521504`],
List[1.2830649217771246`, 0.1664640675381528`],
List[1.3008089017854554`, 0.11683131393049553`],
List[1.3395619533221224`, -0.014559309921745341`],
List[1.3497807250584448`, -0.04656505959061242`],
List[1.360812454108068`, -0.07699189854131996`],
List[1.3705376522004298`, -0.09930216737751288`],
List[1.3807913421137497`, -0.11751844389690592`],
List[1.3866981275031691`, -0.1253500384725379`],
List[1.389824237087754`, -0.12868205488181444`],
```

```
List[1.393168712185386`, -0.13161783385082187`],
List[1.3962490792977973`, -0.1337452708365822`],
List[1.399052023099811`, -0.1352016057928178`],
List[1.4045335760246098`, -0.13673969740956485`],
List[1.4098285490083289`, -0.13660754156569366`],
List[1.4128172144948126`, -0.13584859261362373`],
List[1.4155767598509525`, -0.13472094801234416`],
List[1.4206443460942033`, -0.1316169197489221`],
List[1.4260171995048834`, -0.12693301271192878`],
List[1.436311228356535`, -0.1143182202119677`],
List[1.4459559617815967`, -0.09877639029056667`],
List[1.4679201985139687`, -0.054223359972494584`],
List[1.4887579343823243`, -0.00762072873950793`],
List[1.5115059793294574`, 0.0382345442533714`],
List[1.5219052911983821`, 0.05512766037480968`],
List[1.5329927229796834`, 0.06937991404491133`],
List[1.5383799169472308`, 0.07476578756256917`],
List[1.5434555481172565`, 0.07887721208535046`],
List[1.5480128760885281`, 0.08176009227545963`],
List[1.5529432631496285`, 0.08401317903186631`],
List[1.5577999319927756`, 0.08535730267782553`],
List[1.562334603946491`, 0.08583929036751099`],
List[1.5673100084113765`, 0.08552953552719611`],
List[1.5726443116134108`, 0.08425618533669176`],
List[1.5753219956329856`, 0.08326380557115205`],
List[1.5782350884201926`, 0.08192639160190762`],
List[1.5835319886303225`, 0.07883722949145197`],
List[1.5934602592130633`, 0.07096490215862193`],
List[1.6128244376616974`, 0.04948420103577662`],
List[1.6307422583466153`, 0.02543404632466773`],
List[1.6491132192575328`, 0.00029382516263982446`],
List[1.6684570224553763`, -0.023063944014359752`],
List[1.6785864950413831`, -0.032992068553395676`],
List[1.6896682018240128`, -0.04155430222370065`],
List[1.6945391991512138`, -0.044491418002653244`],
List[1.6997435873078401`, -0.047049906538513514`],
List[1.7046552079069046`, -0.04890844329652339`],
List[1.7091185902618236`, -0.05012830831614879`],
List[1.711901482632756`, -0.05066418084712502`],
List[1.7148481318213324`, -0.05104538406562282`],
List[1.7201314630241844`, -0.051257364803291655`],
List[1.7230685323472925`, -0.05111922764541164`],
List[1.725816724432411`, -0.05082882337742177`],
List[1.7288318026150509`, -0.050335628120028626`],
List[1.7320712690896647`, -0.049608398055116146`],
List[1.7378798445858235`, -0.047816864809224556`],
List[1.743394830049025`, -0.045575434520178475`],
List[1.7537426239565226`, -0.04012069650849713`],
List[1.7737875026298968`, -0.026116835698228463`],
List[1.7956411814708972`, -0.00850661872577543`],
List[1.8180209034331518`, 0.008440018725300728`],
List[1.8289944674381577`, 0.01538353194694639`],
List[1.8390785712365771`, 0.020647516279422223`],
List[1.8482787300585155`, 0.024393268036740877`],
List[1.8529669537056004`, 0.02589139337639477`],
List[1.8579867443579592`, 0.02718112978745319`],
List[1.8630295576621545`, 0.028148004455478503`],
List[1.8677318350704086`, 0.02875436483964524`],
```

```
            List[1.8706049904237505`, 0.028986380194225708`],
            List[1.873323538248825`, 0.0291106936040121`],
            List[1.8784157905220895`, 0.02909979399104083`],
            List[1.8834819593167174`, 0.028783911606270977`],
            List[1.8863644484863291`, 0.028473886915831758`],
            List[1.889067323789397`, 0.0281007099146851`],
            List[1.8945097503439552`, 0.027118590924287478`],
            List[1.9003696902891154`, 0.025740332252873793`],
            List[1.9109593009628465`, 0.022514858084424577`],
            List[1.920657051873991`, 0.018892401701557857`],
            List[1.9665514341784773`, -0.0012774939524654253`],
            List[1.9999998333333333`, -0.0122153593511319`]]]],
         List[Rule[PlotRange, Automatic], Rule[AspectRatio,
           Power[GoldenRatio, -1]],
          RuleDelayed[DisplayFunction, $DisplayFunction],
          Rule[ColorOutput, Automatic],
          Rule[Axes, Automatic],
          Rule[AxesOrigin, Automatic],
          Rule[PlotLabel, None],
          Rule[AxesLabel, None],
          Rule[Ticks, Automatic],
          Rule[GridLines, None],
          Rule[Prolog, List[]],
          Rule[Epilog, List[]],
          Rule[AxesStyle, Automatic],
          Rule[Background, Automatic],
          Rule[DefaultColor, Automatic],
          RuleDelayed[
           DefaultFont, $DefaultFont],
          Rule[RotateLabel, True],
          Rule[Frame, False],
          Rule[FrameStyle, Automatic],
          Rule[FrameTicks, Automatic],
          Rule[FrameLabel, None],
          Rule[PlotRegion, Automatic],
          Rule[ImageSize, Automatic],
          RuleDelayed[
           TextStyle, $TextStyle],
          RuleDelayed[FormatType, $FormatType]]]
```

In[188]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb"]
```

Out[188]=

```
StandardForm
```

In[189]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps"]
```

Out[189]=

```
- Graphics -
```

In[190]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps"] // FullForm
```

Out[190]//FullForm=

```
Graphics[List[List[Line[List[List[-1.9999998333333333`, 0.018315651099163666`],
     List[-1.9210073169479194`, 0.024965282686839303`],
     List[-1.8377320337083367`, 0.034140905879298085`],
     List[-1.7532736698276103`, 0.0462372947296657`], List[-1.660764800562505`,
```

```
     0.06340961817674262`], List[-1.5785622264174572`, 0.08275600472773463`],
    List[-1.4906280683541824`, 0.10839513856535575`],
    List[-1.3165568698541965`, 0.176696449796265`],
    List[-1.1488517715417124`, 0.2671725985341265`],
    List[-0.9875127734167299`, 0.3771239077878865`],
    List[-0.8158732088125825`, 0.5139398247464876`],
    List[-0.6505997443959369`, 0.6548952188218141`],
    List[-0.49169238016679306`, 0.7852435168844217`],
    List[-0.4033296982844994`, 0.8498674817137135`],
    List[-0.32248444945848426`, 0.9012287039308049`],
    List[-0.238040189216551`, 0.9449123267840369`],
    List[-0.1596426189376772`, 0.9748362548994524`],
    List[-0.1210479274390275`, 0.9854542262161895`],
    List[-0.1012837918018679`, 0.9897940315067599`],
    List[-0.07990432059860685`, 0.9936356384406656`],
    List[-0.06766226347739546`, 0.9954322820012288`],
    List[-0.056398986936803536`, 0.9968242078026867`],
    List[-0.04584278769357836`, 0.997900645550053`],
    List[-0.03425568918503252`, 0.9988272359843137`],
    List[-0.029232125698399822`, 0.9991458478229799`],
    List[-0.023846829181437094`, 0.9994314904008477`],
    List[-0.019192329184449866`, 0.9996317223313506`],
    List[-0.01412923021916404`, 0.9998003847791838`],
    List[-0.011162534426455067`, 0.9998754055877068`],
    List[-0.008417754493378532`, 0.9999291439196999`],
    List[-0.003007142183362502`, 0.999990957136776`],
    List[0.001909918777716465`, 0.9999963522169157`],
    List[0.007133677874793013`, 0.9999491119348202`],
    List[0.011909983895002052`, 0.9998581623435324`],
    List[0.017134671691070795`, 0.9997064461214131`],
    List[0.02258923333064591`, 0.9994898567048957`],
    List[0.027664022579297086`, 0.9992349946206686`],
    List[0.03676506101645184`, 0.9986492433825143`],
    List[0.046487856778484764`, 0.9978412127124359`],
    List[0.0687153236097172`, 0.9952893344535297`],
    List[0.08891189341574972`, 0.9921258401122844`],
    List[0.1284035999582669`, 0.9836476901737676`],
    List[0.16568627149973345`, 0.9729214394860457`],
    List[0.24974313299960782`, 0.9395336606784316`],
    List[0.327244529916323`, 0.8984457236123577`],
    List[0.4927372546245758`, 0.7844362264404803`],
    List[0.6518638791453268`, 0.6538178279030031`],
    List[0.8212910701452427`, 0.5094013725987232`],
    List[0.9843521609576569`, 0.37948159673203985`],
    List[1.157713818249236`, 0.26176679030019007`],
    List[1.3247093753533132`, 0.17293232791205787`],
    List[1.4853388322698888`, 0.1101148412255609`],
    List[1.5681280856924957`, 0.08551823139009224`],
    List[1.6562688556656293`, 0.06436234497421982`],
    List[1.7463673615481474`, 0.047368446316283896`],
    List[1.831133345353033`, 0.0349775333298323`],
    List[1.91562847179078`, 0.025485835749798215`],
    List[1.9932656346654332`, 0.01881486762938725`],
    List[1.9999998333333333`, 0.018315651099163666`]]]]],
 List[Rule[PlotRange, Automatic], Rule[AspectRatio,
   Power[GoldenRatio, -1]],
  RuleDelayed[DisplayFunction, $DisplayFunction],
  Rule[ColorOutput, Automatic],
```

```
        Rule[Axes, Automatic],
        Rule[AxesOrigin, Automatic],
        Rule[PlotLabel, None],
        Rule[AxesLabel, None],
        Rule[Ticks, Automatic],
        Rule[GridLines, None],
        Rule[Prolog, List[]],
        Rule[Epilog, List[]],
        Rule[AxesStyle, Automatic],
        Rule[Background, Automatic],
        Rule[DefaultColor, Automatic],
        RuleDelayed[DefaultFont, $DefaultFont],
        Rule[RotateLabel, True],
        Rule[Frame, False],
        Rule[FrameStyle, Automatic],
        Rule[FrameTicks, Automatic],
        Rule[FrameLabel, None],
        Rule[PlotRegion, Automatic],
        Rule[ImageSize, Automatic],
        RuleDelayed[TextStyle, $TextStyle],
        RuleDelayed[FormatType, $FormatType]]
```

In[191]:=

```
Remove[exp1]
```

In[192]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps",exp1]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.ps is not encoded. Mehr…
```

Out[192]=

```
- Graphics -
```

In[193]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps",exp1]; Show[exp1]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.ps is not encoded. Mehr…

Show::gtype : Symbol is not a type of graphics. Mehr…
```

Out[193]=

```
Show[exp1]
```

In[194]:=

```
Get["C:\\work\\MathematicaData\\postscri.ma"]
```

Out[194]=

```
StandardForm
```

In[195]:=

```
Get["C:\\work\\MathematicaData\\postscri.ma"] // FullForm
```

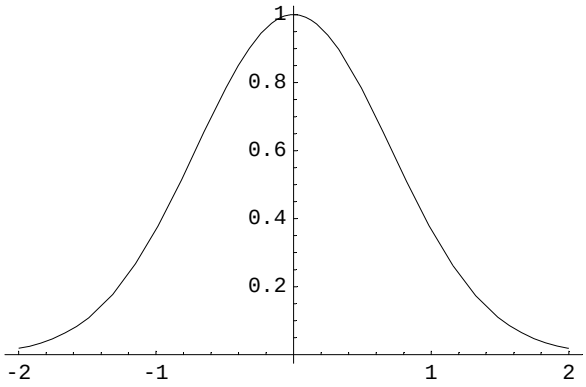
Out[195]//FullForm=

```
StandardForm
```

In[196]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ma",exp1]; Show[exp1]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.ma is not encoded. Mehr…
```



Out[196]=

```
- Graphics -
```

In[197]:=

```
Remove[expCos, exp1]
```

In[198]:=

```
Get["C:\\work\\MathematicaData\\postscri.nb"]
```

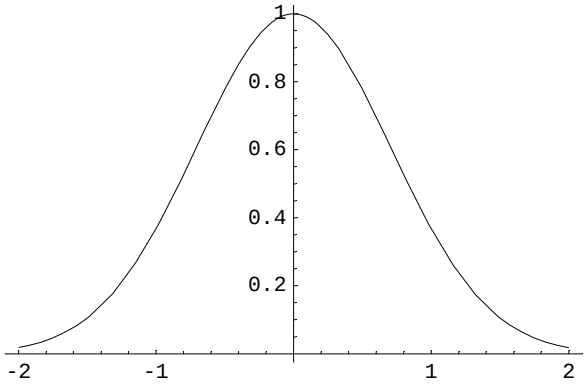
Out[198]=

```
StandardForm
```

In[199]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb",exp1]; Show[exp1]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.nb is not encoded. Mehr…
```



Out[199]=

```
- Graphics -
```

In[200]:=

```
exp1
```

Out[200]=

```
- Graphics -
```

4. 14. Experimentieren mit Animationen
▮ Expérimenter les animations

4.14.1. Nachfolgend zwei Beispiele zu Animationen (Bsp. von U.Altmann)
▮ Ensuite deux exemples d'animations (exemple pris de U.Altmann)

Achtung! Die Rechnungen dauern vielleicht etwas Zeit!
Vorschlag: Erst am Schluss der Arbeit ausführen!

▮

Attention! Il faut du temps aux calculs! Proposition:
Executer à la fin du travail!

Package laden:

▮ Charger Package:

In[201]:=
```
?ImplicitPlot
```
```
Information::notfound : Symbol ImplicitPlot not found. Mehr…
```

In[202]:=
```
Remove[ImplicitPlot]
```

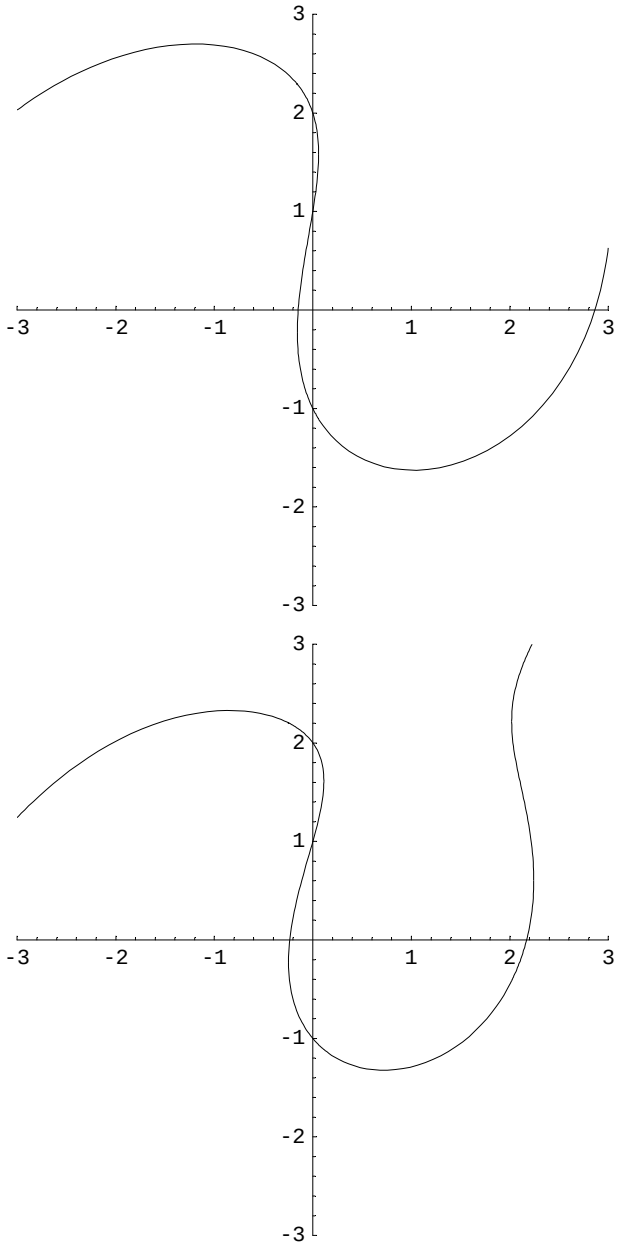
In[203]:=
```
<< Graphics`ImplicitPlot`
```

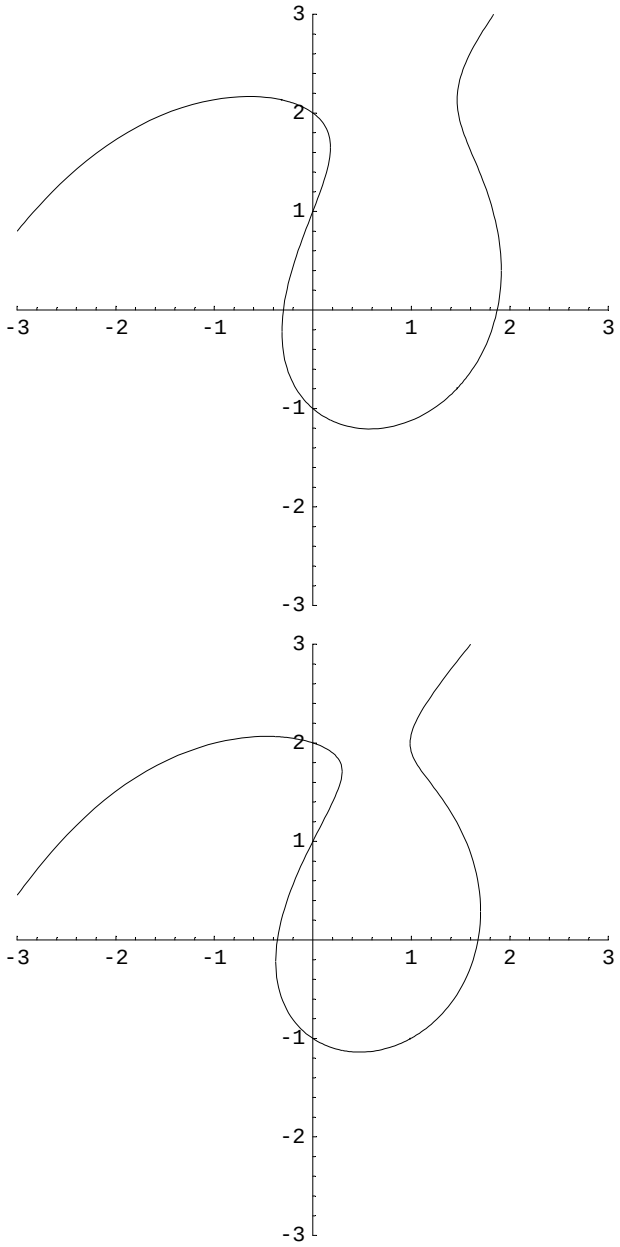
In[204]:=
```
(* Old: *)
(* Needs["Graphics`ImplicitPlot`"] *)
```

Beispiel: Zuerst n = 128 Graphiken rechnen (das dauert!!!). Um Ueberraschungen zu vermeiden, ist n auf 10 eingestellt. Das geht zwar schneller, aber die Animation wird so nicht gut. Sonst kann es schon eine Lektion dauern!

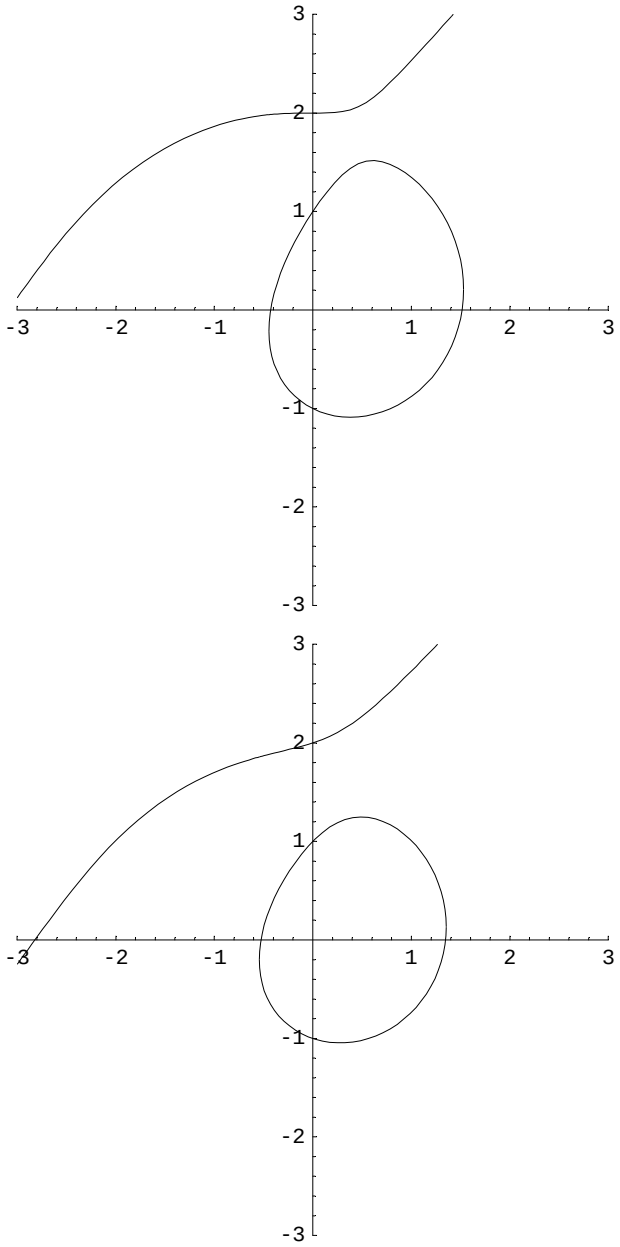
▮ Exemple: Calculer d'abord n = 128 graphiques (cela dure!!!). Pour éviter des surprises, il faut mettre n sur 10. Cela va plus vite, mais l'animation n'est pas si bonne. Si non cela peut durer toute une leçon.

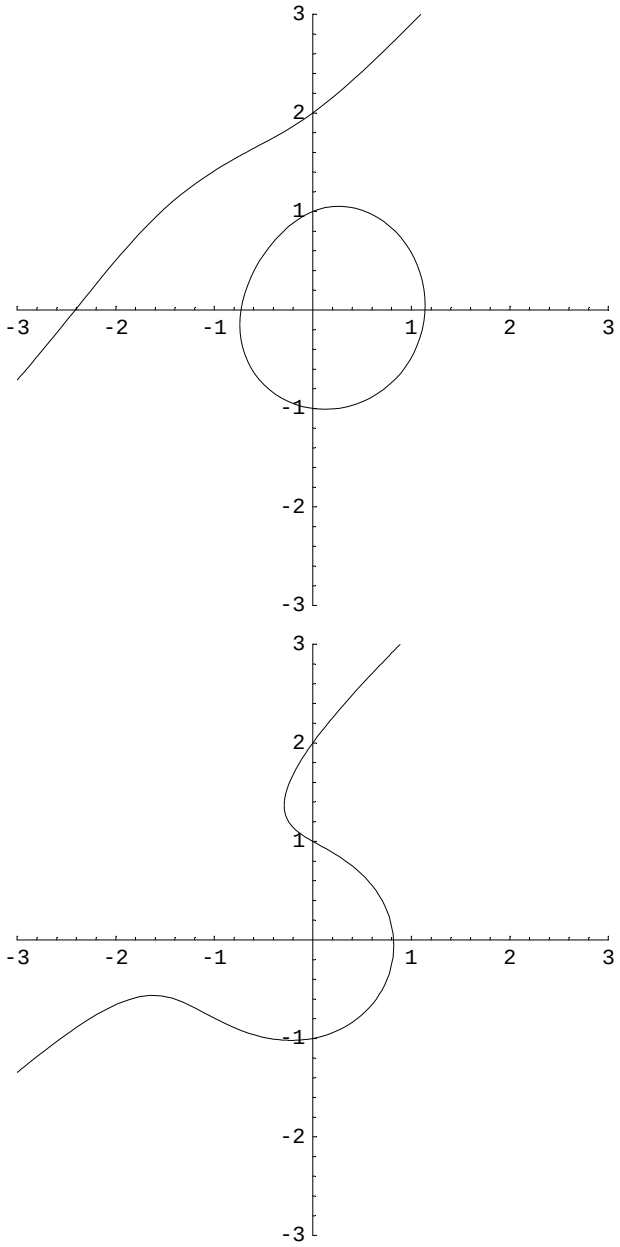
In[206]:=
```
a=3;
(* n=128; *)
n=10;
tabelle1=Table[
  ImplicitPlot[(x-y+2)(x^2+y^2-1) ==
    N[3.(1.-Tan[b])] x, {x,-a,a},
    PlotRange->{{-a,a},{-a,a}}, AspectRatio->1.],
    {b,-Pi/2+Pi/n,Pi/2-Pi/n,Pi/n}];
Prepend[tabelle1,
  ParametricPlot[{0.,t},{t,-a,a},
    PlotRange->{{-a,a},{-a,a}}, AspectRatio->1.]];
```

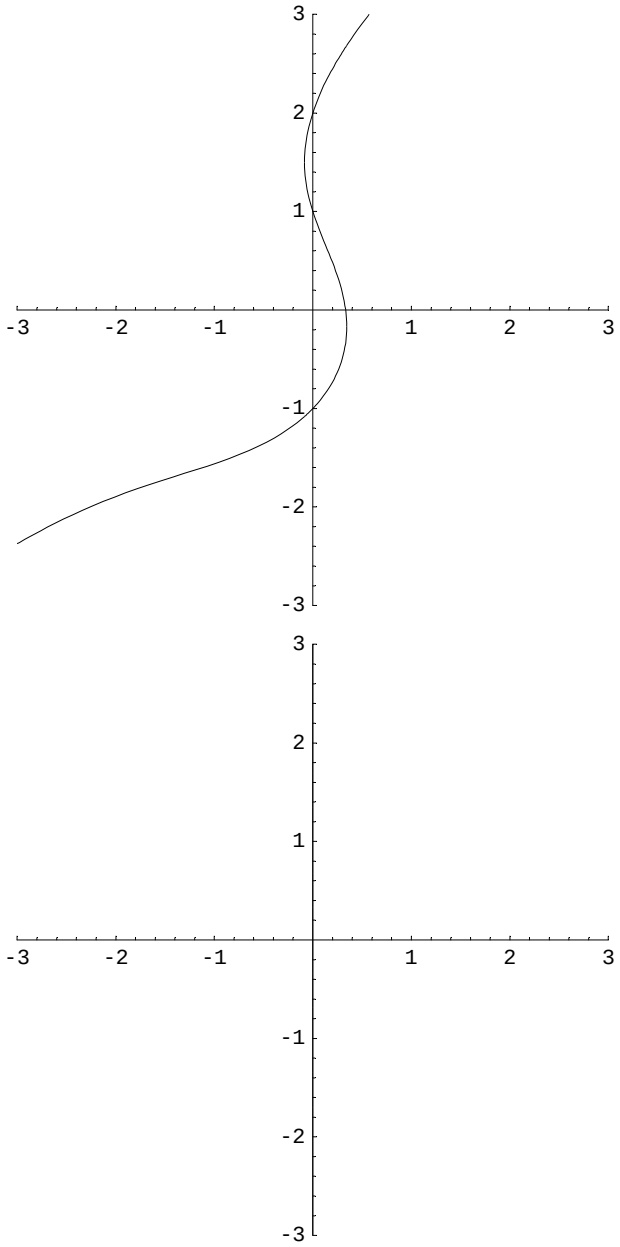

3
2
1
-3
-2
-1
1
2
3
-1
-2
-3
3
2
1
-3
-2
-1
1
2
3
-1
-2
-3


3
2
1
-3
-2
-1
1
2
3
-1
-2
-3

3
2
1
-3
-2
-1
1
2
3
-1
-2
-3


3
2
1
-3
-2
-1
1
2
3
-1
-2
-3
3
2
1
-3
-2
-1
1
2
3
-1
-2
-3


3
2
1
-1
-2
-3
-3
-2
-1
1
2
3
3
2
1
-1
-2
-3
-3
-2
-1
1
2
3



Dann animieren!

■ Ensuite animer!

2. Beispiel:

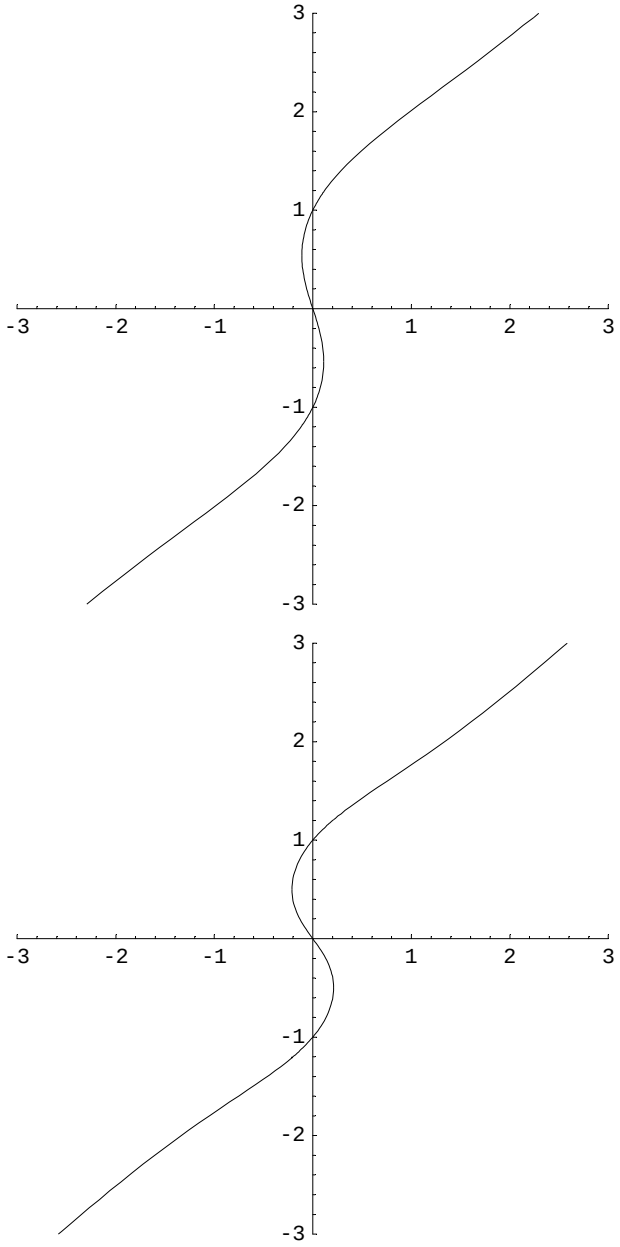
Zuerst n = 128 Graphiken rechnen. (Um Ueberraschungen zu vermeiden, ist n auf 10 eingestellt. Das geht zwar schneller, aber die Animation wird so nicht gut.)

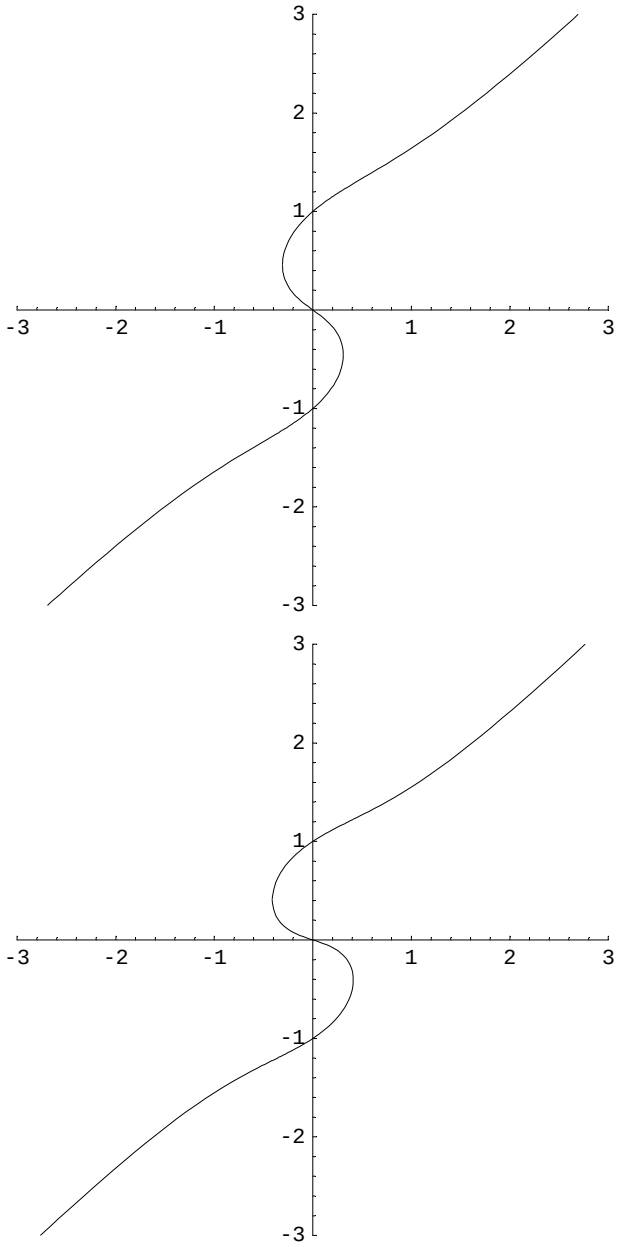
■ 2e exemple:

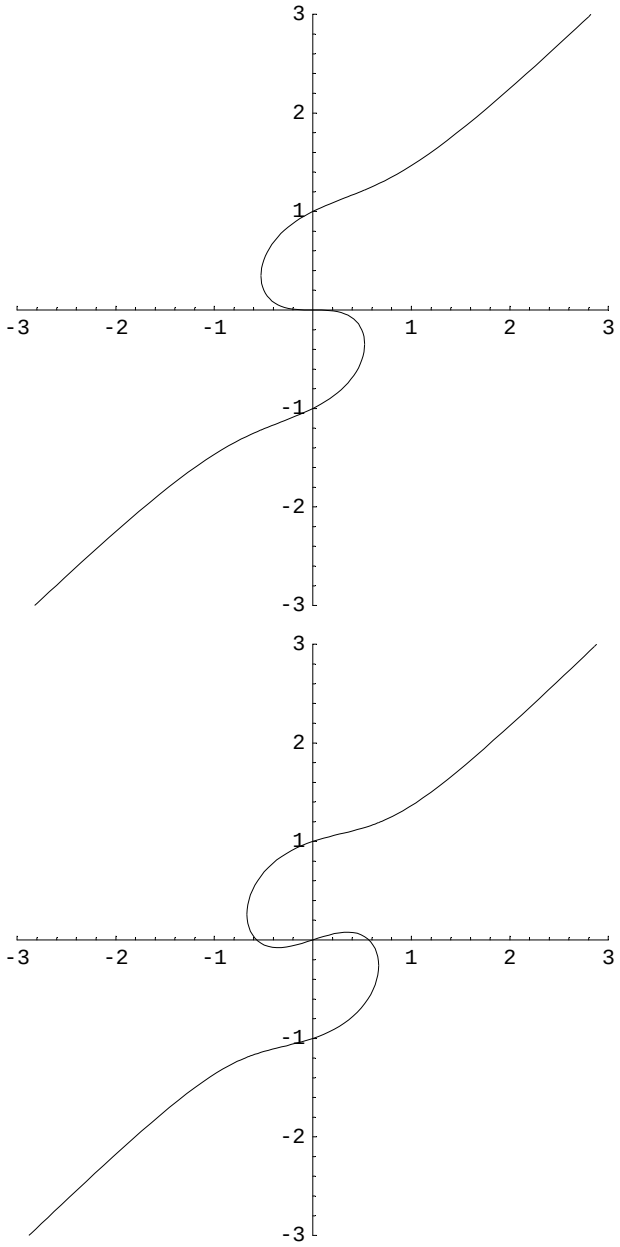
D'abord calculer n = 128 graphiques. (Pour éviter des surprises, il faut mettre n sur 10. Cela va plus vite, mais l'animation n'est pas si bonne.)

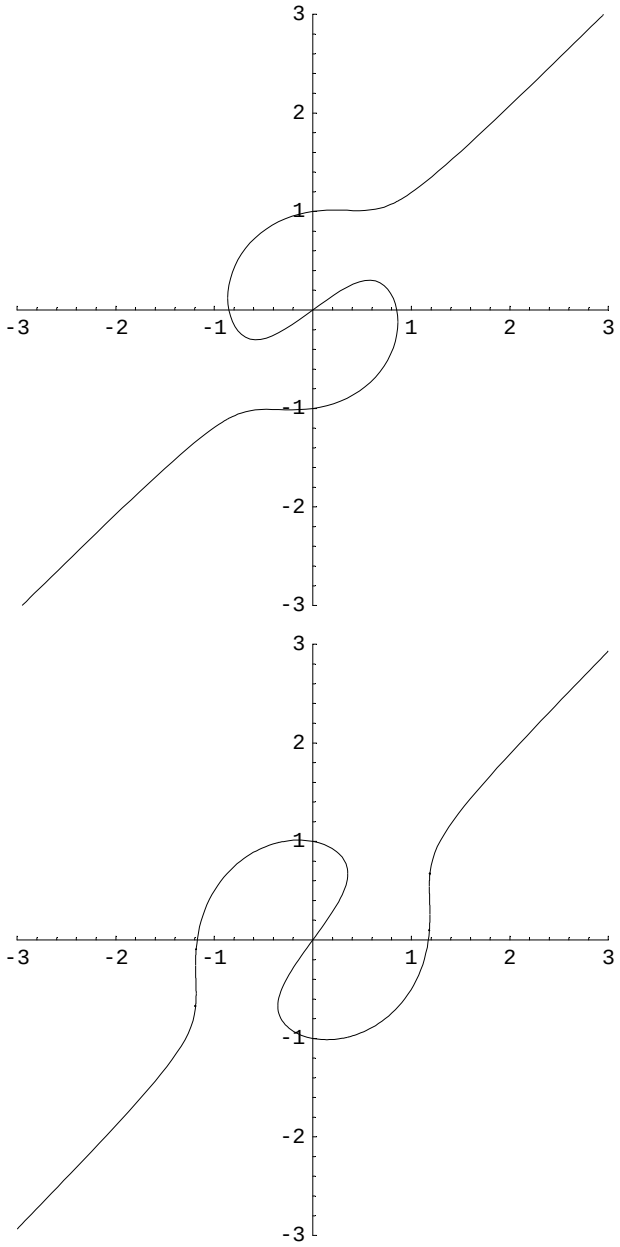
In[211]:=

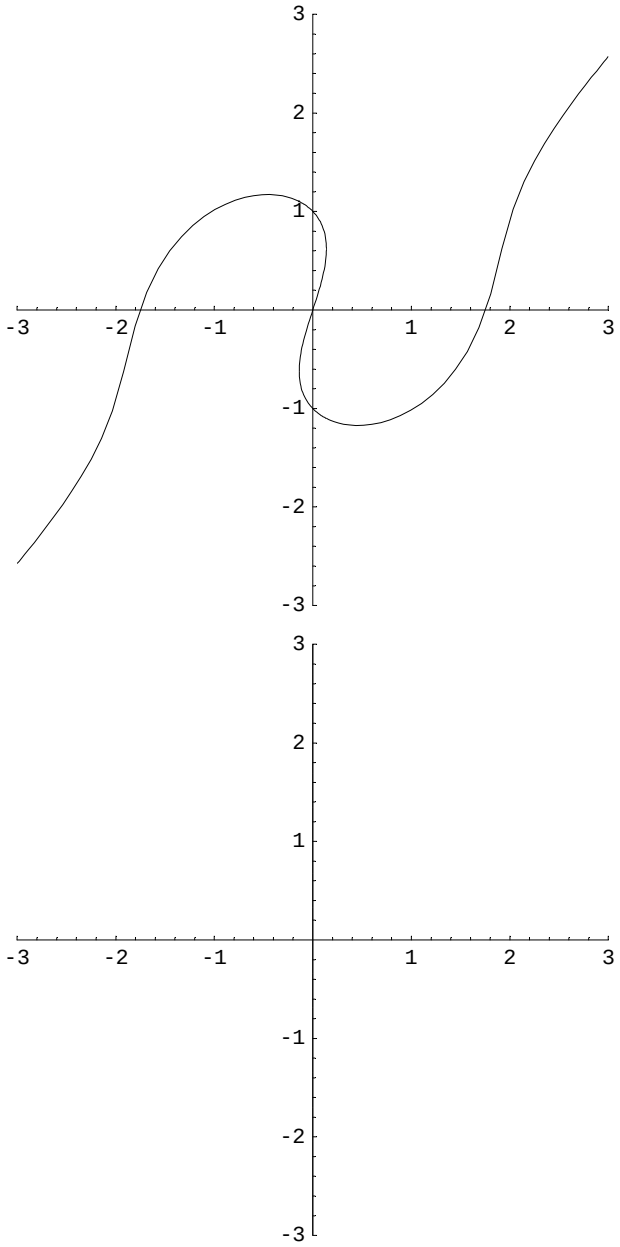
```
a=3;
(* n=128; *)
n=10;
tabelle2=Table[
  ImplicitPlot[(x-y)(x^2+y^2-1)==N[Tan[b]-1] x,
    {x,-a,a}, PlotRange->{{-a,a},{-a,a}},
    AspectRatio->1.],{b,-Pi/2+Pi/n,Pi/2-Pi/n,Pi/n}];
Prepend[tabelle2,
  ParametricPlot[{0.,t},{t,-a,a}, PlotRange->{{-a,a},
  {-a,a}},AspectRatio->1.]];
```




3
2
1
-1
-2
-3
-3
-2
-1
1
2
3
3
2
1
-1
-2
-3
-3
-2
-1
1
2
3


3
2
1
-1
-2
-3
-3
-2
-1
1
2
3
3
2
1
-1
-2
-3
-3
-2
-1
1
2
3


3
2
1
-1
-2
-3
-3
-2
-1
1
2
3
3
2
1
-1
-2
-3
-3
-2
-1
1
2
3



Dann animieren!

Ensuite animer!

4.14.2. Noch ein Beispiel (Simulation einer Welle)
Encore un exemple (simulation d'une onde)

Erst rechnen, dann animieren!

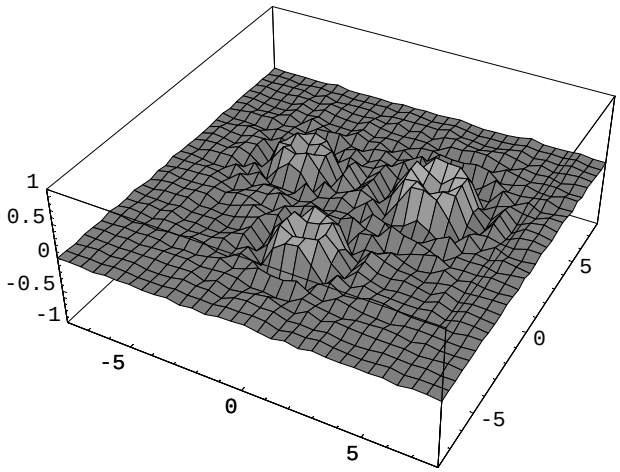
D'abord calculer, ensuite animer!)

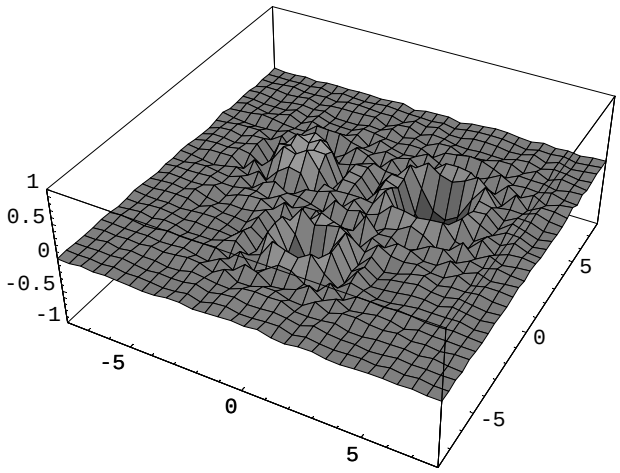
(Um Ueberraschungen zu vermeiden, sind statt 60 nur 5 Schritte eingestellt.
Das geht zwar schneller, aber die Animation wird so nicht gut.)

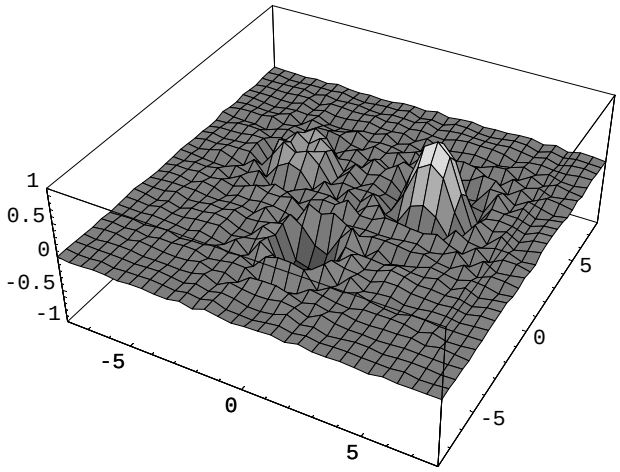
(Pour éviter des surprises, il faut 5 au lieu de 60 étapes. Cela va plus vite, mais l'animation réussit moins bien.)

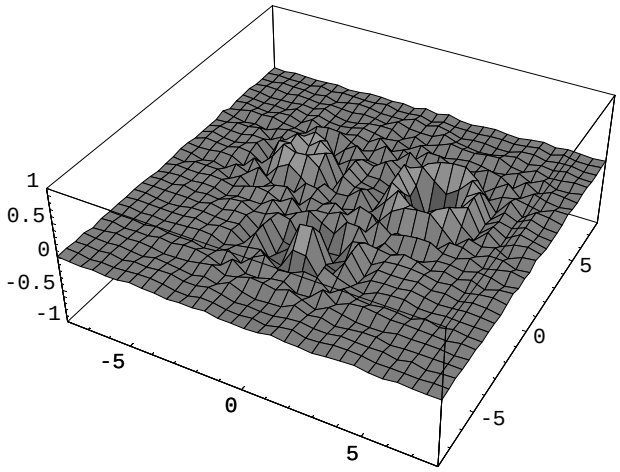
In[216]:=

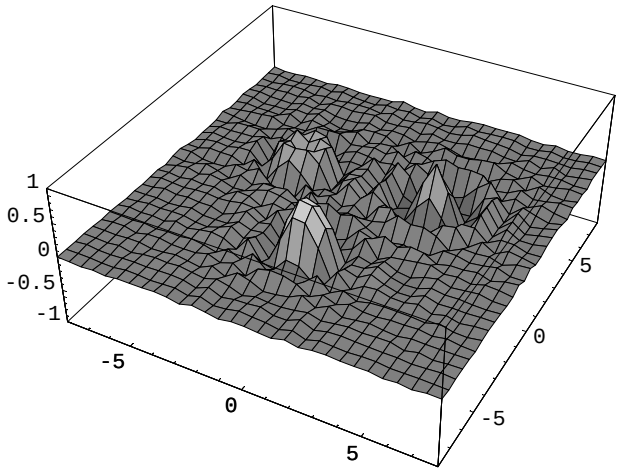
```
Table[Plot3D[
    Sin[3t+(x-3)^2+(y-2)^2]/(1+(x-3)^2+(y-2)^2)+
    3/5Sin[5/3(3t+(x+3)^2+(y-2)^2)]/(1+(x+3)^2+(y-2)^2)+
    3/4Sin[4/3(3t+x^2+(y+3)^2)]/(1+x^2+(y+3)^2),
    {x,-8,8},{y,-8,8},
PlotRange->{{-8,8},{-8,8},{-1,1}},
PlotPoints->30],
(* {t,0,2Pi,2Pi/60}];  *)
{t,0,2Pi,2Pi/5}];
```

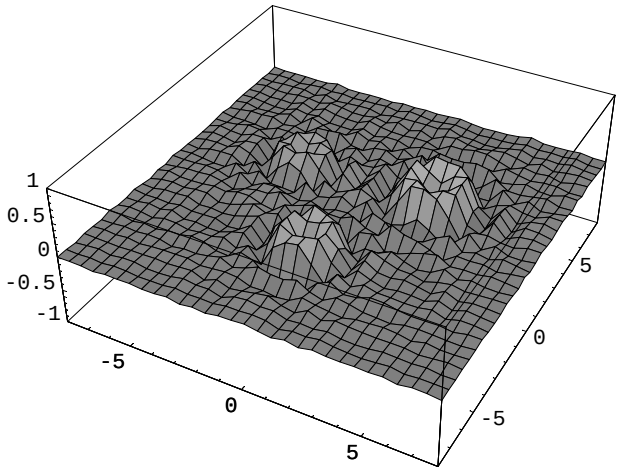





1
0.5
0
-0.5
-1
-5
0
5
5
0
-5


1
0.5
0
-0.5
-1
-5
0
5
5
0
-5


1
0.5
0
-0.5
-1
-5
0
5
5
0
-5



Wähle die Zelle an, die alle ausgegebenen Bilder umschliesst. Animiere mit "Animate Selected Graphics ...y" in Menu ("Graph") neu "Cell". Beobachte, welche Zonen sich bewegen und welche Zonen ruhig bleiben. Gibt es Interferenzen?
■ Choisir la cellule qui inclut toutes les images sorties. Anime par "Animate Selected Graphics ...y" dans le menu ("Graph") nouveau "Cell". Observe quelles zones restent tranquilles. Y a-t-il des interférences?

4.15. Probiere eigene Beispiele aus!
■ Essaie de propres exemples!

"Putzmaschine" einsetzen
■ Employer la "machine de nettoyage"

In[217]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[218]:=
```
Remove["Global`*"]
```

Uebungen ▮ Exercices

4. Graphiken
▮ Graphiques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 4 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 4.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

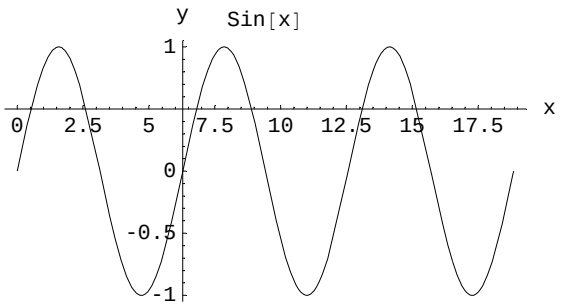
Aufgabe 1 ▮ Problème 1

Spiele ein wenig mit den Options
▮ Joue un peu avec les options

Sinus ohne ";"

▮ Sinus sans ";"

```
In[1]:= Plot[Sin[x],{x,0,6 Pi},
            AspectRatio -> 1/2,
            PlotLabel -> "Sin[x]",
            AxesLabel -> {"x", "y"},
            AxesOrigin -> {2 Pi, 1/2} ]
```

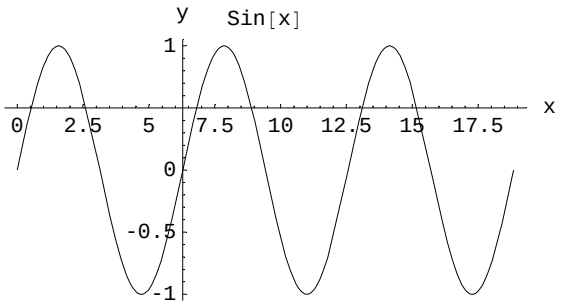



```
Out[1]= - Graphics -
```

Sinus mit ";" --- Unterschied?

■ Sinus avec ";" --- y a-t-il une différence?

```
In[3]:= Plot[Sin[x],{x,0,6 Pi},
            AspectRatio -> 1/2,
            PlotLabel -> "Sin[x]",
            AxesLabel -> {"x", "y"},
            AxesOrigin -> {2 Pi, 1/2} ];
```



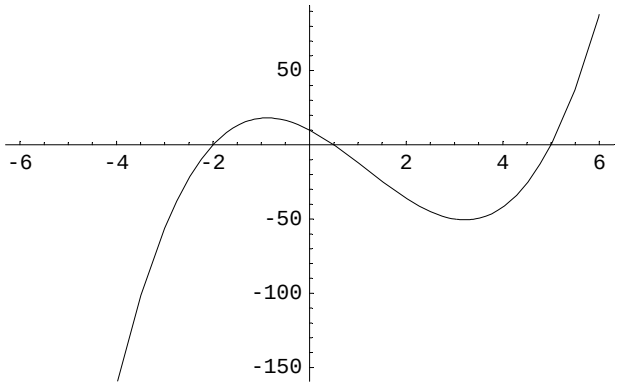
Aufgabe 2 ■ Problème 2

(a) Nullstellen schätzen
■ Evaluer les zéros (estimer...)

Schätze die Nullstellen mit Hilfe des Plots:

■ Evalue les zéros à l'aide de "Plots":

In[5]:= **Plot[2x^3-7x^2-17x+10,{x,-6,6}];**



Verifiziere die Schätzung mit "Solve":
Vérifier l'évaluation par "Solve":

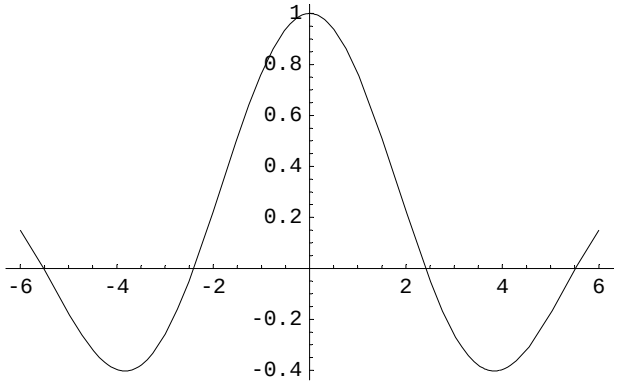
In[6]:= **Solve[2x^3-7x^2-17x+10==0,x]**

Out[6]= $\left\{\{x \to -2\}, \left\{x \to \frac{1}{2}\right\}, \{x \to 5\}\right\}$

(a) Nullstellen schätzen aus Graph:
■ Evaluer les zéros d'un graphe:

Schätze die Nullstellen mit Hilfe des Plots auf 3 Stellen (herauszoomen):
■ Evalue les zéros à l'aide des "Plots" (3 places, zoom):

In[7]:= **Plot[BesselJ[0,x],{x,-6,6}];**



Verifiziere die Schätzung mit "Solve":
■ Vérifie l'évaluation par "Solve":

In[8]:= **Solve[2x^3-7x^2-17x+10==0,x]**

Out[8]= $\left\{\{x \to -2\}, \left\{x \to \frac{1}{2}\right\}, \{x \to 5\}\right\}$

Aufgabe 3 ▮ Problème 3

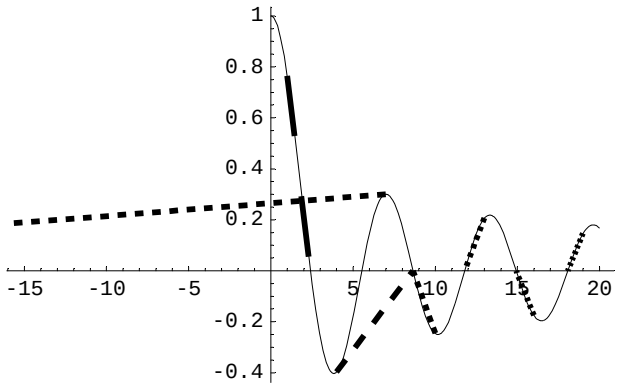
Verschiedene Dinge in einem Graphen: ▮ Plusieurs choses dans un graphe:

Zeichne den Graphen der Besselfunktion J0 sowie Liniensegmente, die vom Startpunkt der Newton-Iteration für die Nullstellen schliesslich zur Nullstelle führen:

▮ Dessine le graphe de la fonction d'après Bessel J0 ainsi que des segments de lignes, qui vont du point de départ de l'itération d'après Newton pour les zéros au zéro:

```
In[9]:= Remove[t, x, y]
```

```
In[10]:= Show[
             Plot[BesselJ[0,x],{x,0,20},
                 DisplayFunction -> Identity],
             Table[Graphics[{Dashing[{1/(10 t)}],
                             Thickness[0.01],
                             Evaluate[Line[{
                                 {t,BesselJ[0,t]},
                                 Evaluate[{y/. FindRoot[BesselJ[0,y]==0,
                                                        {y,t}],0}]
                                 }]]
                             }],
                       {t,1,20,3}],
                       DisplayFunction :> $DisplayFunction
                       ];
```



Aufgabe 4 ▮ Problème 4

(a) Plot einer Funktion und ihrer Potenzreihe: ▮ Plot d'une fonction et sa série de puissances:

f definieren:

▮ Définir f:

In[11]:=
```
Clear[f]; f[x_]:=1/(1 + Sin[Sqrt[Pi^2 + x]^2]); f[x]
```

Out[11]=
$$\frac{1}{1+\text{Sin}[\pi^2+x]}$$

Verwende Series und Normal: Achtung: Viel Output!

■ Utilise "Series" et "Normal": Attention: Beaucoup d'Output!

In[12]:=
```
n[x_]:=Normal[Series[f[x],{x,0,8}]];
nTest[x_]:=Normal[Series[f[x],{x,0,3}]];
nTest[x]
```

Out[14]=
$$-\frac{x\,\text{Cos}[\pi^2]}{(1+\text{Sin}[\pi^2])^2}+\frac{1}{1+\text{Sin}[\pi^2]}+x^3\left(-\frac{\text{Cos}[\pi^2]^3}{(1+\text{Sin}[\pi^2])^4}-\frac{\text{Cos}[\pi^2]\,\text{Sin}[\pi^2]}{(1+\text{Sin}[\pi^2])^3}+\frac{\text{Cos}[\pi^2]}{6\,(1+\text{Sin}[\pi^2])^2}\right)+x^2\left(\frac{\text{Cos}[\pi^2]^2}{(1+\text{Sin}[\pi^2])^3}+\frac{\text{Sin}[\pi^2]}{2\,(1+\text{Sin}[\pi^2])^2}\right)$$

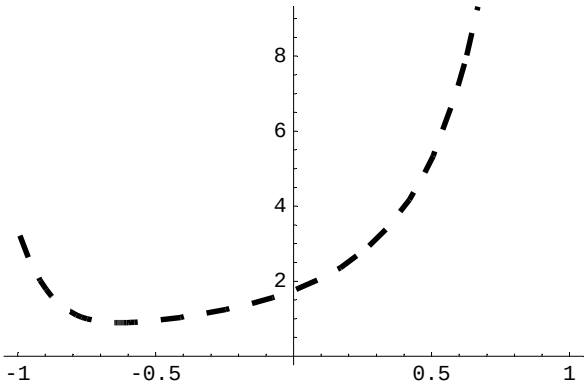
Plot von f und n:

■ Plot de f et n:

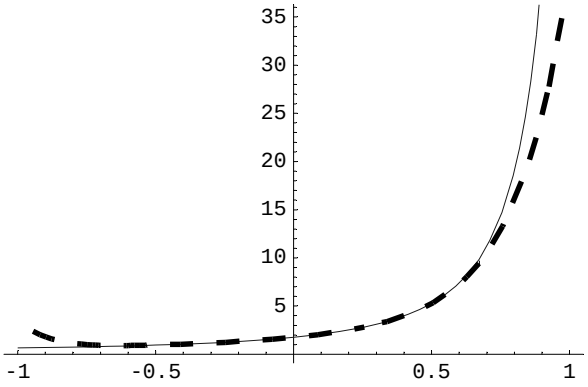
In[15]:=
```
Print[N[n[x]]]
```

$$1.75531+2.78129\,x+3.74403\,x^2+4.41848\,x^3+4.90786\,x^4+5.22979\,x^5+5.41896\,x^6+5.5002\,x^7+5.49547\,x^8$$

In[16]:=
```
a = 1;
Plot[Evaluate[N[n[x]]],Evaluate[{x,-a,a}],
                PlotStyle->{{Thickness[0.01],
        Dashing[{0.04}]}}];
```



```
In[18]:= a = 1; (*PlotRange*)
         Show[
             Plot[f[x],Evaluate[{x,-a,a}],
                 DisplayFunction -> Identity,
                 PlotStyle->{GrayLevel[0.1]}],
             Plot[Evaluate[N[n[x]]],Evaluate[{x,-a,a}],
                 DisplayFunction -> Identity,
                 PlotStyle->{{Thickness[0.01],
                 Dashing[{0.04}]}}],
                 DisplayFunction :> $DisplayFunction];
```



Aufgabe 5 ▮ Problème 5

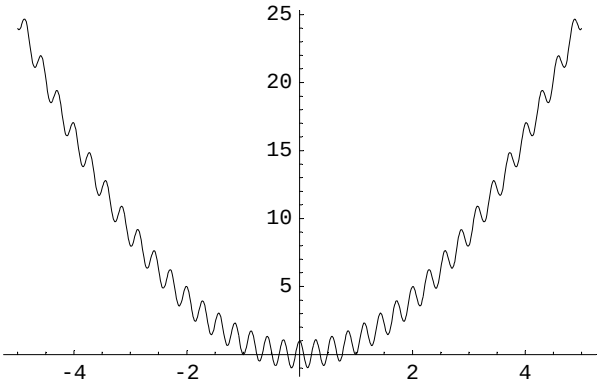
(a) Probleme mit PlotPoints
▮ Problèmes avec Plotpoints

x^2+cos(22x) :

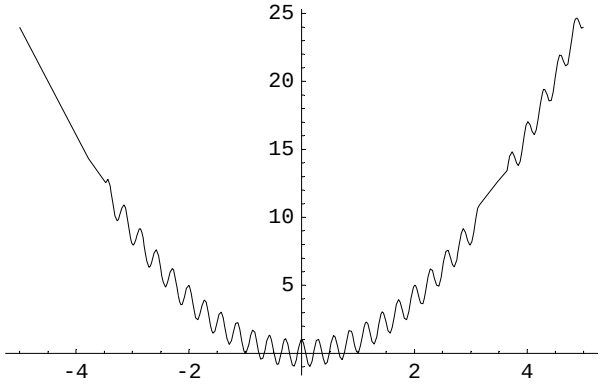
Fehler beim Plotten bei normal vielen PlotPoints (in alten Versionen):

▮ Erreurs qui apparaissent en plotant avec un nombre normal de Plotpoints (vieilles vérsions):

```
In[22]:= Plot[x^2 + Cos[22x], {x,-5,5}];
```



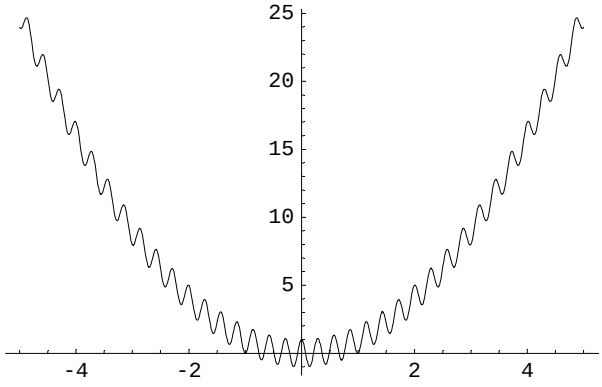
In[23]:=
```
Plot[x^2 + Cos[22 x], {x, -5, 5}, PlotPoints -> 9];
```



(b) Korrektur mit PlotPoints

x^2+cos(22x) :
Wähle 70 PlotPoints:
■ Choisir 70 "PlotPoints":

In[24]:=
```
Plot[x^2 + Cos[22 x], {x, -5, 5}, PlotPoints -> 70];
```

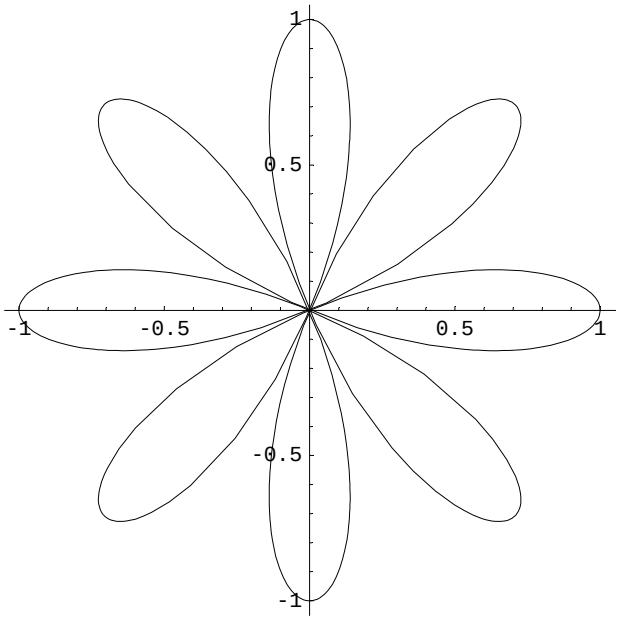


Aufgabe 6 ▮ Problème 6

Einige ParametricPlots:
▮ Quelques "ParametricPlots":

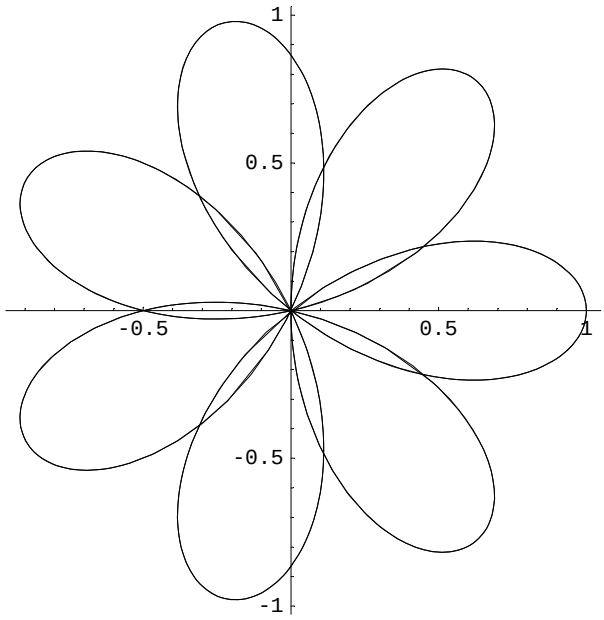
(a)

```
In[25]:= ParametricPlot[{Cos[4t] Cos[t],Cos[4t] Sin[t]},{t,0,2 Pi},
         AspectRatio->Automatic];
```



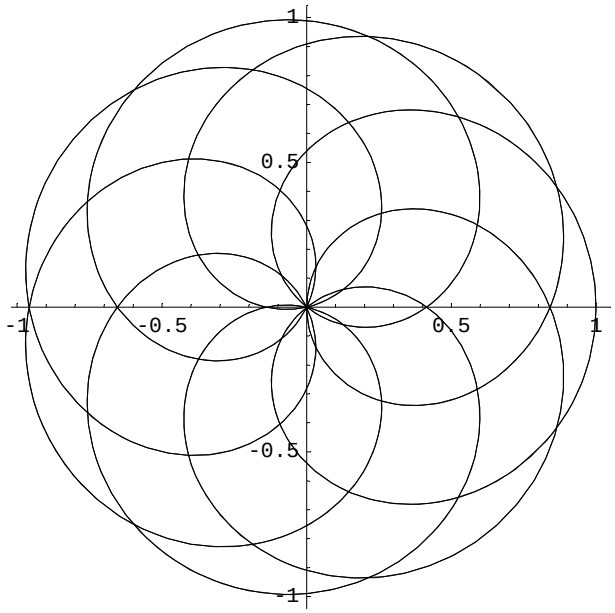
(b)

In[26]:=
```
ParametricPlot[{Cos[7t] Cos[3t],Cos[7t] Sin[3t]},{t,0,2 Pi},
AspectRatio->Automatic];
```



(c)

In[27]:=
```
ParametricPlot[{Cos[7t] Cos[11t],Cos[7t] Sin[11t]},{t,0,2 Pi},
AspectRatio->Automatic];
```



Aufgabe 7 ▮ Problème 7

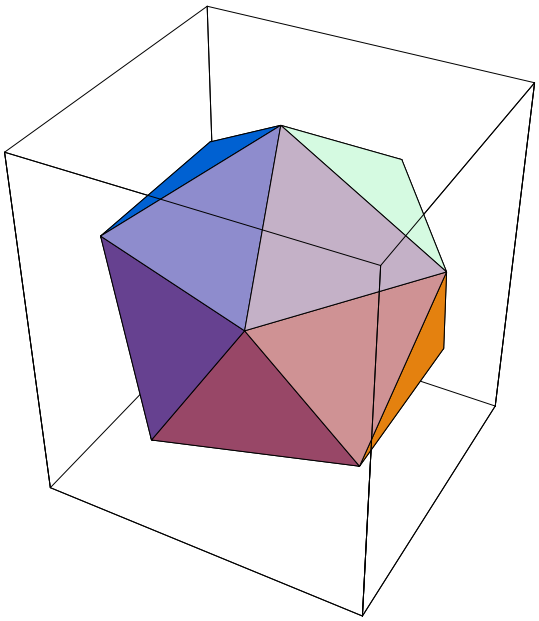
Spiel mit dem Package "Polyhedra.m"
▮ Jeu avec le Package "Polyhedra.m"

Einbinden: ▮ Lier:

```
In[28]:= Needs["Graphics`Polyhedra`"];
```

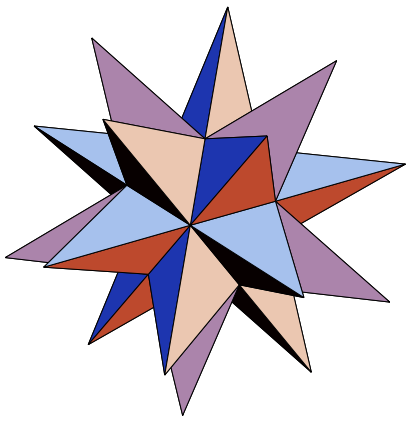
Graphik ansehen:
▮ Regarder le graphique:

```
In[29]:= Show[Graphics3D[Icosahedron[]]];
```



Graphik ansehen:
▮ Regarder le graphique:

```
In[30]:= Show[Graphics3D[Stellate[Icosahedron[],3]],
           Boxed->False];
```

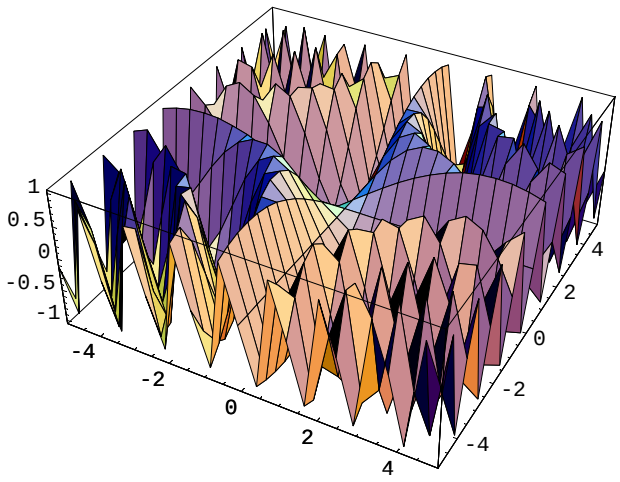


Aufgabe 8 ■ Problème 8

Lighting

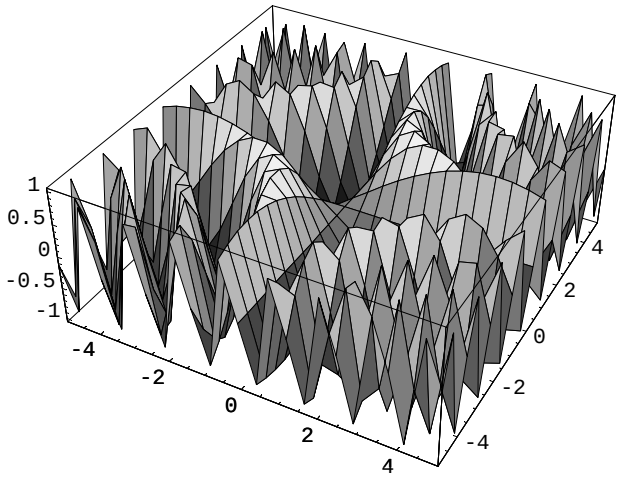
Normal: ■ Normal:

```
In[31]:= Plot3D[Sin[x y],{x,-5,5},{y,-5,5}];
```



Lighting false: ■ Lighting false:

In[32]:= **Plot3D[Sin[x y],{x,-5,5},{y,-5,5},Lighting->False];**

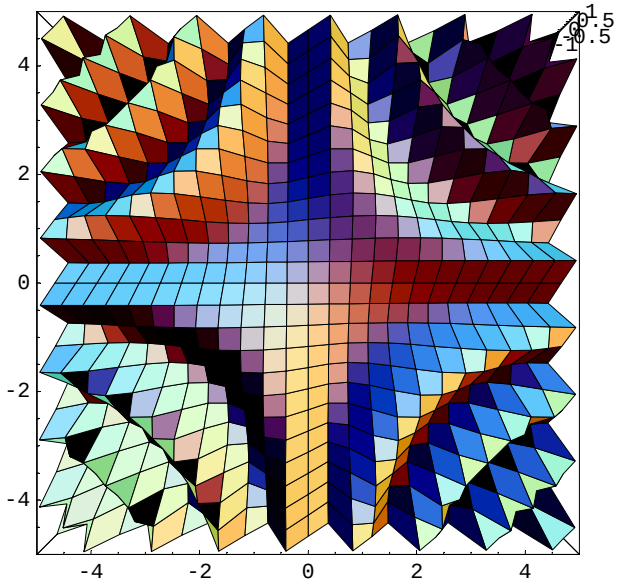


Aufgabe 9 ■ Problème 9

ViewPoint

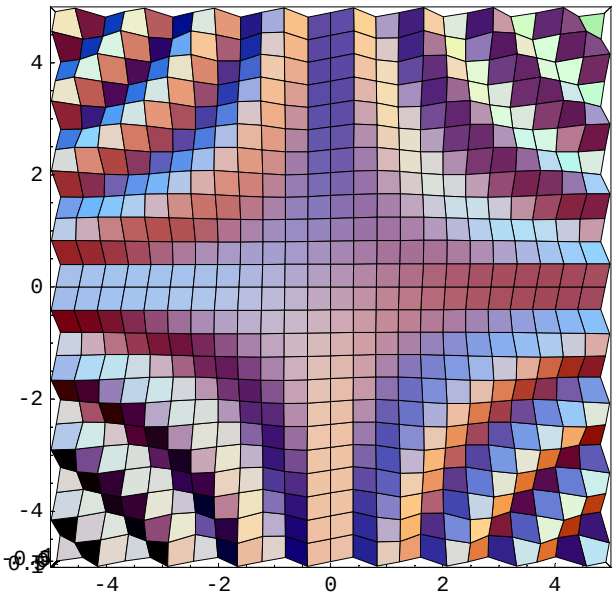
Normal: ■ Normal:

In[33]:= **Plot3D[Sin[x y],{x,-5,5},{y,-5,5},Lighting->True,**
ViewPoint->{0,0,3}];



Anders: ■ Autrement:

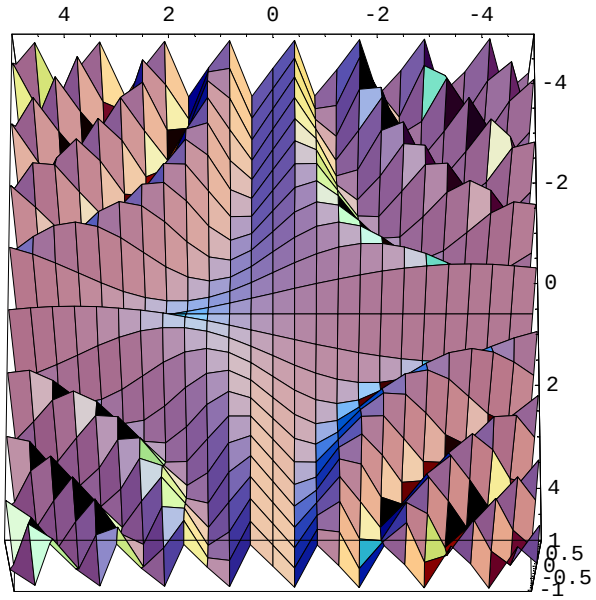
```
In[34]:= Plot3D[Sin[x y],{x,-5,5},{y,-5,5},Lighting->True,
           ViewPoint->{0,0,10}];
```



Nochmals anders:

- Encore autrement:

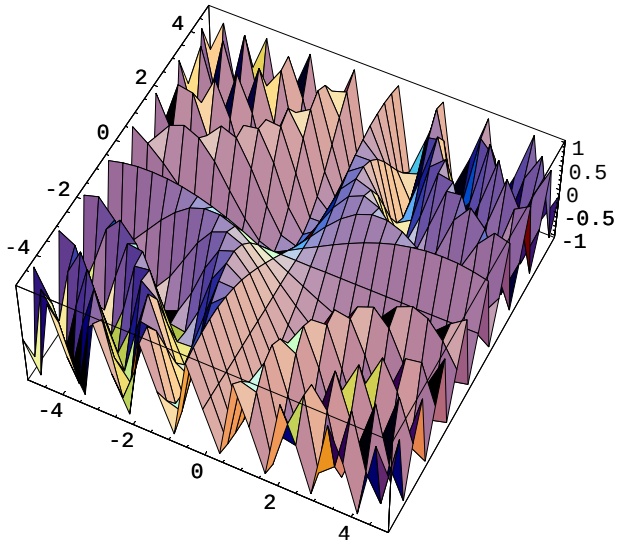
```
In[35]:= Plot3D[Sin[x y],{x,-5,5},{y,-5,5},Lighting->True,
           ViewPoint->{0,3,10}];
```



Oder so:

- Ou ainsi:

In[36]:=
```
Plot3D[Sin[x y],{x,-5,5},{y,-5,5},Lighting->True,
       ViewPoint->{2,-4,6}];
```

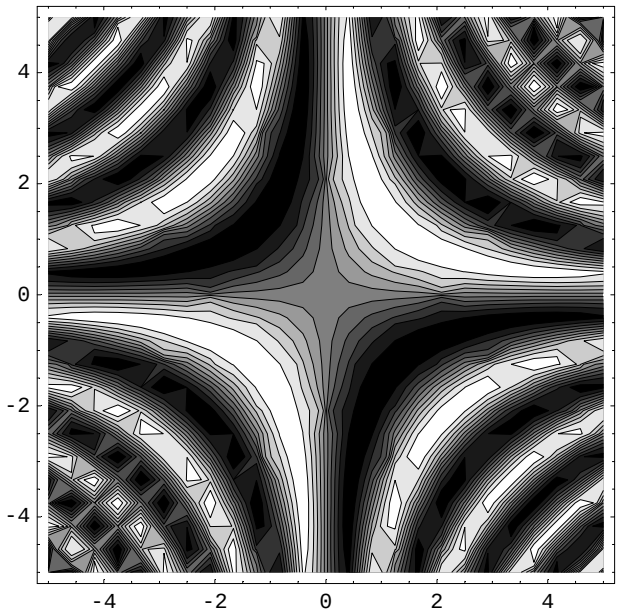


Aufgabe 10 ■ Problème 10

Contour- und DensityPlot
■ Contour- et DensityPlot

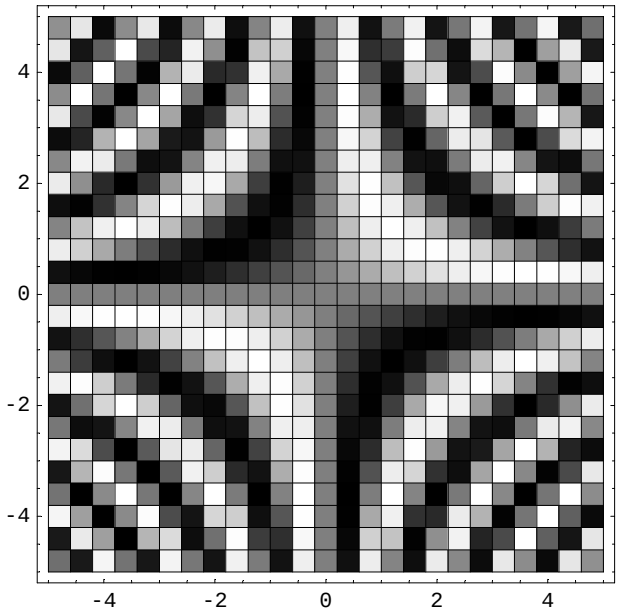
Contour: ■ Contour:

In[37]:=
```
ContourPlot[Sin[x y],{x,-5,5},{y,-5,5}];
```



Density: ■ Density:

In[38]:=
```
DensityPlot[Sin[x y],{x,-5,5},{y,-5,5}];
```

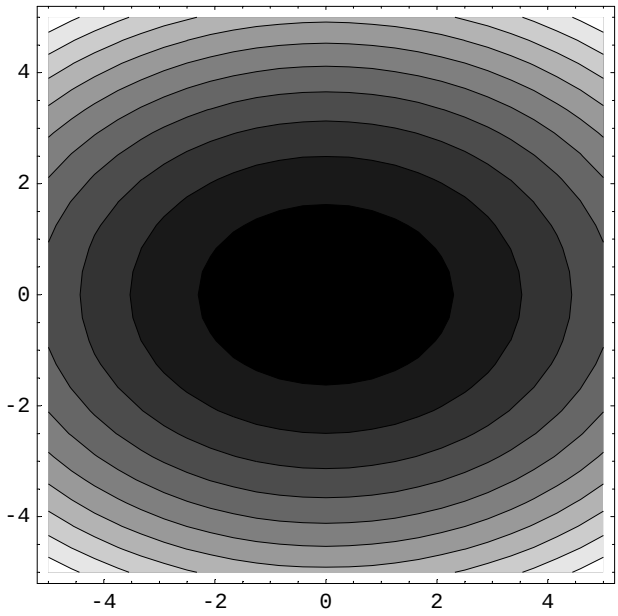


Aufgabe 11 ▮ Problème 11

Mit ContourPlot zu Ellipsen:
▮ Faire des ellipses avec ContourPlot

Contour: ▮ Contour:

In[39]:=
```
ContourPlot[x^2 + 2 y^2,{x,-5,5},{y,-5,5},
 AspectRatio->Automatic];
```



Aufgabe 12 ▮ Problème 12

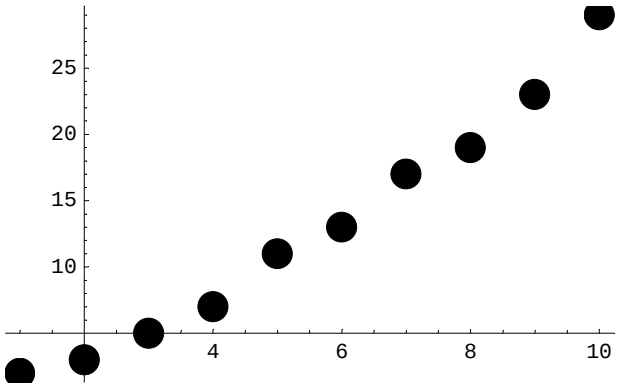
Spiel mit Listen:
▮ Jouer avec des listes

Primzahlen generieren:
▮ Générer des nombres premiers:

```
In[40]:= t = Table[Prime[n],{n,10}]
Out[40]= {2, 3, 5, 7, 11, 13, 17, 19, 23, 29}

In[41]:=
```

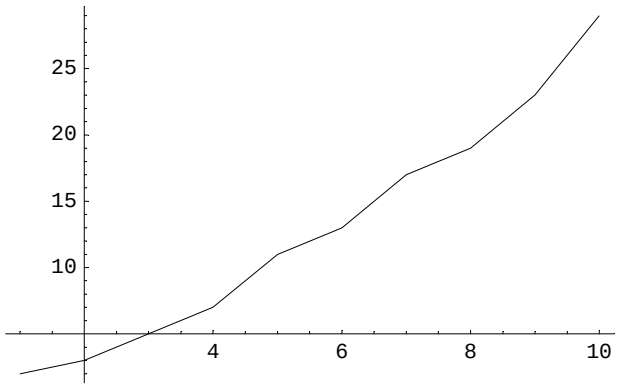
Plot: ▮ Plot:

```
In[41]:= ListPlot[t,PlotStyle->{PointSize[0.05]}];
```



Punkte verbinden:
▮ Relier des points:

```
In[42]:= ListPlot[t,PlotStyle->{PointSize[0.05]},
                  PlotJoined->True];
```



Fit: ▮ Fit:

In[43]:= **?Fit**

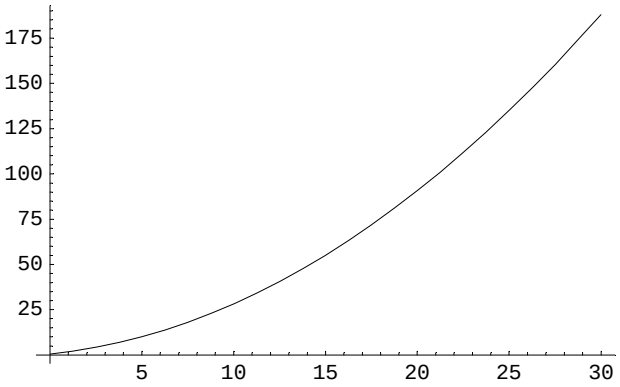
```
Fit[data, funs, vars] finds a least-squares fit to a list of data as a linear combination of
   the functions funs of variables vars. The data can have the form {{x1, y1, ... ,
   f1}, {x2, y2, ... , f2}, ... }, where the number of coordinates x, y, ...  is equal
   to the number of variables in the list vars. The data can also be of the form {f1,
   f2, ... }, with a single coordinate assumed to take values 1, 2, ... . The argument
   funs can be any list of functions that depend only on the objects vars. Mehr...
```

In[44]:= **tfit = Fit[t,{1,x,x^2},x]**

Out[44]= $0.566667 + 1.02273\,x + 0.174242\,x^2$

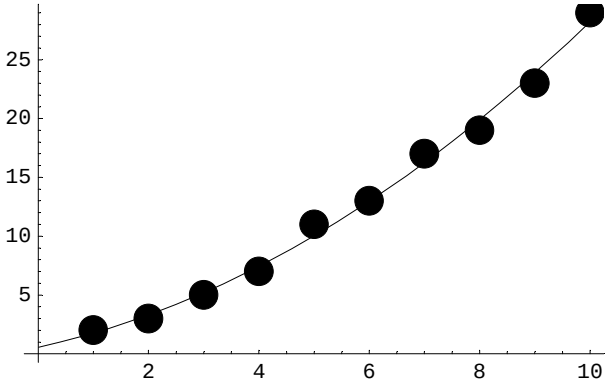
Fit-Kurve ansehen:
Regarder la courbe Fit:

In[45]:= **Plot[tfit,{x,0,30}];**



Kurve zusammen mit Punkten ansehen:
■ Regarder la courbe et les points ensemble:

In[46]:=
```
Show[
    ListPlot[t,PlotStyle->{PointSize[0.05]},
        DisplayFunction -> Identity],
    Plot[tfit,{x,0,10},DisplayFunction -> Identity], DisplayFunction :>
  $DisplayFunction
            ];
```



Abweichungen: ■ Ecarts:

In[47]:=
```
t = Table[Prime[n],{n,10}];
rang=Range[Length[t]]
```

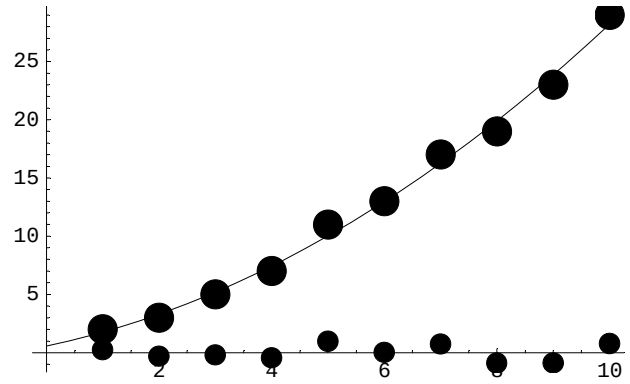
Out[48]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

```
In[49]:= tNew=Transpose[{rang,t}]

Out[49]= {{1, 2}, {2, 3}, {3, 5}, {4, 7}, {5, 11}, {6, 13}, {7, 17}, {8, 19}, {9, 23}, {10, 29}}

In[50]:= Clear[tfit]; tfit[x_] := Fit[tNew,{1,x,x^2},x];

In[51]:= tfit[x]

Out[51]= 0.566667 + 1.02273 x + 0.174242 x^2

In[52]:= tfit[x] /.x->3

Out[52]= 5.20303

In[53]:= te = Table[Prime[n]-(tfit[x]/.x->n),{n,10}]

Out[53]= {0.236364, -0.309091, -0.20303, -0.445455,
          0.963636, 0.0242424, 0.736364, -0.9, -0.884848, 0.781818}
```

Abweichungen im Plot:

■ Ecarts dans le Plot:

```
In[54]:= Show[
             ListPlot[tNew,PlotStyle->{PointSize[0.05]},
                 DisplayFunction -> Identity],
             ListPlot[te,PlotStyle->{PointSize[0.035]},
                 DisplayFunction -> Identity],
             Plot[Evaluate[(tfit[x]/.x->n)],{n,0,10},DisplayFunction -> Identity],
             DisplayFunction :> $DisplayFunction
                        ];
```



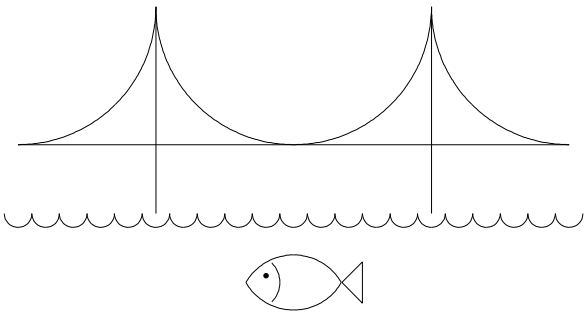
Aufgabe 13 ■ Problème 13

Zeichnen mit *Mathematica*
■ Dessiner avec *Mathematica*

Also los (Golden Gate Bridge):

■ Allez-y (Golden Gate Bridge):

```
In[55]:= Show[
             Graphics[{ (*Brücke*)
                 Circle[{-2,1},1,{3 Pi/2, 2 Pi}],
                 Circle[{0,1},1,{Pi, 2 Pi}],
                 Circle[{2,1},1,{Pi, 3 Pi/2}],
                 Line[{{-2,0},{2,0}}],
                 Line[{{-1,-0.5},{-1,1}}],
                 Line[{{1,-0.5},{1,1}}]        }],
             Graphics[ (*Wasser*)
                 Table[Circle[{x,-0.5},0.1,{Pi, 2Pi}],
                     {x, -2, 2, 0.2}]         ],
             Graphics[{ (*Fisch*)
                 Circle[{0,-1.2},0.4,{Pi/6, 5 Pi/6}],
                 Circle[{0,-0.8},0.4,{7 Pi/6, 11 Pi/6}],
                 Point[{-0.2,-0.95}],
                 Circle[{-0.3,-1},0.2,{- Pi/4, Pi/4}],
                 Line[{{0.35,-1},{0.5,-0.85},
                      {0.5,-1.15},{0.35,-1}}]
                      }],
                 AspectRatio->Automatic,
                 PlotRange->All  ];
```



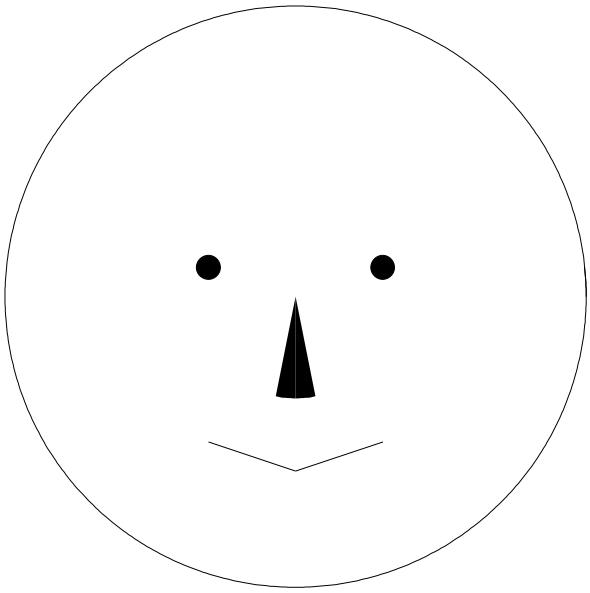
Aufgabe 14 ▮ Problème 14

Zeichnen mit *Mathematica*
▮ Dessiner avec *Mathematica*

Also los (geometrisches Gesicht):

▮ Allez-y (visage géométrique):

```
In[57]:= Show[
             Graphics[{
                 (*Augen*)
                 PointSize[0.04],
                 Point[{-0.3,0.1}],
                 Point[{0.3,0.1}],
                 (*Nase*)
                 Disk[{0,0},0.35,{23 Pi/16, 25 Pi/16}],
                 (*Mund*)
                 Line[{{-0.3,-0.5},{0,-0.6}}],
                 Line[{{0,-0.6},{0.3,-0.5}}],
                 (*Kopf*)
                 Circle[{0,0},1]
                  }],
                 AspectRatio->Automatic,
                 PlotRange->All  ];
```



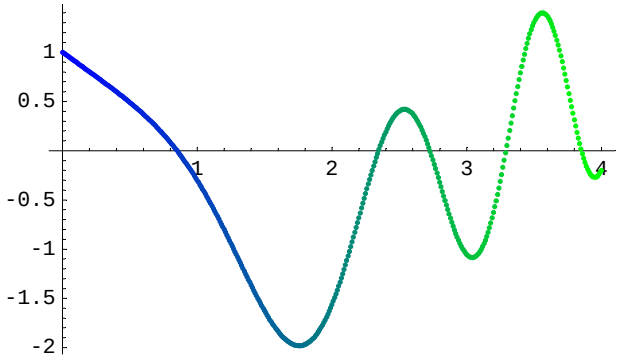
Aufgabe 15 ▮ Problème 15

Zeichnen mit *Mathematica*
▮ Dessiner avec *Mathematica*

Wie wirken Farben?:

▮ Quels sont les effets des couleurs?

