

Fachbereich Mathematik/Informatik

Mathematische Anwendersysteme Sommersemester 1999 Prof. Dr. Heinz Spindler

Fachbereich Mathematik/Informatik der Universität Osnabrück

*...Nun will ich nicht bestreiten, daß es sinnvoll ist, das Einmaleins zu beherrschen und zu wissen, wie man einfache Dreisatz- und Bruchrechnungen auszuführen hat. Aber mit mathematischem Denken hat das alle nichts zu tun. Es ist so, als würde man Menschen in die Musik einführen, indem man sie jahrelang Tonleitern üben läßt. Das Resultat wäre vermutlich lebenslanger Haß auf diese Kunst.
– Hans Magnus Enzensberger*

■ Vorwort

In dieser Veranstaltung wird eine Einführung in moderne Computeralgebrasysteme (CAS) gegeben. Unter den vielen existierenden Produkten sind wohl zur Zeit Maple und Mathematica am bekanntesten. Allerdings werden diese Systeme zur Zeit noch recht wenig in der Schule eingesetzt (siehe: Maths & Fun, Vergleich von Computeralgebrasystemen), obwohl meiner Meinung nach sowohl *Mathematica* als auch *Maple* ideal dazu geeignet sind. Der Einsatz von Computeralgebrasystemen in der Schule könnte den Mathematikunterricht revolutionieren! Er macht es möglich, auch komplexere Beispiele, die nicht mehr mit Papier und Bleistift zu berechnen sind, zu behandeln. Er schärft den Blick dafür, was konstruktive Verfahren sind, die maschinell durchführbar sind. Das mathematische Denken wird geschult, denn die erfolgreiche Behandlung eines komplexen mathematischen Problems mit Computeralgebra setzt das theoretische Verständnis des Problems voraus und führt zu umso besseren Ergebnissen, je besser man das Problem mathematisch beherrscht. Außerdem wird stumpfsinnige Rechnerei jetzt vom CAS übernommen. Termumformungen durchführen, Wertetabellen erstellen, Funktionsgraphen zeichnen, Gleichungen lösen, Grenzwerte, Ableitungen und Integrale berechnen, halt all das, was man so täglich in der Schule insbesondere die Schüler, die vorgeben in Mathe nicht durchzublicken, im Matheunterricht tun läßt, gehört mehr zu den einfachen Dingen, die jedes CAS beherrscht. Natürlich gibt es auch hier Grenzen! Der Trend, daß einfach nur Rezepte zum Lösen standardisierter Aufgaben eingesetzt werden und deren Benutzung intensiv geübt wird, dagegen die Herleitung und Entwicklung dieser Rezepte nicht ausführlich genug behandelt wird, könnte gestoppt werden. Die Herausforderung und das Faszinierende bei der Behandlung einer mathematischen Aufgabe besteht darin ihre Struktur zu erkennen, zum Beispiel die Variablen aufzuspüren, die zwischen den Variablen bestehenden Beziehungen etwa in Gleichungen auszudrücken, zu analysieren, an welcher Stelle man Computeralgebra einsetzen kann. Das schafft eine ungeheure Erweiterung der Möglichkeiten. Dinge von denen man früher nur träumte, sind jetzt im Unterricht realisierbar. Der altmodische Taschenrechner, der einmal den Rechenschieber ablöste, wird jetzt abgelöst durch das Notebook, auf dem ein CAS installiert ist.

Wir beschränken uns in dieser Veranstaltung auf *Mathematica*. Hat man ein CAS kennengelernt, so kann man sich auch mühelos in ein anderes System einarbeiten.

Die mathematischen Grundvorlesungen Lineare Algebra, Einführung in die Algebra, Analysis 1 und Analysis 2 werden vorausgesetzt. Kenntnisse in Geometrie, Funktionentheorie, Zahlentheorie sind nützlich.

■ Links zu allgemeinen Computeralgebrasystemen

Mathematica

Maple

Reduce

Axiom

Macsyma

Mathcad

Mupad

Derive

■ Links zu speziellen Computeralgebrasystemen

CoCoA

Macaulay

Macaulay2

Singular

Gap

Die kleine *Mathematica*–Fibel

Version 1.0

Datum: 15. 2. 99

■ Einleitung

Mathematica is a system for doing mathematics by computer.
– Stephen Wolfram

Dieses Mathematica–Notebook soll helfen, sich mit dem Computeralgebrasystem *Mathematica* vertraut zu machen. Beim Durcharbeiten dieser Fibel, sollte man *Mathematica* gestartet haben und alle vorgestellten Befehle selbst ausprobieren. Nur durch aktives Mitarbeiten lernt man den Umgang mit *Mathematica*. Beachtet man einige wichtige Syntaxregeln, so ist das Arbeiten mit *Mathematica* nach einer relativ kurzen Einarbeitungszeit ein Kinderspiel.