In[59]:=
```
Show[
    Graphics[
        Table[{
            RGBColor[0,x/4.,1.-x/4.],
            Point[{x, Cos[x^2] - Sin[x]}]},
            {x,0,4,0.01}
            ]
            ],
            Axes->Automatic
        ];
```



"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[61]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[62]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

5. Etwas Umgang mit Mathematica
▮ Savoir un peu manier *Mathematica*

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 5 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 5.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Achtung: *Mathematica* neu starten, sonst könnte sehr viel Output auf den Schirm kommen.....
▮ Attention: Démarrer à nouveau *Mathematica*, sinon il pourrait y sortir beaucoup d'output sur l'écran...

5.1. Seiten-Breite setzen
▮ Fixer la largeur des pages

Ein Vorspann
▮ Préparatifs

"Material" erzeugen zur späteren Benutzung:
▮ Générer des "materiaux" à utiliser plus tard:

```
In[1]:= (*n30*)

In[2]:= Clear[n30]
```

In[3]:= **n30[x_]:=N[x,30]**

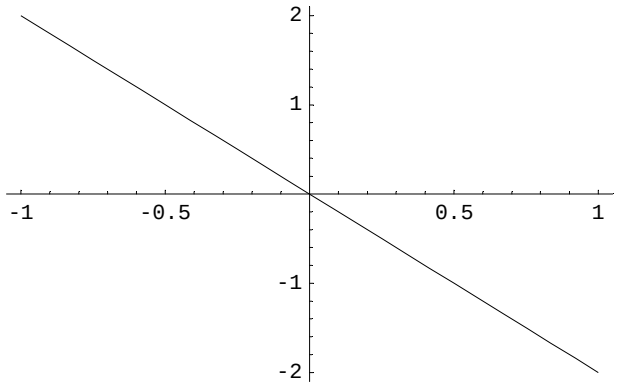
In[4]:= **a = 4**

Out[4]= 4

In[5]:= **n30[Pi]**

Out[5]= 3.14159265358979323846264338328

In[6]:= **Plot[-2x,{x,-1,1}];**



Anzahl Output-Charakter verändern:

- Changer le charactère de l'output:

In[7]:= **??SetOpt***

System`

```
SetOptions SetOptionsPacket
```

In[8]:= **??$Output**

```
$Output gives the list of files and pipes to which standard output from Mathematica is sent.
```
Mehr…

```
$Output = {OutputStream[stdout, 1]}
```

In[9]:= **??Pag***

System`

```
PageBreakAbove  PageFooterLines PageHeaders
PageBreakBelow  PageFooters     PageHeight
PageBreakWithin PageHeaderLines PageWidth
```

In[10]:= **??PageWidth**

```
PageWidth is an option for output streams and for cells which
   specifies how wide each line of text should be allowed to be.
```
Mehr…

```
Attributes[PageWidth] = {Protected}
```

```
In[11]:= SetOptions[$Output, PageWidth -> 45]

Out[11]= {{BinaryFormat → False, FormatType → StandardForm,
           PageWidth → 45, PageHeight → 22, TotalWidth → ∞, TotalHeight → ∞,
           CharacterEncoding :→ Unicode, NumberMarks :→ $NumberMarks}}

In[12]:= Options[$Output]

Out[12]= {BinaryFormat → False, FormatType → StandardForm,
          PageWidth → 45, PageHeight → 22, TotalWidth → ∞, TotalHeight → ∞,
          CharacterEncoding :→ Unicode, NumberMarks :→ $NumberMarks}
```

In 5.2. werden folgende Befehle besprochen: "In", "Out". Probiere:

■ Dans 5.2. on discute les ordres suivants: "In", "Out". Essaie:

```
In[13]:= ??In

        In[n] is a global object that is assigned to have a delayed value of the nth input line.
         Mehr…

        Attributes[In] = {Listable, Protected}

        In[1] := Null

        In[2] := Clear[n30]

        In[3] := n30[x_] := N[x, 30]

        In[4] := a = 4

        In[5] := n30[π]

        In[6] := (Plot[-2 x, {x, -1, 1}];)

        In[7] := Information[SetOpt*, LongForm → True]

        In[8] := Information[$Output, LongForm → True]

        In[9] := Information[Pag*, LongForm → True]

        In[10] := Information[PageWidth, LongForm → True]

        In[11] := SetOptions[$Output, PageWidth → 45]

        In[12] := Options[$Output]

        In[13] := Information[In, LongForm → True]
```

In[14]:=
```
linie = $Line;
??Out
```

```
%n or Out[n] is a global object that is assigned to be the value produced
   on the nth output line. % gives the last result generated. %% gives the
   result before last. %% ... % (k times) gives the kth previous result. Mehr...

Attributes[Out] = {Listable, Protected}

Out[0] = 0

%1 = Null

%2 = Null

%3 = Null

%4 = 4

%5 = 3.14159265358979323846264338328

%6 =
 Graphics[{{Line[{{-1., 2.}, {-0.918866, 1.83773}, {-0.830382, 1.66076}, {-0.747281, 1.49456},
      {-0.667363, 1.33473}, {-0.582295, 1.16459}, {-0.50041, 1.00082}, {-0.413374, 0.826748},
      {-0.329521, 0.659043}, {-0.248852, 0.497704}, {-0.163032, 0.326064},
      {-0.0803953, 0.160791}, {-0.000941658, 0.00188332}, {0.0836623, -0.167325},
      {0.165083, -0.330166}, {0.251654, -0.503309}, {0.335043, -0.670085},
      {0.415248, -0.830495}, {0.500603, -1.00121}, {0.582775, -1.16555}, {0.670098, -1.3402},
      {0.754238, -1.50848}, {0.835194, -1.67039}, {0.921301, -1.8426}, {1., -2.}}]}},
  {PlotRange → Automatic, AspectRatio → 1/GoldenRatio, DisplayFunction :→ $DisplayFunction,
   ColorOutput → Automatic, Axes → Automatic, AxesOrigin → Automatic,
   PlotLabel → None, AxesLabel → None, Ticks → Automatic,
   GridLines → None, Prolog → {}, Epilog → {}, AxesStyle → Automatic,
   Background → Automatic, DefaultColor → Automatic,
   DefaultFont :→ $DefaultFont, RotateLabel → True,
   Frame → False, FrameStyle → Automatic, FrameTicks → Automatic,
   FrameLabel → None, PlotRegion → Automatic, ImageSize → Automatic,
   TextStyle :→ $TextStyle, FormatType :→ $FormatType}]

%7 = {Null}

%8 = {Null}

%9 = {Null}

%10 = {Null}

%11 = {{BinaryFormat → False, FormatType → StandardForm, PageWidth → 45, PageHeight → 22,
   TotalWidth → ∞, TotalHeight → ∞, CharacterEncoding :→ Unicode, NumberMarks :→ $NumberMarks}}

%12 = {BinaryFormat → False, FormatType → StandardForm, PageWidth → 45, PageHeight → 22,
  TotalWidth → ∞, TotalHeight → ∞, CharacterEncoding :→ Unicode, NumberMarks :→ $NumberMarks}

%13 = {Null}

%14 = 14
```

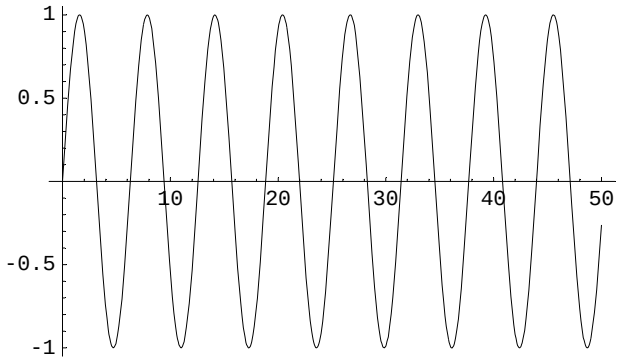
In[16]:=
```
linie
```

Out[16]= 14

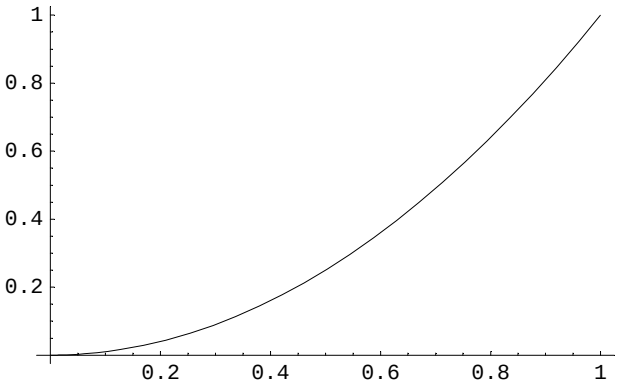
"Material" erzeugen zur späteren Benutzung:

■ Générer des "materiaux" à utiliser plus tard:

In[17]:= **Plot[Sin[x],{x,0,50}];**



In[18]:= **Plot[x^2,{x,0,1}];**



5.2. Listings von Input und Output
▮ Listings d'input et d'output

5.2.1. Input-Lines auflisten
▮ Faire une liste des input-lines

Mit "Recall" in älteren Versionen. Neu mit "InString":

▮ Dans les versions plus vieilles avec "Recall". Dans les versions nouvelles avec "InString":

In[19]:= **Recall[1, 2, 3]**

Out[19]= Recall[1, 2, 3]

In[20]:= **??InString**

```
InString[n] is a global object that is assigned to be the text of the nth input line. Mehr...

Attributes[InString] = {Listable, Protected}

InString[1] = (*n30*)

InString[2] = Clear[n30]

InString[3] = n30[x_]:=N[x,30]

InString[4] = a = 4

InString[5] = n30[Pi]

InString[6] = Plot[-2x,{x,-1,1}];

InString[7] = ??SetOpt*

InString[8] = ??$Output

InString[9] = ??Pag*

InString[10] = ??PageWidth

InString[11] = SetOptions[$Output, PageWidth -> 45]

InString[12] = Options[$Output]

InString[13] = ??In

InString[14] = linie = $Line;

InString[15] = ??Out

InString[16] = linie

InString[17] = Plot[Sin[x],{x,0,50}];

InString[18] = Plot[x^2,{x,0,1}];

InString[19] = Recall[1, 2, 3]

InString[20] = ??InString
```

In[21]:= **InString[{1, 2, 3, 4, 5, 6}]**

Out[21]=
```
{(*n30*), Clear[n30], n30[x_]:=N[x,30], a = 4, n30[Pi], Plot[-2x,{x,-1,1}];}
```

In[22]:= **In[1]**

In[23]:= **Out[1]**

Achtung: Wenn Du *Mathematica* nicht neu gestartet hast, so könnte hier sehr viel Output auf den Schirm kommen..... Es könnte eine Stunde oder mehr dauern!

Attention: Démarrer à nouveau *Mathematica*, sinon il

pourrait y sortir beaucoup d'output sur l'écran... Cela pourrait durer plus d'une heure!

In[24]:= **??In**

```
In[n] is a global object that is assigned to have a delayed value of the nth input line.
```
Mehr…

```
Attributes[In] = {Listable, Protected}

In[1] = Null

In[2] = Clear[n30]

In[3] = n30[x_] := N[x, 30]

In[4] = a = 4

In[5] = n30[π]

In[6] = (Plot[-2 x, {x, -1, 1}];)

In[7] = Information[SetOpt*, LongForm → True]

In[8] = Information[$Output, LongForm → True]

In[9] = Information[Pag*, LongForm → True]

In[10] = Information[PageWidth, LongForm → True]

In[11] = SetOptions[$Output, PageWidth → 45]

In[12] = Options[$Output]

In[13] = Information[In, LongForm → True]

In[14] = (linie = $Line;)

In[15] = Information[Out, LongForm → True]

In[16] = linie

In[17] = (Plot[Sin[x], {x, 0, 50}];)

In[18] = (Plot[x^2, {x, 0, 1}];)

In[19] = Recall[1, 2, 3]

In[20] = Information[InString, LongForm → True]

In[21] = InString[{1, 2, 3, 4, 5, 6}]

In[22] = In[1]

In[23] = %1

In[24] = Information[In, LongForm → True]
```

Das Folgende gibt sehr viel Output (zum Drucken hier gesperrt):
Ce qui suit produit beaucoup d'output (ici défense d'imprimer):

In[25]:= **(* ??Out *)**

5.2.2. In ein File schreiben und wieder abrufen:
▮ Ecrire dans un fichier et rappeler:

In[26]:= **?Save**

```
Save["filename", symbol] appends definitions associated with the specified symbol to a
   file. Save["filename", "form"] appends definitions associated with all symbols
   whose names match the string pattern form. Save["filename", "context`"] appends
   definitions associated with all symbols in the specified context. Save["filename",
   {object1, object2, ... }] appends definitions associated with several objects. Mehr...
```

In[27]:= **Save["C:\\work\\MathematicaData\\ASession.nb",**
In, linie, (*Out*)]

```
Syntax::com :
 Warning: comma encountered with no adjacent expression; the expression will be treated as Null.
 Mehr…
```

In[28]:= **!!C:\work\MathematicaData\ASession.nb**

```
Attributes[In] = {Listable, Protected}

In[1] = Plot[Sin[Removed["x"]], {Removed["x"], 0, 6*Pi}, AspectRatio -> 1/2,
     PlotLabel -> "Sin[x]", AxesLabel -> {"x", "y"},
     AxesOrigin -> {2*Pi, 1/2}]

In[2] = Null

In[3] = (Plot[Sin[Removed["x"]], {Removed["x"], 0, 6*Pi}, AspectRatio -> 1/2,
      PlotLabel -> "Sin[x]", AxesLabel -> {"x", "y"},
      AxesOrigin -> {2*Pi, 1/2}]; )

In[4] = Null

In[5] = (Plot[2*Removed["x"]^3 - 7*Removed["x"]^2 - 17*Removed["x"] + 10,
      {Removed["x"], -6, 6}]; )

In[6] = Solve[2*Removed["x"]^3 - 7*Removed["x"]^2 - 17*Removed["x"] + 10 ==
      0, Removed["x"]]

In[7] = (Plot[BesselJ[0, Removed["x"]], {Removed["x"], -6, 6}]; )

In[8] = Solve[2*Removed["x"]^3 - 7*Removed["x"]^2 - 17*Removed["x"] + 10 ==
      0, Removed["x"]]

In[9] = Remove[Removed["t"], Removed["x"], Removed["y"]]

In[10] = (Show[Plot[BesselJ[0, Removed["x"]], {Removed["x"], 0, 20},
       DisplayFunction -> Identity],
      Table[Graphics[{Dashing[{1/(10*Removed["t"])}], Thickness[0.01],
         Evaluate[Line[{{Removed["t"], BesselJ[0, Removed["t"]]},
            Evaluate[{Removed["y"] /. FindRoot[BesselJ[0, Removed["y"]] == 0,
                {Removed["y"], Removed["t"]}], 0}]}]]}],
       {Removed["t"], 1, 20, 3}], DisplayFunction :> $DisplayFunction]; )

In[11] = (Clear[Removed["f"]]; Removed["f"][Removed["x"]_] :=
      1/(1 + Sin[Sqrt[Pi^2 + Removed["x"]]^2]); Removed["f"][Removed["x"]])

In[12] = (Removed["n"][Removed["x"]_] :=
      Normal[Series[Removed["f"][Removed["x"]], {Removed["x"], 0, 8}]]; )

In[13] = (Removed["nTest"][Removed["x"]_] :=
      Normal[Series[Removed["f"][Removed["x"]], {Removed["x"], 0, 3}]]; )

In[14] = Removed["nTest"][Removed["x"]]

In[15] = Print[N[Removed["n"][Removed["x"]]]]
```

```
In[16] = (Removed["a"] = 1; )

In[17] = (Plot[Evaluate[N[Removed["n"][Removed["x"]]]],
      Evaluate[{Removed["x"], -Removed["a"], Removed["a"]}],
      PlotStyle -> {{Thickness[0.01], Dashing[{0.04}]}}]; )

In[18] = (Removed["a"] = 1; )

In[19] = (Show[Plot[Removed["f"][Removed["x"]],
       Evaluate[{Removed["x"], -Removed["a"], Removed["a"]}],
       DisplayFunction -> Identity, PlotStyle -> {GrayLevel[0.1]}],
      Plot[Evaluate[N[Removed["n"][Removed["x"]]]],
       Evaluate[{Removed["x"], -Removed["a"], Removed["a"]}],
       DisplayFunction -> Identity, PlotStyle ->
        {{Thickness[0.01], Dashing[{0.04}]}}], DisplayFunction :>
       $DisplayFunction]; )

In[20] = Null

In[21] = Null

In[22] = (Plot[Removed["x"]^2 + Cos[22*Removed["x"]],
      {Removed["x"], -5, 5}]; )

In[23] = (Plot[Removed["x"]^2 + Cos[22*Removed["x"]], {Removed["x"], -5, 5},
      PlotPoints -> 9]; )

In[24] = (Plot[Removed["x"]^2 + Cos[22*Removed["x"]], {Removed["x"], -5, 5},
      PlotPoints -> 70]; )

In[25] = (ParametricPlot[{Cos[4*Removed["t"]]*Cos[Removed["t"]],
       Cos[4*Removed["t"]]*Sin[Removed["t"]]}, {Removed["t"], 0, 2*Pi},
      AspectRatio -> Automatic]; )

In[26] = (ParametricPlot[{Cos[7*Removed["t"]]*Cos[3*Removed["t"]],
       Cos[7*Removed["t"]]*Sin[3*Removed["t"]]}, {Removed["t"], 0, 2*Pi},
      AspectRatio -> Automatic]; )

In[27] = (ParametricPlot[{Cos[7*Removed["t"]]*Cos[11*Removed["t"]],
       Cos[7*Removed["t"]]*Sin[11*Removed["t"]]}, {Removed["t"], 0, 2*Pi},
      AspectRatio -> Automatic]; )

In[28] = (Needs["Graphics`Polyhedra`"]; )

In[29] = (Show[Graphics3D[Icosahedron[]]]; )

In[30] = (Show[Graphics3D[Stellate[Icosahedron[], 3]], Boxed -> False]; )

In[31] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}]; )

In[32] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> False]; )

In[33] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {0, 0, 3}]; )

In[34] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {0, 0, 10}]; )

In[35] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {0, 3, 10}]; )

In[36] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {2, -4, 6}]; )

In[37] = (ContourPlot[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}]; )

In[38] = (DensityPlot[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}]; )

In[39] = (ContourPlot[Removed["x"]^2 + 2*Removed["y"]^2,
```

```
      {Removed["x"], -5, 5}, {Removed["y"], -5, 5},
      AspectRatio -> Automatic]; )

In[40] = Removed["t"] = Table[Prime[Removed["n"]], {Removed["n"], 10}]

In[41] = (ListPlot[Removed["t"], PlotStyle -> {PointSize[0.05]}]; )

In[42] = (ListPlot[Removed["t"], PlotStyle -> {PointSize[0.05]},
      PlotJoined -> True]; )

In[43] = Information["Fit", LongForm -> False]

In[44] = Removed["tfit"] = Fit[Removed["t"], {1, Removed["x"],
       Removed["x"]^2}, Removed["x"]]

In[45] = (Plot[Removed["tfit"], {Removed["x"], 0, 30}]; )

In[46] = (Show[ListPlot[Removed["t"], PlotStyle -> {PointSize[0.05]},
       DisplayFunction -> Identity], Plot[Removed["tfit"],
       {Removed["x"], 0, 10}, DisplayFunction -> Identity],
      DisplayFunction :> $DisplayFunction]; )

In[47] = (Removed["t"] = Table[Prime[Removed["n"]], {Removed["n"], 10}]; )

In[48] = Removed["rang"] = Range[Length[Removed["t"]]]

In[49] = Removed["tNew"] = Transpose[{Removed["rang"], Removed["t"]}]

In[50] = (Clear[Removed["tfit"]]; Removed["tfit"][Removed["x"]_] :=
      Fit[Removed["tNew"], {1, Removed["x"], Removed["x"]^2}, Removed["x"]]; )

In[51] = Removed["tfit"][Removed["x"]]

In[52] = Removed["tfit"][Removed["x"]] /. Removed["x"] -> 3

In[53] = Removed["te"] = Table[Prime[Removed["n"]] -
       (Removed["tfit"][Removed["x"]] /. Removed["x"] -> Removed["n"]),
      {Removed["n"], 10}]

In[54] = (Show[ListPlot[Removed["tNew"], PlotStyle -> {PointSize[0.05]},
       DisplayFunction -> Identity], ListPlot[Removed["te"],
       PlotStyle -> {PointSize[0.035]}, DisplayFunction -> Identity],
      Plot[Evaluate[Removed["tfit"][Removed["x"]] /. Removed["x"] ->
          Removed["n"]], {Removed["n"], 0, 10}, DisplayFunction -> Identity],
      DisplayFunction :> $DisplayFunction]; )

In[55] = (Show[Graphics[{Circle[{-2, 1}, 1, {(3*Pi)/2, 2*Pi}],
        Circle[{0, 1}, 1, {Pi, 2*Pi}], Circle[{2, 1}, 1, {Pi, (3*Pi)/2}],
        Line[{{-2, 0}, {2, 0}}], Line[{{-1, -0.5}, {-1, 1}}],
        Line[{{1, -0.5}, {1, 1}}]}], Graphics[
       Table[Circle[{Removed["x"], -0.5}, 0.1, {Pi, 2*Pi}],
        {Removed["x"], -2, 2, 0.2}]], Graphics[
       {Circle[{0, -1.2}, 0.4, {Pi/6, (5*Pi)/6}], Circle[{0, -0.8}, 0.4,
         {(7*Pi)/6, (11*Pi)/6}], Point[{-0.2, -0.95}],
        Circle[{-0.3, -1}, 0.2, {-(Pi/4), Pi/4}],
        Line[{{0.35, -1}, {0.5, -0.85}, {0.5, -1.15}, {0.35, -1}}]}],
      AspectRatio -> Automatic, PlotRange -> All]; )

In[56] = Null

In[57] = (Show[Graphics[{PointSize[0.04], Point[{-0.3, 0.1}],
        Point[{0.3, 0.1}], Disk[{0, 0}, 0.35, {(23*Pi)/16, (25*Pi)/16}],
        Line[{{-0.3, -0.5}, {0, -0.6}}], Line[{{0, -0.6}, {0.3, -0.5}}],
        Circle[{0, 0}, 1]}], AspectRatio -> Automatic, PlotRange -> All]; )

In[58] = Null

In[59] = (Show[Graphics[Table[{RGBColor[0, Removed["x"]/4.,
          1. - Removed["x"]/4.], Point[{Removed["x"], Cos[Removed["x"]^2] -
            Sin[Removed["x"]]}]}, {Removed["x"], 0, 4, 0.01}]],
      Axes -> Automatic]; )

In[60] = Null
```

```
In[61] = Null

In[62] = Remove["Global`*"]

In[63] = Remove["Global`@*"]

In[64] = Null

In[65] = Clear[n30]

In[66] = n30[x_] := N[x, 30]

In[67] = a = 4

In[68] = n30[Pi]

In[69] = (Plot[-2*x, {x, -1, 1}]; )

In[70] = Information["SetOpt*", LongForm -> True]

In[71] = Information["$Output", LongForm -> True]

In[72] = Information["Pag*", LongForm -> True]

In[73] = Information["PageWidth", LongForm -> True]

In[74] = SetOptions[$Output, PageWidth -> 45]

In[75] = Options[$Output]

In[76] = Information["In", LongForm -> True]

In[77] = (linie = $Line; )

In[78] = Information["Out", LongForm -> True]

In[79] = linie

In[80] = (Plot[Sin[x], {x, 0, 50}]; )

In[81] = (Plot[x^2, {x, 0, 1}]; )

In[82] = Recall[1, 2, 3]

In[83] = Information["InString", LongForm -> True]

In[84] = InString[{1, 2, 3, 4, 5, 6}]

In[85] = In[1]

In[86] = Out[1]

In[87] = Information["In", LongForm -> True]

In[88] = Null

In[89] = Save["C:\\work\\ASession.nb", In, linie, Null]

In /: In::usage = "In[n] is a global object that is assigned to have a \
delayed value of the nth input line."

linie = 77

Attributes[Null] = {Protected}
Attributes[In] = {Listable, Protected}

In[1] = Null

In[2] = Remove["Global`*"]

In[3] = Null
```

```
In[4] = Clear[n30]

In[5] = n30[x_] := N[x, 30]

In[6] = a = 4

In[7] = n30[Pi]

In[8] = (Plot[-2*x, {x, -1, 1}]; )

In[9] = Information["SetOpt*", LongForm -> True]

In[10] = Information["$Output", LongForm -> True]

In[11] = Information["Pag*", LongForm -> True]

In[12] = Information["PageWidth", LongForm -> True]

In[13] = SetOptions[$Output, PageWidth -> 45]

In[14] = Options[$Output]

In[15] = Information["In", LongForm -> True]

In[16] = (linie = $Line; )

In[17] = Information["Out", LongForm -> True]

In[18] = linie

In[19] = (Plot[Sin[x], {x, 0, 50}]; )

In[20] = (Plot[x^2, {x, 0, 1}]; )

In[21] = Recall[1, 2, 3]

In[22] = Information["InString", LongForm -> True]

In[23] = InString[{1, 2, 3, 4, 5, 6}]

In[24] = In[1]

In[25] = Out[1]

In[26] = Information["In", LongForm -> True]

In[27] = Null

In[28] = Save["C:\\work\\ASession.nb", In, linie, Null]

In /: In::usage = "In[n] is a global object that is assigned to have a \
delayed value of the nth input line."

linie = 16

Attributes[Null] = {Protected}
Attributes[In] = {Listable, Protected}

In[1] = Plot[Sin[Removed["x"]], {Removed["x"], 0, 6*Pi}, AspectRatio -> 1/2,
     PlotLabel -> "Sin[x]", AxesLabel -> {"x", "y"},
     AxesOrigin -> {2*Pi, 1/2}]

In[2] = Null

In[3] = (Plot[Sin[Removed["x"]], {Removed["x"], 0, 6*Pi}, AspectRatio -> 1/2,
      PlotLabel -> "Sin[x]", AxesLabel -> {"x", "y"},
      AxesOrigin -> {2*Pi, 1/2}]; )

In[4] = Null

In[5] = (Plot[2*Removed["x"]^3 - 7*Removed["x"]^2 - 17*Removed["x"] + 10,
      {Removed["x"], -6, 6}]; )
```

```
In[6] = Solve[2*Removed["x"]^3 - 7*Removed["x"]^2 - 17*Removed["x"] + 10 ==
      0, Removed["x"]]

In[7] = (Plot[BesselJ[0, Removed["x"]], {Removed["x"], -6, 6}]; )

In[8] = Solve[2*Removed["x"]^3 - 7*Removed["x"]^2 - 17*Removed["x"] + 10 ==
      0, Removed["x"]]

In[9] = Remove[Removed["t"], Removed["x"], Removed["y"]]

In[10] = (Show[Plot[BesselJ[0, Removed["x"]], {Removed["x"], 0, 20},
       DisplayFunction -> Identity],
      Table[Graphics[{Dashing[{1/(10*Removed["t"])}], Thickness[0.01],
         Evaluate[Line[{{Removed["t"], BesselJ[0, Removed["t"]]},
            Evaluate[{Removed["y"] /. FindRoot[BesselJ[0, Removed["y"]] == 0,
                {Removed["y"], Removed["t"]}], 0}]}]]}],
       {Removed["t"], 1, 20, 3}], DisplayFunction :> $DisplayFunction]; )

In[11] = (Clear[Removed["f"]]; Removed["f"][Removed["x"]_] :=
      1/(1 + Sin[Sqrt[Pi^2 + Removed["x"]]^2]); Removed["f"][Removed["x"]])

In[12] = (Removed["n"][Removed["x"]_] :=
      Normal[Series[Removed["f"][Removed["x"]], {Removed["x"], 0, 8}]]; )

In[13] = (Removed["nTest"][Removed["x"]_] :=
      Normal[Series[Removed["f"][Removed["x"]], {Removed["x"], 0, 3}]]; )

In[14] = Removed["nTest"][Removed["x"]]

In[15] = Print[N[Removed["n"][Removed["x"]]]]

In[16] = (Removed["a"] = 1; )

In[17] = (Plot[Evaluate[N[Removed["n"][Removed["x"]]]],
      Evaluate[{Removed["x"], -Removed["a"], Removed["a"]}],
      PlotStyle -> {{Thickness[0.01], Dashing[{0.04}]}}]; )

In[18] = (Removed["a"] = 1; )

In[19] = (Show[Plot[Removed["f"][Removed["x"]],
       Evaluate[{Removed["x"], -Removed["a"], Removed["a"]}],
       DisplayFunction -> Identity, PlotStyle -> {GrayLevel[0.1]}],
      Plot[Evaluate[N[Removed["n"][Removed["x"]]]],
       Evaluate[{Removed["x"], -Removed["a"], Removed["a"]}],
       DisplayFunction -> Identity, PlotStyle ->
        {{Thickness[0.01], Dashing[{0.04}]}}], DisplayFunction :>
       $DisplayFunction]; )

In[20] = Null

In[21] = Null

In[22] = (Plot[Removed["x"]^2 + Cos[22*Removed["x"]],
      {Removed["x"], -5, 5}]; )

In[23] = (Plot[Removed["x"]^2 + Cos[22*Removed["x"]], {Removed["x"], -5, 5},
      PlotPoints -> 9]; )

In[24] = (Plot[Removed["x"]^2 + Cos[22*Removed["x"]], {Removed["x"], -5, 5},
      PlotPoints -> 70]; )

In[25] = (ParametricPlot[{Cos[4*Removed["t"]]*Cos[Removed["t"]],
       Cos[4*Removed["t"]]*Sin[Removed["t"]]}, {Removed["t"], 0, 2*Pi},
      AspectRatio -> Automatic]; )

In[26] = (ParametricPlot[{Cos[7*Removed["t"]]*Cos[3*Removed["t"]],
       Cos[7*Removed["t"]]*Sin[3*Removed["t"]]}, {Removed["t"], 0, 2*Pi},
      AspectRatio -> Automatic]; )

In[27] = (ParametricPlot[{Cos[7*Removed["t"]]*Cos[11*Removed["t"]],
       Cos[7*Removed["t"]]*Sin[11*Removed["t"]]}, {Removed["t"], 0, 2*Pi},
      AspectRatio -> Automatic]; )
```

```
In[28] = (Needs["Graphics`Polyhedra`"]; )

In[29] = (Show[Graphics3D[Icosahedron[]]]; )

In[30] = (Show[Graphics3D[Stellate[Icosahedron[], 3]], Boxed -> False]; )

In[31] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}]; )

In[32] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> False]; )

In[33] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {0, 0, 3}]; )

In[34] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {0, 0, 10}]; )

In[35] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {0, 3, 10}]; )

In[36] = (Plot3D[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}, Lighting -> True, ViewPoint -> {2, -4, 6}]; )

In[37] = (ContourPlot[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}]; )

In[38] = (DensityPlot[Sin[Removed["x"]*Removed["y"]], {Removed["x"], -5, 5},
      {Removed["y"], -5, 5}]; )

In[39] = (ContourPlot[Removed["x"]^2 + 2*Removed["y"]^2,
      {Removed["x"], -5, 5}, {Removed["y"], -5, 5},
      AspectRatio -> Automatic]; )

In[40] = Removed["t"] = Table[Prime[Removed["n"]], {Removed["n"], 10}]

In[41] = (ListPlot[Removed["t"], PlotStyle -> {PointSize[0.05]}]; )

In[42] = (ListPlot[Removed["t"], PlotStyle -> {PointSize[0.05]},
      PlotJoined -> True]; )

In[43] = Information["Fit", LongForm -> False]

In[44] = Removed["tfit"] = Fit[Removed["t"], {1, Removed["x"],
       Removed["x"]^2}, Removed["x"]]

In[45] = (Plot[Removed["tfit"], {Removed["x"], 0, 30}]; )

In[46] = (Show[ListPlot[Removed["t"], PlotStyle -> {PointSize[0.05]},
       DisplayFunction -> Identity], Plot[Removed["tfit"],
       {Removed["x"], 0, 10}, DisplayFunction -> Identity],
      DisplayFunction :> $DisplayFunction]; )

In[47] = (Removed["t"] = Table[Prime[Removed["n"]], {Removed["n"], 10}]; )

In[48] = Removed["rang"] = Range[Length[Removed["t"]]]

In[49] = Removed["tNew"] = Transpose[{Removed["rang"], Removed["t"]}]

In[50] = (Clear[Removed["tfit"]]; Removed["tfit"][Removed["x"]_] :=
      Fit[Removed["tNew"], {1, Removed["x"], Removed["x"]^2}, Removed["x"]]; )

In[51] = Removed["tfit"][Removed["x"]]

In[52] = Removed["tfit"][Removed["x"]] /. Removed["x"] -> 3

In[53] = Removed["te"] = Table[Prime[Removed["n"]] -
       (Removed["tfit"][Removed["x"]] /. Removed["x"] -> Removed["n"]),
      {Removed["n"], 10}]

In[54] = (Show[ListPlot[Removed["tNew"], PlotStyle -> {PointSize[0.05]},
       DisplayFunction -> Identity], ListPlot[Removed["te"],
       PlotStyle -> {PointSize[0.035]}, DisplayFunction -> Identity],
```

```
      Plot[Evaluate[Removed["tfit"][Removed["x"]] /. Removed["x"] ->
          Removed["n"]], {Removed["n"], 0, 10}, DisplayFunction -> Identity],
      DisplayFunction :> $DisplayFunction]; )

In[55] = (Show[Graphics[{Circle[{-2, 1}, 1, {(3*Pi)/2, 2*Pi}],
       Circle[{0, 1}, 1, {Pi, 2*Pi}], Circle[{2, 1}, 1, {Pi, (3*Pi)/2}],
       Line[{{-2, 0}, {2, 0}}], Line[{{-1, -0.5}, {-1, 1}}],
       Line[{{1, -0.5}, {1, 1}}]}], Graphics[
      Table[Circle[{Removed["x"], -0.5}, 0.1, {Pi, 2*Pi}],
       {Removed["x"], -2, 2, 0.2}]], Graphics[
      {Circle[{0, -1.2}, 0.4, {Pi/6, (5*Pi)/6}], Circle[{0, -0.8}, 0.4,
        {(7*Pi)/6, (11*Pi)/6}], Point[{-0.2, -0.95}],
       Circle[{-0.3, -1}, 0.2, {-(Pi/4), Pi/4}],
       Line[{{0.35, -1}, {0.5, -0.85}, {0.5, -1.15}, {0.35, -1}}]}],
      AspectRatio -> Automatic, PlotRange -> All]; )

In[56] = Null

In[57] = (Show[Graphics[{PointSize[0.04], Point[{-0.3, 0.1}],
       Point[{0.3, 0.1}], Disk[{0, 0}, 0.35, {(23*Pi)/16, (25*Pi)/16}],
       Line[{{-0.3, -0.5}, {0, -0.6}}], Line[{{0, -0.6}, {0.3, -0.5}}],
       Circle[{0, 0}, 1]}], AspectRatio -> Automatic, PlotRange -> All]; )

In[58] = Null

In[59] = (Show[Graphics[Table[{RGBColor[0, Removed["x"]/4.,
         1. - Removed["x"]/4.], Point[{Removed["x"], Cos[Removed["x"]^2] -
           Sin[Removed["x"]]}]}, {Removed["x"], 0, 4, 0.01}]],
      Axes -> Automatic]; )

In[60] = Null

In[61] = Null

In[62] = Remove["Global`*"]

In[63] = Null

In[64] = Remove["Global`*"]

In[65] = Null

In[66] = Clear[Removed["n30"]]

In[67] = Removed["n30"][Removed["x"]_] := N[Removed["x"], 30]

In[68] = Removed["a"] = 4

In[69] = Removed["n30"][Pi]

In[70] = (Plot[-2*Removed["x"], {Removed["x"], -1, 1}]; )

In[71] = Information["SetOpt*", LongForm -> True]

In[72] = Information["$Output", LongForm -> True]

In[73] = Information["Pag*", LongForm -> True]

In[74] = Information["PageWidth", LongForm -> True]

In[75] = SetOptions[$Output, PageWidth -> 45]

In[76] = Options[$Output]

In[77] = Information["In", LongForm -> True]

In[78] = (Removed["linie"] = $Line; )

In[79] = Information["Out", LongForm -> True]

In[80] = Removed["linie"]

In[81] = (Plot[Sin[Removed["x"]], {Removed["x"], 0, 50}]; )
```

```
In[82] = (Plot[Removed["x"]^2, {Removed["x"], 0, 1}]; )

In[83] = Removed["Recall"][1, 2, 3]

In[84] = Information["InString", LongForm -> True]

In[85] = InString[{1, 2, 3, 4, 5, 6}]

In[86] = In[1]

In[87] = Out[1]

In[88] = Information["In", LongForm -> True]

In[89] = Null

In[90] = Information["Save", LongForm -> False]

In[91] = Save["C:\\work\\ASession.nb", In, Removed["linie"], Null]

In[92] = Null

In[93] = Information["Edit", LongForm -> True]

In[94] = Information["EditDefinition", LongForm -> True]

In[95] = Information["*Edit*", LongForm -> False]

In[96] = Information["EditIn", LongForm -> True]

In[97] = Information["*Edit*", LongForm -> True]

In[98] = Removed["x"]^6 + 12*Removed["x"]^5*Removed["y"] +
     60*Removed["x"]^4*Removed["y"]^2 + 160*Removed["x"]^3*Removed["y"]^3 +
     240*Removed["x"]^2*Removed["y"]^4 + 192*Removed["x"]*Removed["y"]^5 +
     64*Removed["y"]^6

In[99] = Null

In[100] = Clear[Removed["n30"]]

In[101] = Removed["n30"][Removed["x"]_] := N[Removed["x"], 30]

In[102] = Null

In[103] = Clear[Removed["a"]]

In[104] = Removed["a"] = 4

In[105] = Information["n30", LongForm -> True]

In[106] = Information["a", LongForm -> True]

In[107] = Series[Csc[Removed["x"]], {Removed["x"], 0, 7}]

In[108] = Information["Series", LongForm -> False]

In[109] = Information["Csc", LongForm -> False]

In[110] = Information["X* Z*", LongForm -> False]

In[111] = Names["W*"]

In[112] = Names["A*"]

In[113] = Names["*"]

In[114] = Removed["f"][Removed["x"]_] := Sin[Removed["x"]]*Cos[2*Removed["x"]]

In[115] = Removed["f"][Removed["x"]]

In[116] = Removed["f"][Pi]
```

```
In[117] = (Plot[Removed["f"][Removed["x"]], {Removed["x"], 0, 10}]; )

In[118] = (Removed["f1"][Removed["x"]_] := Sin[2*Removed["x"]]*
       Cos[3*Removed["x"]^2]; )

In[119] = (Plot[Removed["f1"][Removed["x"]], {Removed["x"], 0, 10}]; )

In[120] = (Removed["g"][Removed["x"]] := Sin[2*Removed["x"]]*
       Cos[3*Removed["x"]^2]; )

In[121] = (Plot[Removed["g"][Removed["x"]], {Removed["x"], 0, 10}]; )

In[122] = (Removed["g1"][Removed["x"]_] = Sin[2*Removed["x"]]*
       Cos[3*Removed["x"]^2]; )

In[123] = (Plot[Removed["g1"][Removed["x"]], {Removed["x"], 0, 10}]; )

In[124] = (Removed["g2"][Removed["x"]] = Sin[2*Removed["x"]]*
       Cos[3*Removed["x"]^2]; )

In[125] = (Plot[Removed["g2"][Removed["x"]], {Removed["x"], 0, 10}]; )

In[126] = N[Sqrt[2]/2]

In[127] = N[Sqrt[2]/2]

In[128] = Names["W*"]

In[129] = Length[Names["W*"]]

In[130] = Removed["x"] = Pi/2

In[131] = Sin[Removed["x"]]

In[132] = N[Sin[Sin[Removed["x"]]]]

In[133] = N[Sin[Sin[Sin[Removed["x"]]]]]

In[134] = N[Sin[Sin[Sin[Sin[Sin[Removed["x"]]]]]]]

In[135] = N[Sin[Sin[Sin[Sin[Sin[Removed["x"]]]]]]]

In[136] = Sin[4.08]

In[137] = (Sin[3.96]; )

In[138] = %

In[139] = (Removed["a"] = 1; Removed["b"] = 2; Removed["c"] = 3;
     Removed["d"] = 4)

In[140] = (Removed["a"] = 1; Removed["b"] = 2; Removed["c"] = 3;
     Removed["d"] = 4; )

In[141] = (Removed["a"] = 1; Removed["b"] = 2; Print[Removed["b"]];
     Removed["c"] = 3; Removed["d"] = 4; )

In[142] = (Removed["a"] = 1; Removed["b"] = 2; Print[Removed["b"]];
     Removed["c"] = 3; Removed["d"] = 4)

In[143] = (Removed["a"] = 1; Removed["b"] = 2; Plot[Removed["x"],
      {Removed["x"], 0, 1}]; Removed["c"] = 3; Removed["d"] = 4)

In[144] = Timing[N[Pi, 500]]

In[145] = Timing[N[Pi, 500]; ]

In[146] = Timing[N[Pi, 2000]; ]

In[147] = Timing[Plot[Sin[Removed["x"]], {Removed["x"], 0, 50}]]

In[148] = Timing[Plot[Sin[Removed["x"]], {Removed["x"], 0, 50}]; ]
```

```
In[149] = 5000!

In[150] = Timing[1000!]

In[151] = Information["$*", LongForm -> True]

In[152] = Information["$Line", LongForm -> True]

In[153] = $Line

In[154] = $Version

In[155] = $VersionNumber

In[156] = Removed["$Letters"]

In[157] = Information["/", LongForm -> True]

In[158] = Alias["/"]

In[159] = N[2/3]

In[160] = N[2/3]

In[161] = 2/3

In[162] = Divide @@ {2, 3}

In[163] = Information["Apply", LongForm -> True]

In[164] = Null

In[165] = Remove["Global`*"]

In[166] = Null

In[167] = Clear[n30]

In[168] = n30[x_] := N[x, 30]

In[169] = a = 4

In[170] = n30[Pi]

In[171] = (Plot[-2*x, {x, -1, 1}]; )

In[172] = Information["SetOpt*", LongForm -> True]

In[173] = Information["$Output", LongForm -> True]

In[174] = Information["Pag*", LongForm -> True]

In[175] = Information["PageWidth", LongForm -> True]

In[176] = SetOptions[$Output, PageWidth -> 45]

In[177] = Options[$Output]

In[178] = Information["In", LongForm -> True]

In[179] = (linie = $Line; )

In[180] = Information["Out", LongForm -> True]

In[181] = linie

In[182] = (Plot[Sin[x], {x, 0, 50}]; )

In[183] = (Plot[x^2, {x, 0, 1}]; )

In[184] = Recall[1, 2, 3]
```

```
In[185] = Information["InString", LongForm -> True]

In[186] = InString[{1, 2, 3, 4, 5, 6}]

In[187] = In[1]

In[188] = Out[1]

In[189] = Information["In", LongForm -> True]

In[190] = Null

In[191] = Information["Save", LongForm -> False]

In[192] = Save["C:\\work\\MathematicaData\\ASession.nb", In, linie, Null]

In /: In::usage = "In[n] is a global object that is assigned to have a \
delayed value of the nth input line."

linie = 179

Attributes[Null] = {Protected}
Attributes[In] = {Listable, Protected}

In[1] = Null

In[2] = Clear[n30]

In[3] = n30[x_] := N[x, 30]

In[4] = a = 4

In[5] = n30[Pi]

In[6] = (Plot[-2*x, {x, -1, 1}]; )

In[7] = Information["SetOpt*", LongForm -> True]

In[8] = Information["$Output", LongForm -> True]

In[9] = Information["Pag*", LongForm -> True]

In[10] = Information["PageWidth", LongForm -> True]

In[11] = SetOptions[$Output, PageWidth -> 45]

In[12] = Options[$Output]

In[13] = Information["In", LongForm -> True]

In[14] = (linie = $Line; )

In[15] = Information["Out", LongForm -> True]

In[16] = linie

In[17] = (Plot[Sin[x], {x, 0, 50}]; )

In[18] = (Plot[x^2, {x, 0, 1}]; )

In[19] = Recall[1, 2, 3]

In[20] = Information["InString", LongForm -> True]

In[21] = InString[{1, 2, 3, 4, 5, 6}]

In[22] = In[1]

In[23] = Out[1]

In[24] = Information["In", LongForm -> True]

In[25] = Null
```

```
In[26] = Information["Save", LongForm -> False]

In[27] = Save["C:\\work\\MathematicaData\\ASession.nb", In, linie, Null]

In /: In::usage = "In[n] is a global object that is assigned to have a \
delayed value of the nth input line."

linie = 14

Attributes[Null] = {Protected}
```

5.3. Zum Editor
▮ Quant à l'éditeur

Hier für "NeXT"! (Für andere Editoren ev. anders, vgl. *Mathematica*, a Practical Approach von Nancy Blachman oder Literatur)
▮ Ici seulement pour "NeXT"! (Pour d'autres éditeurs voir *Mathematica*, a Practical Approach de Nancy Blachman ou littérature)

Einige Edit-Funktionen:

▮ Quelques fonctions Edit:

In[29]:= **??Edit**

```
Edit[expr___] lets you edit the expressions expr.

Attributes[Edit] = {Protected, ReadProtected}
```

In[30]:= **??EditDefinition**

```
EditDefinition[syms___] lets you edit the rules attached to the symbols syms.

Attributes[EditDefinition] = {HoldAll, Protected, ReadProtected}
```

In[31]:= **?*Edit***

System`

```
CellEditDuplicate  EditCellTagsSettings
Edit               EditDefinition
Editable           EditIn
EditButtonSettings OutputMathEditExpression
```

In[32]:= **??EditIn**

```
EditIn[n__Integer] lets you edit input lines n. EditIn[] edits the previous input line.

Attributes[EditIn] = {Protected, ReadProtected}
```

In[33]:= **??*Edit***

System`

```
CellEditDuplicate  EditCellTagsSettings
Edit               EditDefinition
Editable           EditIn
EditButtonSettings OutputMathEditExpression
```

Beispiele: ■ Exemples

In[34]:= **Edit[Expand[(x+2y)^6]]**

Out[34]= $x^6 + 12\,x^5\,y + 60\,x^4\,y^2 + 160\,x^3\,y^3 + 240\,x^2\,y^4 + 192\,x\,y^5 + 64\,y^6$

Oder: ■ Ou:

In[35]:= **EditDefinition[n30]**

In[38]:= **EditDefinition[a]**

Out[40]= 4

Funktioniert der Befehl auf NeXT?
■ L'ordre fonctionne-t-il sur NeXT?

In[41]:= **??n30**

```
Global`n30

n30[x_] := N[x, 30]
```

In[42]:= **??a**

```
Global`a

a = 4
```

5.4. Zu On-Line Help
■ Quant à On-Line Help

Ein Trick
■ Un truc

Weisst Du beim Schreiben innerhalb eines Befehls nicht mehr weiter, so kannst Du in der selben Zelle die Help-Funktion benutzen.
■ Si tu ne t'en sors plus en écrivant à l'intérieur d'un ordre, tu peux utiliser la fonction Help dans la même cellule.

Beispiel: ■ Exemple:

In[43]:= **Series[Csc[x],{x,0,7}]**

Out[43]= $\frac{1}{x}+\frac{x}{6}+\frac{7\,x^3}{360}+\frac{31\,x^5}{15120}+\frac{127\,x^7}{604800}+O[x]^8$

In[44]:= **?Series**

```
Series[f, {x, x0, n}] generates a power series expansion for f
   about the point x = x0 to order (x - x0)^n. Series[f, {x, x0, nx}, {y, y0,
   ny}] successively finds series expansions with respect to y, then x.
```
Mehr...

Natürlich kannst Du auch eine neue Zelle machen:

■ Naturellement tu peux faire une nouvelle cellule:

In[45]:= **?Csc**

```
Csc[z] gives the cosecant of z.
```
Mehr...

5.5. Ein Blank kann "oder" bedeuten...
■ Un "blanc" peut signifier "ou":

Das Problem besteht nur bei Version 1.2.:

■ Le problème existe seulement pour la version 1.2.:

In[46]:= **?X* Z***

System`

```
XMLElement XMLObject Xor
```

System`

```
ZeroTest ZeroWidthTimes Zeta ZTransform
```

5.6. Symbollisten erzeugen:
■ Générer des listes de symboles

5.6.1. Ein Beispiel, das alles sagt:
■ Un exemple qui dit tout:

In[47]:= **Names["W*"]**

Out[47]= {WeierstrassHalfPeriods, WeierstrassInvariants, WeierstrassP, WeierstrassPPrime, WeierstrassSigma, WeierstrassZeta, Which, While, White, Whitespace, WhitespaceCharacter, WindowClickSelect, WindowElements, WindowFloating, WindowFrame, WindowFrameElements, WindowMargins, WindowMovable, WindowSize, WindowTitle, WindowToolbars, WindowWidth, With, Word, WordBoundary, WordCharacter, WordSearch, WordSeparators, WorkingPrecision, Write, WriteString, WynnDegree}

Oder: ■ Ou:

```
In[48]:= Names["A*"]

Out[48]= {Abort, AbortProtect, Above, Abs, AbsoluteDashing, AbsoluteOptions,
          AbsolutePointSize, AbsoluteThickness, AbsoluteTime, AbsoluteTiming,
          AccountingForm, Accuracy, AccuracyGoal, Active, ActiveItem, AddOnHelpPath,
          AddTo, AdjustmentBox, AdjustmentBoxOptions, After, AiryAi, AiryAiPrime,
          AiryBi, AiryBiPrime, AlgebraicRules, AlgebraicRulesData, Algebraics, Alias,
          AlignmentMarker, All, AllowInlineCells, AllowScriptLevelChange, Alternatives,
          AmbientLight, Analytic, AnchoredSearch, And, AnimationCycleOffset,
          AnimationCycleRepetitions, AnimationDirection, AnimationDisplayTime, Apart,
          ApartSquareFree, AppellF1, Append, AppendTo, Apply, ArcCos, ArcCosh, ArcCot,
          ArcCoth, ArcCsc, ArcCsch, ArcSec, ArcSech, ArcSin, ArcSinh, ArcTan, ArcTanh,
          Arg, ArgumentCountQ, ArithmeticGeometricMean, Array, ArrayDepth, ArrayPlot,
          ArrayQ, ArrayRules, AspectRatio, AspectRatioFixed, Assuming, Assumptions,
          AtomQ, Attributes, AutoDelete, AutoEvaluateEvents, AutoGeneratedPackage,
          AutoIndent, AutoIndentSpacings, AutoItalicWords, AutoloadPath,
          Automatic, AutoOpenNotebooks, AutoOpenPalettes, AutoScroll, AutoSpacing,
          AutoStyleOptions, Axes, AxesEdge, AxesLabel, AxesOrigin, AxesStyle, Axis}
```

Das Problem besteht nur bei Version 1.2.:

■ Le problème existe seulement pour la version 1.2.:

5.6.2. Jetzt ganz *Mathematica* sowie alle bisherigen eigenen Definitionen auf einmal:

■ Maintenant tout *Mathematica* ainsi que toutes les propres définitions jusqu'à présent à la fois:

```
In[49]:= Names["*"]

Out[49]= {a, Abort, AbortProtect, Above, Abs, AbsoluteDashing, AbsoluteOptions,
          AbsolutePointSize, AbsoluteThickness, AbsoluteTime, AbsoluteTiming,
          AccountingForm, Accuracy, AccuracyGoal, Active, ActiveItem, AddOnHelpPath,
          AddTo, AdjustmentBox, AdjustmentBoxOptions, After, AiryAi, AiryAiPrime,
          AiryBi, AiryBiPrime, AlgebraicRules, AlgebraicRulesData, Algebraics, Alias,
          AlignmentMarker, All, AllowInlineCells, AllowScriptLevelChange, Alternatives,
          AmbientLight, Analytic, AnchoredSearch, And, AnimationCycleOffset,
          AnimationCycleRepetitions, AnimationDirection, AnimationDisplayTime, Apart,
          ApartSquareFree, AppellF1, Append, AppendTo, Apply, ArcCos, ArcCosh, ArcCot,
          ArcCoth, ArcCsc, ArcCsch, ArcSec, ArcSech, ArcSin, ArcSinh, ArcTan, ArcTanh,
          Arg, ArgumentCountQ, ArithmeticGeometricMean, Array, ArrayDepth, ArrayPlot,
          ArrayQ, ArrayRules, AspectRatio, AspectRatioFixed, Assuming, Assumptions,
          AtomQ, Attributes, AutoDelete, AutoEvaluateEvents, AutoGeneratedPackage,
          AutoIndent, AutoIndentSpacings, AutoItalicWords, AutoloadPath, Automatic,
          AutoOpenNotebooks, AutoOpenPalettes, AutoScroll, AutoSpacing, AutoStyleOptions,
          Axes, AxesEdge, AxesLabel, AxesOrigin, AxesStyle, Axis, Background,
          BackgroundTasksSettings, Backsubstitution, Backward, BaseForm, Baseline,
          Before, Begin, BeginDialogPacket, BeginFrontEndInteractionPacket, BeginPackage,
          Below, BernoulliB, BesselI, BesselJ, BesselK, BesselY, Beta, BetaRegularized,
          BezoutMatrix, BinaryFormat, BinaryGet, BinaryRead, BinaryReadList,
          BinaryWrite, Binomial, BitAnd, BitNot, BitOr, BitXor, Black, Blank, BlankForm,
          BlankNullSequence, BlankSequence, Block, Blue, Boole, Booleans, Bottom, Bounds,
          Box, BoxBaselineShift, BoxData, BoxDimensions, Boxed, BoxFormFormatTypes,
          BoxFrame, BoxMargins, BoxRatios, BoxRegion, BoxStyle, Break, Brown, Buchberger,
```

Button, ButtonBox, ButtonBoxOptions, ButtonCell, ButtonContents, ButtonData,
ButtonEvaluator, ButtonExpandable, ButtonFrame, ButtonFunction, ButtonMargins,
ButtonMinHeight, ButtonNote, ButtonNotebook, ButtonSource, ButtonStyle,
ButtonStyleMenuListing, Byte, ByteCount, ByteOrdering, C, CallPacket, Cancel,
CarmichaelLambda, Cases, Catalan, Catch, Ceiling, Cell, CellAutoOverwrite,
CellBaseline, CellBoundingBox, CellBracketOptions, CellContents, CellDingbat,
CellEditDuplicate, CellElementsBoundingBox, CellElementSpacings,
CellEvaluationDuplicate, CellFrame, CellFrameColor, CellFrameLabelMargins,
CellFrameLabels, CellFrameMargins, CellGroup, CellGroupData, CellGrouping,
CellGroupingRules, CellHorizontalScrolling, CellLabel, CellLabelAutoDelete,
CellLabelMargins, CellLabelPositioning, CellMargins, CellObject, CellOpen,
CellPasswords, CellPrint, CellSize, CellStyle, CellTags, CellularAutomaton,
Center, CForm, Character, CharacterEncoding, CharacterEncodingsPath,
CharacteristicPolynomial, CharacterRange, Characters, ChebyshevT, ChebyshevU,
Check, CheckAbort, CheckAll, CholeskyDecomposition, Chop, Circle, Clear, ClearAll,
ClearAttributes, ClebschGordan, Clip, ClipboardNotebook, ClipFill, Close, Closed,
ClosingAutoSave, CMYKColor, Coefficient, CoefficientArrays, CoefficientDomain,
CoefficientList, CofactorExpansion, Collect, ColonForm, ColorFunction,
ColorFunctionScaling, ColorOutput, ColorRules, ColorSelectorSettings, Column,
ColumnAlignments, ColumnForm, ColumnLines, ColumnsEqual, ColumnSpacings,
ColumnWidths, CommonDefaultFormatTypes, Compile, Compiled, CompiledFunction,
CompileOptimizations, Complement, CompletionsListPacket, Complex, Complexes,
ComplexExpand, ComplexInfinity, ComplexityFunction, Compose, ComposeList,
ComposeSeries, Composition, CompoundExpression, Condition, ConfigurationPath,
Conjugate, ConjugateTranspose, Connect, ConsoleMessage, ConsoleMessagePacket,
ConsolePrint, Constant, Constants, ConstrainedMax, ConstrainedMin,
ContentsBoundingBox, Context, Contexts, ContextToFilename, ContextToFileName,
Continuation, Continue, ContinuedFraction, ContourGraphics, ContourLines,
ContourPlot, Contours, ContourShading, ContourSmoothing, ContourStyle,
ConversionOptions, ConversionRules, ConvertToBitmapPacket, ConvertToPostScript,
ConvertToPostScriptPacket, Copyable, CopyDirectory, CopyFile, Cos, Cosh,
CoshIntegral, CosIntegral, Cot, Coth, Count, CounterAssignments, CounterBox,
CounterBoxOptions, CounterEvaluator, CounterFunction, CounterIncrements,
CounterStyle, CounterStyleMenuListing, CreateDirectory, CreatePalettePacket,
Cross, Csc, Csch, Cubics, Cuboid, CurrentlySpeakingPacket, Cyan, Cyclotomic,
CylindricalDecomposition, D, DampingFactor, Dashing, DataRange, Date, Debug,
DebugTag, Decimal, DeclarePackage, Decompose, Decrement, DedekindEta, Default,
DefaultColor, DefaultDuplicateCellStyle, DefaultFont, DefaultFontProperties,
DefaultFormatType, DefaultFormatTypeForStyle, DefaultInlineFormatType,
DefaultInputFormatType, DefaultNaturalLanguage, DefaultNewCellStyle,
DefaultNewInlineCellStyle, DefaultNotebook, DefaultOutputFormatType,
DefaultStyleDefinitions, DefaultTextFormatType, DefaultTextInlineFormatType,
DefaultValues, DefineExternal, Definition, Degree, DegreeLexicographic,
DegreeReverseLexicographic, Deletable, Delete, DeleteCases, DeleteContents,
DeleteDirectory, DeleteFile, DeletionWarning, DelimiterFlashTime,
DelimiterMatching, Delimiters, Denominator, DensityGraphics, DensityPlot,
DependentVariables, Depth, Derivative, Det, DiacriticalPositioning,
DiagonalMatrix, Dialog, DialogIndent, DialogLevel, DialogProlog, DialogSymbols,
DifferenceOrder, DigitBlock, DigitCharacter, DigitCount, DigitQ, Dimensions,
DiracDelta, DirectedInfinity, Direction, Directory, DirectoryName,
DirectoryStack, DisableConsolePrintPacket, DiscreteDelta, Disk, Dispatch,
Display, DisplayAnimation, DisplayEndPacket, DisplayFlushImagePacket,
DisplayForm, DisplayFunction, DisplayPacket, DisplayRules, DisplaySetSizePacket,
DisplayString, Distribute, Divide, DivideBy, DivisionFreeRowReduction,
Divisors, DivisorSigma, Do, DOSTextFormat, Dot, DoubleExponential,
DoublyInfinite, Down, DownValues, DragAndDrop, DrawHighlighted, Drop, DSolve,
Dt, DualLinearProgramming, DumpGet, DumpSave, E, EdgeForm, Edit, Editable,
EditButtonSettings, EditCellTagsSettings, EditDefinition, EditIn, Eigensystem,

Eigenvalues, Eigenvectors, Element, Eliminate, EliminationOrder, EllipticE,
EllipticExp, EllipticExpPrime, EllipticF, EllipticK, EllipticLog, EllipticNomeQ,
EllipticPi, EllipticReducedHalfPeriods, EllipticTheta, EllipticThetaPrime,
Empty, EnableConsolePrintPacket, Encode, Encoding, End, EndAdd, EndDialogPacket,
EndFrontEndInteractionPacket, EndOfFile, EndOfLine, EndOfString, EndPackage,
EngineeringForm, Enter, EnterExpressionPacket, EnterTextPacket, Environment,
Epilog, Equal, EqualColumns, EqualRows, EquatedTo, Erf, Erfc, Erfi, ErrorBox,
ErrorBoxOptions, ErrorNorm, ErrorPacket, ErrorsDialogSettings, EulerE,
EulerGamma, EulerPhi, Evaluatable, Evaluate, EvaluatePacket, EvaluationCell,
EvaluationCompletionAction, EvaluationMonitor, EvaluationNotebook, Evaluator,
EvaluatorNames, EvenQ, EventEvaluator, ExactNumberQ, ExactRootIsolation, Except,
ExcludedForms, Exists, Exit, ExitDialog, Exp, Expand, ExpandAll, ExpandDenominator,
ExpandNumerator, ExpIntegralE, ExpIntegralEi, Exponent, ExponentFunction,
ExponentPosition, ExponentStep, Export, ExportAutoReplacements, ExportPacket,
ExportString, Expression, ExpressionPacket, ExpToTrig, ExtendedGCD, Extension,
ExternalCall, ExternalDataCharacterEncoding, Extract, FaceForm, FaceGrids,
Factor, FactorComplete, Factorial, Factorial2, FactorInteger, FactorList,
FactorSquareFree, FactorSquareFreeList, FactorTerms, FactorTermsList, Fail,
False, FEDisableConsolePrintPacket, FEEnableConsolePrintPacket, Fibonacci,
File, FileByteCount, FileDate, FileFormat, FileInformation, FileName,
FileNameDialogSettings, FileNames, FileType, Find, FindFit, FindInstance,
FindList, FindMaximum, FindMinimum, FindRoot, FindSettings, First, Fit,
FitAll, FixedPoint, FixedPointList, Flat, Flatten, FlattenAt, Floor,
FlushPrintOutputPacket, Fold, FoldList, Font, FontColor, FontFamily, FontForm,
FontName, FontPostScriptName, FontProperties, FontReencoding, FontSize,
FontSlant, FontSubstitutions, FontTracking, FontVariations, FontWeight, For,
ForAll, Format, FormatRules, FormatType, FormatTypeAutoConvert, FormatValues,
FormBox, FormBoxOptions, FortranForm, Forward, ForwardBackward, Fourier,
FourierCosTransform, FourierParameters, FourierSinTransform, FourierTransform,
FractionalPart, FractionBox, FractionBoxOptions, FractionLine, Frame, FrameBox,
FrameBoxOptions, FrameLabel, FrameStyle, FrameTicks, FreeQ, FresnelC, FresnelS,
FromCharacterCode, FromContinuedFraction, FromDate, FromDigits, FrontEndExecute,
FrontEndObject, FrontEndStackSize, FrontEndToken, FrontEndTokenExecute,
FrontEndVersion, Full, FullAxes, FullDefinition, FullForm, FullGraphics,
FullOptions, FullSimplify, Function, FunctionExpand, FunctionInterpolation,
Gamma, GammaRegularized, GaussianIntegers, GaussKronrod, GaussPoints, GCD,
GegenbauerC, General, GenerateBitmapCaches, GenerateConditions, GeneratedCell,
GeneratedParameters, Generic, Get, GetBoundingBoxSizePacket, GetContext,
GetFileName, GetFrontEndOptionsDataPacket, GetLinebreakInformationPacket,
GetMenusPacket, GetPageBreakInformationPacket, Glaisher, GlobalPreferences,
GlobalSession, GoldenRatio, Goto, Gradient, Graphics, Graphics3D, GraphicsArray,
GraphicsData, GraphicsGrouping, GraphicsSpacing, Gray, GrayLevel,
Greater, GreaterEqual, Green, Grid, GridBaseline, GridBox, GridBoxOptions,
GridCreationSettings, GridDefaultElement, GridFrame, GridFrameMargins,
GridLines, GroebnerBasis, GroupPageBreakWithin, HarmonicNumber, Hash,
HashTable, Head, HeadCompose, Heads, HeldPart, HelpBrowserLookup,
HelpBrowserNotebook, HelpBrowserSettings, HermiteH, HessenbergDecomposition,
Hessian, HiddenSurface, Hold, HoldAll, HoldAllComplete, HoldComplete,
HoldFirst, HoldForm, HoldPattern, HoldRest, HomeDirectory, Horizontal,
HorizontalForm, HorizontalScrollPosition, HTMLSave, Hue, Hypergeometric0F1,
Hypergeometric0F1Regularized, Hypergeometric1F1, Hypergeometric1F1Regularized,
Hypergeometric2F1, Hypergeometric2F1Regularized, HypergeometricPFQ,
HypergeometricPFQRegularized, HypergeometricU, HyperlinkCreationSettings,
Hyphenation, HyphenationOptions, I, Identity, IdentityMatrix, If, IgnoreCase,
Im, ImageCache, ImageCacheValid, ImageMargins, ImageOffset, ImageRangeCache,
ImageRegion, ImageResolution, ImageRotated, ImageSize, Implies, Import,
ImportAutoReplacements, ImportString, In, IncludeFileExtension,
IncludeSingularTerm, Increment, Indent, IndentingNewlineSpacings,

IndentMaxFraction, Indeterminate, IndexCreationOptions, Inequality,
InexactNumberQ, InexactNumbers, Infinity, Infix, Information, Inherited,
InitializationCell, InitializationCellEvaluation, InitializationCellWarning,
InlineCounterAssignments, InlineCounterIncrements, InlineRules, Inner, Input,
InputAliases, InputAutoFormat, InputAutoReplacements, InputForm, InputGrouping,
InputNamePacket, InputNotebook, InputPacket, InputSettings, InputStream,
InputString, InputStringPacket, InputToBoxFormPacket, InputToInputForm,
InputToStandardForm, Insert, InsertionPointObject, Install, InString, Integer,
IntegerDigits, IntegerExponent, IntegerPart, IntegerQ, Integers, Integrate,
Interactive, Interlaced, InterpolatingFunction, InterpolatingPolynomial,
Interpolation, InterpolationOrder, InterpolationPoints, InterpolationPrecision,
InterpretationBox, InterpretationBoxOptions, InterpretTemplate, Interrupt,
InterruptSettings, Intersection, Interval, IntervalIntersection, IntervalMemberQ,
IntervalUnion, Inverse, InverseBetaRegularized, InverseEllipticNomeQ,
InverseErf, InverseErfc, InverseFourier, InverseFourierCosTransform,
InverseFourierSinTransform, InverseFourierTransform, InverseFunction,
InverseFunctions, InverseGammaRegularized, InverseJacobiCD, InverseJacobiCN,
InverseJacobiCS, InverseJacobiDC, InverseJacobiDN, InverseJacobiDS,
InverseJacobiNC, InverseJacobiND, InverseJacobiNS, InverseJacobiSC,
InverseJacobiSD, InverseJacobiSN, InverseLaplaceTransform, InverseSeries,
InverseWeierstrassP, InverseZTransform, JacobiAmplitude, Jacobian, JacobiCD,
JacobiCN, JacobiCS, JacobiDC, JacobiDN, JacobiDS, JacobiNC, JacobiND,
JacobiNS, JacobiP, JacobiSC, JacobiSD, JacobiSN, JacobiSymbol, JacobiZeta,
Join, JordanDecomposition, K, Khinchin, KleinInvariantJ, KroneckerDelta,
Label, LaguerreL, Language, LanguageCategory, LaplaceTransform, Larger, Last,
LatticeReduce, Launch, LayoutInformation, LCM, LeafCount, Left, LegendreP,
LegendreQ, LegendreType, Length, LerchPhi, Less, LessEqual, LetterCharacter,
LetterQ, Level, Lexicographic, LicenseID, Lighting, LightSources, Limit,
LimitsPositioning, LimitsPositioningTokens, Line, LinearProgramming,
LinearSolve, LinearSolveFunction, LineBreak, LinebreakAdjustments,
LineBreakWithin, LineForm, LineIndent, LineSpacing, LineWrapParts, linie,
LinkActivate, LinkClose, LinkConnect, LinkConnectedQ, LinkCreate, LinkError,
LinkFlush, LinkHost, LinkInterrupt, LinkLaunch, LinkMode, LinkObject, LinkOpen,
LinkOptions, LinkPatterns, LinkProtocol, LinkRead, LinkReadHeld, LinkReadyQ,
Links, LinkWrite, LinkWriteHeld, List, Listable, ListContourPlot, ListConvolve,
ListCorrelate, ListDensityPlot, Listen, ListInterpolation, ListPlay,
ListPlot, ListPlot3D, ListQ, Literal, LiteralSearch, Locked, Log, LogGamma,
LogicalExpand, LogIntegral, LongestMatch, LongForm, Loopback, LowerCaseQ,
LUBackSubstitution, LUDecomposition, MachineID, MachineName, MachineNumberQ,
MachinePrecision, MacintoshSystemPageSetup, Magenta, Magnification, MainSolve,
MakeBoxes, MakeExpression, MakeRules, MantissaExponent, Manual, Map, MapAll,
MapAt, MapIndexed, MapThread, MatchLocalNameQ, MatchLocalNames, MatchQ,
MathematicaNotation, MathieuC, MathieuCharacteristicA, MathieuCharacteristicB,
MathieuCharacteristicExponent, MathieuCPrime, MathieuS, MathieuSPrime,
MathMLForm, MathMLText, MatrixExp, MatrixForm, MatrixPower, MatrixQ, MatrixRank,
Max, MaxBend, Maximize, MaxIterations, MaxMemoryUsed, MaxPlotPoints, MaxPoints,
MaxRecursion, MaxStepFraction, MaxSteps, MaxStepSize, Mean, Median, MeijerG,
MemberQ, MemoryConstrained, MemoryInUse, MenuPacket, Mesh, MeshRange, MeshStyle,
Message, MessageList, MessageName, MessageOptions, MessagePacket, Messages,
MessagesNotebook, MetaCharacters, Method, MethodOptions, Min, Minimize,
Minors, MinRecursion, MinSize, Minus, Mod, Mode, Modular, ModularLambda,
Module, Modulus, MoebiusMu, MonomialOrder, MonteCarlo, Most, MousePointerNote,
MultiDimensional, MultilaunchWarning, MultilineFunction, Multinomial,
MultiplicativeOrder, Multiplicity, N, n30, NameQ, Names, Nand, NBernoulliB,
NDSolve, NeedCurrentFrontEndPackagePacket, NeedCurrentFrontEndSymbolsPacket,
Needs, Negative, Nest, NestedScriptRules, NestList, NestWhile, NestWhileList,
NevilleThetaC, NevilleThetaD, NevilleThetaN, NevilleThetaS, Newton,
Next, NHoldAll, NHoldFirst, NHoldRest, NIntegrate, NMaximize, NMinimize,

```
NonAssociative, NonCommutativeMultiply, NonConstants, None, NonNegative,
NonPositive, Nor, Norm, Normal, NormalGrouping, NormalSelection, NormFunction,
Not, Notebook, NotebookApply, NotebookAutoSave, NotebookClose, NotebookConvert,
NotebookConvertSettings, NotebookCreate, NotebookCreateReturnObject,
NotebookDefault, NotebookDelete, NotebookDirectory, NotebookFind,
NotebookFindReturnObject, NotebookGet, NotebookGetLayoutInformationPacket,
NotebookGetMisspellingsPacket, NotebookInformation, NotebookLocate,
NotebookObject, NotebookOpen, NotebookOpenReturnObject, NotebookPath,
NotebookPrint, NotebookPut, NotebookPutReturnObject, NotebookRead,
NotebookResetGeneratedCells, Notebooks, NotebookSave, NotebookSaveAs,
NotebookSelection, NotebookSetupLayoutInformationPacket, NotebooksMenu,
NotebookWrite, NotElement, NProduct, NProductExtraFactors, NProductFactors,
NRoots, NSolve, NSum, NSumExtraTerms, NSumTerms, Null, NullRecords,
NullSpace, NullWords, Number, NumberForm, NumberFormat, NumberMarks,
NumberMultiplier, NumberPadding, NumberPoint, NumberQ, NumberSeparator,
NumberSigns, NumberString, Numerator, NumericFunction, NumericQ, NValues,
O, OddQ, Off, Offset, OLEData, On, OneIdentity, OneStepRowReduction, Open,
OpenAppend, OpenFunctionInspectorPacket, OpenRead, OpenSpecialOptions,
OpenTemporary, OpenWrite, Operate, OptimizationLevel, Optional,
OptionInspectorSettings, OptionQ, Options, OptionsPacket, OptionValueBox,
OptionValueBoxOptions, Or, Orange, Order, OrderedQ, Ordering, Orderless,
Oscillatory, Out, Outer, OutputAutoOverwrite, OutputForm, OutputFormData,
OutputGrouping, OutputMathEditExpression, OutputNamePacket, OutputStream,
OutputToOutputForm, OutputToStandardForm, Over, Overflow, Overlaps,
Overscript, OverscriptBox, OverscriptBoxOptions, OwnValues, PaddedForm,
PadLeft, PadRight, PageBreakAbove, PageBreakBelow, PageBreakWithin,
PageFooterLines, PageFooters, PageHeaderLines, PageHeaders, PageHeight,
PageWidth, PalettePath, PaperWidth, ParagraphIndent, ParagraphSpacing,
ParameterVariables, ParametricPlot, ParametricPlot3D, ParentConnect,
ParentDirectory, ParentForm, Parenthesize, Part, Partition, PartitionsP,
PartitionsQ, PasteBoxFormInlineCells, Path, Pattern, PatternTest, Pause,
PeriodicInterpolation, Permutations, Pi, Pick, Piecewise, PiecewiseExpand,
Pink, Pivoting, PixelConstrained, Placeholder, Plain, Play, PlayRange,
Plot, Plot3D, Plot3Matrix, PlotDivision, PlotJoined, PlotLabel, PlotPoints,
PlotRange, PlotRegion, PlotStyle, Plus, Pochhammer, Point, PointForm,
PointSize, PolyGamma, Polygon, PolygonIntersections, PolyLog, PolynomialForm,
PolynomialGCD, PolynomialLCM, PolynomialMod, PolynomialQ, PolynomialQuotient,
PolynomialReduce, PolynomialRemainder, Polynomials, Position, Positive,
Postfix, PostScript, Power, PowerExpand, PowerMod, PowerModList, Precedence,
PrecedenceForm, Precision, PrecisionGoal, PreDecrement, PreferencesPath,
Prefix, PreIncrement, Prepend, PrependTo, PreserveStyleSheet, Previous,
PrimaryPlaceholder, Prime, PrimePi, PrimeQ, Primes, PrincipalValue, Print,
PrintAction, PrintForm, PrintingCopies, PrintingOptions, PrintingPageRange,
PrintingStartingPageNumber, PrintingStyleEnvironment, PrintPrecision,
PrivateCellOptions, PrivateEvaluationOptions, PrivateFontOptions,
PrivateNotebookOptions, PrivatePaths, Product, ProductLog, Prolog, PromptForm,
Protect, Protected, PseudoInverse, Purple, Put, PutAppend, QRDecomposition,
Quantile, Quartics, QuasiMonteCarlo, QuasiNewton, Quit, Quotient, RadicalBox,
RadicalBoxOptions, Random, RandomSeed, Range, Raster, RasterArray, Rational,
RationalFunctions, Rationalize, Rationals, Raw, RawData, RawMedium, Re,
Read, ReadList, ReadProtected, Real, RealBlockForm, RealDigits, Reals,
Reap, Recall, Record, RecordLists, RecordSeparators, Rectangle, Red, Reduce,
Refine, Refresh, RegularExpression, Reinstall, Release, ReleaseHold, Remove,
Removed, RenameDirectory, RenameFile, RenderAll, RenderingOptions, Repeated,
RepeatedNull, RepeatedString, Replace, ReplaceAll, ReplaceHeldPart, ReplaceList,
ReplacePart, ReplaceRepeated, Rescale, ResetDirectory, ResetMenusPacket,
Residue, Resolve, Rest, Resultant, ResumePacket, Return, ReturnExpressionPacket,
ReturnInputFormPacket, ReturnPacket, ReturnTextPacket, Reverse, RGBColor,
```

RiemannSiegelTheta, RiemannSiegelZ, Right, Root, RootReduce, Roots, RootSum,
RotateLabel, RotateLeft, RotateRight, Round, Row, RowAlignments, RowBox,
RowLines, RowMinHeight, RowReduce, RowsEqual, RowSpacings, RSolve, Rule,
RuleCondition, RuleDelayed, RuleForm, RulerUnits, Run, RunThrough, SameQ,
SameTest, SampleDepth, SampledSoundFunction, SampledSoundList, SampleRate, Save,
Saveable, SaveAutoDelete, Scaled, Scan, SchurDecomposition, ScientificForm,
ScreenRectangle, ScreenStyleEnvironment, ScriptBaselineShifts, ScriptLevel,
ScriptMinSize, ScriptRules, ScriptSizeMultipliers, ScrollingOptions,
ScrollPosition, Sec, Sech, Second, SectionGrouping, SeedRandom, Select,
Selectable, SelectedNotebook, Selection, SelectionAnimate, SelectionCell,
SelectionCellCreateCell, SelectionCellDefaultStyle, SelectionCellParentStyle,
SelectionCreateCell, SelectionDuplicateCell, SelectionEvaluate,
SelectionEvaluateCreateCell, SelectionMove, SelectionSetStyle,
SelectionStrategy, SendFontInformationToKernel, Sequence, SequenceForm,
SequenceHold, SequenceLimit, Series, SeriesCoefficient, SeriesData, SessionTime,
Set, SetAccuracy, SetAttributes, SetBoxFormNamesPacket, SetDelayed,
SetDirectory, SetEvaluationNotebook, SetFileDate, SetFileLoadingContext,
SetNotebookStatusLine, SetOptions, SetOptionsPacket, SetPrecision,
SetSelectedNotebook, SetSpeechParametersPacket, SetStreamPosition, SetValue,
Shading, Shallow, Share, Short, ShortestMatch, Show, ShowAutoStyles,
ShowCellBracket, ShowCellLabel, ShowCellTags, ShowClosedCellArea,
ShowContents, ShowCursorTracker, ShowGroupOpenCloseIcon, ShowPageBreaks,
ShowSelection, ShowShortBoxForm, ShowSpecialCharacters, ShowStringCharacters,
ShrinkWrapBoundingBox, Sign, Signature, SignPadding, Simplify, Sin,
SingleLetterItalics, SingularityDepth, SingularValueDecomposition,
SingularValueList, SingularValues, Sinh, SinhIntegral, SinIntegral,
SixJSymbol, Skeleton, Skip, Slot, SlotSequence, Smaller, Socket, Solve,
SolveAlways, SolveDelayed, Sort, Sound, SoundAndGraphics, Sow, Space,
SpaceForm, SpanAdjustments, SpanCharacterRounding, SpanLineThickness,
SpanMaxSize, SpanMinSize, SpanningCharacters, SpanSymmetric, Sparse,
SparseArray, SpeakTextPacket, SpellingCorrection, SpellingDictionaries,
SpellingDictionariesPath, SpellingOptions, SpellingSuggestionsPacket,
Spherical, SphericalHarmonicY, SphericalRegion, Splice, Split, Sqrt,
SqrtBox, SqrtBoxOptions, Stack, StackBegin, StackComplete, StackInhibit,
StandardDeviation, StandardForm, StartingStepSize, StartOfLine, StartOfString,
StartupSound, StepMonitor, StieltjesGamma, StirlingS1, StirlingS2, StoppingTest,
StreamPosition, Streams, String, StringBreak, StringByteCount, StringCases,
StringCount, StringDrop, StringExpression, StringForm, StringFreeQ,
StringInsert, StringJoin, StringLength, StringMatchQ, StringPosition,
StringQ, StringReplace, StringReplaceList, StringReplacePart, StringReverse,
StringSkeleton, StringSplit, StringTake, StringToStream, StripBoxes,
StripWrapperBoxes, StructuredSelection, StruveH, StruveL, Stub, StyleBox,
StyleBoxAutoDelete, StyleBoxOptions, StyleData, StyleDefinitions, StyleForm,
StyleMenuListing, StyleNameDialogSettings, StylePrint, StyleSheetPath,
Subresultants, Subscript, SubscriptBox, SubscriptBoxOptions, Subscripted,
Subsets, Subsuperscript, SubsuperscriptBox, SubsuperscriptBoxOptions,
Subtract, SubtractFrom, SubValues, SugarCube, Sum, Superscript, SuperscriptBox,
SuperscriptBoxOptions, SurfaceColor, SurfaceGraphics, SuspendPacket, Switch,
SylvesterMatrix, Symbol, SymbolName, Syntax, SyntaxForm, SyntaxLength,
SyntaxPacket, SyntaxQ, SystemException, SystemHelpPath, SystemStub, Tab,
TabFilling, Table, TableAlignments, TableDepth, TableDirections, TableForm,
TableHeadings, TableSpacing, TabSpacings, TagBox, TagBoxOptions, TaggingRules,
TagSet, TagSetDelayed, TagStyle, TagUnset, Take, Tan, Tanh, TargetFunctions,
Temporary, TemporaryVariable, TensorQ, TensorRank, TeXForm, TeXSave, Text,
TextAlignment, TextBoundingBox, TextData, TextForm, TextJustification,
TextLine, TextPacket, TextParagraph, TextRendering, TextStyle, Thickness,
ThisLink, Thread, ThreeJSymbol, Through, Throw, Ticks, TimeConstrained,
TimeConstraint, Times, TimesBy, TimeUsed, TimeVariable, TimeZone, Timing,

```
TitleGrouping, ToBoxes, ToCharacterCode, ToColor, ToDate, ToExpression,
ToFileName, Together, Toggle, ToggleFalse, ToHeldExpression, TokenWords,
Tolerance, ToLowerCase, TooBig, Top, ToRadicals, ToRules, ToString, Total,
TotalHeight, TotalWidth, ToUpperCase, Tr, Trace, TraceAbove, TraceAction,
TraceBackward, TraceDepth, TraceDialog, TraceForward, TraceInternal, TraceLevel,
TraceOff, TraceOn, TraceOriginal, TracePrint, TraceScan, TraditionalForm,
TraditionalFunctionNotation, TraditionalNotation, TraditionalOrder,
TransformationFunctions, TransparentColor, Transpose, Trapezoidal, TreeForm,
Trig, TrigExpand, TrigFactor, TrigFactorList, TrigReduce, TrigToExp, True,
TrueQ, Tuples, UnAlias, Underflow, Underoverscript, UnderoverscriptBox,
UnderoverscriptBoxOptions, Underscript, UnderscriptBox, UnderscriptBoxOptions,
UndocumentedTestFEParserPacket, UndocumentedTestGetSelectionPacket, Unequal,
Unevaluated, Uninstall, Union, Unique, UnitStep, Unprotect, UnsameQ, Unset, Up,
Update, UpperCaseQ, UpSet, UpSetDelayed, UpValues, URL, Using, V2Get, Value,
ValueBox, ValueBoxOptions, ValueForm, ValueQ, Variables, Variance, VectorQ,
Verbatim, Verbose, VerboseConvertToPostScriptPacket, VerifyConvergence,
VerifySolutions, Version, VersionNumber, Vertical, VerticalForm, ViewCenter,
ViewPoint, ViewPointSelectorSettings, ViewVertical, Visible, VisibleCell,
WeierstrassHalfPeriods, WeierstrassInvariants, WeierstrassP, WeierstrassPPrime,
WeierstrassSigma, WeierstrassZeta, Which, While, White, Whitespace,
WhitespaceCharacter, WindowClickSelect, WindowElements, WindowFloating,
WindowFrame, WindowFrameElements, WindowMargins, WindowMovable, WindowSize,
WindowTitle, WindowToolbars, WindowWidth, With, Word, WordBoundary, WordCharacter,
WordSearch, WordSeparators, WorkingPrecision, Write, WriteString, WynnDegree,
x, XMLElement, XMLObject, Xor, y, Yellow, ZeroTest, ZeroWidthTimes, Zeta,
ZTransform, $, $Aborted, $AddOnsDirectory, $AnimationDisplayFunction,
$AnimationFunction, $Assumptions, $BaseDirectory, $BatchInput, $BatchOutput,
$BoxForms, $ByteOrdering, $CharacterEncoding, $CommandLine, $ConditionHold,
$Context, $ContextPath, $CreationDate, $CurrentLink, $DefaultFont,
$DefaultPath, $Display, $DisplayFunction, $DumpDates, $DumpSupported,
$Echo, $Epilog, $ExportEncodings, $ExportFormats, $Failed, $FormatType,
$FrontEnd, $HistoryLength, $HomeDirectory, $IgnoreEOF, $ImportEncodings,
$ImportFormats, $InitialDirectory, $Input, $Inspector, $InstallationDate,
$InstallationDirectory, $InterfaceEnvironment, $IterationLimit, $Language,
$LaunchDirectory, $LicenseExpirationDate, $LicenseID, $LicenseProcesses,
$LicenseServer, $Line, $Linked, $LinkSupported, $MachineDomain, $MachineEpsilon,
$MachineID, $MachineName, $MachinePrecision, $MachineType, $MaxExtraPrecision,
$MaxLicenseProcesses, $MaxMachineNumber, $MaxNumber, $MaxPiecewiseCases,
$MaxPrecision, $MaxRootDegree, $MessageList, $MessagePrePrint, $Messages,
$MinMachineNumber, $MinNumber, $MinPrecision, $ModuleNumber, $NetworkLicense,
$NewMessage, $NewSymbol, $Notebooks, $NumberMarks, $Off, $OperatingSystem,
$Output, $OutputForms, $Packages, $ParentLink, $ParentProcessID, $PasswordFile,
$Path, $PathnameSeparator, $PipeSupported, $Post, $Pre, $PreferencesDirectory,
$PrePrint, $PreRead, $PrintForms, $PrintLiteral, $ProcessID, $ProcessorType,
$ProductInformation, $ProgramName, $PSDirectDisplay, $RandomState,
$RasterFunction, $RecursionLimit, $ReleaseNumber, $RootDirectory,
$SessionID, $SetParentLink, $SoundDisplay, $SoundDisplayFunction,
$SuppressInputFormHeads, $SyntaxHandler, $System, $SystemCharacterEncoding,
$SystemID, $TemporaryPrefix, $TextStyle, $TimeUnit, $TopDirectory, $TraceOff,
$TraceOn, $TracePattern, $TracePostAction, $TracePreAction, $Urgent,
$UserAddOnsDirectory, $UserBaseDirectory, $UserName, $Version, $VersionNumber}
```

5.7. Verschiedene Möglichkeiten, eine Funktion zu codieren:
▮ Différentes manières de coder une fonction:

5.7.1. Die Definition einer Funktion:
■ La définition d'une fonction:

Beispiel: Die Funktion "f(x) = sin(x) * cos(2x)" soll definiert werden.
Der Unterstrich "_" gibt die Funktionsvariable an. ":=" bedeutet "definiere":
■ Exemple: Il faut définir la fonction "f(x) = sin(x) * cos(2x)". Le trait "_" indique la variable de la fonction. ":=" signifie "définis":

```
In[50]:= f[x_]:= Sin[x] Cos[2 x]
```

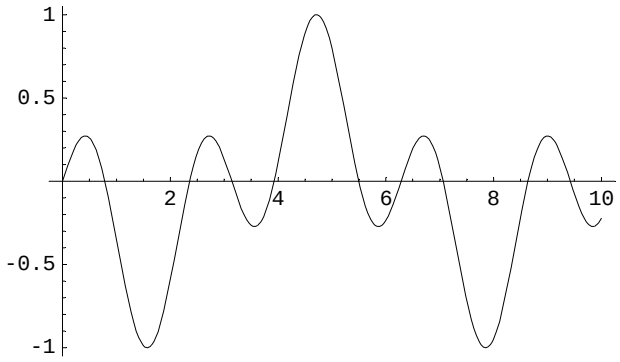
Ausgabe zur Kontrolle:
■ Sortie de contrôle:

```
In[51]:= f[x]

Out[51]= Cos[2 x] Sin[x]
```

Anwendung: ■ Application:

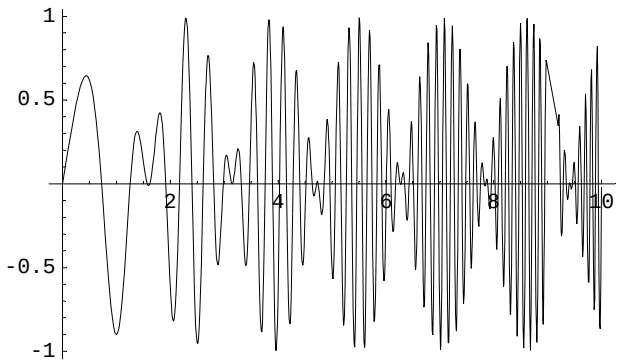
```
In[52]:= f[Pi]

Out[52]= 0
```

```
In[53]:= Plot[f[x],{x, 0, 10}];
```



Kurz in einer Zelle eine andere Funktion:
■ Brièvement une autre fonction dans une cellule:

```
In[54]:= f1[x_]:= Sin[2x] Cos[3 x^2];
         Plot[f1[x],{x, 0, 10}];
```

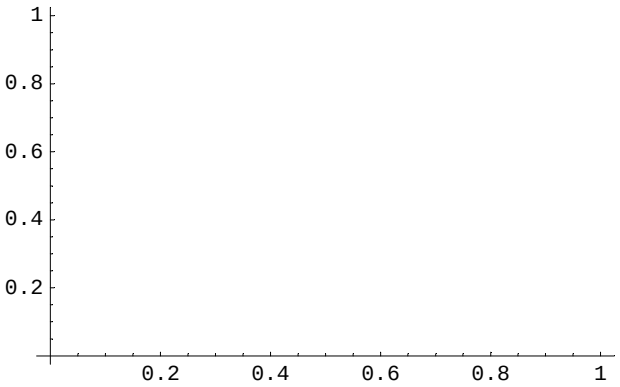


Zum Vergleich, wie es **nicht** funktioniert:

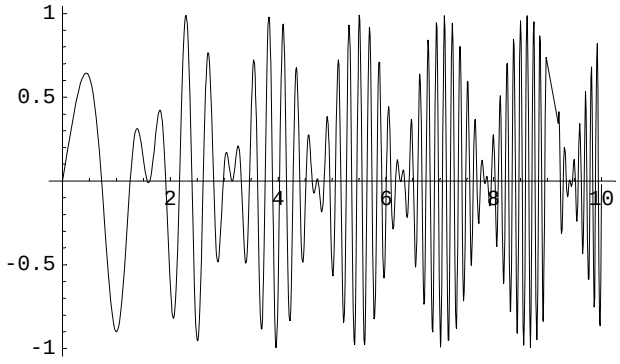
■ En tant que comparaison: Voici comme ça **ne** fonctionne **pas**:

In[56]:=
```
g[x]:= Sin[2x] Cos[3 x^2];
Plot[g[x],{x, 0, 10}];
```

```
Plot::plnr : g[x] is not a machine-size real number at x = 4.1666666666666667`*^-7. Mehr…

Plot::plnr : g[x] is not a machine-size real number at x = 0.40566991572915795`. Mehr…

Plot::plnr : g[x] is not a machine-size real number at x = 0.848087998593737`. Mehr…

General::stop : Further output of Plot::plnr will be suppressed during this calculation. Mehr…
```

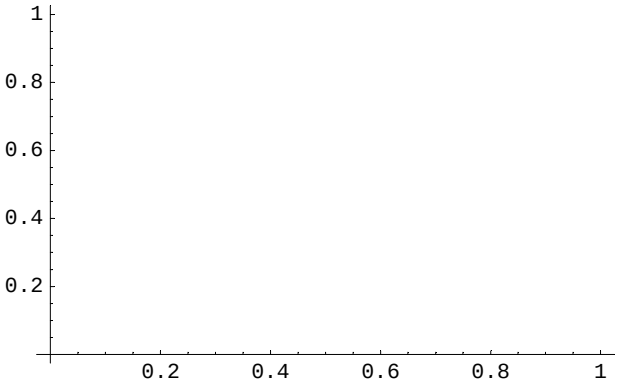


In[58]:=
```
g1[x_]= Sin[2x] Cos[3 x^2];
Plot[g1[x],{x, 0, 10}];
```



```
In[60]:= g2[x]= Sin[2x] Cos[3 x^2];
         Plot[g2[x],{x, 0, 10}];
```

```
Plot::plnr : g2[x] is not a machine-size real number at x = 4.1666666666666667`*^-7. Mehr…

Plot::plnr : g2[x] is not a machine-size real number at x = 0.40566991572915795`. Mehr…

Plot::plnr : g2[x] is not a machine-size real number at x = 0.848087998593737`. Mehr…

General::stop : Further output of Plot::plnr will be suppressed during this calculation. Mehr…
```



5.7.2. Die Prefix-Notation:
▮ La notation préfixe:

Beispiel: Die Funktion "N" mit Klammern "[...]":
▮ Exemple: La fonction "N" avec des parenthèses "[...]":

```
In[62]:= N[Sqrt[2]/2]

Out[62]= 0.707107
```

5.7.3. Die Postfix-Notation:
▮ La notation postfixe:

Beispiel: "N" kommt hinten, abgetrennt durch "//":
▮ Exemple: "N" se place derrière, séparé par "//":

```
In[63]:= Sqrt[2]/2 // N

Out[63]= 0.707107
```

Damit lassen sich "Pipes" bilden. d.h. mehrere Funktionen nacheinander ausführen. Beispiele:
▮ Ainsi on peut former des "pipes", c'est-à-dire générer plusieurs fonctions l'une après l'autre. Exemples:

```
In[64]:= "W*" // Names

Out[64]= {WeierstrassHalfPeriods, WeierstrassInvariants, WeierstrassP, WeierstrassPPrime,
          WeierstrassSigma, WeierstrassZeta, Which, While, White, Whitespace,
          WhitespaceCharacter, WindowClickSelect, WindowElements, WindowFloating,
          WindowFrame, WindowFrameElements, WindowMargins, WindowMovable, WindowSize,
          WindowTitle, WindowToolbars, WindowWidth, With, Word, WordBoundary, WordCharacter,
          WordSearch, WordSeparators, WorkingPrecision, Write, WriteString, WynnDegree}
```

```
In[65]:= "W*" // Names // Length
Out[65]= 32

In[66]:= x = Pi/2
Out[66]= π/2

In[67]:= Sin[x]
Out[67]= 1

In[68]:= Sin[x] // Sin // N
Out[68]= 0.841471

In[69]:= Sin[x] // Sin // Sin // N
Out[69]= 0.745624

In[70]:= Sin[x] // Sin // Sin // Sin // Sin // N
Out[70]= 0.627572

In[71]:= N[Sin[Sin[Sin[Sin[Sin[x]]]]]]
Out[71]= 0.627572
```

5.8. Output unterdrücken
▮ Supprimer l'output

5.8.1. Mit ";" am Schluss
▮ Mettre ";" à la fin

Beispiel: ▮ Exemple:

```
In[72]:= Sin[4.08]
Out[72]= -0.806618

In[73]:= Sin[3.96];
```

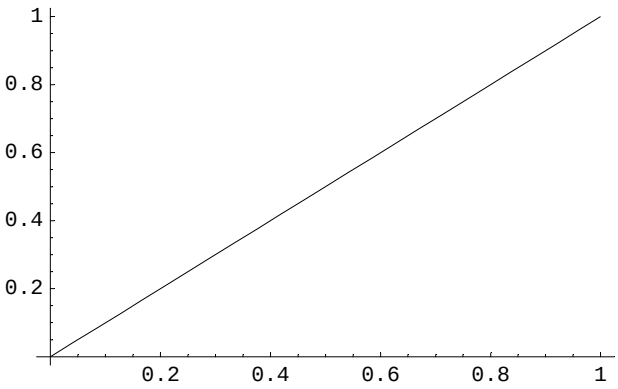
Abrufen: ▮ Appeler:

```
In[74]:= %
Out[74]= -0.730058
```

5.8.2. Diverse Befehle in einer Zelle. Welcher Output kommt?
▮ Différents ordres dans une cellule. Quel sera l'output?

Beispiele: ▮ Exemples:

```
In[75]:= a=1; b=2; c=3; d=4

Out[75]= 4

In[76]:= a=1; b=2; c=3; d=4;

In[77]:= a=1; b=2; Print[b]; c=3; d=4;

         2

In[78]:= a=1; b=2; Print[b]; c=3; d=4

         2

Out[78]= 4

In[79]:= a=1; b=2; Plot[x,{x,0,1}]; c=3; d=4
```




```
Out[79]= 4
```

5.9. Timing - Rechenzeit messen
■ Timing - mesurer le temps de calcul

Mit "Timing" wird die CPU-Zeit gemessen. Beispiele:

■ Avec "Timing" on mesure le temps CPU. Exemple:

```
In[80]:= Timing[N[Pi,500]]

Out[80]= {0. Second,
          3.14159265358979323846264338327950288419716939937510582097494459230781640628620899
           86280348253421170679821480865132823066470938446095505822317253594081284811174502
           84102701938521105559644622948954930381964428810975665933446128475648233786783165
           27120190914564856692346034861045432664821339360726024914127372458700660631558817
           48815209209628292540917153643678925903600113305305488204665213841469519415116094
           33057270365759591953092186117381932611793105118548074462379962749567351885752724
           8912279381830119491}
```

Ohne den Output der Rechnung:

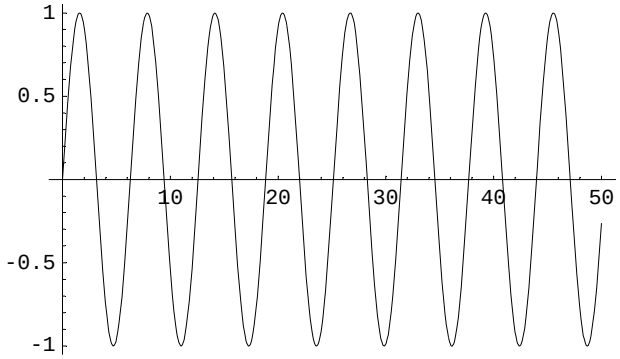
■ Sans l'output du calcul:

```
In[81]:= Timing[N[Pi,500];]

Out[81]= {0. Second, Null}
```

In[82]:= **Timing[N[Pi,2000];]**

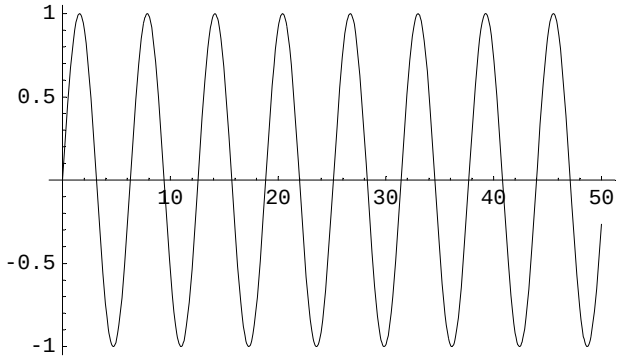
Out[82]= {0. Second, Null}

In[83]:= **Plot[Sin[x],{x, 0, 50}] // Timing**



Out[83]= {0. Second, - Graphics -}

In[84]:= **Plot[Sin[x],{x, 0, 50}]; // Timing**



Out[84]= {0. Second, Null}

5.10 Abbruch und Unterbruch
▮ Rupture et interruption

Während der Rechnungsausführung hast Du im Menu unter ("Action") neu "Kernel" - "Interrupt" die Möglichkeit, mit "Abort" abzubrechen oder mit "Interrupt Calculation" zu unterbrechen.
Der Abbruch ist auf NeXT auch möglich über die Tastatur mit "Command" und "." zusammen gedrückt. U.s.w..
Probiere aus (z.B. 5000 Fakultät....):

▮ Dans le menu sous ("Action") nouveau "Kernel" - "Interrupt" tu as la possibilité d'interrompre le calcul par "Interrupt Calculation" ou définitivement par "Abort".
Sur NeXT l'interruption définitive est aussi possible avec les touches "Command" et ".", pressées simultanément. Etc..
Essaie (p.ex. 5000 factoriel)

In[85]:= **5000!**

Out[85]= 42285779266055435222010642002335844053907866746266467488497824021813580527081082006908990478717063875370847466573006854458784860666838127363372108937727876312793⁄90363058462160643904478986982239871929708896211612652968321775500399242196837031⁄46907264472878789790404754884162215226671928410969236910449565971736352948400223⁄

840381120644820230857671104502306174894755428309761781724040805324809927809328781:
405548619936454829121187625824880218917397790005021321259804363924462646076051131:
588465951086754705858339246552255890354744359883473831789880346330084586315102091:
091509935653820010933047965742556741930917055172805200236075085991197635228755901:
790204336974312350691683121192449597155626740752146219898623308862599830285986481:
575787494459631152869708867100462684236481789899054546908613916132183441741488071:
186234448114831209490361196546872767755617886828720269104814092456410341835975601:
427645816151317857590166107178254415698088335937272999560337137120047104943765621:
911424886053352994996423006999722049181201008190594391406750532650047755338508991:
097945101551091486907004407119572336026243368132330218709287699196806656569752791:
042225826784156108337642578103262920268721107027468139435112860150232619064995911:
718973641763784364912197091098409445148953589591038041769419566578348220717491051:
512752639148381172052604826965162642710094919393332661030104360530459117014557201:
958471435721948246686793467375904872268133410207860903657108806376616249749507 4:
131070774016821805859455264451714092774692300626975113460441745679467358287822611:
629584248675157379172942724178783105429858245117575511884506574424827574660800231:
858837849239624736876150701576772589832112863229553704490251638792512759084179171:
446404669135310473479844649961545955420139963173574763017400367961929199421907621:
895445656261767041799538161133387312823511534152581309087915883638351664797225911:
294427065355714251173732380723263295812179791667969232968709692390100325557478901:
550998074870610472306461959849552396576122086738665141716993075576918979026751571:
342075864796345338446835085965490727326321910504064289713096224505162064669468091:
886991712212740450402068492326624176013291022786668727030528470945252682549661771:
724996452066998369259106908940826374010434983715911264558222806063613941153443161:
771769934353664284928294436414769615881993661388255577487709937004594753907845141:
903443452117456059403991626844469766182138747070532555957793319646099666214537751:
649354741697085623892147732228655071824904300161861421927604523076706211429617671:
274704123616107220009743758647492753665149532164780849075146330071016691313420661:
288256261828386583698363210876071042751607334834778841479673242708041086076184121:
818883071150989821353384066106521470870468747609954274736735094515535997690403671:
353385551052571682650317682405743993414862392331981432579182193321898940450865011:
361099809838399311099635598132800104973158859631213185380120504678764291066936561:
004373056334319848790489985247012933007893445328681566797628804955328463860201331:
480265279836946393384995675049993707814746561543438930431384237878981847802886001:
997108869563298834771186312238278596365311513237793137364739742936941149902875191:
722279995451826154882989511519266821124513553184722099904353559498872999220350621:
039816011086376236539782172380237846650673624510635034423187315338308212043804711:
099941922782103974755271741604389016972396130554937184483611980356589606202500901:
936643993601720073836133544050943290724765189095025077246758419894122246593921631:
116352038147362479528539732089309533421910635702805576629720156556510767780805931:
345363112182956179288767300280245093212277885296841820826177847695564498038569121:
757873726780409591587117339711031652326780607981276092461735041201826668742628051:
385275843979167609007743380748420751185119102921960339376280986753665085212869251:
532153678793252188257410186613705432897373586272537017855880663985135038694403961:
049282588201804191780736496938858025977583988920143897471654659735108526057062341:
402069637065660129535734043582961473427275805630839510667375349259659518575646931:
972321827578000325059389530382053969755887051154307392082742244051629970873959971:
684612062466290981123680125798912848025050940289169597650793954371913113793144271:
405135599630375642214527294341797246187597964074239147838993541565834716156858491:
903677305661135833367087548900413091981676330749041510337597307246885839246941711:
155482957307506185058815819595289926602256269034395733134506669729521152306686961:
227920947779974336574472673471408928071411283888082693377378077293104110767513631:
947620061085804059601963905801576100233746386935222838580143495717812558144586291:
300424794040657368598620079146045902554139299500880447103847589903265480973381661:
940500085452723713571394902463820308668541802838317527668064278489561005755859991:
171896678644915406357001449719424987892085973125427556751457520639911815073639741:
831024907938417256534218942767691165981534300846370877695102954151365517346750541:
015239706042571746001089968440498845985477977905031632568489156557231006499726491:

872148080018177035770150298300887948724388771888441683303470872323950537764223295:
440957732191375823716739247042167230022568831357792303946889006624661825326584905:
724406767024939579697217467485562998183149665611743997680482094166257463879660305:
517127492511922636761533752438165621733077165012952098875485646713186260238761995:
643348679615144083289020610828331808912213258536828564699160079521051669604516955:
430614212305743006877217407155473217957577017595967640563812729153867513698712395:
557054235099922860597546996218619553135413213912643667690046542999681168055073785:
667706659880270629725020018828458861453443687714553613044144656133690928627482765:
981946836480550952968681758714859972973082332924094777085275279923304892719633315:
475156331119274615038921929061678060790138345113706630068437626719988551514368125:
661373199121032354697867564212106248990055535640229243458312642310383634167817195:
908354140411717740185950606674198348143345444247191436828225654380047860390575925:
241707180267064687545421162695874679539854078446465414038175114996527362112354085:
801669902801490332251394608326681709307138688265499773742861277894177847526813285:
371818759103642140881783220739808059714203285309721443041845459183002833408705785:
313828497328376128618292713674516189736620723739613279094498401415440830407440535:
930675407671261825475971308434703113898156953659717885640227506742374003236218505:
094765267521941901241387478279883426470873616812485384444012772521050072279315855:
309627912113116016777207795257261380024068442188545353712134190223637968401238525:
552886071899677256942274333239485950755708390618774501596521844149981554761075485:
008054192318436948191732631430603548399790783307267636729090980772823573558543480:
226006747253709778546456776118180736742436739176986375807214585979148503370059295:
949637933691002834445580898380540176354037371330193112930809582876121073803748005:
660269767842888358265737486556785868822015143046249655995760379768685318192365805:
646919958407184549360692216977613754266223965864498977092147813470912791746087165:
302208219814346542450657312626830895790310128933607886441072301848054003731360145:
216229159146992019884148290014414312800903102107833305090238435726779416177246875:
341150359870000315109281570033108172741562468043297720507045045668389862630170295:
893011453644774168567325123303764778817490360525726055206843706161165397551325415:
369303867783267208227323664249206432363089268768826650939691861683271739757479555:
299324240618699242036378192948536809803525633109244821526927621911625914588639365:
770346534803678871261333671169682264509149970554485212597518700847200256746587525:
403932061045903070039438252019383102480929019684602472171298321628237994627125365:
635997189837442509912067368838373829965389203066284307454755907423534527402921165:
060913463276847495220460104095756073481551016772031875800892244947529220310938415:
661588823584993931745149914395557357641584185479831702428523965451087525425464775:
729459523036094641654199779794713680634491599877240917644313737117854221074057215:
211668686921532404900803842059211926228754408982614789081236989563670808046876285:
524499897440855677969456909042340530355943524640751677873953113928698614347227575:
214494689189609329437547674123490779275433834941232306007876761008994915612693405:
389211483702171933876178233703589258171128695634500013676198971454009866434619225:
197676975930010555225198913002123021780831934330880446592954552165911855939202575:
978112295206535736291447840494647456500311549807205658036066738088957274646437545:
280558193222993050892878068745374013271002744283179253550034515366931721120882275:
603942809788645727306979971285649576934354004030728440581746648376658498039958965:
424337018345415172028533781090411312446243290335396429665110948283688458012758875:
012931560992250445181254601132749860144704377573138810013192761246761166148335285:
935557503106018449788994378274613854651708241613167681463911870000812845144341405:
673998543007277230375811161351109435561489632392975084638315293026358253536178485:
375585196669499722519355159538072078386151421302844500517952397609684331982925985:
921623223582396390262548856855875458198371559008447860086745945709118128793228225:
205176750937186611001319362584522349394982951119928083786052350641276933754813065:
095942644634250776011473342091391285416281831722621437830629624081493919971875285:
106367348876678481602342743230027158192404187686545826519361990687336892886715135:
384024548611098248200448272179949665871225717442904491678119482416563156036034738:
333176651212180527807959582202983306119451640194133155503796629802153576807311245:
530585915969709973988055743550083279071844959752353594644354789680372126344509425:
307025399510286445823745467776101355691621230975228615205321399874567303412767655:

```
033696366823066655520515624911325289261558638685031008491809205076806582659152765:
163719928694258350604859732273949286080260640627521341007801815105623787926212035:
942478183343943387720639580111580908419079432019517823574019054659599028961771175:
761952703540511937272297222484420804400987503694112776865930221330106250318620855:
145076421052980508837197986052557750303949606158442838846866137510968441567309835:
807939434957001302926517795712062555585195131357402989758928347552533440985891145:
006944493084328740050155543325877938950802411285387587259451364008383249444713465:
436826148195406004114845870234072926697740631325878634790667698266181501256117695:
227571529124916482170237288441635760099685110093941144467762818600707227852231495:
410485643962557968082212899357992622085538892211647652208503677064764914961337895:
353761537391569177822237744837614120253342622508007300513473422771427333106345975:
180324024422695045809053932668910361938199883884403623179528243549536248967073415:
559480676885153210730644760778596286278522836572445643064490962775171726569542385:
392941958409527253281659572534531428389629894005886539486824117113929627356938975:
348293585465027868943701479838382600205820885351707321628872521452220526596961495:
621478848401290045077372524246050743396608181829602960191963141249985384220176955:
110361380561701016357743542531148669369994130940908368220071936435111978592782495:
349147705218722654610919959726943915240046790117360252103005188608033708484011485:
102463512882639861700818048883807502035214483487408491547187144788578095745154995:
505005070789428842888410027877777455981132319940624176532148686316581736774410085:
406343695998951928831086912451786634255935345824258941139051646944037756266582155:
778459368299096797545483505104736337708391510338546396027534864016352046338843425:
346714935641429160856846724874244782055113759168236472297793612971080302530934475:
811552773754045896899035480805830938126732359356309854656437620938537105280834465:
071890760033887816180198532737594985669167047034484383635034163683252266403224174:
519476678140428319327482851882140344319384445475456765253419659194332132585432275:
007075903856523966822717130000918912205084518526151462793771759752885297863793175:
112125295294433237579100729090017035587637986124802814630939443919169501293363155:
045285163539312868586427437296109446101235604877439863299611899755965996608749045:
927116768526867533599129758320908955329640952364011606007849500537789278375014735:
441221237779077271341466474044898375894876754232945468993542203416699613666989765:
529978580795899055864050388507083137333076839766882463680992355219727241831735125:
764618911238048588311556947788810175970897768214964434031792444308517030369221415:
376211943886419895083603393064590373618429370287107583219666075461137610763625435:
928614316242890754021082233620012309384737312220374269033838579928678572939434165:
828705376337409193818463226113174093427811791889164244751354347844604054945537985:
345561633538158684416920545186989194341753866639003357567656032643637679067216265:
620330878425545157208117246381251512669846685887209013144861632560461019513371815:
458524998817662992514214501471020619319037367138034766343102970522241478503018825:
751063474462412587079373390850957577243167350668850942087615361644404437558601605:
625837091300574162065273670941888667964570550744724714137001968165215954380698515:
599948336135752132210613188477192664194239535141223354674646149174301347586603735:
381765326045574029254722793602889261893858996956568767830171868739887638872625972:
743976062813263446647679413679726184933395074665820441677989806604203937116666335:
669628256934909734839115586900485603251221924153426852236931603676549104770273355:
215401431683388729684054432969676840360731824353362248654338235981235441675146085:
340781166661858781733980624199254577853462678039039937557802759942957205281043775:
566697939683810934111895947576622019121753509363898546528307869237066251232368435:
902355876362283246571611837140788076611621795178879728018415720196390844002690375:
450381192797170314489871815031319992111563908303017288012610642062005359240278275:
739391802639171772013612598477693398064706376302260888535993759507908878908179185:
021957680333819686051204871076108748984115687401599530206390981389932610955388685:
264084012160831040525974539251576403732889086736948366404734622708560040891610785:
222194340517979455015534768296685532009750190558141991459112418150106225562741125:
315713773586971943741308220273838438159406385713879133375923623304404534872330475:
240668784133333047898995255221468847973813568083995644533005222551320155267768895:
541277032927867082749004117207666311278363815234354768166312118908686499138023625:
817752759460612118133420547918016192203469127603819005280123439735982704614998145:
```

```
51132461819565852823204465827008206493468025155651127282208381156319225650994520:
12226666032260593962470197076685803962869755511151899730490850517587653067857580:
00660424066894170620303846785860257370634352599586885088679654004465187790208942:
93515321731675011373803146603464242948907632222813376329991964133650202862728926:
80875600366137706074635755150790879820997226601304729078257469081754519524055737:
91313113170617323191598673971588373108168916968657704150695512947652386134815766:
96758036476200528906022274453174430549840286304885086955776152865032608094116068:
85706988947620464785008843039731074127741919616974505171103290828152012738886634:
22631492147090220016940636504812047036016738602290671629816411198202268607961324:
73955005756756456820475461904042301106237136739599567894088470597685951450501724:
15177460173514309909726155093783347200000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000000000000000000000000000000000000000000000000000000000000000000000000000000:
00000
```

Braucht wohl die Bildschirmausgabe oder die Rechnung mehr Zeit? Probiere aus!

Est-ce la sortie ou le calcul qui dure plus de temps? Essaie!

In[86]:= **1000! // Timing**

Out[86]= {0. Second,
4023872600770937735437024339230039857193748642107146325437999104299385123986290205920442084869694048004799886101971960586316668729948085589013238296699445909974245040870737599188236277271887325197795059509952761208749754624970436014182780946464962910563938874378864873371191810458257836478499770124766328898359557354325131853239584630755574091142624174743493475534286465766116677973966688202912073791438537195882498081268678383745597317461360853795345242215865932019280908782973084313928444032812315586110369768013573042161687476096758713483120254785893276716913244842623613141250878020800026168315102734182797770478463586817016436502415369139828126481021309276124489635992870511496497541990934222156683257208082133318611681155361583654698404670897560290095053761647584772842188967964624494516076535340819890138544248798495995331910172335555660213945039973628075013783761530712776192684903435262520001588853514733161170210396817592151090778801939317811419454525722386554146106289218796022383897147608850627686296714667469756291123408243920816015378088989396451826324367161676217916890977991190375403127462228998800519544441428201218736174599264295658174662830295557029902432415318162721046583203678690611726015878352075151628422554026517048330422614397428693306169089796848259012545832716822645806652676995865268227280707578139185817888965220816434834482599326604336766017699961283186078838615027946595513115655203609398818061213855860030143569452722420634463179746059468257310379008402443243846565724501440282188525247093519062092902313649327349756551395872055965422874977401141334696271542284586237738753823048386568897646192738381490014076731044664025989949022222176590433990188601856652648506179970235619389701786004081188972991831102117122984590164192106888438712185564612496079872290851929681937238864261483965738229112312502418664935314397013742853192664987533721894069428143411852015801412334482801505139969429015348307764456909907315243327828826986460278986432113908350621709500259738986355427719674282224875758676575234422020757363056949882508796892816275384886339690995982628095612145099487170124451646126037902930912088908694202851064018215439945715680594187274899809425474217358240106367740459574178516082923013535808184009699637252423056085590370062427124341690900415369010593398383577793941097002775347200000000000

```
$AnimationDisplayFunction $MinMachineNumber
$AnimationFunction        $MinNumber
$Assumptions              $MinPrecision
$BaseDirectory            $ModuleNumber
$BatchInput               $NetworkLicense
$BatchOutput              $NewMessage
$BoxForms                 $NewSymbol
$ByteOrdering             $Notebooks
$CharacterEncoding        $NumberMarks
$CommandLine              $Off
$ConditionHold            $OperatingSystem
$Context                  $Output
$ContextPath              $OutputForms
$CreationDate             $Packages
$CurrentLink              $ParentLink
$DefaultFont              $ParentProcessID
$DefaultPath              $PasswordFile
$Display                  $Path
$DisplayFunction          $PathnameSeparator
$DumpDates                $PipeSupported
$DumpSupported            $Post
$Echo                     $Pre
$Epilog                   $PreferencesDirectory
$ExportEncodings          $PrePrint
$ExportFormats            $PreRead
$Failed                   $PrintForms
$FormatType               $PrintLiteral
$FrontEnd                 $ProcessID
$HistoryLength            $ProcessorType
$HomeDirectory            $ProductInformation
$IgnoreEOF                $ProgramName
$ImportEncodings          $PSDirectDisplay
$ImportFormats            $RandomState
$InitialDirectory         $RasterFunction
$Input                    $RecursionLimit
$Inspector                $ReleaseNumber
$InstallationDate         $RootDirectory
$InstallationDirectory    $SessionID
$InterfaceEnvironment     $SetParentLink
$IterationLimit           $SoundDisplay
$Language                 $SoundDisplayFunction
$LaunchDirectory          $SuppressInputFormHeads
$LicenseExpirationDate    $SyntaxHandler
$LicenseID                $System
$LicenseProcesses         $SystemCharacterEncoding
$LicenseServer            $SystemID
$Line                     $TemporaryPrefix
$Linked                   $TextStyle
$LinkSupported            $TimeUnit
$MachineDomain            $TopDirectory
$MachineEpsilon           $TraceOff
$MachineID                $TraceOn
$MachineName              $TracePattern
$MachinePrecision         $TracePostAction
$MachineType              $TracePreAction
$MaxExtraPrecision        $Urgent
```

```
$MaxLicenseProcesses      $UserAddOnsDirectory
$MaxMachineNumber         $UserBaseDirectory
$MaxNumber                $UserName
$MaxPiecewiseCases        $Version
$MaxPrecision             $VersionNumber
$MaxRootDegree
```

Einige Anwendungen:
■ Quelques applications:

```
In[88]:= ??$Line

        $Line is a global variable that specifies the number of the current input line. Mehr…

        $Line = 88

In[89]:= $Line

Out[89]= 89

In[90]:= $Version

Out[90]= 5.2 for Microsoft Windows (June 20, 2005)

In[91]:= $VersionNumber

Out[91]= 5.2

In[92]:= $Letters

Out[92]= $Letters
```

5.12 Spezielle Formen
■ Formes spéciales

Mathematica-Zeichen sind Abkürzungen für Funktionen:
■ Les signes de *Mathematica* sont des abréviations de fonctions:

```
In[93]:= ??/

        x/y or Divide[x, y] is equivalent to x y^-1. Mehr…

        Attributes[Divide] = {Listable, NumericFunction, Protected}

In[94]:= Alias["/"]

Out[94]= Divide

In[95]:= 2/3 // N

Out[95]= 0.666667

In[96]:= Divide[2, 3] // N

Out[96]= 0.666667
```

In[97]:= **Divide[2, 3]**

Out[97]= $\frac{2}{3}$

In[98]:= **Apply[Divide,{2,3}]**

Out[98]= $\frac{2}{3}$

In[99]:= **??Apply**

```
Apply[f, expr] or f @@ expr replaces the head of expr by f. Apply[f, expr,
   levelspec] replaces heads in parts of expr specified by levelspec. Mehr...

Attributes[Apply] = {Protected}

Options[Apply] = {Heads → False}
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[100]:=

```
(* Old Form: Remove["Global`@*"] *)
```

In[101]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

5. Etwas Umgang mit *Mathematica*
▮ *Savoir un peu manier Mathematica*

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 5 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 5.
Testé avec *Mathematica* version 5.2+WIN

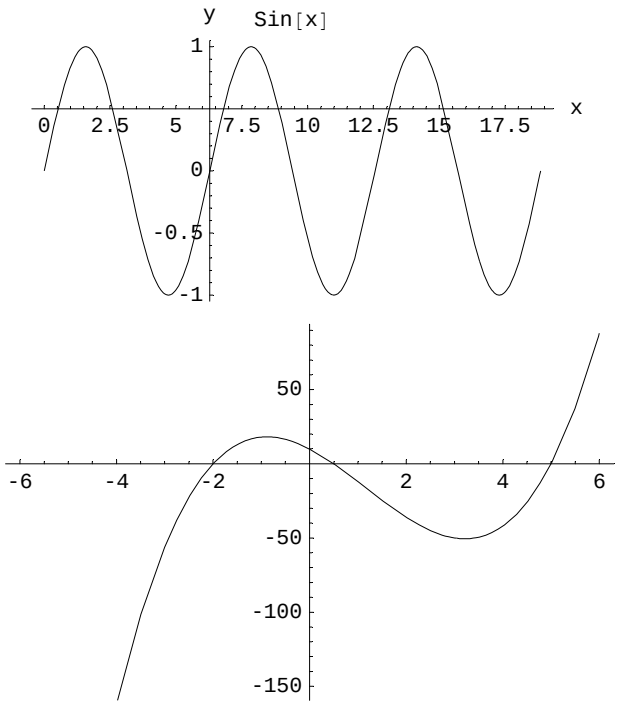
WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Ein Vorspann
▮ Génériques

Sinus ohne " ; ", damit Input-Lines erzeugt werden.
▮ Sinus sans " ; ", pour créer des Input-Lines.

```
In[1]:= Plot[Sin[x],{x,0,6 Pi},
            AspectRatio -> 1/2,
            PlotLabel -> "Sin[x]",
            AxesLabel -> {"x", "y"},
            AxesOrigin -> {2 Pi, 1/2} ];
        Plot[2x^3-7x^2-17x+10,{x,-6,6}];
        Solve[2x^3-7x^2-17x+10==0,x]
```



Out[3]= $\left\{\{x \to -2\}, \left\{x \to \frac{1}{2}\right\}, \{x \to 5\}\right\}$

Input-Lines auflisten
▮ Faire une Liste des Input-Lines

Mit "??": ▮ Avec "??":

In[4]:= **??In**

```
In[n] is a global object that is assigned to have a delayed value of the nth input line.
```
Mehr…

```
Attributes[In] = {Listable, Protected}

In[1] := (Plot[Sin[x], {x, 0, 6 π}, AspectRatio → 1/2,
   PlotLabel → Sin[x], AxesLabel → {x, y}, AxesOrigin → {2 π, 1/2}];)

In[2] := (Plot[2 x^3 - 7 x^2 - 17 x + 10, {x, -6, 6}];)

In[3] := Solve[2 x^3 - 7 x^2 - 17 x + 10 == 0, x]

In[4] := Information[In, LongForm → True]
```

Aufgabe 2 ▮ Problème 2

Auffinden von Kommandos
▮ Trouver des commandements

Finde alle Kommandos mit den Worten "Matrix" und "Power" im Namen:
▮ Trouve tous les commandements qui ont les paroles "Matrix" et Power" dans leur nom:

```
In[5]:= ?*Matrix* *Power*
```

System`

```
BezoutMatrix   MatrixForm  Plot3Matrix
DiagonalMatrix MatrixPower SylvesterMatrix
IdentityMatrix MatrixQ
MatrixExp      MatrixRank
```

System`

```
MatrixPower PowerExpand PowerModList
Power       PowerMod
```

Oder: ▮ Ou:

```
In[6]:= ??*Matrix* *Power*
```

System`

```
BezoutMatrix   MatrixForm  Plot3Matrix
DiagonalMatrix MatrixPower SylvesterMatrix
IdentityMatrix MatrixQ
MatrixExp      MatrixRank
```

System`

```
MatrixPower PowerExpand PowerModList
Power       PowerMod
```

Oder: ▮ Ou:

```
In[7]:= Print[Names["*Matrix*"]];
        Names["*Power*"]

        {BezoutMatrix, DiagonalMatrix, IdentityMatrix, MatrixExp,
         MatrixForm, MatrixPower, MatrixQ, MatrixRank, Plot3Matrix, SylvesterMatrix}

Out[8]= {MatrixPower, Power, PowerExpand, PowerMod, PowerModList}
```

Kurze Neugier:

■ Petite curiosité:

```
In[9]:= ??Names
```

```
Names["string"] gives a list of the names of symbols which match the string. Names["string",
   SpellingCorrection->True] includes names which match after spelling correction. Mehr...

Attributes[Names] = {Protected}

Options[Names] = {IgnoreCase → False, SpellingCorrection → False}
```

Aufgabe 3 ■ Problème 3

CPU-Zeit messen, 3 x 3-Matrix
■ Mesurer le temps CPU, matrice 3 x 3

Invertierung einer 3 x 3-Matrix. Postfix-Notation verwenden.

Erst Matrix generieren:

■ Invertir une matrice 3 x 3. User la notation postfixe. Générer d'abord la matrice:

```
In[10]:= m = Table[Random[ ], {3}, {3}]; MatrixForm[m]
```

```
Out[10]//MatrixForm=
        ⎛0.792512   0.232299  0.946558⎞
        ⎜0.631949   0.677425  0.572377⎟
        ⎝0.0745678  0.852139  0.863563⎠
```

Dann Inverse rechnen und Zeit messen (Postfix-Notation):

■ Ensuite calculer l'inverse et mesurer le temps (notation postfixe):

```
In[11]:= m // Inverse // MatrixForm // Timing
```

```
Out[11]= {0. Second, ⎛ 0.230388  1.43555   -1.20403 ⎞}
                     ⎜-1.19168   1.45405   0.342454 ⎟
                     ⎝ 1.15602   -1.55877  0.924037 ⎠
```

CPU-Zeit messen, 10 x 10-Matrix
■ Mesurer le temps CPU, matrice 10 x 10

Matrix generieren, Inverse rechnen und Zeit messen (Postfix-Notation):

■ Générer une matrice, calculer l'inverse et mesurer le temps. (Notation postfixe):

```
In[12]:= Table[Random[ ], {10}, {10}] // Inverse; // Timing
```

```
Out[12]= {0. Second, Null}
```

Aufgabe 4 ▮ Problème 4

(a) Fakultät berechnen: ▮ Calculer des factoriels

1 ! bis 5 ! ▮ 1! à 5!

```
In[13]:= Do[Print[i!], {i, 5}]
```

```
1
2
6
24
120
```

(a) Grössere Fakultät berechnen: ▮ Calculer des factoriels plus grands:

1 ! bis 250 ! ***Achtung!*** Der Rechner kann damit unmöglich fertig werden.
Die Zahlen sind zu gross! Daher musst Du den Vorgang unterbrechen!
Versuche in **Action**, **Interrupt**, **Abort Calculation** oder **Interrupt Calculation**.

▮ 1 ! à 250 ! ***Attention!*** La calculatrice ne peut pas en venir à bout. Les nombres sont trop grands! Tu dois donc interrompre le processus! Essaie
dans **(Action) Kernel**, **Interrupt**, **Abort Calculation** ou **Interrupt Calculation**.

```
In[14]:= Do[Print[i!], {i, 42}]
```

```
1
2
6
24
120
720
5040
40320
362880
3628800
39916800
479001600
```

```
6227020800

87178291200

1307674368000

20922789888000

355687428096000

6402373705728000

121645100408832000

2432902008176640000

51090942171709440000

1124000727777607680000

25852016738884976640000

620448401733239439360000

15511210043330985984000000

403291461126605635584000000

10888869450418352160768000000

304888344611713860501504000000

8841761993739701954543616000000

265252859812191058636308480000000

8222838654177922817725562880000000

263130836933693530167218012160000000

8683317618811886495518194401280000000

295232799039604140847618609643520000000

10333147966386144929666651337523200000000

371993326789901217467999448150835200000000

13763753091226345046315979581580902400000000

523022617466601111760007224100074291200000000

20397882081197443358640281739902897356800000000

815915283247897734345611269596115894272000000000

33452526613163807108170062053440751665152000000000

1405006117752879898543142606244511569936384000000000
```

Bitte nicht ausdrucken:

■ S.v.p. ne pas imprimer:

```
In[15]:= (* Do[Print[i!], {i, 250}] *)
```

In[16]:= **500!**

Out[16]= 1220136825991110068701238785423046926253574342803192842192413588385845373153881995
76054964475022032818630136164771482035841633787220781772004807852051593292854779
07571939330603772960859086270429174547882424912726344305670173270769461062802310
45264421887878946575477714986349436778103764427403382736539747138647787849543848
95955375379904232410612713269843277457155463099772027810145610811883737095310163
56324432987029563896628911658974769572087926928871281780070265174507768410719624
39039432253642260523494585012991857150124870696156814162535905669342381300885624
92468915641267756544818865065938479517753608940057452389403357984763639449053130
62323749066445048824665075946735862074637925184200459369692981022263971952597190
94521782333175693458150855233282076282002340262690789834245171200620771464097945
61161276291459512372299133401695523638509428855920187274337951730145863575708283
55780158735432768888680120399882384702151467605445407663535984174430480128938313
89688163948746965881750450692636533817505547812864000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000

Aufgabe 5 ▮ Problème 5

Einige Variablen:
▮ Quelques variables:

In[17]:= **??$Line**

```
$Line is a global variable that specifies the number of the current input line. Mehr...

$Line = 17
```

In[18]:= **??$Version**

```
$Version is a string that represents the version of Mathematica you are running. Mehr...

Attributes[$Version] = {Locked, Protected}

$Version = 5.2 for Microsoft Windows (June 20, 2005)
```

In[19]:= **??$VersionNumber**

```
$VersionNumber is a real number which gives the current Mathematica
   kernel version number, and increases in successive versions. Mehr...

Attributes[$VersionNumber] = {Locked, Protected}

$VersionNumber = 5.2
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[20]:= **(* Old Form: Remove["Global`@*"] *)**

In[21]:= **Remove["Global`*"]**

Kurs ▮ Cours

6. Manipulation von Listen
▮ Manipulation de listes

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 6 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 6.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

6.0. Einleitung
▮ Introduction

Listen - was sind Listen - was ist das?
▮ Des listes - qu'est - ce?

```
In[1]:= ??List

        {e1, e2, ... } is a list of elements. Mehr...

        Attributes[List] = {Locked, Protected}
```

Eingabe einer Liste:
▮ Entrer une liste:

```
In[2]:= list0 = {5, 4.78, a, 8x + y, Sin[x], D,
         Integer, x->E}

Out[2]= {5, 4.78, a, 8 x + y, Sin[x], D, Integer, x → e}
```

Oft gibt *Mathematica* auch Listen aus. Beispiel:
▮ *Mathematica* sort souvent des listes. Exemple:

```
In[3]:= Solve[{2x + 5y == 19, 3x + 7y == 27}]

Out[3]= {{x → 2, y → 3}}
```

Hier ist eine Liste mit einer Liste ausgegeben worden.

- Ici on a sorti une liste qui contient une liste.

Ebenso z.B. bei "Options":

- Egalement p.ex. avec "Options":

```
In[4]:= Options[NIntegrate]

Out[4]= {AccuracyGoal → ∞, Compiled → True, EvaluationMonitor → None, GaussPoints → Automatic,
          MaxPoints → Automatic, MaxRecursion → 6, Method → Automatic, MinRecursion → 0,
          PrecisionGoal → Automatic, SingularityDepth → 4, WorkingPrecision → MachinePrecision}
```

Hier ist eine Liste ausgegeben worden.

- Ici on a sorti une liste.

6.1. Erzeugung von Listen
▮ Générer des listes

6.1.1. Befehle
▮ Ordres

Zur Erzeugung von Listen stehen verschiedene Befehle zur Verfügung:

- Pour créer des listes on a plusieurs ordres à disposition:

```
In[5]:= ??Range

        Range[imax] generates the list {1, 2, ... , imax}. Range[imin, imax] generates
           the list {imin, ... , imax}. Range[imin, imax, di] uses step di. Mehr...

        Attributes[Range] = {Listable, Protected}

In[6]:= ??Table

        Table[expr, {imax}] generates a list of imax copies of expr. Table[
           expr, {i, imax}] generates a list of the values of expr when i runs from
           1 to imax. Table[expr, {i, imin, imax}] starts with i = imin. Table[expr,
           {i, imin, imax, di}] uses steps di. Table[expr, {i, imin, imax}, {j, jmin,
           jmax}, ... ] gives a nested list. The list associated with i is outermost. Mehr...

        Attributes[Table] = {HoldAll, Protected}

In[7]:= ??Array

        Array[f, n] generates a list of length n, with elements f[i]. Array[
           f, {n1, n2, ... }] generates an n1 by n2 by ... array of nested lists,
           with elements f[i1, i2, ... ]. Array[f, {n1, n2, ... }, {r1, r2, ... }]
           generates a list using the index origins ri (default 1). Array[f, dims,
           origin, h] uses head h, rather than List, for each level of the array. Mehr...

        Attributes[Array] = {Protected}
```

6.1.2. Erzeugung einer Liste mit "Range"
▮ Créer une liste avec "Range"

Beispiele: ▮ Exemples:

```
In[8]:= Range[5]
Out[8]= {1, 2, 3, 4, 5}

In[9]:= Range[7.896]
Out[9]= {1, 2, 3, 4, 5, 6, 7}

In[10]:= Range[6,12]
Out[10]= {6, 7, 8, 9, 10, 11, 12}

In[11]:= Range[3.4, 8.7]
Out[11]= {3.4, 4.4, 5.4, 6.4, 7.4, 8.4}

In[12]:= Range[3.4, 8.7, 1.2]
Out[12]= {3.4, 4.6, 5.8, 7., 8.2}
```

6.1.3. Erzeugung einer Liste mit "Table"
▮ Générer des listes avec "Table"

Beispiele: ▮ Exemples:

```
In[13]:= Table[hoi,{6}]
Out[13]= {hoi, hoi, hoi, hoi, hoi, hoi}

In[14]:= Table[i!,{i, 6}]
Out[14]= {1, 2, 6, 24, 120, 720}

In[15]:= Table[3i,{i, 6, 14}]
Out[15]= {18, 21, 24, 27, 30, 33, 36, 39, 42}

In[16]:= Table[i,{i, 4, 40, 5}]
Out[16]= {4, 9, 14, 19, 24, 29, 34, 39}

In[17]:= Table[{i, i^2},{i, 5}]
Out[17]= {{1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}

In[18]:= Table[a[i] + b[j], {i, 1, 3}, {j, 4, 6}]
Out[18]= {{a[1] + b[4], a[1] + b[5], a[1] + b[6]},
          {a[2] + b[4], a[2] + b[5], a[2] + b[6]}, {a[3] + b[4], a[3] + b[5], a[3] + b[6]}}
```

Das ist eine Liste von Listen, d.h. eine Matrix.
Zur besseren Darstellung verwenden wir "MatrixForm":

■ C'est une liste de listes, c'est-à-dire une matrice.
Pour plus de clarté nous utilisons "MatrixForm"

```
In[19]:= MatrixForm[%]
```

```
Out[19]//MatrixForm=
        ⎛a[1] + b[4]  a[1] + b[5]  a[1] + b[6]⎞
        ⎜a[2] + b[4]  a[2] + b[5]  a[2] + b[6]⎟
        ⎝a[3] + b[4]  a[3] + b[5]  a[3] + b[6]⎠
```

Ein Iterator kann auch von einem andern abhängen:
■ Un itérateur peut aussi dépendre d'un autre:

```
In[20]:= Table[a[i] + b[j], {i, 1, 3}, {j, i, 6}]
```

```
Out[20]= {{a[1] + b[1], a[1] + b[2], a[1] + b[3], a[1] + b[4], a[1] + b[5], a[1] + b[6]},
          {a[2] + b[2], a[2] + b[3], a[2] + b[4], a[2] + b[5], a[2] + b[6]},
          {a[3] + b[3], a[3] + b[4], a[3] + b[5], a[3] + b[6]}}
```

```
In[21]:= MatrixForm[%]
```

```
Out[21]//MatrixForm=
        ⎛{a[1] + b[1], a[1] + b[2], a[1] + b[3], a[1] + b[4], a[1] + b[5], a[1] + b[6]}⎞
        ⎜{a[2] + b[2], a[2] + b[3], a[2] + b[4], a[2] + b[5], a[2] + b[6]}              ⎟
        ⎝{a[3] + b[3], a[3] + b[4], a[3] + b[5], a[3] + b[6]}                          ⎠
```

6.1.4. "Array"

■ "Array"

Mit "Array" kann man eine symbolische Matrix erzeugen:
■ Avec "Array" on peut créer une matrice symbolique:

```
In[22]:= Clear[a]
```

```
In[23]:= Array[a,{2, 3}]
```

```
Out[23]= {{a[1, 1], a[1, 2], a[1, 3]}, {a[2, 1], a[2, 2], a[2, 3]}}
```

```
In[24]:= MatrixForm[%]
```

```
Out[24]//MatrixForm=
        ⎛a[1, 1]  a[1, 2]  a[1, 3]⎞
        ⎝a[2, 1]  a[2, 2]  a[2, 3]⎠
```

```
In[25]:= a = 17; Array[a,{2, 3}] // MatrixForm
```

```
Out[25]//MatrixForm=
        ⎛17[1, 1]  17[1, 2]  17[1, 3]⎞
        ⎝17[2, 1]  17[2, 2]  17[2, 3]⎠
```

6.2. Umordnen von Listen
▮ Ranger autrement des listes

6.2.1. Befehle
▮ Ordres

In[26]:= **??Sort**

```
Sort[list] sorts the elements of list into canonical
   order. Sort[list, p] sorts using the ordering function p.
```
Mehr...

```
Attributes[Sort] = {Protected}
```

In[27]:= **??Reverse**

```
Reverse[expr] reverses the order of the elements in expr.
```
Mehr...

```
Attributes[Reverse] = {Protected}
```

In[28]:= **??RotateLeft**

```
RotateLeft[expr, n] cycles the elements in expr n positions to the left.
   RotateLeft[expr] cycles one position to the left. RotateLeft[expr, {n1,
   n2, ... }] cycles elements at successive levels ni positions to the left.
```
Mehr...

```
Attributes[RotateLeft] = {Protected}
```

In[29]:= **??RotateRight**

```
RotateRight[expr, n] cycles the elements in expr n positions to the right.
   RotateRight[expr] cycles one position to the right. RotateRight[expr, {n1,
   n2, ... }] cycles elements at successive levels ni positions to the right.
```
Mehr...

```
Attributes[RotateRight] = {Protected}
```

In[30]:= **??Permutations**

```
Permutations[list] generates a list of all possible permutations of the elements in list.
```
Mehr...

```
Attributes[Permutations] = {Protected}
```

In[31]:= **??Drop**

```
Drop[list, n] gives list with its first n elements dropped. Drop[list,
   -n] gives list with its last n elements dropped. Drop[list, {n}] gives list
   with its nth element dropped. Drop[list, {m, n}] gives list with elements m
   through n dropped. Drop[list, {m, n, s}] gives list with elements m through
   n in steps of s dropped. Drop[list, seq1, seq2, ... ] gives a nested list in
   which elements specified by seqi have been dropped at level i in list.
```
Mehr...

```
Attributes[Drop] = {NHoldRest, Protected}
```

In[32]:= **??Take**

```
Take[list, n] gives the first n elements of list. Take[list, -n] gives the last n elements of
   list. Take[list, {m, n}] gives elements m through n of list. Take[list, {m, n,
   s}] gives elements m through n in steps of s. Take[list, seq1, seq2, ... ] gives a
   nested list in which elements specified by seqi are taken at level i in list. Mehr...

Attributes[Take] = {NHoldRest, Protected}
```

In[33]:= **??First**

```
First[expr] gives the first element in expr. Mehr...

Attributes[First] = {Protected}
```

In[34]:= **??Last**

```
Last[expr] gives the last element in expr. Mehr...

Attributes[Last] = {Protected}
```

In[35]:= **??Part**

```
expr[[i]] or Part[expr, i] gives the ith part of expr. expr[[-i]]
   counts from the end. expr[[0]] gives the head of expr. expr[[i, j, ... ]]
   or Part[expr, i, j, ... ] is equivalent to expr[[i]] [[j]] ... . expr[[
   {i1, i2, ... } ]] gives a list of the parts i1, i2, ...  of expr. Mehr...

Attributes[Part] = {NHoldRest, Protected, ReadProtected}
```

In[36]:= **??Rest**

```
Rest[expr] gives expr with the first element removed. Mehr...

Attributes[Rest] = {Protected}
```

In[37]:= **??Select**

```
Select[list, crit] picks out all elements ei of list for which crit[ei] is True. Select[
   list, crit, n] picks out the first n elements for which crit[ei] is True. Mehr...

Attributes[Select] = {Protected}
```

6.2.2. Anwendungen
▮ Applications

Hier einige Beispiele:
▮ Voici quelques exemples:

In[38]:= **listA = Table[Random[Integer, {0, 12}],{16}]**

Out[38]= {3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

In[39]:= **Sort[listA]**

Out[39]= {0, 1, 1, 1, 2, 2, 3, 3, 3, 5, 6, 6, 8, 10, 11, 12}

In[40]:= **Union[listA]**

Out[40]= {0, 1, 2, 3, 5, 6, 8, 10, 11, 12}

"Sort" sortiert die Liste, "Union" sortiert sie auch, eliminiert aber Duplikate von Elementen.
■ "Sort" trie la liste, "Union" la trie aussi, mais élimine les doubles des éléments.

```
In[41]:= Reverse[listA]

Out[41]= {1, 10, 3, 11, 2, 3, 2, 8, 6, 6, 1, 1, 12, 5, 0, 3}

In[42]:= RotateLeft[%,3]

Out[42]= {11, 2, 3, 2, 8, 6, 6, 1, 1, 12, 5, 0, 3, 1, 10, 3}

In[43]:= RotateRight[%,4]

Out[43]= {3, 1, 10, 3, 11, 2, 3, 2, 8, 6, 6, 1, 1, 12, 5, 0}

In[44]:= Permutations[{a, b, c}]

Out[44]= {{17, b, c}, {17, c, b}, {b, 17, c}, {b, c, 17}, {c, 17, b}, {c, b, 17}}

In[45]:= Permutations[{a, b, b}]

Out[45]= {{17, b, b}, {b, 17, b}, {b, b, 17}}
```

6.3. Erweiterung und Verkürzung von Listen
■ Allonger et raccourcir des listes

6.3.1. Befehle
■ Ordres

Probiere aus: ■ Essaie:

```
In[46]:= ?Append

        Append[expr, elem] gives expr with elem appended. Mehr...

In[47]:= ?AppendTo

        AppendTo[s, elem] appends elem to the value of s, and resets s to the result. Mehr...

In[48]:= ?Drop

        Drop[list, n] gives list with its first n elements dropped. Drop[list,
           -n] gives list with its last n elements dropped. Drop[list, {n}] gives list
           with its nth element dropped. Drop[list, {m, n}] gives list with elements m
           through n dropped. Drop[list, {m, n, s}] gives list with elements m through
           n in steps of s dropped. Drop[list, seq1, seq2, ... ] gives a nested list in
           which elements specified by seqi have been dropped at level i in list. Mehr...

In[49]:= ?Insert

        Insert[list, elem, n] inserts elem at position n in list. If n is negative,
           the position is counted from the end. Insert[expr, elem, {i, j, ... }]
           inserts elem at position {i, j, ... } in expr. Insert[expr, elem, {{i1,
           j1, ... }, {i2, j2, ... }, ... }] inserts elem at several positions. Mehr...
```

In[50]:= **?Prepend**

```
Prepend[expr, elem] gives expr with elem prepended.
```
Mehr...

In[51]:= **?PrependTo**

```
PrependTo[s, elem] prepends elem to the value of s, and resets s to the result.
```
Mehr...

In[52]:= **?Rest**

```
Rest[expr] gives expr with the first element removed.
```
Mehr...

In[53]:= **?Take**

```
Take[list, n] gives the first n elements of list. Take[list, -n] gives the last n elements of
   list. Take[list, {m, n}] gives elements m through n of list. Take[list, {m, n,
   s}] gives elements m through n in steps of s. Take[list, seq1, seq2, ... ] gives a
   nested list in which elements specified by seqi are taken at level i in list.
```
Mehr...

In[54]:= **?First**

```
First[expr] gives the first element in expr.
```
Mehr...

In[55]:= **?Last**

```
Last[expr] gives the last element in expr.
```
Mehr...

In[56]:= **?Part**

```
expr[[i]] or Part[expr, i] gives the ith part of expr. expr[[-i]]
   counts from the end. expr[[0]] gives the head of expr. expr[[i, j, ... ]]
   or Part[expr, i, j, ... ] is equivalent to expr[[i]] [[j]] ... . expr[[
   {i1, i2, ... } ]] gives a list of the parts i1, i2, ...  of expr.
```
Mehr...

In[57]:= **?Select**

```
Select[list, crit] picks out all elements ei of list for which crit[ei] is True. Select[
   list, crit, n] picks out the first n elements for which crit[ei] is True.
```
Mehr...

6.3.2. Beispiele
▮ Exemples

Probiere aus: ▮ Essaie:

In[58]:=
```
Print[listA];
Rest[listA]
```

```
{3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}
```

Out[59]=
```
{0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}
```

Was ist passiert?
▮ Que s'est-il passé?

In[60]:=
```
Drop[listA, 10]
```

Out[60]=
```
{3, 2, 11, 3, 10, 1}
```

```
In[61]:= Drop[listA, -5]
Out[61]= {3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3}

In[62]:= Drop[listA, {2}]
Out[62]= {3, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

In[63]:= Drop[listA, {1}]
Out[63]= {0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

In[64]:= Drop[listA, {3}]
Out[64]= {3, 0, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

In[65]:= Drop[listA, {1,8}]
Out[65]= {8, 2, 3, 2, 11, 3, 10, 1}

In[66]:= Drop[listA, {4,12}]
Out[66]= {3, 0, 5, 11, 3, 10, 1}

In[67]:= Take[listA,4]
Out[67]= {3, 0, 5, 12}

In[68]:= Take[listA,-6]
Out[68]= {3, 2, 11, 3, 10, 1}

In[69]:= Take[listA,{4,6}]
Out[69]= {12, 1, 1}

In[70]:= Append[listA, 99]
Out[70]= {3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1, 99}

In[71]:= Prepend[listA, 100]
Out[71]= {100, 3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

In[72]:= Insert[listA, 999, 4]
Out[72]= {3, 0, 5, 999, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

In[73]:= listA
Out[73]= {3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}
```

Bei all diesen Operationen ist "listA" unverändert geblieben. Nicht aber bei den folgenden Operationen:

■ Pendant toutes ces opérations "listA" n'a pas changé. Cela n'est pas le cas pour les opérations suivantes:

```
In[74]:= Print[listA];
         AppendTo[listA, 2000]

        {3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1}

Out[75]= {3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1, 2000}
```

In[76]:= **PrependTo[listA, 555]**

Out[76]= {555, 3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1, 2000}

6.4. Zählung der Elemente einer Liste
▮ Compter les éléments d'une liste

6.4.1. Befehle:
▮ Ordres:

Probiere aus: ▮ Essaie:

In[77]:= **?Length**

Length[expr] gives the number of elements in expr. Mehr...

In[78]:= **?Dimensions**

Dimensions[expr] gives a list of the dimensions of expr. Dimensions[
expr, n] gives a list of the dimensions of expr down to level n. Mehr...

6.4.2. Anwendungen
▮ Applications

In[79]:= **Length[listA]**

Out[79]= 18

In[80]:= **Length[{a, 4, 2^8, t^2 + 2t - 6, matrix}]**

Out[80]= 5

In[81]:= **Length[{{a, 4, 2^8, t^2 + 2t - 6, matrix},
listA}]**

Out[81]= 2

In[82]:= **Dimensions[{{a, 4, 2^8, t^2 + 2t - 6, matrix},
listA}]**

Out[82]= {2}

In[83]:= **Dimensions[{{a, 4, 2^8, t^2 + 2t - 6, matrix},
listA}, 0]**

Out[83]= {}

In[84]:= **Dimensions[{{a, 4, 2^8, t^2 + 2t - 6, matrix},
listA}, 1]**

Out[84]= {2}

```
In[85]:= Dimensions[{{a, 4, 2^8, t^2 + 2t - 6, matrix},
                     listA}, 2]

Out[85]= {2}

In[86]:= MatrixForm[{{a, 4, 2^8, t^2 + 2t - 6, matrix},
                     listA}]

Out[86]//MatrixForm=
         ( {17, 4, 256, -6 + 2 t + t^2, matrix}                                  )
         ( {555, 3, 0, 5, 12, 1, 1, 6, 6, 8, 2, 3, 2, 11, 3, 10, 1, 2000} )

In[87]:= Dimensions[{{a, 4, 2^8, t^2 + 2t - 6},
                     {1, 2, 3, 4}}, 2]

Out[87]= {2, 4}

In[88]:= MatrixForm[{{a, 4, 2^8, t^2 + 2t - 6},
                     {1, 2, 3, 4}}]

Out[88]//MatrixForm=
         ( 17  4  256  -6 + 2 t + t^2 )
         ( 1   2  3    4              )
```

6.5. Zusammenfügen von Listen, Beziehungen zwischen Listen
∎ Assemblage de listes, relations entre listes

6.5.1. Befehle
∎ Ordres

Probiere aus: ∎ Essaie

```
In[89]:= ?Complement

         Complement[eall, e1, e2, ... ] gives the elements in eall which are not in any of the ei.
          Mehr…

In[90]:= ?Union

         Union[list1, list2, ... ] gives a sorted list of all the distinct
            elements that appear in any of the listi. Union[list] gives a sorted version
            of a list, in which all duplicated elements have been dropped. Mehr…

In[91]:= ?Join

         Join[list1, list2, ... ] concatenates lists together. Join
            can be used on any set of expressions that have the same head. Mehr…

In[92]:= ?Intersection

         Intersection[list1, list2, ... ] gives a sorted list of the elements common to all the listi.
          Mehr…
```

6.5.2. Anwendungen
▮ Applications

```
In[93]:= Complement[{1, 2, 3, 4, 5},{1, 2, 3}]

Out[93]= {4, 5}

In[94]:= Intersection[{a, b, c, d, e, f},
                      {d, e, f, g, h, i, j, k}]

Out[94]= {d, e, f}

In[95]:= Union[{a, b, c, d, e, f},
                                {d, e, f, g, h, i, j, k}]

Out[95]= {17, b, c, d, e, f, g, h, i, j, k}

In[96]:= Union[{1, 1, 1, 1, 2, 1, 3, 3, 3, 2, 4, 2, 4, 4,
                      3, 3, 3}]

Out[96]= {1, 2, 3, 4}

In[97]:= Join[{a, b, c, d, e, f},{d, e, f, g, h, i, j, k}]

Out[97]= {17, b, c, d, e, f, d, e, f, g, h, i, j, k}

In[98]:= Join[{1, 2, 1, 1, 2, 2},{1, 2, 2, 1, 3, 2, 1,
                      3, 3}]

Out[98]= {1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 3, 2, 1, 3, 3}

In[99]:= Union[{1, 2, 1, 1, 2, 2},{1, 2, 2, 1, 3, 2, 1,
                      3, 3}]

Out[99]= {1, 2, 3}
```

6.6. Veränderung der Form einer Liste
▮ Changer la forme d'une liste

6.6.1. Befehle
▮ Ordres

Probiere aus: ▮ Essaie:

```
In[100]:=
       ?Flatten

        Flatten[list] flattens out nested lists. Flatten[list, n] flattens
           to level n. Flatten[list, n, h] flattens subexpressions with head h. Mehr...
```

In[101]:=

```
?Partition
```

```
Partition[list, n] partitions list into non-overlapping sublists of length n. Partition[list,
   n, d] generates sublists with offset d. Partition[list, {n1, n2, ... }] partitions
   a nested list into blocks of size n1 × n2 × … . Partition[list, {n1, n2, ... },
   {d1, d2, ... }] uses offset di at level i in list. Partition[list, n, d, {kL, kR}]
   specifies that the first element of list should appear at position kL in the first
   sublist, and the last element of list should appear at or after position kR in the
   last sublist. If additional elements are needed, Partition fills them in by treating
   list as cyclic. Partition[list, n, d, {kL, kR}, x] pads if necessary by repeating
   the element x. Partition[list, n, d, {kL, kR}, {x1, x2, ... }] pads if necessary by
   cyclically repeating the elements xi. Partition[list, n, d, {kL, kR}, {}] uses no
   padding, and so can yield sublists of different lengths. Partition[list, nlist, dlist,
   {klistL, klistR}, padlist] specifies alignments and padding in a nested list. Mehr...
```

In[102]:=

```
?Transpose
```

```
Transpose[list] transposes the first two levels in list.
   Transpose[list, {n1, n2, ... }] transposes list so that the levels 1,
   2, ... in list correspond to levels n1, n2, ... in the result. Mehr...
```

6.6.2. Anwendungen
▮ Applications

In[103]:=

```
vector1 = Range[10]
```

Out[103]=

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In[104]:=

```
matrix1 = Partition[vector1, 5]
```

Out[104]=

```
{{1, 2, 3, 4, 5}, {6, 7, 8, 9, 10}}
```

In[105]:=

```
MatrixForm[matrix1]
```

Out[105]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

In[106]:=

```
matrix2 = Partition[vector1, 4]
```

Out[106]=

```
{{1, 2, 3, 4}, {5, 6, 7, 8}}
```

In[107]:=

```
matrix3 = Partition[vector1, 3]
```

Out[107]=

```
{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
```

In[108]:=
```
matrix4 = Partition[vector1, 2.5]
```

```
Partition::ilsmp : Single or list of positive machine-size integers
   expected at position 2 of Partition[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 2.5]. Mehr…
```

Out[108]=
```
Partition[{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 2.5]
```

Was ist passiert?

■ Que s'est-il passé?

In[109]:=
```
Partition[{1, 35, 432, 2, 5467, 4, 3, 987, 87},
                                              3]
```

Out[109]=
```
{{1, 35, 432}, {2, 5467, 4}, {3, 987, 87}}
```

In[110]:=
```
Partition[{1, 2, 3, 4, 5, 6, 7}, 3, 1]
```

Out[110]=
```
{{1, 2, 3}, {2, 3, 4}, {3, 4, 5}, {4, 5, 6}, {5, 6, 7}}
```

In[111]:=
```
Partition[{{1, 2, 3}, {4, 5, 6}, {7, 8, 9},
            {a, b, c}}, 2, 1]
```

Out[111]=
```
{{{1, 2, 3}, {4, 5, 6}}, {{4, 5, 6}, {7, 8, 9}}, {{7, 8, 9}, {17, b, c}}}
```

In[112]:=
```
Print[matrix1];
Flatten[matrix1]
```

```
{{1, 2, 3, 4, 5}, {6, 7, 8, 9, 10}}
```

Out[113]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In[114]:=
```
m = {{1, 2, 3, a},{4, 5, 6, b}}
```

Out[114]=
```
{{1, 2, 3, 17}, {4, 5, 6, b}}
```

In[115]:=
```
Transpose[m]
```

Out[115]=
```
{{1, 4}, {2, 5}, {3, 6}, {17, b}}
```

In[116]:=
```
Print[MatrixForm[m]];
MatrixForm[Transpose[m]]
```

$$\begin{pmatrix} 1 & 2 & 3 & 17 \\ 4 & 5 & 6 & b \end{pmatrix}$$

Out[117]//MatrixForm=

$$\begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \\ 17 & b \end{pmatrix}$$

6.7. Elemente herauspicken
▮ Choisir quelques éléments

6.7.1. Befehle
▮ Ordres

Oben sind schon "First", "Last" und "Part erwähnt worden.

▮ Plus haut on a déjà nommé "First", "Last", et "Post".

6.7.2. Anwendungen
▮ Applications

In[118]:=
```
liste1 = {a, b, c, d, e, f, g}
```

Out[118]=
```
{17, b, c, d, e, f, g}
```

In[119]:=
```
First[liste1]
```

Out[119]=
```
17
```

In[120]:=
```
Last[liste1]
```

Out[120]=
```
g
```

In[121]:=
```
liste1[[4]]
```

Out[121]=
```
d
```

In[122]:=
```
Print[{liste1[[2]],liste1[[3]],liste1[[4]],
        liste1[[5]]}]
```

```
{b, c, d, e}
```

In[123]:=
```
liste1[[ Range[2, 5] ]]
```
Out[123]=
```
{b, c, d, e}
```

In[124]:=
```
array1 = Array[a, {2, 2}]
```
Out[124]=
```
{{17[1, 1], 17[1, 2]}, {17[2, 1], 17[2, 2]}}
```

In[125]:=
```
array1[[2]]
```
Out[125]=
```
{17[2, 1], 17[2, 2]}
```

In[126]:=
```
array1[[1, 2]]
```
Out[126]=
```
17[1, 2]
```

6.8. Auswahl von Daten
▮ Choisir des données

6.8.1. Befehle
▮ Ordres

Probiere aus: ▮ Essaie:

In[127]:=
```
?Select
```
```
Select[list, crit] picks out all elements ei of list for which crit[ei] is True. Select[
   list, crit, n] picks out the first n elements for which crit[ei] is True. Mehr...
```

In[128]:=
```
?*Q
```

System`

ArgumentCountQ	MatrixQ
ArrayQ	MemberQ
AtomQ	NameQ
DigitQ	NumberQ
EllipticNomeQ	NumericQ
EvenQ	OddQ
ExactNumberQ	OptionQ
FreeQ	OrderedQ
HypergeometricPFQ	PartitionsQ
InexactNumberQ	PolynomialQ
IntegerQ	PrimeQ
IntervalMemberQ	SameQ
InverseEllipticNomeQ	StringFreeQ
LegendreQ	StringMatchQ
LetterQ	StringQ
LinkConnectedQ	SyntaxQ
LinkReadyQ	TensorQ
ListQ	TrueQ
LowerCaseQ	UnsameQ
MachineNumberQ	UpperCaseQ
MatchLocalNameQ	ValueQ
MatchQ	VectorQ

In[129]:=
```
?DigitQ
```

DigitQ[string] yields True if all the characters in the string
   are digits in the range 0 through 9, and yields False otherwise. Mehr...

In[130]:=
```
?IntegerQ
```

IntegerQ[expr] gives True if expr is an integer, and False otherwise. Mehr...

In[131]:=
```
?LetterQ
```

LetterQ[string] yields True if all the characters
   in the string are letters, and yields False otherwise. Mehr...

In[132]:=
```
?LowerCaseQ
```

LowerCaseQ[string] yields True if all the characters in
   the string are lower-case letters, and yields False otherwise. Mehr...

In[133]:=
```
?MachineNumberQ
```

MachineNumberQ[expr] returns True if expr is a machine-
   precision real or complex number, and returns False otherwise. Mehr...

In[134]:=
?MatrixQ

MatrixQ[expr] gives True if expr is a list of lists or a two-dimensional SparseArray object that can represent a matrix, and gives False otherwise. MatrixQ[expr, test] gives True only if test yields True when applied to each of the matrix elements in expr. Mehr...

In[135]:=
?NameQ

NameQ["string"] yields True if there are any symbols whose names match the string pattern given, and yields False otherwise. Mehr...

In[136]:=
?NumberQ

NumberQ[expr] gives True if expr is a number, and False otherwise. Mehr...

In[137]:=
?OddQ

OddQ[expr] gives True if expr is an odd integer, and False otherwise. Mehr...

In[138]:=
?EvenQ

EvenQ[expr] gives True if expr is an even integer, and False otherwise. Mehr...

In[139]:=
?PrimeQ

PrimeQ[expr] yields True if expr is a prime number, and yields False otherwise. Mehr...

In[140]:=
?UpperCaseQ

UpperCaseQ[string] yields True if all the characters in the string are upper-case letters, and yields False otherwise. Mehr...

In[141]:=
?ValueQ

ValueQ[expr] gives True if a value has been defined for expr, and gives False otherwise. Mehr...

In[142]:=
?VectorQ

VectorQ[expr] gives True if expr is a list or a one-dimensional SparseArray object, none of whose elements are themselves lists, and gives False otherwise. VectorQ[expr, test] gives True only if test yields True when applied to each of the elements in expr. Mehr...

In[143]:=
?Positiv*

Positive[x] gives True if x is a positive number. Mehr...

In[144]:=
?Negat*

Negative[x] gives True if x is a negative number. Mehr...

6.8.2. Anwendungen
▮ Applications

Probiere aus: ▮ Essaie:

In[145]:=
```
MemberQ[liste1, 3]
```

Out[145]=
```
False
```

In[146]:=
```
MemberQ[liste1, a]
```

Out[146]=
```
True
```

In[147]:=
```
menge = {-4, 3, 5.678, emil, Emma, 4, 0, -2, -4,
            Pi, 2^(1/2)}
```

Out[147]=
$$\left\{-4, 3, 5.678, \text{emil}, \text{Emma}, 4, 0, -2, -4, \pi, \sqrt{2}\right\}$$

In[148]:=
```
Select[menge, Positive]
```

Out[148]=
$$\left\{3, 5.678, 4, \pi, \sqrt{2}\right\}$$

In[149]:=
```
Select[{a, b, c, d, x, y, z}, Positive]
```

Out[149]=
```
{17}
```

Bei "Select" steht also hinten das Selektionskriterium. Solche Kriterien kann man selbst definieren. Beispiel:
▮ Quant à "Select", le critère de sélection se trouve à la fin. On peut définir soi-même de tels critères. Exemple:

In[150]:=
```
?Precision
```

```
Precision[x] gives the effective number of digits of precision in the number x. Mehr…
```

In[151]:=
```
?===
```

```
lhs === rhs yields True if the expression
   lhs is identical to rhs, and yields False otherwise. Mehr…
```

In[152]:=
```
exakt[x_] := Precision[x] === Infinity
```

In[153]:=
```
Select[{-6, 0, 3, 3.1414333, 1.111111, Pi, E,
        N[Sqrt[2], 10], 5.66}, exakt]
```

Out[153]=
$$\{-6, 0, 3, \pi, e\}$$

6.9. Rechnen mit Listen
▮ Calculer avec des listes

6.9.1. Befehle
▮ Ordres

In[154]:=
```
?Sum
```

```
Sum[f, {i, imax}] evaluates the sum of the expressions f as
   evaluated for each i from 1 to imax. Sum[f, {i, imin, imax}] starts with
   i = imin. Sum[f, {i, imin, imax, di}] uses steps di. Sum[f, {i, imin,
   imax}, {j, jmin, jmax}, ... ] evaluates a sum over multiple indices. Mehr...
```

In[155]:=
```
?Plus
```

```
x + y + z represents a sum of terms. Mehr...
```

In[156]:=
```
?Apply
```

```
Apply[f, expr] or f @@ expr replaces the head of expr by f. Apply[f, expr,
   levelspec] replaces heads in parts of expr specified by levelspec. Mehr...
```

6.9.2. Wie man's macht:
▮ C'est ainsi que cela se fait:

In[157]:=
```
Sum[n^2, {n, 7}]
```

Out[157]=
```
140
```

In[158]:=
```
meineDaten = {3.4, 6.1, 5.3}
```

Out[158]=
```
{3.4, 6.1, 5.3}
```

In[159]:=
```
Sum[meineDaten[[n]], {n, Length[meineDaten]}]
```

Out[159]=
```
14.8
```

In[160]:=
```
Plus[1, 1, 1, 1, 1]
```

Out[160]=
```
5
```

In[161]:=
```
Plus[1, 2, 3, 4, 5]
```
Out[161]=
```
15
```

In[162]:=
```
Plus[a, b, c, d]
```
Out[162]=
```
17 + b + c + d
```

In[163]:=
```
Plus[{a, b, c, d}]
```
Out[163]=
```
{17, b, c, d}
```

In[164]:=
```
??Plus
```
```
x + y + z represents a sum of terms. Mehr...

Attributes[Plus] = {Flat, Listable, NumericFunction, OneIdentity, Orderless, Protected}

Default[Plus] := 0
```

"Plus" hat das Attribut "Listable".

■ "Plus" a l'attribut "Listable".

In[165]:=
```
??Listable
```
```
Listable is an attribute that can be assigned to a symbol f to indicate that the function f
   should automatically be threaded over lists that appear as its arguments. Mehr...

Attributes[Listable] = {Protected}
```

"Plus" auf eine Liste angewandt addiert nur ein Element, nämlich die Liste selbst. Wie nun "Plus" aber auf die Elemente anwenden? - So:

■ "Plus" appliqué à une liste additionne seulement un élément, c'est-à-dire la liste même. Comment appliquer "Plus" aux éléments? - Ainsi:

In[166]:=
```
Apply[Plus, {a, b, c, d}]
```
Out[166]=
```
17 + b + c + d
```

Was geht nun schneller, "Sum" oder "Plus" ?

■ Qu'est-ce qui est plus rapide, "Sum" ou "Plus"?

In[167]:=
```
Timing[Apply[Plus, Range[500]]]
```
Out[167]=
```
{0. Second, 125250}
```

In[168]:=
```
Timing[Sum[n, {n, 500}]]
```

Out[168]=
```
{0. Second, 125250}
```

6.10. Auf Listen anwendbare arithmetische Funktionen
▮ Fonctions arithmétiques applicables à des listes

Beispiele
▮ Exemples

Wir wenden "Log" auf eine Liste an:
▮ Nous appliquons "Log" à une liste:

In[169]:=
```
??Log
```

```
Log[z] gives the natural logarithm of z (
   logarithm to base e). Log[b, z] gives the logarithm to base b. Mehr...

Attributes[Log] = {Listable, NumericFunction, Protected}
```

In[170]:=
```
Log[{1, 2, 3, 4, 5, 6, 7}]
```

Out[170]=
```
{0, Log[2], Log[3], Log[4], Log[5], Log[6], Log[7]}
```

In[171]:=
```
N[%]
```

Out[171]=
```
{0., 0.693147, 1.09861, 1.38629, 1.60944, 1.79176, 1.94591}
```

Oder mit "Sinus":
▮ Ou avec "Sinus":

In[172]:=
```
Sin[{1, 2, 3, 4, 5, 6, 7}]
```

Out[172]=
```
{Sin[1], Sin[2], Sin[3], Sin[4], Sin[5], Sin[6], Sin[7]}
```

In[173]:=
```
Sin[{1, 2, 3, 4, 5, 6, 7}] // N
```

Out[173]=
```
{0.841471, 0.909297, 0.14112, -0.756802, -0.958924, -0.279415, 0.656987}
```

Oder "Plus":
▮ Ou "Plus":

In[174]:=
```
?Plus
```

x + y + z represents a sum of terms. Mehr...

In[175]:=
```
?*Form*
```

System`

AccountingForm	MathMLForm
BaseForm	MatrixForm
BinaryFormat	NumberForm
BlankForm	NumberFormat
BoxFormFormatTypes	OutputForm
CForm	OutputFormData
ColonForm	OutputToOutputForm
ColumnForm	OutputToStandardForm
CommonDefaultFormatTypes	PaddedForm
DefaultFormatType	ParentForm
DefaultFormatTypeForStyle	PasteBoxFormInlineCells
DefaultInlineFormatType	PointForm
DefaultInputFormatType	PolynomialForm
DefaultOutputFormatType	PrecedenceForm
DefaultTextFormatType	PrintForm
DefaultTextInlineFormatType	PromptForm
DisplayForm	RealBlockForm
DOSTextFormat	ReturnInputFormPacket
EdgeForm	RuleForm
EngineeringForm	ScientificForm
ExcludedForms	SequenceForm
FaceForm	SetBoxFormNamesPacket
FileFormat	ShowShortBoxForm
FontForm	SpaceForm
Format	StandardForm
FormatRules	StringForm
FormatType	StyleForm
FormatTypeAutoConvert	SyntaxForm
FormatValues	TableForm
FormBox	TeXForm
FormBoxOptions	TextForm
FortranForm	TraditionalForm
FullForm	TreeForm
HoldForm	ValueForm
HorizontalForm	VerticalForm
InputAutoFormat	$BoxForms
InputForm	$ExportFormats
InputToBoxFormPacket	$FormatType
InputToInputForm	$ImportFormats
InputToStandardForm	$OutputForms
LineForm	$PrintForms
LongForm	$SuppressInputFormHeads

```
In[176]:=
FullForm[a + b]

Out[176]//FullForm=
Plus[17, b]

In[177]:=
Plus[{1, 2, 3, 4}, {a, b, c, d}]

Out[177]=
{18, 2 + b, 3 + c, 4 + d}

In[178]:=
{1, 2, 3, 4} + {a, b, c, d}

Out[178]=
{18, 2 + b, 3 + c, 4 + d}
```

Attribute einiger Funktionen:

■ Attributs de quelques fonctions:

```
In[179]:=
Attributes[{Exp, Log, Sin, Cos, Plus, Times}]

Out[179]=
{{Listable, NumericFunction, Protected, ReadProtected},
 {Listable, NumericFunction, Protected},
 {Listable, NumericFunction, Protected}, {Listable, NumericFunction, Protected},
 {Flat, Listable, NumericFunction, OneIdentity, Orderless, Protected},
 {Flat, Listable, NumericFunction, OneIdentity, Orderless, Protected}}

In[180]:=
Attributes[Attributes]

Out[180]=
{HoldAll, Listable, Protected}
```

"Attributes" hat also auch das Attribut "Listable".

■ "Attributes" a donc aussi l'attribut "Listable".

6.11. Listable und Map

■ Listable et Map

6.11.1. Befehle

■ Ordres

Probiere: ■ Essaie:

In[181]:=
```
??Map
```

```
Map[f, expr] or f /@ expr applies f to each element on the first level in expr. Map[
   f, expr, levelspec] applies f to parts of expr specified by levelspec. Mehr…

Attributes[Map] = {Protected}

Options[Map] = {Heads → False}
```

In[182]:=
```
?Listable
```

```
Listable is an attribute that can be assigned to a symbol f to indicate that the function f
   should automatically be threaded over lists that appear as its arguments. Mehr…
```

In[183]:=
```
?Apply
```

```
Apply[f, expr] or f @@ expr replaces the head of expr by f. Apply[f, expr,
   levelspec] replaces heads in parts of expr specified by levelspec. Mehr…
```

6.11.2. Anwendungen
▮ Applications

Es gibt zwei Wege, eine Funktion auf eine Liste anzuwenden: mit "Listable" als Attribut und mit der Funktion "Map". Probiere:

▮ Il y a deux façons d'appliquer une fonction à une liste: avec l'attribut "List" et avec la fonction "Map". Essaie:

In[184]:=
```
f[{1, 2, 3, 4, 5, 6}]
```

Out[184]=
```
f[{1, 2, 3, 4, 5, 6}]
```

In[185]:=
```
?f
```

```
Global`f
```

f ist nicht "Listable".

▮ F n'est pas "Listable"

In[186]:=
```
Map[f, {1, 2, 3, 4, 5, 6}]
```

Out[186]=
```
{f[1], f[2], f[3], f[4], f[5], f[6]}
```

So geht's! Oder so:

▮ Voilà! Ou ainsi:

In[187]:=
```
Map[f, a x^2 + b x + c]
```

Out[187]=
```
f[c] + f[b x] + f[17 x^2]
```

Map wirkt auf die Teile eines Ausdrucks! Beachte:

■ Map a un effet sur les parties d'une expression! Remarque:

In[188]:=
```
Map[f, {a, b, c, d, e}]
```
Out[188]=
```
{f[17], f[b], f[c], f[d], f[e]}
```

In[189]:=
```
Apply[f, {a, b, c, d, e}]
```
Out[189]=
```
f[17, b, c, d, e]
```

Mit Apply wird f angewandt auf die Elemente der Menge als Ganzes, während mit Map diejenige Menge gebildet wird, die als neue Elemente die Funktionswerte der einzelnen alten Elemente enthält.

■ Avec Apply on applique f aux éléments de l'ensemble comme un tout, tandis qu'avec Map on forme l'ensemble qui contient en tant qu'éléments nouveaux les valeurs des fonctions des vieux éléments.

Statt Apply und Map kann man auch /@ und @@ benutzen. Beispiele:

■ On peut employer aussi /@ et @@ au lieu de Apply et Map. Exemple:

In[190]:=
```
f/@{a, b, c, d, e}
```
Out[190]=
```
{f[17], f[b], f[c], f[d], f[e]}
```

In[191]:=
```
f@@{a, b, c, d, e}
```
Out[191]=
```
f[17, b, c, d, e]
```

In[192]:=
```
Sin/@{0, Pi, 2 Pi, 2, 3, 4, 5, 6}
```
Out[192]=
```
{0, 0, 0, Sin[2], Sin[3], Sin[4], Sin[5], Sin[6]}
```

In[193]:=
```
f1[x_, y_,z_]:= Sin[x z^2] Cos[y]
```

In[194]:=
```
f1@@{1, 2, 3}
```
Out[194]=
```
Cos[2] Sin[9]
```

6.12. Formatierung von Listen
▮ Formater des listes

6.12.1. Befehle
▮ Ordres

Probiere: ▮ Essaie:

In[195]:=
```
?ColumnForm
```

```
ColumnForm[{e1, e2, ... }] prints as a column with e1 above e2, etc. ColumnForm[list,
   horiz] specifies the horizontal alignment of each element. ColumnForm[list,
   horiz, vert] also specifies the vertical alignment of the whole column. Mehr...
```

In[196]:=
```
?MatrixForm
```

```
MatrixForm[list] prints with the elements of list arranged in a regular array. Mehr...
```

In[197]:=
```
?TableForm
```

```
TableForm[list] prints with the elements of list arranged in an array of rectangular cells.
 Mehr...
```

6.12.2. Beispiele
▮ Exemples

In[198]:=
```
?Binomial
```

```
Binomial[n, m] gives the binomial coefficient. Mehr...
```

In[199]:=
```
pascal[n_]:= Table[Binomial[n, i],{i, 0, n}]
```

In[200]:=
```
pascal[5]
```

Out[200]=
```
{1, 5, 10, 10, 5, 1}
```

In[201]:=
```
Table[pascal[n],{n, 0, 5}]
```

Out[201]=
```
{{1}, {1, 1}, {1, 2, 1}, {1, 3, 3, 1}, {1, 4, 6, 4, 1}, {1, 5, 10, 10, 5, 1}}
```

In[202]:=

```
ColumnForm[Table[pascal[n],{n, 0, 5}]]
```

Out[202]=

```
{1}
{1, 1}
{1, 2, 1}
{1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
```

In[203]:=

```
ColumnForm[Table[pascal[n],{n, 0, 5}], Center]
```

Out[203]=

```
        {1}
      {1, 1}
    {1, 2, 1}
  {1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
```

In[204]:=

```
ColumnForm[Table[pascal[n],{n, 0, 5}], Right]
```

Out[204]=

```
                {1}
              {1, 1}
            {1, 2, 1}
          {1, 3, 3, 1}
        {1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
```

In[205]:=

```
ColumnForm[Table[pascal[n],{n, 0, 5}], Left]
```

Out[205]=

```
{1}
{1, 1}
{1, 2, 1}
{1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}
```

In[206]:=

```
TableForm[Table[pascal[n],{n, 0, 5}]]
```

Out[206]//TableForm=

```
1
1     1
1     2     1
1     3     3     1
1     4     6     4     1
1     5     10    10    5     1
```

In[207]:=

```
MatrixForm[Table[pascal[n],{n, 0, 5}]]
```

Out[207]//MatrixForm=

$$\begin{pmatrix} \{1\} \\ \{1, 1\} \\ \{1, 2, 1\} \\ \{1, 3, 3, 1\} \\ \{1, 4, 6, 4, 1\} \\ \{1, 5, 10, 10, 5, 1\} \end{pmatrix}$$

In[208]:=

```
MatrixForm[Table[n - i,{i,5},{n, 5}]]
```

Out[208]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ -1 & 0 & 1 & 2 & 3 \\ -2 & -1 & 0 & 1 & 2 \\ -3 & -2 & -1 & 0 & 1 \\ -4 & -3 & -2 & -1 & 0 \end{pmatrix}$$

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[209]:=

```
(* Old Form: Remove["Global`@*"] *)
```

In[210]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

6. Manipulation von Listen
▮ Manipulation de listes

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 6 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 6.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Erzeugung einer Liste der ersten 29 ungeraden natürlichen Zahlen
▮ Générer une liste des premiers 29 nombres naturels impairs

Mit Range ▮ Avec Range

```
In[1]:= Range[1,20,2]

Out[1]= {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
```

Mit Table ▮ Avec Table

```
In[2]:= oddList = Table[2i-1,{i,10}]

Out[2]= {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
```

Aufgabe 2 ▮ Problème 2

Herausnehmen und weglassen von 4 Elementen ▮ Enlever 4 éléments

Die ersten 4 Elemente nehmen:
▮ Prendre les 4 premiers éléments:

```
In[3]:= Take[oddList,4]

Out[3]= {1, 3, 5, 7}
```

Die letzten 4 Elemente weglassen:
▮ Supprimer les 4 derniers éléments:

```
In[4]:= Drop[oddList, -4]

Out[4]= {1, 3, 5, 7, 9, 11}
```

Aufgabe 3 ▮ Problème 3

Funktionen "Dimension" und "Length", Interpretation des Resultats ▮ Fonctions "Dimension" et "Length", interprétation des résultats

Eingabe der Matrizen
▮ Entrer les matrices

```
In[5]:= m = {{2, 4, 6}, {5, 7, 9}}; n = {{1, 2, 3}, {4,
                                                5}};
        Print[MatrixForm[m]]; Print["***"]; Print[MatrixForm[n]];

        (2  4  6)
        (5  7  9)

        ***

        ({1, 2, 3})
        ({4, 5}   )
```

Dimension ▮ Dimension

```
In[7]:= Dimensions[m]

Out[7]= {2, 3}
```

```
In[8]:= Dimensions[n]

Out[8]= {2}
```

Länge ▮ Longueur

```
In[9]:= Length[m]

Out[9]= 2

In[10]:=
        Length[n]

Out[10]= 2
```

Aufgabe 4 ▮ Problème 4

Einige Listenmanipulationen ▮ Quelques manipulations de listes

(a) Listen generieren ▮ Générer des listes

Zwei Listen kreieren:
▮ Créer deux listes

```
In[11]:= li1 = Table[Random[Integer,{0,10}],{i,10}]

Out[11]= {5, 4, 1, 2, 3, 10, 9, 2, 0, 3}

In[12]:= li2 = Table[Random[Integer,{0,10}],{i,10}]

Out[12]= {2, 3, 3, 1, 5, 4, 6, 7, 9, 9}
```

Als Mengen:
▮ Comme ensembles:

```
In[13]:= Union[li1]

Out[13]= {0, 1, 2, 3, 4, 5, 9, 10}

In[14]:= Union[li2]

Out[14]= {1, 2, 3, 4, 5, 6, 7, 9}
```

(b) Sortieren
▮ Trier

Sortieren von li1:
▮ Trier li1:

```
In[15]:=
        Sort[li1]

Out[15]= {0, 1, 2, 2, 3, 3, 4, 5, 9, 10}
```

(c) Anfügen
▮ Ajouter

An li1: ▮ A li1:

```
In[16]:= Append[li1, 5]

Out[16]= {5, 4, 1, 2, 3, 10, 9, 2, 0, 3, 5}
```

(d) Schnittmenge
▮ Intersection

von li1 und li2:
▮ de li1 et li2:

```
In[17]:= Intersection[li1, li2]

Out[17]= {1, 2, 3, 4, 5, 9}
```

Vereinigung
▮ Réunir

von li1 und li2:
▮ li1 et li2:

```
In[18]:= Union[li1, li2]

Out[18]= {0, 1, 2, 3, 4, 5, 6, 7, 9, 10}
```

(e) Zusammenhängen
▮ Relier

von li1 und li2:
▮ li1 et li2:

```
In[19]:= Join[li1, li2]

Out[19]= {5, 4, 1, 2, 3, 10, 9, 2, 0, 3, 2, 3, 3, 1, 5, 4, 6, 7, 9, 9}
```

(f) Verschiedene Elemente
▮ Différents éléments

in li1 und li2:
▮ dans li1 et li2:

```
In[20]:= Complement[Union[li1, li2],Intersection[li1, li2]]

Out[20]= {0, 6, 7, 10}
```

(f) Differenz
▮ Différence

li1 ohne li2:
▮ li1 sans li2:

```
In[21]:= Complement[li1, li2]

Out[21]= {0, 10}
```

(f) Komplement
▮ Complément

von li1 und li2 bezüglich Range[10]:
▮ de li1 et li2 par rapport à Range[10]:

```
In[22]:= Complement[Range[10],Union[li1, li2]]

Out[22]= {8}
```

Aufgabe 5 ▮ Problème 5

Test, ob alle Elemente einer Liste dieselben sind
▮ Test, si tous les éléments d'une liste sont les mêmes

Liste von Listen von Zahlen:
▮ Liste de listes de nombres:

```
In[23]:= myList = {{-1,-1,2},{-1,-1,2},{-1,-1,2},{-1,-1,2}}

Out[23]= {{-1, -1, 2}, {-1, -1, 2}, {-1, -1, 2}, {-1, -1, 2}}
```

```
In[24]:= Length[Union[myList]]

Out[24]= 1
```

Liste von Polynomen:
■ Liste de polynômes:

```
In[25]:= myList = {x^2-1,x^2-1,x^2-1,x^2-1,(x-1)(x+1)}

Out[25]= {-1 + x^2, -1 + x^2, -1 + x^2, -1 + x^2, (-1 + x) (1 + x)}

In[26]:= Length[Union[myList]]

Out[26]= 2
```

Aufgabe 6 ■ Problème 6

Listenmanipulationen
■ Manipulations de listes

(a) Liste generieren
■ Générer une liste

Liste generieren mit 16 Zahlen zwischen 0 und 100:
■ Générer une liste de 16 nombres entre 0 et 100:

```
In[27]:= li3 = Table[Random[Integer,{0,100}],{i,16}]

Out[27]= {46, 75, 21, 51, 6, 11, 30, 5, 95, 28, 46, 88, 96, 26, 98, 64}
```

(b) Partitionen
■ Partitions

mit 4 Teilmengen
■ avec 4 sous-ensembles

```
In[28]:= li4 = Partition[li3,4]

Out[28]= {{46, 75, 21, 51}, {6, 11, 30, 5}, {95, 28, 46, 88}, {96, 26, 98, 64}}
```

Matrizenform:
■ Forme de matrice:

In[29]:= **MatrixForm[li4]**

Out[29]//MatrixForm=

$$\begin{pmatrix} 46 & 75 & 21 & 51 \\ 6 & 11 & 30 & 5 \\ 95 & 28 & 46 & 88 \\ 96 & 26 & 98 & 64 \end{pmatrix}$$

(c) 3. Zeile der Matrix
▮ 3. ligne de la matrice

In[30]:= **li4[[3]]**

Out[30]= {95, 28, 46, 88}

(d) 3. Spalte der Matrix
▮ 3. colonne de la matrice

In[31]:= **Transpose[li4][[3]]**

Out[31]= {21, 30, 46, 98}

In[32]:= **Transpose[li4][[3]] // MatrixForm**

Out[32]//MatrixForm=

$$\begin{pmatrix} 21 \\ 30 \\ 46 \\ 98 \end{pmatrix}$$

(e) Mittelwert und Median
▮ Valeur moyenne et médiane

Mittelwet: ▮ Valeur moyenne:

In[33]:= **mean = Sum[li3[[i]],{i,1,Length[li3]}]/Length[li3]**

```
General::spell1 :
 Possible spelling error: new symbol name "mean" is similar to existing symbol "Mean". Mehr…
```

Out[33]= $\frac{393}{8}$

Oder: ▮ Ou:

In[34]:= **Apply[Plus,li3]/Length[li3]**

Out[34]= $\frac{393}{8}$

Median: ▮ Médiane:

Erst Liste sortieren:
▮ D'abord trier la liste:

```
In[35]:= sLi3 = Sort[li3]

Out[35]= {5, 6, 11, 21, 26, 28, 30, 46, 46, 51, 64, 75, 88, 95, 96, 98}
```

Mittleres Element resp. Durchschnitt der beiden mittleren Elemente rechnen, erst Funktion definieren:
■ Calculer l'élément du milieu, resp. la moyenne des deux éléments, d'abord définir la fonction:

```
In[36]:= median[x_List] := (x[[Ceiling[ Length[x]   /2]]] +
                            x[[  Floor[(Length[x]+2)/2]]]
                    ) / 2

         General::spell1 :
          Possible spelling error: new symbol name "median" is similar to existing symbol "Median". Mehr…
```

Versuche, dieses Programm zu begreifen!
■ Essaie de comprendre ce programme!

Rechnen: ■ Calculer la médiane:

```
In[37]:= median[sLi3]

Out[37]= 46
```

Liste mit ungerader Anzahl Elemente definieren, sortieren:
■ Définir une liste avec nombre impair d'éléments, trier:

```
In[38]:= sLi4 = Sort[Append[li3,99]]

Out[38]= {5, 6, 11, 21, 26, 28, 30, 46, 46, 51, 64, 75, 88, 95, 96, 98, 99}
```

Median rechnen:
■ Calculer la médiane:

```
In[39]:= median[sLi4]

Out[39]= 46
```

Aufgabe 7 ■ Problème 7

Mit "Select" Elemente herauspicken:
■ Choisir avec "Select" des éléments:

Liste: ■ Liste:

```
In[40]:= li3

Out[40]= {46, 75, 21, 51, 6, 11, 30, 5, 95, 28, 46, 88, 96, 26, 98, 64}
```

Programm: ■ Programme:

```
In[41]:= greater50[x_]:= x > 50
```

Anwenden: ■ Appliquer:

In[42]:= **Select[li3, greater50]**

Out[42]= {75, 51, 95, 88, 96, 98, 64}

Aufgabe 8 ▮ Problème 8

Spiel mit "Apply" und "Map": ▮ Jeu avec "Apply" et "Map":

Liste eingeben:
▮ Entrer une liste:

In[43]:= **Clear[myList]; myList = {{a,b},{c,d}}**

Out[43]= {{a, b}, {c, d}}

(a) ausprobieren:
▮ essayer:

In[44]:= **Apply[f, myList]**

Out[44]= f[{a, b}, {c, d}]

(b) ausprobieren:
▮ essayer:

In[45]:= **Map[f, myList]**

Out[45]= {f[{a, b}], f[{c, d}]}

(c) On-line help:
▮ On-line help:

In[46]:= **??MapAt**

MapAt[f, expr, n] applies f to the element at position n in expr. If n is negative, the position is counted from the end. MapAt[f, expr, {i, j, ... }] applies f to the part of expr at position {i, j, ... }. MapAt[f, expr, {{i1, j1, ... }, {i2, j2, ... }, ... }] applies f to parts of expr at several positions. Mehr...