Abschnitt 1 (Getting Started) gibt einige erste Orientierungshilfen. In Abschnitt 2 (Einführung in *Mathematica*) wird dann an mehr oder weniger willkürlich zusammengestellten Beispielen vorgeführt, wie man mit *Mathematica* arbeitet. Dabei habe ich manchmal die verwendeten Befehle mit Absicht nicht im Detail erklärt. Vielmehr sollte durch Betrachten des

Outputs oder Variation des Inputs ihre Wirkungsweise klar werden. Bleibt etwas unklar, so kann man immer auf den Help Browser zurückgreifen. Sehr zu empfehlen ist auch das *Mathematica*-Buch von Wolfram, die Bibel des *Mathematica*-Benutzers. Das Buch ist auch in deutscher Sprache erschienen. (Nähere Informationen dazu findet man hier. Deutschsprachige Bücher zu *Mathematica* findet man hier.)

Die folgenden Abschnitte behandeln dann etwas systematischer den Einsatz von *Mathematica* bei der Behandlung von Themen aus verschiedenen Gebieten der Mathematik und Physik. Dabei habe ich darauf wert gelegt, daß die ausgewählten Themen auch für die Schule interessant sein könnten, etwa für Projekte in der Informatik oder den Leistungskursen in Mathematik

Jeder Abschnitt endet mit einem Aufgabenteil. Es werden Aufgaben von verschiedenem Schwierigkeitsgrad und Umfang gestellt. Die mit © versehenen Aufgaben dienen nur zum Einüben der eingeführten Befehle und enthalten nichts Tiefsinniges.

Vom Leser dieser Fibel wird erwartet, daß er schon Erfahrungen mit dem PC besitzt, weiß wie man sich mit der Maus unter Windows bewegt und schon einmal ein Textverarbeitungsprogramm wie Word benutzt hat.

■ Literatur

[1] Kevin R. Coombes et al.: The *Mathematica*-Primer, Cambridge University Press 1998 (ISBN 0-521-63715-5)

[2] Roman E. Maeder: Informatik für Mathematiker und Naturwissenschaftler, Eine Einführung mit *Mathematica*, Addison-Wesley 1993 (ISBN 3-89319-519-X)

[3] Wolfram, S: The *Mathematica*-Book, 3rd edition, Wolfram Media/Cambridge University Press 1996

[4] Wolfram, S.: Das *Mathematica*-Buch, Addison-Wesley 1998 (ISBN 3-8273-1338-4)

Weiterführende Literatur:

[5] Roman Maeder: The *Mathematica* Programmer, Academic Press 1994

[6] Richard Gass: *Mathematica* for Scientists and Engineers Using *Mathematica* to do Science, Prentice Hall 1998

■ 1. Getting Started

■ Programmstart

Auf den NT-Rechnern im PC-Raum 31/149 startet man *Mathematica* 3.0 über

Start ⇒ Programme ⇒ Mathematik & Statistik ⇒ Mathematica

Im LAN Mathematik startet man *Mathematica* unter X-Windows etwa auf dem Rechner **rowan** mit dem Befehl

mathematica30

Das *Mathematica*-Programm benötigt sehr viel 'Memory'.
Es ist deshalb ratsam, alle anderen Anwendungen zu schließen.

Das Notebook-Interface wird gestartet. Es erscheint ein Menübalken, ein Notebook-Fenster mit dem Namen "Untitled-1" und eine Palette für einige mathematische Symbole.

Man startet nun den *Mathematica*-Kern über das *Mathematica*-Menü durch

Kernel \Rightarrow Start Kernel \Rightarrow Local.

■ Schreiben im Notebook

Zunächst klickt man in das Notebook-Fenster, um das Fenster zu aktivieren.

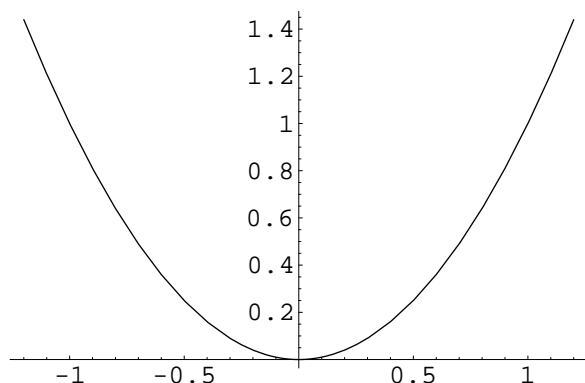
Man kann dann wie in einem Textverarbeitungsprogramm arbeiten. Gibt man etwa **37+13** ein, und drückt anschließend **Shift+Return** oder auch **Enter**, so erscheint im Fenster folgendes

```
In[1]:= 37 + 13
```

```
Out[1]= 50
```

Tippt man **Plot[x², {x, -1.2, 1.2}]**, so erhält man ein Bild der Normalparabel im Bereich $-1.2 \leq x \leq 1.2$.

```
In[2]:= Plot[x2, {x, -1.2, 1.2}]
```



```
Out[2]= - Graphics -
```

Die Potenz x^2 wird dabei mit Hilfe der Palette eingetragen. Dazu klickt man auf das Feld und tippt dann nacheinander **x** **TAB** **2** und dann, um die Zeile des Exponenten wieder zu verlassen, gleichzeitig **CTRL+SPACE**. Statt x^2 kann man aber auch (ohne Palettenbenutzung) einfach x^2 über die Tastatur eingeben.

Wählt man nun im Menü **Format** den **Style Text** aus, so befindet man sich in einer Textzelle und man kann normalen Text eingeben, den *Mathematica* nicht auswertet. (siehe Abschnitt 2 Input und Output) Befindet man sich im Textmodus, so gibt es einen kleinen Unterschied beim Eingeben von "zweidimensionalen" Ausdrücken wie x^2 . Durch Eingeben des Feldes wird innerhalb des Textes eine Grafizelle (vergleichbar dem mathematischen Modus bei *TeX*) erzeugt. Man geht dann zunächst wie oben beschrieben vor. Um dann die Grafizelle zu verlassen, muß man noch einmal die Kombination **CTRL+SPACE** drücken.

■ Speichern des Notebooks

Man wählt im *Mathematica*-Menü: **File \Rightarrow Save As**. Dann erscheint ein neues Fenster, in dem man wie gewohnt fortfährt. Der Filename sollte die Endung ".nb" besitzen. Nennen wir das Notebook etwa "Mein erstes Notebook.nb" Dann kann man es schließen, ohne die *Mathematica*-Sitzung zu verlassen, indem man **File \Rightarrow Close** wählt (oder die bekannten Windows-Mausklicks ausführt).

■ Öffnen eines Notebooks

Über **File** ⇒ **Open** kann man das vorher gespeicherte und geschlossene Notebook wieder öffnen. Es erscheint ein neues Notebook-Fenster mit dem Namen "Mein erstes Notebook.nb". Jetzt kann man Änderungen vornehmen oder weitere Befehle ausführen lassen. Mit **File** ⇒ **Save** werden die Änderungen gespeichert.

■ Der Help Browser

Wählen Sie **Help**⇒**Help** aus dem *Mathematica*-Menü. Der Help Browser wird geöffnet. Er bietet sehr umfangreiche Informationsmöglichkeiten. Zum Beispiel ist das komplette *Mathematica*-Buch (allerdings nur in englischer Sprache) zugänglich.

Um einen kleinen Eindruck zu gewinnen, beginnen Sie etwa mit

Getting Started/Demos ⇒ **Demos** ⇒ **Notebook-Demos** ⇒ **Calculating Pi**.

Sie werden etwas über die Kreiszahl π erfahren. Um die Zellen zu öffnen, muß man nur an die äußeren Zellenklammern doppelklicken.

Versuchen Sie dann:

Getting Started/Demos ⇒ **GraphicsGallery** ⇒ **Animations**

Sie erhalten eine Liste von Animationen. Klicken Sie zum Beispiel auf "Rolling Square". Es erscheint ein Bild. Klicken Sie nun auf die äußere rechte senkrechte blaue Linie. Sie wird fett-schwarz. Jetzt können Sie einfach die Tastenkombination **CTRL**+**Y** drücken (Sie können auch mit der Maus im Menü **Cell** nach **Animate Selected Graphics** suchen oder noch einfacher einfach in das Bild doppelklicken) und so die Animation starten.

■ Beenden der *Mathematica*-Sitzung

Man beendet eine *Mathematica*-Sitzung mit **File** ⇒ **Exit**

■ Aufgaben zu Abschnitt 1

⊗ 1. Stellen Sie ein Notebook mit einer oder mehreren Textzellen (siehe Menü **Format**), einer Inputzelle und der ausgewerteten Outputzelle (Enter drücken!) her. Speichern Sie das Notebook unter einem selbsterdachten Namen mit der Extension **.nb** Schließen Sie das Notebook. Öffnen Sie das Notebook wieder. Fügen Sie etwas hinzu, z.B eine Grafik mit dem Befehl **Plot**. Speichern Sie die Änderungen.