Attributes[MapAt] = {Protected}

In[47]:= **??MapAll**

MapAll[f, expr] or f //@ expr applies f to every subexpression in expr. Mehr...

Attributes[MapAll] = {Protected}

Options[MapAll] = {Heads → False}

(d) ausprobieren:
▮ essayer:

```
In[48]:= MapAt[f, myList, {2}]

Out[48]= {{a, b}, f[{c, d}]}
```

(e) ausprobieren:
■ essayer:

```
In[49]:= MapAll[f, myList]

Out[49]= f[{f[{f[a], f[b]}], f[{f[c], f[d]}]}]
```

Aufgabe 9 ■ Problème 9

Schreibe ein *Mathematica*-Programm, das aus zwei Zahlenreihen positionsweise immer die kleinere heraussucht:
■ Ecris un programme de *Mathematica* qui, de deux ensembles de nombres, choisit pour chaque position le plus petit nombre:

Zahlenreihen: ■ Ensembles de nombres:

```
In[50]:= r1 = {15, 17, 18, 32, 29}

Out[50]= {15, 17, 18, 32, 29}
```

```
In[51]:=
         r2 = {14, 18, 22, 29, 26}

Out[51]= {14, 18, 22, 29, 26}
```

Programm: ■ Programme:

```
In[52]:= Map[Min, Transpose[{r1, r2}]]

Out[52]= {14, 17, 18, 29, 26}
```

Dazu Details:
■ Quelques détails:

```
In[53]:= Transpose[{r1, r2}] // MatrixForm

Out[53]//MatrixForm=
         ⎛15  14⎞
         ⎜17  18⎟
         ⎜18  22⎟
         ⎜32  29⎟
         ⎝29  26⎠
```

Aufgabe 10 ▮ Problème 10

Eine Aufgabe mit 3D-Graphik
▮ Un problème avec des graphiques 3D

Generiere eine Liste von 80 Punkten im 3-dimensionalen Raum:

▮ Générer une Liste de 80 points dans l'espace à trois dimensions

```
In[54]:= li5 = Table[Table[Random[Integer,{0,100}],{i,3}],
                {j,80}]
```

```
Out[54]= {{76, 76, 11}, {99, 80, 45}, {20, 10, 86}, {7, 5, 56}, {36, 70, 52}, {11, 81, 74},
          {55, 46, 40}, {50, 93, 26}, {55, 42, 20}, {72, 10, 98}, {58, 77, 27},
          {34, 93, 75}, {54, 76, 68}, {75, 24, 33}, {50, 59, 94}, {18, 69, 89},
          {53, 83, 62}, {90, 12, 64}, {94, 78, 21}, {44, 76, 66}, {88, 87, 14},
          {37, 22, 93}, {23, 40, 20}, {53, 73, 10}, {17, 15, 99}, {69, 100, 39},
          {0, 99, 13}, {39, 8, 69}, {43, 89, 33}, {94, 34, 75}, {42, 8, 91}, {29, 13, 78},
          {66, 79, 41}, {87, 55, 26}, {61, 52, 82}, {95, 33, 61}, {58, 54, 73}, {19, 19, 4},
          {56, 93, 28}, {19, 22, 81}, {2, 39, 31}, {91, 61, 33}, {73, 78, 13}, {81, 22, 3},
          {3, 82, 94}, {18, 12, 1}, {42, 64, 18}, {97, 88, 13}, {92, 34, 82}, {65, 41, 95},
          {3, 31, 64}, {87, 96, 80}, {20, 93, 23}, {30, 10, 79}, {26, 60, 41}, {83, 48, 30},
          {70, 69, 39}, {23, 36, 93}, {73, 13, 26}, {89, 81, 2}, {38, 8, 55}, {26, 10, 12},
          {46, 5, 65}, {10, 6, 9}, {84, 93, 38}, {14, 7, 33}, {5, 4, 90}, {81, 95, 71},
          {62, 76, 94}, {19, 58, 78}, {69, 99, 69}, {23, 93, 72}, {98, 5, 54}, {66, 2, 90},
          {29, 76, 8}, {18, 72, 92}, {44, 69, 27}, {87, 99, 6}, {65, 97, 14}, {75, 87, 100}}
```

```
In[55]:= MatrixForm[%]
```

```
Out[55]//MatrixForm=
```

76	76	11
99	80	45
20	10	86
7	5	56
36	70	52
11	81	74
55	46	40
50	93	26
55	42	20
72	10	98
58	77	27
34	93	75
54	76	68
75	24	33
50	59	94
18	69	89
53	83	62
90	12	64
94	78	21
44	76	66
88	87	14
37	22	93
23	40	20
53	73	10
17	15	99
69	100	39

$$\begin{pmatrix}
0 & 99 & 13 \\
39 & 8 & 69 \\
43 & 89 & 33 \\
94 & 34 & 75 \\
42 & 8 & 91 \\
29 & 13 & 78 \\
66 & 79 & 41 \\
87 & 55 & 26 \\
61 & 52 & 82 \\
95 & 33 & 61 \\
58 & 54 & 73 \\
19 & 19 & 4 \\
56 & 93 & 28 \\
19 & 22 & 81 \\
2 & 39 & 31 \\
91 & 61 & 33 \\
73 & 78 & 13 \\
81 & 22 & 3 \\
3 & 82 & 94 \\
18 & 12 & 1 \\
42 & 64 & 18 \\
97 & 88 & 13 \\
92 & 34 & 82 \\
65 & 41 & 95 \\
3 & 31 & 64 \\
87 & 96 & 80 \\
20 & 93 & 23 \\
30 & 10 & 79 \\
26 & 60 & 41 \\
83 & 48 & 30 \\
70 & 69 & 39 \\
23 & 36 & 93 \\
73 & 13 & 26 \\
89 & 81 & 2 \\
38 & 8 & 55 \\
26 & 10 & 12 \\
46 & 5 & 65 \\
10 & 6 & 9 \\
84 & 93 & 38 \\
14 & 7 & 33 \\
5 & 4 & 90 \\
81 & 95 & 71 \\
62 & 76 & 94 \\
19 & 58 & 78 \\
69 & 99 & 69 \\
23 & 93 & 72 \\
98 & 5 & 54 \\
66 & 2 & 90 \\
29 & 76 & 8 \\
18 & 72 & 92 \\
44 & 69 & 27 \\
87 & 99 & 6 \\
65 & 97 & 14 \\
75 & 87 & 100
\end{pmatrix}$$

Graphik-Objekte herstellen:

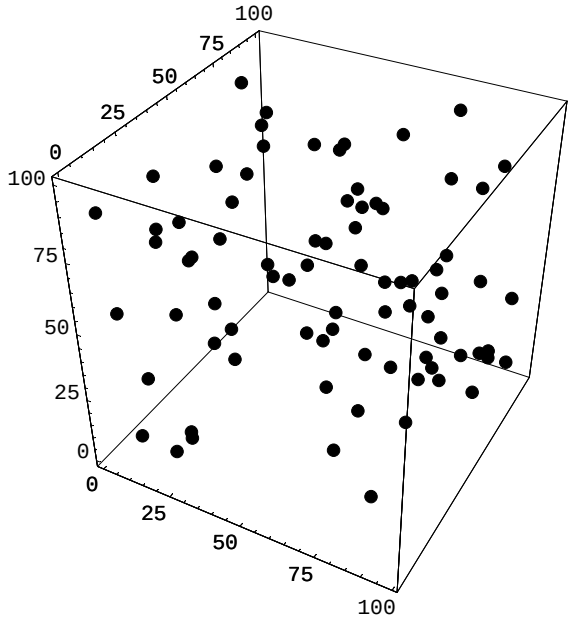
- Faire des objets graphiques

In[56]:= **punkte = Map[Point, li5]**

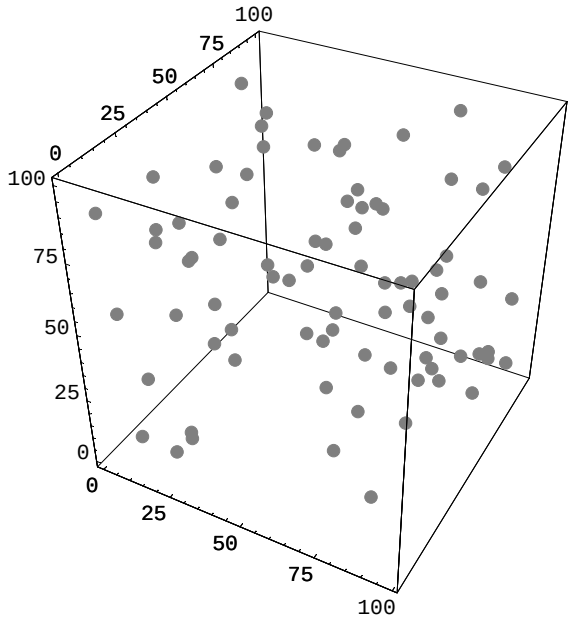
Out[56]=
```
{Point[{76, 76, 11}], Point[{99, 80, 45}], Point[{20, 10, 86}], Point[{7, 5, 56}],
 Point[{36, 70, 52}], Point[{11, 81, 74}], Point[{55, 46, 40}], Point[{50, 93, 26}],
 Point[{55, 42, 20}], Point[{72, 10, 98}], Point[{58, 77, 27}], Point[{34, 93, 75}],
 Point[{54, 76, 68}], Point[{75, 24, 33}], Point[{50, 59, 94}], Point[{18, 69, 89}],
 Point[{53, 83, 62}], Point[{90, 12, 64}], Point[{94, 78, 21}], Point[{44, 76, 66}],
 Point[{88, 87, 14}], Point[{37, 22, 93}], Point[{23, 40, 20}], Point[{53, 73, 10}],
 Point[{17, 15, 99}], Point[{69, 100, 39}], Point[{0, 99, 13}], Point[{39, 8, 69}],
 Point[{43, 89, 33}], Point[{94, 34, 75}], Point[{42, 8, 91}], Point[{29, 13, 78}],
 Point[{66, 79, 41}], Point[{87, 55, 26}], Point[{61, 52, 82}], Point[{95, 33, 61}],
 Point[{58, 54, 73}], Point[{19, 19, 4}], Point[{56, 93, 28}], Point[{19, 22, 81}],
 Point[{2, 39, 31}], Point[{91, 61, 33}], Point[{73, 78, 13}], Point[{81, 22, 3}],
 Point[{3, 82, 94}], Point[{18, 12, 1}], Point[{42, 64, 18}], Point[{97, 88, 13}],
 Point[{92, 34, 82}], Point[{65, 41, 95}], Point[{3, 31, 64}], Point[{87, 96, 80}],
 Point[{20, 93, 23}], Point[{30, 10, 79}], Point[{26, 60, 41}], Point[{83, 48, 30}],
 Point[{70, 69, 39}], Point[{23, 36, 93}], Point[{73, 13, 26}], Point[{89, 81, 2}],
 Point[{38, 8, 55}], Point[{26, 10, 12}], Point[{46, 5, 65}], Point[{10, 6, 9}],
 Point[{84, 93, 38}], Point[{14, 7, 33}], Point[{5, 4, 90}], Point[{81, 95, 71}],
 Point[{62, 76, 94}], Point[{19, 58, 78}], Point[{69, 99, 69}], Point[{23, 93, 72}],
 Point[{98, 5, 54}], Point[{66, 2, 90}], Point[{29, 76, 8}], Point[{18, 72, 92}],
 Point[{44, 69, 27}], Point[{87, 99, 6}], Point[{65, 97, 14}], Point[{75, 87, 100}]}
```

Ploten: ■ "Ploter":

In[57]:=
```
Show[Graphics3D[{PointSize[0.025], punkte}],
                          Axes -> Automatic];
```



```
In[58]:= Show[Graphics3D[{RGBColor[0.5,0.5,0.5],
                  PointSize[0.025], punkte}],
                            Axes -> Automatic];
```

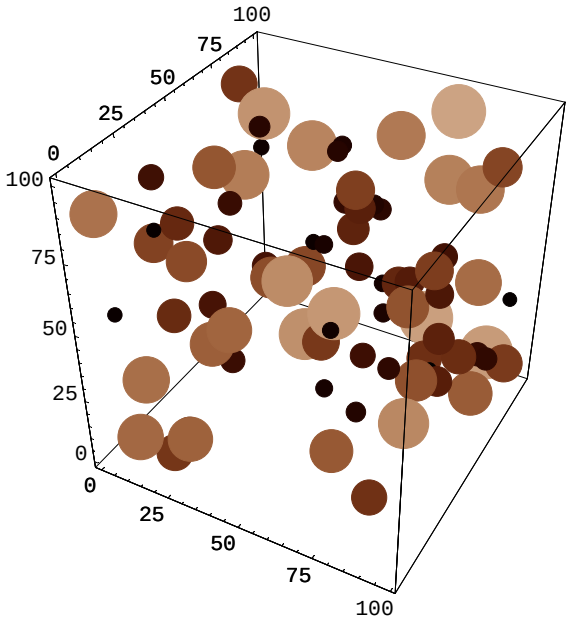


Aufgabe 11 ▮ Problème 11

Probiere in Aufgabe 10 die Graustufe vom Punkt abhängig zu machen, so dass jeder Punkt eine andere Graustufe bekommt!
▮ Prends le problème 10 et essaie de rendre l'intensité du gris dépendante du point, ainsi que chaque point aie une autre intensité de gris!

Eine mögliche Lösung:
▮ Une solution possible:

In[59]:=
```
Show[Graphics3D[
                    Table[{
                        RGBColor[i/100,(i/100)^2,(i/100)^3],
                        PointSize[0.025+i/1000],
                        punkte[[i]]
                          },{i,Length[punkte]}]
                ],
                        Axes -> Automatic
        ];
```



"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[60]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[61]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

7. Probleme zum Thema "Zuordnungen und Regeln"
▮ Problèmes au sujet des "applications et règles"

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 7 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 7.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Es geht hier darum zu zeigen, wie ein Name einem Wert oder einer Operation zugewiesen wird. Oder wie Symbole in Ausdrücken durch Werte ersetzt werden.
▮ Il s'agit de montrer ici comme un nom est assigné à une valeur ou à une opération. Ou comme on remplace des symboles dans des expressions par des valeurs.

7.1. Zuordnungen
▮ Applications

7.1.1. Allgemeines
▮ Généralités

Namen bestehen aus Buchstaben, Ziffern oder "$".
Eingebaute Mathematica-Namen beginnen mit Grossbuchstaben oder "$".
Namen mit "$" am Anfang sind globale Variablen. Probiere aus:
▮ Les noms sont formés de lettres, chiffres ou "$". Les noms incorporés de *Mathematica* commencent par une majuscule ou par "$". Les noms qui commencent par "$" sont des variables globales. Essaie:

```
In[1]:= a = 5

Out[1]= 5
```

In[2]:= **a + 4**

Out[2]= 9

Statt Werte kann man auch Ausdrücke zuordnen und damit rechnen:
■ Au lieu des valeurs on peut aussi assigner des expressions et calculer avec celles-ci:

In[3]:= **a = 3x + y**

Out[3]= 3 x + y

In[4]:= **1/a**

Out[4]= $\frac{1}{3\,x + y}$

"." heisst "missing value". Weist man "." zu, so ist die Variable "leer":
■ "." veut dire "missing value". Si on assigne ".", la variable est "vide":

In[5]:= **a = .**

In[6]:= **a**

Out[6]= a

7.1.2. Unmittelbare und verzögerte Zuordnungen
■ Applications immédiates et retardées

"=" oder "Set" bewirkt eine sofortige Zuordnung. Probiere aus:
■ "=" ou "Set" produisent une application immédiate. Essaie:

In[7]:= **?Set**

```
lhs = rhs evaluates rhs and assigns the result to be the value of lhs. From then on, lhs is
   replaced by rhs whenever it appears. {l1, l2, ... } = {r1, r2, ... } evaluates
   the ri, and assigns the results to be the values of the corresponding li. Mehr...
```

In[8]:= **?=**

```
lhs = rhs evaluates rhs and assigns the result to be the value of lhs. From then on, lhs is
   replaced by rhs whenever it appears. {l1, l2, ... } = {r1, r2, ... } evaluates
   the ri, and assigns the results to be the values of the corresponding li. Mehr...
```

In[9]:= **Set[a, 33444]**

Out[9]= 33444

In[10]:= **a**

Out[10]= 33444

In[11]:= **random1 = Random[]**

Out[11]= 0.529863

In[12]:= **2**

Out[12]= 2

In[13]:= **3 + random1**

Out[13]= 3.52986

":=" oder "SetDelayed" bewirkt die Zuordnung erst dann, wenn die nachstehende Funktion aufgerufen wird. Probiere:
■ ":=" ou "SetDelayed" produisent l'application seulement, quand la fonction suivante est appelé. Essaie:

In[14]:= **?:=**

```
lhs := rhs assigns rhs to be the delayed value of lhs. rhs is maintained in an unevaluated
  form. When lhs appears, it is replaced by rhs, evaluated afresh each time. Mehr…
```

In[15]:= **?SetDelayed**

```
lhs := rhs assigns rhs to be the delayed value of lhs. rhs is maintained in an unevaluated
  form. When lhs appears, it is replaced by rhs, evaluated afresh each time. Mehr…
```

In[16]:= **random2 := Random[]**

Kein Output, da random2 nicht gerufen, sondern erst gesetzt worden ist.
Vergleiche und erkläre:
■ Pas de "Output" car random2 n'a pas été appelé, mais seulement été placé. Compare et explique:

In[17]:= **Table[random1,{5}]**

Out[17]= {0.529863, 0.529863, 0.529863, 0.529863, 0.529863}

In[18]:= **Table[random2,{5}]**

Out[18]= {0.320157, 0.260433, 0.706004, 0.425192, 0.767409}

Was ist der Unterschied und wieso tritt er auf? (Du kannst auch mehrmals probieren.) Anwendung auf die Definition von Funktionen:
■ Quelle est la différence et pourquoi apparaît-elle? (Tu peux essaier plusieurs fois.) Application à la définition de fonctions:

In[19]:= **Remove[f];Remove[g];**

a) Unmittelbare Zuweisung. Beobachte, was b bewirkt:
■ Application immédiate. Observe ce que produit b:

In[20]:=
```
b = 3;
f[x_] = x + b
```

Out[21]= 3 + x

Veränderung des Wertes und Aufruf: Was bewirkt b?
■ Changement des valeurs et appel: Que produit b?

In[22]:=
```
b = 100;
f[5]
```

Out[23]= 8

b) Verzögerte Zuweisung. Beobachte, was b bewirkt:
■ Application retardée. Observe ce que produit b:

```
In[24]:= b = 3;
         g[x_] := x + b
```

Veränderung des Wertes und Aufruf: Was bewirkt b?

■ Changement de la valeur et de l'appel: Que produit b?

```
In[26]:= b = 100;
         g[5]

Out[27]= 105
```

Abruf der Definitionen:

■ Appel des définitions:

```
In[28]:= ?f

         Global`f

         f[x_] = 3 + x

In[29]:= ?g

         Global`g

         g[x_] := x + b
```

7.1.3. Mehrfache Zuordnungen
■ Applications multiples

In einem Schritt können mehrere Zuordnungen gemacht werden. *Mathematica* arbeitet die Zuordnungen von rechts nach links ab:

■ On peut faire plusieurs applications en une démarche. *Mathematica* achève les applications de droite à gauche:

```
In[30]:= a1 = a2 = a3

Out[30]= a3
```

Oder listenweise:

■ Ou par listes:

```
In[31]:= {u, v} = {4, 15}

Out[31]= {4, 15}

In[32]:= {s, t} = {u, v}

Out[32]= {4, 15}

In[33]:= {t, s}

Out[33]= {15, 4}
```

7.1.4. Einschub für 7.4.1.
(dort nicht anwendbar, wegen zu grossem Output)

Insertion pour 7.4.1.
(ne peut pas être appliqué à cause du trop grand output)

In[34]:= **??Out**

```
%n or Out[n] is a global object that is assigned to be the value produced
   on the nth output line. % gives the last result generated. %% gives the
   result before last. %% ... % (k times) gives the kth previous result. Mehr...

Attributes[Out] = {Listable, Protected}

Out[0] = 0

%1 = 5

%2 = 9

%3 = 3 x + y

%4 = 1/(3 x+y)

%5 = Null

%6 = a

%7 = {Null}

%8 = {Null}

%9 = 33444

%10 = 33444

%11 = 0.529863

%12 = 2

%13 = 3.52986

%14 = {Null}

%15 = {Null}

%16 = Null

%17 = {0.529863, 0.529863, 0.529863, 0.529863, 0.529863}

%18 = {0.320157, 0.260433, 0.706004, 0.425192, 0.767409}

%19 = Null

%20 = 3

%21 = 3 + x

%22 = 100

%23 = 8

%24 = 3

%25 = Null

%26 = 100

%27 = 105
```

```
%28 = {Null}

%29 = {Null}

%30 = a3

%31 = {4, 15}

%32 = {4, 15}

%33 = {15, 4}
```

7.1.5. Rekursive Zuordnungen
▮ Application récursive

Durch rekursive Zuweisung (rekursive Funktion) kann eine Liste erzeugt werden. Hier das Beispiel der Fakultäten:
▮ Par une application (fonction) récursive on peut créer une liste. Voici l'exemple pour des factoriels:

```
In[35]:= Remove[fac];
         fac[1]  = 1;
         fac[n_]:= fac[n] = n fac[n-1];

In[38]:= ?fac

         Global`fac

         fac[1] = 1

         fac[n_] := fac[n] = n fac[n - 1]

In[39]:= Table[fac[n],{n,1,5}]

Out[39]= {1, 2, 6, 24, 120}

In[40]:= fac[50];
         Timing[fac[50]]

Out[41]= {0. Second, 30414093201713378043612608166064768844377641568960512000000000000}
```

Probiere: ▮ Essaie:

```
In[42]:= Remove[fak];
         fak[1]  = 1;
         fak[n_]:= n fak[n-1];
         Table[fak[n],{n,1,5}]

Out[45]= {1, 2, 6, 24, 120}

In[46]:= fak[50];
         Timing[fak[50]]

Out[47]= {0. Second, 30414093201713378043612608166064768844377641568960512000000000000}
```

Wieso braucht *Mathematica* das zweite Mal weniger Zeit? Das erste Mal speichert es wegen der Zwischenzuweisung alle Werte ab und kann den gewünschten Wert dann aus dem Speicher holen. Das zweite Mal muss das Programm rechnen!
▮ Pourquoi *Mathematica* emploie-t-il moins de temps la deuxième fois? La première fois il mémorise toutes les valeurs

à cause des applications intermédiaires et peut sortir la valeur desirée de la mémoire. La deuxième fois le programme doit calculer!

7.2. Löschen der Wertzuweisung
▮ Effacer l'assignation des valeurs

7.2.1. Missing value einsetzen:
▮ Appliquer missing value

Beispiel: ▮ Exemple:

```
In[48]:= Remove[x] ; x = 1 ; Print["1)   ", x] ;
         x = . ; Print["2)   ",x]

         1)   1

         2)   x

In[50]:= ??x

         Global`x

In[51]:= x = 1 ; Print["1)   ", x]

         1)   1

In[52]:= ??x

         Global`x

         x = 1

In[53]:= x = . ; Print["2)   ",x]

         2)   x

In[54]:= ??x

         Global`x
```

7.2.2. Funktion löschen:
▮ Effacer la fonction

```
In[55]:= ?fak

         Global`fak

         fak[1] = 1

         fak[n_] := n fak[n - 1]

In[56]:= fak[1] = .

In[57]:= fak[n_] = .
```

In[58]:= **?fak**

```
Global`fak
```

Zwar ist kein Wert mehr zugewiesen, doch die Funktion ist noch bekannt.

- Aucune valeur n'est plus assignée, cependant la fonction est encore connue.

Anderes Beispiel:

- Autre exemple:

In[59]:= **kubus[x_] = x^3**

Out[59]= x^3

In[60]:= **?kubus**

```
Global`kubus

kubus[x_] = x^3
```

In[61]:= **Clear[kubus]**

In[62]:= **?kubus**

```
Global`kubus
```

Zwar ist keine Funktionsvorschrift mehr zugewiesen, doch der Name der Funktion ist noch bekannt.

- Aucune prescription de fonction n'est assignée. Cependant le nom de la fonction est encore connu.

In[63]:= **Attributes[kubus] = {Listable}**

Out[63]= {Listable}

In[64]:= **Clear[kubus]**

In[65]:= **?kubus**

```
Global`kubus

Attributes[kubus] = {Listable}
```

Sogar das Attribut ist noch bekannt.

- Même l'attribut est encore connu.

In[66]:= **ClearAll[kubus]**

In[67]:= **?kubus**

```
Global`kubus
```

Jetzt ist das Attribut nicht mehr bekannt.

- Maintenant l'attribut n'est plus connu.

In[68]:= **Remove[kubus]**

In[69]:= **?kubus**

```
Information::notfound : Symbol kubus not found. Mehr…
```

Jetzt ist auch der Name nicht mehr bekannt.
■ Maintenant même le nom n'est plus connu.

In[70]:= **Remove[Apply]**

```
Remove::rmptc : Symbol Apply is Protected and cannot be removed. Mehr…
```

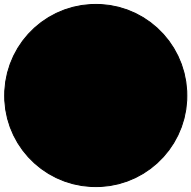
(Hier handelt es sich um einen *Mathematica*-Namen!)
■ (Ici il s'agit d'un nom de *Mathematica*))

7.2.3. Probleme mit Packages:
■ Problèmes de Packages:

Es kann passieren, dass man eine Funktion aus einem Package aufruft, bevor das Package geladen worden ist. Nach dem Aufruf erscheint vielleicht eine Meldung, worauf man dann den Aufruf des Packages nachholt. Da aber vom ersten Aufruf her die Funktion schon registriert ist, lässt sie sich jetzt nicht einfach überschreiben. Man muss sie erst wieder löschen. Die mit dem Package geladene Funktion bleibt dann bestehen. Beispiel:
■ Il peut arriver qu'on appelle une fonction d'un package avant le changement du package. Après l'appel il apparaît peut-être un message de rattraper l'appel du package. Mais comme la fonction a déjà été enregistrée au premier appel, elle ne se laisse pas surcharger maintenant. Il faut d'abord l'effacer. La fonction chargée du Package reste. Exemple:

In[71]:=
```
Show[Graphics[{CMYColor[0.6,0.5,0.4],PointSize[0.3],
          Point[{0,0}]}]];
```

```
General::spell1 :
 Possible spelling error: new symbol name "CMYColor" is similar to existing symbol "CMYKColor". Mehr…

Graphics::gprim :
 CMYColor[0.6, 0.5, 0.4] was encountered where a Graphics primitive or directive was expected. Mehr…
```



In[72]:= **Needs["Graphics`Colors`"]**

```
CMYColor::shdw : Symbol CMYColor appears in multiple contexts {Graphics`Colors`, Global`};
   definitions in context Graphics`Colors` may shadow or be shadowed by other definitions. Mehr…
```

In[73]:= **?CMYColor**

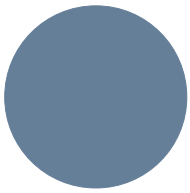
```
Global`CMYColor
```

In[74]:= **Remove[CMYColor]**

In[75]:= **?CMYColor**

```
CMYColor[c,m,y] represents a color in the CMY (cyan-magenta-yellow) system. Mehr…
```

```
In[76]:= Show[Graphics[{CMYColor[0.6, 0.5, 0.4], PointSize[0.3], Point[{0, 0}]}]];
```



7.2.4. Löschen aller eigenen Variablen und Funktionen:
▮ Effacer toutes les propres variables et fonctions:

Folgender Befehl löscht alle globalen Variablen und Funktionen, die mit einem Kleinbuchstabe oder mit "$" beginnen und nicht"protected" sind.
▮ L'ordre suivant efface toutes les variables et fonctions globales qui commencent par une minuscule ou par "$" et qui ne sont pas "prodected".

```
In[77]:= ?fak

         Global`fak
```

```
In[78]:= Remove["Global`*"]
```

```
In[79]:= ?fak

         Information::notfound : Symbol fak not found. Mehr…
```

Der folgende Befehl löscht alle Variablen und Funktionen, die nicht"protected" sind. Achtung vor der Anwendung!
▮ L'ordre suivant efface toutes les variables et fonctions qui ne sont pas "protected".

```
In[80]:= (*   Remove["@*"] old, Remove["Global`*"] new   *)
```

Falls was schief gegangen ist, kannst Du im Menu unter ("Action") "Kernel", "Kernels and Tasks..." einen neuen Kernel starten.
▮ Si quelque chose n'a pas fonctionné, tu peux démarrer un nouveau Kernel dans le menu sous ("Action") "Kernel", "Kernels and Tasks...".

7.3. Regeln
∎ Règles

7.3.1. Allgemeines
∎ Généralités

```
In[81]:= ?/.

         expr /. rules applies a rule or list of rules in
            an attempt to transform each subpart of an expression expr. Mehr…

In[82]:= ?ReplaceAll

         expr /. rules applies a rule or list of rules in
            an attempt to transform each subpart of an expression expr. Mehr…

In[83]:= ?Replace

         Replace[expr, rules] applies a rule or list of rules in an
            attempt to transform the entire expression expr. Replace[expr, rules,
            levelspec] applies rules to parts of expr specified by levelspec. Mehr…
```

In *Mathematica* gibt es also die Möglichkeit, nach der Berechnung durch Anwendung einer Ersetzungsregel einen Ausdruck durch etwas anderes zu ersetzen. Beispiele:

∎ Dans *Mathematica* il y a la possibilité, après un calcul où on emploie une règle de remplacement, de remplacer une expression par autre chose. Exemple:

```
In[84]:= x + 7y

Out[84]= x + 7 y

In[85]:= x + 7y /. x->4

Out[85]= 4 + 7 y

In[86]:= x + 7y /. {x->4, y->3}

Out[86]= 25

In[87]:= x + 7y

Out[87]= x + 7 y
```

Die Ersetzung ist also nicht bleibend! Beachte:

∎ Le remplacement n'est pas définitif! Observe:

```
In[88]:= 3x + 7x /. {3x->y, y->3}

Out[88]= 10 x
```

Da zuerst gerechnet wird, kommt die Ersetzungsregel gar nie zur Ausführung.

∎ Comme on commence par le calcul, la règle de remplacement n'arrive pas à être utilisée.

7.3.2. Verzögerte Ausführung der Regel
▮ Exécution retardée de la règle

Vergleiche: ▮ Compare:

```
In[89]:= ?->
```

```
lhs -> rhs represents a rule that transforms lhs to rhs. Mehr…
```

```
In[90]:= ?:>
```

```
lhs :> rhs represents a rule that transforms
   lhs to rhs, evaluating rhs only when the rule is used. Mehr…
```

Man hat also eine ähnliche Situation wie bei "=" und ":=". Beispiele:
▮ On a donc une situation semblable comme pour "=" et ":=". Exemple:

```
In[91]:= Table[x, {6}]
Out[91]= {x, x, x, x, x, x}

In[92]:= Table[x, {6}] /. x-> Random[]
Out[92]= {0.357011, 0.357011, 0.357011, 0.357011, 0.357011, 0.357011}
```

Offenbar ist erst gerechnet worden. Dann wurde Random[ ] einmal ausgeführt, darauf zugewiesen. x bleibt leer:
▮ Apparemment on a d'abord calculé. Ensuite on a exécuté une fois Random[ ], puis assigné. x reste vide.

```
In[93]:= x
Out[93]= x

In[94]:= Table[x, {6}] /. x:> Random[]
Out[94]= {0.799552, 0.201374, 0.507658, 0.690089, 0.183678, 0.286869}
```

Hier ist demnach Random[ ] jeweils vor der nächsten Zuweisung verzögert ausgeführt worden.
▮ On a donc retardé chaque fois l'exécution de Random[ ] avant la prochaine assignation.

7.3.3. Wiederholte Ersetzung
▮ Remplacement répété:

Studiere: ▮ Etudie:

```
In[95]:= ?//.
```

```
expr //. rules repeatedly performs replacements until expr no longer changes. Mehr…
```

Vergleiche: ▮ Compare:

```
In[96]:= log[a b c d] /. log[x_ y_] -> log[x] + log[y]

         General::spell1 :
          Possible spelling error: new symbol name "log" is similar to existing symbol "Log". Mehr…

Out[96]= log[a] + log[b c d]

In[97]:= log[a b c d] //. log[x_ y_] -> log[x] + log[y]

Out[97]= log[a] + log[b] + log[c] + log[d]
```

7.3.4. Ersetzung bei graphischen Optionen
▮ Remplacement lors d'options graphiques

Studiere, welche Optionen im nächsten Beispiel verzögert und welche gewöhnlich sind:

▮ Etudie quelles options, dans l'exemple suivant, sont retardées et lesquelles sont normales:

```
In[98]:= Options[ContourPlot]

Out[98]= {AspectRatio → 1, Axes → False, AxesLabel → None, AxesOrigin → Automatic,
           AxesStyle → Automatic, Background → Automatic, ColorFunction → Automatic,
           ColorFunctionScaling → True, ColorOutput → Automatic, Compiled → True,
           ContourLines → True, Contours → 10, ContourShading → True,
           ContourSmoothing → True, ContourStyle → Automatic, DefaultColor → Automatic,
           DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction,
           Epilog → {}, FormatType :→ $FormatType, Frame → True, FrameLabel → None,
           FrameStyle → Automatic, FrameTicks → Automatic, ImageSize → Automatic,
           PlotLabel → None, PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic,
           Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

7.3.5. Eine Anwendung: Labels bei Plots
▮ Une application: "Labels" des "Plots"

Aufgabe: Jeder Plot soll als Label das Kommando erhalten, das ihn generiert hat.

Studiere, was nun folgt:

▮ Problème: Chaque Plot reçoit en tant que Label le commandement qui l'a généré. Etudie ce qui en suit:

```
In[99]:= ?$Line

         $Line is a global variable that specifies the number of the current input line. Mehr…

In[100]:=
         $Line

Out[100]=
         100

In[101]:=
         ?InString

         InString[n] is a global object that is assigned to be the text of the nth input line. Mehr…
```

In[102]:=

```
?ToString
```

```
ToString[expr] gives a string corresponding to
   the printed form of expr in OutputForm. ToString[expr, form] gives
   the string corresponding to output in the specified form. Mehr...
```

In[103]:=

```
?StringJoin
```

```
"s1" <> "s2" <> ... , StringJoin["s1", "s2", ... ] or StringJoin[
   {"s1", "s2", ... }] yields a string consisting of a concatenation of the si. Mehr...
```

In[104]:=

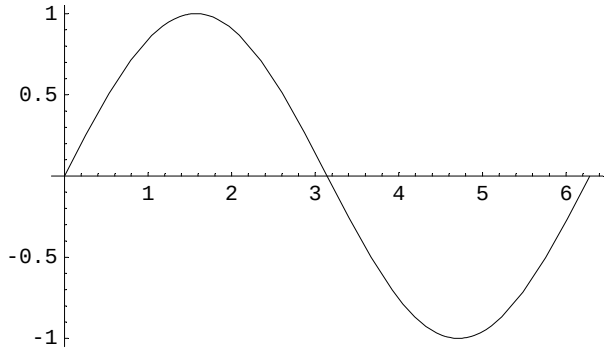
```
?SetOptions
```

```
SetOptions[s, name1->value1, name2->value2, ... ] sets the specified default
   options for a symbol s. SetOptions[stream, ... ] or SetOptions["name", ... ]
   sets options associated with a particular stream. SetOptions[object, ... ] sets
   options associated with an external object such as a NotebookObject. Mehr...
```

In[105]:=

```
SetOptions[Plot,PlotLabel :> StringJoin[
            "In[", ToString[$Line], "] :> ",
            InString[$Line]
            ]];
Plot[Sin[x],{x, 0, 2 Pi}];
```

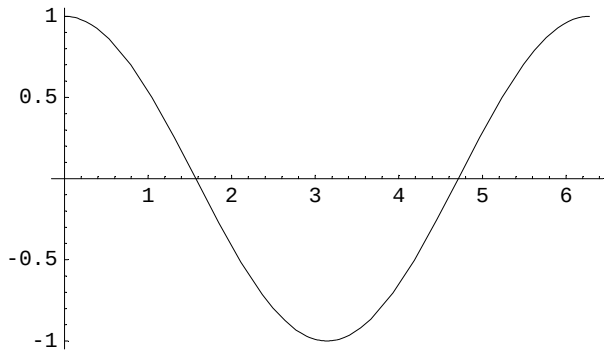
```
In[106] :> Plot[Sin[x],{x, 0, 2 Pi}];
```



In[107]:=

```
Plot[Cos[x],{x, 0, 2 Pi}];
```

```
In[107] :> Plot[Cos[x],{x, 0, 2 Pi}];
```



7.3.6. Vereinfachung von trigonometrischen Ausdrücken
▮ Simplification d'expressions trigonométriques

Dazu gab es ein Package. (Neu bei Build-in Functions)
▮ Pour cela il existait un Package. (Nouveau avec Build-in Functions)

In[108]:=
```
(* Needs["Algebra`Trigonometry`"] old ! *)
```

In[109]:=
```
??TrigReduce
```

```
TrigReduce[expr] rewrites products and powers of trigonometric functions
   in expr in terms of trigonometric functions with combined arguments. Mehr…

Attributes[TrigReduce] = {Protected}

Options[TrigReduce] = {Modulus → 0}
```

In[110]:=
```
?Trig*
```

System`

```
Trig       TrigFactor     TrigReduce
TrigExpand TrigFactorList TrigToExp
```

In[111]:=
```
(* ??TrigReduceRel Old *)
```

Offenbar kann (konnte) man Dinge nicht ansehen, die mit "Private" spezifiziert sind (waren). Beachte die Anwendung:
▮ Apparemment on ne peut (pouvait) pas regarder les choses qui sont (étaient) spécifiées par "Private". Tiens compte de l'application:

In[112]:=
```
TrigReduce[Sin[x + Pi]]
```

Out[112]=
```
-Sin[x]
```

Beachte die Ersetzungsregeln im Aufbau des folgenden Programms.
(Die Sache wird in einem späteren Kapitel erklärt. Hier ist nur Text - so läuft es nicht.)
▮ Considère les règles de remplacement dans la construction (structure) du programme suivant. (La chose sera expliquée dans un chapitre ultérieur. Ici il n'y a que du texte - cela ne va (tourne) pas ainsi.)

```
BeginPackage["Algebra`Trigonometry`", "Global`"]

TrigCanonical::usage = "TrigCanonical[expr] is obsolete.
      Its functionality is now built-in."
TrigExpand::usage = "TrigExpand[expr] is obsolete.
      Its functionality is provided by Expand[expr, Trig->True]."
TrigFactor::usage = "TrigFactor[expr] tries to write sums of trigonometric
      functions as products."
```

```
TrigReduce::usage = "TrigReduce[expr] writes trigonometric functions of
      multiple angles as sums of products of trigonometric functions of
      that angle."
TrigReduce::notes = "TrigReduce simplifies the arguments of trigonometric
      functions. It is, in a way, the inverse of TrigExpand."
TrigToComplex::usage = "TrigToComplex[expr] writes trigonometric functions
      in terms of complex exponentials."
ComplexToTrig::usage = "ComplexToTrig[expr] writes complex exponentials
      as trigonometric functions of a real angle."

Begin["`Private`"]
{`x, `y, `r, `n, `m, `a, `b, `c, `i, `e};

TrigCanonical[e_] := e

TrigExpand[___] := $Failed /;
   Message[TrigExpand::obsfn, TrigExpand, Expand]

TrigExpand[e_]:=Expand[e,Trig->True]

`TrigFactorRel = {
   a_. Sin[x_] + a_. Sin[y_] :> 2 a Sin[x/2+y/2] Cos[x/2-y/2],
   a_. Sin[x_] + b_. Sin[y_] :> 2 a Sin[x/2-y/2] Cos[x/2+y/2] /; a+b == 0,
   a_. Cos[x_] + a_. Cos[y_] :> 2 a Cos[x/2+y/2] Cos[x/2-y/2],
   a_. Cos[x_] + b_. Cos[y_] :> 2 a Sin[x/2+y/2] Sin[y/2-x/2] /; a+b == 0,
   a_. Tan[x_] + a_. Tan[y_] :> a Sin[x+y]/(Cos[x] Cos[y]),
   a_. Tan[x_] + b_. Tan[y_] :> a Sin[x-y]/(Cos[x] Cos[y]) /; a+b == 0,

   a_. Sin[x_] Cos[y_] + a_. Sin[y_] Cos[x_] :> a Sin[x + y],
   a_. Sin[x_] Cos[y_] + b_. Sin[y_] Cos[x_] :> a Sin[x - y] /; a+b == 0,
   a_. Cos[x_] Cos[y_] + b_. Sin[x_] Sin[y_] :> a Cos[x + y] /; a+b == 0,
   a_. Cos[x_] Cos[y_] + a_. Sin[x_] Sin[y_] :> a Cos[x - y]

}
TrigFactorRel = Dispatch[TrigFactorRel]
Protect[TrigFactorRel]

TrigFactor[e_] := FixedPoint[(# /. TrigFactorRel)&, e]

`TrigReduceRel = {

   Cos[n_Integer x_] :> 2^(n-1) Cos[x]^n +
        Sum[ Binomial[n-i-1, i-1] (-1)^i n/i 2^(n-2i-1) Cos[x]^(n-2i),
          {i, 1, n/2} ]              /; n > 0,

   Sin[m_Integer?OddQ x_] :>
        Block[{`p = -(m^2-1)/6, `s = Sin[x], `k},
            Do[s += p Sin[x]^k;
           p *= -(m^2 - k^2)/(k+2)/(k+1),
```

```
            {k, 3, m, 2}];
             m s]                                  /; m > 0,

   Sin[n_Integer?EvenQ x_] :>
        Sum[ Binomial[n, i] (-1)^((i-1)/2) Sin[x]^i Cos[x]^(n-i),
           {i, 1, n, 2} ]                  /; n > 0,

   Tan[n_Integer x_] :> Sin[n x]/Cos[n x],

   Sin[x_ + y_] :> Sin[x] Cos[y] + Sin[y] Cos[x],
   Cos[x_ + y_] :> Cos[x] Cos[y] - Sin[x] Sin[y],
   Tan[x_ + y_] :> (Tan[x] + Tan[y])/(1 - Tan[x] Tan[y]),

     Sin[r_Rational x_] :> (Sin[Numerator[r] `symb] /. TrigReduceRel /.
        `symb -> x/Denominator[r])            /; Numerator[r] != 1,
   Cos[r_Rational x_] :> (Cos[Numerator[r] `symb] /. TrigReduceRel /.
        `symb -> x/Denominator[r])            /; Numerator[r] != 1,

     Tan[x_/2] :> (1 - Cos[x])/Sin[x],
   Cos[x_/2]^(n_Integer?EvenQ) :>
                        ((1 + Cos[x])/2)^(n/2),
   Sin[x_/2]^(n_Integer?EvenQ) :>
                        ((1 - Cos[x])/2)^(n/2),
   Sin[x_/2]^n_. Cos[x_/2]^m_. :> Tan[x/2]^n                  /; m == -n,
   Sin[r_ x_.] Cos[r_ x_.]       :> Sin[2 r x]/2 /; IntegerQ[2r]

}
TrigReduceRel = Dispatch[TrigReduceRel]

TrigReduce[e_] := e //. TrigReduceRel

`TrigToComplexRel = {
   Sin[x_] :> -I/2*(-E^(-I*x) + E^(I*x)),
   Cos[x_] :>  (E^(-I*x) + E^(I*x))/2,
   Tan[x_] :> (-I*(-E^(-I*x) + E^(I*x)))/(E^(-I*x) + E^(I*x)),
   Csc[x_] :> (2*I)/(-E^(-I*x) + E^(I*x)),
   Sec[x_] :> 2/(E^(-I*x) + E^(I*x)),
   Cot[x_] :> (I*(E^(-I*x) + E^(I*x)))/(-E^(-I*x) + E^(I*x)),
   Si[x_]  :> -I/2(ExpIntegralE[1, I x] - ExpIntegralE[1, -I x]) + Pi/2,
   Ci[x_]  :> -1/2(ExpIntegralE[1, I x] + ExpIntegralE[1, -I x])
}
TrigToComplexRel = Dispatch[TrigToComplexRel]

TrigToComplex[e_] := e //. TrigToComplexRel

`ComplexToTrigRel = {
   Exp[a_ b_Plus] :> Exp[Expand[a b]],
   Exp[c_Complex x_. + y_.] :> Exp[Re[c] x + y] (Cos[Im[c] x] + I Sin[Im[c] x])
}
```

```
ComplexToTrigRel = Dispatch[ComplexToTrigRel]

ComplexToTrig[e_] := Cancel[e //. ComplexToTrigRel]

End[]   (* Algebra`Trigonometry`Private` *)

Protect[ TrigCanonical, TrigExpand, TrigFactor, TrigReduce,
        TrigToComplex, ComplexToTrig ]

EndPackage[]
```

Das Original befindet (befand) sich im Root-Verzeichnis unter "LocalLibrary", "Mathematica", "Packages", "Algebra", "Trigonometry.m" (NeXT).

■ L'original se trouve (touvait) dans le directoire Root sous "LocalLibrary", "Mathematica", "Packages", "Algebra", "Trigonometry.m" (NeXT).

7.3.7. Wertelisten, Extraktion von Werten aus Zuweisungsregeln

■ Listes de valeurs, extraction de valeurs de règles d'assignation

Studiere die folgende Anwendung:

■ Etudie l'application suivante:

In[113]:=

```
Needs["Statistics`DescriptiveStatistics`"]
```

In[114]:=

```
dReport = DispersionReport[{3,4,2,3,6,7,4,8,9,
                            5,2,4}]
```

Out[114]=

$$\left\{\texttt{Variance} \to \frac{233}{44},\ \texttt{StandardDeviation} \to \frac{\sqrt{\frac{233}{11}}}{2},\ \texttt{SampleRange} \to 7,\right.$$
$$\left.\texttt{MeanDeviation} \to \frac{15}{8},\ \texttt{MedianDeviation} \to \frac{3}{2},\ \texttt{QuartileDeviation} \to \frac{7}{4}\right\}$$

Das Resultat ist also eine Liste von Regeln! Nun wollen wir einen Wert extrahieren:

■ Le résultat est donc une liste de règles! Maintenant nous en extrayons une valeur:

In[115]:=

```
MeanDeviation /. dReport
```

Out[115]=

$$\frac{15}{8}$$

Oder wir können das Resultat in eine Variable speichern:

■ Oubien nous pouvons mémoriser le résultat dans une variable:

In[116]:=
```
x1 = MeanDeviation /. dReport
```

Out[116]=
$$\frac{15}{8}$$

In[117]:=
```
x1
```

Out[117]=
$$\frac{15}{8}$$

7.3.8. Weiterrechnen mit Werten aus Output in Form von Ersetzungsregeln
▮ Continuer de calculer avec des valeurs de l'output en forme de règles de remplacement

Manchmal erscheint der Output in Form von Ersetzungsregeln.
Mit ihnen kann z.B. wie folgt direkt weitergerechnet werden (Beispiel):

▮ Parfois l'output apparaît en forme de règles de remplacement. Avec celles-ci on peut continuer de calculer par exemple comme il suit:

In[118]:=
```
Remove[x]
```

In[119]:=
```
wurzeln = Solve[x^2 + 13x - 34 == 0]
```

Out[119]=
$$\left\{\left\{x \to \frac{1}{2}\left(-13-\sqrt{305}\right)\right\}, \left\{x \to \frac{1}{2}\left(-13+\sqrt{305}\right)\right\}\right\}$$

Hier ist ein solcher Output in Form von Ersetzungsregeln gegeben. Wir verwenden diese Regeln, die nun den Namen "wurzeln" haben, direkt hinter "/." weiter. Wir wollen die Lösungen wieder in die Gleichung einsetzen:

▮ Voici un tel output en forme de règles de remplacement. Nous continuons d'employer ces règles qui ont dès maintenant le nom "racines" directement derrière "/." . Nous allons remplacer la solution dans l'équation:

In[120]:=
```
x^2 + 13x - 34 == 0 /. wurzeln
```

```
N::meprec : Internal precision limit $MaxExtraPrecision =
   49.99999999999999` reached while evaluating -34 + 13/2 (-13 - Sqrt[305]) + 1/4 (-13 - Sqrt[305])^2. Mehr…

N::meprec : Internal precision limit $MaxExtraPrecision =
   49.99999999999999` reached while evaluating -34 + 13/2 (-13 + Sqrt[305]) + 1/4 (-13 + Sqrt[305])^2. Mehr…
```

Out[120]=
$$\left\{-34+\frac{13}{2}\left(-13-\sqrt{305}\right)+\frac{1}{4}\left(-13-\sqrt{305}\right)^2==0,\ -34+\frac{13}{2}\left(-13+\sqrt{305}\right)+\frac{1}{4}\left(-13+\sqrt{305}\right)^2==0\right\}$$

In[121]:=
```
Simplify[%]
```

Out[121]=
```
{True, True}
```

Oder: ■ Ou:

In[122]:=
```
x /. wurzeln[[1]]
```

Out[122]=
$$\frac{1}{2}\left(-13-\sqrt{305}\right)$$

Die erste Regel wurde extrahiert! Oder:
■ Cette règle a été extraite! Ou:

In[123]:=
```
2x + 13 /. wurzeln[[1]]
```

Out[123]=
$$-\sqrt{305}$$

7.4. Gleichheit
■ Egalité

7.4.1. Allgemeines
■ Généralités

Studiere: ■ Etudie:

In[124]:=
```
?==
```

```
lhs == rhs returns True if lhs and rhs are identical. Mehr…
```

Anwendung: ■ Application:

In[125]:=
```
Expand[(x + y)^2] == x^2 + 2x y + y^2
```

Out[125]=
```
True
```

Vergleich mit "=":
■ Comparaison avec "=":

In[126]:=
```
Expand[(x + y)^2] =  x^2 + 2x y + y^2
```

```
Set::write : Tag Expand in Expand[(x + y)^2] is Protected. Mehr…
```

Out[126]=
$$x^2 + 2\,x\,y + y^2$$

Eingebaute Funktionen sind "protected". Man kann dies allerdings ändern. Beispiel:

■ Les fonctions incorporées sont "prodected". Mais on peut changer cela. Exemple:

In[127]:=
```
?Out
```
```
%n or Out[n] is a global object that is assigned to be the value produced
   on the nth output line. % gives the last result generated. %% gives the
   result before last. %% ... % (k times) gives the kth previous result. Mehr…
```

In[128]:=
```
Out = 1
```
```
Set::wrsym : Symbol Out is Protected. Mehr…
```

Out[128]=
```
1
```

In[129]:=
```
Attributes[Out]
```

Out[129]=
```
{Listable, Protected}
```

Achtung: Das gibt viel Output..... Vgl. 7.1.4.

■ Attention: Cela fait beaucoup d'output.... Compare 7.1.4.

In[130]:=
```
(*  ??Out   *)
```

In[131]:=
```
?Unprotect
```
```
Unprotect[s1, s2, ... ] removes the attribute Protected
   for the symbols si. Unprotect["form1", "form2", ... ] unprotects
   all symbols whose names textually match any of the formi. Mehr…
```

In[132]:=
```
Unprotect[Out]
```

Out[132]=
```
{Out}
```

In[133]:=
```
Out = 2
```

Out[133]=
```
2
```

In[134]:=
```
Clear[Out]; Protect[Out]
```

Out[134]=
```
{Out}
```

Wieviel Output ist es noch?

■ Combien d'output cela fait-il encore?

In[135]:=
```
Out = 3
```
```
Set::wrsym : Symbol Out is Protected. Mehr…
```
Out[135]=
```
3
```

7.4.2. Konvertierung von Gleichheiten in Zuweisungsregeln
▮ Convertir des églaités en règles d'assignation

Studiere: ▮ Etudie:

In[136]:=
```
?ToRules
```
```
ToRules[eqns] takes logical combinations of equations, in the form generated by Roots and
   Reduce, and converts them to lists of rules, of the form produced by Solve. Mehr…
```
In[137]:=
```
NRoots[x^3 - 5 x^2 - 19 x + 66 == 0 , x]
```
Out[137]=
```
x == -3.9708 || x == 2.61524 || x == 6.35557
```

Der Output kommt hier in Form einer logischen oder-Verknüpfpung.
Wir wollen daraus Regeln machen:
▮ L'output se manifeste ici en forme de composition (avec "ou"). Nous voulons en faire des règles:

In[138]:=
```
ToRules[%]
```
Out[138]=
```
Sequence[{x → -3.9708}, {x → 2.61524}, {x → 6.35557}]
```

7.4.2. Weitere Probleme mit "Gleichheit"
▮ Autres problèmes avec "églaité"

Probiere aus: ▮ Essaie:

In[139]:=
```
?=
```
```
lhs = rhs evaluates rhs and assigns the result to be the value of lhs. From then on, lhs is
   replaced by rhs whenever it appears. {l1, l2, ... } = {r1, r2, ... } evaluates
   the ri, and assigns the results to be the values of the corresponding li. Mehr…
```
In[140]:=
```
?==
```
```
lhs == rhs returns True if lhs and rhs are identical. Mehr…
```
In[141]:=
```
?===
```
```
lhs === rhs yields True if the expression
   lhs is identical to rhs, and yields False otherwise. Mehr…
```

In[142]:=
```
?!=
```
lhs != rhs returns False if lhs and rhs are identical. Mehr...

In[143]:=
```
?=!=
```
lhs =!= rhs yields True if the expression
lhs is not identical to rhs, and yields False otherwise. Mehr...

In[144]:=
```
?Set
```
lhs = rhs evaluates rhs and assigns the result to be the value of lhs. From then on, lhs is replaced by rhs whenever it appears. {l1, l2, ... } = {r1, r2, ... } evaluates the ri, and assigns the results to be the values of the corresponding li. Mehr...

In[145]:=
```
?Equal
```
lhs == rhs returns True if lhs and rhs are identical. Mehr...

In[146]:=
```
?SameQ
```
lhs === rhs yields True if the expression
lhs is identical to rhs, and yields False otherwise. Mehr...

In[147]:=
```
?Unequal
```
lhs != rhs returns False if lhs and rhs are identical. Mehr...

In[148]:=
```
?UnsameQ
```
lhs =!= rhs yields True if the expression
lhs is not identical to rhs, and yields False otherwise. Mehr...

Studiere: ■ Etudie:

In[149]:=
```
N[Pi, 5] == N[Pi, 30]
```
Out[149]=
```
True
```

Hier hat *Mathematica* nur die Gleichheit bis auf die kleinere angegebene Stellenzahl geprüft.
■ Ici *Mathematica* n'a examiné l'égalité que jusqu'au plus petit degré indiqué.

In[150]:=
```
Clear[x]; Clear[y];
```
In[151]:=
```
x == y
```
Out[151]=
```
x == y
```

Hier gab es nichts zu prüfen.
■ Ici il n'y avait rien à examiner.

In[152]:=
```
1 != 2
```

Out[152]=
```
True
```

Das stimmt. ■ C'est vrai.

Im folgenden Beispiel wird die syntaktische Gleichheit geprüft, auch wieder bis auf die kleinste angegebene Genauigkeit:
■ Dans l'exemple suivant on examine l'égalité syntaxique de nouveau jusqu'à la plus petite exactitude donnée.

In[153]:=
```
N[Pi, 3] === N[Pi, 30]
```

Out[153]=
```
True
```

Oder: ■ Ou:

In[154]:=
```
N[Pi, 3] =!= N[Pi, 30]
```

Out[154]=
```
False
```

In[155]:=
```
x =!= y
```

Out[155]=
```
True
```

In[156]:=
```
x === y
```

Out[156]=
```
False
```

"Putzmaschine" einsetzen
■ Employer la "machine de nettoyage"

In[157]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[158]:=
```
Remove["Global`*"]
```

Uebungen ▮ Exercices

7. Probleme zum Thema "Zuordnungen und Regeln"
▮ Problèmes au sujet des "applications et règles"

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 7 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 7.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

f und g seien verschiedentlich definiert. Finde f[3+a] sowie g[3+a].
▮ f et g soient définis différemment. Trouve f[3+a] et g[3+a].

```
In[1]:= Clear[f];Clear[g];
```

a)

```
In[2]:= f[x_]=x/.a->7
Out[2]= x
```

b)

```
In[3]:= g[x_]:=x/.a->7
```

c)

```
In[4]:= f[3+a]
Out[4]= 3 + a
```

d)

```
In[5]:= g[3+a]
Out[5]= 10
```

e)

```
In[6]:= f[x_]=x + %
Out[6]= 10 + x
```

f)

```
In[7]:= g[x_]:=x + %
```

g)

```
In[8]:= f[x_]=N[x]
Out[8]= x
```

h)

```
In[9]:= g[x_]:=N[x]
```

i)

```
In[10]:= f[x_]=Expand[2x]
Out[10]= 2 x
```

j)

```
In[11]:= g[x_]:=Expand[2x]
```

k)

```
In[12]:= ?f
        Global`f
         f[x_] = 2 x
```

l)

```
In[13]:= ?g
        Global`g
         g[x_] := Expand[2 x]
```

Aufgabe 2 ▮ Problème 2

Probiere aus:
▮ Essaie:

```
In[14]:= Integrate[x^2,x]
```

$$\text{Out[14]= } \frac{x^3}{3}$$

Wieso wohl schreibt *Mathematica* vor die Input line .... := und vor die Output line .... =, ohne den Doppelpunkt?
▮ Pourquoi est-ce que *Mathematica* écrit devant la ligne Input.....:= et devant la ligne Output ....=, sans le double point?

Aufgabe 3 ▮ Problème 3

a)

Verdoppelung jeweils des zweiten Elements:
▮ Redoublement du deuxième élément:

```
In[15]:= {{1,1},{2,2},{3,3}}/.{x_,y_}:>{x,2y}
Out[15]= {{1, 2}, {2, 4}, {3, 6}}
```

b)

Gleiches Vorgehen wie in a), nur mit kleinerer Menge:
▮ Même procédé que dans a), seulement avec un ensemble plus petit:

```
In[16]:= {{1,1},{2,2}      }/.{x_,y_}:>{x,2y}
Out[16]= {{1, 1}, {4, 4}}
```

Wieso hat die Regel jetzt anders funktioniert? (Hinweis: Welche Möglichkeiten bestehen jeweils, das Paar zu interpretieren?)
▮ Pourquoi la règle a-t-elle fonctionné autrement? (Indication: Quelles possibilités y a-t-il pour interpréter la paire?)

Aufgabe 4 ▮ Problème 4

Eigene Regel schreiben:
▮ Ecrire une propre règle:

Schreibe eine Regel, die einen Punkt (x, y, z) des 3-dimensionalen Raumes projiziert in den Punkt des 2-dimensionalen Raumes (x, y):
▮ Ecris une règle, qui projette un point (x, y, z) de l'espace à 3 dimensions dans un point de l'espace à 2 dimensions (x, y):

Regel: ▮ Règle:

```
In[17]:= p[x_,y_,z_]={x,y,z}/.{x_,y_,z_}:>{x,y}
Out[17]= {x, y}
```

Anwendung: ▮ Application:

```
In[18]:= p[7,4,9]
Out[18]= {7, 4}
```

Aufgabe 5 ▮ Problème 5

Eigene Regel schreiben:
▮ Ecrire une propre règle:

Schreibe eine Regel, die aus einer Liste von 4 Paaren {xi, yi} und einer Liste von 4 Werten zi eine neue Liste von Tripeln {xi, yi, zi} macht:
▮ Ecris une règle qui fait d'une liste de 4 paires{xi, yi} et d'une liste de 4 valeurs zi une nouvelle liste de triples {xi, yi, zi}:

Paare: ▮ Paires:

```
In[19]:= paare = Table[{x[i],y[i]},{i,4}]
Out[19]= {{x[1], y[1]}, {x[2], y[2]}, {x[3], y[3]}, {x[4], y[4]}}
```

z-Werte: ▮ Valeurs z

```
In[20]:= zWerte = Table[z[i],{i,4}]
Out[20]= {z[1], z[2], z[3], z[4]}
```

Noch nicht gewünschte Liste:
▮ List pas encore désirée:

In[21]:= **{paare,zWerte}**

Out[21]= {{{x[1], y[1]}, {x[2], y[2]}, {x[3], y[3]}, {x[4], y[4]}}, {z[1], z[2], z[3], z[4]}}

In[22]:= **{paare,zWerte}//MatrixForm**

Out[22]//MatrixForm=

$$\begin{pmatrix} \{x[1], y[1]\} & \{x[2], y[2]\} & \{x[3], y[3]\} & \{x[4], y[4]\} \\ z[1] & z[2] & z[3] & z[4] \end{pmatrix}$$

In[23]:= **{paare,zWerte}//Transpose//MatrixForm**

Out[23]//MatrixForm=

$$\begin{pmatrix} \{x[1], y[1]\} & z[1] \\ \{x[2], y[2]\} & z[2] \\ \{x[3], y[3]\} & z[3] \\ \{x[4], y[4]\} & z[4] \end{pmatrix}$$

In[24]:=
```
{paare,zWerte}/.{xyListe_List,zListe_List}:>
 Transpose[xyListe] // MatrixForm
```

Out[24]//MatrixForm=

$$\begin{pmatrix} x[1] & x[2] & x[3] & x[4] \\ y[1] & y[2] & y[3] & y[4] \end{pmatrix}$$

In[25]:=
```
{paare,zWerte}/.{xyListe_List,zListe_List}:>
 Append[Transpose[xyListe],zListe] // MatrixForm
```

Out[25]//MatrixForm=

$$\begin{pmatrix} x[1] & x[2] & x[3] & x[4] \\ y[1] & y[2] & y[3] & y[4] \\ z[1] & z[2] & z[3] & z[4] \end{pmatrix}$$

Gewünschte Liste:

■ Liste désirée:

In[26]:=
```
{paare,zWerte}/.{xyListe_List,zListe_List}:>
 Transpose[Append[Transpose[xyListe],zListe]]//
                  MatrixForm
```

Out[26]//MatrixForm=

$$\begin{pmatrix} x[1] & y[1] & z[1] \\ x[2] & y[2] & z[2] \\ x[3] & y[3] & z[3] \\ x[4] & y[4] & z[4] \end{pmatrix}$$

Aufgabe 6 ▮ Problème 6

Was geschieht?
▮ Que se passe-t-il?

(a) Wieso findet trotz der Regel kein Ausmultiplizieren statt?
▮ Pourquoi ne se produit-il pas de multiplication malgré la règle?

```
In[27]:= y (x+2)(x-2)/. y z_ -> y Expand[z]
Out[27]= (-2 + x) (2 + x) y
```

(b) Verbesserung
▮ Correction

Schreibe eine Regel, die das Produkt (x+2)(x-2) ausmultipliziert.
▮ Ecris une règle qui multiple le produit (x+2)(x-2).

```
In[28]:= y (x+2)(x-2)/. y z_ :> y Expand[z]
Out[28]= (-4 + x^2) y
```

Hinweis: Was ist der Unterschied bezüglich Parameter bei fixer und bei verzögerter Zuordnung?
▮ Indication: Quelle est la différence par rapport aux paramètres lors d'une assignation fixe et d'une assignation retardée?

Aufgabe 7 ▮ Problème 7

Frage:
▮ Question:

Ein altes Package enthält die folgende Regel:
▮ Un vieux Package contient la règle suivante:

```
In[29]:= a_. Cos[x_] Cos[y_] + a_. Sin[x_] Sin[y_] :>
                a Cos[x - y]
Out[29]= Cos[x_] Cos[y_] a_. + a_. Sin[x_] Sin[y_] :> a Cos[x - y]
```

Wieso enthält es dazu nicht auch die folgende Regel?
▮ Pourquoi ne contient-il pas non plus la règle suivante?

```
In[30]:= a. Cos[x_ - y_] :>
         a Cos[x] Cos[y] + a Sin[x] Sin[y]

Out[30]= a.Cos[x_ - y_] :> a Cos[x] Cos[y] + a Sin[x] Sin[y]
```

Hinweise: Unendlicher loop .....
Indications: Loop infini .....

Aufgabe 8 Problème 8

Anwendung auf eine Differentialgleichung: Application pour une équation différentielle:

Verwende DSolve, um die folgende D'Gl. zu lösen: y'=y:
Utilise DSolve pour résoudre l'équation différentielle suivante: y'=y:

```
In[31]:= DSolve[y'[x]==y[x],y[x],x]

Out[31]= {{y[x] → ℯ^x C[1]}}
```

Probiere, aus der Lösungsliste die Resultatsfunktion herauszugreifen und wieder in die Gleichung einzusetzen, um zu überprüfen, ob das Resultat stimmt!
Essaie de prender la fonction des résultats dans la liste des solutions et de l'insérer dans l'équation différentielle!

```
In[32]:= r = Flatten[DSolve[y'[x]==y[x],y[x],x]]

Out[32]= {y[x] → ℯ^x C[1]}

In[33]:= s[x]:=y[x]/.r ; s[x]

Out[33]= ℯ^x C[1]
```

Einsetzen: Insérer:

```
In[34]:= D[s[x],x] == s[x]

Out[34]= True
```

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

```
In[35]:= (* Old Form: Remove["Global`@*"] *)

In[36]:= Remove["Global`*"]
```

Kurs ▮ Cours

8. Datentypen
▮ Types de données

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 8 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 8.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Mathematica kennt keine Deklarationen von Typen für die Variablen wie z.B. in klassischen Programmiersprachen, z.B. Modula II. Trotzdem.... *Mathematica* kann selber merken, um welchen Typ es sich handelt...
▮ *Mathematica* ne connaît pas de déclarations de types pour les variables comme p.ex. dans des langues de programmation classiques, p.ex. Modula II. Pourtant.... *Mathematica* s'aperçoit de quel type il s'agit....

8.1. Atomare Typen
▮ Types atomiques

8.1.1. Rufe die Information selbst ab
▮ Appelle l'information toi-même

Jeder Ausdruck ist aus folgenden "atomaren Typen" zusammengesetzt: Integer, Real, Rational, Complex, Symbol, String.
▮ Chaque expression est composée des "types atomiques" suivants: Integer, Real, Rational, Complex, String.

```
In[1]:= ??Integer

        Integer is the head used for integers. Mehr…

        Attributes[Integer] = {Protected}
```

In[2]:= **?Real**

Real is the head used for real (floating-point) numbers. Mehr...

In[3]:= **?Rational**

Rational is the head used for rational numbers. Mehr...

In[4]:= **?Complex**

Complex is the head used for complex numbers. Mehr...

In[5]:= **?Symbol**

Symbol["name"] refers to a symbol with the specified name. Mehr...

In[6]:= **?String**

String is the head of a character string "text". Mehr...

8.1.2. Probiere aus:
▮ Essaie:

In[7]:= **?Head**

Head[expr] gives the head of expr. Mehr...

In[8]:= **Head[3452]**

Out[8]= Integer

In[9]:= **Head[67.94]**

Out[9]= Real

In[10]:= **Head[3 + 4 I]**

Out[10]= Complex

In[11]:= **Head[E]**

Out[11]= Symbol

In[12]:= **Head[Pi]**

Out[12]= Symbol

In[13]:= **Head["Tschou zämä!"]**

Out[13]= String

8.2. Andere Typen und Prädikatenfunktionen
▮ Autres types et fonctions de prédicats

8.2.1. Beispiele
▮ Exemples

Neben den atomaren Typen gibt es übergeordnete Typen. Z.B. Integer, Real,
Rational u.s.w. sind Zahlen. Ein Vektor ist eine Liste etc.. Um zu prüfen, um welchen Typ es sich gerade handelt, existieren sogenannte Prädikatenfunktionen. Beispiele:

▮ A part des types atomiques il y a des types supérieurs. P.ex. Integer, Real, Rational etc. sont des nombres. Un vecteur est une liste etc.. Pour examiner de quel type il s'agit, il existe les dites fonctions de prédicats.

```
In[14]:= NumberQ[5/2 + 8 I/5]
Out[14]= True

In[15]:= NumberQ[5/2 + 8 I/5 - Pi]
Out[15]= False

In[16]:= VectorQ[{a, b, c, d, e, f, g}]
Out[16]= True
```

8.2.2. Uebersicht
▮ Vue d'ensemble

Die Prädikatenfunktionen enden mit "Q". Ihr Wertebereich ist {True, False}. Hier sind sie alle:

▮ Les fonctions de prédicats se terminent par "Q". Leur domaine de valeurs est "True, False". Voici tous:

In[17]:= **?*Q**

System`

```
ArgumentCountQ       LinkConnectedQ  PartitionsQ
ArrayQ               LinkReadyQ      PolynomialQ
AtomQ                ListQ           PrimeQ
DigitQ               LowerCaseQ      SameQ
EllipticNomeQ        MachineNumberQ  StringFreeQ
EvenQ                MatchLocalNameQ StringMatchQ
ExactNumberQ         MatchQ          StringQ
FreeQ                MatrixQ         SyntaxQ
HypergeometricPFQ    MemberQ         TensorQ
InexactNumberQ       NameQ           TrueQ
IntegerQ             NumberQ         UnsameQ
IntervalMemberQ      NumericQ        UpperCaseQ
InverseEllipticNomeQ OddQ            ValueQ
LegendreQ            OptionQ         VectorQ
LetterQ              OrderedQ
```

8.3. Interne Darstellung in *Mathematica*
▮ Représentation intérieure dans *Mathematica*

8.3.1. Allgemeines
▮ Généralités

Operationszeichen oder spezielle Symbole wie "+", "/", "#", "/@" u.s.w. werden intern dargestellt durch Funktionen. Mit den Funktionen "FullForm[...]", "FullForm[Hold[...]]" oder "TreeForm" kann die interne Darstellung in *Mathematica* sichtbar gemacht werden.

▮ Signes d'opérations ou symboles spéciaux tels que "+", "/", "#", "/@" etc. sont représentés intérieurement par des fonctions. Par les fonctions "FullForm[...]", "FullForm[Hold[...]]" ou "TreeForm" on peut rendre visible la représentation intérieure dans *Mathematica*.

In[18]:= **?FullForm**

```
FullForm[expr] prints as the full form of expr, with no special syntax.
```
Mehr…

In[19]:= **?TreeForm**

```
TreeForm[expr] prints with different levels in expr shown at different depths.
```
Mehr…

Beispiele: ▮ Exemples:

In[20]:= **FullForm[4/7]**

Out[20]//FullForm=
```
Rational[4, 7]
```

```
In[21]:= FullForm[4 - 7 I]

Out[21]//FullForm=
        Complex[4, -7]

In[22]:= FullForm[f[x]]

Out[22]//FullForm=
        f[x]

In[23]:= FullForm[x y^z / w]

Out[23]//FullForm=
        Times[Power[w, -1], x, Power[y, z]]

In[24]:= TreeForm[x y^z / w]

Out[24]//TreeForm=
        Times[ |            , x, |          ]
              Power[w, -1]     Power[y, z]

In[25]:= FullForm[2 + 3 15]

Out[25]//FullForm=
        47
```

8.3.2. Erzwungene oder verhinderte Ausführung
■ Exécution obtenue de force ou empêchée

Im letzten Beispiel ist zuerst die Rechnung ausgeführt worden. Erst darauf wurde dann die Funktion "FullForm" angewandt. Die gültige Abarbeitungshierarchie kann aber geändert werden. Dazu dienen folgende Befehle:
■ Dans le dernier exemple on a d'abord exécuté le calcul. Ensuite seulement on a appliqué la fonction "FullForm". On peut changer la hiérarchie de l'exécution valable par les ordres suivants:

```
In[26]:= ?Hold

        Hold[expr] maintains expr in an unevaluated form. Mehr…

In[27]:= ?HoldForm

        HoldForm[expr] prints as the expression expr, with expr maintained in an unevaluated form.
         Mehr…

In[28]:= ?ReleaseHold

        ReleaseHold[expr] removes Hold, HoldForm, HoldPattern and HoldComplete in expr. Mehr…

In[29]:= ?Evaluate

        Evaluate[expr] causes expr to be evaluated even if it appears as the argument of
           a function whose attributes specify that it should be held unevaluated. Mehr…
```

Beispiele: ■ Exemples

```
In[30]:= FullForm[Hold[2 + 7 - 4]]

Out[30]//FullForm=
        Hold[Plus[2, 7, -4]]
```

In[31]:=
```
Hold[2 + 7 - 4]
```

Out[31]=
```
Hold[2 + 7 - 4]
```

In[32]:=
```
HoldForm[2 + 7 - 4]
```

Out[32]=
```
2 + 7 - 4
```

In[33]:=
```
ReleaseHold[Hold[2 + 7 - 4]]
```

Out[33]=
```
5
```

In[34]:=
```
meineListe = {"Or", "And", "Abs"};
Attributes[meineListe]
```

Out[35]=
```
{}
```

In[36]:=
```
Attributes[Evaluate[meineListe]]
```

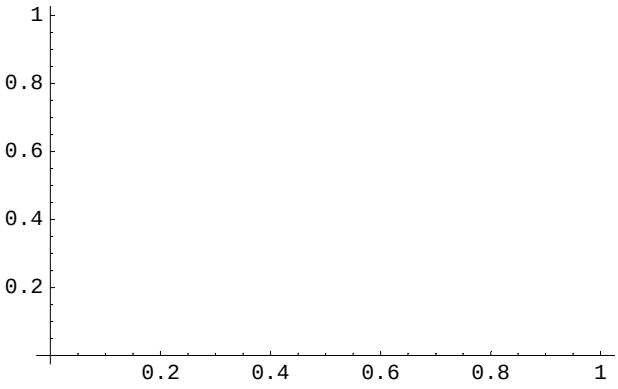
Out[36]=
```
{{Flat, HoldAll, OneIdentity, Protected},
 {Flat, HoldAll, OneIdentity, Protected}, {Listable, NumericFunction, Protected}}
```

In[37]:=
```
t[x_] = Table[x^n, {n,4}]
```

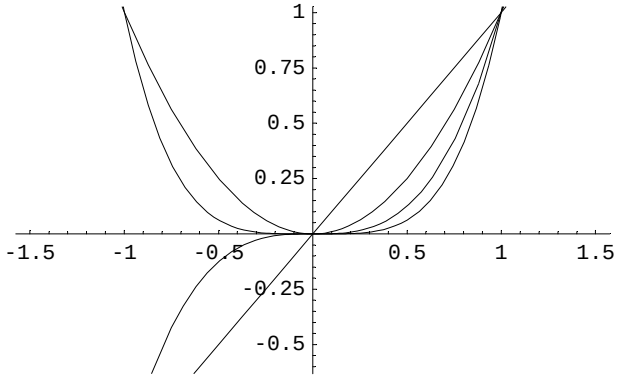
Out[37]=
```
{x, x^2, x^3, x^4}
```

In[38]:=
```
Plot[t[x], {x, -1.5, 1.5}];
```

```
Plot::plnr : t[x] is not a machine-size real number at x = -1.5. Mehr…

Plot::plnr : t[x] is not a machine-size real number at x = -1.3783. Mehr…

Plot::plnr : t[x] is not a machine-size real number at x = -1.24557. Mehr…

General::stop : Further output of Plot::plnr will be suppressed during this calculation. Mehr…
```



In[39]:= **Plot[Evaluate[t[x]], {x, -1.5, 1.5}];**



Was ist passiert?

■ Que s'est-il passé?

8.3.3. Der "Kopf" eines Ausdrucks und "FullForm"
■ La "tête" d'une expression et "FullForm"

Vergleiche: ■ Comparer:

In[40]:= **Head[x + y]**

Out[40]= Plus

In[41]:= **FullForm[x + y]**

Out[41]//FullForm=
```
Plus[x, y]
```

In[42]:= **Head[FullForm[x + y]]**

Out[42]= FullForm

In[43]:= **FullForm[{x, y}]**

Out[43]//FullForm=
```
List[x, y]
```

In[44]:= **Head[{x, y}]**

Out[44]= List

In[45]:= **FullForm[Cos[x^3 - 4 x^2]]**

Out[45]//FullForm=
```
Cos[Plus[Times[4, Power[x, 2]], Times[-1, Power[x, 3]]]]
```

In[46]:= **Head[Cos[x^3 - 4 x^2]]**

Out[46]= Cos

8.3.4. Veränderung des "Kopfs"
▮ Changement de la "tête"

Vergleiche: ▮ Comparer:

```
In[47]:= FullForm[{x, y, z, w}]

Out[47]//FullForm=
        List[x, y, z, w]

In[48]:= FullForm[x + y + z + w]

Out[48]//FullForm=
        Plus[w, x, y, z]

In[49]:= {x, y, z, w} /. List -> Plus

Out[49]= w + x + y + z
```

Was ist passiert? Der "Kopf" ist ausgewechselt worden! Doch *Mathematica* kann noch mehr: Mit "Apply" lässt sich ein weiterer "Kopf" einem Ausdruck voranstellen:
▮ Que s'est-il passé? On a remplacé la "tête"! *Mathematica* peut cependant davantage. Pour "Apply" on peut placer devant une expression une autre "tête" encore.

```
In[50]:= FullForm[{x, y, z, w}]

Out[50]//FullForm=
        List[x, y, z, w]

In[51]:= summe = Apply[Plus, {x, y, z, w}]

Out[51]= w + x + y + z

In[52]:= FullForm[summe]

Out[52]//FullForm=
        Plus[w, x, y, z]

In[53]:= Head[summe]

Out[53]= Plus
```

Oder so: ▮ Ou ainsi:

```
In[54]:= Apply[Plus, Range[1000]]

Out[54]= 500500
```

Das ist die Summe der natürlichen Zahlen von 1 bis 1000.
▮ Voici la somme des nombres naturels de 1 à 1000.

8.4. Herauslesen von gewissen Teilen von Ausdrücken
■ Choisir certaines parties d'expressions

Allgemeines
■ Généralités

Studiere: ■ Etudier:

```
In[55]:= ?Position
```

```
Position[expr, pattern] gives a list of the positions at which objects
   matching pattern appear in expr. Position[expr, pattern, levspec] finds
   only objects that appear on levels specified by levspec. Position[expr,
   pattern, levspec, n] gives the positions of the first n objects found. Mehr...
```

```
In[56]:= ?[[
```

```
expr[[i]] or Part[expr, i] gives the ith part of expr. expr[[-i]]
   counts from the end. expr[[0]] gives the head of expr. expr[[i, j, ... ]]
   or Part[expr, i, j, ... ] is equivalent to expr[[i]] [[j]] ... . expr[[
   {i1, i2, ... } ]] gives a list of the parts i1, i2, ...  of expr. Mehr...
```

```
In[57]:= ?Part
```

```
expr[[i]] or Part[expr, i] gives the ith part of expr. expr[[-i]]
   counts from the end. expr[[0]] gives the head of expr. expr[[i, j, ... ]]
   or Part[expr, i, j, ... ] is equivalent to expr[[i]] [[j]] ... . expr[[
   {i1, i2, ... } ]] gives a list of the parts i1, i2, ...  of expr. Mehr...
```

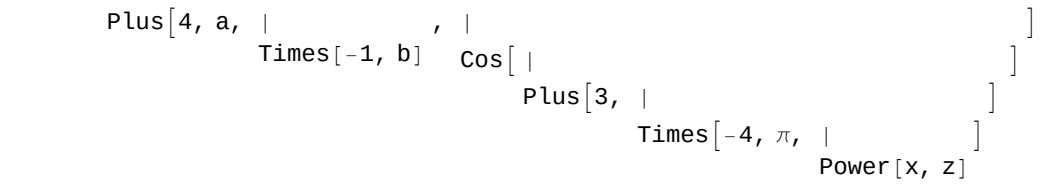
```
In[58]:= neueListe = 2 Range[9]

Out[58]= {2, 4, 6, 8, 10, 12, 14, 16, 18}

In[59]:= neueListe[[4]]

Out[59]= 8

In[60]:= Head[neueListe[[4]]]

Out[60]= Integer

In[61]:= HoldForm[neueListe[[4]]]

Out[61]= neueListe〚4〛
```

Bearbeite: ■ Travailler:

```
In[62]:= ausdruck = 4 + a - b + Cos[4 Pi x^z - 3]

Out[62]= 4 + a - b + Cos[3 - 4 π x^z]
```

In[63]:= **TreeForm[ausdruck]**

Out[63]//TreeForm=



In[64]:= **ausdruck[[3]]**

Out[64]= $-b$

In[65]:= **ausdruck[[4]]**

Out[65]= $\text{Cos}[3 - 4\,\pi\,x^z]$

In[66]:= **ausdruck[[4, 1]]**

Out[66]= $3 - 4\,\pi\,x^z$

In[67]:= **ausdruck[[4, 1, 2]]**

Out[67]= $-4\,\pi\,x^z$

Was ist passiert? --- Studiere weiter die Funktion "Position":
■ Que s'est-il passé? --- Continuer d'étudier la fonction "Position":

In[68]:= **ausdruck1 = Expand[(r + s + w)^3]**

Out[68]= $r^3 + 3\,r^2\,s + 3\,r\,s^2 + s^3 + 3\,r^2\,w + 6\,r\,s\,w + 3\,s^2\,w + 3\,r\,w^2 + 3\,s\,w^2 + w^3$

In[69]:= **Position[ausdruck1,r]**

Out[69]= {{1, 1}, {2, 2, 1}, {3, 2}, {5, 2, 1}, {6, 2}, {8, 2}}

In[70]:= **Position[ausdruck1,s]**

Out[70]= {{2, 3}, {3, 3, 1}, {4, 1}, {6, 3}, {7, 2, 1}, {9, 2}}

Deute diesen Output! --- Damit kann man auch komponieren:
Interpréter cet output! --- On peut aussi en faire des compositions:

In[71]:=
```
u = {ausdruck1[[2]], ausdruck1[[2,2]],
     ausdruck1[[2,2,1]], ausdruck1[[5,2]]}
```

Out[71]= $\{3\,r^2\,s,\ r^2,\ r,\ r^2\}$

In[72]:= **v = Apply[Plus, u]**

Out[72]= $r + 2\,r^2 + 3\,r^2\,s$

In[73]:= **v /. r->3**

Out[73]= $21 + 27\,s$

8.5. Symbole wie "unendlich" etc.
▮ Symboles tels que "infini" etc.

Allgemeines
▮ Généralités

Studiere: ▮ Etudier

In[74]:= **?Infinity**

Infinity is a symbol that represents a positive infinite quantity. Mehr...

In[75]:= **Head[Infinity]**

Out[75]= DirectedInfinity

In[76]:= **1 a**

Out[76]= a

In[77]:= **0 a**

Out[77]= 0

In[78]:= **0 5**

Out[78]= 0

In[79]:= **0 Infinity**

∞::indet : Indeterminate expression 0 ∞ encountered. Mehr…

Out[79]= Indeterminate

In[80]:= **?Limit**

Limit[expr, x->x0] finds the limiting value of expr when x approaches x0. Mehr...

In[81]:= **Limit[1/x, x->0]**

Out[81]= ∞

In[82]:= **Limit[Log[x], x->0]**

Out[82]= -∞

In[83]:= **Head[-Infinity]**

Out[83]= DirectedInfinity

In[84]:= **FullForm[-Infinity]**

Out[84]//FullForm=

DirectedInfinity[-1]

```
In[85]:= Limit[Log[1/x], x->0]
```

```
Out[85]= ∞
```

```
In[86]:= -5 Infinity
```

```
Out[86]= -∞
```

```
In[87]:= 7 / 0
```

```
Power::infy : Infinite expression 1/0 encountered. Mehr…
```

```
Out[87]= ComplexInfinity
```

```
In[88]:= ?ComplexInfinity
```

```
ComplexInfinity represents a quantity
   with infinite magnitude, but undetermined complex phase. Mehr…
```

```
In[89]:= 0 / 0
```

```
Power::infy : Infinite expression 1/0 encountered. Mehr…

∞::indet : Indeterminate expression 0 ComplexInfinity encountered. Mehr…
```

```
Out[89]= Indeterminate
```

```
In[90]:= ?Indeterminate
```

```
Indeterminate is a symbol that represents a
   numerical quantity whose magnitude cannot be determined. Mehr…
```

8.6. Memory: Wie *Mathematica* speichert
■ Memory: La façon de mémoriser de *Mathematica*

8.6.1. Allgemeines
■ Généralités

Studiere: ■ Etudier:

```
In[91]:= ?MemoryInUse
```

```
MemoryInUse[ ] gives the number of bytes currently
   being used to store all data in the current Mathematica session. Mehr…
```

Beispiele: ■ Exemples:

```
In[92]:= MemoryInUse[]
```

```
Out[92]= 3398800
```

"Putzmaschine" einsetzen:
■ Effacer:

```
In[93]:= Remove["Global`@*"]

In[94]:= MemoryInUse[]

Out[94]= 3398832

In[95]:= t = 1;

In[96]:= MemoryInUse[]

Out[96]= 3399616
```

Was hat t sowie die ausgeführten Operationen "gekostet"?
■ Qu'est-ce que t ainsi que les opérations exécutées ont-ils "coûté"?

```
In[97]:= vorher = MemoryInUse[]

Out[97]= 3400160

In[98]:= Table[Random[],{10000}];

In[99]:= nachher = MemoryInUse[]

Out[99]= 3482472

In[100]:=
       N[nachher - vorher] / 1000

Out[100]=
        82.312
```

Wieviel Speicher hat *Mathematica* "verbraten"?
■ Combien de mémoire *Mathematica* a-t-il "gaspillé"?

8.6.2. Gemeinsame Speichernutzung (sharing)
■ Utilisation commune de la mémoire (sharing)

Studiere: ■ Etudier:

```
In[101]:=
       (   Clear[a, b];
           vorher1 = MemoryInUse[];
           a = Table[Random[],{10000}];
           nachher1 = MemoryInUse[];
           b = a;
           nachher2 = MemoryInUse[];
           {nachher1 - vorher1, nachher2- nachher1}
       )

Out[101]=
        {80128, 16}
```

Beachte: b hat keinen Speicher mehr belegt. b belegt erst dann Speicher, wenn a manipuliert wird. Offenbar gibt es von a und b aus Zeiger auf diesselbe Speicheradresse. Das kann mit der Funktion "Share" auch künstlich bewirkt werden. Im folgenden Beispiel werden 1000 Pseudozufallszahlen im Intervall zwischen 10'000 und 10'100 gerechnet. Da dieses Intervall nur 100 Zahlen enthält, gibt es sicher Duplikate. Mit "Share" kann dabei eine Mehrfachspeicherung vermieden werden. Probiere:

■ Retenir: b n'a pas plus occupé de mémoire. b occupe la mémoire seulement si l'on manipule a. Il est évident qu'il a de a et de b des indicateurs pour la même adresse dans la mémoire. On peut obtenir cela aussi artificiellement par la fonction "Share". Dans l'exemple suivant on calcule 1'000 nombres pseudo-aléatoires dans l'intervalle entre 10'000 et 10'100. Comme cet intervalle ne compte que 100 nombres, il y aura certainement des doubles. Par "Share" on peut éviter une mise en mémoire répétitive. Essaie:

In[102]:=
```
?Share
```

```
Share[expr] changes the way expr is stored internally, to try and minimize the amount of
   memory used. Share[ ] tries to minimize the memory used to store all expressions. Mehr…
```

In[103]:=
```
(   Clear[c];
    vorherSpeicher = MemoryInUse[];
    c = Table[Random[Integer, {10000, 10100}],
                    {10000}];
    nachherSpeicher = MemoryInUse[];
    c = Share[c];
    nachherShare = MemoryInUse[];
    {nachherSpeicher - vorherSpeicher,
     nachherShare- vorherSpeicher}
)
```

Out[103]=
```
{40464, 1416}
```

Probiere: ■ Essaie:

In[104]:=
```
?ByteCount
```

```
ByteCount[expr] gives the number of bytes used internally by Mathematica to store expr.
  Mehr…
```

In[105]:=
```
t = 2
```

Out[105]=
```
2
```

In[106]:=
```
ByteCount[t]
```

Out[106]=
```
16
```

In[107]:=
```
ByteCount[neueVariable]
```

Out[107]=
```
0
```

In[108]:=
```
neueVariable = 1;
```

In[109]:=
```
ByteCount[neueVariable]
```

Out[109]=
```
16
```

In[110]:=
```
ByteCount[a]
```
Out[110]=
```
80056
```

In[111]:=
```
ByteCount[{a, b}]
```
Out[111]=
```
160144
```

Beispiele für Speicherbelegung:

■ Exemple pour l'occupation de mémoire:

In[112]:=
```
z = 15;  {Head[z], ByteCount[z]}
```
Out[112]=
```
{Integer, 16}
```

In[113]:=
```
z = 1;  {Head[z], ByteCount[z]}
```
Out[113]=
```
{Integer, 16}
```

In[114]:=
```
z = 100;  {Head[z], ByteCount[z]}
```
Out[114]=
```
{Integer, 16}
```

In[115]:=
```
z = 1000000000000000;  {Head[z], ByteCount[z]}
```
Out[115]=
```
{Integer, 32}
```

In[116]:=
```
z = 3.1;  {Head[z], ByteCount[z]}
```
Out[116]=
```
{Real, 16}
```

In[117]:=
```
z = 3.14159666666666;  {Head[z], ByteCount[z]}
```
Out[117]=
```
{Real, 16}
```

In[118]:=
```
z = 3/8;  {Head[z], ByteCount[z]}
```
Out[118]=
```
{Rational, 64}
```

In[119]:=
```
z = 3 + 5 I;  {Head[z], ByteCount[z]}
```
Out[119]=
```
{Complex, 64}
```

In[120]:=
```
z = Range[10];  {Head[z], ByteCount[z]}
```
Out[120]=
```
{List, 96}
```

In[121]:=
```
z = Range[10000];  {Head[z], ByteCount[z]}
```
Out[121]=
```
{List, 40056}
```

In[122]:=
```
z = "tschou";  {Head[z], ByteCount[z]}
```
Out[122]=
```
{String, 24}
```

In[123]:=
```
z = "t";  {Head[z], ByteCount[z]}
```
Out[123]=
```
{String, 24}
```

In[124]:=
```
z = "tschou zaeaeaeaeaeaemmaeaeaeaeaeaeaeae";
     {Head[z], ByteCount[z]}
```
Out[125]=
```
{String, 56}
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[126]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[127]:=
```
Remove["Global`*"]
```

Uebungen ▮ Exercices

8. Datentypen
▮ Types de données

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 8 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 8.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Untersuche die Interne Darstellung von:
▮ Examine la représentation intérieure de

a) 35 a

```
In[1]:= FullForm[35a]

Out[1]//FullForm=
        Times[35, a]
```

b) a 35

```
In[2]:= FullForm[a35]

Out[2]//FullForm=
        a35
```

c) - x

In[3]:= **FullForm[-x]**

Out[3]//FullForm=
```
Times[-1, x]
```

d) - 3

In[4]:= **FullForm[-3]**

Out[4]//FullForm=
```
-3
```

e) x - 3

In[5]:= **FullForm[x - 3]**

Out[5]//FullForm=
```
Plus[-3, x]
```

f) 3 - x

In[6]:= **FullForm[3 - x]**

Out[6]//FullForm=
```
Plus[3, Times[-1, x]]
```

g) Sqrt[5]

In[7]:= **FullForm[Sqrt[5]]**

Out[7]//FullForm=
```
Power[5, Rational[1, 2]]
```

h) Series[Tan[x],{x,0,9}]

In[8]:= **Series[Tan[x],{x,0,9}]**

Out[8]= $x + \frac{x^3}{3} + \frac{2\,x^5}{15} + \frac{17\,x^7}{315} + \frac{62\,x^9}{2835} + O[x]^{10}$

In[9]:= **FullForm[Series[Tan[x],{x,0,9}]]**

Out[9]//FullForm=
```
SeriesData[x, 0, List[1, 0, Rational[1, 3], 0,
  Rational[2, 15], 0, Rational[17, 315], 0, Rational[62, 2835]], 1, 10, 1]
```

In[10]:=

g) Normal[Series[Tan[x],{x,0,9}]]

In[10]:= **Normal[FullForm[Series[Tan[x],{x,0,9}]]]**

Out[10]//FullForm=
```
Plus[x, Times[Rational[1, 3], Power[x, 3]], Times[Rational[2, 15], Power[x, 5]],
 Times[Rational[17, 315], Power[x, 7]], Times[Rational[62, 2835], Power[x, 9]]]
```

Aufgabe 2 ▮ Problème 2

a) Vergleich von (x^y)^z mit x^(y^z)
▮ Comparaison de (x^y)^z avec x^(y^z)

x^y^z

```
In[11]:= 2^3^4
Out[11]= 241785163922925834941235 2
```

x^(y^z)

```
In[12]:= 2^(3^4)
Out[12]= 241785163922925834941235 2
```

(x^y)^z

```
In[13]:= (2^3)^4
Out[13]= 4096
```

b) Vergleich der FullForm
▮ Comparaison de FullForm

x^y^z

```
In[14]:= 2^3^4
Out[14]= 241785163922925834941235 2

In[15]:= FullForm[2^3^4]
Out[15]//FullForm=
        241785163922925834941235 2
```

x^(y^z)

```
In[16]:= 2^(3^4)
Out[16]= 241785163922925834941235 2

In[17]:= FullForm[2^(3^4)]
Out[17]//FullForm=
        241785163922925834941235 2
```

(x^y)^z

```
In[18]:= (2^3)^4

Out[18]= 4096

In[19]:= FullForm[(2^3)^4]

Out[19]//FullForm=
       4096
```

Aufgabe 3 ∎ Problème 3

Untersuchung der Memory-Belegung bei Operationen ∎ Examiner l'occupation de la mémoire lors d'opérations

Operation:
Liste von 1000 Integer-Zufallszahlen zwischen 100 und 200 rechnen:
∎ Opération:
Calculer une liste de 1000 nombres aléatoires "Integer" entre 100 et 200:

```
In[20]:= aufruf1 = MemoryInUse[];
         kleineInt = Table[Random[Integer,{100,200}],
                           {1000}];

In[22]:= aufruf2 = MemoryInUse[];
         aufruf2 - aufruf1

Out[23]= 6776
```

Operation:
Liste von 1'000 Integer-Zufallszahlen zwischen 300 und 400 rechnen:
∎ Opération:
Calculer une liste de 1'000 nombres aléatoires integer entre 300 et 400:

```
In[24]:= kleineInt = Table[Random[Integer,{300,400}],
                           {1000}];

In[25]:= aufruf3 = MemoryInUse[];
         aufruf3 - aufruf2

Out[26]= 5368
```

Wieviel Memory ist mehr benutzt worden?
(*Mathematica* ist so gebaut worden, weil eine Klasse mehr benutzt wird...)
∎ Combien de mémoire en plus a-t-on occupé?
(*Mathematica* a été construit ainsi, parce qu'on utilise une classe de plus...)

Operation:
Liste von 1'000 Real-Zufallszahlen (default-Präzision) rechnen:
∎ Opération:
Calculer une liste de 1'000 nombres aléatoires réels (précision default):

```
In[27]:= realInt = Table[Random[],{1000}];
```

In[28]:=
```
aufruf4 = MemoryInUse[];
aufruf4 - aufruf3
```

Out[29]= 9568

Aufgabe 4 ▮ Problème 4

Was ist die grösste speicherbare quadratische Matrix bei 4 Megabytes? ▮ Quelle est la plus grande matrice carrée qu'on peut mémoriser?

Annahme: Zur Speicherung einer Floating-point-Zahl braucht es 32 = 2^5 Bytes.

▮ Supposition: Pour mémoriser un nombre à virgule flottante on utilise 32 = 2^5 bytes.

In[30]:=
```
NSolve[2^x==10^6,x]
```

```
Solve::ifun : Inverse functions are being used by Solve, so some
   solutions may not be found; use Reduce for complete solution information. Mehr…
```

Out[30]= {{x → 19.9316}}

In[31]:=
```
?NSolve
```

```
NSolve[lhs==rhs, var] gives a list of numerical approximations
   to the roots of a polynomial equation. NSolve[{eqn1, eqn2, ... },
   {var1, var2, ... }] solves a system of polynomial equations. Mehr…
```

1 Mb entspricht etwa 2^20 Bytes.

▮ 1 Mb correspond à environ 2^20 bytes.

Anzahl Floating-point Zahlen in 4 Mb:

▮ Nombre (quantité) de nombres à virgule flottante dans 4 Mb:

In[32]:=
```
a = 4 2^19.9316/2^5
```

Out[32]= 125003.

Seitenlänge der quadratischen Matrix:

▮ Longueur d'un côté de la matrice carrée:

In[33]:=
```
s = Floor[Sqrt[a]]
```

Out[33]= 353

"Putzmaschine" einsetzen ▮ Employer la "machine de nettoyage"

In[34]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[35]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

9. Funktionen definieren
▮ Définir des fonctions

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 9 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 9.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

9.1. Einfache Funktionen
▮ Fonctions simples

9.1.1. Funktionen selber schreiben: Falls in *Mathematica* die gewollte Funktion nicht schon vorhanden ist, kann man versuchen, sie selber zu schreiben.
▮ Ecrire les fonctions soi-même: Si la fonction voulue n'est pas contenue dans *Mathematica*, on peut l'écrire soi-même.

Erst "Clear", damit keine Ueberraschung auftritt! Beispiel: Quadrieren...

▮ D'abord "Clear", pour ne pas avoir de susrprise! Exemple: Elever au carré...

```
In[1]:= Clear[quadrat];
        quadrat[x]:=x^2;
        quadrat[x]
```

Out[3]= x^2

```
In[4]:= {quadrat[y], quadrat[5], quadrat[12]}
```

```
Out[4]= {quadrat[y], quadrat[5], quadrat[12]}
```

Offenbar klappt es bei dieser Funktion nicht mit demVariablenwechsel resp. mit dem Einsetzen von Werten. Wieso?
■ Il est évident que le changement de variables ne fonctionne pas pour cette fonction, le placement de valeurs non plus. Pourquoi?

```
In[5]:= ??quadrat

        Global`quadrat

        quadrat[x] := x^2

In[6]:= ?_

        _ or Blank[ ] is a pattern object that can stand for any Mathematica
           expression. _h or Blank[h] can stand for any expression with head h. Mehr…
```

Die Variable x ist kein "Muster", das für irgend einen andern Ausdruck oder Wert stehen kann. Nehmen wir "x_" statt "x", so klappt es:
■ La variable x n'est pas un "patron" qui peut remplacer une expression ou une valeur quelconque. Si nous prenons "x_" au lieu de "x", cela fonctionne:

```
In[7]:= Clear[quadrat1];
        quadrat1[x_]:=x^2;
        quadrat1[x]

Out[9]= x^2

In[10]:= {quadrat1[y], quadrat1[5], quadrat1[12]}

Out[10]= {y^2, 25, 144}

In[11]:= ??quadrat1

         Global`quadrat1

         quadrat1[x_] := x^2
```

Oder: ■ Ou:

```
In[12]:= Table[quadrat1[n],{n,12}]

Out[12]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144}
```

Oder: ■ Ou:

```
In[13]:= quadrat1[9z^3+2z-1]

Out[13]= (-1 + 2 z + 9 z^3)^2

In[14]:= quadrat1[9z^3+2z-1] // Expand

Out[14]= 1 - 4 z + 4 z^2 - 18 z^3 + 36 z^4 + 81 z^6
```

Oder: ■ Ou:

9.1.2. Funktionen mit mehreren Variablen:
▮ Fonction avec plusieurs variables:

Ein Beispiel, dass alles sagt (Cosinus-Satz):
▮ Un exemple qui dit tout (théorème du cosinus):

```
In[15]:= Clear[c];
         c[a_, b_, gamma_] = Sqrt[a^2 + b^2
                               - 2 a b Cos[gamma]];
         c[u, v, w]
```

$$\text{Out[17]}= \sqrt{u^2 + v^2 - 2\,u\,v\,\text{Cos}[w]}$$

```
In[18]:= c[3, 5, Pi/3]
```

$$\text{Out[18]}= \sqrt{19}$$

9.1.3. Funktionen definiert in mehreren Schritten:
▮ Fonctions définies en plusieurs étapes:

Beispiel: Umrechnung in Polarkoordinaten:
▮ Exemple: Transformation en coordonnées polaires:

```
In[19]:= PolarTransformation[r_, phi_] := (
                     x = r Cos[phi];
                     y = r Sin[phi];
                     {x, y}               );
         PolarTransformation[a, b]
```

$$\text{Out[20]}= \{a\,\text{Cos}[b],\ a\,\text{Sin}[b]\}$$

Neudefinition: Die Umrechnung in Polarkoordinaten könnte auch für einen Vektor aufgerufen werden. Daher ist eine zweite Definition ratsam:
▮ Nouvelle définition: La transformation en coordonnées polaires pourrait être appelée aussi pour un vecteur. C'est pourquoi on conseille une deuxième définition:

```
In[21]:= PolarTransformation[{r_, phi_}] :=
                          PolarTransformation[r, phi];
         PolarTransformation[{a, b}]
```

$$\text{Out[22]}= \{a\,\text{Cos}[b],\ a\,\text{Sin}[b]\}$$

```
In[23]:= PolarTransformation[{3, 2}]
```

$$\text{Out[23]}= \{3\,\text{Cos}[2],\ 3\,\text{Sin}[2]\}$$

```
In[24]:= PolarTransformation[3, 2]
```

$$\text{Out[24]}= \{3\,\text{Cos}[2],\ 3\,\text{Sin}[2]\}$$

```
In[25]:= PolarTransformation[3, 2] // N
```

$$\text{Out[25]}= \{-1.24844,\ 2.72789\}$$

9.2. Typenprüfung
▮ Examen de types

9.2.1. f[arg_typ] ▮ f[arg_typ]

"f[arg_typ]" funktioniert nur, falls gilt "Head[arg] === typ". Probiere:

▮

"f[arg_typ]" fonctionne seulement s'il vaut "Head[arg] === typ". Essaie:

```
In[26]:= ?Binomial

         Binomial[n, m] gives the binomial coefficient. Mehr…

In[27]:= Clear[pascal];
         pascal[n_Integer]:=Table[Binomial[n,i],{i,0,n}];

In[29]:= pascal[0]

Out[29]= {1}

In[30]:= pascal[1]

Out[30]= {1, 1}

In[31]:= pascal[2]

Out[31]= {1, 2, 1}

In[32]:= pascal[3]

Out[32]= {1, 3, 3, 1}

In[33]:= pascal[4]

Out[33]= {1, 4, 6, 4, 1}

In[34]:= pascal[5]

Out[34]= {1, 5, 10, 10, 5, 1}

In[35]:= pascal[5.6]

Out[35]= pascal[5.6]

In[36]:= pascal[6/5]

Out[36]= pascal[6/5]
```

Ein anderes Beispiel (ausprobieren):

▮ Autre exemple (essayer):

```
In[37]:= Clear[mittelWert];
```

```
In[38]:= mittelWert[x_List] := Apply[Plus, x]/Length[x];
         mittelWert[{1,2,3,4,5}]

Out[39]= 3

In[40]:= mittelWert[1,2,3,4,5]

Out[40]= mittelWert[1, 2, 3, 4, 5]
```

Test, ob ein Ausdruck ein Polynom, eine Primzahl, ein Vektor u.s.w. ist, d.h. ob ein Ausdruck unter einer Prädikatenfunktion "wahr" ergibt. Beispiel:

■ Tester si une expression est un polynôme, un nombre premier, un vecteur etc., c'est-à-dire, si une expression est "vraie" sous une fonction de prédicats. Exemple:

```
In[41]:= ?*Q
```

System`

```
ArgumentCountQ       LinkConnectedQ  PartitionsQ
ArrayQ               LinkReadyQ      PolynomialQ
AtomQ                ListQ           PrimeQ
DigitQ               LowerCaseQ      SameQ
EllipticNomeQ        MachineNumberQ  StringFreeQ
EvenQ                MatchLocalNameQ StringMatchQ
ExactNumberQ         MatchQ          StringQ
FreeQ                MatrixQ         SyntaxQ
HypergeometricPFQ    MemberQ         TensorQ
InexactNumberQ       NameQ           TrueQ
IntegerQ             NumberQ         UnsameQ
IntervalMemberQ      NumericQ        UpperCaseQ
InverseEllipticNomeQ OddQ            ValueQ
LegendreQ            OptionQ         VectorQ
LetterQ              OrderedQ
```

```
In[42]:= {PrimeQ[45], PrimeQ[17]}

Out[42]= {False, True}
```

Anwendung auf die Definition einer Funktion:

■ Application à la définition d'une fonction:

```
In[43]:= Clear[halb];
         halb[x_?EvenQ] := x/2;
         {halb[1], halb[2], halb[3], halb[4], halb[5]}

Out[45]= {halb[1], 1, halb[3], 2, halb[5]}

In[46]:= {halb[2], halb[4], halb[6], halb[8], halb[9]}

Out[46]= {1, 2, 3, 4, halb[9]}
```

Was ist passiert?

■ Que s'est-il passé?

Ein Beispiel in mehreren Schritten:
■ Un exemple en plusieurs étapes:

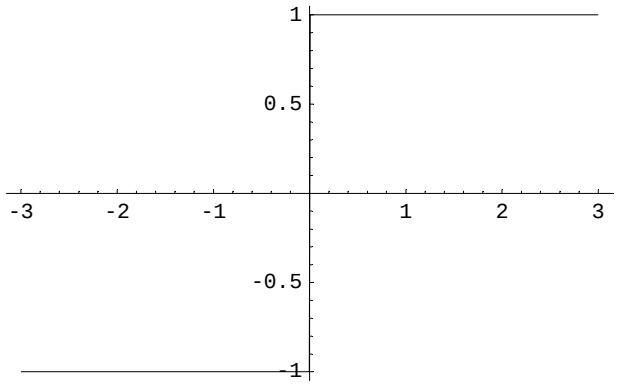
```
In[47]:= Remove[signum];
         signum[0                       ] :=  0;
         signum[x_/; Positive[x]==True ] :=  1;
         signum[x_/; Negative[x]==True ] := -1;

In[51]:= {signum[3], signum[-1], signum[0], signum[2]}

Out[51]= {1, -1, 0, 1}

In[52]:= ??Positive

         Positive[x] gives True if x is a positive number. Mehr…

         Attributes[Positive] = {Listable, Protected}

In[53]:= ??Integer

         Integer is the head used for integers. Mehr…

         Attributes[Integer] = {Protected}

In[54]:= Positive[1]

Out[54]= True

In[55]:= signum[1]

Out[55]= 1

In[56]:= ??signum

        Global`signum

         signum[0] := 0

         signum[x_ /; Positive[x] == True] := 1

         signum[x_ /; Negative[x] == True] := -1
```

9.3. Bedingte Ausführung
■ Exécution conditionnée

Betrachte das obige Beispiel mit "signum"!
■ Considère l'exemple ci-dessus avec "signum"!

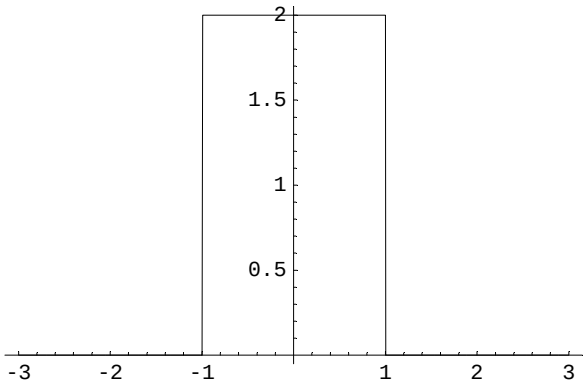
Bei der Definition von "signum" ist in der Klammer "[ ]" nach "x_" das Symbol "/;" angegeben. Darauf folgt jeweils eine Bedingung! Damit können Funktionen punktweise oder intervallweise definiert werden!!!!!! Anwendungen:
■ Dans la définition de "signum", le symbole "/;" est indiqué dans la parenthèse "[ ]" après "x_". Il suit chaque fois une condition! Ainsi on peut définir des fonctions ponctuellement ou par intervalles! Applications:

```
In[57]:= Plot[signum[x], {x, -3, 3}];
```



Oder: ■ Ou:

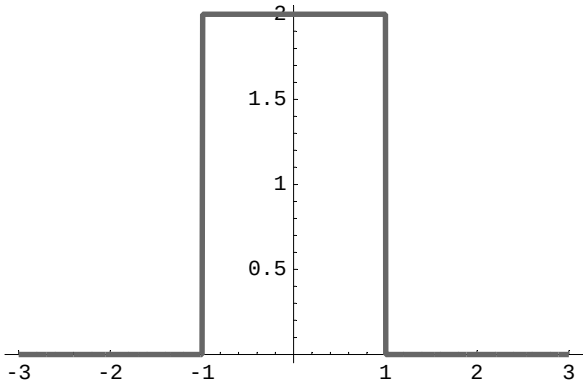
```
In[58]:= Clear[rechteck];
         rechteck[x_; (-1<=x && x<=1) ]:= 2;
         rechteck[x_; Abs[x] > 1      ]:= 0;
         Plot[rechteck[x], {x, -3, 3}];
```



Etwas "schöner":

■ Un peu plus esthétiquement:

```
In[62]:= Plot[rechteck[x], {x, -3, 3},
             PlotStyle -> {{Thickness[0.01],
                            GrayLevel[0.4]}}];
```



Solche Funktionen können sogar integriert werden:

■ On peut même intégrer de telles fonctions:

In[63]:= **??rechteck**

```
Global`rechteck

rechteck[x_ /; -1 ≤ x && x ≤ 1] := 2

rechteck[x_ /; Abs[x] > 1] := 0
```

In[64]:= **NIntegrate[rechteck[x], {x, -1, 1}]**

Out[64]= 4.

In[65]:= **NIntegrate[rechteck[x], {x, -3, 3}]**

```
NIntegrate::slwcon :
 Numerical integration converging too slowly; suspect one of the following: singularity, value of
   the integration being 0, oscillatory integrand, or insufficient WorkingPrecision. If your
   integrand is oscillatory try using the option Method->Oscillatory in NIntegrate. Mehr…

NIntegrate::ncvb : NIntegrate failed to converge to
   prescribed accuracy after 7 recursive bisections in x near x = -1.00781. Mehr…
```

Out[65]= 3.99241

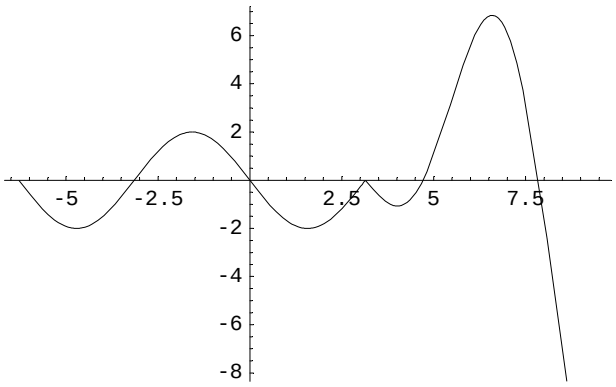
Was ist passiert? Studiere:

■ Que s'est-il passé? Etudie:

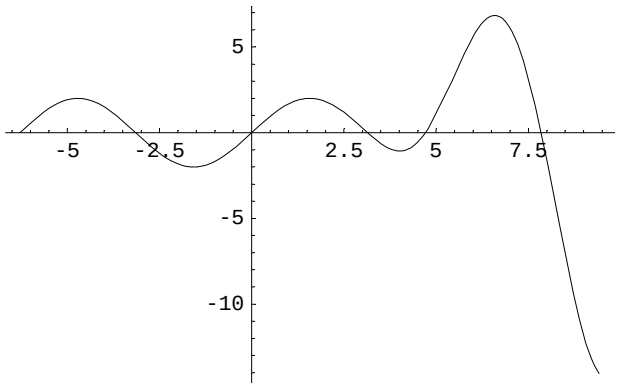
In[66]:= **?Which**

```
Which[test1, value1, test2, value2, ... ] evaluates each of the testi in turn, returning
   the value of the valuei corresponding to the first one that yields True. Mehr...
```

In[67]:=
```
Remove[funkt];
funkt[x_] := Which[x<=Pi,-2 Sin[x],x>Pi,
            (x^1.3-Pi^1.3) Cos[x]];
Plot[funkt[x],{x,-2Pi,3Pi}];
```



```
In[70]:= Remove[funkt];
         funkt[x_] := Which[x<=Pi,2 Sin[x],x>Pi,
                      (x^1.3-Pi^1.3) Cos[x]];
         Plot[funkt[x],{x,-2Pi,3Pi}];
```



9.4. Vorgegebene Werte
▮ Valeurs données

9.4.1. Das Muster "x_.":
▮ Le patron ("Pattern") "x_.":

Das Muster "x_." bewirkt, dass x "default" mit dem neutralen Element der am "stärksten" mit x verknüpften Operation belegt wird. Dazu ein Beispiel:

▮ Le patron "x_." fait que x "default" est occupé par l'élément neutre de l'opération qui est le plus "fortement" lié à x. Voici un exemple:

```
In[73]:= Clear[f];
         f[x_. + y_. z_.^e_.] := {x, y, z, e};
```

```
Pattern::nodef : No default setting found for Power in position 1 when length is 2. Mehr…
```

```
Pattern::nodef : No default setting found for Power in position 1 when length is 2. Mehr…
```

```
In[75]:= Clear[f];
         f[x_. + y_. z_^e_.] := {x, y, z, e};
```

```
In[77]:= ?f

         Global`f

         f[x_. + y_. z_^e_.] := {x, y, z, e}
```

```
In[78]:= f[4]

Out[78]= {0, 1, 4, 1}
```

```
In[79]:= {f[0], f[1], f[2], f[3], f[4], f[5]}

Out[79]= {{0, 1, 0, 1}, {0, 1, 1, 1}, {0, 1, 2, 1}, {0, 1, 3, 1}, {0, 1, 4, 1}, {0, 1, 5, 1}}
```

```
In[80]:= {x, y, z, e}

Out[80]= {3 Cos[2], 3 Sin[2], z, e}
```

Finde heraus, welche Werte "default" jeweils für x, y, z und e genommen worden sind und wieso!

- Cherche quelles valeurs "default" ont été prises respectivement pour x, y, z et e et pour quelle raison!

9.4.2. Das Muster "x_:" :
Le patron "x_:" :

Das Muster "x_:Wert" bewirkt, dass x "default" mit Wert belegt wird. Dazu ein Beispiel:

- Le patron "x_:Valeur" fait que x "default" est occupé par valeur. Voici un exemple:

```
In[81]:= Clear[UnserBereich];
         UnserBereich[start_:34, ziel_] :=
                              Range[start, ziel];
         UnserBereich[100]

Out[83]= {34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
           50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66,
           67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
           84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}

In[84]:= UnserBereich[80, 100]

Out[84]= {80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}

In[85]:= UnserBereich[36]

Out[85]= {34, 35, 36}

In[86]:= UnserBereich[34]

Out[86]= {34}

In[87]:= UnserBereich[30]

Out[87]= {}
```

9.5. Die Abarbeitungshierarchie bei Regeln

Mathematica bearbeitet die spezielleren Regeln vor den generelleren Regeln. Mit "?" kann die Abarbeitungshierarchie eingesehen werden.

- *Mathematica* travaille les règles spéciales avant les règles générales. Avec "?" on peut voir la hiérarchie de travail.

```
In[88]:= Clear[r];
         r[x_]       := x^2;
         r[x_Integer]:= x;
         r[5]        := 11;
         ?r

        Global`r

         r[5] := 11

         r[x_Integer] := x

         r[x_] := x^2
```

Beachte, dass die Reihenfolge der Ausgabe nicht mit der der Eingabe übereinstimmt. Wieso wohl?
Considère que l'ordre de la sortie ne correspond pas à celui de l'entrée. Pourquoi?

```
In[93]:= {r[5], r[5.5], r[6], r[r.5], r[7]}

Out[93]= {11, 30.25, 6, 0.25 r^2, 7}
```

9.6. Zusammenfügen von Regeln mit "f/: "
▮ Assemblage de règles par "f/: "

Operationen links vom Zeichen ":= "
▮ Opérations à gauche du signe ":= "

Z.B. f[x_] + g[x_] := h[x_] "funktioniert" nicht. Beispiel:
▮ P.ex. f[x_] + g[x_] := h[x_] ne "fonctionne" pas. Exemple:

```
In[94]:= f[x_] + g[x_] := h[x_]

         SetDelayed::write : Tag Plus in g[x_] + {0, 1, x_, 1} is Protected. Mehr…

Out[94]= $Failed
```

Wie lässt sich das Funktionieren trotzdem erzwingen? -
Dafür ist das Symbol "f/ : " vorgesehen. Studiere das folgende Beispiel:
▮ Comment obtenir quand même le fonctionnement? -
Pour cela est prévu le symbole "f/ : ". Etudie l'exemple suivant:

```
In[95]:= Remove[f];
         f/: f[x_] + g[x_] := h[x_]

         RuleDelayed::rhs : Pattern x_ appears on the right-hand side of rule f[x_] + g[x_] :> h[x_]. Mehr…

In[97]:= ?f

        Global`f

         f /: f[x_] + g[x_] := h[x_]

In[98]:= f[5] + g[5]

Out[98]= h[Pattern[5, _]]
```

f wird hier assoziiert mit f[x]+g[x] := h[x]. Wird f[x]+g[x] gerufen, so erscheint h[x]. Andernfalls erscheint f[x] oder f[x1]+g[x2].
■ Ici on associe f avec f[x]+g[x] := h[x]. Si on appelle f[x]+g[x], il arrive h[x]. Autrement il apparaît f[x] ou f[x1]+g[x2].

In[99]:=
```
f[5]
```
Out[99]=
```
f[5]
```

In[100]:=
```
f[5] + g[6]
```
Out[100]=
```
f[5] + g[6]
```

Beispiel: ■ Exemple:

In[101]:=
```
Remove[f];
f/: f[x_] + Sin[x_] := x^3;
f[3] + Sin[3]
```
Out[103]=
```
27
```

Vergleiche: ■ Compare:

In[104]:=
```
??_
```
```
_ or Blank[ ] is a pattern object that can stand for any Mathematica
   expression. _h or Blank[h] can stand for any expression with head h. Mehr…

Attributes[Blank] = {Protected}
```

In[105]:=
```
?_.
```
```
p:v is a pattern object which represents an expression
   of the form p, which, if omitted, should be replaced by v. Mehr…
```

In[106]:=
```
?:
```
```
Information::notfound : Symbol : not found. Mehr…
```

In[107]:=
```
?/;
```
```
patt /; test is a pattern which matches only if the evaluation of test yields True. lhs :>
   rhs /; test represents a rule which applies only if the evaluation of test yields
   True. lhs := rhs /; test is a definition to be used only if test yields True. Mehr…
```

In[108]:=
```
???
```
```
p?test is a pattern object that stands for any expression
   which matches p, and on which the application of test gives True. Mehr…

Attributes[PatternTest] = {HoldRest, Protected}
```

9.7. Dokumentieren der eigenen Funktionen
▮ Documenter les propres fonctions

9.7.1. Verschiedenes Verhalten einer Transformationsregel:
▮ Comportement différent d'une règle de transformation

Beachte, dass die Regel s[-x_] :> -s[x] den Ausdruck s[-a] in -s[a] transformiert, s[-3] aber nicht transformiert. Was wird mit s[1-x] geschehen und wieso?

▮ Considère que la règle s[-x_] :> -s[x] transforme l'expression s[-a] en -s[a], mais ne transforme pas s[-3]. Que se passe-t-il avec s[1-x] et pourquoi?

```
In[109]:=
sRegel = s[-x_] :> -s[x]
```

```
Out[109]=
s[-x_] :> -s[x]
```

```
In[110]:=
s[-a] /. sRegel
```

```
Out[110]=
-s[a]
```

```
In[111]:=
s[-3] /. sRegel
```

```
Out[111]=
s[-3]
```

Wieso geschieht das? Beobachte, wie *Mathematica* -a und -3 interpretiert:

▮ Pourquoi cela arrive-t-il? Observe comment *Mathematica* interprète -a et -3:

```
In[112]:=
FullForm[-a]
```

```
Out[112]//FullForm=
Times[-1, a]
```

```
In[113]:=
FullForm[-3]
```

```
Out[113]//FullForm=
-3
```

Der Ausdruck -a ist mit dem Muster -(x_) verträglich, der Ausdruck -3 nicht.

▮ L'expression -a est compatible avec le patron -(x_), l'expression -3 ne l'est pas.

In[114]:=
```
FullForm[1-x]
```

Out[114]//FullForm=
```
Plus[1, Times[-3, Cos[2]]]
```

Der Ausdruck (1-x) ist mit dem Muster -(x_) nicht verträglich:
■ L'expression (1-x) n'est pas compatible avec le patron -(x_).

In[115]:=
```
s[1-x] /. sRegel
```

Out[115]=
```
s[1 - 3 Cos[2]]
```

9.7.2. Der Ausdruck ' Funktion ::usage = "Erklärung" ' ■ L'expression ' Funktion ::usage = "explication" '

Bisher ist öfters schon das Symbol "?" erschienen. Damit ist einige Information abrufbar. Beispiel:
■ Jusqu'à présent le symbole "?" est apparu plusieurs fois. Par cela on peut appeler bien de l'information. Exemple:

In[116]:=
```
?f
```
```
Global`f
```
```
f /: f[x_] + Sin[x_] := x^3
```

In[117]:=
```
?Sin
```
```
Sin[z] gives the sine of z. Mehr…
```

In[118]:=
```
??Sin
```
```
Sin[z] gives the sine of z. Mehr…
```
```
Attributes[Sin] = {Listable, NumericFunction, Protected}
```

Manchmal möchte man aber gern dem System eigene Kommentare einverleiben, die dann wieder abgerufen werden können. Das ist möglicn mit "...::usage=...".
Beispiel: Wir verwenden die oben definierte Funktion "pascal".
■ Parfois on aimerait incorporer au système ses propres commentaires, qu'on pourrait de nouveau appeler. Cela est possible avec "...::usage=...".
Exemple: Nous employons la fonction "pascal" définie ci-dessus.

In[119]:=
```
Clear[pascal];
pascal[n_Integer]:=Table[Binomial[n,i],{i,0,n}];
```

In[121]:=
```
?pascal
```
```
Global`pascal
```
```
pascal[n_Integer] := Table[Binomial[n, i], {i, 0, n}]
```

Wir fügen nun an:
■ Nous ajoutons:

In[122]:=
```
pascal::usage = "pascal[n] gibt die n-te Zeile
des Pascalschen Dreiecks aus - sort la ligne n
du triangle de Pascal."
```

Out[122]=
```
pascal[n] gibt die n-te Zeile  des Pascalschen
  Dreiecks aus - sort la ligne n du triangle de Pascal.
```

Abfragen: ■ Appeler:

In[123]:=
```
?pascal
```

```
pascal[n] gibt die n-te Zeile  des
  Pascalschen Dreiecks aus - sort la ligne n du triangle de Pascal.
```

In[124]:=
```
??pascal
```

```
pascal[n] gibt die n-te Zeile  des
  Pascalschen Dreiecks aus - sort la ligne n du triangle de Pascal.

pascal[n_Integer] := Table[Binomial[n, i], {i, 0, n}]
```

Gute Sache, nicht?
■ C'est bon, n'est-ce pas?

9.8. Attribute
■ Attributs

Spezielle Funktionen bestimmen:
■ Déterminer des fonctions spéciales:

Funktionen haben Attribute. Beispiel "Sinus":
■ Les fonctions ont des attributs. Exemple "Sinus":

In[125]:=
```
Attributes[Sin]
```

Out[125]=
```
{Listable, NumericFunction, Protected}
```

In[126]:=
```
??Attributes
```

```
Attributes[symbol] gives the list of attributes for a symbol. Mehr...

Attributes[Attributes] = {HoldAll, Listable, Protected}
```

In[127]:=

```
??Attributes
```

```
Attributes[symbol] gives the list of attributes for a symbol. Mehr…

Attributes[Attributes] = {HoldAll, Listable, Protected}
```

Die möglichen Attribute sind vom System vorgegeben. Einige Beispiele:

Les attributs possibles sont donnés par le système. Quelques exemples:

In[128]:=

```
??Constant
```

```
Constant is an attribute which indicates
   zero derivative of a symbol with respect to all parameters. Mehr…

Attributes[Constant] = {Protected}
```

In[129]:=

```
?Flat
```

```
Flat is an attribute that can be assigned to a symbol f to
   indicate that all expressions involving nested functions f should be
   flattened out. This property is accounted for in pattern matching. Mehr…
```

In[130]:=

```
?Hold*
```

System`

```
Hold    HoldAllComplete HoldFirst HoldPattern
HoldAll HoldComplete    HoldForm  HoldRest
```

In[131]:=

```
?HoldAll
```

```
HoldAll is an attribute which specifies that all arguments
   to a function are to be maintained in an unevaluated form. Mehr…
```

In[132]:=

```
?HoldFirst
```

```
HoldFirst is an attribute which specifies that the first
   argument to a function is to be maintained in an unevaluated form. Mehr…
```

In[133]:=

```
?HoldRest
```

```
HoldRest is an attribute which specifies that all but the first
   argument to a function are to be maintained in an unevaluated form. Mehr…
```

In[134]:=

```
?Listable
```

```
Listable is an attribute that can be assigned to a symbol f to indicate that the function f
   should automatically be threaded over lists that appear as its arguments. Mehr…
```

In[135]:=

```
?Locked
```

```
Locked is an attribute which, once assigned,
   prevents modification of any attributes of a symbol. Mehr…
```

In[136]:=
```
?Orderless
```

```
Orderless is an attribute that can be assigned to a symbol f to indicate that the elements
   ei in expressions of the form f[e1, e2, ... ] should automatically be sorted
   into canonical order. This property is accounted for in pattern matching. Mehr…
```

In[137]:=
```
?Protected
```

```
Protected is an attribute which prevents
   any values associated with a symbol from being modified. Mehr…
```

In[138]:=
```
?ReadProtected
```

```
ReadProtected is an attribute which prevents values associated with a symbol from being seen.
 Mehr…
```

9.9. Die Art der Abarbeitung eines Ausdrucks
▮ La façon de travailler une expression

9.9.1. Normalfall
▮ Cas normal

Folgende Reihenfolge ist im System voreingestellt: Zuerst Kopf des Ausdrucks abarbeiten. Dann Elemente (Teile) des Ausdrucks abarbeiten.
Wenn ein Element kein Atomarer Ausdruck ist: Teile des Elements abarbeiten etc.. Dieses Procedere wird rekursiv angewandt.
▮ L'ordre suivant est déjà réglé dans le système: Travailler d'abord l'en-tête de l'expression. Travailler ensuite les éléments (parties) de l'expression. Si un élément n'est pas une expression atomique: Procéder par parties etc.. Ce procédé est appliqué récursivement.

9.9.2. Ausnahmefall
▮ Exception

Wenn ein Element oder Ausdruck das Attribut HoldFirst, HoldAll, HoldRest hat oder die Funktion Evaluate aufruft, wird die Voreinstellung durchbrochen. Erkunde, wo solche Attribute vorkommen:
▮ Quand un élément en une expression a l'attribut HoldFirst, HoldAll, HoldRest ou appellent la fonction Evaluate, le règlement préalable est interrompu. Cherche où se trouvent de tels attributs:

In[139]:=
```
??Sin
```

```
Sin[z] gives the sine of z. Mehr…

Attributes[Sin] = {Listable, NumericFunction, Protected}
```

In[140]:=
```
??Plot
```

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
   xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. Mehr...

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic,
  Compiled → True, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
  DisplayFunction :→ $DisplayFunction, Epilog → {}, FormatType :→ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
  ImageSize → Automatic, MaxBend → 10., PlotDivision → 30., PlotLabel → None,
  PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[141]:=
```
??Table
```

```
Table[expr, {imax}] generates a list of imax copies of expr. Table[
   expr, {i, imax}] generates a list of the values of expr when i runs from
   1 to imax. Table[expr, {i, imin, imax}] starts with i = imin. Table[expr,
   {i, imin, imax, di}] uses steps di. Table[expr, {i, imin, imax}, {j, jmin,
   jmax}, ... ] gives a nested list. The list associated with i is outermost. Mehr...

Attributes[Table] = {HoldAll, Protected}
```

In[142]:=
```
??Attributes
```

```
Attributes[symbol] gives the list of attributes for a symbol. Mehr...

Attributes[Attributes] = {HoldAll, Listable, Protected}
```

In[143]:=
```
??Evaluate
```

```
Evaluate[expr] causes expr to be evaluated even if it appears as the argument of
   a function whose attributes specify that it should be held unevaluated. Mehr...

Attributes[Evaluate] = {Protected}
```

Vergleiche die Resultate bei Abänderung der Abarbeitungsreihenfolge und erkläre den Unterschied der Resultate:

■ Compare les résultats lors des changements de l'ordre de travail et explique la différence des résultats:

In[144]:=
```
Table[Random[], {5}]
```

Out[144]=
```
{0.258329, 0.518034, 0.790079, 0.821881, 0.860525}
```

In[145]:=
```
Table[Evaluate[Random[]], {5}]
```

Out[145]=
```
{0.581881, 0.581881, 0.581881, 0.581881, 0.581881}
```

Wieso kommt das so, wie es kommt?

■ Pourquoi cela se déroule ainsi?

9.10. Diskrete Funktionen
▮ Fonctions discrètes

Ein Beispiel:

▮ Un exemple:

In[146]:=

```
Clear[ungerade, t];
ungerade[x_?OddQ ] := (ungerade[x] = 1);
ungerade[x_?EvenQ] := (ungerade[x] = -1);
```

In[149]:=

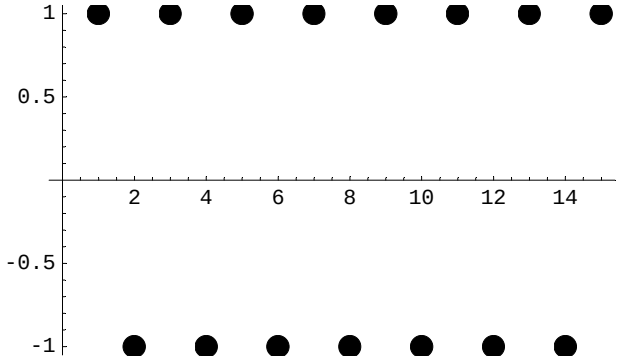
```
t = Table[ungerade[x], {x,1, 15}]
```

Out[149]=

```
{1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1}
```

In[150]:=

```
ListPlot[t, PlotStyle -> PointSize[0.04]];
```



In[151]:=

```
?ListPlot
```

```
ListPlot[{y1, y2, ... }] plots points corresponding to a list of values. The x
   coordinates are by default taken to be 1, 2, ... . ListPlot[{{x1, y1}, {x2,
   y2}, ... }] plots a list of points with specified x and y coordinates. Mehr...
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[152]:=

```
(* Old Form: Remove["Global`@*"] *)
```

In[153]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

9. Funktionen definieren
▮ Définir des fonctions

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 9 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 9.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Untersuche, was die unten definierte Funktion f macht:
▮ Examine ce que fait la fonction f définie ci-dessous:

Definition: ▮ Définition:

```
In[1]:= f[2]  = 3    ;
        f[u] := 2 u^2;
        f[v_]:= 1/v  ;
        u    = 2;
```

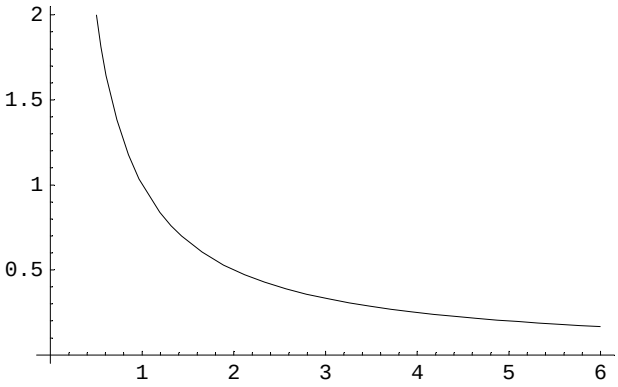
Was ist f[u] ?
▮ Qu'est-ce f[u] ?

```
In[5]:= f[u]
```

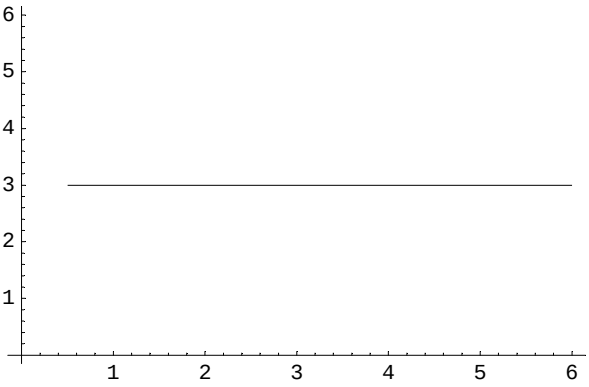
```
Out[5]= 3
```

Was ist mit f passiert? Untersuche die möglichen Graphen!
Que s'est-il passé? Examine les graphes possibles!

In[6]:= **Plot[f[x],{x,0.5,6}];**



In[7]:= **Plot[f[u],{x,0.5,6}];**



In[8]:= **f[2]**

Out[8]= 3

Was ist passiert?
■ Que s'est-il passé?

Aufgabe 2 ■ Problème 2

Definiere eine Funktion mitte[x], die das Integer- Resultat ausgibt für ungerade Werte von x!
■ Définis une fonction mitte[x], qui sort le résultat Integer pour les valeurs impairs de x!

Definition: ■ Définition:

In[9]:=
```
Clear[f];
f[x_?OddQ] := (x+1)/2
```

Ausprobieren: ■ Essayer:

In[11]:= **f[3.2]**

Out[11]= f[3.2]

```
In[12]:= f[3]

Out[12]= 2

In[13]:= f[2]

Out[13]= f[2]
```

Was ist passiert? ▮ Que s'est-il passé?

Aufgabe 3 ▮ Problème 3

Definiere eine Funktion, die als Argument ein Zahlenpaar {a, b} hat und als Wert das Paar {a, 2b} ausgibt. Wende die Funktion an auf die Liste {{1, 1}, {2, 2}}:
▮ Définis une fonction, qui a pour argument une paire de nombres {a, b} et qui sort comme valeur la paire {a, 2b}. Applique la fonction à la liste {{1, 1}, {2, 2}}:

Wende mit "Map" die Funktion an auf die Liste {{1, 1}, {2, 2}}.
▮ Applique par "Map" la fonction à la liste {{1, 1}, {2, 2}}.

```
In[14]:= Clear[verdoppelnY];
         verdoppelnY[{x_, y_}]:={x, 2y}

In[16]:= Map[verdoppelnY, {{1, 1},{2, 2}}]

Out[16]= {{1, 2}, {2, 4}}
```

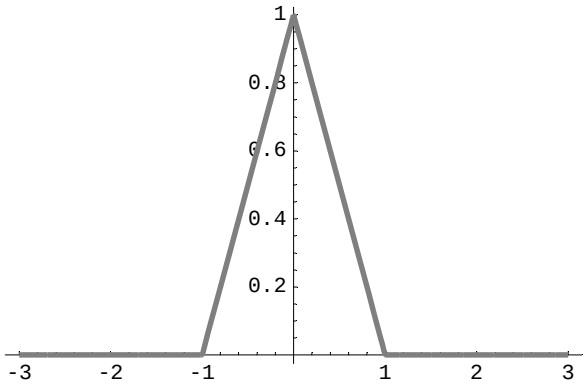
Was macht "Map" hier?
▮ Que fait "Map" ici?

Aufgabe 4 ▮ Problème 4

Definiere eine Funktion "dreieck[x]" wie folgt:
▮ Définis une fonction "dreieck[x]"

dreieck[x] ist gleich 1 - Abs[x] für x<1. Sonst ist der Funktionswert null.
Mache einen Plot!
▮ dreieck[x] est égal 1 - Abs[x] pour x<1. Autrement la valeur de la fonction est zéro. Fais un plot!

In[17]:=
```
Clear[dreieck];
dreieck[x_/; (-1<=x && x<=1)]:= 1 - Abs[x];
dreieck[x_/; Abs[x]>1]:= 0;
Plot[dreieck[x],{x,-3,3},PlotStyle -> {{
    Thickness[0.01],GrayLevel[0.5]}}];
```



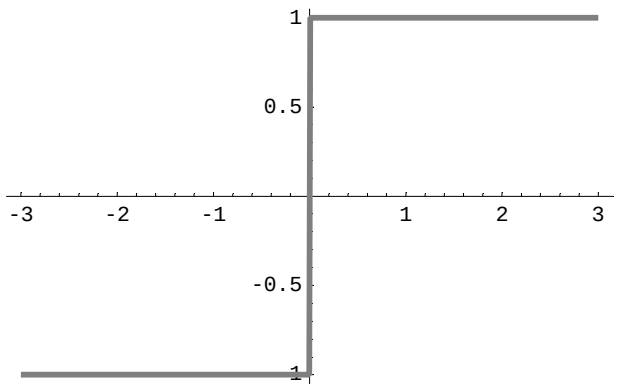
Aufgabe 5 ▮ Problème 5

Definiere eine Funktion "signum[x]" wie folgt:
▮ Définis une fonction "signum[x]" comme il suit:

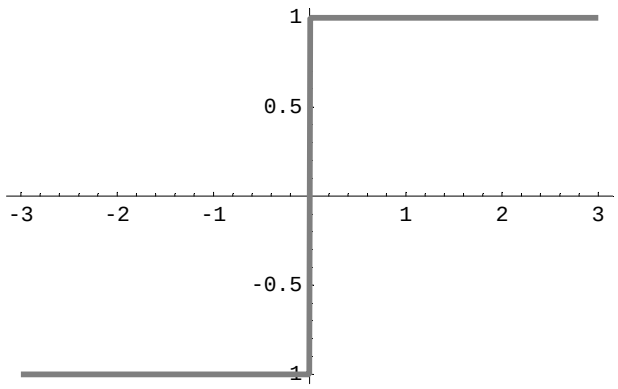
signum[x] ist gleich 1 für x>0, -1 für x<0 und 0 für x=0. Mache einen Plot! (Definiere dazu signum für Floating-Point-Zahlen 0.0!) Schreibe einen Kommentar für die Dokumentation!

▮ signum[x] est égal 1 pour x>0, -1 pour x<0 et 0 pour x=0. Fais un plot. (Définis signum pour les nombres à virgule flottante 0.0!) Ecris un commentaire pour la documentation!

In[21]:=
```
Clear[signum];
signum[0] = 0        ;
signum[x_/; x>0]:= 1 ;
signum[x_/; x<0]:= -1;
signum::usage="signum[x] ist gleich 1 für
positive x, -1 für negative x und 0 für
x = 0 - est égal 1 pour x positif, -1 pour
x négatif et 0 pour x = 0.";
Plot[signum[x],{x,-3,3},PlotStyle -> {{
    Thickness[0.01],GrayLevel[0.5]}}];
```



```
In[27]:= Clear[signum];
         signum[0.] = 0.      ;   (*Geändert!*)
         signum[x_/; x>0]:= 1 ;
         signum[x_/; x<0]:= -1;
         signum::usage="signum[x] ist gleich 1 für
         positive x, -1 für negative x und 0 für
         x = 0 - est égal 1 pour x positif, -1 pour
         x négatif et 0 pour x = 0.";
         Plot[signum[x],{x,-3,3},PlotStyle -> {{
             Thickness[0.01],GrayLevel[0.5]}}];
```




```
In[33]:= ?signum

         signum[x] ist gleich 1 für  positive x, -1 für negative x und 0
           für  x = 0 - est égal 1 pour x positif, -1 pour x négatif et 0 pour x = 0.

In[34]:= ??signum

         signum[x] ist gleich 1 für  positive x, -1 für negative x und 0
           für  x = 0 - est égal 1 pour x positif, -1 pour x négatif et 0 pour x = 0.

         signum[0.] = 0.

         signum[x_/; x > 0] := 1

         signum[x_/; x < 0] := -1
```

Aufgabe 6 ▮ Problème 6

Definiere die Funktion "myRange" wie folgt:
▮ Définis la fonction "myRange" comme il suit:

myRange[x] gibt für ein positives Integer-Argument n die Zahlen von 0 bis n aus. Für zwei Argumente m und n mit "n < m" gibt die Funktion alle natürlichen Zahlen von n bis m aus. Für drei Argumente n, m und d gibt sie die Zahlen n, n+d, n+2d, ... , n+kd aus mit n+kd<m.

▮ myRange[x] sort pour un argument Integer positif n les nombres de 0 à n. Pour deux arguments m et n avec "n < m", la fonction sort tous les nombres naturels de n à m. Pour trois arguments n, m et d, elle sort les nombres n, n+d, n+2d, ... , n+kd avec n+kd<m.

```
In[35]:= Clear[myRange];
         myRange[n_?Integer,m_?Integer,d_?Integer/;
         {n>=0 && m>=n && d>=0}]:= Range[n,m,d];
         myRange[n_?Integer,m_?Integer /;
         {n>=0 && m>=n}]:= Range[n,m];
         myRange[start_:0, n_?Integer /; n>=0]:=
         Range[start,n];
         Print[{6,Range[6]}];
         Print[{-6,Range[-6]}];
         Print[{0,Range[0]}];
         Print[{{5,11},Range[{5,11}]}];
         Print[{5,11,Range[5,11]}];
         Print[{5,-3,Range[5,-3]}];
         Print[{5,3,Range[5,3]}];
         Print[{5,18,4,Range[5,18,4]}];
```

```
{6, {1, 2, 3, 4, 5, 6}}

{-6, {}}

{0, {}}

{{5, 11}, {{1, 2, 3, 4, 5}, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}}}

{5, 11, {5, 6, 7, 8, 9, 10, 11}}

{5, -3, {}}

{5, 3, {}}

{5, 18, 4, {5, 9, 13, 17}}
```

Aufgabe 7 ▮ Problème 7

Verschiedenes Verhalten einer Transformationsregel: ▮ Comportement différent d'une règle de transformation:

Beachte, dass die Regel "s[-x_] :> -s[x]" den Ausdruck "s[-a]" in "-s[a]" transformiert, "s[-3]" aber nicht transformiert. Was wird mit "s[1-x]" geschehen und wieso?

▮ Considère que la règle "s[-x_] :> -s[x]" transforme l'expression "s[-a]" en "-s[a]", mais ne transforme pas "s[-3]". Que se passe-t-il avec "s[1-x]" et pourquoi?

```
In[47]:= sRegel = s[-x_] :> -s[x]
```

```
Out[47]= s[-x_] :> -s[x]
```

```
In[48]:= s[-a] /. sRegel
```

```
Out[48]= -s[a]
```

```
In[49]:= s[-3] /. sRegel
```

```
Out[49]= s[-3]
```

Wieso geschieht das? Beobachte, wie *Mathematica* -a und -3 interpretiert:

▮ Pourquoi cela arrive-t-il? Observe, comme *Mathematica* interprète -a et -3:

```
In[50]:= FullForm[-a]

Out[50]//FullForm=
        Times[-1, a]

In[51]:= FullForm[-3]

Out[51]//FullForm=
        -3
```

Der Ausdruck -a ist mit dem Muster -(x_) verträglich, der Ausdruck -3 nicht.
■ L'expression -a est compatible avec le patron -(x_) , l'expression -3 ne l'est pas.

```
In[52]:= FullForm[1-x]

Out[52]//FullForm=
        Plus[1, Times[-1, x]]
```

Der Ausdruck (1-x) ist mit dem Muster -(x_) nicht verträglich:
■ L'expression (1-x) n'est pas compatible avec le patron -(x_):

```
In[53]:= s[1-x] /. sRegel

Out[53]= s[1 - x]
```

Aufgabe 8 ■ Problème 8

Approximative Berechnung von Pi/4 durch eine rekursive Funktion: ■ Calcul approximatif de Pi/4 par une fonction récursive:

Definiere die rekursive Funktion piDurchVier[n_] um approximativ Pi/4 zu berechnen wie folgt:
■ Définis la fonction récursive piDurchVier[n_] pour calculer approximativement Pi/4 comme il suit:

Pi/4 ist etwa: ■ Pi/4 est environ:

2*4*4*6*6*8*8*10*10...2n / (3*3*5*5*7*7*9*9*...(2n+1)).

Entwickle: ■ Développe:

piDurchVier[1] = 2/3
piDurchVier[2] = 2*4*4/(3*3*5) = piDurchVier[1]*4*4/(3*5)
piDurchVier[3] = 2*4*4*6*6/(3*3*5*5*7)
= piDurchVier[2]*6*6/(5*7)
piDurchVier[4] = piDurchVier[3]*8*8/(7*9)

```
In[54]:= RemoveAll[piDurchVier];
         piDurchVier[1] = 2/3;
         piDurchVier[n_] :=
             piDurchVier[n-1] (2n)^2 / ((2n-1)(2n+1));
         ??piDurchVier

        Global`piDurchVier

        piDurchVier[1] = 2/3

        piDurchVier[n_] := piDurchVier[n-1] (2 n)^2 / ((2 n-1) (2 n+1))
```

```
In[58]:= Table[piDurchVier[i],{i,5}]

Out[58]= {2/3, 32/45, 128/175, 8192/11025, 32768/43659}
```

```
In[59]:= Table[piDurchVier[i],{i,5}] // N

Out[59]= {0.666667, 0.711111, 0.731429, 0.743039, 0.750544}
```

Kontrolle: ■ Contrôle:

```
In[60]:= N[Pi/4,10]

Out[60]= 0.7853981634
```

```
In[61]:= N[piDurchVier[200]]

Out[61]= 0.784419
```

Was hälst Du von der Sache?
■ Qu'en penses-tu?

Aufgabe 9 ■ Problème 9

Spezielle Funktionen bestimmen:
■ Déterminer des fonctions spéciales:

Bestimme die Funktionen, deren Namen mit den Buchstaben A - E beginnen und das Attribut "Listable" haben.
Hinweis: Was tun die Funktionen "Names", "MemberQ", "Attributes" und "Select"?
■ Détermine les fonctions dont le nom commence par les lettres A - E et qui ont l'attribut "Listable".
Indication: Que font les fonctions "Names", "MembreQ", "Attributes" et "Select"?

```
In[62]:= ??Names

        Names["string"] gives a list of the names of symbols which match the string. Names["string",
           SpellingCorrection->True] includes names which match after spelling correction. Mehr...

        Attributes[Names] = {Protected}

        Options[Names] = {IgnoreCase → False, SpellingCorrection → False}
```

In[63]:= **??MemberQ**

```
MemberQ[list, form] returns True if an element of list matches form, and False otherwise.
   MemberQ[list, form, levelspec] tests all parts of list specified by levelspec. Mehr...

Attributes[MemberQ] = {Protected}

Options[MemberQ] = {Heads → False}
```

In[64]:= **??Attributes**

```
Attributes[symbol] gives the list of attributes for a symbol. Mehr...

Attributes[Attributes] = {HoldAll, Listable, Protected}
```

In[65]:= **??Select**

```
Select[list, crit] picks out all elements ei of list for which crit[ei] is True. Select[
   list, crit, n] picks out the first n elements for which crit[ei] is True. Mehr...

Attributes[Select] = {Protected}
```

Suche alle "Listablen" Funktionen:

- Cherche toutes les fonctions "Listables":

In[66]:=
```
RemoveAll[listbareQ];
listbareQ[symbol_String]:=
    MemberQ[Attributes[symbol], Listable]
```

Was tun die Teile?

- Qu'en font les parties?

In[68]:= **Attributes["Cos"]**

Out[68]= {Listable, NumericFunction, Protected}

In[69]:= **MemberQ[Attributes["Cos"], Listable]**

Out[69]= True

In[70]:= **listbareQ["Cos"]**

Out[70]= True

Versuchen wir, die Funktionen zu bestimmen:

- Essayons de déterminer les fonctions:

In[71]:= **Select[Names["A* B* C* D* E*"], listbareQ]**

Out[71]= {}

In[72]:= **Select[Names["A*"], listbareQ]**

Out[72]=
```
{Abs, AiryAi, AiryAiPrime, AiryBi, AiryBiPrime, Apart, ArcCos,
 ArcCosh, ArcCot, ArcCoth, ArcCsc, ArcCsch, ArcSec, ArcSech, ArcSin,
 ArcSinh, ArcTan, ArcTanh, Arg, ArithmeticGeometricMean, Attributes}
```

In[73]:= **Select[Names["B*"], listbareQ]**

Out[73]=
```
{BernoulliB, BesselI, BesselJ, BesselK, BesselY, Beta,
 BetaRegularized, Binomial, BitAnd, BitNot, BitOr, BitXor, Boole}
```

In[74]:= **Select[Names["F*S*"], listbareQ]**

Out[74]= {FactorSquareFree}

In[75]:= **sN[x_]:=Select[Names[x], listbareQ]**

In[76]:= **sN["A*"]**

Out[76]= {Abs, AiryAi, AiryAiPrime, AiryBi, AiryBiPrime, Apart, ArcCos,
ArcCosh, ArcCot, ArcCoth, ArcCsc, ArcCsch, ArcSec, ArcSech, ArcSin,
ArcSinh, ArcTan, ArcTanh, Arg, ArithmeticGeometricMean, Attributes}

In[77]:=
```
Union[sN["A*"],sN["B*"],sN["C*"],sN["D*"],
                    sN["E*"]]
```

Out[77]= {Abs, AiryAi, AiryAiPrime, AiryBi, AiryBiPrime, Apart, ArcCos, ArcCosh, ArcCot,
ArcCoth, ArcCsc, ArcCsch, ArcSec, ArcSech, ArcSin, ArcSinh, ArcTan, ArcTanh,
Arg, ArithmeticGeometricMean, Attributes, BernoulliB, BesselI, BesselJ,
BesselK, BesselY, Beta, BetaRegularized, Binomial, BitAnd, BitNot, BitOr,
BitXor, Boole, Cancel, CarmichaelLambda, Ceiling, Characters, ChebyshevT,
ChebyshevU, Coefficient, Conjugate, ContinuedFraction, Cos, Cosh, Cot, Coth,
CreateDirectory, Csc, Csch, Decompose, Denominator, DiracDelta, Divide,
Divisors, DivisorSigma, EllipticE, EllipticF, EllipticK, EllipticPi,
EllipticTheta, EllipticThetaPrime, Erf, Erfc, Erfi, EulerE, EulerPhi,
EvenQ, Exp, ExpIntegralE, ExpIntegralEi, Exponent, ExpToTrig, ExtendedGCD}

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[78]:= **(* Old Form: Remove["Global`@*"] *)**

In[79]:= **Remove["Global`*"]**

Kurs ▮ Cours

10. Lokale Variablen und prozedurales Programmieren
▮ Variables locales et programmation procédurale

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....
Hinweis: Kapitel 10 lesen!
Run mit WIN+*Mathematica* Version 5.2

▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 10.
Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

10.1. Globale Variablen
▮ Variables globales

Mathematica stört es nicht, dass im folgenden Beispiel zweimal dieselbe globale Variable (hier "i") in verschiedenen Bedeutungen gebraucht wird:
▮ Cela ne dérange pas *Mathematica* que dans l'exemple suivant on emploie deux fois la même variable globale (ici "i") avec deux significations différentes:

```
In[1]:= f[n_]:=Table[g[i],  {i,n}];
        g[m_]:=Table[Log[i],{i,m}];

In[3]:= f[3]

Out[3]= {{0}, {0, Log[2]}, {0, Log[2], Log[3]}}

In[4]:= ??i

        Global`i
```

Zum Vergleich: ▮ Pour comparer:

```
In[5]:= f[n_]:=Table[g[i],  {i,n}];
        g[m_]:=Table[Log[j],{j,m}];
```

```
In[7]:= f[3]

Out[7]= {{0}, {0, Log[2]}, {0, Log[2], Log[3]}}
```

10.2. Lokale Variablen im Unterschied zu globalen Variablen
▮ Variables locales à la différence de variables globales

10.2.1. Lokale Variablen in einer "Block"-Prozedur
▮ Variables locales dans une procédure "Block"

```
In[8]:= ??Block

        Block[{x, y, ... }, expr] specifies that expr is
           to be evaluated with local values for the symbols x, y, ... . Block[
           {x = x0, ... }, expr] defines initial local values for x, ... . Mehr...

        Attributes[Block] = {HoldAll, Protected}
```

Betrachte das folgende Beispiel:
▮ Considère l'exemple suivant:

```
In[9]:= Clear[f,g];
        f[n_] := Block[{i}, Table[  g[i], {i, n}]];
        g[m_] := Block[{i}, Table[Log[i], {i, m}]]
```

Ausprobieren: ▮ Essayer:

```
In[12]:= i

Out[12]= i

In[13]:= ?f

        Global`f

        f[n_] := Block[{i}, Table[g[i], {i, n}]]

In[14]:= f[n]

        Table::iterb : Iterator {i, n} does not have appropriate bounds. Mehr…

        Table::iterb : Iterator {i, n} does not have appropriate bounds. Mehr…

Out[14]= Table[g[i], {i, n}]

In[15]:= f[3]

Out[15]= {{0}, {0, Log[2]}, {0, Log[2], Log[3]}}

In[16]:= f[5]

Out[16]= {{0}, {0, Log[2]}, {0, Log[2], Log[3]},
          {0, Log[2], Log[3], Log[4]}, {0, Log[2], Log[3], Log[4], Log[5]}}
```

In[17]:=
```
i
```

Out[17]=
```
i
```

Was ist passiert? (Welcher Wert ist in "i" gespeichert?) Betrachte:

■ Que s'est-il passé? (Quelle valeur est mémorisée dans "i"?) Considère:

In[18]:=
```
Clear[f,g,i];
Block[{i}, i]
```

Out[19]=
```
i
```

10.2.2. Lokale Variablen in einer "Module"-Prozedur
■ Variables locales dans une procédure "Module"

In[20]:=
```
??Module
```

```
Module[{x, y, ... }, expr] specifies that occurrences
   of the symbols x, y, ...  in expr should be treated as local. Module[
   {x = x0, ... }, expr] defines initial values for x, ... . Mehr...

Attributes[Module] = {HoldAll, Protected}
```

Vergleiche mit dem letzten Beispiel:

■ Compare au dernier exemple:

In[21]:=
```
Clear[f];
Module[{i}, i]
```

Out[22]=
```
i$36
```

In[23]:=
```
Clear[i];
Module[{i}, i]
```

Out[24]=
```
i$37
```

Was bedeutet wohl die ausgegebene Zahl?

■ Que signifie le nombre sorti?

Betrachte das folgende Beispiel:

■ Considère l'exemple suivant:

In[25]:=
```
Clear[f,g];
f[n_] := Module[{i}, Table[  g[i], {i, n}]];
g[m_] := Module[{i}, Table[Log[i], {i, m}]]
```

Ausprobieren: ■ Essayer:

In[28]:=
```
?f
```

```
Global`f

f[n_] := Module[{i}, Table[g[i], {i, n}]]
```

```
In[29]:= f[n]

        Table::iterb : Iterator {i$42, n} does not have appropriate bounds. Mehr…

Out[29]= Table[g[i$42], {i$42, n}]

In[30]:= f[3]

Out[30]= {{0}, {0, Log[2]}, {0, Log[2], Log[3]}}

In[31]:= f[5]

Out[31]= {{0}, {0, Log[2]}, {0, Log[2], Log[3]},
          {0, Log[2], Log[3], Log[4]}, {0, Log[2], Log[3], Log[4], Log[5]}}

In[32]:= i

Out[32]= i
```

Was ist passiert? (Welcher Wert ist in "i" gespeichert?)

- Que s'est-il passé? (Quelle valeur est mémorisée dans "i"?)

10.2.3. Lokale Variablen mit "With"
Variables locales avec "With"

Damit lassen sich Ausdrücke manipulieren:

- Par cela on peut manipuler des expressions:

```
In[33]:= ?With

        With[{x = x0, y = y0, ... }, expr] specifies that in expr occurrences
           of the symbols x, y, ...  should be replaced by x0, y0, ... . Mehr...

In[34]:= With[{x=2, y=a, z=b^3}, y Log[x^z]]

Out[34]= a Log[2^b^3]

In[35]:= Clear[x,y,z,r,s];
         Solve[{x+2==r-s,s+1==y+2z,2r+z==-4x},{x,r}]

Out[36]= {{x → 1/6 (-4 - 2 s - z), r → 1/6 (8 + 4 s - z)}}

In[37]:= With[{z=4, s=Sin[z^2]},
         Solve[{x+2==r-s,s+1==y+2z,2r+z==-4x},{x,r}]]

Out[37]= {{x → 1/3 (-4 - Sin[z^2]), r → 2/3 (1 + Sin[z^2])}}
```

10.3. Prozedurale Programmierung
▮ Programmation procédurale

Mathematica erlaubt prozedurale Programmierung, d.h. Befehlssequenzen mit Wiederholungen und Verzweigungen. Studiere die folgenden Befehle:

▮ *Mathematica* permet la programmation procédurale, c'est-à-dire des séquences d'ordres avec des répétitions et des ramifications. Etudier les ordres suivants:

```
In[38]:= ??Do
```

```
Do[expr, {imax}] evaluates expr imax times. Do[expr, {i, imax}] evaluates expr
   with the variable i successively taking on the values 1 through imax (in steps
   of 1). Do[expr, {i, imin, imax}] starts with i = imin. Do[expr, {i, imin,
   imax, di}] uses steps di. Do[expr, {i, imin, imax}, {j, jmin, jmax}, ... ]
   evaluates expr looping over different values of j, etc. for each i. Mehr...

Attributes[Do] = {HoldAll, Protected}
```

```
In[39]:= ?For
```

```
For[start, test, incr, body] executes start, then
   repeatedly evaluates body and incr until test fails to give True. Mehr...
```

```
In[40]:= ?While
```

```
While[test, body] evaluates test, then
   body, repetitively, until test first fails to give True. Mehr...
```

```
In[41]:= ?If
```

```
If[condition, t, f] gives t if condition evaluates to True, and f if it evaluates to False.
   If[condition, t, f, u] gives u if condition evaluates to neither True nor False. Mehr...
```

```
In[42]:= ?Which
```

```
Which[test1, value1, test2, value2, ... ] evaluates each of the testi in turn, returning
   the value of the valuei corresponding to the first one that yields True. Mehr...
```

```
In[43]:= ?Switch
```

```
Switch[expr, form1, value1, form2, value2, ... ] evaluates
   expr, then compares it with each of the formi in turn, evaluating and
   returning the valuei corresponding to the first match found. Mehr...
```

10.3.1. Arithmetische Operationen auf bestehenden Variablen wie Inkrementierung u.s.w.
▮ Opérations arithmétiques sur des variables existantes comme "increment" etc.

Studiere die folgenden Beispiele:

▮ Etudie les exemples suivants:

```
In[44]:= ??++
```

```
x++ increases the value of x by 1, returning the old value of x. Mehr...

Attributes[Increment] = {HoldFirst, Protected, ReadProtected}
```

In[45]:= **?+=**

x += dx adds dx to x and returns the new value of x. Mehr...

In[46]:= **?-=**

x -= dx subtracts dx from x and returns the new value of x. Mehr...

In[47]:= **?*=**

x *= c multiplies x by c and returns the new value of x. Mehr...

In[48]:= **?/=**

x /= c divides x by c and returns the new value of x. Mehr...

In[49]:= **?DivideBy**

x /= c divides x by c and returns the new value of x. Mehr...

In[50]:= **i=1**

Out[50]= 1

In[51]:= **++i**

Out[51]= 2

In[52]:= **i**

Out[52]= 2

In[53]:= **i++**

Out[53]= 2

In[54]:= **i**

Out[54]= 3

In[55]:= **i**

Out[55]= 3

In[56]:= **i +=6**

Out[56]= 9

In[57]:= **i -=5**

Out[57]= 4

In[58]:= **i *=7**

Out[58]= 28

In[59]:= **i /=4**

Out[59]= 7

10.3.2. Iterative Konstruktionen
▮ Constructions itératives

Studiere die folgenden Beispiele:
▮ Etudie les exemples suivants:

```
In[60]:= Do[Print[3 n^2], {n,7}]
```

```
3
12
27
48
75
108
147
```

```
In[61]:= For[n=1, n<=7, ++n, Print[3 n^2]]
```

```
3
12
27
48
75
108
147
```

```
In[62]:= n=1;
         While[n<=7, (Print[3 n^2]; n++)]
```

```
3
12
27
48
75
108
147
```

10.3.3. Logik
▮ Logique

Studiere die folgenden Befehle:
▮ Etudie les ordres suivants:

In[64]:= **?&&**

e1 && e2 && ... is the logical AND function. It evaluates its arguments in order, giving
False immediately if any of them are False, and True if they are all True. Mehr...

In[65]:= **?And**

e1 && e2 && ... is the logical AND function. It evaluates its arguments in order, giving
False immediately if any of them are False, and True if they are all True. Mehr...

In[66]:= **?||**

e1 || e2 || ... is the logical OR function. It evaluates its arguments in order, giving
True immediately if any of them are True, and False if they are all False. Mehr...

In[67]:= **?Or**

e1 || e2 || ... is the logical OR function. It evaluates its arguments in order, giving
True immediately if any of them are True, and False if they are all False. Mehr...

In[68]:= **?!**

Information::notfound : Symbol ! not found. Mehr…

In[69]:= **?!!**

n!! gives the double factorial of n. Mehr...

In[70]:= **?Not**

!expr is the logical NOT function. It gives False if expr is True, and True if it is False.
Mehr…

In[71]:= **?Xor**

Xor[e1, e2, ...] is the logical XOR (exclusive OR) function. It gives
True if an odd number of the ei are True, and the rest are False. It gives
False if an even number of the ei are True, and the rest are False. Mehr...

Studiere die folgenden Beispiele:

■ Etudie les exemples suivants:

In[72]:= **(1 < 2) || (4 > 5/0)**

Out[72]= True

In[73]:= **(4 > 5/0) || (1 < 2)**

Power::infy : Infinite expression $\frac{1}{0}$ encountered. Mehr…

Greater::nord : Invalid comparison with ComplexInfinity attempted. Mehr…

Out[73]= True

Von wo her wird also geklammert?

■ D'où viennent les parenthèses?

10.3.4. Wahrheitstabellen
▮ Tableaux de verité

Ein Beispiel: Die folgende Aussageform soll untersucht werden:

▮ Un exemple: La forme propositionelle suivante soit examinée:

```
In[74]:= Clear[f];
         f[x_:0, y_:0, z_:0, w_:0, u_:0, v_:0]:=
                 !(!x || (x && y)) || (x || ! y) && w;
         AppendTo[Attributes[f], Listable]

Out[76]= {Listable}

In[77]:= ??f

        Global`f

         Attributes[f] = {Listable}

         f[x_ : 0, y_ : 0, z_ : 0, w_ : 0, u_ : 0, v_ : 0] := ! (! x || (x && y)) || ((x || ! y) && w)
```

Die nachstehende Funktion definiert den "Input" in die Wahrheitstabelle:

▮ La fonction ci-dessous définit l'"Input" dans le tableau de vérité:

```
In[78]:= Remove[tabteil,tabelle];
         tabteil[n_, k_]:= Permutations[
                         Table[Ceiling[Floor[n/i]/k],
                                 {i,1,k}]];
         tabelle = {};
         t[k_]:=(Do[AppendTo[tabelle,tabteil[n,k]],
                                 {n,0,k}];
                 tabelle  = Sort[Flatten[tabelle,1]];
                 (tabelleTF=tabelle /. {0->False,
                                        1->True});
                 Print[MatrixForm[tabelle]];
                 MatrixForm[tabelleTF]);
         t[3]
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

```
Out[82]//MatrixForm=
```

$$\begin{pmatrix} \text{False} & \text{False} & \text{False} \\ \text{False} & \text{False} & \text{True} \\ \text{False} & \text{True} & \text{False} \\ \text{False} & \text{True} & \text{True} \\ \text{True} & \text{False} & \text{False} \\ \text{True} & \text{False} & \text{True} \\ \text{True} & \text{True} & \text{False} \\ \text{True} & \text{True} & \text{True} \end{pmatrix}$$

```
In[83]:= Remove[tabteil,tabelle];
         tabteil[n_, k_]:= Permutations[
                         Table[Ceiling[Floor[n/i]/k],
                                        {i,1,k}]];
         tabelle = {};
         t[k_]:=(Do[AppendTo[tabelle,tabteil[n,k]],
                                        {n,0,k}];
                 tabelle  = Sort[Flatten[tabelle,1]];
                 (tabelleTF=tabelle /. {0->False,
                                        1->True});
                 Print[MatrixForm[tabelle]];
                 MatrixForm[tabelleTF]);
         t[4]
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Out[87]//MatrixForm=

$$\begin{pmatrix} \text{False} & \text{False} & \text{False} & \text{False} \\ \text{False} & \text{False} & \text{False} & \text{True} \\ \text{False} & \text{False} & \text{True} & \text{False} \\ \text{False} & \text{False} & \text{True} & \text{True} \\ \text{False} & \text{True} & \text{False} & \text{False} \\ \text{False} & \text{True} & \text{False} & \text{True} \\ \text{False} & \text{True} & \text{True} & \text{False} \\ \text{False} & \text{True} & \text{True} & \text{True} \\ \text{True} & \text{False} & \text{False} & \text{False} \\ \text{True} & \text{False} & \text{False} & \text{True} \\ \text{True} & \text{False} & \text{True} & \text{False} \\ \text{True} & \text{False} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{False} & \text{False} \\ \text{True} & \text{True} & \text{False} & \text{True} \\ \text{True} & \text{True} & \text{True} & \text{False} \\ \text{True} & \text{True} & \text{True} & \text{True} \end{pmatrix}$$

Und hier der "Output" in der entsprechenden Reihenfolge:

■ Et voici l'"Output" dans l'ordre correspondant:

```
In[88]:= ??Map
```

```
Map[f, expr] or f /@ expr applies f to each element on the first level in expr. Map[
   f, expr, levelspec] applies f to parts of expr specified by levelspec. Mehr...

Attributes[Map] = {Protected}

Options[Map] = {Heads → False}
```

In[89]:=
```
f @@ Transpose[tabelleTF]
```

Out[89]=
```
{False, True, False, True, False, False, False,
 False, True, True, True, True, False, True, False, True}
```

In[90]:=
```
MatrixForm[Transpose[
    Join[Transpose[tabelleTF],
    {Table["|",{i,1,
        Length[f @@ Transpose[tabelleTF]]}]},
    {f @@ Transpose[tabelleTF]}]]]
```

Out[90]//MatrixForm=

False	False	False	False	|	False
False	False	False	True	|	True
False	False	True	False	|	False
False	False	True	True	|	True
False	True	False	False	|	False
False	True	False	True	|	False
False	True	True	False	|	False
False	True	True	True	|	False
True	False	False	False	|	True
True	False	False	True	|	True
True	False	True	False	|	True
True	False	True	True	|	True
True	True	False	False	|	False
True	True	False	True	|	True
True	True	True	False	|	False
True	True	True	True	|	True

```
In[91]:= Remove[tabteil,tabelle];
         tabteil[n_, k_]:= Permutations[
                         Table[Ceiling[Floor[n/i]/k],
                                              {i,1,k}]];
         tabelle = {};
         t[k_]:=(Do[AppendTo[tabelle,tabteil[n,k]],
                                              {n,0,k}];
                 tabelle  = Sort[Flatten[tabelle,1]];
                 (tabelleTF=tabelle /. {0->False,
                                              1->True});
                 MatrixForm[Transpose[
                 Join[Transpose[tabelleTF],
                 {Table["|",{i,1,
                     Length[f @@ Transpose[tabelleTF]]}]},
                 {f @@ Transpose[tabelleTF]}]]]);
         (*Anwendung*)
         t[5]
```

Out[96]//MatrixForm=

False	False	False	False	False	|	False
False	False	False	False	True	|	False
False	False	False	True	False	|	True
False	False	False	True	True	|	True
False	False	True	False	False	|	False
False	False	True	False	True	|	False
False	False	True	True	False	|	True
False	False	True	True	True	|	True
False	True	False	False	False	|	False
False	True	False	False	True	|	False
False	True	False	True	False	|	False
False	True	False	True	True	|	False
False	True	True	False	False	|	False
False	True	True	False	True	|	False
False	True	True	True	False	|	False
False	True	True	True	True	|	False
True	False	False	False	False	|	True
True	False	False	False	True	|	True
True	False	False	True	False	|	True
True	False	False	True	True	|	True
True	False	True	False	False	|	True
True	False	True	False	True	|	True
True	False	True	True	False	|	True
True	False	True	True	True	|	True
True	True	False	False	False	|	False
True	True	False	False	True	|	False
True	True	False	True	False	|	True
True	True	False	True	True	|	True
True	True	True	False	False	|	False
True	True	True	False	True	|	False
True	True	True	True	False	|	True
True	True	True	True	True	|	True

10.3.5. Verzweigungen
∎ Ramification

Probiere aus: ∎ Essaie:

In[97]:= **?If**

If[condition, t, f] gives t if condition evaluates to True, and f if it evaluates to False.
If[condition, t, f, u] gives u if condition evaluates to neither True nor False. Mehr...

In[98]:= **?PrintForm**

System`PrintForm

Attributes[PrintForm] = {Protected}

In[99]:= **PrintForm[Print"gagag"]**

Out[99]= HorizontalForm[{11, 1, 0, 0, {10, 4, 10, 2, 6}},
{Times, 400, None}, , , gagag, , Print,]

In[100]:=
?HorizontalForm

HorizontalForm is an internal symbol used for formatting and printing.

In[101]:=
?Write

Write[channel, expr1, expr2, ...] writes the expressions expri in
sequence, followed by a newline, to the specified output channel. Mehr...

In[102]:=
?WriteString

WriteString[channel, expr1, expr2, ...] converts the expri to strings,
and then writes them in sequence to the specified output channel. Mehr...

In[103]:=
?SequenceForm

SequenceForm[expr1, expr2, ...] prints as the
textual concatenation of the printed forms of the expri. Mehr...

In[104]:=

```
Clear[t];
Do[(t=Table[(If[Random[]>0.5,
    SequenceForm["omumo"],SequenceForm["wvovw"],
    SequenceForm["bubu"]]),{i,7}];
Print[t[[1]],t[[2]],t[[3]],t[[4]],t[[5]],
        t[[6]],t[[7]]]),{k,25}]
```

```
wvovwomumowvovwomumoomumowvovwwvovw

wvovwwvovwomumowvovwomumoomumoomumo

wvovwwvovwomumowvovwwvovwwvovwwvovw

omumoomumowvovwomumowvovwwvovwwvovw

wvovwwvovwwvovwomumowvovwwvovwomumo

wvovwwvovwwvovwomumowvovwomumowvovw

wvovwwvovwomumowvovwomumoomumoomumo

wvovwomumoomumowvovwwvovwomumoomumo

wvovwomumoomumowvovwomumowvovwwvovw

wvovwwvovwwvovwomumowvovwomumoomumo

wvovwwvovwomumoomumowvovwomumoomumo

omumoomumoomumowvovwwvovwomumoomumo

wvovwomumowvovwwvovwwvovwomumoomumo

wvovwomumoomumowvovwomumoomumowvovw

wvovwwvovwwvovwwvovwwvovwwvovwomumo

wvovwomumowvovwomumoomumowvovwwvovw

wvovwomumoomumowvovwwvovwwvovwwvovw

wvovwomumowvovwwvovwwvovwwvovwomumo

omumowvovwwvovwwvovwwvovwomumoomumo

omumoomumoomumoomumoomumowvovwwvovw

wvovwwvovwwvovwwvovwomumowvovwwvovw

omumowvovwomumoomumoomumoomumoomumo

omumowvovwwvovwwvovwomumoomumoomumo

omumowvovwwvovwomumowvovwomumoomumo

omumoomumoomumowvovwomumoomumoomumo
```

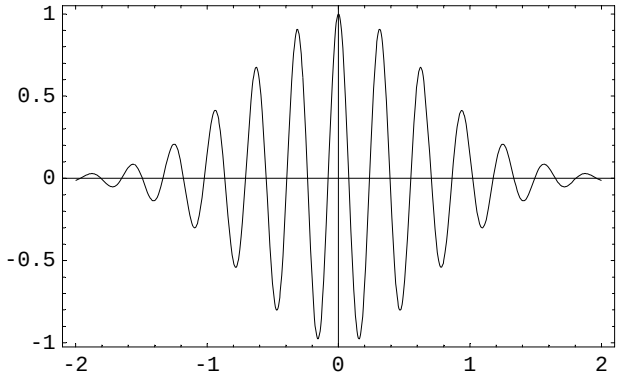
Verzweigung je nach *Mathematica*-Version:

■ Ramification selon la version de *Mathematica*:

In[106]:=

```
If[$VersionNumber < 2.0,
    Plot[E^(-x^2) Cos[20x],{x,-2,2},Framed->True],
    Plot[E^(-x^2) Cos[20x],{x,-2,2},Frame ->True]];
```

```
General::spell1 :
 Possible spelling error: new symbol name "Framed" is similar to existing symbol "Frame". Mehr…
```



Mit Switch: ■ Avec Switch:

In[107]:=

```
x=Random[Integer,{1,5}];
Print[x];
Switch[x^2,1,x,4,2x,9,3x,16,4x,25,-5x]
```

```
1
```

Out[109]=

```
1
```

In[110]:=

```
x=Random[Integer,{1,5}];
Print[x];
Switch[x^2,1,x,4,2x,9,3x,16,4x,25,-5x]
```

```
1
```

Out[112]=

```
1
```

In[113]:=

```
x=Random[Integer,{1,5}];
Print[x];
Switch[x^2,1,x,4,2x,9,3x,16,4x,25,-5x]
```

```
4
```

Out[115]=

```
16
```

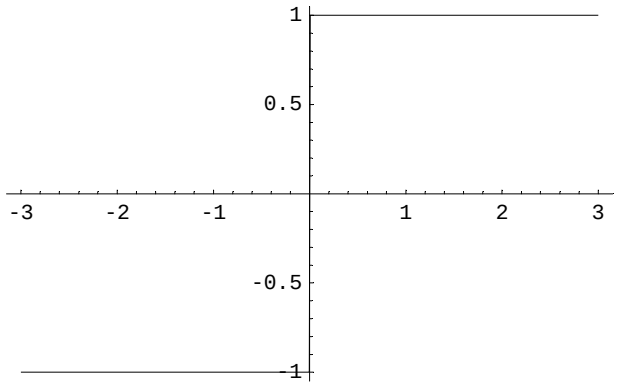
Mit Which: ■ Avec Which:

In[116]:=

```
signum[x_]:=Which[x<0.,-1,x==0.,0,x>0.,1]
```

In[117]:=

```
Plot[signum[x],{x,-3,3}];
```



10.3.6: Spaghetti-Code (resp. die "Chaos-Erzeugung"...)
■ Code spaghetti (résp. la "génération du chaos"...)

Davon ist abzuraten!

■ On le déconseille!

In[118]:=

```
?Goto
```

```
Goto[tag] scans for Label[tag], and transfers control to that point. Mehr...
```

In[119]:=

```
?Label
```

```
Label[tag] represents a point in a compound
   expression to which control can be transferred using Goto. Mehr...
```

In[120]:=

```
?Throw
```

```
Throw[value] stops evaluation and returns value as the
   value of the nearest enclosing Catch. Throw[value, tag] is caught only
   by Catch[expr, form] where form is a pattern that matches tag. Mehr...
```

In[121]:=

```
?Catch
```

```
Catch[expr] returns the argument of the first Throw generated in the evaluation
   of expr. Catch[expr, form] returns value from the first Throw[value, tag] for
   which form matches tag. Catch[expr, form, f] returns f[value, tag]. Mehr...
```

10.3.7: Beispiel aus der Statistik
■ Exemple pris de la statistique

Der Median: ■ La médiane:

In[122]:=
```
median[liste_List]:=
    Block[{
            sl,
            len
          },
          len = Length[liste];
          sl = Sort[liste];
          If[
             OddQ[len],
             sl[[(len+1)/2]],
             (sl[[len/2]]+sl[[len/2+1]])/2
          ]
    ]
```

```
General::spell1 :
 Possible spelling error: new symbol name "median" is similar to existing symbol "Median". Mehr…
```

In[123]:=
```
median[{53,64,78,24,63,78}] // N
```

Out[123]=
```
63.5
```

In[124]:=
```
median[{76, 56, 23, 78, 34}]
```

Out[124]=
```
56
```

In[125]:=
```
median[{0,1,2,3,4,5,6}]
```

Out[125]=
```
3
```

In[126]:=
```
median[{1,2,3,4,5,6}]  // N
```

Out[126]=
```
3.5
```

In[127]:=
```
median[{1,2,3,4,5,6,7}]
```

Out[127]=
```
4
```

Was ist der Median?

Qu'est-ce la médiane?

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

In[128]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[129]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

10. Lokale Variablen und prozedurales Programmieren
▮ Variables locales et programmation procédurale

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 10 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 10.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Bestimme die Anzahl Iterationsschritte, die notwendig sind, um "Wurzel aus 2" auf 30 Dezimalen mittels der Newton-Methode zu approximieren.
▮ Détermine le nombre de pas d'itération qui sont nécessaires pour approximer "racine de 2" jusqu'à 30 décimales par la méthode de Newton.

Starte mit dem Wert 2. Berechne die Wurzel mit der Formel x(i+1)=(x(i)+2/x(i))/2. Erkläre wie diese Formel entsteht. (Löse x^2 - 2 = 0 ...)
▮ Démarre avec la valeur 2. Calcule la racine par la formule x(i+1)=(x(i)+2/x(i))/2. Explique comment cette formule s'est formée. (Résous x^2 - 2 = 0 ...)

```
In[1]:= Clear[newtonMethode, counter, zahl, approx];
        newtonMethode[x_]:=
            Module[{
                    counter = 1,
                    zahl = N[x,30],
                    approx = N[x,40]
                },
                While[approx^2 != zahl,
                    Print[counter, ":  ", N[approx,30]];
                    approx = (approx + 2/approx)/2;
                    ++counter;
                ];
                Print[counter, ":  ", N[approx,30]];
            ]
```

```
In[3]:= newtonMethode[2]
```

```
1:  2.000000000000000000000000000000
2:  1.500000000000000000000000000000
3:  1.416666666666666666666666666667
4:  1.414215686274509803921568627450
5:  1.414213562374689910626295578890
6:  1.414213562373095048801689623500
7:  1.414213562373095048801688724210
```

Vergleich: ▮ Comparaison:

```
In[4]:= N[Sqrt[2],30]
```

```
Out[4]= 1.41421356237309504880168872421
```

Aufgabe 2 ▮ Problème 2

Definiere eine Funktion, die als Argumente drei Zahlen aufnimmt und die Summe der Quadrate der beiden grösseren Zahlen ausgibt.
▮ Définis une fonction qui prend 3 nombres comme arguments et sort la somme des carrés des deux nombres plus grands.

Definition: ▮ Définition:

```
In[5]:= Clear[f];
        f[x_,y_,z_]:=
            Module[{i,j,liste,min},
                min=Min{x,y,z};
                liste=Sort[{x,y,z}];
                restQuadr=Rest[{x,y,z}]^2;
                Print[Apply[Plus,restQuadr]]
                ]
```

```
General::spell1 :
 Possible spelling error: new symbol name "min" is similar to existing symbol "Min". Mehr…
```

Ausprobieren: ▮ Essayer:

```
In[7]:= {f[0,0,0],f[0,0,1],f[0,1,1],f[1,1,1],f[1,1,2],f[1,2,3]}

        0

        1

        2

        2

        5

        13

Out[7]= {Null, Null, Null, Null, Null, Null}

In[8]:= f[3]

Out[8]= f[3]
```

Was ist passiert?
▮ Que s'est-il passé?

Aufgabe 3 ▮ Problème 3

Annahme: Familien haben Kinder, bis erstmals ein Knabe kommt. Mache eine Simulation mit 1000 Familien um herauszufinden, wieviele Kinder etwa durchschnittlich eine Familie haben wird (Abschätzung). Wieviele Töchter und wieviele Söhne werden in einer Familie zu erwarten sein?
▮ Supposition: Des familles ont des enfants jusqu'à la naissance du 1er garçon. Fais une simulation avec 1000 familles pour obtenir le nombre moyen d'enfants qu'une famille aura (estimation). A combien de filles et à combien de fils pourra s'attendre une famille?

Lösung von Nancy Blachman:
▮ Solution de Nancy Blachman:

```
In[9]:= Clear[makeFamily];
        makeFamily[ ] :=
            Module[{
                    children = {}
                    },
                    While[Random[Integer] == 0,
                        AppendTo[children, "girl"]
                    ];
                    Append[children, "boy"]
                ]

In[11]:= makeFamily::usage="makeFamily[ ] returns a list
                           of children."

Out[11]= makeFamily[ ] returns a list      of children.
```

```
In[12]:= Clear[numChildren];
         numChildren[n_Integer]:=
             Module[{
                     allChildren
                     },
                     allChildren = Flatten[Table[
                             makeFamily[ ],{n}]];
                     {   avgChildren -> Length[
                                 allChildren]/n,
                         avgBoys -> Count[allChildren,
                                     "boy"]/n,
                         avgGirls -> Count[allChildren,
                                     "girl"]/n
                         }
                 ]

In[14]:= numChildren::usage="numChildren[n] returns
                     statistics on the number of children
                     from n families."

Out[14]= numChildren[n] returns    statistics on the number of children    from n families.

In[15]:= numChildren[1000]

Out[15]= {avgChildren → 493/250, avgBoys → 1, avgGirls → 243/250}
```

Aufgabe 4 ▮ Problème 4

Mischen von Kartenspielen:
▮ Mélanger des jeux de cartes:

Beim perfekten Mischen teilt der Spieler das Kartenspiel genau in der Mitte und mischt das Spiel im "Reissverschlussverfahren": Abwechslungsweise kommt von jeder Seite je eine Karte, wobei sicher sein muss, dass die erste Karte von der ersten Hälfte zuerst "hingelegt" wird (das Spiel wird umgekehrt gehalten). Z.B. wenn man ein Spiel mit 10 Karten hat, ursprünglich geordnet in der Reihenfolge {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, so ist es nach einmal perfektem Mischen in der Reihenfolge {1, 6, 2, 7, 3, 8, 4, 9, 5, 10}.
a) Schreibe eine Funktion "perfektMischen", die das perfekte Mischen einer geraden Anzahl von Karten simuliert.
b) Schreibe die Funktion "wiederOrdnung", die die Anzahl perfekter Mischungen berechnet um das Spiel wieder in die urspüngliche Ordnung zu bringen. Wieviel mal ist es bei 52 Karten?

▮

Un joueur qui mélange les cartes à la perfection, divise le jeu de cartes exactement par la moitié et mélange les cartes par la méthode de la "fermeture éclair": De chaque côté il arrive une carte alternativement, mais on doit être sûr que la 1ère carte vient de la 1ère moitié (on tient le jeu à l'envers). P.ex. si on a un jeu de 10 cartes dans l'ordre d'origine {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, on obtient l'ordre suivant en mélangeant une fois {1, 6, 2, 7, 3, 8, 4, 9, 5, 10}.
a) Ecris une fonction "perfectMischen" qui simule un mélange parfait d'un nombre pair de cartes.
b) Ecris la fonction "wiederOrdnung" qui calcule le nombre de mélange parfaits pour remettre le jeu dans l'ordre d'origine. Combien de mélanges faut-il pour un jeu de 52 cartes?

```
In[16]:= Clear[mischen, perfektMischen, mischen1,
                     perfektMischen1];

         mischen::usage="mischen[ ] erzeugt eine Liste
```

```
            von perfekt gemischten Zahlen -- donne
            une liste de nombres mélangés à la
            perfection.";
mischen[n_]:=
    Module[{matrix1,matrix2},
        listeIn=Range[n];
        matrix1=Partition[listeIn,n/2];
        matrix2=Transpose[matrix1];
        listeOut=Flatten[matrix2];
        Print["Liste_In: ", listeIn, "   ",
         "Liste_Out: ", listeOut]
     ];

perfektMischen::usage="perfektMischen[ ] greift
        auf den Modul mischen[ ] im Falle einer
        eingegebenen geraden natuerlichen Zahl --
        utilise le Modul mischen[ ] dans le cas
        d-un nombre donné pair, naturel.";

perfektMischen[n_Integer]:=
    Module[{listeIn},
        If[(OddQ[n] || n <= 0),
        Print["Bitte gerade, positive Zahl
        eingeben -- S.v.p. entrer nombre positif
        pair."], mischen[n],
        Print["Bitte gerade, positive Zahl
        eingeben -- S.v.p. entrer nombre positif
        pair."]]
        ];

mischen1::usage="mischen1[ ] erzeugt eine Liste
            von perfekt gemischten Zahlen ohne
            Output -- donne une liste de nombres
            mélangés à la perfection sans output.";
mischen1[n_]:=
    Module[{matrix1,matrix2},
        listeIn=Range[n];
        matrix1=Partition[listeIn,n/2];
        matrix2=Transpose[matrix1];
        listeOut=Flatten[matrix2];
        ];

perfektMischen1::usage="perfektMischen1[ ]
        greift auf den Modul mischen[ ]
        im Falle einer eingegebenen geraden
        natuerlichen Zahl. Im Unterschied zu
        perfektMischen[ ]
        wird hier die Ausgabe unterdruekt
        -- utilise le Modul mischen[ ] dans le cas
        d-un nombre donné pair, naturel. Par contre
        à perfektMischen[ ] l-output est retenu.";

perfektMischen1[n_Integer]:=
    Module[{},
        If[(OddQ[n] || n <= 0),
        Print["Bitte gerade, positive Zahl
        eingeben -- S.v.p. entrer nombre positif
        pair."], mischen1[n],
        Print["Bitte gerade, positive Zahl
        eingeben -- S.v.p. entrer nombre positif
        pair."]]];
```

In[27]:= **perfektMischen[100]**

```
Liste_In: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
  21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
  42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
  63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
  84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}   Liste_Out:
 {1, 51, 2, 52, 3, 53, 4, 54, 5, 55, 6, 56, 7, 57, 8, 58, 9, 59, 10, 60, 11, 61, 12, 62, 13, 63,
  14, 64, 15, 65, 16, 66, 17, 67, 18, 68, 19, 69, 20, 70, 21, 71, 22, 72, 23, 73, 24, 74, 25, 75, 26,
  76, 27, 77, 28, 78, 29, 79, 30, 80, 31, 81, 32, 82, 33, 83, 34, 84, 35, 85, 36, 86, 37, 87, 38, 88,
  39, 89, 40, 90, 41, 91, 42, 92, 43, 93, 44, 94, 45, 95, 46, 96, 47, 97, 48, 98, 49, 99, 50, 100}
```

In[28]:= **listeOut**

```
Out[28]= {1, 51, 2, 52, 3, 53, 4, 54, 5, 55, 6, 56, 7, 57, 8, 58, 9, 59, 10, 60, 11, 61,
  12, 62, 13, 63, 14, 64, 15, 65, 16, 66, 17, 67, 18, 68, 19, 69, 20, 70, 21, 71,
  22, 72, 23, 73, 24, 74, 25, 75, 26, 76, 27, 77, 28, 78, 29, 79, 30, 80, 31, 81,
  32, 82, 33, 83, 34, 84, 35, 85, 36, 86, 37, 87, 38, 88, 39, 89, 40, 90, 41,
  91, 42, 92, 43, 93, 44, 94, 45, 95, 46, 96, 47, 97, 48, 98, 49, 99, 50, 100}
```

In[29]:=
```
Clear[mischenAnzahl];

mischenAnzahl::usage="mischenAnzahl[ ] greift auf
        den Modul perfektMischen1[ ] und wendet
        diesen an, bis die ursprüngliche
        Reihenfolge wieder hergestellt ist. Ein
        Zaehler zaehlt die Anzahl Mischablaeufe
        -- utilise le Modul perfektMischen1[ ]
        et l-utilise jusqu-à l-ordre original est
        remis. Un compteur compte le nombre de
        mélanges.";

mischenAnzahl[n_]:=
    Module[{liste, counter=1, matrix1, matrix2},
        mischen1[n];
        liste  =listeIn;
        While[((listeOut != liste) &&
            counter < 10000), Print[counter,"  ",
            listeIn,"  ", listeOut];
            matrix1=Partition[listeOut,n/2];
            matrix2=Transpose[matrix1];
            listeOut=Flatten[matrix2];
            ++counter;
            ]
        Print[counter,"  ",listeIn,"  ", listeOut];
        Print["Anzahl Mischungen --
            Nombre de mélanges:   ", counter];

     ]

perfektMischenAnzahl::usage="perfektMischenAnzahl[ ]
        greift auf den Modul mischenAnzahl[ ] im
        Falle einer eingegebenen geraden
        natuerlichen Zahl  --
        utilise le Modul mischenAnzahl[ ] dans le
        cas d-un nombre donné pair, naturel..";


perfektMischenAnzahl[n_Integer]:=
    Module[{},
        If[(OddQ[n] || n <= 0),
        Print["Bitte gerade, positive Zahl
        eingeben-- S.v.p. entrer nombre positif
        pair."], mischenAnzahl[n],
        Print["Bitte gerade, positive Zahl
        eingeben-- S.v.p. entrer nombre positif
        pair."]]]
```

```
Syntax::newl:
   The newline character after "           ]"
     is understood as a multiplication operator.
     Mehr…
```

In[34]:=
```
perfektMischenAnzahl[1]
```

```
Bitte gerade, positive Zahl   eingeben-- S.v.p. entrer nombre positif   pair.
```

In[35]:=
```
perfektMischenAnzahl[0]
```

```
Bitte gerade, positive Zahl   eingeben-- S.v.p. entrer nombre positif   pair.
```

In[36]:= **perfektMischenAnzahl[2]**

```
1  {1, 2}  {1, 2}

Anzahl Mischungen --    Nombre de mélanges:   1
```

In[37]:= **perfektMischenAnzahl[4]**

```
1  {1, 2, 3, 4}  {1, 3, 2, 4}

2  {1, 2, 3, 4}  {1, 2, 3, 4}

Anzahl Mischungen --    Nombre de mélanges:   2
```

In[38]:= **perfektMischenAnzahl[6]**

```
1  {1, 2, 3, 4, 5, 6}  {1, 4, 2, 5, 3, 6}

2  {1, 2, 3, 4, 5, 6}  {1, 5, 4, 3, 2, 6}

3  {1, 2, 3, 4, 5, 6}  {1, 3, 5, 2, 4, 6}

4  {1, 2, 3, 4, 5, 6}  {1, 2, 3, 4, 5, 6}

Anzahl Mischungen --    Nombre de mélanges:   4
```

In[39]:= **perfektMischenAnzahl[8]**

```
1  {1, 2, 3, 4, 5, 6, 7, 8}  {1, 5, 2, 6, 3, 7, 4, 8}

2  {1, 2, 3, 4, 5, 6, 7, 8}  {1, 3, 5, 7, 2, 4, 6, 8}

3  {1, 2, 3, 4, 5, 6, 7, 8}  {1, 2, 3, 4, 5, 6, 7, 8}

Anzahl Mischungen --    Nombre de mélanges:   3
```

In[40]:= **perfektMischenAnzahl[10]**

```
1  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  {1, 6, 2, 7, 3, 8, 4, 9, 5, 10}

2  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  {1, 8, 6, 4, 2, 9, 7, 5, 3, 10}

3  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  {1, 9, 8, 7, 6, 5, 4, 3, 2, 10}

4  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  {1, 5, 9, 4, 8, 3, 7, 2, 6, 10}

5  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  {1, 3, 5, 7, 9, 2, 4, 6, 8, 10}

6  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Anzahl Mischungen --    Nombre de mélanges:   6
```

In[41]:= **perfektMischenAnzahl[12]**

```
1  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 7, 2, 8, 3, 9, 4, 10, 5, 11, 6, 12}

2  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 4, 7, 10, 2, 5, 8, 11, 3, 6, 9, 12}

3  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 8, 4, 11, 7, 3, 10, 6, 2, 9, 5, 12}

4  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 10, 8, 6, 4, 2, 11, 9, 7, 5, 3, 12}

5  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 12}

6  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 6, 11, 5, 10, 4, 9, 3, 8, 2, 7, 12}

7  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 9, 6, 3, 11, 8, 5, 2, 10, 7, 4, 12}

8  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 5, 9, 2, 6, 10, 3, 7, 11, 4, 8, 12}

9  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 3, 5, 7, 9, 11, 2, 4, 6, 8, 10, 12}

10  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Anzahl Mischungen --    Nombre de mélanges:   10
```

In[42]:= **perfektMischenAnzahl[14]**

```
1  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 8, 2, 9, 3, 10, 4, 11, 5, 12, 6, 13, 7, 14}

2  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 11, 8, 5, 2, 12, 9, 6, 3, 13, 10, 7, 4, 14}

3  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 6, 11, 3, 8, 13, 5, 10, 2, 7, 12, 4, 9, 14}

4  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 10, 6, 2, 11, 7, 3, 12, 8, 4, 13, 9, 5, 14}

5  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 12, 10, 8, 6, 4, 2, 13, 11, 9, 7, 5, 3, 14}

6  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 14}

7  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 7, 13, 6, 12, 5, 11, 4, 10, 3, 9, 2, 8, 14}

8  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 4, 7, 10, 13, 3, 6, 9, 12, 2, 5, 8, 11, 14}

9  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 9, 4, 12, 7, 2, 10, 5, 13, 8, 3, 11, 6, 14}

10  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 5, 9, 13, 4, 8, 12, 3, 7, 11, 2, 6, 10, 14}

11  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 3, 5, 7, 9, 11, 13, 2, 4, 6, 8, 10, 12, 14}

12  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

Anzahl Mischungen --    Nombre de mélanges:   12
```

In[43]:= **perfektMischenAnzahl[16]**

```
1  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 9, 2, 10, 3, 11, 4, 12, 5, 13, 6, 14, 7, 15, 8, 16}

2  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 4, 8, 12, 16}

3  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 3, 5, 7, 9, 11, 13, 15, 2, 4, 6, 8, 10, 12, 14, 16}

4  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
   {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}

Anzahl Mischungen --    Nombre de mélanges:   4
```

In[44]:= **perfektMischenAnzahl[18]**

```
1  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 10, 2, 11, 3, 12, 4, 13, 5, 14, 6, 15, 7, 16, 8, 17, 9, 18}

2  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 14, 10, 6, 2, 15, 11, 7, 3, 16, 12, 8, 4, 17, 13, 9, 5, 18}

3  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 16, 14, 12, 10, 8, 6, 4, 2, 17, 15, 13, 11, 9, 7, 5, 3, 18}

4  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 18}

5  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 9, 17, 8, 16, 7, 15, 6, 14, 5, 13, 4, 12, 3, 11, 2, 10, 18}

6  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 5, 9, 13, 17, 4, 8, 12, 16, 3, 7, 11, 15, 2, 6, 10, 14, 18}

7  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 3, 5, 7, 9, 11, 13, 15, 17, 2, 4, 6, 8, 10, 12, 14, 16, 18}

8  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
   {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}

Anzahl Mischungen --    Nombre de mélanges:   8
```

In[45]:= **(* perfektMischenAnzahl[20] *)**

In[46]:= **(* perfektMischenAnzahl[52] *)**

Aufgabe 5 ▮ Problème 5

Zum Unterschied zwischen "Block" und "Module" ▮ Quant à la différence entre "Block" er "Module"

Betrachte die folgenden Beispiele. Sie sind bis auf die Anweisungen "Block" resp. "Module" identisch. Wieso gibt *Mathematica* verschiedene Resultate aus?

▮ Considère les exemples suivants. Ils sont identiques, sauf les directives "Block" resp. "Module". Pourquoi est-ce que *Mathematica* sort des résultats différents?

In[47]:=
```
Clear[a, i];
a = i;
i = 3;
Block[{i},
        Table[a, {i, 2}]
    ]
```

Out[50]= {1, 2}

In[51]:=
```
Clear[a, i];
a = i;
i = 3;
Module[{i},
        Table[a, {i, 2}]
    ]
```

Out[54]= {3, 3}

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[55]:= **(* Old Form: Remove["Global`@*"] *)**

In[56]:= **Remove["Global`*"]**

Kurs ▮ Cours

11. Problemsammlung zur Musterentsprechung (Mustererkennen, musterkonformes Abarbeiten)
▮ Collection de problèmes concernant les correspondances et patrons (reconnaître les patrons, travailler conformement aux patrons)

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 11 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 11.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

11.1. Mustererkennung bei einer Sequenz
▮ Reconnaître un patron lors d'une séquence

Ein Beispiel
▮ Un exemple

Bisher bekannte Variablen mit "_": Z.B. in f[x_]:=x wird bei der Anwendung f[y] x durch y ersetzt. Oder bei f[3] x durch 3. Oder bei f[x+2y-Sin[z]] wird x durch x+2y-Sin[z] ersetzt. *Mathematica* erkennt hier das "Muster" des zu ersetzenden Ausdrucks. Wie lässt sich damit z.B. der Mittelwert einer
beliebig grossen Zahlenmenge berechnen? Beispiel:
▮ Variables avec "_" déjà connues: P.ex. dans f[x_]:=x on remplace f[y] x par y à l'application.Ou dans f[3] x par 3. Ou dans f[x+2y-Sin[z]] on remplace x par x+2y-Sin[z]. *Mathematica* reconnaît ici le "patron" de l'expression à remplacer. Comment peut-on calculer p.ex. la valeur moyenne d'un ensemble d'une grandeur quelconque de nombres? Exemple:

```
In[1]:= mittel[a_]:= a;
        mittel[a_, b_]:= (a+b)/2;
        mittel[a_, b_, c_]:= (a+b+c)/3;  (*etc. ....*)
```

In[4]:= **mittel[4,6]**

Out[4]= 5

In[5]:= **mittel[9]**

Out[5]= 9

In[6]:= **mittel[1,2,3]**

Out[6]= 2

In[7]:= **mittel[1,2,3,4]**

Out[7]= mittel[1, 2, 3, 4]

Was tun? - Wir können auch das Symbol "__" (doppelter Unterstrich) verwenden resp. "___" (dreifach).
■ Que faire? Nous pouvons employer aussi le symbole "__" (double sous-ligne) resp. "___" (triple).

In[8]:= **??__**

__ (two _ characters) or BlankSequence[ ] is a pattern object that can stand for any sequence of one or more Mathematica expressions. __h or BlankSequence[h] can stand for any sequence of one or more expressions, all of which have head h. Mehr...

Attributes[BlankSequence] = {Protected}

In[9]:= **?___**

___ (three _ characters) or BlankNullSequence[ ] is a pattern object that can stand for any sequence of zero or more Mathematica expressions. ___h or BlankNullSequence[h] can stand for any sequence of expressions, all of which have head h. Mehr...

Beispiel: ■ Exemple:

In[10]:=
```
Remove[mittel];
mittel[x__] := Plus[x]/Length[{x}]
```

In[12]:= **mittel[1,2,3,4,5,6,7,8,9]**

Out[12]= 5

In[13]:= **mittel[a,b,c,d,e,f,g]**

Out[13]= $\frac{1}{7}(a+b+c+d+e+f+g)$

In[14]:= **mittel[{a,b,c,d,e,f,g}]**

Out[14]= {a, b, c, d, e, f, g}

Um die Funktion auch auf eine Liste anwendbar zu machen, definieren wir:
■ Pour rendre la fonction aussi applicable à une liste, nous définissons:

In[15]:= **mittel[{x__}] := mittel[x]**

In[16]:= **mittel[{a,b,c,d,e,f,g}]**

Out[16]= $\frac{1}{7}(a+b+c+d+e+f+g)$

Nun klappt es!
Voilà, ça marche!

11.2. Nachbau von Funktionen
Copier (imiter) des fonctions

11.2.1. "First" (herauspicken des 1. Elements einer Liste)
"First" (choisir le 1er élément d'une liste)

"Von Hand":
"A la main":

Probiere: Essaie:

```
In[17]:= {a, b, c, d, e, f, g} /. {x_, y___} :> x
         (* 3 Unterstriche! *)

Out[17]= a
```

Definition einer Funktion:
Définition d'une fonction:

```
In[19]:= nimmErstes[{x_, y___}] := x
```

Ausprobieren: Essayer:

```
In[20]:= nimmErstes[{a, b, c, d, e, f, g}]

Out[20]= a
```

11.2.2. "Last" (herauspicken des letzten Elements einer Liste)
"Last" (choisir le dernier élément d'une liste)

"Von Hand":
"A la main":

Probiere: Essaie:

```
In[21]:= {a, b, c, d, e, f, g} /. {x___, y_} :> y

Out[21]= g
```

Definition einer Funktion:
Définition d'une fonction:

```
In[22]:= nimmLetztes[{x___, y_}] := y
```

Ausprobieren: ▮ Essayer:

```
In[23]:= nimmLetztes[{a, b, c, d, e, f, g}]

Out[23]= g
```

11.2.3. "Head"
▮ "Head"

"Von Hand":
▮ "A la main":

Probiere: ▮ Essaie:

```
In[24]:= {a, b, c, d, e, f, g} /. h_[x___] :> h
         (* 3 Unterstriche! *)

Out[24]= List
```

Definition einer Funktion:
▮ Définition d'une fonction:

```
In[26]:= nimmKopf[h_[x___]] := h
```

Ausprobieren: ▮ Essayer:

```
In[27]:= nimmKopf[{a, b, c, d, e, f, g}]

Out[27]= List
```

11.2.4. Neuordnung von Elementen
▮ Mettre des éléments dans un nouvel ordre

Zum Beispiel Vertauschung der Reihenfolge
▮ Par exemple renverser l'ordre

Probiere: ▮ Essaie:

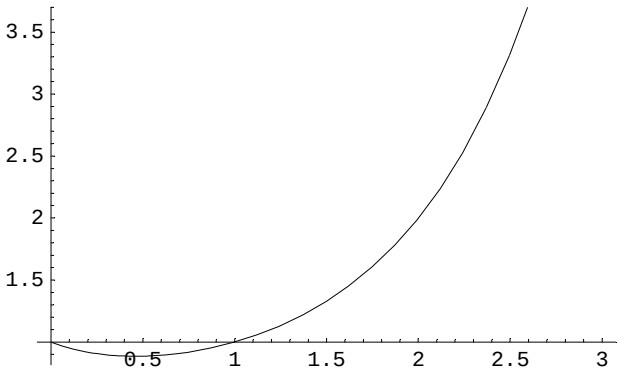
```
In[28]:= {{1,3},{2,4},{3,5},{4,6}} /. {x_, y_} :> {y, x}

Out[28]= {{3, 1}, {4, 2}, {5, 3}, {6, 4}}
```

11.3. "Polymorphe" Definitionen
▮ Définitions "polymorphes"

Hier geht es um datenabhängige Definitionen!
▮ Il s'agit ici de définitions dépendantes de données!

Gemeint sind da Definitionen, die ihre Form ändern je nach den Daten, auf die sie angewandt werden. Beispiel:
▮ Il s'agit ici de définitions qui changent leur forme selon les données auxquelles on les applique. Exemple:

```
In[29]:= Remove[pasc];
         pasc[n_Integer] := Table[Binomial[n,i],
                                  {i,0,n}];
         pasc[n_Real   ] := n!;

In[32]:= pasc[5]

Out[32]= {1, 5, 10, 10, 5, 1}

In[33]:= pasc[5.5]

Out[33]= 287.885

In[34]:= Map[pasc,{1,1.2,1.4,1.6,1.8,2,2.2,2.4,2.6,2.8,
                                          3,3.2}]

Out[34]= {{1, 1}, 1.1018, 1.24217, 1.42962, 1.67649, {1, 2, 1},
          2.42397, 2.98121, 3.71702, 4.69417, {1, 3, 3, 1}, 7.75669}

In[35]:= Plot[n!,{n,0,3}];
```



11.4. Benennung von Ausdrücken
▮ Nommer des expressions

Das Muster "x:Muster"
▮ Le patron "x:Patron"

Komplette Ausdrücke können mit Namen versehen werden. Beispiel:
▮ Les expressions complètes peuvent être pourvues de noms. Exemple:

```
In[36]:= 2 + 3 Cos[x] /.
           a:(x_ +y_ Cos[z_]) :>
               {{"a", a},{"x", x},{"y",y},{"z",z}}

Out[36]= {{a, 2 + 3 Cos[x]}, {x, 2}, {y, 3}, {z, x}}
```

Identifiziere die Namen!
▮ Identifier les noms!

11.5. Ausdrücke auffinden, die mit Mustern übereinstimmen
▮ Chercher des expressions qui correspondent à des patrons

11.5.1. Beispiele mit "Cases"
▮ Exemples avec "Cases"

Probiere aus: ▮ Essaie:

```
In[37]:= ??Cases
```

```
Cases[{e1, e2, ... }, pattern] gives a list of the ei that match the pattern. Cases[
   {e1, ... }, pattern -> rhs] gives a list of the values of rhs corresponding to
   the ei that match the pattern. Cases[expr, pattern, levspec] gives a list of all
   parts of expr on levels specified by levspec which match the pattern. Cases[expr,
   pattern -> rhs, levspec] gives the values of rhs which match the pattern. Cases[expr,
   pattern, levspec, n] gives the first n parts in expr which match the pattern. Mehr...

Attributes[Cases] = {Protected}

Options[Cases] = {Heads → False}
```

Studieren: ▮ Etudier:

```
In[38]:= Cases[{1, 2.3, {a, b}, x^2 +7, x+y+z},  x_]

Out[38]= {1, 2.3, {a, b}, 7 + x^2, x + y + z}
```

Einschränkung auf Integers oder Reals:

■ Se limiter à des Integers ou Reals:

```
In[39]:= Cases[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                                        x_Integer]

Out[39]= {1}

In[40]:= Cases[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                                        x_Real]

Out[40]= {2.3, 3.}

In[41]:= ?|

        p1 | p2 | ...  is a pattern object which represents any of the patterns pi. Mehr...
```

Einschränkung auf eine Alternative von Integers oder Reals:

■ Se limiter à une alternative de Integers ou de Reals:

```
In[42]:= Cases[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                 (x_Integer | x_Real)]

Out[42]= {1, 2.3, 3.}
```

Einschränkung auf Listen:

■ Se limiter à des listes:

```
In[43]:= Cases[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                                           x_List]

Out[43]= {{a, b}}
```

Einschränkung auf "Numbers":

■ Se limiter à "Numbers":

```
In[44]:= Cases[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                                        x_?NumberQ]

Out[44]= {1, 2.3, 3.}
```

11.5.2. Beispiel mit "Select"
■ Exemple avec "Select"

Nochmals die letzte Operation, aber mit "Select":

■ Encore une fois la dernière opération, mais avec "Select":

```
In[45]:= Select[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                                        NumberQ]

Out[45]= {1, 2.3, 3.}
```

11.5.3. Zusammengesetzte Muster
▮ Patrons composés

Z.B. stimmt der Ausdruck "a+b+c+d" überein mit dem Muster "x_ + y__":
▮ P.ex. l'expression "a+b+c+d" correspond au patron "x_ + y__":

```
In[46]:= Cases[{1, 2.3, 3., {a, b}, x^2 +7, x+y+z},
                                          x_+y__]
```

```
Out[46]= {7 + x^2, x + y + z}
```

Oder: ▮ Ou:

```
In[47]:= Cases[{1, a + I 5, 3., {a, b}, 2 - 3I, x+y+z},
                 x_+I y_]
```

```
Out[47]= {}
```

```
In[48]:= Cases[{1, a + 5I, 3., {a, b}, 2 - 3I, x+y+z},
                 x_+I y_]
```

```
Out[48]= {}
```

Wieso hat das jetzt nicht "funktioniert"? - Studiere:
▮ Pourquoi cela n'a-t-il pas fonctionné? - Etudie:

```
In[49]:= FullForm[a + 5I]
```

```
Out[49]//FullForm=
       Plus[Complex[0, 5], a]
```

```
In[50]:= FullForm[a + I 5]
```

```
Out[50]//FullForm=
       Plus[Complex[0, 5], a]
```

("Complex" ist atomar. Man kann also diesen "Head" suchen.)
▮ ("Complex" est atomique. On peut donc chercher ce "Head".)

```
In[51]:= Cases[{1, a + 5I, 3., {a, b}, 2 - 3I, x+y+z},
                 x_Complex]
```

```
Out[51]= {2 - 3 i}
```

```
In[52]:= Cases[{1, a + 5I, 3., {a, b}, 2 - 3I, x+y+z},
                Plus[x_Complex,y_]]
```

```
Out[52]= {5 i + a}
```

11.6. Das Attribut "Orderless"
▮ L'attribut "Orderless"

Studiere das folgende Beispiel:
▮ Etudie l'exemple suivant:

```
In[53]:= f[a_Integer, b_Complex, c_Real] := {a, b, c}

In[54]:= f[5, 3-8I, 3.346]

Out[54]= {5, 3 - 8 i, 3.346}

In[55]:= f[3-8I, 3.346, 5]

Out[55]= f[3 - 8 i, 3.346, 5]
```

Offenbar spielt die Reihenfolge der Argumente eine Rolle. Das kann aber umgangen werden! Dazu muss allerdings für f das Attribut "Orderless" gesetzt und f neu definiert werden! Probiere aus:

■ Evidemment l'ordre des arguments joue un rôle. Cependant on peut éviter cela! Pour cela il faut remplacer f par l'attribut "Orderless" et définir à nouveau f ! Essaie:

```
In[56]:= Attributes[f]

Out[56]= {}

In[57]:= ClearAll[f];
         SetAttributes[f, Orderless];
         f[a_Integer, b_Complex, c_Real] := {a, b, c}

In[60]:= f[3-8I, 3.346, 5]

Out[60]= {5, 3 - 8 i, 3.346}
```

Achtung! Die Attribute Orderless, Flat, HoldAll, FoldFirst, HoldRest und Listable müssen vor der Definition der Funktion gesetzt werden!

■ Attention! Les attributs Orderless, Flat, HoldAll, FoldFirst, HoldRest et Listable doivent être placés avant la définition de la fonction!

11.7. Beispiele mit Mustererkennung
■ Exemples avec reconnaissance de patron

11.7.1. Selektives Ausmultiplizieren
■ Multiplications sélectives

Beispiel
■ Exemple

Es soll eine Funktion geschrieben werden, die es erlaubt, ausschliesslich nur die Terme auszumultiplizieren, die unter einer Logarithmus-Funktion stehen. Eine Lösung:

■ Qu'on écrive une fonction qui permette de multiplier exclusivement les termes qui sont placés sous une fonction de logarithme. Voici une solution:

```
In[61]:= Clear[logAusmult];
         logAusmult[x_] := x /. Log[y_] :> Log[Expand[y]]
```

```
In[63]:= logAusmult[(4-3x)^5 Log[-(1-x)^3] /(x-1)^3]
```

$$\text{Out[63]}= \frac{(4-3x)^5 \,\text{Log}[-1+3x-3x^2+x^3]}{(-1+x)^3}$$

11.7.2. Manipulation von Klammern
▮ Manipulation de parenthèses

Beispiele
▮ Exemples

Eine Liste von Listen soll in eine Liste verwandelt werden:

▮ Qu'on transforme une liste de listes en une liste:

```
In[64]:= {{a, b, c, d},{e, f, g}} /.
                        {{x__},{y__}} :> {x, y}

Out[64]= {a, b, c, d, e, f, g}

In[65]:= {{a, b, c, d},{e, f, g}} /.
                        {x__} :> x

Out[65]= Sequence[{a, b, c, d}, {e, f, g}]

In[66]:= Remove[f];
         Map[f,{{a, b, c, d},{e, f, g}}]

Out[67]= {f[{a, b, c, d}], f[{e, f, g}]}

In[68]:= Map[f,{{a, b, c, d},{e, f, g}}] /.
                        f[{x__}] :> f[x]

Out[68]= {f[a, b, c, d], f[e, f, g]}
```

11.7.3. Eine eigene Definition von "Map"
▮ Une propre définition de "Map"

Studiere das folgend Beispiel:

▮ Etudie l'exemple suivant:

```
In[69]:= fktMap[f_, {}] := {};
         fktMap[f_, {a_, b___}] :=
                     Prepend[fktMap[f, {b}], f[a]];


In[71]:= fktMap[Sin,{a, 2, 3, 4, 5, 6, 7}]

Out[71]= {Sin[a], Sin[2], Sin[3], Sin[4], Sin[5], Sin[6], Sin[7]}
```

11.7.4. Ziffern zählen
▮ Compter des Chiffres

Beispiel
▮ Exemple

Es soll die Frage beantwortet werden, ob es sechs aufeinanderfolgende Ziffern "9" gibt unter den ersten 1000 Dezimalstellen von Pi. Falls ja, so soll die Anzahl Dezimalstellen zwischen Komma und dieser Sechsergruppe ausgegeben werden. Studiere die Lösung:
▮ Il faut répondre à la question si le chiffre "9" apparaît 6 fois de suite parmi les 1000 premières unités décimales de Pi. Si cela est le cas, il faut sortir le nombre d'unités décimales situées entre la virgule et ce groupe de six "9". Etudie la situation:

```
In[72]:= Characters[ToString[1234567]]
```

```
Out[72]= {1, 2, 3, 4, 5, 6, 7}
```

```
In[73]:= charListe = Characters[ToString[N[Pi, 10]]]
```

```
Out[73]= {3, ., 1, 4, 1, 5, 9, 2, 6, 5, 4}
```

Ohne Output (1000....)!
Sans output (1000...)!

```
In[74]:= charListe = Characters[ToString[N[Pi, 1000]]];
```

```
In[75]:= findeNeun[{"3", ".", x___,
                "9","9","9","9","9","9",___}] :=
                           Length[{x}];
```

```
In[76]:= findeNeun[charListe]
```

```
Out[76]= 761
```

11.7.5. Uebergabe von Optionen (Options)
▮ Transmission d'options

Beispiel
▮ Exemple

Studiere: ▮ Etudie

```
In[77]:= Needs["Graphics`Graphics`"]
```

In[78]:= **??Graphics**

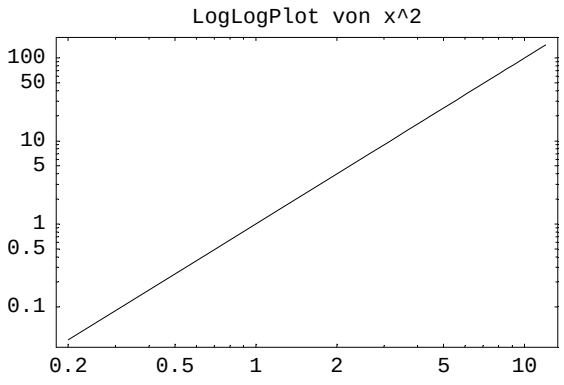
```
Graphics[primitives, options] represents a two-dimensional graphical image. Mehr...

Attributes[Graphics] = {Protected, ReadProtected}

Options[Graphics] = {AspectRatio → 1/GoldenRatio, Axes → False, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
  ColorOutput → Automatic, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
  DisplayFunction :→ $DisplayFunction, Epilog → {}, FormatType :→ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
  ImageSize → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

Beachte: ■ Observe:

Der zu verwendende Modul sieht etwa so aus
(- man sieht sofort, was übergeben werden kann...):
■ Le module à utiliser a environ cet aspect
(- on voit tout de suite ce qui peut être transmis...):

```
LogLogPlot[f_, {x_, xmin_, xmax_}, opts___] :=
      Module[{r,
        g = ParametricPlot[{Log[10,x], Log[10,f]}, {x, xmin, xmax},
            Ticks->{LogScale, LogScale},
            FrameTicks -> {LogScale, LogScale,
                  LogScale, LogScale},
            DisplayFunction -> Identity, opts]},
    r = PlotRange[g];
    Show[g, DisplayFunction -> $DisplayFunction,
            PlotRange -> r, AxesOrigin -> Map[#[[1]]&,r]]]
```

In[79]:=
```
LogLogPlot[x^2, {x, 0.2, 12},
           PlotLabel -> "LogLogPlot von x^2",
           Frame -> True];
```



Versuche dieses Beispiel mit anderen Optionen!
■ Essaie cet exemple avec d'autres options!

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[80]:= (* Old Form: Remove["Global`@*"] *)

In[81]:= Remove["Global`*"]
```

Uebungen ▮ Exercices

11. Problemsammlung zur Musterentsprechung (Mustererkennen, musterkonformes Abarbeiten)
▮ Collection de problèmes concernant les correspondances et patrons (reconnaître les patrons, travailler conformement aux patrons)

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 11 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 11.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Schreibe eine Funktion "geometrisches Mittel", die das geometrische Mittel einer Liste von Zahlen ausgibt.
Ecris une fonction "moyenne géométrique", qui donne la moyenne géométrique d'une liste de nombres.

Z.B. das geometrische Mittel von {a,b,c,d,e} ist (a b c d e)^(1/5).
▮ P.ex. la moyenne géométrique de {a,b,c,d,e} est (a b c d e)^(1/5).

```
In[1]:= Clear[geometrMittel];
        geometrMittel[{x__}]:= Times[x]^(1/Length[{x}])

In[3]:= geometrMittel[{1,1,1,1,1}]

Out[3]= 1
```

```
In[4]:= geometrMittel[{2,16,8,4,32}]
Out[4]= 8

In[5]:= geometrMittel[{2,3,5}]
Out[5]= 30^(1/3)

In[6]:= geometrMittel[{2,3,5}] // N
Out[6]= 3.10723
```

Aufgabe 2 ▮ Problème 2

Schreibe eine eigene Version der "Join-Funktion". ▮ Ecris ta propre version de la fonction "Join".

"Join" fuegt zwei Listen hintereinander im Gegensatz zur Mengenvereinigung, wo mehrfach vorkommende Elemente nur einmal aufgefuehrt werden.
▮ "Join" rassemble deux listes l'une après l'autre. Dans la réunion des ensembles par contre les éléments doubles, triples etc. n'apparaissent qu'une seule fois.

```
In[7]:= Clear[joinFkt];
        joinFkt[{x__},{}]:= {x};
        joinFkt[{},{x__}]:= {x};
        joinFkt[{x__},{y__}]:= Flatten[{{x},{y}}]
```

Ausprobieren: ▮ Essayer:

```
In[11]:= joinFkt[{a,b,c,d},{e,f,g}]
Out[11]= {a, b, c, d, e, f, g}

In[12]:= joinFkt[{a,b,c,d},{a,e,b,f,g}]
Out[12]= {a, b, c, d, a, e, b, f, g}

In[13]:= joinFkt[{a,b,c,d},{}]
Out[13]= {a, b, c, d}

In[14]:= joinFkt[{},{}]
Out[14]= joinFkt[{}, {}]
```

Was ist passiert? Erweitere die Definition sinngemäss!
▮ Que s'est-il passé? Elargis la définition d'une façon sensée!

Aufgabe 3 ▮ Problème 3

Baue eine Funktion "Fold" wie unten beschrieben
▮ Construis une fonction "Fold" comme il est décris ci-dessous.

a) Implementiere wie nachstehend folgt. Beantworte dann die nachstehenden Fragen b), c), d) und vergleiche mit den implementierten Funktionen "Fold" resp. "FoldList".

▮ "Implement" comme il suit. Réponds aux questions suivantes b), c), d) et compare avec les fonctions "implémentées" "Fold" resp. "FoldList".

```
In[15]:= Clear[falte];
         falte::usage = "falte[f, basis, liste] ergibt
                     --donne  f[...[f[f[basis,x1]x2],...xn]
                     wobei --étant  list = {x1,x2,...,xn}";
         falte[f_,basis_,{}]:=basis;
         falte[f_,basis_,{x1_,xrest___}]:=
                falte[f,f[basis,x1],{xrest}];
```

b) Was macht "falte[Plus,0,liste]"? Z.B. liste={a,b,c}.

▮ Que fait "falte[Plus,0,liste]"? P.ex. liste={a,b,c}.

```
In[19]:= falte[Plus,0,{a,b,c}]
```

```
Out[19]= a + b + c
```

```
In[20]:= falte[Plus,0,{1,2,3,4}]
```

```
Out[20]= 10
```

```
In[21]:= falte[Plus,5,{1,2,3,4}]
```

```
Out[21]= 15
```

c) Was macht "falte[Max,-Infinity,liste]"? Z.B. liste={a,b,c}.

▮ Que fait "falte[Max,-Infinity,liste]"? P.ex. liste={a,b,c}.

```
In[22]:= falte[Max,-Infinity,{a,b,c}]
```

```
Out[22]= Max[a, b, c, -∞]
```

```
In[23]:= falte[Max,-Infinity,{6,8,3}]
```

```
Out[23]= 8
```

```
In[24]:= falte[Max,9,{6,8,3}]
```

```
Out[24]= 9
```

d) Vergleiche mit "Fold" und "FoldList":

▮ Compare avec "Fold" et "FoldList":

In[25]:= **??Fold**

```
Fold[f, x, list] gives the last element of FoldList[f, x, list]. Mehr...

Attributes[Fold] = {Protected}
```

In[26]:= **??FoldList**

```
FoldList[f, x, {a, b, ... }] gives {x, f[x, a], f[f[x, a], b], ... }. Mehr...

Attributes[FoldList] = {Protected}
```

Aufgabe 4 ▮ Problème 4

Schreibe die APL-Funktion "Deal" resp. "?" in *Mathematica*: ▮ Ecris la fonction APL "Deal" resp. "?" dans *Mathematica*:

"L ? R" in APL wählt (streicht) aus dem "Range[r]" pseudo-zufällig L Zahlen aus, ohne sie in R zu ersetzen.
▮ "L ? R" dans APL choisit (biffe) dans le "Range[r]" d'une façon pseudo-aléatoire L nombres, sans les remplacer dans R.

a) Was ist "Range" - Wie erzeuge ich die "Zufallszahlen"?
▮ Qu'est "Range" - Comment générér les "nombres aléatoires"?

In[27]:= **??Range**

```
Range[imax] generates the list {1, 2, ... , imax}. Range[imin, imax] generates
   the list {imin, ... , imax}. Range[imin, imax, di] uses step di. Mehr...

Attributes[Range] = {Listable, Protected}
```

In[28]:= **Range[14]**

Out[28]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

In[29]:= **??Random**

```
Random[ ] gives a uniformly distributed pseudorandom Real in the range
   0 to 1. Random[type, range] gives a pseudorandom number of the specified
   type, lying in the specified range. Possible types are: Integer, Real and
   Complex. The default range is 0 to 1. You can give the range {min, max}
   explicitly; a range specification of max is equivalent to {0, max}. Mehr...

Attributes[Random] = {Protected}
```

b) Die Beschreibung der Funktion "deal" in *Mathematica*:
▮ La description de la fonction "deal" dans *Mathematica*:

```
In[30]:= RemoveAll[deal];
         deal::usage="deal[n,r] streicht zufällig n Integers
                      aus der Liste  Range[r] ohne Ersetzung.
                      -- biffe de façon aléatoire n Integers
                      dans la liste Range[r] sans les
                      remplacer."
```

```
Out[31]= deal[n,r] streicht zufällig n Integers               aus der Liste
           Range[r] ohne Ersetzung.              -- biffe de façon aléatoire n
           Integers              dans la liste Range[r] sans les               remplacer.
```

c) Hier eine mögliche Variante:

■ Voici une variante possible:

```
In[32]:= Clear[deal];
         deal[0,r_] :=Range[r];
         deal[n_,r_]:=Sort[Rest[RotateLeft[deal[n-1,r],
                               Random[Integer,{1,r}]]]]
```

d) Ausprobieren:

■ Essaier

```
In[35]:= deal[2,6]
```

```
Out[35]= {1, 2, 4, 5}
```

e) Erklärungen:

■ Explications:

```
In[36]:= ??Rest
```

```
Rest[expr] gives expr with the first element removed. Mehr...

Attributes[Rest] = {Protected}
```

```
In[37]:= ??RotateLeft
```

```
RotateLeft[expr, n] cycles the elements in expr n positions to the left.
   RotateLeft[expr] cycles one position to the left. RotateLeft[expr, {n1,
   n2, ... }] cycles elements at successive levels ni positions to the left. Mehr...

Attributes[RotateLeft] = {Protected}
```

```
In[38]:= RotateLeft[{a,b,c},11]
```

```
Out[38]= {c, a, b}
```

```
In[39]:= ??Sort
```

```
Sort[list] sorts the elements of list into canonical
   order. Sort[list, p] sorts using the ordering function p. Mehr...

Attributes[Sort] = {Protected}
```

Aufgabe 5 ∎ Problème 5

Zu Funktionen, die als Argument eine variable Zahl von Listen aufnehmen: Quant aux fonctions qui acceptent comme argument un nombre variable de listes:

Das Muster "x___List" als Argument in einer Funktion lässt eine variable Zahl von Listen zu. Wie kann man auf einen Schlag alle Längen einer Anzahl beteiligter Listen finden? Schreibe eine Funktion dafür!
∎ Le patron "x___List" comme argument dans une fonction permet un nombre variable de listes. Comment peut-on, d'un coup, trouver toutes les longueurs d'un nombre de listes faisant partie du problème? Ecris une fonction pour cela!

```
In[40]:= Clear[listenLaengen];
         listenLaengen[x___List]:=Map[Length, {x}]
```

Ausprobieren: ∎ Essayer:

```
In[42]:= listenLaengen[{1,2,3}, {1,2,3,4,5}, {1,1,1,1},
                       {a,b,c,d,e,f,g}]
Out[42]= {3, 5, 4, 7}
```

Aufgabe 6 ∎ Problème 6

Lauflängencodierung in Listen (Häufigkeitszählungen) ∎ Comper des fréquences dans les listes

Konstruiere eine kurze *Mathematica*-Funktion, die die Häufigkeit der Elemente einer Liste zählt. Die Funktion "laufCod" soll eine Liste von Paaren {Listenelement, absolute Häufigkeit} ausgeben Beispiel:
liste = {e,b,a,b,e,c,c,d,e,a,a,d,u};
laufCod[liste] soll das Resultat {{a, 3}, {b, 2}, {c, 2}, {d, 2}, {e, 3}, {u, 1}} ergeben. Die Funktion "laufDeCod" soll aus dem Output von "laufCod" wieder die ursprüngliche Liste machen.
∎ Construis une fonction de *Mathematica* courte qui compte la fréquence des éléments d'une liste. La fonction "laufCod" doit sortir une liste de paires {élément de liste, fréquence absolue}. Exemple:
liste = {e,b,a,b,e,c,c,d,e,a,a,d,u};
laufCod[liste]doit donner le résultat
{{a, 3}, {b, 2}, {c, 2}, {d, 2}, {e, 3}, {u, 1}}. La fonction "laufDeCod" doit transformer l'output de "laufCod" de nouveau dans la liste d'origine.

a) laufCod

```
In[43]:= laufCod[liste_List]:=Transpose[{Union[liste],
                       Table[Count[liste,Part[Union[liste]
                       ,n]], {n,Length[Union[liste]]}]}]
```

(Hat Schweiss gekostet!) Aber jetzt ausprobieren:
∎ (Cela nous a fait transpirer!) Mais maintenant on essaie:

```
In[44]:= RemoveAll[liste];
         liste = {1,1,2,2,2,3,3,3,3};
         output = laufCod[liste]

Out[46]= {{1, 2}, {2, 3}, {3, 4}}

In[47]:= liste = {e,b,a,b,e,c,c,d,e,a,a,d,u};
         output = laufCod[liste]

Out[48]= {{a, 3}, {b, 2}, {c, 2}, {d, 2}, {e, 3}, {u, 1}}
```

a) laufDeCod

```
In[49]:= laufDeCod[output_]:= If[MatrixQ[output],
                     output/.{x_,n_}:> Flatten[
                                   Table[x,{n}]],                    Print["Bitte
                    Print["Bitte Matrix eingeben --
                           S.v.p. entrer la matrice"]]

In[50]:= laufDeCod[{{1,2},{3,5},{4,3}}]

Out[50]= {{1, 1}, {3, 3, 3, 3, 3}, {4, 4, 4}}

In[51]:= laufDeCod[{{a,3},{b,5},{c,2},{d,4}}]

Out[51]= {{a, a, a}, {b, b, b, b, b}, {c, c}, {d, d, d, d}}
```

c) Eine andere Variante für "laufCod" mit Ersetzungsregeln (funktioniert nur für eine Liste von Integers)

Une autre variante pour "laufCod" avec des règles de remplacement (ne fonctionne que pour une liste d'Integers)

```
In[52]:= RemoveAll[f];
         f[liste_List]:=Module[{i},
         i=0;
         a=liste;
         a//.{y___,x_,x_,z___}:>
         {y,x,++i b,x,z}//.{y___,x_Integer,w_Integer,z___}:>
         {y,x,d, c,w,z}/. {x___}:>{c, x, d}//.
         {x___,y___ b,z___}:>{x,z}//. {x___,b,z___}:>{x,z}//.
         {y___,c,x__,d,z___}:> {y,{x},z}
         ];
         laufCod1[liste_List]:={Map[First,f[liste]],
                                Map[Length,f[liste]]}

In[55]:= f[{1,1,1,2,2,2,2,2,3,3,3,3,4,5,5}]

Out[55]= {{1, 1, 1}, {2, 2, 2, 2, 2}, {3, 3, 3, 3}, {4}, {5, 5}}

In[56]:= laufCod1[{1,1,1,2,2,2,2,2,3,3,3,3,4,5,5}]

Out[56]= {{1, 2, 3, 4, 5}, {3, 5, 4, 1, 2}}

In[57]:= laufCod1[{0,0,0,1,1,2,2,2,3,3,3,3,4,4,4,5,5,6,6,6}]

Out[57]= {{0, 1, 2, 3, 4, 5, 6}, {3, 2, 3, 4, 3, 2, 3}}
```

Was ist los?

Que se passe-t-il?

```
In[58]:= laufCod1[{a,a,a,1,b,b,b,2,2,2,3,3,3,3,4,4,4,5}]

Out[58]= {{{0, 0, 0, d, {1, 1}, {2, 2, 2}, {3, 3, 3, 3}, {4, 4, 4}, {5, 5}, c, 6, 6, 6}, 3, 4, 5},
          {7, 4, 3, 1}}
```

d) Noch eine andere Variante:

■ Encore une autre variante:

d) Noch eine andere Variante für "laufCod" mit Hilfe anonymer Funktionen (vgl. Kap. 12):

■ Encore une autre variante pour "laufCod" à l'aide de fonctions anonymes (comparer chap. 12):

```
In[59]:= Attributes[mal]={Flat};
         mal[{elem_,m_},{elem_,n_}]:=mal[{elem,m+n}];
         laufCod2[liste_List]:= List @@ mal @@({#,1}& /@
                                liste)

In[62]:= laufCod2[{1,1,1,2,2,2,2,2,3,3,3,3,4,5,5}]

Out[62]= {{1, 3}, {2, 5}, {3, 4}, {4, 1}, {5, 2}}
```

Wie arbeitet dieses Programm?

■ Comment ce programme travaille-t-il?

steht für "anonyme Variable", & für "anonyme Funktion", /@ für "Map".
Durchlaufen wird die Liste.

■ # signifie "variable anonyme", & signifie "fonction anonyme", /@ signifie "Map". La liste est parcourue.

```
In[63]:= out1=({#,1}& /@ {1,1,1,2,2,2,2,2,3,3,3,3,4,5,5})

Out[63]= {{1, 1}, {1, 1}, {1, 1}, {2, 1}, {2, 1}, {2, 1}, {2, 1},
          {2, 1}, {3, 1}, {3, 1}, {3, 1}, {3, 1}, {4, 1}, {5, 1}, {5, 1}}
```

@@ steht für "Apply".

■ @@ signifie "Apply".

```
In[64]:= out2=mal @@out1

Out[64]= mal[{1, 3}, {2, 5}, {3, 4}, {4, 1}, {5, 2}]

In[65]:= ??List

        {e1, e2, ... } is a list of elements. Mehr...

        Attributes[List] = {Locked, Protected}

In[66]:= out3=List @@ out2

Out[66]= {{1, 3}, {2, 5}, {3, 4}, {4, 1}, {5, 2}}
```

Probiere eigene Varianten aus!

■ Essaie des variantes à toi!

Testgelände:
Terrain de test:

In[67]:=

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

In[68]:= **(* Old Form: Remove["Global`@*"] *)**

In[69]:= **Remove["Global`*"]**

Kurs ▮ Cours

12. Anonyme Funktionen
▮ Fonctions anonymes

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 12 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 12.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Anonyme Funktionen sind solche, die keinen Namen haben ....
Hier werden wir sehen, was es damit um sich hat!
▮ Les fonctions anonymes sont celles qui n'ont pas de nom...
Ici nous verrons ce qui en est d'elles!

12.1. "Function"
▮ "Function"

"Function" ist eine allgemeine Funktion ohne speziellen Namen, die gerade "unmittelbar" angewandt wird.
▮ "Function" est une fonction générale sans nom spécial, on peut l'utiliser "immédiatement".

Studiere: ▮ Etudie:

```
In[1]:= ??Function
```

```
Function[body] or body& is a pure function. The formal parameters are # (or #1), #2, etc.
   Function[x, body] is a pure function with a single formal parameter x. Function[
   {x1, x2, ... }, body] is a pure function with a list of formal parameters. Mehr...

Attributes[Function] = {HoldAll, Protected}
```

Beispiele: ▮ Exemple:

```
In[2]:= Function[x, x^2][4]

Out[2]= 16

In[3]:= Function[x, x^2 - x][5]

Out[3]= 20

In[4]:= Function[x, x^2][{1, 2, 3, 4, 5, 6, 7}]

Out[4]= {1, 4, 9, 16, 25, 36, 49}
```

Einfach, nicht? ▮ Simple, n'est-ce pas?

12.2. Anwendung auf Datenselektion ▮ Application à la sélection de données

12.2.1. Mit Hilfe einer klassischen Funktion mit Name (hier "groes45") ▮ A l'aide d'une fonction classique avec nom (ici "groes45")

Beispiel (erst Daten erzeugen, dann Funktion definieren, dann anwenden):
▮ Exemple (d'abord générer des données, ensuite définir la fonction, puis appliquer):

```
In[5]:= data = Table[Random[Integer, {0, 100}], {14}]

Out[5]= {33, 89, 77, 26, 3, 33, 66, 53, 95, 45, 49, 69, 15, 90}

In[6]:= groes45[x_] := x > 45

In[7]:= Map[groes45,data]

Out[7]= {False, True, True, False, False, False,
          True, True, True, False, True, True, False, True}

In[8]:= groes45[data]

Out[8]= {33, 89, 77, 26, 3, 33, 66, 53, 95, 45, 49, 69, 15, 90} > 45
```

Anwendung zur Selektion:
▮ Application à la sélection:

```
In[9]:= Select[data, groes45]

Out[9]= {89, 77, 66, 53, 95, 49, 69, 90}
```

12.2.2. Mit Hilfe einer anonymen Funktion
▮ A l'aide d'une fonction anonyme

Beispiel (erst ausprobieren):
▮ Exemple (essayer d'abord):

```
In[10]:= Function[x, x > 45][68]
Out[10]= True

In[11]:= Function[x, x > 45][24]
Out[11]= False
```

Anwendung zur Selektion:
▮ Application à la sélection:

```
In[12]:= Select[data, Function[x, x > 45]]
Out[12]= {89, 77, 66, 53, 95, 49, 69, 90}
```

12.2.3. Mit Hilfe der Abkürzung "#" für die Variable der anonymen Funktion
▮ A l'aide de l'abréviation "#" pour les variables de la fonction anonyme

```
In[13]:= Select[data, Function[# > 45]]
Out[13]= {89, 77, 66, 53, 95, 49, 69, 90}
```

12.2.4. Mit Hilfe der Abkürzung "( )&" für Function[ ]
▮ A l'aide de l'abréviation "( )&" pour Function[ ]

```
In[14]:= Select[data,  (# > 45)& ]
Out[14]= {89, 77, 66, 53, 95, 49, 69, 90}
```

12.3. Neuauflage einer früher definierten Funktion
▮ Nouvel emploi d'une fonction définie auparavant

Präzision 20 Stellen bei numerischen Rechnungen
▮ Précision de 20 unités lors de calculs numériques

Das Beispiel:
▮ L'exemple:

```
In[15]:= nZwanzig[x_] := N[x, 20]

In[16]:= $Post = nZwanzig

Out[16]= nZwanzig

In[17]:= Sin[1]

Out[17]= 0.84147098480789650665
```

Mit anonymer Funktion:
▮ Avec une fonction anonyme:

```
In[18]:= $Post := N[#, 40]&     (* vierzig *)

In[19]:= Sin[1]

Out[19]= 0.8414709848078965066525023216302989996226
```

12.4. Transformation von Wertepaaren in Regeln
▮ Transformation de paires de valeurs en règles

Aus zwei Listen {x1, x2, x3,... } und {y1, y2, y3,... } soll eine Liste {x1 - > y1, x2 -> y2, x3 -> y3,... } gemacht werden.
▮ Faire de deux listes {x1, x2, x3,... } et {y1, y2, y3,... } une seule {x1 - > y1, x2 -> y2, x3 -> y3,... }.

Listen erzeugen:
▮ Générer des listes de paires:

```
In[20]:= list1 = {x1, x2, x3, x4, x5, x6};
         list2 = {y1, y2, y3, y4, y5, y6};
```

Paarliste erzeugen:
Générer des listes de paires:

```
In[22]:= Transpose[{list1, list2}]

Out[22]= {{x1, y1}, {x2, y2}, {x3, y3}, {x4, y4}, {x5, y5}, {x6, y6}}
```

Anonyme Funktion (#[[1]] -> #[[2]])& auf die Liste setzen (mit "Map"):
■ Appliquer des fonctions anonymes (#[[1]] -> #[[2]])& à des listes (avec "Map"):

```
In[23]:= Map[(#[[1]] -> #[[2]])&, Transpose[{list1,
                                    list2}]]

Out[23]= {x1 → y1, x2 → y2, x3 → y3, x4 → y4, x5 → y5, x6 → y6}
```

12.5. Mehrere Argumente
■ Plusieurs arguments

Beispiel: ■ Exemple:

```
In[24]:= Function[{x, y}, x > y][1, 2]

Out[24]= False

In[25]:= Function[{x, y}, x > y][2, 1]

Out[25]= True

In[26]:= Function[{x, y}, x > y][u, 2]

Out[26]= u > 2.000000000000000000000000000000000000000
```

Was passiert, wenn die Anzahl der übergebenen Argumente nicht stimmt?
■ Que se passe-t-il quand le nombre des arguments transmis n'est pas correct?

```
In[27]:= Function[{x, y}, x > y][1]

        Function::fpct : Too many parameters in {x, y} to be filled from Function[{x, y}, x > y][1]. Mehr…

        Function::fpct : Too many parameters in {x, y} to be filled
           from Function[{x, y}, x > y][1.000000000000000000000000000000000000000]. Mehr…

Out[27]= Function[{x, y}, x > y][1.000000000000000000000000000000000000000]

In[28]:= Function[{x, y}, x > y][1, 2, 3]

Out[28]= False

In[29]:= Function[{x, y}, x > y][a, b, c]

Out[29]= a > b
```

Bei der Kurzschreibweise mit "&" und "#" kommt nun das Problem der Identifikation der verschiedenen Variablen. Die Variablen können dabei nummeriert werden. Beispiel:
■ Les abréviations "&" et "#" posent le problème de l'identification des différentes variables. On peut numéroter les variables. Exemple

```
In[30]:= ??#
```

```
# represents the first argument supplied to a pure function. #n represents the nth argument.
 Mehr…

Attributes[Slot] = {NHoldAll, Protected}
```

```
In[31]:= ?##
```

```
## represents the sequence of arguments supplied to a pure function. ##n represents the
   sequence of arguments supplied to a pure function, starting with the nth argument. Mehr…
```

```
In[32]:= ??&
```

```
Function[body] or body& is a pure function. The formal parameters are # (or #1), #2, etc.
   Function[x, body] is a pure function with a single formal parameter x. Function[
   {x1, x2, ... }, body] is a pure function with a list of formal parameters. Mehr…

Attributes[Function] = {HoldAll, Protected}
```

```
In[33]:= ?&&
```

```
e1 && e2 && ...  is the logical AND function. It evaluates its arguments in order, giving
   False immediately if any of them are False, and True if they are all True. Mehr…
```

```
In[34]:= (#1 < #2)&[4, 6]

Out[34]= True
```

Die anonyme Funktion ist jetzt (#1 < #2)&.

La fonction anonyme est maintenant (#1 < #2)&.

12.6. Daten filtern
Filtrer des données

Aufgabe
Problème

Ersetze eine Datenmenge nach einem "Filter" durch die Menge der gewichteten Mittelwerte je dreier sich folgender Elemente. Studiere dazu das folgende Beispiel!

Remplace un ensemble de données après un "Filtre" par l'ensemble des moyennes pondérées de trois éléments qui se suivent. Etudie l'exemple suivant!

Lösung in Einzelschritten:
Solution par étapes:

Zuerst Filter und Daten generieren:

Générer d'abord le filtre et les données:

```
In[35]:= filter = {0.1, 0.8, 0.1};
         datenF  = Table[Random[Integer, {0, 100}]/10,
                                        {10}]
```

```
Out[36]= {3.4000000000000000000000000000000000000000,
          8.8000000000000000000000000000000000000000,
          8.5000000000000000000000000000000000000000,
          1.8000000000000000000000000000000000000000,
          3.3000000000000000000000000000000000000000,
          0.9000000000000000000000000000000000000000,
          8.8000000000000000000000000000000000000000,
          7.0000000000000000000000000000000000000000,
          7.2000000000000000000000000000000000000000,
          6.3000000000000000000000000000000000000000}
```

Ordnen in Dreiergruppen, die dann gewichtet gemittelt werden können:

■ Ordonner en groupes de trois dont on peut faire ensuite la moyenne pondérée:

```
In[37]:= matrix = Partition[datenF, Length[filter], 1]
```

```
Out[37]= {{3.4000000000000000000000000000000000000000,
           8.8000000000000000000000000000000000000000,
           8.5000000000000000000000000000000000000000},
          {8.8000000000000000000000000000000000000000,
           8.5000000000000000000000000000000000000000,
           1.8000000000000000000000000000000000000000},
          {8.5000000000000000000000000000000000000000,
           1.8000000000000000000000000000000000000000,
           3.3000000000000000000000000000000000000000},
          {1.8000000000000000000000000000000000000000,
           3.3000000000000000000000000000000000000000,
           0.9000000000000000000000000000000000000000},
          {3.3000000000000000000000000000000000000000,
           0.9000000000000000000000000000000000000000,
           8.8000000000000000000000000000000000000000},
          {0.9000000000000000000000000000000000000000,
           8.8000000000000000000000000000000000000000,
           7.0000000000000000000000000000000000000000},
          {8.8000000000000000000000000000000000000000,
           7.0000000000000000000000000000000000000000,
           7.2000000000000000000000000000000000000000},
          {7.0000000000000000000000000000000000000000,
           7.2000000000000000000000000000000000000000,
           6.3000000000000000000000000000000000000000}}
```

```
In[38]:= % // MatrixForm
```

```
Out[38]//MatrixForm=
3.4000000000000000000000000000000000000000  8.8000000000000000000000000000000000000000
8.8000000000000000000000000000000000000000  8.5000000000000000000000000000000000000000
8.5000000000000000000000000000000000000000  1.8000000000000000000000000000000000000000
1.8000000000000000000000000000000000000000  3.3000000000000000000000000000000000000000
3.3000000000000000000000000000000000000000  0.9000000000000000000000000000000000000000
0.9000000000000000000000000000000000000000  8.8000000000000000000000000000000000000000
8.8000000000000000000000000000000000000000  7.0000000000000000000000000000000000000000
7.0000000000000000000000000000000000000000  7.2000000000000000000000000000000000000000
```

Die Gewichtung geschieht, indem man die erste Zahl einer Dreiergruppe mit der ersten Zahl des Filters multipliziert, die zweite Zahl des Filters mit der zweiten der Dreiergruppe etc. und das jeweils addiert. Diese Multiplikation mit nachfolgender Addition ist jeweils ein Skalarprodukt entsprechender Vektoren:
■ L pondération se fait en multipliant le 1er nombre d'un groupe de trois avec le 1er nombre du filtre, le 2e nombre du filtre avec le 2e nombre du groupe etc.. On en fait chaque fois l'addition. Cette multiplication suivie de l'addition est un produit scalaire des vecteurs correspondants:

```
In[39]:= Map[filter .#&, matrix]

Out[39]= {8.23, 7.86, 2.62, 2.91, 1.93, 7.83, 7.2, 7.09}
```

Lösung alles auf einmal:
■ Solution en un coup:

```
In[40]:= zusrollen[filterVar_List,
             dataVar_List]:= Map[filterVar . #&,
             Partition[dataVar, Length[filterVar], 1] ]

In[41]:= zusrollen[filter, datenF]

Out[41]= {8.23, 7.86, 2.62, 2.91, 1.93, 7.83, 7.2, 7.09}
```

"Putzmaschine" einsetzen
■ Employer la "machine de nettoyage"

```
In[42]:= (* Old Form: Remove["Global`@*"] *)

In[43]:= Remove["Global`*"]
```

Uebungen ▮ Exercices

12. Anonyme Funktionen
▮ Fonctions anonymes

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 12 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 12.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Leseübung: Schreibe benannte Funktionen, die dasselbe ausgeben wie die folgenden anonymen Funktionen:
▮ Exercice de lecture: Ecris des fonctions dénommées qui sortent la même chose que les fonctions anonymes suivantes:

(a) Anonym: (#^3)&
▮ Anonyme: (#^3)&

```
In[1]:= {(#^3)&[0],(#^3)&[1],(#^3)&[2],(#^3)&[3],(#^3)&[4]}

Out[1]= {0, 1, 8, 27, 64}
```

Benannt: ▮ Dénommé:

```
In[2]:= f[x_]:=x^3;
        {f[0],f[1],f[2],f[3],f[4]}

Out[3]= {0, 1, 8, 27, 64}
```

(b) Anonym: (#^#)&
▮ Anonyme: (#^#)&

```
In[4]:= {(#^#)&[0],(#^#)&[1],(#^#)&[2],(#^#)&[3],(#^#)&[4],
         (#^#)&[5]}

        Power::indet : Indeterminate expression 0^0 encountered. Mehr…

Out[4]= {Indeterminate, 1, 4, 27, 256, 3125}
```

Benannt: ■ Dénommé:

```
In[5]:= Remove[f];
        f[x_]:=x^x;
        {f[0],f[1],f[2],f[3],f[4],f[5]}

        Power::indet : Indeterminate expression 0^0 encountered. Mehr…

Out[7]= {Indeterminate, 1, 4, 27, 256, 3125}
```

Benannt: ■ Dénommé:

```
In[8]:= f[x_]:=x^3;
        {f[0],f[1],f[2],f[3],f[4]}

Out[9]= {0, 1, 8, 27, 64}

In[10]:=
```

(c) Anonym: {#,#^2}&

■ Anonyme: (#^#^2)&

```
In[10]:= {{#,#^2}&[0],{#,#^2}&[1],{#,#^2}&[2],{#,#^2}&[3],
          {#,#^2}&[4],{#,#^2}&[5]}

Out[10]= {{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
```

Benannt: ■ Dénommé:

```
In[11]:= Remove[f];
         f[x_]:={x,x^2};
         {f[0],f[1],f[2],f[3],f[4],f[5]}

Out[13]= {{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}}
```

(d) Anonym: If[#>0,#,-#]&

■ Anonyme: If[#>0,#,-#]&

```
In[14]:= {If[#>0,#,-#]&[-2],If[#>0,#,-#]&[-1],If[#>0,#,
         -#]&[0], If[#>0,#,-#]&[1],If[#>0,#,-#]&[2]}

Out[14]= {2, 1, 0, 1, 2}
```

Benannt: ■ Dénommé:

```
In[15]:= Remove[f];
         f[x_]:=Abs[x];
         {f[-2],f[-1],f[0],f[1],f[2],f[3]}

Out[17]= {2, 1, 0, 1, 2, 3}
```

(e) Anonym: (#/.x->y)&
■ Anonyme: (#/.x->y)&

```
In[18]:= {(#/.x->y)&[-2],(#/.x->y)&[-1],(#/.x->y)&[0],
          (#/.x->y)&[1],(#/.x->y)&[2],(#/.x->y)&[3],
          (#/.x->y)&[x],(#/.x->y)&[y],(#/.x->y)&[z]}

Out[18]= {-2, -1, 0, 1, 2, 3, y, y, z}
```

Benannt: ■ Dénommé:

```
In[19]:= Remove[f];
         f[z_]:=z /. x->y;
         {f[-2],f[-1],f[0],f[1],f[2],f[3],f[x],f[y],f[z]}

Out[21]= {-2, -1, 0, 1, 2, 3, y, y, z}
```

Aufgabe 2 ■ Problème 2

Schreibe eine anonyme Funktion, die die 3. Potenz des Arguments berechnet.
■ Ecris une fonction anonyme qui calcule la puissance 3 des arguments.

Zum Beispiel (#^3)&
■ Par exemple (#^3)&

```
In[22]:=
         {(#^3)&[-3],(#^3)&[-2],(#^3)&[-1],(#^3)&[0],
          (#^3)&[1],(#^3)&[2],(#^3)&[3]}

Out[22]= {-27, -8, -1, 0, 1, 8, 27}
```

Aufgabe 3 ■ Problème 3

Verwende "Select" sowie eine anonyme Funktion, um aus einer Liste von Paaren, diejenigen Paare herauszusuchen, bei denen die erste Zahl grösser ist als die zweite.
■ Utilise "Select" ainsi qu'une fonction anonyme pour choisir dans une liste de paires les paires dont le premier nombre est plus grand que le deuxième.

Ein Beispiel:
■ Un exemple:

```
In[23]:= newList = {{1,2},{2,1},{3,1},{2,3},{6,4},{4,7}};
         Select[newList,(#[[1]] > #[[2]])&]

Out[24]= {{2, 1}, {3, 1}, {6, 4}}
```

Aufgabe 4 ▮ Problème 4

Definiere eine anonyme Funktion, um die Option "PlotLabel" bei den plotgenerierenden Kommandos zu setzen. Bilde die Funktion ab auf ContourPlot, DensityPlot, ParametricPlot, Plot, Plot3D.
▮ Définis une fonction anonyme pour joindre l'option "PlotLabel" aux commandements qui génèrent des plots. "Applique" la fonction à ContourPlot, DensityPlot, ParametricPlot, Plot, Plot3D.

Was ist "PlotLabel"?
▮ Qu'est "PlotLabel"?

```
In[25]:= ??PlotLabel
```

```
PlotLabel is an option for graphics functions that specifies an overall label for a plot.
 Mehr…

Attributes[PlotLabel] = {Protected}
```

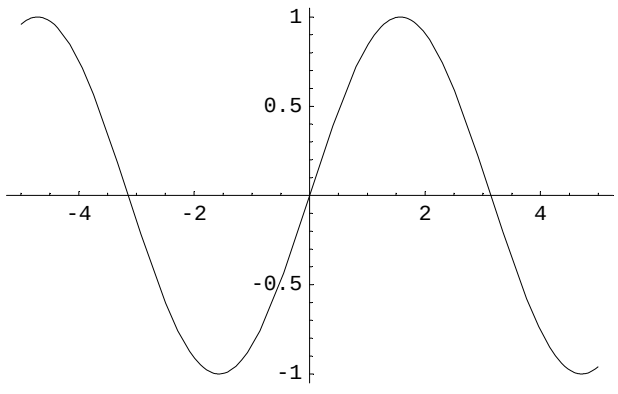
Wie wird es eingesetzt?
▮ Comment l'applique-t-on?

```
In[26]:= ??Plot
```

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
   xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. Mehr…

Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
  AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic, ColorOutput → Automatic,
  Compiled → True, DefaultColor → Automatic, DefaultFont :→ $DefaultFont,
  DisplayFunction :→ $DisplayFunction, Epilog → {}, FormatType :→ $FormatType, Frame → False,
  FrameLabel → None, FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
  ImageSize → Automatic, MaxBend → 10., PlotDivision → 30., PlotLabel → None,
  PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
  Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

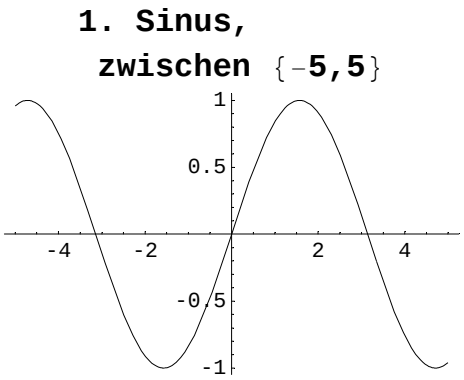
Ein Plot: ▮ Un plot:

```
In[27]:= Plot[Sin[x],{x,-5,5}];
```



Mit Label: ■ Avec Label:

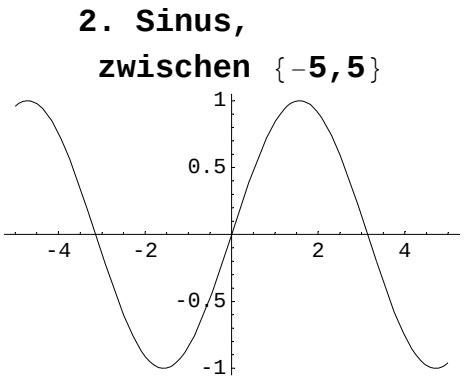
```
In[28]:= Plot[Sin[x],{x,-5,5},PlotLabel->FontForm[
         "1. Sinus,\n zwischen {-5,5}",
                        {"Courier-Bold",15}]];
```



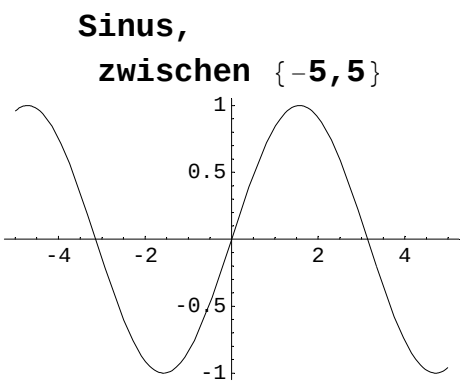
Mit Hilfe einer anonymen Funktion:

■ A l'aide d'une fonction anonyme:

```
In[29]:= (Plot[Sin[x],{x,-5,5},PlotLabel->#])&[FontForm[
         "2. Sinus,\n zwischen {-5,5}",
                        {"Courier-Bold",15}]];
```



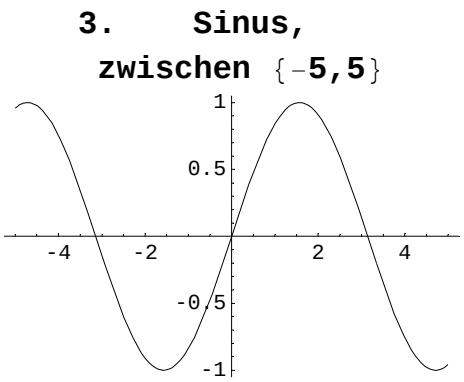

```
In[30]:= (Plot[Sin[x],{x,-5,5},PlotLabel->#])&[FontForm[
         "Sinus,\n zwischen {-5,5}",{"Courier-Bold",15}]];
```



Etwas geschraubter:

■ Un peu plus sophistiqué:

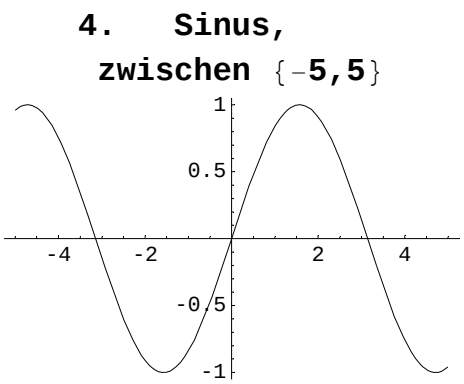
```
In[31]:= Apply[Plot,{#[[1]],#[[2]],#[[3]]}]&[{Sin[x],
           {x,-5,5},(PlotLabel->#)}&[FontForm["3.
           Sinus,\n zwischen {-5,5}",{"Courier-Bold",15}]]];
```



Noch etwas geschraubter:

■ Ecore plus sophistiqué:

```
In[32]:= Plot @@ {#[[1]],#[[2]],#[[3]]}&[{Sin[x],{x,-5,5},
          (PlotLabel->#)}&[FontForm["4.
          Sinus,\n zwischen {-5,5}",{"Courier-Bold",15}]]];
```



Ein Versuch, die Options zu ändern:

■ Un essai de changer les options:

```
In[33]:= Options[Plot]
```

```
Out[33]= {AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
          AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
          ColorOutput → Automatic, Compiled → True, DefaultColor → Automatic,
          DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction, Epilog → {},
          FormatType :→ $FormatType, Frame → False, FrameLabel → None, FrameStyle → Automatic,
          FrameTicks → Automatic, GridLines → None, ImageSize → Automatic,
          MaxBend → 10., PlotDivision → 30., PlotLabel → None, PlotPoints → 25,
          PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
          Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[34]:=
```
Unprotect[Options[Plot]];
Drop[Options[Plot],{11}];
Options[Plot]=Union[Options[Plot],{PlotLabel->
FontForm["5. Neues Bild",{"Courier-Bold",15}],
Frame->True}];
(*Protect[Plot]*)
```

```
Protect::pssl : Options[Plot] is not a string, symbol, or list of strings and symbols. Mehr…

Set::write : Tag Plot in Options[Plot] is Protected. Mehr…
```

In[38]:=
```
Attributes[Plot]
```

Out[38]=
```
{HoldAll, Protected}
```

In[39]:=
```
Options[Plot]
```

Out[39]=
```
{AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
 AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
 ColorOutput → Automatic, Compiled → True, DefaultColor → Automatic,
 DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction, Epilog → {},
 FormatType :→ $FormatType, Frame → False, FrameLabel → None, FrameStyle → Automatic,
 FrameTicks → Automatic, GridLines → None, ImageSize → Automatic,
 MaxBend → 10., PlotDivision → 30., PlotLabel → None, PlotPoints → 25,
 PlotRange → Automatic, PlotRegion → Automatic, PlotStyle → Automatic,
 Prolog → {}, RotateLabel → True, TextStyle :→ $TextStyle, Ticks → Automatic}
```

In[40]:=
```
Unprotect[Plot];
Drop[Options[Plot],{11}];
Options[Plot]=Union[Options[Plot],{PlotLabel->
FontForm["5. Neues Bild",{"Courier-Bold",15}],
Frame->True}];
(*Protect[Plot]*)
```

In[44]:=
```
Attributes[Plot]
```

Out[44]=
```
{HoldAll}
```

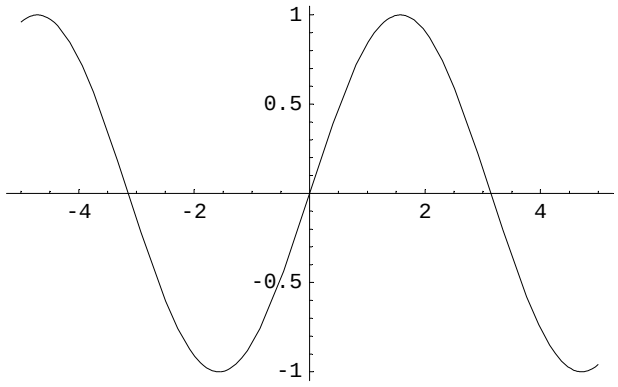
In[45]:=
```
Options[Plot]
```

Out[45]=
```
{AspectRatio → 1/GoldenRatio, Axes → Automatic, AxesLabel → None,
 AxesOrigin → Automatic, AxesStyle → Automatic, Background → Automatic,
 ColorOutput → Automatic, Compiled → True, DefaultColor → Automatic,
 Epilog → {}, Frame → False, Frame → True, FrameLabel → None,
 FrameStyle → Automatic, FrameTicks → Automatic, GridLines → None,
 ImageSize → Automatic, MaxBend → 10., PlotDivision → 30., PlotLabel → None,
 PlotLabel → FontForm[5. Neues Bild, {Courier-Bold, 15}],
 PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic,
 PlotStyle → Automatic, Prolog → {}, RotateLabel → True, Ticks → Automatic,
 DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction,
 FormatType :→ $FormatType, TextStyle :→ $TextStyle}
```

Was hat er nicht getan?

■ Que n'a-t-il pas fait?

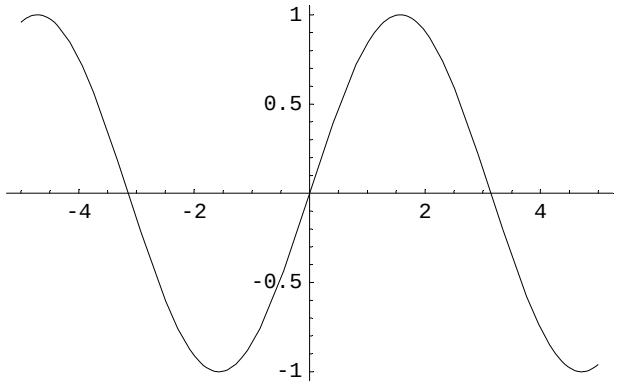
In[46]:= **Plot[Sin[x],{x,-5,5}];**



Wieso kam der Titel nicht?

■ Pourquoi le titre n'est-il pas apparu?

In[47]:= **Show[%];**



Uebungsfeld: ■ Terrain d'exercice:

In[48]:=

Aufgabe 5 ■ Problème 5

Schreibe nochmals eine Funktion, die die Häufigkeit eines Elements in einer Liste zählt.
■ Ecris encore une fonction qui compte la fréquence d'un élément dans une liste.

Verwende Union, Count und Map.

■ Utilise Union, Count et Map.

Diese Funktion zählt die Häufigkeit von verschiedenen Elementen
einer Liste:

■ Cette fonction compte la fréquence de différents éléments d'une liste:

```
In[48]:= frequenz[x_List]:= Module[{elemente=Union[x]},
                    Map[{#,Count[x,#]}&,elemente]];
         frequenz::usage="frequenz[x_List] gibt eine Liste \
                    von verschiedenen Elementen zusammen
                    mit ihrer Häufigkeit ---- \
                    donne une liste avec différents \
                    éléments avec leurs fréquences.";
         testListe = Table[Random[Integer,{1,20}],{18}];
         Sort[testListe]

Out[51]= {2, 3, 3, 6, 7, 8, 8, 9, 10, 15, 15, 16, 16, 18, 18, 19, 19, 19}

In[52]:= frequenz[testListe]

Out[52]= {{2, 1}, {3, 2}, {6, 1}, {7, 1}, {8, 2},
          {9, 1}, {10, 1}, {15, 2}, {16, 2}, {18, 2}, {19, 3}}

In[53]:= ??frequenz

        frequenz[x_List] gibt eine Liste    von verschiedenen Elementen zusammen    mit ihrer
          Häufigkeit ----    donne une liste avec différents    éléments avec leurs fréquences.

        frequenz[x_List] := Module[{elemente = Union[x]}, ({#1, Count[x, #1]} &) /@ elemente]
```

Probiere eigene Listen!

■ Essaie des listes à toi!

```
In[54]:=
```

Aufgabe 6 ■ Problème 6

Graphisches ■ Graphiques

Eine Aufgabe in mehreren Schritten

■ Un problème en plusieurs étapes

(a) Generiere eine Liste von 10 Pseudo-Zufallspunkten im Raum! ■ Génère une liste de 10 points pseudo-aléatoires dans l'espace!

```
In[54]:= punkte = Partition[
                  Table[Random[Integer,{1,20}],{30}],3]

Out[54]= {{9, 6, 10}, {11, 17, 18}, {7, 11, 12}, {6, 4, 12}, {7, 11, 9},
          {9, 1, 10}, {16, 11, 15}, {6, 15, 11}, {9, 1, 16}, {17, 9, 3}}
```

(b) Generiere eine Liste von 10 Pseudo-Zufallszahlen für das Grau-Niveau!

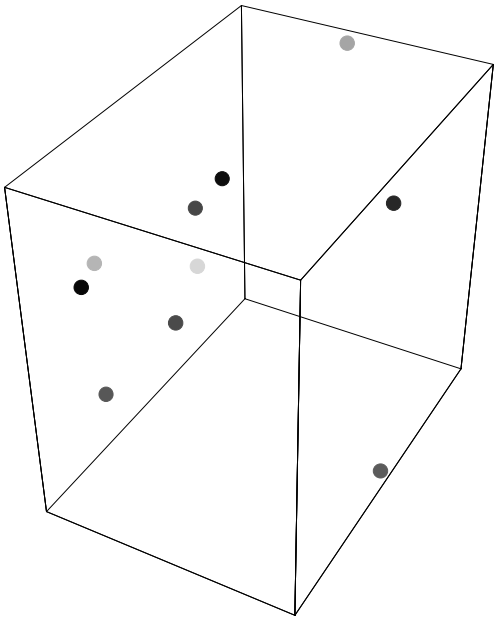
Génère une liste de 10 points pseudo-aléatoires pour le niveau gris!

```
In[55]:= grau = Table[Random[Real,{0,1}],{10}]

Out[55]= {0.292196, 0.647111, 0.279979, 0.029356,
          0.842525, 0.343356, 0.155026, 0.0516282, 0.709496, 0.358394}
```

(c)

```
In[56]:= grau1=Partition[grau,1]

Out[56]= {{0.292196}, {0.647111}, {0.279979}, {0.029356}, {0.842525},
          {0.343356}, {0.155026}, {0.0516282}, {0.709496}, {0.358394}}

In[57]:= trans=Transpose[{punkte,grau1}];trans

Out[57]= {{{9, 6, 10}, {0.292196}}, {{11, 17, 18}, {0.647111}},
          {{7, 11, 12}, {0.279979}}, {{6, 4, 12}, {0.029356}}, {{7, 11, 9}, {0.842525}},
          {{9, 1, 10}, {0.343356}}, {{16, 11, 15}, {0.155026}},
          {{6, 15, 11}, {0.0516282}}, {{9, 1, 16}, {0.709496}}, {{17, 9, 3}, {0.358394}}}

In[58]:= RemoveAll[mix];
         mix=Map[Join[#[[1]],#[[2]]]&,trans]

Out[59]= {{9, 6, 10, 0.292196}, {11, 17, 18, 0.647111},
          {7, 11, 12, 0.279979}, {6, 4, 12, 0.029356}, {7, 11, 9, 0.842525},
          {9, 1, 10, 0.343356}, {16, 11, 15, 0.155026},
          {6, 15, 11, 0.0516282}, {9, 1, 16, 0.709496}, {17, 9, 3, 0.358394}}

In[60]:= menge=Map[({PointSize[0.03],GrayLevel[#[[4]]],
                     Point[{#[[1]],#[[2]],#[[3]]}]
                     })&,
                     mix];
         menge

Out[61]= {{PointSize[0.03], GrayLevel[0.292196], Point[{9, 6, 10}]},
          {PointSize[0.03], GrayLevel[0.647111], Point[{11, 17, 18}]},
          {PointSize[0.03], GrayLevel[0.279979], Point[{7, 11, 12}]},
          {PointSize[0.03], GrayLevel[0.029356], Point[{6, 4, 12}]},
          {PointSize[0.03], GrayLevel[0.842525], Point[{7, 11, 9}]},
          {PointSize[0.03], GrayLevel[0.343356], Point[{9, 1, 10}]},
          {PointSize[0.03], GrayLevel[0.155026], Point[{16, 11, 15}]},
          {PointSize[0.03], GrayLevel[0.0516282], Point[{6, 15, 11}]},
          {PointSize[0.03], GrayLevel[0.709496], Point[{9, 1, 16}]},
          {PointSize[0.03], GrayLevel[0.358394], Point[{17, 9, 3}]}}
```

In[62]:= **Show[Graphics3D[menge]];**



Einige Erklärungen:

■ Quelques explications:

In[63]:= **??Show**

```
Show[graphics, options] displays two- and three-dimensional graphics, using
   the options specified. Show[g1, g2, ... ] shows several plots combined. Mehr...

Attributes[Show] = {Protected}
```

In[64]:= **??Graphics3D**

```
Graphics3D[primitives, options] represents a three-dimensional graphical image. Mehr...

Attributes[Graphics3D] = {Protected, ReadProtected}

Options[Graphics3D] =
 {AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → False, AxesEdge → Automatic,
  AxesLabel → None, AxesStyle → Automatic, Background → Automatic, Boxed → True,
  BoxRatios → Automatic, BoxStyle → Automatic, ColorOutput → Automatic, DefaultColor → Automatic,
  DefaultFont :→ $DefaultFont, DisplayFunction :→ $DisplayFunction, Epilog → {},
  FaceGrids → None, FormatType :→ $FormatType, ImageSize → Automatic, Lighting → True,
  LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]},
    {{0., 1., 1.}, RGBColor[0, 0, 1]}}, Plot3Matrix → Automatic, PlotLabel → None,
  PlotRange → Automatic, PlotRegion → Automatic, PolygonIntersections → True,
  Prolog → {}, RenderAll → True, Shading → True, SphericalRegion → False,
  TextStyle :→ $TextStyle, Ticks → Automatic, ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

In[65]:= **??DefaultColor**

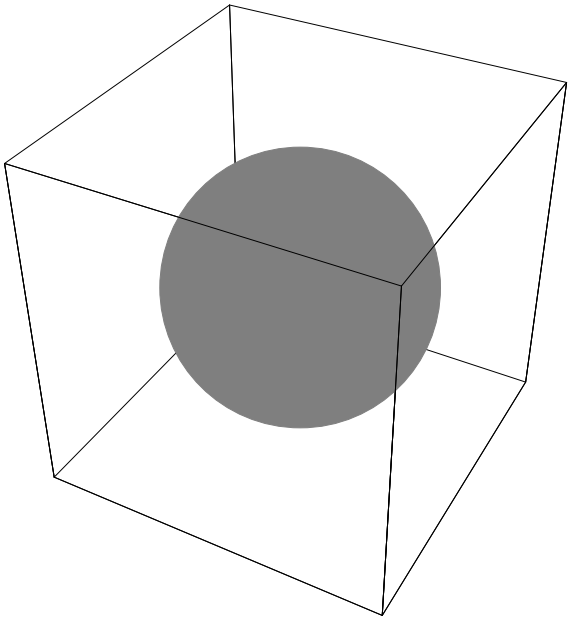
```
DefaultColor is an option for graphics functions
   which specifies the default color to use for lines, points, etc. Mehr...

Attributes[DefaultColor] = {Protected}
```

In[66]:= **??GrayLevel**

GrayLevel[level] is a graphics directive which specifies the gray-level intensity with which objects that follow should be displayed. Mehr…

Attributes[GrayLevel] = {Protected}

In[67]:=
```
{PlotStyle->GrayLevel[1]};
Show[Graphics3D[{PointSize[0.5],GrayLevel[0.5],
Point[{1,1,1}]
}]];
```



In[69]:= **??#**

represents the first argument supplied to a pure function. #n represents the nth argument. Mehr…

Attributes[Slot] = {NHoldAll, Protected}

Aufgabe 7 ▮ Problème 7

Das Problem deines "signifikanten Partners" ▮ Le problème de ton "partenaire significatif"

Hier geht es um die Analyse einer besondern Strategie zum Auffinden Deines "signifikanten Partners". Angenommen, da seinen n Personen des andern Geschlechts, die Du in Betracht ziehen willst. Du selbst bist noch ledig, jedoch heiratswillig. Jeder der n Personen ist ein unterscheidbarer Wert zugewiesen, den Du nicht kennst, bis Du das Individuum triffst. Jede Person kannst Du nur einmal treffen. Nach jedem Treffen musst Du entscheiden, ob das eben getroffene Individuum Dein "signifikanter Partner" ist und ihm dann einen Heiratsantrag stellen. Im Falle der Ablehnung gehst Du zm nächsten Treffen, wo dieselben Regeln gelten. Eine abgelehnte Person kannst Du nicht nochmals treffen, entsprechend den hier geltenden Gepflogenheiten.
Du sollst jetzt folgende Strategie analysieren: Du triffst (n/e) Partner als Testmenge, e = Eulersche Zahl. Nachdem Du jetzt einen Eindruck gewonnen hast, wählst Du den ersten Partner Deines Wunsches, der Dir besser gefällt als alle bisherigen Partner.
Wähle n = 30 und lasse eine Simulation 50 mal laufen. Bezeichne im voraus einen besten Partner. Ein Lauf sei positiv, wenn Du den vorher bezeichneten besten Partner durch das Verfahren auch findest.
▮
Il s'agit ici d'une stratégie spéciale pour trouver ton "partenaire significatif". Supposons n personnes de l'autre sexe que tu veux considérer. Toi-même tu es encore célibataire, mais décidé de te marier. A chacune des n personnes est attribuée une valeur qu'on peut distinguer, que tu ne connais pas, jusqu'à ce que tu rencontres l'individu. Tu ne peux rencontrer chaque personne qu'une seule fois. après chaque rencontre tu dois décider si l'individu que tu viens de rencontrer est ton "partenaire significatif" et lui faire ensuite une proposition de mariage. En cas de refus tu te rends au prochain rendez-vous, où valent les mêmes règles. Une personne refusée tu ne peux plus la rencontrer, selon les us et coutumes en vigeur ici.
Analyse la stratégie suivante: Tu rencontres (n/e) partenaires comme ensemble de test, e = nombre d'Euler. Après cette première impression tu choisis le partenaire qui te plaît le mieux que tous les précédents. Choisis n = 30 et fais marcher la simulation 50 fois. Désigne à l'avance le meilleur partenaire. Un parcours soit positif si tu trouves le partenaire désigné à l'anvance par cette méthode.

(a) Berechne den prozentualen Anteil der positiven Läufe. ▮ Calcule la partie proportionelle des parcours positifs:

Definiere zuerst eine Funktion "zufall", die eine Liste von Leuten in zufälliger Reihenfolge wieder ausgibt.
▮ Définis d'abord une fonction "hasard", qui sort une liste de personnes en ordre aléatoire.

Definiere zuerst die Funktion "zufall1". Diese Funktion stellt die Liste rekursiv zufällig um.
▮ Définis d'abord la fonction "zufall1" ("hasard"). Cette fonction redonne la liste de façon récursive aléatoire.

```
In[70]:= zufall1[{}]:={};
         zufall1[personen_List]:=
           Block[{n=Random[Integer,{1,Length[personen]}]},
             Prepend[zufall1[Drop[personen,{n}]],
             personen[[n]]]];
```

Einige Details zum Test:
▮ Quelques détails quant au test:

```
In[72]:= personen={a,b,c,d};
         zufall1[personen]

Out[73]= {a, b, d, c}

In[74]:= n=Random[Integer,{1,Length[personen]}]

Out[74]= 2

In[75]:= n=Random[Integer,{1,Length[personen]}];
         Drop[personen,{n}]

Out[76]= {a, b, c}

In[77]:= zufall1[Drop[{a,b,c,d},{2}]]

Out[77]= {a, c, d}
```

Oder definiere dann die Funktion "zufall2". Diese Funktion tut dasselbe wie "zufall1":

■ Ou bien définis la fonction "zufall2". Cette fonction fait la même chose que "zufall2":

```
In[78]:= zufall2[{}]:={};
         zufall2[personen_List]:=
          Block[{n=Random[Integer,{0,Length[personen]-1}],
                         shifted},
                 shifted=RotateLeft[personen,n];
                 Prepend[zufall2[Drop[shifted,1]],
                 shifted[[1]]]];
```

Einige Details zum Test:

■ Quelques détails quant au test:

```
In[80]:= personen={a,b,c,d};
         zufall2[personen]

Out[81]= {d, a, c, b}

In[82]:= n=Random[Integer,{0,Length[personen]-1}]

Out[82]= 3

In[83]:= n=Random[Integer,{0,Length[personen]-1}];
         shifted=RotateLeft[personen,n]

Out[84]= {b, c, d, a}

In[85]:= n=Random[Integer,{0,Length[personen]-1}];
         shifted=RotateLeft[personen,n];
         Drop[shifted,1]

Out[87]= {a, b, c}
```

Oder definiere dann die Funktion "zufall" mit Hilfe der Funktion "Sort".
Hier werden Paare {a,b} benützt, wobei a eine Zufallszahl und b die
Ordnungszahl der Person ist. Die Paare werden dann nach a geordnet. Diese Funktion tut dasselbe wie "zufall1":

■ Ou bien définis la fonction "zufall" à l'aide de la fonction "Sort". Ici on emploie des paires {a,b}, a étant le nombre aléatoire, b le nombre ordinal de la personne. Les paires sont rangées selon a. Cette fonction fait la même chose que "zufall1":

```
In[88]:= zufall[{}]:={};
         zufall[personen_List]:=
             Map[Last,
                 Sort[Map[{Random[], #}&, personen],
                      (#1[[1]] > #2[[1]])&
                 ]];
         zufall::usage="zufall[list] gibt eine zufällige \
                     Permutation der Elemente der Liste \
                     --- donne une permutation aléatoire \
                     des éléments de la liste."

Out[90]= zufall[list] gibt eine zufällige    Permutation der Elemente der
           Liste    --- donne une permutation aléatoire    des éléments de la liste.
```

Einige Details zum Test:

Quelques détails quant au test:

```
In[91]:= personen={a,b,c,d};
         zufall[personen]

Out[92]= {c, d, a, b}

In[93]:= {Random[], #}&[personen]

Out[93]= {0.0532293, {a, b, c, d}}

In[94]:= Map[{Random[], #}&, personen]

Out[94]= {{0.0669755, a}, {0.648717, b}, {0.148611, c}, {0.232326, d}}

In[95]:= Sort[Map[{Random[], #}&, personen],
                      (#1[[1]] > #2[[1]])&]

Out[95]= {{0.494212, c}, {0.463157, a}, {0.397076, d}, {0.113987, b}}

In[96]:= Map[Last, Sort[Map[{Random[], #}&, personen],
                      (#1[[1]] > #2[[1]])& ]]


Out[96]= {a, c, b, d}
```

Gib nun den Personen statt Namen Ordnungszahlen von 1 bis n. 1 bedeutet "am meisten bevorzugt" etc..

Donne aux personnes au lieu de noms des nombres ordinaux de 1 jusqu'à n. 1 signifie "préféré le plus" etc..

```
In[98]:= n = 30;
         personen = zufall[Range[n]]

Out[99]= {24, 2, 18, 27, 4, 11, 1, 30, 5, 9, 15, 8, 13, 10,
          12, 29, 3, 16, 23, 25, 21, 19, 17, 6, 26, 22, 7, 20, 14, 28}
```

Die Funktion "wahlVorzug" soll nun n Personen nehmen und diese in zufällige Reihenfolge bringen. Sie soll die ersten (n/e) (gerundet) Personen nehmen und den Wert der meist bevorzugten Person in dieser Menge in die Variable mmin speichern. Der Wert der ersten noch mehr bevorzugten Person, die dann folgt, soll in die Variable wahl gespeichert werden. Wenn wahl eine Zahl grösser gleich 1 enthält, soll wahlVorzug diese Zahl als Element einer Menge ausgeben. Andernfalls soll die leere Menge kommen.

La fonction "wahlVorzug" prenne maintenant n personnes et les mette dans un ordre aléatoire. Qu'elle prenne les premières (n/e) (arrondi) personnes et qu'elle mémorise dans la variable mmin la valeur de la personne préférée de cet ensemble. La valeur de la personne préférée encore davantage, qui suit, doit être mémorisée dans la variable wahl. Si

wahl contient un nombre plus grand ou égal 1, wahl doit sortir ce nombre comme élément d'un ensemble. Autrement doit apparaître l'ensemble vide.

In[100]:=
```
wahlVorzug[n_Integer]:=
        Module[{personen=zufall[Range[n]],
        m=Round[N[n/E]],
        mmin, wahl},
        mmin=Min[Take[personen,m]];
        wahl=Select[personen,(#<mmin)&];
        If[Length[wahl]>0,First[wahl],{}]];
anzahlPersonen=30;
anzahlLauf=50;
resultat=Table[wahlVorzug[anzahlPersonen],
        {anzahlLauf}]
```

Out[103]=
```
{{}, 1, {}, {}, 1, 1, 2, 1, 1, {}, {}, {}, 3, {}, {}, 1, 2, 1, {}, 4, {}, {}, {}, {}, {},
 6, 1, {}, {}, 3, 2, {}, 1, 1, {}, 1, 2, 3, {}, {}, 2, 1, {}, {}, 1, 4, 1, {}, 1, 1}
```

Anzahl positive Simulationen mit Resultat 1, 2 oder 3:

Nombre de simulations avec le résultat 1, 2 ou 3:

In[104]:=
```
posSim1={Map[(N[Count[resultat,#]/anzahlLauf])&,
                                    {1,2,3}]}
```

Out[104]=
```
{{0.32, 0.1, 0.06}}
```

Kürzer mit Vergleich mit (1/e):

Plus brièvement avec la comparaison avec (1/e):

In[105]:=
```
posSim1={(N[Count[resultat,#]/anzahlLauf
                            ])&/@{1,2,3},N[1/E]}
```

Out[105]=
```
{{0.32, 0.1, 0.06}, 0.367879}
```

(b) Prozentualer Anteil der Läufe, die mit einem signifikanten Partner enden:

Partie propositionelle des parcours qui terminent par un partenaire significatif:

Studiert man den Output, so sieht man, dass meistens die Anzahl der Fälle, in denen die leere Menge herauskommt, sehr gering ist. Dann hat die Strategie keinen Erfolg. Falls die Strategie eine Zahl ergibt, so wollen wir von Erfolg insofern reden, dass dann eine kleine Nummer erwartet werden kann. Wir sagen dann, es hätte sich ein "signifikanter Partner" finden lassen. Wie oft kommt das prozentual vor?

Si on étudie l'output, on voit que pour la plupart le nombre des cas où sort l'ensemble vide et très petit. Dans ces cas la stratégie n'a pas de succès. Si la stratégie donne un nombre, nous pouvons nous attendre à un nombre bas. Nous dirons qu'un "partenaire significatif" a pu être trouvé. Dans quel pourcentage cela arrive-t-il?

In[106]:=
```
signifikant = 1- N[Count[resultat,{}]/50]
```

Out[106]=
```
0.54
```

c) Wie gut oder schlecht war die Auswahl nun?
▮ Combien le choix était donc bon ou mauvais?

Wir berechnen, in wievielen Läufen prozentual die Simulation mit einer Person endete, die unter "den favorisierten 10%" einzureihen ist.

▮ Nous calculons dans combien de parcours la simulation s'est terminée par une personne qu'on a pu placer parmi les "10 % facorisés".

In[107]:=
```
signifikant10Prozent =
        N[Length[Select[resultat,
        (#<=Round[N[anzahlPersonen 0.1]])&]]/50]
```

Out[107]=
```
0.48
```

(d) Bei einer grösseren Laufzahl:
▮ Lors d'un nombre supérieur de parcours:

In[108]:=
```
anzahlLauf=50;
resultat=Table[wahlVorzug[anzahlPersonen],
                    {anzahlLauf}];
posSim1={Map[(N[Count[resultat,#]/anzahlLauf])&,
                    {1,2,3}]};
signifikant = 1- N[Count[resultat,{}]/50];
signifikant10Prozent =
        N[Length[Select[resultat,
        (#<=Round[N[anzahlPersonen 0.1]])&]]/50];
Print["resultat = ", resultat];
Print["posSim1              = ", posSim1];
Print["signifikant          = ", signifikant];
Print["signifikant10Prozent = ",
                            signifikant10Prozent]
```

```
resultat = {2, 1, {}, 1, {}, {}, {}, 5, 2, {}, 1, 1, {}, {}, 1, {}, 1, 1, {}, {}, 1, 8, {},
  {}, 1, 1, 3, {}, 1, 1, 1, {}, 1, 6, 1, 2, {}, 2, {}, 1, {}, {}, 1, 3, 2, {}, {}, 1, 1, 2}

posSim1     = {{0.38, 0.12, 0.04}}

signifikant    = 0.6

signifikant10Prozent = 0.54
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[117]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[118]:=

```
Remove["Global`*"]
```

Kurs ▮ Cours

13. Fallstricke und "Debugging" (Fehler eliminieren)
▮ Embûches et "Debugging" (éliminer les erreurs)

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 13 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 13.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

13.1. Fehlermeldungen
▮ Annonces d'erreurs

Beispiel: ▮ Exemples:

```
In[1]:= 1 + 2 = 3

        Set::write : Tag Plus in 1 + 2 is Protected. Mehr…

Out[1]= 3
```

Hier kommt eine Fehlermeldung. Die Fehlermeldungen sind im Technical Report Number 9 ff von Wolfram Research beschrieben. Achtung! Orthographische, syntaktische Fehler (programmlogische Fehler, Durchbrechung der Programmierregeln) können entdeckt werden, jedoch nicht semantische Fehler (falsche Interpretation der eigenen Absicht in der Niederschrift).
▮ Voici une annonce d'erreurr.(Les annonces d'erreurs sont décrites dans le Technical Report Number 9 ff de Wolfram Research.) Attention! Erreurs d'orthographe ou de syntaxe (erreurs concernant la logique de programmation ou les règles de programmation) peuvent être découvertes, tandis que les erreurs sémantiques (fausse interprétation de la propre intention dans le manscrit) ne le peuvent pas.

Beispiel: ▮ Exemple:

In[2]:= **2 -)5 + 7**

```
Syntax::bktx: "2 - " has extra ")" after it. Mehr…
```

Hinweis zum Notebook Front End: Mit (<Command>-B (B gross) --- NeXT) <Shift Control> B (B gross) ist eine Klammerüberprüfung möglich. Probiere das aus (setze den Cursor in die Lücke und verwende (<Command>-B (B gross) --- NeXT) <Shift Control> B (B gross)):

■ Indication au sujet de Notebook Front End: Avec (<Command>-B (B grand) --- NeXT) <Shift Control> B (B grand) il est possible de faire un examen des parenthèses. Essaie (met le cursor dans l'espace vide et emploie (<Command>-B (B grand) --- NeXT) <Shift Control> B (B grand)):

In[2]:= **((([[cjklf]r]}}]**

```
Syntax::bktmch:
   "([ [cjklf]r]" must be followed by ")", not "}". Mehr…
```

13.2. Fehler durch vordefinierte Variablen
■ Erreur par des variables définies d'avance

Beispiel: ■ Exemple:

Wird einer Variablen ein Wert zugewiesen, so wird auf den Wert und nicht auf die Variable gegriffen:

■ Quand une valeur est assignée à une variable, c'est la valeur qui est prise et non pas la variable:

In[2]:=
```
t = 5;
Print["t = ", t];
Integrate[Sin[t], t]
```

```
t = 5

Integrate::ilim : Invalid integration variable or limit(s) in 5. Mehr…
```

Out[4]= $\int \text{Sin}[5]\, d5$

Abhilfe: Z.B. mit lokalen Variablen arbeiten.

■ Secours: P.ex. travailler avec des variables locales.

13.3. Unvollständige Befehle
■ Ordres incomplets

13.3.1.Fehlende oder nicht korrekte Namen
■ Noms qui manquent ou qui ne sont pas corrects.

Studiere: ■ Etudie:

In[5]:= **D[x^3]**

Out[5]= x^3

```
In[6]:= D[x^3, x]
```

$$3\,x^2$$

```
In[7]:= D[x^3, y]
```

```
Out[7]= 0
```

```
In[8]:= D[x^3, 1]
```

```
General::ivar : 1 is not a valid variable. Mehr…
```

$$\partial_1 x^3$$

```
In[9]:= D[x^3 Sin[y], x, y]
```

$$3\,x^2\,\text{Cos}[y]$$

D[x^3] ergibt offenbar die 0-te Ableitung von x^3.
■ D[x^3] donne apparemment la 0-ième dérivation de x^3.

13.3.2.Fehlende oder nicht korrekte Punktuation
■ Ponctuation qui manque ou qui n'est pas correcte

Studiere: ■ Etudie:

```
In[10]:= Clear[f];
         f[x_] := If[x > 0,  x -x];
         {f[5], f[-5]}
```

```
Out[12]= {0, Null}
```

Mit Komma:
■ Avec virgule:

```
In[13]:= Clear[f];
         f[x_] := If[x > 0,  x, -x];
         {f[5], f[-5]}
```

```
Out[15]= {5, 5}
```

Studiere: ■ Etudie:

```
In[16]:= Clear[f];
         f[x_, y_] := xy;
         f[2, 3]
```

```
Out[18]= xy
```

```
In[19]:= Clear[f];
         f[x_, y_] := x y;
         f[2, 3]
```

```
Out[21]= 6
```

```
In[22]:= D[x^3 Sin[y], x, y]
```

$$3\,x^2\,\text{Cos}[y]$$

13.4. Falsche Klammerung
■ Erreurs de parenthèses

Beispiel: ■ Exemple:

```
In[23]:= Clear[g]

In[24]:= g[0]

Out[24]= g[0]

In[25]:= g(0)

Out[25]= 0

In[26]:= g(a)

Out[26]= a g
```

13.5. Verwechslung von Zeilen- und Spaltenvektor
■ Confusion entre vecteur de ligne et vecteur de colonnes

Skalarprodukt
■ Produit scalaire

```
In[27]:= {a, b, c}.{e, f, g}

Out[27]= a e + b f + c g

In[28]:= {a, b, c}.{e, f, g, h}

        Dot::dotsh : Tensors {a, b, c} and {e, f, g, h} have incompatible shapes. Mehr…

Out[28]= {a, b, c}.{e, f, g, h}
```

Matrixprodukt
■ Produit de matrice

Uebersicht: ■ Vue d'ensemble:

```
In[29]:= {{a},{b},{c}} // MatrixForm

Out[29]//MatrixForm=
        ⎛a⎞
        ⎜b⎟
        ⎝c⎠

In[30]:= Transpose[{{a},{b},{c}}] // MatrixForm

Out[30]//MatrixForm=
        (a  b  c)
```

Diverse Produkte:
■ Produits divers:

```
In[31]:= {{a},{b},{c}} . {{e},{f},{g}} // MatrixForm

        Dot::dotsh : Tensors {{a}, {b}, {c}} and {{e}, {f}, {g}} have incompatible shapes. Mehr…

Out[31]//MatrixForm=
        {{a}, {b}, {c}}.{{e}, {f}, {g}}
```

```
In[32]:= {{a},{b},{c}} . Transpose[{{e},{f},{g}}] //
                                MatrixForm
```

Out[32]//MatrixForm=

$$\begin{pmatrix} a\,e & a\,f & a\,g \\ b\,e & b\,f & b\,g \\ c\,e & c\,f & c\,g \end{pmatrix}$$

```
In[33]:= Transpose[{{a},{b},{c}}] . {{e},{f},{g}} //
                                MatrixForm
```

Out[33]//MatrixForm=

$$(a\,e + b\,f + c\,g)$$

```
In[34]:= Transpose[{{a},{b},{c}}] .
             Transpose[{{e},{f},{g}}] // MatrixForm

        Dot::dotsh : Tensors {{a, b, c}} and {{e, f, g}} have incompatible shapes. Mehr…

Out[34]//MatrixForm=
        {{a, b, c}}.{{e, f, g}}
```

Generierung: ■ Génération:

```
In[35]:= Table[u[[i]] v[[j]], {i, Length[u]}, {j,
                                Length[v]}]

Out[35]= {}
```

Was weiss man über u und v?
■ Que sait-on de u et v?

13.6. Weiter mit "Return" statt "Enter"
■ Continuer par "Return" au lieu de "Enter"

Beispiele ■ Exemple

```
In[36]:= a = 5
           + 7
           + 9

Out[36]= 5

Out[37]= 7

Out[38]= 9
```

```
In[39]:= a

Out[39]= 5

In[40]:= a = 5 +
           7 +
           9

Out[40]= 21

In[41]:= a

Out[41]= 21
```

Was ist hier los?

■ Que se passe-t-il ici?

13.7. Probleme mit Wurzeln (Mehrdeutigkeiten)
■ Problèmes de racines (différentes acceptions, équivoques)

Beispiele
■ Exemples

```
In[42]:= (x^2)^(0.5)

Out[42]= (x^2)^0.5

In[43]:= Sqrt[x^2]

Out[43]= Sqrt[x^2]

In[44]:= Integrate[Sqrt[1 - Cos[x]^2], {x, 0, Pi/2}]

Out[44]= 1

In[45]:= Integrate[Cos[x], {x, Pi, 2Pi}]

Out[45]= 0

In[46]:= Integrate[Sqrt[1 - Sin[x]^2], {x, Pi, 2Pi}]

Out[46]= 2
```

Was ist hier los?

■ Que se passe-t-il ici?

```
In[47]:= ??Sqrt

         Sqrt[z] gives the square root of z. Mehr...

         Attributes[Sqrt] = {Listable, NumericFunction, Protected}

In[48]:= Sqrt[(-2)^2]

Out[48]= 2
```

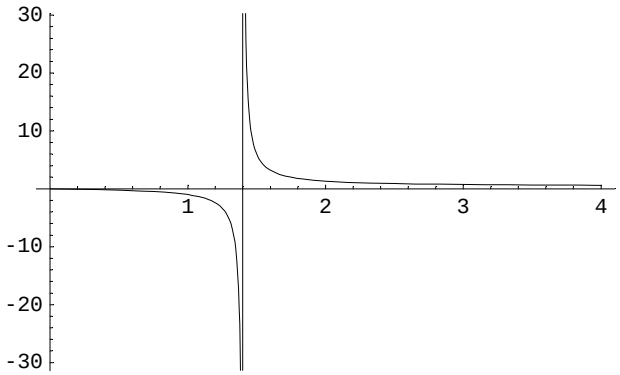
13.8. Abarbeitungsreihenfolge
▮ Ordre de travail

Probiere: ▮ Essaie:

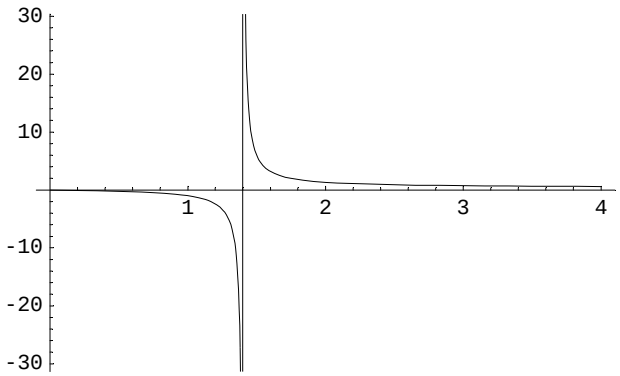
```
In[49]:= s = 2 x /(5x - 7)
```

$$\text{Out[49]= } \frac{2\,x}{-7+5\,x}$$

```
In[50]:= Plot[s, {x, 0, 4}];
```

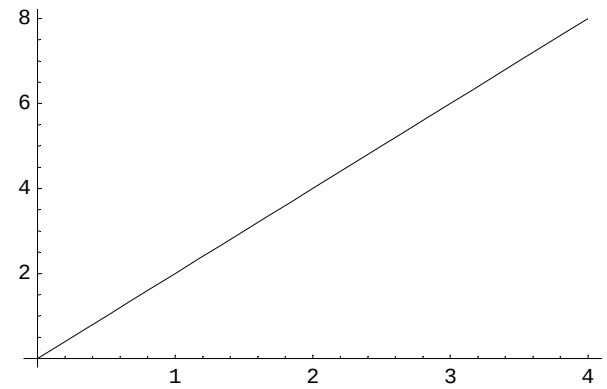



```
In[51]:= Plot[Numerator[s], {x, 0, 4}];
```



Wieso der Fehler? Probiere:
▮ Pourquoi l'erreur? Essaie:

```
In[52]:= Plot[Evaluate[Numerator[s]], {x, 0, 4}];
```



13.9. Ordnung von Regeln
▮ Ordre de règles

Probiere und erkläre die Prioritäten! Beispiel:
▮ Essaie et explique les priorités!

```
In[53]:=
        Clear[f];
        f[x_] := x^2;
        f[y_] := y^3;

In[56]:= {f[x],f[y]}

Out[56]= {x^3, y^3}

In[57]:= ??f

       Global`f

        f[y_] := y^3
```

Beispiel: ▮ Exemple:

```
In[58]:= Clear[ung];
         ung[x_?OddQ ] := (ung[x] = 1);
         ung[x_?EvenQ] := (ung[x] = 0);


In[61]:= Table[ung[x], {x,10}]

Out[61]= {1, 0, 1, 0, 1, 0, 1, 0, 1, 0}

In[62]:= ??ung

       Global`ung

        ung[1] = 1

        ung[2] = 0

        ung[3] = 1

        ung[4] = 0

        ung[5] = 1

        ung[6] = 0

        ung[7] = 1

        ung[8] = 0

        ung[9] = 1

        ung[10] = 0

        ung[x_?OddQ] := ung[x] = 1

        ung[x_?EvenQ] := ung[x] = 0
```

13.10. "Protect" und Funktionen
■ "Protect" et fonctions

In[63]:= **??Protect**

```
Protect[s1, s2, ... ] sets the attribute Protected
   for the symbols si. Protect["form1", "form2", ... ] protects all
   symbols whose names match any of the string patterns formi. Mehr...

Attributes[Protect] = {HoldAll, Protected}
```

In[64]:= **??Unprotect**

```
Unprotect[s1, s2, ... ] removes the attribute Protected
   for the symbols si. Unprotect["form1", "form2", ... ] unprotects
   all symbols whose names textually match any of the formi. Mehr...

Attributes[Unprotect] = {HoldAll, Protected}
```

Probiere: ■ Essaie:

In[65]:= **Integrate[f[x_], x_] := F[x]**

SetDelayed::write : Tag Integrate in $\int x_^3 \partial x_$ is Protected. Mehr…

Out[65]= $Failed

In[66]:=
```
Remove[f, F];
f/: Integrate[f[x_], x_] := F[x];
Integrate[f[x], x]
```

Out[68]= F[x]

Probiere: ■ Essaie:

In[69]:= **Pi = 3.1416**

Set::wrsym : Symbol π is Protected. Mehr…

Out[69]= 3.1416

In[70]:= **{Pi, Pi // N}**

Out[70]= {π, 3.14159}

In[71]:=
```
Unprotect[Pi];
Pi = 3.1416;
Pi
```

Out[73]= 3.1416

In[74]:=
```
Protect[Pi];
Pi // N
```

Out[75]= 3.1416

In[76]:= **??Pi**

Pi is the constant pi, with numerical value approximately equal to 3.14159. Mehr...

Attributes[π] = {Constant, Protected, ReadProtected}

In[77]:=
```
Unprotect[Pi];
Remove[Pi];
Pi // N
```

Out[79]= Pi

In[80]:= **??Pi**

Pi is the constant pi, with numerical value approximately equal to 3.14159.

13.11. Debugging
∎ Debugging

13.11.1. Allgemeine Regeln
∎ Règles générales

Beachte beim Programmieren:

Dokumentiere die eigenen Funktionen.
Beachte Gross- und Kleinschreibung.
Beachte Klammerung.
Beachte die Initialisierung von Variablen.
Beachte die Funktionennamen.
Teste die Statements.
Beachte die Argumente von Funktionen.
Beachte die Punktuation.
Kontrolliere, ob Befehle oder Zeichen vergessen worden sind.
Mache Tests in trivialen Fällen.
Teste den Code mit Testdaten.
Beachte das Zuladen von Packages.
Beachte die "Contexts" (Erklärung in Kap. 15.)

Schreibe die Programme so, dass ein anderer nicht viel Zeit benötigt, um sie zu lesen. Wer die eigenen Kriterien den andern aufzwingen will, ist nicht sehr teamfähig. Denke auch an die andern, obwohl Du ja für Dich alleine sicher sehr gescheit bist!

∎

Considère en programmant:

Documente les propres fonctions.
Tiens compte des majuscules et minuscules.
Tiens compte des initiales des variables.
Tiens compte des noms et fonctions.
Examine et teste les statements.
Tiens compte des arguments de fonction.
Tiens compte de la ponctuation.
Contrôle si des ordres ou des signes ont été oubliés.
Fais des tests dans des cas triviaux.
Teste le code par des données aptes à cela.
Tiens compte des Packages utilisés.
Tiens compte des "Contexts". (Explication au chapitre 15.)

Ecris les programmes de façon qu'une autre personne ne perde pas beaucoup de temps à les lire. Qui veut imposer ses propres critères aux autres, n'est pas capable de travailler en groupe. Pense aux autres, bien que tu n'aies pas de problèmes en travaillant seul.

Studiere: ∎ Etudie:

In[81]:= **?On**

```
On[symbol::tag] switches on a message, so that it can be printed.
   On[s] switches on tracing for the symbol s. On[m1, m2, ... ] switches
   on several messages. On[ ] switches on tracing for all symbols. Mehr...
```

In[82]:= **?Off**

```
Off[symbol::tag] switches off a message, so that it is no longer printed. Off[s]
   switches off tracing messages associated with the symbol s. Off[m1, m2, ... ]
   switches off several messages. Off[ ] switches off all tracing messages. Mehr...
```

In[83]:= **?Print**

```
Print[expr1, expr2, ... ] prints the expri concatenated together. Mehr...
```

In[84]:= **?Debug**

```
Debug is an obsolete function, superseded by Trace. Debug[expr] evaluates
  expr, allowing you to stop at certain points in some control structures,
  and see what's happening or run commands. Debug[expr, {f1, f2, ...}]
  stops only in evaluation of the functions fi (and everything they call).
```

In[85]:= **?Trace**

```
Trace[expr] generates a list of all expressions used in the evaluation of expr. Trace[expr,
   form] includes only those expressions which match form. Trace[expr, s] includes
   all evaluations which use transformation rules associated with the symbol s. Mehr...
```

In[86]:= **?Input**

```
Input[ ] interactively reads in one Mathematica expression.
   Input["prompt"] requests input, using the specified string as a prompt. Mehr...
```

In[87]:= **?Interrupt**

```
Interrupt[ ] generates an interrupt. Mehr...
```

13.11.2. Traceing

▮ Traceing

Probiere aus: ▮ Essaie:

In[88]:=
```
Remove[f, g];
f[x_]:= x^3 + g[x];
g[x_]:= x - 14
```

In[91]:= **On[f, g]**

In[92]:= **f[4]**

```
f::trace : f[4] --> 4^3 + g[4]. Mehr…

g::trace : g[4] --> 4 - 14. Mehr…
```

Out[92]= 54

In[93]:= **Off[f, g]**

In[94]:= **f[4]**

Out[94]= 54

13.11.3. Print
▮ Print

Von 10.3.7:
▮ De 10.3.7:

```
In[95]:= median[liste_List]:=
             Block[{
                    sl,
                    len
                  },
                  len = Length[liste];
                  Print["Länge der Liste: ", len];
                  sl = Sort[liste];
                  Print["Sortierte Liste: ", sl];
                  Print["Median: "];
                  If[
                     OddQ[len],
                     sl[[(len+1)/2]],
                     (sl[[len/2]]+sl[[len/2+1]])/2
                  ]
             ];
             median[{24, 32, 36, 47, 49, 52}]
```

```
General::spell1 :
 Possible spelling error: new symbol name "median" is similar to existing symbol "Median". Mehr…

Länge der Liste: 6

Sortierte Liste: {24, 32, 36, 47, 49, 52}

Median:
```

$$\text{Out[96]=}\ \frac{83}{2}$$

13.11.4. Debug (für Version 1.2)
▮ Debug (pour version 1.2)

Beispiel: ▮ Exemple:

```
In[97]:= ?$Version

$Version is a string that represents the version of Mathematica you are running. Mehr…

In[98]:= ?$VersionNumber

$VersionNumber is a real number which gives the current Mathematica
   kernel version number, and increases in successive versions. Mehr…

In[99]:= $VersionNumber < 2

Out[99]= False
```

In[100]:=
```
If[$VersionNumber < 2,
   Debug[For[n = 1, n < 4, ++n, Print[n^3]]],
   Print["Achtung: Bei Debug ab Version 2:
            Systemabsturz! \
   Attention: Debug dès version 2:
            système interrompu!"]]
```

```
Achtung: Bei Debug ab Version 2:        Systemabsturz!
  Attention: Debug dès version 2:               système interrompu!
```

13.11.5. Trace

▮ Trace

Studiere: ▮ Etudie:

In[101]:=
```
2 5 + 8 9
```

Out[101]=
```
82
```

In[102]:=
```
Trace[2 5 + 8 9]
```

Out[102]=
```
{{2 5, 10}, {8 9, 72}, 10 + 72, 82}
```

In[103]:=
```
Trace[2 5 + 8 9, TraceDepth -> 1]
```

Out[103]=
```
{10 + 72, 82}
```

In[104]:=
```
Trace[2 5 + 8 9, Plus]
```

Out[104]=
```
{10 + 72, 82}
```

In[105]:=
```
Trace[2 5 + 8 9, Times]
```

Out[105]=
```
{{2 5, 10}, {8 9, 72}}
```

In[106]:=
```
Trace[2 5 + 8 9, x_ y_]
```

Out[106]=
```
{{2 5}, {8 9}}
```

In[107]:=
```
Trace[2 5 + 8 9, TraceDepth -> 1]
```

Out[107]=
```
{10 + 72, 82}
```

13.11.6. Input

Input

Studiere (obiges Beispiel abgeändert):
Falls während der Rechnung plötzlich ein Fenster geöffnet wird so kannst Du eine Variable abfragen. Schreibe z.B. "?len" ins Fenster und beobachte, was passiert!

Etudie (Exemple ci-dessus changé):
Si pendant le calcul il s'ouvre tout à coup une fenêtre, tu peux demander le contenu d'une variable. Ecris p.ex. "?len" dans la fenêtre et observe ce qui se passe!

In[108]:=
```
median[liste_List]:=
    Block[{
            sl,
            len
          },
          len = Length[liste];
          sl = Sort[liste];
          Print[Input[]];
          If[
             OddQ[len],
             sl[[(len+1)/2]],
             (sl[[len/2]]+sl[[len/2+1]])/2
          ]
    ];
    median[{24, 32, 36, 47, 49, 52}]
```

```
Global`len

len = 6

Null
```

Out[109]=

$$\frac{83}{2}$$

13.11.7. Interrupt

Inerrupt

Beispiel: n! soll berechnet werden. Die Rechnung soll wegen der Rechnungszeit aber nur durchgeführt werden für n <= 1000.

Exemple: n! doit être calculé, mais à cause du temps de calcul on ne le calculera que pour des n >= 1000.

In[110]:=
```
Remove[f];
f::gross = "Eingegebene Zahl grösser tausend. \
        Rechenzeit möglicherweise sehr lang.";
f[n_] := If[n <= 500, Print[n,"! = ", n!],
        f::gross]
```

In[113]:=

```
f[10]
```

```
10! = 3628800
```

In[114]:=

```
f[100]
```

```
100! =
 933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761
  565182862536979208272237582511852109168640000000000000000000000000
```

In[115]:=

```
f[500]
```

```
500! =
 1220136825991110068701238785423046926253574342803192842192413588385845373153881997605496447
  022032818630136164771482035841633787220781772004807852051593292854779075719393306037729608
  908627042917454788242491272634430567017327076946106280231045264421887878946575477714986349
  367781037644274033827365397471386477878495438489595537537990423241061271326984327745715546
  099772027810145610811883737095310163563244329870295638966289116589747695720879269288712817
  007026517450776841071962439039432253642260523494585012991857150124870696156814162535905669
  423813008856249246891564126775654481886506593847951775360894005745238940335798476363944905
  130623237490664450488246650759467358620746379251842004593696929810222639719525971909452178
  333175693458150855233282076282002340262690789834245171200620771464097945611612762914595123
  229913340169552363850942885592018727433795173014586357570828355780158735432768888680120399
  823847021514676054454076635359841744304801289383138968816394874696588175045069263653381750
  547812864000000000000000000000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000000000
```

In[116]:=

```
f[501]
```

Out[116]=

```
Eingegebene Zahl grösser tausend.   Rechenzeit möglicherweise sehr lang.
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[117]:=

```
(* Old Form: Remove["Global`@*"] *)
```

In[118]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

13. Fallstricke und "Debugging" (Fehler eliminieren)
▮ Embûches et "Debugging" (éliminer les erreurs)

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 13 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 13.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

"Putzmaschine" einsetzen (hier sehr wichtig)
▮ Employer la "machine de nettoyage" (très important dans ce cas)

```
In[1]:= x = 1

Out[1]= 1

In[2]:= Clear[]; x

Out[2]= 1

In[3]:= (* Old Form: Remove["Global`@*"] *)

In[4]:= Remove["Global`*"]
```

Versuche, in den folgenden Beispielen die Fehler zu finden!
▮ Essaie de trouver les erreurs dans les exemples suivants!

Aufgabe 1 ▮ Problème 1

```
In[5]:= 3 Plus 4

Out[5]= 12 Plus
```

(Ueberlege, wie "Plus" wirkt.)
▮ (Réfléchis aux effets de "Plus")

```
In[6]:= Plus[3, 4]

Out[6]= 7
```

Aufgabe 2 ▮ Problème 2

```
In[7]:= mean[x_] := Add[x]/Length[x];
        mean[{1,2,3}]

        General::spell1 :
         Possible spelling error: new symbol name "mean" is similar to existing symbol "Mean". Mehr…

Out[8]= 1/3 Add[{1, 2, 3}]
```

Schreibe den Code richtig!
▮ Ecris le Code correctement!

```
In[9]:= ?Plus

        x + y + z represents a sum of terms. Mehr…

In[10]:= Remove[mean];
         mean[{x__}] := Plus[x]/Length[{x}];
         mean[{1,2,3}]

         General::spell1 :
          Possible spelling error: new symbol name "mean" is similar to existing symbol "Mean". Mehr…

Out[12]= 2
```

Aufgabe 3 ▮ Problème 3

Wieso kommen keine Primzahlen?
▮ Pourquoi il n'apparaît pas de nombre premier?

```
In[13]:= Select[{2, 3, 7, 9, 11, 12}, PrimeQ[x_]]

Out[13]= {}
```

Schreibe den Code richtig!

■ Ecris le code correctement!

```
In[14]:= ?Select
```

```
Select[list, crit] picks out all elements ei of list for which crit[ei] is True. Select[
   list, crit, n] picks out the first n elements for which crit[ei] is True. Mehr…
```

```
In[15]:= Trace[Select[{2, 3, 7, 9, 11, 12}, PrimeQ[x_]]]
```

```
Out[15]= {{PrimeQ[x_], False}, Select[{2, 3, 7, 9, 11, 12}, False], {}}
```

```
In[16]:= Select[{2, 3, 7, 9, 11, 12}, PrimeQ]
```

```
Out[16]= {2, 3, 7, 11}
```

Aufgabe 4 ■ Problème 4

Berechne das Integral!

■ Calcule l'intégral!

```
In[17]:= Integral[xCos(x), (x, 0, pi)]
```

```
General::spell1 :
 Possible spelling error: new symbol name "xCos" is similar to existing symbol "Cos". Mehr…

Syntax::sntxf:
   "Integral[xCos(x), (x" cannot be followed by
    ", 0, pi)]". Mehr…
```

Schreibe den Code richtig!

■ Ecris le code correctement!

```
In[17]:= Integrate[x Cos[x], {x, 0, Pi}]
```

```
Out[17]= -2
```

Aufgabe 5 ■ Problème 5

Mit der folgenden Funktion sollen 6 Lottozahlen ausgewählt werden. Wieso kommt nur eine?

■ Avec la fonction suivante on choisit 6 nombres de loto. Pourquoi il n'en vient qu'un?

```
In[18]:= machLotto:=
            Block[{
                n = 1
                  },
                  While[n <= 6;
                    Print[
                        Random[Integer, {0, 49}]
                    ];
                    n++
                  ]
            ];
          machLotto
```

```
42
```

Schreibe den Code richtig!

■ Ecris le code correctement!

```
In[20]:= machLotto:=
             Block[{
                 n = 1
                   },
                   While[n <= 6,
                     Print[
                         Random[Integer, {0, 49}]
                     ];
                     n++
                   ]
             ];
          machLotto
```

```
27

26

13

1

43

6
```

Aufgabe 6 ■ Problème 6

Studiere die folgenden Funktionen:

■ Etudie les fonctions suivantes:

```
In[22]:= Remove[aoi, eip];
         aoi[f_[x_], min_, max_] := NIntegrate[f[x],
                                        {x, min, max}];
         eip[x_                ] := x^2 + 5 Exp[x];
         aoi[eip[t], 5, 10]
```

Out[25]= aoi[5 ℯ^t + t^2, 5, 10]

Wieso kommt nicht ein numerischer Wert? Hier einige Studien und Lösungsmöglichkeiten:

■ Pourquoi il n'apparaît pas une valeur numérique? Voici quelques études et quelques possibilités de solution:

```
In[26]:= ?aoi
```

```
Global`aoi

aoi[f_[x_], min_, max_] := NIntegrate[f[x], {x, min, max}]
```

```
In[27]:= ?On
```

```
On[symbol::tag] switches on a message, so that it can be printed.
   On[s] switches on tracing for the symbol s. On[m1, m2, ... ] switches
   on several messages. On[ ] switches on tracing for all symbols. Mehr...
```

```
In[28]:= On[aoi, eip]
```

In[29]:= **aoi[eip[t], 5, 10]**

eip::trace : eip[t] --> $t^2 + 5\,e^t$. Mehr…

aoi::trace : aoi[eip[t], 5, 10] --> aoi[$5\,e^t + t^2$, 5, 10]. Mehr…

Out[29]= aoi[$5\,e^t + t^2$, 5, 10]

In[30]:=
```
Remove[aoi, eip];
On[aoi, eip];
aoi[f_[x_], y_, min_, max_] := NIntegrate[f[x],
                                    {y, min, max}];
eip[z_                ] := z^2 + 5 Exp[z];
aoi[eip[t], 5, 10]
```

eip::trace : eip[t] --> $t^2 + 5\,e^t$. Mehr…

aoi::trace : aoi[eip[t], 5, 10] --> aoi[$5\,e^t + t^2$, 5, 10]. Mehr…

Out[34]= aoi[$5\,e^t + t^2$, 5, 10]

In[35]:=
```
Remove[aoi, eip];
On[aoi, eip];
aoi[f_, y_, min_, max_] := NIntegrate[f,
                                    {y, min, max}];
eip := x^2 + 5 Exp[x];
aoi[eip, t, 5, 10]
```

eip::trace : eip --> $x^2 + 5\,e^x$. Mehr…

aoi::trace : aoi[eip, t, 5, 10] --> aoi[$5\,e^x + x^2$, t, 5, 10]. Mehr…

aoi::trace : aoi[$5\,e^x + x^2$, t, 5, 10] --> NIntegrate[$5\,e^x + x^2$, {t, 5, 10}]. Mehr…

NIntegrate::inum : Integrand $5\,e^x + x^2$ is not numerical at {t} = {7.5}. Mehr…

Out[39]= NIntegrate[$5\,e^x + x^2$, {t, 5, 10}]

In[40]:= **aoi[eip, x, 5, 10]**

eip::trace : eip --> $x^2 + 5\,e^x$. Mehr…

aoi::trace : aoi[eip, x, 5, 10] --> aoi[$5\,e^x + x^2$, x, 5, 10]. Mehr…

aoi::trace : aoi[$5\,e^x + x^2$, x, 5, 10] --> NIntegrate[$5\,e^x + x^2$, {x, 5, 10}]. Mehr…

Out[40]= 109682.

In[41]:=
```
Remove[aoi, eip];
On[aoi, eip];
aoi[f_, x_, min_, max_] := NIntegrate[f,
                                    {x, min, max}];
eip[y_                ] := y^2 + 5 Exp[y];
aoi[eip[t], t, 5, 10]
```

eip::trace : eip[t] --> $t^2 + 5\,e^t$. Mehr…

aoi::trace : aoi[eip[t], t, 5, 10] --> aoi[$5\,e^t + t^2$, t, 5, 10]. Mehr…

aoi::trace : aoi[$5\,e^t + t^2$, t, 5, 10] --> NIntegrate[$5\,e^t + t^2$, {t, 5, 10}]. Mehr…

Out[45]= 109682.

Aufgabe 7 ▮ Problème 7

Schreibe die folgende Funktion kürzer und besser:

▮ Ecris la fonction suivante d'une daçon plus courte et meilleure:

```
In[46]:= Remove[f];
         f[g_, x_List]:=
             Block[{
                 l = {},
                 n = 1
             },
             Do[
                 l = Join[l, {g[x[[i]]]}],
                 {i, Length[x]}
             ];
             l
         ];
         f[g, {a, b, c}]
```

```
Out[48]= {g[a], g[b], g[c]}
```

Einfacher: ▮ Plus simplement:

```
In[49]:= Clear[f];
         f[g_, x_List] := Map[g, x];
         f[g, {a, b, c}]
```

```
Out[51]= {g[a], g[b], g[c]}
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

```
In[52]:= (* Old Form: Remove["Global`@*"] *)
```

```
In[53]:= Remove["Global`*"]
```

Kurs ▮ Cours

14. Input und Output
▮ Input et output

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 14 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 14.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Problem: Datenimport und Datenexport in und von *Mathematica*.
▮ Problème: Importation et exportation de données dans et de *Mathematica*.

14.1. Input
▮ Input

14.1.1. Dateneingabe via Tastatur während des Programmlauf
▮ Entrée des données par les touches pendant que le programme marche

Studiere: ▮ Etudie:

```
In[1]:= ?Input

        Input[ ] interactively reads in one Mathematica expression.
           Input["prompt"] requests input, using the specified string as a prompt. Mehr...

In[2]:= ?InputString

        InputString[ ] interactively reads in a character string.
           InputString["prompt"] requests input, using the specified string as a prompt. Mehr...
```

```
In[3]:= ?BaseForm

        BaseForm[expr, n] prints with the numbers in expr given in base n. Mehr...

In[4]:= BaseForm[15,2]+BaseForm[17,2]

Out[4]= 1111_2 + 10001_2

In[5]:= benutzerInput = Input[
        "Schreibe eine Zahl zwischen 1 und 10: \
        Ecris un nombre entre 1 et 10:"]

Out[5]= 8
```

Achtung: Das nachfolgende Programm bricht erst ab, wenn keine Integer-Zahl eingegeben wird! Probiere:

Attention: Le programme suivant s'interrompt seulement si on n'entre pas le nombre Integer! Essaie:

```
In[6]:= dualUmwandlung:=
            Module[{benutzerInput},
                    While[
                        IntegerQ[
                            userInput = Input[
            "Schreibe eine Zahl zwischen 1 und 10: \
            Ecris un nombre entre 1 et 10:"]
                                ],
                        Print[userInput, " = ",
                                BaseForm[userInput, 2]]
                    ]
            ];

        dualUmwandlung

        1 = 1_2

        2 = 10_2

        3 = 11_2

        4 = 100_2

        5 = 101_2

        6 = 110_2

        7 = 111_2

        8 = 1000_2

        9 = 1001_2
```

(* Input ==> 0.1 !!! *)

```
In[8]:=
```

14.1.2. Datenimport von einem File
Importation de données d'un fichier

Achtung: Achte auf das zu deinem Betriebssystem lauffähige Beispiel!
Der Umgang mit externen Dateien von *Mathematica* aus hat ab Version 3.0 eine

grosse Veränderung erfahren!
■ Attention: Prends seulement les exemples qui tournent sur ton propre système!
Le traitement des fichiers externes depuis *Mathematica* a beaucoup changé depuis la version 3.0!

Beispiel:
(Das File AAAdaten) befindet sich wahrscheinlich in Deinem Verzeichnis C:\Programme\WolframResearch\-*Mathematica*\3.0 (win98). - Sonst musst Du Deinen mir nicht bekannten Pfad suchen!)
■ Exemple:
(Le file AAAdanten.nb (aadaten.ma) se trouve probablement dans le directoire C:\Programme\WolframResearch\-*Mathematica*\3.0 (win98). - Si non tu dois chercher ton "path" que je ne connais pas!)

Mathematica 3.02, Windows98
Beispiel: neues Verzeichnis erstellen:
■ Faire un nouveau directoire:

```
In[239]:=
        CreateDirectory["C:\work\MathematicaData\NewDir"]

Out[239]=
        C:\work\MathematicaData\NewDir
```

```
In[9]:= x = Sin[3] // N >> AAADaten

In[10]:= << AAADaten

Out[10]= 0.14112
```

Beispiel: Erstelle ein Verzeichnis C:\work (falls nicht vorhanden).
(Das File AABDaten befindet sich in Deinem Verzeichnis "work" (win98), (aadaten.ma: "Mate" (win3.1, 95....), "WirzMathematicaFiles/Daten" (NeXT) usw.) . - Sonst musst Du Deinen mir nicht bekannten Pfad angeben!)
■ Exemple: Ouvre un directoire C:\work (si cela n'existe pas encore).
(Le file AABDaten se trouve dans le directoire "work" (win98), (aadaten.ma: "Mate" (win3.1, 95....), "Wirz-MathematicaFiles/Daten" (NeXT) etc.). - Si non tu dois m'indiquer ton "path" que je ne connais pas!)

Schreibe Daten ins File C:\work\AABDaten:
■ Ecrire des données dans le fichier C:\work\AABDaten:

```
In[11]:= "3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo
            7777 6666 4.321 1.11111, {1,2,3,4}" >>
          C:\\work\\MathematicaData\\AABDaten;

In[12]:= {"Cos[3]//N", Cos[3] // N} >> C:\\work\\MathematicaData\\AACDaten;

In[13]:= {3 4 5 1 2 3 6 6, "3 4 5 1 2 3 6 6",
            {3 4 5 1 2 3 6 6}} >> C:\\work\\MathematicaData\\AACDaten;
```

Lese das File C:\work\AABDaten:
■ Lire le fichier C:\work\AABDaten:

```
In[14]:= << C:\\work\\MathematicaData\\AABDaten

Out[14]= 3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}
```

In[15]:= **!! C:\\work\\MathematicaData\\AABDaten**

```
"3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}"
```

win98:
Erstelle nötigenfalles eine Textdatei "c:\work\AAEDaten.txt" mit Inhalt:
■ Si nécessaire composer un fichier "c:\work\AAEDaten.txt" avec le contenu:
1 2 3 4 5 6 7
1.1 1.2 1.3 1.4
11 12 13 14 115
a b c d e f g
Sin[5]\\N hallo
"hallo"

Erstelle ebenso eine Textdatei "c:\work\AAFDaten.txt" mit Inhalt:
■ Composer aussi un fichier "c:\work\AAEDaten.txt" avec le contenu:
1 2 3 4 5 6 7
1.1 1.2 1.3 1.4
11 12 13 14 115

Inhalt anschauen:
■ Regarder le contenu:

In[16]:= **!! "C:\\work\\MathematicaData\\AAFdaten.txt"**

```
1234  k34ljök4j8sn3i 43 45 dkr kjkj
fjkldjklf
12345
677890
```

In[17]:= **(* !!f:\Mathe/Daten/aadaten0 (* win3.1 etc. *) *)**

File einlesen und ausführen:
■ Lire le fichier et exécuter:

In[18]:= **<< C:\\work\\MathematicaData\\AAFdaten.txt**

Out[18]= 677890

In[19]:= **(* <<f:\Mathe/Daten/aadaten1 (* win3.1 etc. *) *)**

Was ist los gewesen? Schauen wir weiter:
■ Que s'est-il passé? Continuons d'observer:

In[20]:= **?<<**

```
<<name reads in a file, evaluating each expression in it, and returning the last one.
```
Mehr…

In[21]:= **?Get**

```
<<name reads in a file, evaluating each expression in it, and returning the last one.
```
Mehr…

In[22]:= **!! C:\\work\\MathematicaData\\AAEDaten.txt**

```
1234  99 2 3 4  5 11 1.2  1.12334455
1.123344335
12345
677890
```

In[23]:= **(* !!f:\Mathe/Daten/aadaten1 (* win3.1 etc. *) *)**

In[24]:= **<< C:\\work\\MathematicaData\\AAEDaten.txt**

Out[24]= 677890

In[25]:= **(* <<f:\Mathe/Daten/aadaten1 (* win3.1 etc. *) *)**

In[26]:= **Get["C:\\work\\MathematicaData\\AAEDaten.txt"]**

Out[26]= 677890

In[27]:= **(* Get["f:\Mathe/Daten/aadaten1"]**
(* win3.1 etc. *) *)

In[28]:= **1 7 3 4 2 9 (*Produkt! *)**

Out[28]= 1512

Spezifisch einlesen:

Lire spécifiquement:

In[29]:= **?ReadList**

```
ReadList["file"] reads all the remaining expressions in a file, and returns
   a list of them. ReadList["file", type] reads objects of the specified type
   from a file, until the end of the file is reached. The list of objects read
   is returned. ReadList["file", {type1, type2, ... }] reads objects with a
   sequence of types, until the end of the file is reached. ReadList["file",
   types, n] reads only the first n objects of the specified types. Mehr...
```

In[31]:= **ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", Real]**

Out[31]= {1234., 99., 2., 3., 4., 5., 11., 1.2, 1.12334, 1.12334, 12345., 677890.}

In[32]:= **(* ReadList["f:\Mathe/Daten/aadaten0", Real] *)**
(* win3.1 etc. *)

In[34]:= **?Real**

```
Real is the head used for real (floating-point) numbers. Mehr...
```

In[35]:= **ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", Number]**

Out[35]= {1234, 99, 2, 3, 4, 5, 11, 1.2, 1.12334, 1.12334, 12345, 677890}

In[36]:= **(* ReadList["f:\Mathe/Daten/aadaten0", Number] *)**
(* win3.1 etc. *)

In[38]:= **?Number**

```
Number represents an exact integer or an approximate real number in Read. Mehr...
```

In[39]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Byte]
```

Out[39]=
```
{49, 50, 51, 52, 32, 32, 107, 51, 52, 108, 106, 246, 107, 52, 106,
 56, 115, 110, 51, 105, 32, 52, 51, 32, 52, 53, 32, 100, 107, 114, 32,
 107, 106, 107, 106, 10, 102, 106, 107, 108, 100, 106, 107, 108, 102,
 32, 10, 49, 50, 51, 52, 53, 10, 54, 55, 55, 56, 57, 48, 10, 32, 32, 32}
```

In[40]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", Byte] *)
                                  (* win3.1 etc. *)
```

In[42]:=
```
?Byte
```

Byte represents a single byte of data in Read. Mehr...

In[43]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Character]
```

Out[43]=
```
{1, 2, 3, 4,  ,  , k, 3, 4, l, j, ö, k, 4, j, 8,
 s, n, 3, i,  , 4, 3,  , 4, 5,  , d, k, r,  , k, j, k, j,
, f, j, k, l, d, j, k, l, f,  ,
, 1, 2, 3, 4, 5,
, 6, 7, 7, 8, 9, 0,
,  ,  ,  }
```

In[44]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0",
                       Character]  (* win3.1 etc. *) *)
```

In[45]:=
```
?Character
```

Character represents a single character in Read. Mehr...

In[46]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", String]
```

Out[46]=
```
{1234  k34ljök4j8sn3i 43 45 dkr kjkj, fjkldjklf , 12345, 677890,    }
```

In[47]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", String]
                                  (* win3.1 etc. *) *)
```

In[48]:=
```
?String
```

String is the head of a character string "text". Mehr...

In[49]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Record]
```

Out[49]=
```
{1234  k34ljök4j8sn3i 43 45 dkr kjkj, fjkldjklf , 12345, 677890,    }
```

In[50]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", Record] *)
                                  (* win3.1 etc. *)
```

In[52]:=
```
?Record
```

Record represents a record in Read, Find and related functions. Mehr...

In[53]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Word]
```

Out[53]=
```
{1234, k34ljök4j8sn3i, 43, 45, dkr, kjkj, fjkldjklf, 12345, 677890}
```

In[54]:=
```
(* ReadList["f:\Mathe/Daten/aadaten0", Word] *)
                                  (* win3.1 etc. *)
```

In[56]:=
```
?Word
```

Word represents a word in Read, Find and related functions. Mehr...

```
In[57]:= (* ReadList["f:\Mathe/Daten/aadaten1", Real] *)
                                  (* win3.1 etc. *)

In[59]:=  (* ReadList["f:\Mathe/Daten/aadaten1", Number] *)
                                  (* win3.1 etc. *)

In[61]:= (* ReadList["f:\Mathe/Daten/aadaten2", Number] *)
                                  (* win3.1 etc. *)

In[63]:= (* !!f:\Mathe/Daten/aadaten2    (* win3.1 etc. *) *)

In[64]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", {Number, Number}]

Out[64]= {{1234, 99}, {2, 3}, {4, 5}, {11, 1.2}, {1.12334, 1.12334}, {12345, 677890}}

In[65]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                {Number, Number}] *)    (* win3.1 etc. *)

In[66]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt",
         {Number, Number, Number, Number}]

Out[66]= {{1234, 99, 2, 3}, {4, 5, 11, 1.2}, {1.12334, 1.12334, 12345, 677890}}

In[67]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                {Number, Number, Number, Number}] *)
                                  (* win3.1 etc. *)

In[69]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt", Cos[Number]]

Out[69]= {Cos[1234], Cos[99], Cos[2], Cos[3], Cos[4], Cos[5],
          Cos[11], 0.362358, 0.43267, 0.43267, Cos[12345], Cos[677890]}

In[70]:= ReadList["C:\\work\\MathematicaData\\AAEDaten.txt",Cos[Number]] // N

Out[70]= {-0.798551, 0.0398209, -0.416147, -0.989992, -0.653644,
          0.283662, 0.0044257, 0.362358, 0.43267, 0.43267, 0.111436, -0.961386}

In[71]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                Cos[Number]] *)     (* win3.1 etc. *)

In[72]:= ReadList["C:\\work\\MathematicaData\\AAFDaten.txt", Cos[Number]] // N

        Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAFDaten.txt. Mehr…

Out[72]= {-0.798551}

In[73]:= (* ReadList["f:\Mathe/Daten/aadaten1",
                Cos[Number]] // N *)    (* win3.1 etc. *)

In[74]:= (* !!f:\Mathe/Daten/aadaten3 *) (* win3.1 etc. *)

In[75]:= ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
         {String, Number, Number}]

        Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAFDaten.txt. Mehr…

Out[75]= {{1234  k34ljök4j8sn3i 43 45 dkr kjkj, $Failed, EndOfFile}}

In[76]:= (* ReadList["f:\Mathe/Daten/aadaten3",
         {String, Number, Number}] *)    (* win3.1 etc. *)

In[77]:= (* !!f:\Mathe/Daten/aadaten4 *) (* win3.1 etc. *)
```

```
In[78]:= ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
          {Number, Number, String}]
```

```
Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAFDaten.txt. Mehr…
```

```
Out[78]= {{1234, $Failed, EndOfFile}}
```

```
In[79]:= (*ReadList["f:\Mathe/Daten/aadaten4",
          {Number, Number, String}] *)  (* win3.1 etc. *)
```

```
In[80]:= (*!!f:\Mathe/Daten/aadaten5 *)  (* win3.1 etc. *)
```

```
In[81]:= (* ReadList["f:\Mathe/Daten/aadaten5",
          {Number, Number, String}] *)    (* win3.1 etc. *)
```

Probiere eigene files!

- Essaie tes propres fichiers!

14.1.3. Einlesen von Records und Words
Lire Records et Words

Studiere die "Befehle":

- Etudie les "ordres":

```
In[82]:= ?*Record*
```

System`

```
NullRecords  RecordLists
Record       RecordSeparators
```

```
In[83]:= ?RecordLists
```

```
RecordLists is an option for ReadList which specifies whether objects
   from separate records should be returned in separate sublists. Mehr…
```

```
In[84]:= ?RecordSeparators
```

```
RecordSeparators is an option for Read, Find and related functions which
   specifies the list of strings to be taken as delimiters for records. Mehr…
```

```
In[85]:= ?WordSeparators
```

```
WordSeparators is an option for Read, Find and related functions which
   specifies the list of strings to be taken as delimiters for words. Mehr…
```

```
In[86]:= ?NullRecords
```

```
NullRecords is an option for Read and related functions which specifies whether null
   records should be taken to exist between repeated record separators. Mehr…
```

```
In[87]:= ?NullWords
```

```
NullWords is an option for Read and related functions which specifies whether
   null words should be taken to exist between repeated word separators. Mehr…
```

In[88]:= **?TokenWords**

```
TokenWords is an option for Read and related functions
   which gives a list of token words to be used to delimit words. Mehr...
```

Studiere die Anwendungen:

Etudie les applications:

In[89]:= **!!C:\\work\\MathematicaData\\AABDaten**

```
"3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}"
```

In[90]:= **!! C:\\work\\MathematicaData\\AAGDaten.txt**

```
asdf 5  34 dkfjke jkdf
```

In[91]:= **(* !!f:\Mathe/Daten/spsheet0 *) (* win3.1 etc. *)**

In[92]:=
```
ReadList["C:\\work\\MathematicaData\\AAEDaten.txt",
 {Number, Number, Number}]
```

Out[92]= {{1234, 99, 2}, {3, 4, 5}, {11, 1.2, 1.12334}, {1.12334, 12345, 677890}}

In[93]:=
```
(* ReadList["f:\Mathe/Daten/spsheet0",
 {Number, Number, Number}] *) (* win3.1 etc. *)
```

In[94]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
 Word, RecordLists -> True]
```

Out[94]= {{1234, k34ljök4j8sn3i, 43, 45, dkr, kjkj}, {fjkldjklf}, {12345}, {677890}}

In[95]:=
```
(*ReadList["f:\Mathe/Daten/spsheet0",
 Word, RecordLists ->True] *) (* win3.1 etc. *)
```

In[96]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",  Word]
```

Out[96]= {1234, k34ljök4j8sn3i, 43, 45, dkr, kjkj, fjkldjklf, 12345, 677890}

In[97]:=
```
(* ReadList["f:\Mathe/Daten/spsheet0",
 Word] *)          (* win3.1 etc. *)
```

In[98]:=
```
ReadList["C:\\work\\MathematicaData\\AAFDaten.txt",
          RecordLists -> True]
```

Out[98]= {{2387790 dkr k34ljök4j8sn3i kjkj}, {fjkldjklf}, {12345}, {677890}, {Null}}

In[99]:=
```
(* ReadList["f:\Mathe/Daten/spsheet0",
       RecordLists ->True] *) (* win3.1 etc. *)
```

In[100]:=
```
Clear[x];
(* x = ReadList["f:\Mathe/Daten/spsheet0",
        Word,
        WordSeparators -> {"\t"},
        RecordLists ->True] *) (* win3.1 etc. *)
```

In[102]:=
```
Clear[x];
x = ReadList["C:\\work\\MathematicaData\\AAGDaten.txt",
        Word,
        WordSeparators -> {"\t"},
        RecordLists -> True]
```

Out[103]=
```
{{asdf 5  34 dkfjke jkdf}, {   }}
```

In[104]:=
```
?Word
```

```
Word represents a word in Read, Find and related functions. Mehr...
```

In[105]:=
```
ToExpression[x]
```

Out[105]=
```
{{170 asdf dkfjke jkdf}, {Null}}
```

In[106]:=
```
?ToExpression
```

```
ToExpression[input] gives the expression obtained by interpreting strings
   or boxes as Mathematica input. ToExpression[input, form] uses interpretation
   rules corresponding to the specified form. ToExpression[input, form, h]
   wraps the head h around the expression produced before evaluating it. Mehr...
```

In[107]:=
```
FullForm[x]
```

Out[107]//FullForm=
```
List[List["asdf 5  34 dkfjke jkdf"], List["   "]]
```

In[108]:=
```
(* ReadList["f:\Mathe/Daten/spsheet1",
        {Word, Number, Number, Number},
        WordSeparators -> {" "},
        RecordLists ->True] *) (* win3.1 etc. *)
```

In[109]:=
```
ReadList["C:\\work\\MathematicaData\\AAGDaten.txt",
        {Word, Number, Number, Number},
        WordSeparators -> {" "},
        RecordLists -> True]
```

```
Read::readn : Syntax error reading a real number from C:\work\MathematicaData\AAGDaten.txt. Mehr…
```

Out[109]=
```
{{{asdf, 5, 34, $Failed}}}
```

In[110]:=
```
FullForm[%]
```

Out[110]//FullForm=
```
List[List[List["asdf", 5, 34, $Failed]]]
```

14.1.3. Ein längeres Beispiel
▮ Un exemple plus long

Zuerst definieren wir Funktionen. Damit kŏnnen wir obigen Befehl "nachbilden":
▮ D'abord nous définissons des fonctions. Avec cela nous pouvons "imiter" l'ordre ci-dessus.

In[111]:=
```
Clear[tabSeparate, tabSep];
tabSeparate[str_String]:= Apply[tabSep,
        Characters[str]];
tabSep[x__String,"\t",y___String]:=
        Flatten[{tabSep[x],tabSep[y]},1];
tabSep[x___String]:=
        ToExpression[StringJoin[x]]/;
        FreeQ[{x},"\t"];
ReadList["C:\\work\\MathematicaData\\AAGDaten.txt",
        tabSeparate[String]]
```

Out[115]=
```
{170 asdf dkfjke jkdf, Null}
```

In[116]:=
```
(*Clear[tabSeparate, tabSep];
tabSeparate[str_String]:=Apply[tabSep,
        Characters[str]];
tabSep[x__String,"\t",y___String]:=
        Flatten[{tabSep[x],tabSep[y]},1];
tabSep[x___String]:=
        ToExpression[StringJoin[x]]/;
        FreeQ[{x},"\t"];
ReadList["f:\Mathe/Daten/spsheet1",
        tabSeparate[String]](* win3.1 etc. *)*)
```

In[117]:=
```
?Characters
```

```
Characters["string"] gives a list of the characters in a string. Mehr…
```

In[118]:=
```
?FreeQ
```

```
FreeQ[expr, form] yields True if no subexpression in expr
   matches form, and yields False otherwise. FreeQ[expr, form, levelspec]
   tests only those parts of expr on levels specified by levelspec. Mehr…
```

Achtung! *Mathematica* ist nicht für Datamanagement und Datamanipulation gebaut. Man kann Daten in speziellen Datamanagement-Sprachen oder Programmen (low-level- Programmierung) viel schneller parsen (einlesen und interpretieren) sowie manipulieren. Dies ist bei grösseren Datenmengen wesentlich. Man denke z. B. an (1000 x 1000)-Matrizen in der Statistik, z.B. bei 1000 Beobachtungen von 1000 Variablen.
▮ Attention: *Mathematica* n'est pas conçu pour le "management" ni pour la manipulation des données. On peut lire et intérpréter ("parser") de même que manipuler beaucoup plus rapidement les données à l'aide de langages "datamanagement" ou programmes (programmation low-level). Cela est essentiel pour des quantités plus importantes de données. On pense p.ex. aux matrices (1000 x 1000) dans la statistique, p.ex. 1000 observations de 1000 variables.

14.1.4. Externe Kommandos bis exklusive Version 3.0 (für WIN98 nicht zu empfehlen)
▮ Commandements extérieurs jusqu'à version 3.0, exclusive (pour WIN98 pas recommandé)

Mit "!Kommando" kann ein externes Kommando ausgeführt werden. Leider haben wir auf NeXT keinen ModulaII-Compiler oder Java-Compiler, so dass wir selbst solche Kommandos bauen könnten. Eine andere Programmiersprache, die den Zweck erfüllen würde, wird an der Schule momentan nicht unterrichtet. Hier ist für NeXT das Programm "aa_ascii", das Umlaute in TeX-Format übersetzt. Studiere das Beispiel:

▮ Par "commandement !" on peut exécuter un commandement extérieur. Malheureusement nous n'avons pas de compilateur pour ModulaII ou Jave sur NeXT pour pouvoir construire de tels commandements. Une autre langue de programmation qui serve pour le même but n'est actuellement pas enseigné à cette école. Ici on utilise pour NeXT le programme "aa_ascii" qui transforme les "Umlaute" (métaphonie, voyelle infléchie) de l'allemand dans le format TeX. Etudie l'exemple:

```
In[119]:=
        !!C:\\work\\MathematicaData\\AABDaten

        "3 4 5 1 2 3 6 6.3 7.8 4 r s t hallo  7777 6666 4.321 1.11111, {1,2,3,4}"

In[120]:=
        (* !!f:\Mathe/Daten/spsheet2 *)

In[121]:=
        !C:\\work\\MathematicaData\\AABDaten

In[122]:=
        (* !f:\Mathe/Daten/aa_ascii
                f:\Mathe/Daten/spsheet2
                f:\Mathe/Daten/tex -tex;
        ReadList["f:\Mathe/Daten/tex",
                {Word, Number, Number, Number},
                 WordSeparators -> {" "},
                 RecordLists ->True] *)
```

Das "!Kommando" kann auch in ReadList stehen, falls es sich auf eine Operation auf einem File bezieht (und nicht zwei Files aufruft)! Folgendes geht nicht:

▮ Le "commandement !" peut se trouver aussi dans ReadList, s'il se rapporte à une opération sur un fichier (et n'appelle pas deux fichiers)! La chose suivante ne va pas:

```
In[123]:=
        (* ReadList["!f:\Mathe/Daten/aa_ascii
                f:\Mathe/Daten/spsheet2
                f:\Mathe/Daten/tex -tex ",
                {Word, Number, Number, Number},
                 WordSeparators -> {" "},
                 RecordLists ->True] *)
```

Externe Kommandos ab Version 3.0
▮ Commandements extérieurs à partir de version 3.0

Verwendung externer Programme für Datenmanipulationen von *Mathematica* aus mit Hilfe der Befehle "Install", "Links", "MLPut...",
"MLGet...", u.s.w. ..... .
Momentan sind keine Programme vorhanden, die problemlos auf irgendwelchen hier unbekannten Systemen laufen. Daher wird auf ein konkretes Beispiel verzichtet. Probiere daher mit eigenen Programmen.

▮ Utiliser des Programmes externes pour manipuler des données depuis *Mathematica* à l'aide des ordres "Install", "Links", "MLPut...",
"MLGet...", etc. .... .
Dans ce moment nous n'avons pas de programmes qui tournent sans problèmes sur n'importe quel système inconnu chez nous. C'est pourquoi nous renonçons à présenter un exemple concret. Essaie avec tes propres programmes.

14.2. Datenexport
▮ Exportation de données

14.2.1. Zeichenweise in ein File schreiben
▮ Ecrire les signes l'un après l'autre dans un fichier

Beispiel
▮ Exemple

Zuerst Daten erzeugen, die exportiert werden sollen:
▮ D'abord générer les données qui doivent être exportées:

```
In[124]:=
Clear[daten];
daten = Table[Random[Integer,{1,9}],{2},{6}]
```

```
Out[125]=
{{3, 1, 3, 2, 2, 3}, {9, 2, 2, 3, 3, 4}}
```

Studiere die Befehle:
▮ Etudie les ordres:

```
In[126]:=
?>>
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

```
In[127]:=
?Put
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

In[128]:=
```
??WriteString
```

```
WriteString[channel, expr1, expr2, ... ] converts the expri to strings,
   and then writes them in sequence to the specified output channel. Mehr…

Attributes[WriteString] = {Protected}
```

In[129]:=
```
?OpenWrite
```

```
OpenWrite["file"] opens a file to write output to it, and returns an OutputStream object.
 Mehr…
```

In[130]:=
```
?Close
```

```
Close[stream] closes a stream. Mehr…
```

In[131]:=
```
?Scan
```

```
Scan[f, expr] evaluates f applied to each element of expr in turn. Scan[f,
   expr, levelspec] applies f to parts of expr specified by levelspec. Mehr…
```

In[132]:=
```
?Rest
```

```
Rest[expr] gives expr with the first element removed. Mehr…
```

Zum Exportieren wollen wir eine Funktion definieren, die die Generierte in der Art Matrix "Zeilenvektor auf Zeile ... (Record) im File" schreibt:

■ Nous allons définir une fonction pour exporter, qui écrit la générée dans la matrice dans le sens "vecteur de ligne sur ligne .... (Record) dans le fichier":

In[133]:=
```
Remove[schreibMatrix];
Off[neufile];
schreibMatrix[fileName_String, data_List] :=
    Block[{ neufile = OpenWrite[fileName]},
        Scan[(WriteString[neufile,First[#]];
                  Scan[WriteString[neufile,
                  "\t",#]&,Rest[#]];
             WriteString[neufile,"\n"])&,
             data];
        Close[neufile]];
```

In[136]:=
```
daten = {{1, 2}, {3, 4, 5, 6}, {7, 8, 9, 10, a}, {b}}
```

Out[136]=
```
{{1, 2}, {3, 4, 5, 6}, {7, 8, 9, 10, a}, {b}}
```

In[137]:=
```
schreibMatrix["C:\\work\\MathematicaData\\AAHDaten", daten]
```

Out[137]=
```
C:\work\MathematicaData\AAHDaten
```

In[138]:=
```
(* schreibMatrix["f:\Mathe/Daten/neuDaten", daten] (* win3.1 etc. *) *)
```

In[139]:=

```
!! "C:\\work\\MathematicaData\\AAHDaten"
```

```
1   2
3   4   5   6
7   8   9   10  a
b
```

In[140]:=

```
(* !!f:\Mathe/Daten/neuDaten *) (* win3.1 etc. *)
```

In[141]:=

```
?schreibMatrix
```

```
Global`schreibMatrix

schreibMatrix[fileName_String, data_List] :=
 Block[{neufile = OpenWrite[fileName]}, Scan[(WriteString[neufile, First[#1]];
     Scan[WriteString[neufile,     , #1] &, Rest[#1]]; WriteString[neufile,
])&, data]; Close[neufile]]
```

14.2.2. Abspeichern von Definitionen
▮ Mémoriser des définitions

Beispiel
▮ Exemple

Studiere: ▮ Etudie:

In[142]:=

```
?Save
```

```
Save["filename", symbol] appends definitions associated with the specified symbol to a
   file. Save["filename", "form"] appends definitions associated with all symbols
   whose names match the string pattern form. Save["filename", "context`"] appends
   definitions associated with all symbols in the specified context. Save["filename",
   {object1, object2, ... }] appends definitions associated with several objects. Mehr...
```

In[143]:=

```
f[x_] := Sin[x^2] / (E^(-x) Cos[x]);
Save["C:\\work\\MathematicaData\\AA_f", f]
```

In[145]:=

```
(* f[x_] := Sin[x^2]/(E^(-x) Cos[x]);
Save["f:\Mathe/Daten/AAA_f", f]
                          (* win3.1 etc. *) *)
```

In[146]:=

```
!! C:\\work\\MathematicaData\\AA_f
```

```
f[x_] := Sin[x^2]/(Cos[x]/E^x)

x = {{"dddt  ejkrjke   dfjj  asdf  \\t dkfjke \\t jkdf"}, {"   "}}
f[x_] := Sin[x^2]/(Cos[x]/E^x)

x = {{"dddt  ejkrjke   dfjj  asdf  \\t dkfjke \\t jkdf"}, {"   "}}
f[x_] := Sin[x^2]/(Cos[x]/E^x)

x = {{"asdf 5  34 dkfjke jkdf"}, {"   "}}
```

In[147]:=
```
(* !!f:\Mathe/Daten/AAA_f  (* win3.1 etc. *) *)
```

14.2.3.Abspeichern von Output
▮ Mémoriser l'output

Studiere: ▮ Etudie:

In[148]:=
```
?>
```

```
x > y yields True if x is determined to be greater than y. x1 >
   x2 > x3 yields True if the xi form a strictly decreasing sequence. Mehr…
```

In[149]:=
```
?>>
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

In[150]:=
```
?>>>
```

```
expr >>> filename appends expr to a file. PutAppend[expr1,
   expr2, ... , "filename"] appends a sequence of expressions expri to a file. Mehr…
```

In[151]:=
```
?Put
```

```
expr >> filename writes expr to a file. Put[expr1,
   expr2, ... , "filename"] writes a sequence of expressions expri to a file. Mehr…
```

Probiere aus: ▮ Essaie:

In[152]:=
```
Table[f[n], {n, 20}] >>
    C:\\work\\MathematicaData\\AAneuDaten
```

In[153]:=
```
(* Table[f[n], {n, 20}] >>
    f:\Mathe/Daten/neuDaten  (* win3.1 etc. *) *)
```

In[154]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten
```

```
{E*Tan[1], E^2*Sec[2]*Sin[4], E^3*Sec[3]*Sin[9], E^4*Sec[4]*Sin[16],
 E^5*Sec[5]*Sin[25], E^6*Sec[6]*Sin[36], E^7*Sec[7]*Sin[49],
 E^8*Sec[8]*Sin[64], E^9*Sec[9]*Sin[81], E^10*Sec[10]*Sin[100],
 E^11*Sec[11]*Sin[121], E^12*Sec[12]*Sin[144], E^13*Sec[13]*Sin[169],
 E^14*Sec[14]*Sin[196], E^15*Sec[15]*Sin[225], E^16*Sec[16]*Sin[256],
 E^17*Sec[17]*Sin[289], E^18*Sec[18]*Sin[324], E^19*Sec[19]*Sin[361],
 E^20*Sec[20]*Sin[400]}
```

In[155]:=
```
(* !!f:\Mathe/Daten/neuDaten   (* win3.1 etc. *) *)
```

In[156]:=
```
Put[Table[f[n], {n, 20}], Sin[x], "Hallo",
 "C:\\work\\MathematicaData\\AAneuDaten"]
```

In[157]:=
```
(* Put[Table[f[n], {n, 20}], Sin[x], "Hallo",
 "f:\Mathe/Daten/neuDaten"] *)  (* win3.1 etc. *)
```

In[158]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten
```

```
{E*Tan[1], E^2*Sec[2]*Sin[4], E^3*Sec[3]*Sin[9], E^4*Sec[4]*Sin[16],
 E^5*Sec[5]*Sin[25], E^6*Sec[6]*Sin[36], E^7*Sec[7]*Sin[49],
 E^8*Sec[8]*Sin[64], E^9*Sec[9]*Sin[81], E^10*Sec[10]*Sin[100],
 E^11*Sec[11]*Sin[121], E^12*Sec[12]*Sin[144], E^13*Sec[13]*Sin[169],
 E^14*Sec[14]*Sin[196], E^15*Sec[15]*Sin[225], E^16*Sec[16]*Sin[256],
 E^17*Sec[17]*Sin[289], E^18*Sec[18]*Sin[324], E^19*Sec[19]*Sin[361],
 E^20*Sec[20]*Sin[400]}
{{Sin["asdf 5  34 dkfjke jkdf"]}, {Sin["   "]}}
"Hallo"
```

In[159]:=
```
(* !!f:\Mathe/Daten/neuDaten *)  (* win3.1 etc. *)
```

Was ist mit den Daten von 14.2.3 passiert?
Aue s'est-il passé avec les données de 14.2.3?

In[160]:=
```
2.^0.5
```

Out[160]=
```
1.41421
```

In[161]:=
```
2.^0.5 >> C:\\work\\MathematicaData\\AAneuDaten1
```

In[162]:=
```
(* 2.^0.5 >> f:\Mathe/Daten/neuDaten  *)
                                (* win3.1 etc. *)
```

In[164]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten1
```

```
1.4142135623730951
```

In[165]:=
```
(* !!f:\Mathe/Daten/neuDaten *)   (* win3.1 etc. *)
```

Wieso ist das abgespeicherte Resultat genauer als das auf dem Schirm ausgegebene Resultat?
- Pourquoi le résultat mémorisé est-il plus exact que le résultat sorti sur l'écran?

14.3. Manipulation von Strings
Manipulation de Strings

Studiere die folgenden Befehle:
- Etudie les ordres suivants:

In[166]:=
```
?Read
```

```
Read[stream] reads one expression from an input stream, and returns the expression.
   Read[stream, type] reads one object of the specified type. Read[stream, {type1,
   type2, ... }] reads a sequence of objects of the specified types. Mehr...
```

In[167]:=

?ReadList

ReadList["file"] reads all the remaining expressions in a file, and returns a list of them. ReadList["file", type] reads objects of the specified type from a file, until the end of the file is reached. The list of objects read is returned. ReadList["file", {type1, type2, ... }] reads objects with a sequence of types, until the end of the file is reached. ReadList["file", types, n] reads only the first n objects of the specified types. Mehr...

In[168]:=

?OpenAppend

OpenAppend["file"] opens a file to append output to it, and returns an OutputStream object. Mehr...

In[169]:=

?OpenRead

OpenRead["file"] opens a file to read data from, and returns an InputStream object. Mehr...

In[170]:=

?StringDrop

StringDrop["string", n] gives "string" with its first n characters dropped. StringDrop["string", -n] gives "string" with its last n characters dropped. StringDrop["string", {n}] gives "string" with its nth character dropped. StringDrop["string", {m, n}] gives "string" with characters m through n dropped. StringDrop[{s1, s2, ... }, spec] gives the list of results for each of the si. Mehr...

In[171]:=

?StringInsert

StringInsert["string", "snew", n] yields a string with "snew" inserted starting at position n in "string". StringInsert["string", "snew", -n] inserts at position n from the end of "string". StringInsert["string", "snew", {n1, n2, ... }] inserts a copy of "snew" at each of the positions ni. StringInsert[{s1, s2, ... }, "snew", n] gives the list of results for each of the si. Mehr...

In[172]:=

?StringJoin

"s1" <> "s2" <> ... , StringJoin["s1", "s2", ...] or StringJoin[{"s1", "s2", ... }] yields a string consisting of a concatenation of the si. Mehr...

In[173]:=

?StringReplace

StringReplace["string", s -> sp] or StringReplace["string", {s1 -> sp1, s2 -> sp2, ... }] replaces the string expressions si by spi whenever they appear as substrings of "string". StringReplace["string", srules, n] does only the first n replacements. StringReplace[{s1, s2, ... }, srules] gives the list of results for each of the si. Mehr...

In[174]:=

?StringReverse

StringReverse["string"] reverses the order of the characters in "string". Mehr...

In[175]:=

?StringTake

StringTake["string", n] gives a string containing the first n
characters in "string". StringTake["string", -n] gives the last n characters
in "string". StringTake["string", {n}] gives the nth character in "string".
StringTake["string", {m, n}] gives characters m through n in "string". StringTake[
{s1, s2, ... }, spec] gives the list of results for each of the si. Mehr...

In[176]:=

?StringLength

StringLength["string"] gives the number of characters in a string. Mehr...

In[177]:=

?StringPosition

StringPosition["string", "sub"] gives a list of the starting and ending character positions
at which "sub" appears as a substring of "string". StringPosition["string", patt] gives
all positions at which substrings matching the general string expression patt appear
in "string". StringPosition["string", patt, n] includes only the first n occurrences of
patt. StringPosition["string", {patt1, patt2, ... }] gives positions of all the patti.
StringPosition[{s1, s2, ... }, p] gives the list of results for each of the si. Mehr...

In[178]:=

?StringMatchQ

StringMatchQ["string", patt] tests whether string matches the
string pattern patt. StringMatchQ["string", RegularExpression["regex"]]
tests whether string matches the specified regular expression. StringMatchQ[
{s1, s2, ... }, p] gives the list of results for each of the si. Mehr...

In[179]:=

?StringQ

StringQ[expr] gives True if expr is a string, and False otherwise.

In[180]:=

?StringToStream

StringToStream["string"] opens an input stream for reading from a string. Mehr...

In[181]:=

?WriteString

WriteString[channel, expr1, expr2, ...] converts the expri to strings,
and then writes them in sequence to the specified output channel. Mehr...

In[182]:=

?ToString

ToString[expr] gives a string corresponding to
the printed form of expr in OutputForm. ToString[expr, form] gives
the string corresponding to output in the specified form. Mehr...

Probiere aus: ■ Essaie:

In[183]:=

```
Table[StringJoin["daten.", ToString[i]],{i,6}]
```

Out[183]=

{daten.1, daten.2, daten.3, daten.4, daten.5, daten.6}

14.4. Eingabe- und Ausgabeform
▮ Formes d'entrée et de sortie

Studiere die folgenden Befehle:
▮ Etudie les ordres suivants:

In[184]:=
```
?InputForm
```
```
InputForm[expr] prints as a version of expr suitable for input to Mathematica. Mehr...
```

In[185]:=
```
?OutputForm
```
```
OutputForm[expr] prints as a two-dimensional
   representation of expr using only keyboard characters. Mehr...
```

Beispiele: ▮ Exemples:

In[186]:=
```
Clear[a,x,y];
a = x^5/x^2 y +x^y^5
```

Out[187]=
$$x^{y^5} + x^3\, y$$

In[188]:=
```
InputForm[a]
```

Out[188]//InputForm=
```
x^y^5 + x^3*y
```

In[189]:=
```
OutputForm[a]
```

Out[189]//OutputForm=
```
         5
        y
x^3 + x   y
```

In[190]:=
```
a >> C:\\work\\MathematicaData\\AAneuDaten2
```

In[191]:=
```
(* a >> f:\Mathe/Daten/neuDaten *)  (* win3.1 etc. *)
```

In[192]:=
```
!! C:\\work\\MathematicaData\\AAneuDaten2
```
```
x^y^5 + x^3*y
```

In[193]:=
```
(* !!f:\Mathe/Daten/neuDaten *)   (* win3.1 etc. *)
```

In[194]:=
```
Read["C:\\work\\MathematicaData\\AAneuDaten2"]
```

Out[194]=
$$x^{y^5} + x^3\, y$$

```
In[195]:=
(* Read["f:\Mathe/Daten/neuDaten"](* win3.1 etc. *) *)

In[196]:=
b = Read["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[196]=
EndOfFile

In[197]:=
(* b = Read["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)

In[199]:=
b

Out[199]=
EndOfFile

In[200]:=
!! C:\\work\\MathematicaData\\AAneuDaten2

x^y^5 + x^3*y

In[201]:=
(* !!f:\Mathe/Daten/neuDaten *)   (* win3.1 etc. *)

In[202]:=
Close["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[202]=
C:\work\MathematicaData\AAneuDaten2

In[203]:=
(* Close["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)

In[205]:=
b = Read["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[205]=
x^(y^5) + x^3 y

In[206]:=
(* b = Read["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)

In[208]:=
Close["C:\\work\\MathematicaData\\AAneuDaten2"]

Out[208]=
C:\work\MathematicaData\AAneuDaten2

In[209]:=
(* Close["f:\Mathe/Daten/neuDaten"] *)
                                (* win3.1 etc. *)
```

14.5. Uebersetzen von Ausdrücken in andere Programmiersprachen und Manipulation von Sourcecode

∎ Traduire des expressions en d'autres languages de programmation et manipulation de sourcecode

Uebersetzungen
∎ Tradcuction

Studiere die folgenden Befehle:
∎ Etudie les ordres suivants:

In[211]:=
```
?CForm
```
```
CForm[expr] prints as a C language version of expr. Mehr...
```

In[212]:=
```
?FortranForm
```
```
FortranForm[expr] prints as a Fortran language version of expr. Mehr...
```

In[213]:=
```
?TeXForm
```
```
TeXForm[expr] prints as a TeX version of expr. Mehr...
```

In[214]:=
```
?Splice
```
```
Splice["file"] splices Mathematica output into an external file. It takes text
   enclosed between (* and *) in the file, evaluates the text as Mathematica
   input, and replaces the text with the resulting Mathematica output. Mehr...
```

Beispiele: ∎ Exemple:

In[215]:=
```
Clear[a];
a = x^5/z^2 y +x^y^5 +
                ( x^2 Sin[x^5/z^2 y +x^y^5]);
```

In[217]:=
```
CForm[a]
```

Out[217]//CForm=
```
Power(x,Power(y,5)) + (Power(x,5)*y)/Power(z,2) +
   Power(x,2)*Sin(Power(x,Power(y,5)) +
      (Power(x,5)*y)/Power(z,2))
```

In[218]:=
```
FortranForm[a]
```

Out[218]//FortranForm=
```
            x**y**5 + (x**5*y)/z**2 +
          -  x**2*Sin(x**y**5 + (x**5*y)/z**2)
```

In[219]:=
```
TeXForm[a]
```

Out[219]//TeXForm=
```
x^{y^5}+\frac{y x^5}{z^2}+\sin
   \left(x^{y^5}+\frac{y x^5}{z^2}\right) x^2
```

Manipulation von Sourcecode
▮ Manipulation de sourcecode

Im folgenden Beispiel wird gezeigt, wie in den Source-Code eines externen Programms (z.B. in C) direkt *Mathematica*-Ausdrücke eingesetzt und dann von *Mathematica* übersetzt werden kŏnnen. Der externe Programmteil befindet sich im File AAA.mc. Das übersetzte Programm wird in AAA.c gespeichert!!!!!!!

▮ Dans l'exemple suivant on montre comme on peut mettre des expressions de *Mathematica* directement dans le Source-Code d'un autre programme extérieur (p.ex. dand C), ensuite ses expressions peuvent être traduites par *Mathematica*. La partie extérieur du programme se trouve dans le fichier AAA.mx resp. AAA.mc. Le programme traduit est mémorisé dans AAA.x resp. AAA.c !!!!!!!

In[220]:=
```
Sin[3] + Cos[3] >> C:\\work\\MathematicaData\\AAA.mx
```

In[221]:=
```
!! C:\\work\\MathematicaData\\AAA.mx

Cos[3] + Sin[3]
```

In[222]:=
```
(* !!f:\Mathe/Daten/AAA.mc  *)  (* win3.1 etc. *)
```

In[223]:=
```
?Splice

Splice["file"] splices Mathematica output into an external file. It takes text
   enclosed between <* and *> in the file, evaluates the text as Mathematica
   input, and replaces the text with the resulting Mathematica output. Mehr...
```

In[224]:=
```
Splice["C:\\work\\MathematicaData\\AAA.mx"]
```

Out[224]=
```
C:\work\MathematicaData\AAA.mx
```

In[225]:=
```
(* Splice["f:\Mathe/Daten/AAA.mc"]  *)
                                (* win3.1 etc. *)
```

In[227]:=
```
!! C:\\work\\MathematicaData\\AAA.mx

Cos[3] + Sin[3]
```

In[228]:=
```
(* !!f:\Mathe/Daten/AAA.c *) (* win3.1 etc. *)
```

14.6. Formate
Formats

Allgemeines
Généralités

Studiere die folgenden Befehle:
Etudie les ordres suivants:

```
In[229]:=
?Subscripted
```

```
System`Subscripted
```

```
In[230]:=
?Superscript
```

```
Superscript[x, y] is an object that formats as x with a superscript y. Mehr...
```

```
In[231]:=
?SequenceForm
```

```
SequenceForm[expr1, expr2, ... ] prints as the
   textual concatenation of the printed forms of the expri. Mehr...
```

Probiere aus: Etudie:

```
In[232]:=
Subscripted[abc[145]]
```

```
Out[232]=
```

$$abc_{145}$$

```
In[233]:=
SequenceForm[uvw, Superscript[251]]
```

```
Out[233]=
```

$$uvw^{251}$$

Ein Anwendungsbeispiel
Un exemple d'application

```
In[234]:=
Remove[a, n, m];
Format[a[n_, m_]] := Subscripted[a[n, m]];
Array[a, {2, 2}] // MatrixForm
```

```
Out[236]//MatrixForm=
```

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[237]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[238]:=
```
Remove["Global`*"]
```

Uebungen ▮ Exercices

14. Input und Output
▮ Input et output

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 14 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 14.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Studiere die nachfolgenden Beispiele und den Befehl Input:
▮ Etudie les exemples suivants et l'ordre Input

```
In[1]:= Remove[MeinReadList];
        MeinReadList[fileName_String,
                thing_:Expression]:=
            Module[{file, expr, list = {}},
                file = OpenRead[fileName];
                If[ file === $Failed, Return[list] ];
                While[ True,
                    expr = Read[file, thing];
                    If[ expr === EndOfFile, Break[] ];
                    AppendTo[list, expr]
                ];
                Close[file];
                list
            ];
```

```
General::spell1 :
 Possible spelling error: new symbol name "file" is similar to existing symbol "File". Mehr…

General::spell1 :
 Possible spelling error: new symbol name "list" is similar to existing symbol "List". Mehr…
```

In[3]:=
```
MeinReadList["C:\\work\\MathematicaData\\AAADaten", Number]
```

Out[3]=
```
{1, 2, 3, 4, 5, 5, 6, 7, 6, 4, 12, 34, 123, 34, 44}
```

In[4]:=
```
?Expression
```

```
Expression is a symbol that represents an
   ordinary Mathematica expression in Read and related functions. Mehr...
```

In[5]:=
```
?OpenRead
```

```
OpenRead["file"] opens a file to read data from, and returns an InputStream object. Mehr...
```

In[6]:=
```
(* MeinReadList["C:\\work\\MathematicaData\\AAAdaten", Number]   *)
```

In[7]:=
```
Remove[MeinReadList];
MeinReadList[thing_:Expression] :=
    Module[{file, fileName, expr, list = {}},
        fileName = Input["Bitte einzulesende
        Datei in Anführungszeichen angeben.
        (Das File C:\work\AAFdaten.txt ...
        sollte vorhanden sein.) -- S.v.p
        enterer le fichier à lire entre
        parenthèses.(Le fichier
        C:\work\AAFdaten.txt doit exister.)"];

        file = OpenRead[fileName];
        If[ file === $Failed, Return[list] ];
        While[ True,
            expr = Read[file, thing];
            If[ expr === EndOfFile, Break[] ];
            AppendTo[list, expr]
        ];
        Close[file];
        list
    ];
    MeinReadList[Number]
```

```
Syntax::stresc : Unknown string escape \w. Mehr…

Syntax::stresc : Unknown string escape \A. Mehr…

Syntax::stresc : Unknown string escape \w. Mehr…

Syntax::stresc : Unknown string escape \A. Mehr…
```

Out[9]=
```
{1, 2, 3, 4, 5, 5, 6, 7, 6, 4, 12, 34, 123, 34, 44}
```

Aufgabe 2 ▮ Problème 2

Schreibe eine Funktion, mit der sich das Multiplizieren der Zahlen von 1 bis 12 im Kopf trainieren und korrigieren lässt!

▮ Ecris une fonction par laquelle on peut exercer et corriger le calcul mental de la multiplication des nombres de 1 à 12 !

Beispiel einer Lösung:
Exemple d'une solution:

```
In[10]:= Remove[mult];
         mult:=
             Module[{faktor1, faktor2,
                     korrektur, resultat},

                     Clear[resultat];
                     resultat = ".";

               While[ resultat != "y" || resultat !
                     = "Y" || resultat != "<y>" ||
                     resultat != "<Y>", faktor1 =
                     Random[Integer,{1,12}]; faktor2
                     = Random[Integer,{1,12}];
                     korrektur = faktor1 faktor2;
                     Print["Was gibt ",faktor1,
                     " mal ",faktor2,"?"];
                     resultat = Input[
             "Bitte Resultat eingeben, oder beenden
             mit dem Buchstaben y. -- S.V.P entrer le
             résultat ou terminer à l'aide de y."];
                  If[ resultat == korrektur,
                         Print[resultat,
                         " ist richtig, bravo! ---
                         est juste, bravissimo!"],
                         Print[resultat,
                         "? - Richtig ist ---
                         résultat juste:",korrektur]
                    ]
                  ];
                  Print["Ende"]
             ];
          mult

         Set::write : Tag Or in resultat$32 ≠ y || resultat$32! is Protected. Mehr…

         Was gibt 1 mal 1?

         Set::write : Tag Or in resultat$32 ≠ y || resultat$32! is Protected. Mehr…

         Ende
```

In[13]:= **mult**

```
Set::write : Tag Or in resultat$35 ≠ y || resultat$35! is Protected. Mehr…

General::stop : Further output of Set::write will be suppressed during this calculation. Mehr…

Was gibt 9 mal 8?

72 ist richtig, bravo! ---           est juste, bravissimo!

Was gibt 12 mal 1?

12 ist richtig, bravo! ---           est juste, bravissimo!

Was gibt 7 mal 1?

7 ist richtig, bravo! ---           est juste, bravissimo!

Was gibt 9 mal 11?

99 ist richtig, bravo! ---           est juste, bravissimo!

Was gibt 8 mal 6?

48 ist richtig, bravo! ---           est juste, bravissimo!

Was gibt 8 mal 6?

Ende
```

Aufgabe 3 ■ Problème 3

Uebersetze die folgenden Ausdrücke mit Hilfe von *Mathematica* in C und in Fortran:
■ Traduis les expressions suivantes à l'aide de *Mathematica* dans C et dans Fortran:
a) D[z Exp[z]
b) a^x
c) Sin[Pi/5]

Lösung: ■ Solution:

In[14]:=
```
Map[{CForm[#], FortranForm[#]}&,
 {D[z Exp[z], z], a^x, Sin[Pi/5]}]
```

Out[14]=
```
{{Power(E, z) + Power(E, z) * z, E ** z + E ** z * z}, {Power(a, x), a ** x},
 {Sqrt((5 - Sqrt(5)) / 2.) / 2., Sqrt((5 - Sqrt(5)) / 2.) / 2.}}
```

Aufgabe 4 ■ Problème 4

Schreibe ein C-Programm, ein Fortran-Programm oder ein TeX-Programm, in dem Splice zur Anwendung kommt.
■ Ecris un programme C, un programme Fortran ou un Programme TeX, dans lequel on applique Splice

Ein Beispiel: Manipulation von Sourcecode bei TeX
Un exemple: Manipulation de Sourcecode dans TeX

Im folgenden Beispiel wird gezeigt, wie in den Source-Code eines externen Programms (hier in TeX) direkt *Mathematica*-Ausdrücke eingesetzt und dann von *Mathematica* übersetzt werden kŏnnen. Der externe Programmteil befindet sich im File C:\work\AAATeX.txt. Das übersetzte Programm wird in C:\work\AAA.tex gespeichert !!!!!!!

Dans l'exemple suivant on montre comme on peut mettre directement des expressions de *Mathematica* dans le sorce-code d'un programme extérieur (ici dans TeX), ensuite elles peuvent être traduites par *Mathematica*. La partie extérieure du programme se trouve dans le fichier C:\work\AAATeX.txt. Le programme traduit dans le fichier C:\work\-AAA.tex !!!!!!!!

```
In[15]:= TeXForm[Integrate[Sin[2 x + 3],x];
            Sin[x] == 4 / Tan[y];
            Print["Hallo!"]]

         Hallo!

Out[15]//TeXForm=
        \text{Null}

In[16]:= TeXForm[Integrate[Sin[2 x + 3],x]]

Out[16]//TeXForm=
        -\frac{1}{2} \cos (2 x+3)

In[17]:= TeXForm[Integrate[Sin[2 x + 3], x]] >> C:\\work\\MathematicaData\\AAA1TeX.txt

In[18]:= !!C:\\work\\MathematicaData\\AAA1TeX.txt

        -\frac{1}{2} \cos (2 x+3)

In[19]:= CopyFile["C:\\work\\MathematicaData\\AAA1TeX.txt",
           "C:\\work\\MathematicaData\\AAA3TeX.mtex"]

Out[19]= C:\work\MathematicaData\AAA3TeX.mtex

In[20]:= !! C:\\work\\MathematicaData\\AAA1TeX.mtex

        -\frac{1}{2} \cos (2 x+3)

In[21]:= TeXForm[MatrixForm[{{1, 2}, {3, 4}}]] >> C:\\work\\MathematicaData\\AAA2TeX.mtex

In[22]:= !! C:\\work\\MathematicaData\\AAA2TeX.mtex

        \left(
        \begin{array}{ll}
         1 & 2 \\
         3 & 4
        \end{array}
        \right)

In[23]:= Splice["C:\\work\\MathematicaData\\AAA2TeX.mtex"]

Out[23]= C:\work\MathematicaData\AAA2TeX.mtex

In[24]:= (*Splice["f:\Mathe/Daten/AAA.mtex"]*) (*win31...*)
```

In[25]:= `!! C:\\work\\MathematicaData\\AAA2TeX.mtex`

```
\left(
\begin{array}{ll}
 1 & 2 \\
 3 & 4
\end{array}
\right)
```

In[26]:= `(*!!f:\Mathe/Daten/AAA.tex*) (*win31...*)`

Aufgabe 5 ■ Problème 5

Gegeben sei b[c, d]. Schreibe einen Befehl, der c als Subscript und d als Superscript wiedergibt!
■ Soit donné b[c, d]. Ecris un ordre qui rend c comme subscript et d comme superscript.

Eine Lðsung: ■ Une solution:

In[27]:= `?Format`

```
Format[expr] prints as the formatted form of expr. Assigning
   values to Format[expr] defines print forms for expressions. Format[
   expr, form] gives a format for the specified form of output. Mehr...
```

In[28]:=
```
Format[b[c_,d_]] :=
  SequenceForm[Subscripted[b[c]],
  Superscript[d]];
 b[c, d]
```

Out[29]= b_c^d

In[30]:= `b[3, 5]`

Out[30]= b_3^5

In[31]:= `c[3, 5]`

Out[31]= `c[3, 5]`

Was ist passiert?
■ Que s'est-il passé?

Aufgabe 6 ■ Problème 6

Schreibe eine Funktion zur Berechnung der Glieder der Fibonacci-Folge bis zum n-ten Glied und speichere die Glieder sowie die Funktion in ein File.
■ Ecris une fonction pour calculer les membres de la suite de Fibonacci jusqu'au n-ième membre et mémorise les membres ainsi que la fonction dans un fichier.

Beispiel für das Programm
▮ Exemple pour le programme

```
In[32]:= Remove[fib];
         fib[0]  = 1;
         fib[1]  = 1;
         fib[n_Integer]  = fib[n-1]+fib[n-2];
         fib[n_]  = 0 ;
         ?fib

        Global`fib

         fib[0] = 1

         fib[1] = 1

         fib[n_Integer] = fib[-2 + n] + fib[-1 + n]

         fib[n_] = 0

In[38]:= {fib[0],fib[1],fib[2],fib[3],fib[4],fib[5],
           "Test",fib[1.3]}

Out[38]= {1, 1, 2, 3, 5, 8, Test, 0}

In[39]:= fib[10]

Out[39]= 89
```

Speichern der Funktion
▮ Mémoriser la fonction

```
In[40]:= Save["C:\\work\\MathematicaData\\AAAfib.txt", fib];

In[41]:= << C:\\work\\MathematicaData\\AAAfib.txt

Out[41]= 0

In[42]:= Save["C:\\work\\MathematicaData\\AAAfib",fib];
```

In[43]:= **!!C:\\work\\MathematicaData\\AAAfib**

```
fib[0] = 1

fib[1] = 1

fib[n_Integer] = 0

fib[n_] = 0
fib[0] = 1

fib[1] = 1

fib[n_Integer] = 0

fib[n_] = 0
fib[0] = 1

fib[1] = 1

fib[n_Integer] = 0

fib[n_] = 0
fib[0] = 1

fib[1] = 1

fib[n_Integer] = 0

fib[n_] = 0
```

In[44]:= **<< C:\\work\\MathematicaData\\AAAfib**

Out[44]= 0

In[45]:=
```
(*Save["f:\Mathe/Daten/AAA_fib", fib];
!!f:\Mathe/Daten/AAA_fib*)   (*win31...*)
```

Erst rechnen und dann speichern
D'abord calculer et ensuite mémoriser

In[46]:= **Table[fib[n], {n, 0, 10}] >> C:\\work\\MathematicaData\\AAAfib1**

In[47]:= **!! C:\\work\\MathematicaData\\AAAfib1**

```
{1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

In[48]:=
```
(*Table[fib[n], {n, 0, 10}] >>
"f:\Mathe/Daten/AAA_fib";
!!f:\Mathe/Daten/AAA_fib*) (*win31...*)
```

Ein Programm zum direkten Speichern der Glieder
▮ Un programme pour mémoriser directement les menbres

In[49]:=
```
Remove[schreibFib];

schreibFib[fileName_String, zahl_Integer] :=
    Block[{ neufile = OpenWrite[fileName],
        n = 0},
        While[n<=zahl,
        WriteString[neufile,fib[n]];
        WriteString[neufile,"\t"];
        n++];
        (*WriteString[neufile,"\t"];*)
        Close[neufile]];
```

In[51]:=
```
schreibFib["C:\\work\\MathematicaData\\fib2", 12];
```

In[52]:=
```
!! C:\\work\\MathematicaData\\fib2
```

```
1   1   0   0   0   0   0   0   0   0   0   0   0
```

"Putzmaschine" einsetzen
▮ Employer la "machine de nettoyage"

In[53]:=
```
(* Old Form: Remove["Global`@*"] *)
```

In[54]:=
```
Remove["Global`*"]
```

Kurs ▮ Cours

15. Packages

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 15 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 15.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Problem: Externe Programme und Programmpackete schreiben.
Mathematica selbst ist in C geschrieben. Externe Programme, sogenannte Packages, sind aber in Mathematica selbst geschrieben. Bsp.: "Calculus`DSolveIntegrals`"
▮ Problème: Ecris les programmes extérieurs et des paquets de programmes. *Mathematica* est écrit en C. Les programmes extérieurs, les dits Packages, sont écrits dans *Mathematica* même. Ex.:
"Calculus`DSolveIntegrals`"

15.1. Wieso Packages? Wie zugreifen?
▮ Pourquoi des Packages? Comment les saisir?

15.1.1. Wieso Packages?
▮ Pourquoi des Packages?

Es ist klar, dass man häufig gebrauchte Funktionen oder Prozeduren etc. nicht immer wieder neu schreibt. Andere möchten auch gerne profitieren. So schreibt man eben externe Programme, die nach Bedarf zuladbar sind.
▮ Il est clair qu'on n'écrit pas chaque fois à nouveau des fonctions ou des procédures ets. fréquentes. Les autres aimeraient aussi en profiter. Ainsi on écrit des programmes extérieurs qu'on peut ajouter quand on en a besoin.

15.1.2. Wie greife ich auf ein mitgeliefertes Package?
▮ Comment saisir un Package livré

Die mitgelieferten Packages sieht man unter (Info, Open Function Browser, Packages) neu Help, Add-ons, Standard Packages, ... Nach dem Anklicken (Evaluate Template und dann Loaded Packages, Update Packages kann man im Browser das geladene Package anklicken und sieht die darin enthaltenen Funktionen) neu COPY-PASTE.

▮ Les Packages livrés se trouvent sous (Info, Open Function Browser, Packages) nouveau Help, Add-ons, Standard Packages, ... Après avoir marqué à la souris (Evaluate Template et ensuite Loaded Packages, Update Packages on peut marquer à la souris le Package chargé dans le Browser, on y voit les fonctions qui y sont contenues.) nouveau COPY-PASTE.

Lade das nachfolgende Package und schaue im Browser nach, welche Funktionen darin enthalten sind (Browser: Loaded Packages, Update Packages, anklicken etc.) neu Help, Add-ons, Standard Packages, ...

▮ Charge le Package suivant et regarde dans le Browser quelles fonctions y sont contenues. (Browser: Loaded Packages, Update Packages, marquer à la souris etc.) nouveau Help, Add-ons, Standard Packages, ...

```
In[1]:= Needs["Statistics`DescriptiveStatistics`"]

In[2]:= ?Mean

        Mean[list] gives the statistical mean of the elements in list. Mehr...

In[3]:= <<Algebra`FiniteFields`

In[4]:= ?GF

        GF[p, ilist][elist] represents an element of a finite field (Galois field) with prime
           characteristic p and where the coefficients of the irreducible polynomial are given by
           ilist (constant term listed first). The particular element is specified by elist, the
           list of coefficients of the polynomial representation of the element. GF[p, d], p prime
           and d a positive integer, is a special form giving a Galois field that is a degree d
           extension of the prime field of p elements. The resulting field has p^d elements. Mehr...
```

15.2. Contexts
▮ Contextes

15.2.1. Das Wesen des Contexts
▮ L'essence du contexte

Ein Context ist vergleichbar mit einem Directory (Verzeichnis) oder einem Subdirectory (Folder). An die Namen der Variablen oder Symbolen wird dort der Contextname angehängt, so dass es sich dann um lokale Objekte handelt. Das verhindert Namenkollision. Contexte können geschachtelt sein. Bisher haben wir immer im globalen Context gearbeitet. Probiere aus:

▮ Un contexte est comparable à un directoire (index) ou à un sous-directoire (Subdirectory, Folder). On y ajoute le nom du contexte aux noms des variables ou des symboles, de façon qu'il s'agisse ensuite d'objets locaux. Cela évite des collisions de noms. Les contextes peuvent être emboîtés. Jusqu'à présent nous avons toujours travaillé dans le contexte global. Essaie:

```
In[5]:= ?$Path
```

```
$Path gives the default list of directories to search in attempting to find an external file.
```
Mehr…

```
In[6]:= $Path
```

```
Out[6]= {C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Applications,
 ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

15.2.2. Der Context-Pfad
Le chemin ("path") du contexte

Studiere die Befehle:

Etudie les ordres:

```
In[7]:= ?Context
```

```
Context[ ] gives the current context.
   Context[symbol] gives the context in which a symbol appears.
```
Mehr…

```
In[8]:= ?$Context
```

```
$Context is a global variable that gives the current context.
```
Mehr…

```
In[9]:= ?$ContextPath
```

```
$ContextPath is a global variable that gives a list of contexts, after
   $Context, to search in trying to find a symbol that has been entered.
```
Mehr…

```
In[10]:= ?*Context*
```

System`

Context	ContextToFileName	$Context
Contexts	GetContext	$ContextPath
ContextToFilename	SetFileLoadingContext	

In[11]:= **?Dir***

System`

```
DiracDelta       Direction DirectoryName
DirectedInfinity Directory DirectoryStack
```

In[12]:= **?Directory**

`Directory[ ]` gives the current working directory. Mehr...

Probiere aus: ■ Essaie:

In[13]:= **$ContextPath**

Out[13]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`, NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`, Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}

In[14]:= **Context[ParametricPlot]**

Out[14]= System`

In[15]:= **Context[Integrate]**

Out[15]= System`

15.2.2. Context und Symbole
■ Contexte et symboles

Beispiele
■ Essaie

Studiere: ■ Etudie:

In[16]:=
```
x = 2;
Remove[a];
{Context[x], Context[a], Context[Tuten]}
```

Out[18]= {Global`, Global`, Global`}

In welchem Context sind die geschaffenen Symbole?
■ Dans quel contexte se trouvent les symboles crées?

In[19]:=
```
Benzin = 50 Liter;
Auto`Benzin =
"Interessiert den Umweltschutz. --
Interessant pour la protection naturelle.";
```

In[21]:= **?Benzin**

Global`Benzin

Benzin = 50 Liter

```
In[22]:= ?*`Benzin
```

Global`
Benzin
Auto`
Benzin

```
In[23]:= ?Auto`Benzin

         Auto`Benzin

         Auto`Benzin = Interessiert den Umweltschutz. -- Interessant pour la protection naturelle.
```

Wird der Context "Global" angegeben bei einer Abfrage?
Teste das aus:
■ Est-ce que le contexte "Global" est iniqué lors d'une interrogation? Fais un test:

```
In[24]:= `StudentError`Hoppla =
         "Das war wohl nicht so gemeint!
         Ce n'est certainement pas l'intention!";

In[25]:= `StudentError`Hoppla

Out[25]= Das war wohl nicht so gemeint! Ce n'est certainement pas l'intention!

In[26]:= Global`StudentError`Hoppla

Out[26]= Das war wohl nicht so gemeint! Ce n'est certainement pas l'intention!
```

15.3. Context-Wechsel
■ Changement de contexte

15.3.1. Allgemeines
■ Généralités

Dies ist möglich mit der Variable $Context. Studiere:
■ Cela est possible avec la variable $Context. Etudie:

```
In[27]:= Benzin

Out[27]= 50 Liter

In[28]:= $Context = "Auto`"

Out[28]= Auto`

In[29]:= Benzin

Out[29]= Interessiert den Umweltschutz. -- Interessant pour la protection naturelle.
```

```
In[30]:= $Context

Out[30]= Auto`

In[31]:= $Context = "Global`"

Out[31]= Global`

In[32]:= Benzin

Out[32]= 50 Liter
```

15.3.2. Der Befehl "Begin"
▮ L'ordre "Begin"

Mit den Befehlen Begin, BeginPackage, End, EndPackage wird der Context gewechselt. Studiere:
▮ Par les ordres Begin, BeginPackage, End, EndPackage on change de contexte. Etudie:

```
In[33]:= ?Begin

        Begin["context`"] resets the current context. Mehr...

In[34]:= ?BeginPackage

        BeginPackage["context`"] makes context` and System` the only active contexts.
           BeginPackage["context`", {"need1`", "need2`", ... }] calls Needs on the needi. Mehr...

In[35]:= ?End

        End[ ] returns the present context, and reverts to the previous one. Mehr...

In[36]:= ?EndPackage

        EndPackage[ ] restores $Context and $ContextPath to their values before the preceding
           BeginPackage, and prepends the current context to the list $ContextPath. Mehr...
```

Probiere aus: ▮ Essaie:

```
In[37]:= $Context

Out[37]= Global`

In[38]:= $ContextPath

Out[38]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}

In[39]:= Begin["Versuch`"]

Out[39]= Versuch`

In[40]:= $Context

Out[40]= Versuch`
```

```
In[41]:= $ContextPath

Out[41]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

Nun führen wir eine Variable ein:

■ Nous introduisons une variable:

```
In[42]:= dumm = "frech und beschränkt --
                 mal poli et borné"

Out[42]= frech und beschränkt --   mal poli et borné

In[43]:= Context[dumm]

Out[43]= Versuch`

In[44]:= End[ ]

Out[44]= Versuch`

In[45]:= $Context

Out[45]= Global`

In[46]:= $ContextPath

Out[46]= {Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
          NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
          Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}

In[47]:= ?dumm

         Information::notfound : Symbol dumm not found. Mehr…

In[48]:= ?Versuch`dumm

        Versuch`dumm

         Versuch`dumm = frech und beschränkt --   mal poli et borné
```

In welchem Context sind wir jetzt und wie ist der Pfad gesetzt?

■ Dans quel contexte somme-nous maintenant et comment le chemin est-il posé?

15.3.3. Der Befehl "Begin"
■ L'ordre "Begin"

Probiere aus: ■ Essaie:

```
In[49]:= BeginPackage["HalloWach`"]

Out[49]= HalloWach`

In[50]:= $Context

Out[50]= HalloWach`
```

In[51]:= **$ContextPath**

Out[51]= {HalloWach`, System`}

Was ist der Unterschied zwischen Begin und BeginPackage?
Probiere jetzt früher definierte Variablen aus:

- Quelle est la différence entre Begin et BeginPackage?

Essaie maintenant des variables définies il y a un certain temps:

In[52]:= **?Benzin**

```
Information::notfound : Symbol Benzin not found. Mehr…
```

In[53]:= **?Global`Benzin**

```
Global`Benzin

Global`Benzin = 50 Global`Liter
```

In[54]:= **dumm**

Out[54]= dumm

In[55]:=
```
dumm =
"blöd und lieb zur Hälfte und beides
zusammen -- idiot et aimable à moitié
ou toutes les deux ensembles"
```

Out[55]=
```
blöd und lieb zur Hälfte und beides  zusammen --
  idiot et aimable à moitié ou toutes les deux ensembles
```

In[56]:= **EndPackage[]**

In[57]:= **$Context**

Out[57]= Global`

In[58]:= **$ContextPath**

Out[58]=
```
{HalloWach`, Algebra`FiniteFields`, Algebra`PolynomialPowerMod`,
 NumberTheory`NumberTheoryFunctions`, NumberTheory`FactorIntegerECM`,
 Statistics`DescriptiveStatistics`, Statistics`DataManipulation`, Global`, System`}
```

In[59]:= **dumm**

Out[59]=
```
blöd und lieb zur Hälfte und beides  zusammen --
  idiot et aimable à moitié ou toutes les deux ensembles
```

In[60]:= **Context[dumm]**

Out[60]= HalloWach`

In[61]:= **{Versuch`dumm, dumm}**

Out[61]=
```
{frech und beschränkt --   mal poli et borné,
 blöd und lieb zur Hälfte und beides  zusammen --
   idiot et aimable à moitié ou toutes les deux ensembles}
```

15.4. Zum Laden von Packages
▮ Pour charger des Packages

15.4.1. Befehle
▮ Ordres

Zur Verfügung stehen die Befehle <<, Get und Needs.
Studiere die Befehle:

▮ Les ordres suivants sont à disposition: <<, Get et Needs.
Etudie les ordres:

```
In[62]:= ?<<

        <<name reads in a file, evaluating each expression in it, and returning the last one. Mehr…

In[63]:= ?Get

        <<name reads in a file, evaluating each expression in it, and returning the last one. Mehr…

In[64]:= ?Needs

        Needs["context`"] loads an appropriate file if the
           specified context is not already in $Packages. Needs["context`", "file"]
           loads file if the specified context is not already in $Packages. Mehr…

In[65]:= ?$Path

        $Path gives the default list of directories to search in attempting to find an external file.
         Mehr…

In[66]:= $Path

Out[66]= {C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
          C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
          C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
          C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
          C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
          C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
          C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
            Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
          C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
          C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

Hinweis: Die Variable $Path wird im File init.m gesetzt! (C:\Programme\Wolfram Research\Mathematica\3.0\-Configuration\Kernel.) Dieses File kann von einem selbst modifiziert werden. (NeXT: Es soll sich bei privaten Installationen im eigenen Root befinden.) Beim Start von *Mathematica* wird es gelesen.

▮ Retenir: La variable $Path est mise dans le fichier init.m! (C:\Programme\Wolfram Research\Mathematica\3.0\-

Configuration\Kernel.) On peut modifier soi-même ce fichier. (NeXT: Il doit se trouver dans le propre Root quant il s'agit d'installations privées.) Il est lu quand on démarre *Mathematica*.

15.4.2. Laden mit Get resp. <<
▮ Charger avec Get resp. <<

Bemerkung: "Packages" befindet sich in $Path! Führe die folgenden Befehle nacheinander aus:
▮ Remarque: "Packages" se trouve dans $Path! Exécute les ordres suivants l'un après l'autre:

```
In[67]:= $Path

Out[67]= {C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
          C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
          C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
          C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
          C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
          C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
          C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\
            Mathematica\Applications, ., C:\Dokumente und Einstellungen\Rolf,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
          C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
          C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
          C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}

In[68]:= ?Helix

        Information::notfound : Symbol Helix not found. Mehr…

In[69]:= Remove[Helix]

In[70]:= << Graphics/Shapes.m;
         ?Helix

        Helix[(r:1, h:0.5, (m:2, n:20r))] is a list of n*m
           polygons approximating a helix with half height h and m turns. Mehr...

In[72]:= $ContextPath

Out[72]= {Graphics`Shapes`, Geometry`Rotations`, HalloWach`, Algebra`FiniteFields`,
          Algebra`PolynomialPowerMod`, NumberTheory`NumberTheoryFunctions`,
          NumberTheory`FactorIntegerECM`, Statistics`DescriptiveStatistics`,
          Statistics`DataManipulation`, Global`, System`}

In[73]:= << Graphics`Shapes`;
         ?Helix

        Helix[(r:1, h:0.5, (m:2, n:20r))] is a list of n*m
           polygons approximating a helix with half height h and m turns. Mehr...

In[75]:= << Graphics/Shapes.m;
         ?Helix

        Helix[(r:1, h:0.5, (m:2, n:20r))] is a list of n*m
           polygons approximating a helix with half height h and m turns. Mehr...
```

```
In[77]:= Needs["Graphics`Shapes`"]

In[78]:= $ContextPath

Out[78]= {Graphics`Shapes`, Geometry`Rotations`, HalloWach`, Algebra`FiniteFields`,
          Algebra`PolynomialPowerMod`, NumberTheory`NumberTheoryFunctions`,
          NumberTheory`FactorIntegerECM`, Statistics`DescriptiveStatistics`,
          Statistics`DataManipulation`, Global`, System`}
```

15.4.3. Laden mit Needs
▮ Charger avec Needs

Was ist der Unterschied gewesen zwischen Get und Needs?
▮ Quelle était la différence entre Get et Needs?

```
In[79]:= Get["Graphics`Shapes`"]

In[80]:= Needs["Graphics`Shapes`"]
```

Versuche: ▮ Essaie:

```
In[81]:= BeginPackage["Graphics`Shapes`",
          "Geometry`Rotations`"]

Out[81]= Graphics`Shapes`

In[82]:= $ContextPath

Out[82]= {Graphics`Shapes`, Geometry`Rotations`, System`}
```

15.4.3. Wie sieht man den Inhalt eines Packages (Funktionen)?
▮ Comment voit-on le contenu d'un Package (fonctions?)

Versuche: ▮ Essaie:

```
In[83]:= ?Rotate2D

         Rotate2D[vec, theta, (pt:{0,0})] rotates the vector vec by angle theta about point pt. Mehr...

In[84]:= ?Geometry`Rotations`Rotate2D

         Rotate2D[vec, theta, (pt:{0,0})] rotates the vector vec by angle theta about point pt. Mehr...

In[85]:= ?Graphics`Shapes`Helix (*Old Vers.*)

         Helix[(r:1, h:0.5, (m:2, n:20r))] is a list of n*m
            polygons approximating a helix with half height h and m turns. Mehr...

In[86]:= ?Geometry`Rotations`*
```

Geometry`Rotations`

Rotate2D Rotate3D RotationMatrix2D RotationMatrix3D

In[87]:= **?Graphics`Shapes`* (*Old Vers.*)**

Graphics`Shapes`

```
AffineShape Helix             RotateShape    TranslateShape
Cone        MoebiusStrip      ShrinkPolygons WireFrame
Cylinder    OutlinePolygons   Sphere
DoubleHelix PerforatePolygons Torus
```

In[88]:= **?Geometry`*`***

Geometry`Rotations`

```
Rotate2D Rotate3D RotationMatrix2D RotationMatrix3D
```

Geometry`Rotations`Private`

```
phi psi pt theta vec
```

Achtung: private bedeutet "zugriffgeschützt"! Es wird keine spezifische Information abgegeben. Beispiel:

■ Attention: Privé signifie "ne peut pas être saisi"! On ne donne pas d'information spécifique. Exemple:

In[89]:= **?Geometry`Rotations`Private`phi**

```
Geometry`Rotations`Private`phi
```

In[90]:= **?Geometry`Rotations`Private`***

Geometry`Rotations`Private`

```
phi psi pt theta vec
```

In[91]:= **?Geometry`Rotations`Private`vec**

```
Geometry`Rotations`Private`vec
```

Welche Packages sind da? Wir suchen z.B. ein Objekt, in dessen Namen hinter dem "`" ein "a" vorkommt:

■ Quels Packages y a-t-il? Nous cherchons p.ex. un objet, dans le nom duquel il y a un "`" suivi d'un "a" :

In[92]:= **?*`a***

Global`

```
a
```

Algebra`FiniteFields`Private`

```
a a$
```

BoxForm`

```
a    apat$ args  arguments  arg$
apat arg   args$ arguments$ a$
```

FE`

```
a allrules alphabetrules any
```

Graphics`Shapes`Private`

```
a averagepoints
```

Integrate`

```
aaaaa
```

NumberTheory`FactorIntegerECM`Private`

```
a
```

NumberTheory`NumberTheoryFunctions`Private`

```
a ac alen alen$ any args as as$ a$
```

Solve`

```
a
```

Statistics`DescriptiveStatistics`Private`

```
a
```

System`Convert`MathMLDump`

```
annotations
```

System`Dump`

```
a                alltags$         arg2    asp$
addmessagebutton any              arg2$   axes
addmessagebuttons applyUsageStyles argname axes$
allowprotect     arg              args    a$
allowprotect$    arg1             args$
alltags          arg1$            asp
```

System`FEDump`

```
allNotebooks allNotebooks$ arg args
```

System`HypergeometricDump`

```
a
```

System`LanguageEnhancements`

```
a
```

System`Private`

```
auto
```

15.5. Der Package-Stil
▮ Le style Package

Studiere die folgenden Beispiele konsequent durch!

▮ Fais une étudie approfondie des exemples suivants!

Hier zwei Beispiele aus der *Mathematica*-Bibliothek
■ Voici deux exemples de la bibliothèque de *Mathematica*:

Beispiel 1 (Kopie aus der old Library, etwas kommentiert)
■ Exemple 1 (Copie de la old Librery, un peu commentée)

```
(*************************************************************************

    Adapted from
    Roman E. Maeder: Programming in Mathematica,
    Second Edition, Addison-Wesley, 1991.
Wichtig: Gute Dokumentation! ■  Important: Bonne documentation!

 *************************************************************************)


BeginPackage["Graphics`ArgColors`"]
(* Wichtig: $Context und $ContextPath wechseln!                    *)
(*■ Important: Changer $Context et $ContextPath !                  *)
(* Wichtig: Die hier definierten Funktionen, z.B. ArgColor, sind   *)
(*        nach aussen sichtbar. Die Symbole dafür werden in diesem *)
(*        Context kreiert.                                         *)
(* ■ Important: Les fonctions définies ici, p.ex. ArgColor, sont   *)
(*        visibles à l'extérieur. Les symboles en sont crées dans ce *)
(*        contexte.                                                *)

ArgColor::usage = "ArgColor[z] gives a color value whose hue is proportional
        to the argument of the complex number z."
ArgShade::usage = "ArgShade[z] gives a gray level proportional
        to the argument of the complex number z."

ColorCircle::usage = "ColorCircle[r, (light:1)] gives a color value whose hue
        is proportional to r (mod 2Pi) with lightness light."
(* Wichtig: usage-Statement für alle aussen sichtbaren Funktionen *)
(* ■ Important: usage-Statement visibles pour toutes les fonctions *)
(*    visibles à l'extérieur. *)

Begin["`Private`"]
(* Wichtig: $Context wechselt auf `Private`,                       *)
(* ■ Important: $Context devient `Private`,                        *)
(*        $ContextPath bleibt ■ reste.  *)
(* Wichtig: Hier neu definierte Funktionen sind aussen unsichtbar! *)
(* ■ Important: Les fonctions définies ici nouvellement ne sont pas*)
(*    visibles à l'extérieur! *)

HelperFunction[z_] := Print["Mich siehst Du von aussen nicht", z]
```

```
ArgColor[z_] := RGBColor[1, 1, 1] /; z == 0.0

ArgColor[z_] := ColorCircle[ Arg[z] ] /; NumberQ[N[arg]]

ArgShade[z_] := GrayLevel[1] /; z == 0.0

ArgShade[z_] := GrayLevel[N[(Pi + Arg[z])/(2Pi)] ] /; NumberQ[N[Arg[z]]]

ColorCircle[arg_, light_:1.0] := Hue[ N[arg/2/Pi], 1, light] /; NumberQ[N[arg]]

End[ ]
(* Wichtig: $Context wechselt zurück                    *)
(*          auf den Wert vor dem Aufruf von `Private`    *)
(* ■  Important: $Context reprend la forme d'origine    *)
(*          sur la valeur d'avant l'apppel de  `Private`  *)

Protect[ArgColor, ArgShade, ColorCircle]
(* Wichtig: Die aussen sichtbaren Funktionen werden geschützt.    *)
(* ■ Important: Les fonctions visibles à l'extérieur sont protégées.    *)

EndPackage[ ]
(* Wichtig: $Context wird zurückgewechselt auf den alten Wert.    *)
(* Wichtig: Zu $ContextPath wird der Name                  *)
(*          "Graphics`ArgColors`" hinzugefügt.             *)
(* ■ Important: $Context reprend la valeur d'origine.      *)
(* ■ Important: On ajoute à $ContextPath le nom            *)
(*          "Graphics`ArgColors`" .                        *)
```

In[93]:=
```
(*********************************************************************

 Adapted from
 Roman E. Maeder: Programming in Mathematica,
 Second Edition, Addison-Wesley, 1991.
 (Obige Kopie aus der Library etwas abgeändert,
  so dass sie auf NeXT... lauffähig ist.
  ■ La copie de Libary ci cessus un peu changée,
   de façon qu'elle puisse fonctionner sur NeXT
   .....)

  *********************************************************************)

 BeginPackage["Test`"];

 argColor1::usage =
 "ArgColor[z] gives a color value whose hue is
     proportional;
     to the argument of the complex number z."
 argShade1::usage =
 "ArgShade[z] gives a gray level proportional
     to the argument of the complex number z.";

 colorCircle1::usage =
 "ColorCircle[r, (light:1)] gives a color value
      whose hue is proportional to r (mod 2Pi)
      with lightness light.";

 Begin["`Private`"];

 argColor1[z_] := RGBColor[1, 1, 1] /;
                  z == 0.0  ;

 argColor1[z_] := ColorCircle[ Arg[z] ] /;
                  NumberQ[N[Arg]]  ;

 argShade1[z_] := GrayLevel[1] /; z == 0.0
 argShade1[z_] := GrayLevel[N[(Pi +
         Arg[z])/(2Pi)] ] /;
         NumberQ[N[Arg[z]]]   ;

 colorCircle1[arg_, light_:1.0] :=
         Hue[ N[arg/2/Pi], 1, light] /;
         NumberQ[N[Arg]]   ;

 End[];

 Protect[argColor1, argShade1, colorCircle1]

 EndPackage[];
```

Out[95]=
```
ArgColor[z] gives a color value whose hue
  is      proportional; to the argument of the complex number z.
```

Out[105]=
```
{argColor1, argShade1, colorCircle1}
```

In[107]:=
```
(*********************************************************************

 Adapted from
```

```
Roman E. Maeder: Programming in Mathematica,
Second Edition, Addison-Wesley, 1991.
(Obige Kopie aus der Library etwas abgeändert,
 so dass sie auf NeXT.. lauffähig ist, jetzt mit
 Abfrage der Context-Variablen und des Context-
 Pfades.
 ■ La copie de Libary ci cessus un peu changée,
  de façon qu'elle puisse fonctionner sur NeXT
  .., maintenant avec interrogation (appel) des
  variables du contexte et du chemin du
  contexte.)

 ************************************************************************)

Print["1. Abfrage: $ContextPath = ",
    $ContextPath, " $Context = ", $Context];

BeginPackage["Test`"];
Print["2. Abfrage: $ContextPath = ",
    $ContextPath, " $Context = ", $Context];

argColor1::usage =
"ArgColor[z] gives a color value whose hue is
    proportional;
    to the argument of the complex number z."
argShade1::usage =
"ArgShade[z] gives a gray level proportional
    to the argument of the complex number z.";

colorCircle1::usage = "ColorCircle[r,
    (light:1)] gives a color value
     whose hue is proportional to r (mod 2Pi)
     with lightness light.";

Begin["`Private`"];
Print["3. Abfrage: $ContextPath = ",
    $ContextPath, " $Context = ", $Context];

argColor1[z_] := RGBColor[1, 1, 1] /;
        z == 0.0  ;

argColor1[z_] := ColorCircle[ Arg[z] ] /;
        NumberQ[N[Arg]]  ;

argShade1[z_] := GrayLevel[1] /;
        z == 0.0
argShade1[z_] := GrayLevel[N[(Pi +
        Arg[z])/(2Pi)] ] /;
        NumberQ[N[Arg[z]]]   ;

colorCircle1[arg_, light_:1.0] :=
        Hue[ N[arg/2/Pi], 1, light] /;
        NumberQ[N[Arg]]   ;

End[];
Print["4. Abfrage: $ContextPath = ",
    $ContextPath, " $Context = ", $Context];

Protect[argColor1, argShade1, colorCircle1]

EndPackage[];
Print["5. Abfrage: $ContextPath = ",
```

```
    $ContextPath, " $Context = ", $Context];
```

1. Abfrage: $ContextPath = {Test`, Graphics`Shapes`, Geometry`Rotations`, System`} $Context = Graphics`Shapes`

2. Abfrage: $ContextPath = {Test`, System`} $Context = Test`

Out[111]=

ArgColor[z] gives a color value whose hue
  is proportional; to the argument of the complex number z.

3. Abfrage: $ContextPath = {Test`, System`} $Context = Test`Private`

SetDelayed::write : Tag argColor1 in argColor1[z_] is Protected. Mehr…

SetDelayed::write : Tag argColor1 in argColor1[z_] is Protected. Mehr…

SetDelayed::write : Tag argShade1 in argShade1[z_] is Protected. Mehr…

Out[118]=

$Failed

SetDelayed::write : Tag argShade1 in argShade1[z_] is Protected. Mehr…

SetDelayed::write : Tag colorCircle1 in colorCircle1[arg_, light_ : 1.] is Protected. Mehr…

4. Abfrage: $ContextPath = {Test`, System`} $Context = Test`

Out[123]=

{}

5. Abfrage: $ContextPath = {Test`, Graphics`Shapes`, Geometry`Rotations`, System`} $Context = Graphics`Shapes`

Beispiel 2 (Kopie aus der Library, ohne Kommentar)
■ Exemple 2 (Copie de Library, sans commentaire)

```
(*************************************************************************

        Copyright 1991 by Roman E. Maeder

        Adapted from
        Roman E. Maeder: Programming in Mathematica,
        Second Edition, Addison-Wesley, 1991.

        Permission is hereby granted to make copies of this file for
        any purpose other than direct profit, or as part of a
        commercial product, provided this copyright notice is left
        intact.  Sale, other than for the cost of media, is prohibited.

        Permission is hereby granted to reproduce part or all of
        this file, provided that the source is acknowledged.

 *************************************************************************)

        (* Enclosed with Mathematica by permission *)


BeginPackage["Examples`Collatz`"]

Collatz::usage =
        "Collatz[n] gives a list of the iterates in the 3n+1 problem,
        starting from n. The conjecture is that this sequence always
        terminates."

StoppingTime::usage = "StoppingTime[n] finds the total stopping time
        of the integer n. This is the length of the Collatz sequence
        before hitting 1."

FindMaxima::usage = "FindMaxima[from] reports successive maxima of the
        total stopping time starting the search with the integer from."

Begin["`Private`"]

Collatz[n_Integer] := AppendCollatz[{}, n]

AppendCollatz[sofar_, 1] := Flatten[{sofar, 1}]

AppendCollatz[sofar_, n_Integer] :=
        AppendCollatz[{sofar, n}, 3 n + 1]    /; OddQ[n]

AppendCollatz[sofar_, n_Integer] :=
```

```
        AppendCollatz[{sofar, n}, n / 2]        /; EvenQ[n] && n != 0

StoppingTime[n_Integer?Positive] :=
        Module[{i=1, m=n}, While[m != 1, m = If[OddQ[m], 3m+1, m/2]; i++];i ]

FindMaxima[low_] :=
        Module[{m=0, n=0, i=low, j},
                While[ True,
                        j = StoppingTime[i];
                        If[j > m, m = j; n = i;
                                Print["StoppingTime[", n, "] = ", m] ];
                        i++;
                        If[ Mod[i, 100] == 0, Print["i = ", i] ]
                ]
        ]

End[]

Protect[ Collatz, StoppingTime, FindMaxima ]

EndPackage[ ]
```

15.6. Namenskonflikte
▮ Conflits de noms

15.6.1. Beispiel eines Konflikts
▮ Exemple d'un conflit

Mehrfach verwendete Namen in verschiedenen Umgebungen
▮ Noms utilisés de plusieurs façons dans des environs différents.

Studiere die folgende Abfolge:
▮ Etudier la suite suivante:

In[126]:=
```
LaplaceTransform[(1+4t+5t^2) Exp[-3t], t, s]
```

Out[126]=

$$\frac{10}{(3+s)^3}+\frac{4}{(3+s)^2}+\frac{1}{3+s}$$

Die gerufene Funktion war noch unbekannt. Das Symbol LaplaceTransform ist aber jetzt kreiert worden, allerdings ohne konkreten Inhalt.
▮ La fonction appelée était encore inconnue. Le symbole LaplaceTransform a été crée maintenant, cependant sans contenu concret.

In[127]:=
```
?LaplaceTransform
```

```
LaplaceTransform[expr, t, s] gives the Laplace
   transform of expr. LaplaceTransform[expr, {t1, t2, ... }, {s1,
   s2, ... }] gives the multidimensional Laplace transform of expr. Mehr…
```

Laden des Packages:

■ Charger le Package:

In[128]:=
```
Needs["Calculus`LaplaceTransform`"]
```

```
Get::noopen : Cannot open Calculus`LaplaceTransform`. Mehr…

Needs::nocont : Context Calculus`LaplaceTransform` was not created when Needs was evaluated. Mehr…
```

Out[128]=
```
$Failed
```

In[129]:=
```
?LaplaceTransform
```

```
LaplaceTransform[expr, t, s] gives the Laplace
   transform of expr. LaplaceTransform[expr, {t1, t2, ... }, {s1,
   s2, ... }] gives the multidimensional Laplace transform of expr. Mehr…
```

In[130]:=
```
<<Calculus`FourierTransform`
```

In[131]:=
```
NFourierTransform[Exp[-t^2]/(1 + Abs[t]), t, 1]
```

Out[131]=
```
0.404024 + 0. i
```

Die in Global` vorher kreierte Funktion überschattet immer noch die in einem "Unter-Context" zugeladene Funktion mit demselben Namen.

■ La fonction créée auparavant dans Global` recouvre encore la fonction du même nom qui se trouve dans un sous-contexte.

In[132]:=
```
Remove[FourierTransform]
```

```
Remove::rmptc : Symbol FourierTransform is Protected and cannot be removed. Mehr…
```

Die in Global` existierende Funktion wird eliminiert. Die darunter liegende zeigt sich jetzt.

■ La fonction existant dans un Global` est élininée. Celle qui se trouve dessous devient visible.

In[133]:=
```
?FourierTransform
```

```
FourierTransform[expr, t, ω] gives the symbolic
   Fourier transform of expr. FourierTransform[expr, {t1, t2, ... }, {ω1,
   ω2, ... }] gives the multidimensional Fourier transform of expr. Mehr…
```

Nun kann man sie auch verwenden:

■ Maintenant on peut aussi l'utiliser:

In[134]:=
```
FourierTransform[(1+4t+5t^2), t, ω]
```

Out[134]=
$$\sqrt{2\pi}\ \text{DiracDelta}[\omega] - 4\,i\,\sqrt{2\pi}\ \text{DiracDelta}'[\omega] - 5\sqrt{2\pi}\ \text{DiracDelta}''[\omega]$$

Was gibt es sonst noch im geladenen Package?

■ Qu'y a-t-il encore dans le Package chargé?

In[135]:=
```
?Calculus`FourierTransform`*
```

Calculus`FourierTransform`

```
DTFourierTransform           NFourierCosSeriesCoefficient
FourierCoefficient           NFourierCosTransform
FourierCosCoefficient        NFourierExpSeries
FourierCosSeriesCoefficient  NFourierExpSeriesCoefficient
FourierExpSeries             NFourierSeries
FourierExpSeriesCoefficient  NFourierSinCoefficient
FourierFrequencyConstant     NFourierSinSeriesCoefficient
FourierOverallConstant       NFourierSinTransform
FourierSample                NFourierTransform
FourierSeries                NFourierTrigSeries
FourierSinCoefficient        NInverseDTFourierTransform
FourierSinSeriesCoefficient  NInverseFourierCoefficient
FourierTrigSeries            NInverseFourierCosTransform
InverseDTFourierTransform    NInverseFourierSinTransform
InverseFourierCoefficient    NInverseFourierTransform
NDTFourierTransform          $FourierFrequencyConstant
NFourierCoefficient          $FourierOverallConstant
NFourierCosCoefficient
```

15.6.2. Der empfohlene Stil
■ Le style conseillé

Im Folgenden wird ein Prototyp eines Packages demonstriert, das ursprünglich von Henry Cejtin und Theo Gray stammen soll:

■ Dans ce qui suit on démontre un prototype d'un Package qui a été créé, soit-disant, par Henry Cejtin et Theo Gray.

In[136]:=
```
 (* OpenWrite["C:\\work\\MathematicaData\\AAKurs15.m"];
WriteString["C:\\work\\MathematicaData\\AAKurs15.m", "\n",
"Du sollst das Package von Hand in das File",
"\n","AAKurs15.m kopieren."
 ];   (* End WriteString  etc. – continued .... *)  *)

OpenWrite["AAKurs15.m"];
WriteString["AAKurs15.m", "\n",
"Du sollst das Package von Hand in das File",
"\n", "AAKurs15.m kopieren."
 ] ;  (* End WriteString  etc. – continued .... *)
```

In[138]:=

```
" " >> AAKurs15.m;
```

In[139]:=

```
OpenWrite[
"AAKurs15.m"];
WriteString[
"AAKurs15.m",
 "\n",
"!!!!!!
Du sollst das Package von Hand in das File",
"\n","AAKurs15.m kopieren."
"Il faut copier le Package à la main dans le
 File", "\n","AAKurs15.m kopieren." ];
(* End WriteString   *)

 Close["AAKurs15.m"];


(*
 * AAKurs15.m
 *
 * This package demonstrates the recommended
 * package style.
 *
 *)

(*
 * Prepend the context ProtoNotebook` to
 * $ContextPath.
 * Then Needs will be able to determine when
 * the package
 * has been loaded.
 *)
BeginPackage["AAKurs15`"];
EndPackage[];

(*
 * Now the usage statements.
 *)
ExamplePlot::usage = "ExamplePlot[n,m] draws \
    n starbursts and m spirals. \
    ExamplePlot[n] draws n starbursts and
    no spirals.";

SamplePlot::usage =
    "SamplePlot is Vaporware. It \
    does nothing.";

(*
 * Now the global objects.
 *)
ExamplePlot;
SamplePlot;

(*
 * Now the private context. The objects here
 * are not \
 * intended to be accessed by the user.
 *)
 Begin["AAKurs15`Private`"];
```

```
(*
 * Local objects.
 *)
`starBurst;
`spiral;

(*
 * Define starBurst and spiral.
 *)
 starBurst[center_, radius_, n_] :=
    Block[
        { `r  },
        Table[
            Line[{center,
                  center + radius {Cos[r],
                  Sin[r]}
                 }],
                 {r, 0, 2Pi - 2Pi/n, 2Pi/n}
             ]
          ];

spiral[center_, radius_, loops_] :=
    Block[
        { `t  },
        Line[
            Table[center +
                  radius t {Cos[t], Sin[t]}/(2Pi
                                   loops),
                  {t, 0, 2Pi loops, 2Pi/30}
                 ]
             ]
          ];

ExamplePlot[stars_] :=
    Show[
        Graphics[
            Table[
                starBurst[
                    4 {Random[], Random[]},
                    0.3 + Random[],
                    20
                         ],
                         {stars}
                  ]
                ],
                AspectRatio -> Automatic
          ];

ExamplePlot[stars_, spins_] :=
    Show[
        Graphics[{
            Table[
                starBurst[
                    4 {Random[], Random[]},
                    0.3 + Random[],
                    20
                         ],
                         {stars}
                  ],
             Table[
                spiral[
                    4 {Random[], Random[]},
```

```
                    0.3 + Random[],
                    Random[Integer, {2, 5}]
                         ],
                         {spins}
                    ]
               }],
               AspectRatio -> Automatic
          ];

     (*
      * Protect now the function and lock the private
      * functions.
      *)
      Protect[ExamplePlot, SamplePlot];
      Append[Attributes[ExamplePlot], Locked];
      Append[Attributes[SamplePlot], Locked];

      End[];
```

```
General::aofil : AAKurs15.m already open as AAKurs15.m. Mehr…
```

```
OpenWrite::noopen : Cannot open AAKurs15.m. Mehr…
```

```
Syntax::newl:
   The newline character after
    "               radius t {Cos[t], Sin[t]}/(2Pi"
     is understood as a multiplication operator. Mehr…
```

Schaue ins File ■ Regarde le fichier
C:\Programme\Wolfram Research\Mathematica\3.0\AAKurs15.nb !!!!!!!!!!!!

In[175]:=

```
Needs["AAKurs15`"]
```

In[176]:=

```
Needs["C:\\work\\MathematicaData\\AAKurs15.m"]
```

```
Needs::cxt : Invalid context specified at position 1 in Needs[C:\work\MathematicaData\AAKurs15.m].
   A context must consist of valid symbol names separated by and ending with `. Mehr…
```

Out[176]=

```
Needs[C:\work\MathematicaData\AAKurs15.m]
```

In[177]:=

```
<< AAKurs15.m
```

```
General::noopen : Cannot open Du sollst das Package von Hand in das File. Mehr…
```

```
Syntax::sntx:
   Syntax error in or before
    "AAKurs15.m kopieren. Il faut copier le Package a` la main".
        dans le   File
                                                    ^
     (line 6 of "AAKurs15.m") Mehr…
```

In[178]:=

```
? AA*
```

```
Graphics`Shapes`AAKurs15
```

In[179]:=
```
$Path
```

Out[179]=
```
{C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\JLink,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\NETLink,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\Rolf\Anwendungsdaten\Mathematica\Applications,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Kernel,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Autoload,
 C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Mathematica\Applications,
 ., C:\Dokumente und Einstellungen\Rolf,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\StandardPackages\StartUp,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Autoload,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\Applications,
 C:\Programme\Wolfram Research\Mathematica\5.2\AddOns\ExtraPackages,
 C:\Programme\Wolfram Research\Mathematica\5.2\SystemFiles\Graphics\Packages,
 C:\Programme\Wolfram Research\Mathematica\5.2\Configuration\Kernel}
```

In[180]:=
```
Needs["AAKurs15`"]
```

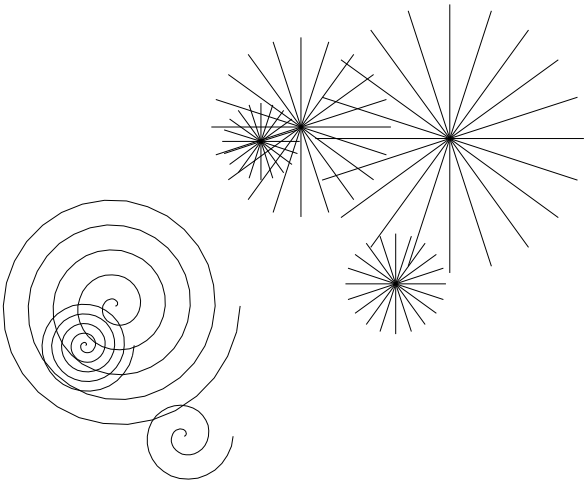
In[181]:=
```
?SamplePlot
```

```
SamplePlot is Vaporware. It     does nothing. Mehr...
```

In[182]:=
```
?ExamplePlot
```

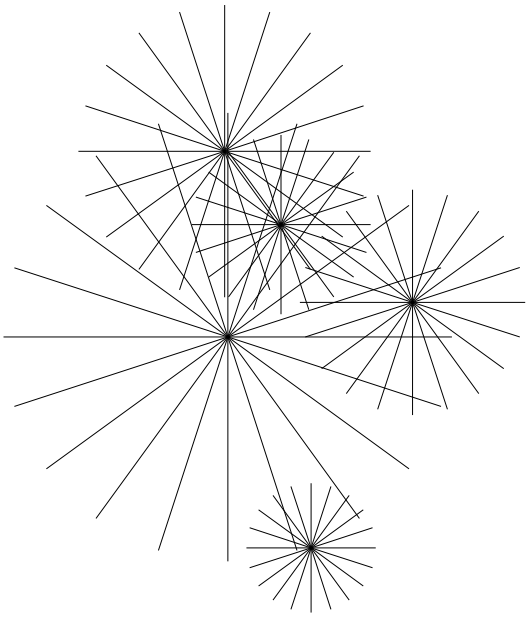
```
ExamplePlot[n,m] draws  n starbursts and m
   spirals.  ExamplePlot[n] draws n starbursts and  no spirals. Mehr...
```

In[183]:=
```
ExamplePlot[4, 3]
```



Out[183]=
```
- Graphics -
```

In[184]:=
```
ExamplePlot[5]
```



Out[184]=
```
- Graphics -
```

In[185]:=
```
SamplePlot[5]
```

Out[185]=
```
SamplePlot[5]
```

15.6.3. Einlesen eines externen Files in einen andern Context
▮ Lire un fichier extérieur et le mettre dans un autre contexte

Im folgenden Beispiel wird gezeigt, wie ProtoNotbook` in den Context PlottingExamples` eingelesen wird. Probiere die verschiedenen Situationen durch:

▮ Dans l'exemple suivant on montre comme ProtoNotbook` est mis dans le contexte PlottingExamples`. Essaie les différentes situations:

In[186]:=
```
?SymbolicSum
```

```
Information::notfound : Symbol SymbolicSum not found. Mehr…
```

In[187]:=
```
Remove["Global`*"]
```

In[188]:=
```
End[]
```

Out[188]=
```
Graphics`Shapes`
```

In[189]:=
```
?*SamplePlot
```
```
SamplePlot is Vaporware. It     does nothing. Mehr...
```

In[190]:=
```
$Context
```
Out[190]=
```
Global`
```

In[191]:=
```
Begin["PlottingExamples`"];
$Context
```
Out[192]=
```
PlottingExamples`
```

In[193]:=
```
$ContextPath
```
Out[193]=
```
{AAKurs15`, Calculus`FourierTransform`, Utilities`FilterOptions`,
 Test`, Graphics`Shapes`, Geometry`Rotations`, System`}
```

In[194]:=
```
?SymbolicSum
```
```
Information::notfound : Symbol SymbolicSum not found. Mehr…
```

In[195]:=
```
<<Calculus`VectorAnalysis`
```

In[196]:=
```
?JacobianMatrix
```
```
JacobianMatrix[pt] gives the Jacobian matrix of the transformation from the
   default coordinate system to the Cartesian coordinate system at the point pt.
   JacobianMatrix[pt, coordsys] gives the Jacobian matrix of the transformation
   from the coordinate system coordsys to the Cartesian coordinate system at the
   point pt. If pt is not given, the default coordinate variables are used. Mehr...
```

In[197]:=
```
?*SamplePlot
```
```
SamplePlot is Vaporware. It     does nothing. Mehr...
```

In[198]:=
```
Needs["Calculus`VectorAnalysis`"]
```

In[199]:=
```
?Calculus`VectorAnalysis`*
```

Calculus`VectorAnalysis`

ArcLengthFactor	Div	Pphi
Biharmonic	DotProduct	ProlateSpheroidal
Bipolar	Eeta	Rr
Bispherical	EllipticCylindrical	ScalarTripleProduct
Cartesian	Grad	ScaleFactors
ConfocalEllipsoidal	JacobianDeterminant	SetCoordinates
ConfocalParaboloidal	JacobianMatrix	Toroidal
Conical	Laplacian	Ttheta
CoordinateRanges	Llambda	Uu
Coordinates	Mmu	Vv
CoordinatesFromCartesian	Nnu	Xx
CoordinatesToCartesian	OblateSpheroidal	Xxi
CoordinateSystem	ParabolicCylindrical	Yy
CrossProduct	Paraboloidal	Zz
Curl	ParameterRanges	
Cylindrical	Parameters	

In[200]:=
```
$Context
```

Out[200]=
```
PlottingExamples`
```

In[201]:=
```
$ContextPath
```

Out[201]=
```
{Calculus`VectorAnalysis`, AAKurs15`, Calculus`FourierTransform`,
 Utilities`FilterOptions`, Test`, Graphics`Shapes`, Geometry`Rotations`, System`}
```

In[202]:=
```
a = 1
```

Out[202]=
```
1
```

In[203]:=
```
a
```

Out[203]=
```
1
```

In[204]:=
```
End[]
```

Out[204]=
```
PlottingExamples`
```

In[205]:=
```
a
```

Out[205]=
```
a
```

In[206]:=
```
$Context
```

Out[206]=
```
Global`
```

In[207]:=
```
$ContextPath
```

Out[207]=
```
{Calculus`VectorAnalysis`, AAKurs15`, Calculus`FourierTransform`,
 Utilities`FilterOptions`, Test`, Graphics`Shapes`, Geometry`Rotations`, System`}
```

In[208]:=
```
?Calculus`VectorAnalysis`*
```

Calculus`VectorAnalysis`

ArcLengthFactor	Div	Pphi
Biharmonic	DotProduct	ProlateSpheroidal
Bipolar	Eeta	Rr
Bispherical	EllipticCylindrical	ScalarTripleProduct
Cartesian	Grad	ScaleFactors
ConfocalEllipsoidal	JacobianDeterminant	SetCoordinates
ConfocalParaboloidal	JacobianMatrix	Toroidal
Conical	Laplacian	Ttheta
CoordinateRanges	Llambda	Uu
Coordinates	Mmu	Vv
CoordinatesFromCartesian	Nnu	Xx
CoordinatesToCartesian	OblateSpheroidal	Xxi
CoordinateSystem	ParabolicCylindrical	Yy
CrossProduct	Paraboloidal	Zz
Curl	ParameterRanges	
Cylindrical	Parameters	

15.6.4. Notebooks

▮ Notebooks

Dieses File hier ist ein Notebook. In solchen Notebooks oder allgemein bei der interaktiven Benützung von Mathematica mit einem Front-End, das mit Zellen arbeitet, können Probleme mit dem Context auftreten, wenn man nicht aufpasst. Studiere das folgende Beispiel:

▮ Le fichier est un Notebook. Dans des Notebooks de ce genre ou plus généralement quand on utilise interactivement *Mathematica* avec un Front-End qui travaille avec des cellules, il peut y avoir des problèmes avec le contexte quand on ne fait pas attention. Etudie l'exemple suivant:

In[209]:=
```
Remove["Global`@*"]
```

In[210]:=
```
esel = "Dummes menschliches Individuum,
Anwesende immer ausgeschlosen. --- Etre humain
bête, personnes présents exclus."
```

Out[210]=
```
Dummes menschliches Individuum,  Anwesende immer
  ausgeschlosen. --- Etre humain  bête, personnes présents exclus.
```

In[211]:=
```
Context[esel]
```

Out[211]=
```
Global`
```

In[212]:=
```
BeginPackage["Zoologie"];
esel = "Säugetier im Dienste des Menschen.
       --- Mammifèr au service des hommes";
```

```
BeginPackage::cxt : Invalid context specified at position 1 in BeginPackage[Zoologie].
   A context must consist of valid symbol names separated by and ending with `. Mehr…
```

In[214]:=
```
esel
```

Out[214]=
```
Säugetier im Dienste des Menschen.        --- Mammifèr au service des hommes
```

In[215]:=
```
Context[esel]
```

Out[215]=
```
Global`
```

In[216]:=
```
Global`esel
```

Out[216]=
```
Säugetier im Dienste des Menschen.        --- Mammifèr au service des hommes
```

In[217]:=
```
Zoologie`esel
```

Out[217]=
```
Zoologie`esel
```

In[218]:=
```
EndPackage[]
```

```
EndPackage::noctx : No previous context defined. Mehr…
```

Out[218]=
```
EndPackage[]
```

In[219]:=
```
esel
```

Out[219]=
```
Säugetier im Dienste des Menschen.        --- Mammifèr au service des hommes
```

Wo ist der eingangs eingegebene "Wert" der Variablen "esel" geblieben?

Où a disparu la "valeur" de la variable "esel" entrée au début?

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

In[220]:=

```
(* Old Form: Remove["Global`@*"] *)
```

In[221]:=

```
Remove["Global`*"]
```

Uebungen ▮ Exercices

15. Packages

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach.... Hinweis: Kapitel 15 lesen!
▮ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....
Indication: Lire le chapitre 15.

Run mit WIN+*Mathematica* Version 5.2
▮ Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

Aufgabe 1 ▮ Problème 1

Lade das Package "Colors.m" aus "Graphics". Liste alle Farben auf.
▮ Charger le Package "Colors.m" de "Graphics". Fais une liste des couleurs.

```
In[1]:= Needs["Graphics`Colors`"]
```

```
In[2]:= ?Graphics`Colors`*
```

Graphics`Colors`

```
AliceBlue          Goldenrod          PaleGreen
AlizarinCrimson    GoldenrodDark      PaleTurquoise
AllColors          GoldenrodLight     PaleVioletRed
Antique            GoldenrodPale      PapayaWhip
Apricot            GoldOchre          Peach
Aquamarine         GreenDark          PeachPuff
AquamarineMedium   GreenishUmber      Peacock
AureolineYellow    GreenPale          PermanentGreen
Azure              GreenYellow        PermanentRedViolet
Banana             HLSColor           Peru
Beige              Honeydew           PinkLight
Bisque             HotPink            Plum
BlanchedAlmond     HSBColor           PowderBlue
BlueLight          IndianRed          PrussianBlue
BlueMedium         Indigo             PurpleMedium
BlueViolet         Ivory              Raspberry
Brick              IvoryBlack         RawSienna
```

```
Brick              IvoryBlack          RawSienna
BrownMadder        Khaki               RawUmber
BrownOchre         KhakiDark           RoseMadder
Burlywood          LampBlack           RosyBrown
BurntSienna        Lavender            RoyalBlue
BurntUmber         LavenderBlush       SaddleBrown
Cadet              LawnGreen           Salmon
CadetBlue          LemonChiffon        SandyBrown
CadmiumLemon       LightBeige          SapGreen
CadmiumOrange      LightBlue           SeaGreen
CadmiumRedDeep     LightCadmiumRed     SeaGreenDark
CadmiumRedLight    LightCadmiumYellow  SeaGreenLight
CadmiumYellow      LightCoral          SeaGreenMedium
CadmiumYellowLight LightGoldenrod      Seashell
Carrot             LightGray           Sepia
Cerulean           LightPink           Sienna
Chartreuse         LightSalmon         SkyBlue
Chocolate          LightSeaGreen       SkyBlueDeep
ChromeOxideGreen   LightSkyBlue        SkyBlueLight
CinnabarGreen      LightSlateBlue      SlateBlue
CMYColor           LightSlateGray      SlateBlueDark
Cobalt             LightSteelBlue      SlateBlueLight
CobaltGreen        LightViridian       SlateBlueMedium
CobaltVioletDeep   LightYellow         SlateGray
ColdGray           LimeGreen           SlateGrayDark
Coral              Linen               SlateGrayLight
CoralLight         MadderLakeDeep      Smoke
CornflowerBlue     ManganeseBlue       Snow
Cornsilk           Maroon              SpringGreen
CyanWhite          MarsOrange          SpringGreenMedium
DarkGoldenrod      MarsYellow          SteelBlue
DarkGreen          MediumAquamarine    SteelBlueLight
DarkKhaki          MediumBlue          TerreVerte
DarkOliveGreen     MediumOrchid        Thistle
DarkOrange         MediumPurple        Titanium
DarkOrchid         MediumSeaGreen      Tomato
DarkSeaGreen       MediumSlateBlue     Turquoise
DarkSlateBlue      MediumSpringGreen   TurquoiseBlue
DarkSlateGray      MediumTurquoise     TurquoiseDark
DarkTurquoise      MediumVioletRed     TurquoiseMedium
DarkViolet         Melon               TurquoisePale
DeepCadmiumRed     MidnightBlue        Ultramarine
DeepCobaltViolet   Mint                UltramarineViolet
DeepMadderLake     MintCream           VanDykeBrown
DeepNaplesYellow   MistyRose           VenetianRed
DeepOchre          Moccasin            Violet
DeepPink           NaplesYellowDeep    VioletDark
DeepSkyBlue        Navajo              VioletRed
DimGray            Navy                VioletRedMedium
DodgerBlue         NavyBlue            VioletRedPale
Eggshell           Oak                 ViridianLight
EmeraldGreen       OldLace             WarmGray
EnglishRed         Olive               Wheat
Firebrick          OliveDrab           YellowBrown
Floral             OliveGreenDark      YellowGreen
ForestGreen        OrangeRed           YellowLight
Gainsboro          Orchid              YellowOchre
GeraniumLake       OrchidDark          YIQColor
```

```
Ghost              OrchidMedium       Zinc
Gold               PaleGoldenrod
```

Aufgabe 2 ▮ Problème 2

Schreibe ein *Mathematica*-Package, das den Mittelwert (mean) und den Median einer Liste berechnet. In der folgenden Lösung wird die Standard-Lösung aus den mitgelieferten Packages kopiert und präsentiert.
▮ Ecris un Package de *Mathematica*, qui calcule la valeur moyenne et la médiane d'une liste. Dans la solution suivante, la solution standard est copiée et présentée dans les Packages livrées.

Standard-Package einlesen:
▮ Lire le Package standard:

In[3]:= **Needs["DiscreteMath`Combinatorica`"]**

In[4]:= **?DiscreteMath`Combinatorica`***

DiscreteMath`Combinatorica`

```
AcyclicQ                    MinimumChangePermutations
AddEdge                     MinimumSpanningTree
AddEdges                    MinimumVertexColoring
AddVertex                   MinimumVertexCover
AddVertices                 MultipleEdgesQ
Algorithm                   MultiplicationTable
AllPairsShortestPath        MycielskiGraph
AlternatingGroup            NecklacePolynomial
AlternatingGroupIndex       Neighborhood
AlternatingPaths            NetworkFlow
AnimateGraph                NetworkFlowEdges
AntiSymmetricQ              NextBinarySubset
Approximate                 NextComposition
ApproximateVertexCover      NextGrayCodeSubset
ArticulationVertices        NextKSubset
Automorphisms               NextLexicographicSubset
Backtrack                   NextPartition
BellB                       NextPermutation
BellmanFord                 NextSubset
BiconnectedComponents       NextTableau
BiconnectedQ                NoMultipleEdges
BinarySearch                NonLineGraphs
BinarySubsets               NoPerfectMatchingGraph
BipartiteMatching           NormalDashed
BipartiteMatchingAndCover   NormalizeVertices
BipartiteQ                  NoSelfLoops
BooleanAlgebra              NthPair
BreadthFirstTraversal       NthPermutation
Brelaz                      NthSubset
BrelazColoring              NumberOf2Paths
Bridges                     NumberOfCompositions
ButterflyGraph              NumberOfDerangements
CageGraph                   NumberOfDirectedGraphs
CartesianProduct            NumberOfGraphs
ChangeEdges                 NumberOfInvolutions
```

```
ChangeEdges                 NumberOfInvolutions
ChangeVertices              NumberOfKPaths
ChromaticNumber             NumberOfNecklaces
ChromaticPolynomial         NumberOfPartitions
ChvatalGraph                NumberOfPermutationsByCycles
CirculantGraph              NumberOfPermutationsByInversions
CircularEmbedding           NumberOfPermutationsByType
CircularVertices            NumberOfSpanningTrees
CliqueQ                     NumberOfTableaux
CoarserSetPartitionQ        OctahedralGraph
CodeToLabeledTree           OddGraph
Cofactor                    One
CompleteBinaryTree          Optimum
CompleteGraph               OrbitInventory
CompleteKaryTree            OrbitRepresentatives
CompleteKPartiteGraph       Orbits
CompleteQ                   Ordered
Compositions                OrientGraph
ConnectedComponents         OutDegree
ConnectedQ                  PairGroup
ConstructTableau            PairGroupIndex
Contract                    Parent
CostOfPath                  ParentsToPaths
CoxeterGraph                PartialOrderQ
CubeConnectedCycle          PartitionLattice
CubicalGraph                PartitionQ
Cut                         Partitions
Cycle                       PathConditionGraph
CycleIndex                  PerfectQ
Cycles                      PermutationGraph
CycleStructure              PermutationGroupQ
Cyclic                      PermutationQ
CyclicGroup                 PermutationToTableaux
CyclicGroupIndex            PermutationType
DeBruijnGraph               PermutationWithCycle
DeBruijnSequence            Permute
Degrees                     PermuteSubgraph
DegreeSequence              PetersenGraph
DegreesOf2Neighborhood      PlanarQ
DeleteCycle                 PointsAndLines
DeleteEdge                  Polya
DeleteEdges                 PseudographQ
DeleteFromTableau           RadialEmbedding
DeleteVertex                Radius
DeleteVertices              RandomComposition
DepthFirstTraversal         RandomGraph
DerangementQ                RandomHeap
Derangements                RandomInteger
Diameter                    RandomKSetPartition
Dihedral                    RandomKSubset
DihedralGroup               RandomPartition
DihedralGroupIndex          RandomPermutation
Dijkstra                    RandomPermutation1
DilateVertices              RandomPermutation2
Directed                    RandomRGF
Distances                   RandomSetPartition
DistinctPermutations        RandomSubset
Distribution                RandomTableau
DodecahedralGraph           RandomTree
```

```
DodecahedralGraph            RandomTree
DominatingIntegerPartitionQ  RandomVertices
DominationLattice            RankBinarySubset
DurfeeSquare                 RankedEmbedding
Eccentricity                 RankGraph
Edge                         RankGrayCodeSubset
EdgeChromaticNumber          RankKSetPartition
EdgeColor                    RankKSubset
EdgeColoring                 RankPermutation
EdgeConnectivity             RankRGF
EdgeDirection                RankSetPartition
EdgeLabel                    RankSubset
EdgeLabelColor               ReadGraph
EdgeLabelPosition            RealizeDegreeSequence
Edges                        ReflexiveQ
EdgeStyle                    RegularGraph
EdgeWeight                   RegularQ
EmptyGraph                   RemoveMultipleEdges
EmptyQ                       RemoveSelfLoops
EncroachingListSet           ResidualFlowGraph
EquivalenceClasses           RevealCycles
EquivalenceRelationQ         ReverseEdges
Equivalences                 RGFQ
Euclidean                    RGFs
Eulerian                     RGFToSetPartition
EulerianCycle                RobertsonGraph
EulerianQ                    RootedEmbedding
ExactRandomGraph             RotateVertices
ExpandGraph                  Runs
ExtractCycles                SamenessRelation
FerrersDiagram               SelectionSort
FindCycle                    SelfComplementaryQ
FindSet                      SelfLoopsQ
FiniteGraphs                 SetEdgeLabels
FirstLexicographicTableau    SetEdgeWeights
FolkmanGraph                 SetGraphOptions
FranklinGraph                SetPartitionListViaRGF
FromAdjacencyLists           SetPartitionQ
FromAdjacencyMatrix          SetPartitions
FromCycles                   SetPartitionToRGF
FromInversionVector          SetVertexLabels
FromOrderedPairs             SetVertexWeights
FromUnorderedPairs           ShakeGraph
FruchtGraph                  ShortestPath
FunctionalGraph              ShortestPathSpanningTree
GeneralizedPetersenGraph     ShowGraph
GetEdgeLabels                ShowGraphArray
GetEdgeWeights               ShowLabeledGraph
GetVertexLabels              ShuffleExchangeGraph
GetVertexWeights             SignaturePermutation
Girth                        Simple
Graph                        SimpleQ
GraphCenter                  Small
GraphComplement              SmallestCyclicGroupGraph
GraphDifference              Spectrum
GraphicQ                     SpringEmbedding
GraphIntersection            StableMarriage
GraphJoin                    Star
```

```
GraphOptions                StirlingFirst
GraphPolynomial             StirlingSecond
GraphPower                  Strings
GraphProduct                Strong
GraphSum                    StronglyConnectedComponents
GraphUnion                  SymmetricGroup
GrayCode                    SymmetricGroupIndex
GrayCodeKSubsets            SymmetricQ
GrayCodeSubsets             TableauClasses
GrayGraph                   TableauQ
Greedy                      Tableaux
GreedyVertexCover           TableauxToPermutation
GridGraph                   TetrahedralGraph
GrotztschGraph              Thick
HamiltonianCycle            ThickDashed
HamiltonianPath             Thin
HamiltonianQ                ThinDashed
Harary                      ThomassenGraph
HasseDiagram                ToAdjacencyLists
Heapify                     ToAdjacencyMatrix
HeapSort                    ToCanonicalSetPartition
HeawoodGraph                ToCycles
HerschelGraph               ToInversionVector
HideCycles                  ToOrderedPairs
Highlight                   TopologicalSort
HighlightedEdgeColors       ToUnorderedPairs
HighlightedEdgeStyle        TransitiveClosure
HighlightedVertexColors     TransitiveQ
HighlightedVertexStyle      TransitiveReduction
Hypercube                   TranslateVertices
IcosahedralGraph            TransposePartition
IdenticalQ                  TransposeTableau
IdentityPermutation         TravelingSalesman
IncidenceMatrix             TravelingSalesmanBounds
InDegree                    Tree
IndependentSetQ             TreeIsomorphismQ
Index                       TreeQ
InduceSubgraph              TreeToCertificate
InitializeUnionFind         TriangleInequalityQ
InsertIntoTableau           Turan
IntervalGraph               TutteGraph
Invariants                  TwoColoring
InversePermutation          Type
InversionPoset              Undirected
Inversions                  UndirectedQ
InvolutionQ                 UnionSet
Involutions                 Uniquely3ColorableGraph
IsomorphicQ                 UnitransitiveGraph
Isomorphism                 UnrankBinarySubset
IsomorphismQ                UnrankGrayCodeSubset
Josephus                    UnrankKSetPartition
KnightsTourGraph            UnrankKSubset
KSetPartitions              UnrankPermutation
KSubsetGroup                UnrankRGF
KSubsetGroupIndex           UnrankSetPartition
KSubsets                    UnrankSubset
LabeledTreeToCode           UnweightedQ
Large                       UpperLeft
```

```
LastLexicographicTableau     UpperRight
LeviGraph                    V
LexicographicPermutations    VertexColor
LexicographicSubsets         VertexColoring
LineGraph                    VertexConnectivity
ListGraphs                   VertexConnectivityGraph
ListNecklaces                VertexCover
LNorm                        VertexCoverQ
LongestIncreasingSubsequence VertexLabel
LoopPosition                 VertexLabelColor
LowerLeft                    VertexLabelPosition
LowerRight                   VertexNumber
M                            VertexNumberColor
MakeDirected                 VertexNumberPosition
MakeGraph                    VertexStyle
MakeSimple                   VertexWeight
MakeUndirected               Vertices
MaximalMatching              WaltherGraph
MaximumAntichain             Weak
MaximumClique                WeaklyConnectedComponents
MaximumIndependentSet        WeightingFunction
MaximumSpanningTree          WeightRange
McGeeGraph                   Wheel
MeredithGraph                WriteGraph
MinimumChainPartition        Zoom
```

In[5]:= **?Grid**

```
System`Grid

MakeBoxes[
  Grid[BoxForm`matrix_/; TensorRank[Unevaluated[BoxForm`matrix], 2] == 2, BoxForm`opts___],
  BoxForm`fmt_] ^:= Function[BoxForm`e, TagBox[GridBox[BoxForm`e, BoxForm`opts], Grid]][
  Map[Function[BoxForm`entry, MakeBoxes[BoxForm`entry, BoxForm`fmt], HoldAllComplete],
   Unevaluated[BoxForm`matrix], {2}]]
```

In[6]:= **?Vertices**

```
Vertices[g] gives the embedding of graph g, that is, the coordinates
   of each vertex in the plane. Vertices[g, All] gives the embedding of the
   graph along with graphics options associated with each vertex. Mehr...
```

In[7]:= **<<Graphics`Arrow`**

In[8]:= **?Graphics`Arrow`***

Graphics`Arrow`

```
Absolute HeadCenter HeadScaling HeadWidth ZeroShape
Arrow    HeadLength HeadShape   Relative
```

In[9]:= **?Arrow**

```
Arrow[start, finish, (opts)] is a graphics primitive
   representing an arrow starting at start and ending at finish. Mehr...
```

Standard-Package anschauen:

- Regarde le Package standard:

In[10]:= **!!Graphics`Arrow`**

```
(* :Title:  Arrow Graphics Primitives *)

(* :Context: Graphics`Arrow` *)

(* :Author: John M. Novak *)

(* :Summary:
This package introduces the Arrow[start, finish] graphics
primitive and various style directives.
*)

(* :Package Version: 1.0.3 *)

(* :Mathematica Version: 2.2 *)

(* :Copyright: Copyright 1992-2005, Wolfram Research, Inc.*)

(* :History:
    V 0.9 June 1992 by John M. Novak.
    V 1.0 October 1992 by John M. Novak--substantial revisions.
    V 1.0.1 March 1994 by John M. Novak -- bug fixes, including zero check
           in the PostScript.
    V 1.0.2 February 1997 by John M. Novak -- bug fix for DisplayString.
    V 1.0.3 February 1998 by John M. Novak -- fix to allow Arrow objects
           in Epilog or Prolog of any graphics object
*)

(* :Keywords:
    Arrow, Vector, PostScript, Graphics
*)

(* :Sources:
    PostScript Language Reference Manual, Adobe Systems
*)

(* :Limitations:  Size of arrowhead cannot be taken into account
    for autoscaling of plot ranges, since determination of
    plot ranges are done in Mathematica, and the arrow heads are
    generated in pure PostScript. *)

(* :Discussion: *)

BeginPackage["Graphics`Arrow`"]

Arrow::usage =
"Arrow[start, finish, (opts)] is a graphics primitive representing an
arrow starting at start and ending at finish.";

HeadShape::usage =
"HeadShape is an option to the Arrow primitive; it specifies
the shape of the arrow's head by Automatic, which specifies that the
shape is described by the parameters HeadLength, HeadCenter, and
HeadWidth, or it can be a list of a subset of the Mathematica graphics
primitives, drawn in the coordinate system scaled by HeadScaling.  The
coordinate system is centered at the head of the arrow, with the negative
direction moving towards the tail of the arrow.";

HeadScaling::usage =
"HeadScaling is an option to the Arrow primitive; it specifies
the scaling used in the coordinate system for drawing the
arrowhead.  Automatic scales the system to the graphic, where
{0,0} is at the head of the arrow, and the system is rotated along
the arrow, and the distance between 0 and 1 is equivalent to the width of
the graphic.  Relative scales the coordinates of the arrowhead so that {0,0}
is at the head of the arrow, {-1,0} at the tail. Absolute scales to
the same coordinate system used in the device coordinate
```

```
system, rotated along the arrow, with {0,0} at the head.";

ZeroShape::usage =
"ZeroShape is an option to the Arrow primitive; it specifies
the shape of an arrow with no length (and hence no direction) in a form
similar to that of the HeadShape option.  Note that the
parameterized form of HeadShape is not available. The coordinate system
is not rotated, but is scaled to HeadScaling.  Automatic sets
the default zero arrow (a point.)";

HeadLength::usage =
"HeadLength is an option to the Arrow primitive.  It is used when
HeadShape -> Automatic. It describes the length of the arrowhead, scaled
according to HeadScaling.";

HeadWidth::usage =
"HeadWidth is an option to the Arrow primitive.  It is used when
HeadShape -> Automatic.  It describes the width of the arrowhead, relative
to the length of the arrowhead (specified by HeadLength.)";

HeadCenter::usage =
"HeadCenter is an option to the Arrow primitive.  It is used when
HeadShape -> Automatic.  It describes the location of the center of the
base of the arrowhead along the length of the arrow, as a factor of the
length of the arrowhead.  That is, if HeadCenter -> 0, the arrow will be
two lines; if HeadCenter -> 1, the arrowhead will be a perfect triangle;
otherwise, the arrowhead will be four-sided.";

Relative::usage =
"Relative is a possible value for the HeadScaling option to Arrow.
It specifies that the coordinate system in which the arrowhead is rendered
should be scaled to the length of the arrow, where {0,0} is at the head
of the arrow and {-1,0} is at the tail of the arrow.";

Absolute::usage =
"Absolute is a possible value for the HeadScaling option to Arrow.
It specifies that the device scaling should be used for the
arrowhead.";

Begin["`Private`"]

(* some global (but in private context) variables for caching the  arrow
styles. *)
{$$HeadDescriptions, $$HeadRoutines, $$ZeroDescriptions, $$ZeroRoutines};

(* A little utility function for checking numeric values. *)

numberQ[x_] := NumberQ[N[x]]

(* putting numbers in PostScript -
    To make it easy to put numbers into PostScript, this utility hacks the
    PostScript operator.  Any number followed by a string is joined to the
    string in the proper format; a number at the end is turned into a string.
*)

numtostring[num_]/;Abs[num] > 10^38 := numtostring[Sign[num] * 10^38]

numtostring[num_]/;Abs[num] < 10^-38 && Abs[num] > 0 :=
    numtostring[Sign[num] * 9.9999999^-37]

numtostring[num_] :=
    ToString[NumberForm[N[num], 8,
            ExponentFunction -> (If[Abs[#] > 7, #, Null] &),
            NumberFormat -> (If[#3 =!= "", StringJoin[#1, "e", #3], #1] &)
    ]]

Unprotect[PostScript];

PostScript[x___, y_?numberQ] :=
    PostScript[x,
        numtostring[y]]

PostScript[a___, y_?numberQ, z_String, b___] :=
```

```
    PostScript[a,
        numtostring[y] <> " " <> z,
        b
    ]

Protect[PostScript]

(* define PostScript operators *)

(* math to ps coordinates; note that this implementation is
used in part because of the sparsity of Mathematica PostScript
(e.g., transform does not exist.)  The MBeginOrig/MEndOrig
transform needs to be done every time, because this can be
different in a subgraph (i.e., I can't cache the transform
matrix once at the beginning.)  Because the transform operator
doesn't exist, I use this moveto method; unfortunately, this
requires the gsave/grestore, which undoubtedly leads to some
performance hit. *)

mathtops =
    PostScript[
        "/mathtops {",   (* stack: mathx mathy *)
            "gsave",
            "MBeginOrig",
            "moveto",    (* stack: - *)
            "MEndOrig",
            "currentpoint", (* stack : psx psy *)
            "grestore",
        "} bind def"
    ];

tocoords =
    PostScript[
        "/MAtocoords {",   (* stack: beginx beginy endx endy *)
            "mathtops 4 2 roll mathtops", (* x2 y2 x1 y1 *)
            "4 copy pop pop",    (* x2 y2 x1 y1 x2 y2 *)
            "3 -1 roll sub",     (* x2 y2 x1 x2 y2-y1 *)
            "/arry exch def",
            "exch sub",          (* x2 y2 x2-x1 *)
            "/arrx exch def",
            "arrx dup mul",    (* x2 y2 (x2-x1)^2 *)
            "arry dup mul",   (* x2 y2 (x2-x1)^2 (y2-y1)^2 *)
            "add sqrt",          (* x2 y2 (sqrt((x2-x1)^2+(y2-y1)^2) *)
            "/arrl exch def",   (* x2 y2 *)
            "translate",         (* - *)
        "} bind def"
    ];

(* The following sets up the call to the doarrow routine. *)

Arrow::bad =
"Arguments `1` to Arrow are not valid.";

Options[Arrow] =
    {HeadScaling -> Automatic,
    HeadLength -> Automatic,
    HeadCenter -> 1,
    HeadWidth -> .5,
    HeadShape -> Automatic,
    ZeroShape -> Automatic};

evalarrow[{bx_?numberQ,by_?numberQ}, {ex_?numberQ,ey_?numberQ},
        opts:((_Rule | _RuleDelayed)...)] :=
    Module[{head, zero, scale, len, cent, width,
            nbx = N[bx], nby = N[by], nex = N[ex], ney = N[ey]},
        {head, zero, scale, len, cent, width} = {HeadShape, ZeroShape,
                    HeadScaling, HeadLength, HeadCenter, HeadWidth}/.
            {opts}/.Options[Arrow];
        If[numtostring[nbx] === numtostring[nex] &&
                  numtostring[nby] === numtostring[ney],
            arrow = generatezero[zero, scale, nbx, nby],
            arrow = generatehead[head, scale, len, cent, width,
                        nbx, nby, nex, ney]
```

```
        ];
        {Line[{{bx,by},{ex,ey}}], arrow}
     ]

evalarrow[args___] :=
    (Message[Arrow::bad, {args}]; {})

(* This routine checks whether the arguments correspond to a cached
arrowhead; if so, it returns the PostScript call to the routine.  If not,
it build the PostScript string, caches the arrow description and the
PostScript, and returns the PostScript call to the cached routine.  Note
that the preparearrows[] routine is the one to emit the cached routines.
*)

generatehead[head_, scale_, len_, cent_, width_, bx_, by_, nex_, ney_] :=
    Module[{pos},
        If[(pos = Position[$$HeadDescriptions, {head, scale, len, cent, width},
                           {1}, Heads -> False]) != {},
            PostScript[bx, by, nex, ney,
                "MAarrowhead"<>ToString[ pos[[1,1]] ]],
        (* else *)
            buildhead[head, scale, len, cent, width];
            PostScript[bx, by, nex, ney,
                "MAarrowhead"<>ToString[Length[$$HeadRoutines]]]
        ]
    ]

generatezero[zero_, scale_, bx_, by_] :=
    Module[{pos},
        If[(pos = Position[$$ZeroDescriptions, {zero, scale},
                           {1}, Heads -> False]) != {},
            PostScript[bx, by, "MAarrowzero"<>ToString[ pos[[1,1]] ]],
        (* else *)
            buildzero[zero, scale];
            PostScript[bx, by, "MAarrowzero"<>ToString[Length[$$ZeroRoutines]]]
        ]
    ]

(* These build the PostScript descriptions, and add descriptions to the
proper caching variables. *)

buildhead[head_, scale_, len_, cent_, width_] :=
    Module[{routine, pshead},
        AppendTo[$$HeadDescriptions, {head, scale, len, cent, width}];
        If[head === Automatic,
            pshead = fromparams[scale, len, cent, width],
            pshead = fromdescription[head]
        ];
        routine = {PostScript["gsave", "MAtocoords", "arrl 0. eq",
                              "{ 0 0 Mdot }", "{"], arrowscale[scale],
            rotatesystem, pshead, PostScript["} ifelse", "grestore"]};
        AppendTo[$$HeadRoutines, routine]
    ]

buildzero[head_, scale_] :=
    Module[{routine, zhead, zscale},
        AppendTo[$$ZeroDescriptions, {head, scale}];
        If[head === Automatic,
            zhead = PostScript["0 0 Mdot"],
            zhead = fromdescription[head]
        ];
        If[scale === Absolute,
            zscale = arrowscale[scale],
            zscale = {}
        ];
        routine = {PostScript["gsave", "mathtops translate"], zscale,
            zhead, PostScript["grestore"]};
        AppendTo[$$ZeroRoutines, routine]
    ]

(* PostScript for rotating coordinate system *)

rotatesystem =
```

```
    PostScript[
        "[ arrx arrl div",   (* [ cos(t) *)
        "arry arrl div",    (* [ cos(t) sin(t) *)
        "-1 arry mul arrl div",
        "arrx arrl div",
        "0 0 ]",             (* [cos(t) sin(t) -sin(t) cos(t) 0 0] *)
        "concat"             (* - *)
    ];


(* This generate the primitives describing a parameterized arrowhead. *)
(* fromparams determines the default length parameter depending on scaling; it
    calls fromsparams *)

fromparams[Automatic, Automatic, c_, w_] :=
    fromsparams[.05, c, w]

fromparams[Relative, Automatic, c_, w_] :=
    fromsparams[.2, c, w]

fromparams[Absolute, Automatic, c_, w_] :=
    fromsparams[15, c, w]

fromparams[_, l_, c_, w_] := fromsparams[l,c,w]

fromsparams[l_, _?(#==0&), w_] :=
    PostScript[
        -l, (w l)/2, " moveto 0 0 lineto ",
        -l, -(w l)/2, " lineto stroke"
    ]

(* note that filled arrowheads are outlined by a line to hopefully cause a
    more graceful resizing... *)
fromsparams[l_, _?(#==1&), w_] :=
    Module[{nls = -l, ws = (w l)/2, nws = -(w l)/2},
        PostScript[
            nls, ws, "moveto 0 0 lineto",
            nls, nws, "lineto fill",
            nls, ws, "moveto 0 0 lineto",
            nls, nws, "lineto",
            nls, ws, "lineto stroke"
        ]
    ]

fromsparams[l_, c_, w_] :=
    Module[{nls = -l, ncs = -c l, ws = (w l)/2, nws = -(w l)/2},
        PostScript[
            nls, ws, "moveto 0 0 lineto",
            nls, nws, "lineto",
            ncs, "0 lineto fill",
            nls, ws, "moveto 0 0 lineto",
            nls, nws, "lineto",
            ncs, "0 lineto",
            nls, ws, "lineto stroke"
        ]
    ]

(* Create PostScript from a Mathematica-like syntax for describing head *)

fromdescription[head_] := head/.{Polygon[g_] :> frompoly[g],
    Line[l_] :> fromline[l],
    Point[p_] :> frompoint[p],
    Thickness[t_] :> PostScript[t, "w"]}

frompoly[g_] :=
    Module[{str = listtostring[g]},
        PostScript[
            str,
            "fill"
        ]
    ]

fromline[g_] :=
```

```
    Module[{str = listtostring[g]},
        PostScript[
            str,
            "stroke"
        ]
    ]

frompoint[{x_,y_}] :=
    PostScript[
        x, y, "Mdot"
    ]

listtostring[l_] :=
    With[{initial = numtostring[#1]<>" "<>numtostring[#2]<>
            " moveto " & @@ First[l]},
        Fold[#1<>numtostring[First[#2]]<>" "<>numtostring[Last[#2]]<>
                " lineto "&,initial, Rest[l]
        ]
    ]

(* PostScript for coordinate scaling for arrowhead *)

HeadScaling::bad =
"Arguments `1` to HeadScaling are not Automatic, Relative, or Absolute; using
default of Automatic.";

arrowscale[] := arrowscale[Automatic]

arrowscale[Automatic] := {}

arrowscale[Relative] :=
    PostScript[
        "arrl arrl scale"
    ]

arrowscale[Absolute] :=
    PostScript[
        "currentlinewidth 1 Mabswid",
        "currentlinewidth dup scale setlinewidth"
    ]

arrowscale[x___] :=
    (Message[HeadScaling::bad, {x}]; arrowscale[Automatic])

(* generate PostScript from cached descriptions, set up routines for preparing
to do arrows (This gets dumped into the start of the Prolog. *)

preparearrows[] :=
    Module[{heads, zeros},
        heads = MapIndexed[
            {PostScript["/MAarrowhead"<>ToString[First[#2]]<>" {"],
             #1, PostScript["} def"]}&, $$HeadRoutines];
        zeros = MapIndexed[
            {PostScript["/MAarrowzero"<>ToString[First[#2]]<>" {"],
             #1, PostScript["} def"]}&, $$ZeroRoutines];
        Flatten[{mathtops, tocoords, heads, zeros}]//.
            {x___, PostScript[a___], PostScript[b___], y___} :>
                {x,PostScript[a,b], y}
    ]

(* remember that all the $$etc. are global... *)
(* was originally coded to only operate on Graphics objects, but it
   turns out that Arrow objects can work in Epilog or Prolog, so at
   cost of poorer error checking, the code was trivially modified to
   check anything being passed to Display for an Arrow. This may
   mean trouble for design of a 3D arrow, if it shares the head Arrow. *)
Unprotect[Display];

Display[stream_,
        ((ghead_)[igarg_, igopts___])?
                    (!FreeQ[#,Arrow]&),
        opts___] :=
    Module[{oldgropts, oldgaopt, prolog, p},
```

```
        $$HeadDescriptions = {}; $$HeadRoutines = {}; $$ZeroDescriptions ={};
        $$ZeroRoutines = {};
        Unprotect[Graphics,GraphicsArray];
        oldgropts = Options[Graphics];
        oldgaopts = Options[GraphicsArray];
        Options[Graphics] = Options[Graphics]/.
                Arrow[p___] :> evalarrow[p];
        Options[GraphicsArray] = Options[GraphicsArray]/.
                Arrow[p___] :> evalarrow[p];
        prolog = (Prolog/.Flatten[{igopts}]/.Options[ghead])/.
                    Arrow[p___] :> evalarrow[p];
        garg = igarg/.Arrow[p___] :> evalarrow[p];
        gopts = {igopts}/.Arrow[p___] :> evalarrow[p];
        If[Head[prolog] === List,
            prolog = Join[preparearrows[],
                prolog],
            prolog = Append[preparearrows[], prolog]
        ];
        Display[stream,
            ghead[garg, Flatten[{Prolog -> prolog, Sequence @@ gopts}]],
            opts
        ];
        Unprotect[Graphics, GraphicsArray];
        Options[Graphics] = oldgropts;
        Options[GraphicsArray] = oldgaopts;
        Protect[Graphics, GraphicsArray];
        ghead[igarg,igopts]
    ]

Protect[Display];

Unprotect[DisplayString];

DisplayString[
        ((ghead_)[igarg_, igopts___])?
                    (!FreeQ[#,Arrow]&),
        opts___] :=
    Module[{oldgropts, oldgaopt, prolog, p, grout},
        $$HeadDescriptions = {}; $$HeadRoutines = {}; $$ZeroDescriptions ={};
        $$ZeroRoutines = {};
        Unprotect[Graphics,GraphicsArray];
        oldgropts = Options[Graphics];
        oldgaopts = Options[GraphicsArray];
        Options[Graphics] = Options[Graphics]/.
                Arrow[p___] :> evalarrow[p];
        Options[GraphicsArray] = Options[GraphicsArray]/.
                Arrow[p___] :> evalarrow[p];
        prolog = (Prolog/.Flatten[{igopts}]/.Options[ghead])/.
                    Arrow[p___] :> evalarrow[p];
        garg = igarg/.Arrow[p___] :> evalarrow[p];
        gopts = {igopts}/.Arrow[p___] :> evalarrow[p];
        If[Head[prolog] === List,
            prolog = Join[preparearrows[],
                prolog],
            prolog = Append[preparearrows[], prolog]
        ];
        grout = DisplayString[
            ghead[garg, Flatten[{Prolog -> prolog, Sequence @@ gopts}]],
            opts
        ];
        Unprotect[Graphics, GraphicsArray];
        Options[Graphics] = oldgropts;
        Options[GraphicsArray] = oldgaopts;
        Protect[Graphics, GraphicsArray];
        grout
    ]

Protect[DisplayString];

(* OK, this is an ugly hack for several "graphic characteristic determination"
   functions (FullOptions, FullGraphics, FullAxes, PlotRange).
   This is not a complete solution, since it doesn't spot Arrows that are
   set in default options. Also, FullGraphics is (for the time being) not
```

```
      being handled, since that would involve generating the proper PostScript
      primitives. Hopefully, this will all be able to be supersceded by some
      kernel design, so I am only making this temporary fix. --JMN, 10.93
*)

Unprotect[FullAxes];

FullAxes[gr_?(!FreeQ[#,Arrow]&), rest___] :=
   FullAxes[gr/.Arrow[beg_, end_, ___] :> Line[{beg, end}], rest]

Protect[FullAxes];

Unprotect[PlotRange];

PlotRange[gr_?(!FreeQ[#,Arrow]&), rest___] :=
   PlotRange[gr/.Arrow[beg_, end_, ___] :> Line[{beg, end}], rest]

Protect[PlotRange];

Unprotect[FullOptions]

FullOptions[gr_?(!FreeQ[#,Arrow]&), rest___] :=
   FullOptions[gr/.Arrow[beg_, end_, ___] :> Line[{beg, end}], rest]

Protect[FullOptions];


End[]

EndPackage[]
```

In[11]:= **?ReadList**

```
ReadList["file"] reads all the remaining expressions in a file, and returns
   a list of them. ReadList["file", type] reads objects of the specified type
   from a file, until the end of the file is reached. The list of objects read
   is returned. ReadList["file", {type1, type2, ... }] reads objects with a
   sequence of types, until the end of the file is reached. ReadList["file",
   types, n] reads only the first n objects of the specified types. Mehr...
```

In[12]:= **Options[Put]**

Out[12]= {}

In[13]:= **?NullRecords**

```
NullRecords is an option for Read and related functions which specifies whether null
   records should be taken to exist between repeated record separators. Mehr...
```

In[14]:=
```
readIn = ReadList[
    "Graphics`Arrow`",
        Record,RecordSeparators -> {"\n","\t"}];
readIn >> C:\\work\\MathematicaData\\DataRec;
readIn >> C:\\work\\MathematicaData\\DataRec.ma;
readIn = ReadList[
    "Graphics`Arrow`",
                    String];
readIn >> C:\\work\\MathematicaData\\DataOutS;
readIn >> C:\\work\\MathematicaData\\DataOutS.ma;
```

Diese Files können nun editiert und von Hand manipuliert werden. Z.B. wie folgt:

Ces fichiers peuvent être édités maintenant et manipulés à la main. P. ex. de la façon suivante:

In[20]:=
```
OpenWrite["Statisti.m"];
WriteString["Statisti.m",
"\n",
"Du sollst das Package von Hand in das File","\n",
"Statisti.m kopieren."
];   (* End WriteString   *)

Close["Statisti.m"];


(* Author:  Student Heimlich Fileklau  *)
(* History: Original version ... 1989. *)
(*Revised 1995.                        *)
(* Summary: This package computes descriptive *)
(*                           statistics *)
(* Keywords: mean, median *)

BeginPackage["AAAStatistics`"];
EndPackage[ ];

mean::usage =
    "mean[list] gives the mean of the entries \
                          in list.";
median::usage =
    "median[list] gives the median of the  \
                    entries in list.";

Begin["`Private`"];
mean[list_] := Apply[Plus, list]/Length[list] /;
               VectorQ[list] && Length[list] > 0;
median[list_] := Sort[list][[
                         (Length[list]+1)/2]] /;
               VectorQ[list] &&
                         OddQ[Length[list]];
median[list_] := Block[{s, n},
            s =Sort[list] ;
            n = Length[list] ;
            (s[[n/2]] + s[[n/2 + 1]]) / 2] /;
            VectorQ[list] && EvenQ[Length[list]];
End[ ];

(* Protect[mean, median];
SetAttributes[ mean, ReadProtected];
SetAttributes[ median,ReadProtected];  *)
```

```
General::spell1 :
 Possible spelling error: new symbol name "mean" is similar to existing symbol "Mean". Mehr…

General::spell1 :
 Possible spelling error: new symbol name "median" is similar to existing symbol "Median". Mehr…
```

In[40]:=
```
Needs["AAAStatistics`"]
```

In[41]:=
```
?AAAStatistics`*
```

```
Information::nomatch : No symbol matching AAAStatistics`* found. Mehr…
```

In[42]:=
```
?mean
```

```
mean[list] gives the mean of the entries                          in list.
```

```
In[43]:= ?median

        median[list] gives the median of the                    entries in list.

In[44]:= mean[{1,2,3,4,5,6,7,8,9}]

Out[44]= 5

In[45]:= mean[{1,2,3,4,5,6,7,8}]

Out[45]= 9/2

In[46]:= median[{1,2,3,4,5,6,7,8}]

Out[46]= 9/2

In[47]:= median[{1,2,3,4,5,6,7,8,9}]

Out[47]= 5
```

Zum Beispielprogramm: Funktioniert es immer gut und somit tadellos? - Wenn nicht, so korrigiere es! Wenn Du Erfolg hast und es besser machen kannst, so hast Du ein grosses Ziel erreicht!

Quant au programme exemple: Fonctionne-t-il toujours bien et donc sans erreurs? - Si non, corrige-le. Si tu as du succès et que tu peux le faire mieux, tu as atteint un grand but!

"Putzmaschine" einsetzen
Employer la "machine de nettoyage"

```
In[48]:= (* Old Form: Remove["Global`@*"] *)

In[49]:= Remove["Global`*"]
```