⊗ 2. Informieren Sie sich über den Befehl **Plot** mit Hilfe der verschiedenen Möglichkeiten im Help Browser. Plotten Sie die "Funktion" $x^3 - x$ im Intervall $[-2, 2]$.

⊗ 3. Machen Sie sich mit dem Menü **Edit** vertraut. Kopieren Sie Text von einem Notebookfenster in ein anderes (**CTRL**C und **CTRL**V). Öffnen Sie dazu zunächst ein neues Fenster mit **File** ⇒ **New**. Es funktioniert wie in einem Standardtextverarbeitungsprogramm.

⊗ 4. Folgen Sie dem folgenden Link im Help Browser: **Other Information** ⇒ **Contact Information** ⇒ **World Wide Web** oder hier . Man schaue hier zum Beispiel unter *MathSource*. Gibt es hier Programme zu den Themen Geometry, Calculus, Linear Algebra?

⊗ 5. Berechnen Sie nacheinander 2^{64} , $(23 + 567)^3$, $\sqrt[3]{12345}$, $\sum_{k=1}^{100} k$. (Benutzen Sie die Palette **Basic Input**)

■ 2. Mathematica-Einführung

■ Input und Output

Ein *Mathematica*-Notebook besteht aus verschiedenen Zellen. Es gibt im wesentlichen drei Typen von Zellen: Input-Zellen, Output-Zellen und Text-Zellen. Am rechten Rand des Fensters werden Zellen-Klammern angezeigt. Bewegt man die Maus in eine Zelle, so bekommt der Mauszeiger die Gestalt eines senkrechten Balkens. Zieht man die Maus zwischen zwei Zellen oder ans Ende des Notebooks, so bekommt der Mauszeiger die Form eines waagerechten Balkens. Klickt man dann auf die linke Maustaste, so entsteht eine durchgezogene waagerechte Linie, und man kann eine neue Zelle beginnen. Mit `ALT+7` wechselt man zum Textmodus. Der Zelleninhalt einer Text-Zelle wird von *Mathematica* nicht ausgewertet. Die Default-Einstellung für das Format einer Zelle ist Input. Eine Input-Zelle wird, wenn sie aktiv ist, durch gleichzeitiges Drücken von `SHIFT+RET` oder von `Enter` (ganz rechts auf der Tastatur) ausgewertet. Eine Zelle ist aktiv, wenn sich der Mauszeiger in der Zelle befindet, oder wenn die Zellen-Klammer fett-schwarz ist.

Beispiel: Man tippe $(a + b)^3$ und werte dies dann wie oben beschrieben aus. Das Ergebnis sollte so aussehen:

```
In[3]:= (a + b)3
```

```
Out[3]= (a + b)3
```

Schade! *Mathematica* beherrscht die binomischen Formeln nicht. Dem ist natürlich nicht so. Wir sollten `Expand[(a + b)3]` tippen. Dann ergibt Auswerten das gewünschte Ergebnis:

```
In[4]:= Expand[(a + b)3]
```

```
Out[4]= a3 + 3 a2 b + 3 a b2 + b3
```

Wir sehen, daß wir schon 4 Eingaben gemacht haben. Manchmal wird man Fehler machen. Dann wird eine Fehlermeldung ausgegeben, und man hat die Möglichkeit, die Eingabe zu korrigieren. Ein Beispiel:

```
In[5]:= Expand[b (a + b)3]
```

```
Syntax::bktmop : Expression "Expand[b (a + b)3]" has no opening "(".
```

```
Expand[b (a + b)3]
```

Man kann mit der Maus in die fehlerhafte Input-Zelle gehen und den Fehler verbessern.

```
In[5]:= Expand[b (a + b)3]
```

```
Out[5]= a3 b + 3 a2 b2 + 3 a b3 + b4
```

Strebt man an, Text und Input in der üblichen mathematischen Notation zu schreiben, so es ist es ratsam, die Paletten zu benutzen. Dazu gehe man zum Menü **File** ⇒ **Palettes**. Neben **Basic Input** ist die Palette **Complete Characters** wichtig. Die Bedienung ist selbsterklärend. Mit der Palette **Basic Input** kann man zum Beispiel ein bestimmtes Integral in der üblichen Form eingeben. Dabei werden nach und nach die Einträge in die geschwärtzten Felder vorgenommen. Man springt mit der Tabulatortaste `TAB` von einem Feld zum nächsten.

$$\text{In}[6] := \int_1^2 \frac{1}{1+x^2} dx$$

$$\text{Out}[6] = -\frac{\pi}{4} + \text{ArcTan}[2]$$

Man versuche es nachzumachen! Falls es nicht klappt, informiere man sich im Help Browser unter **Getting Started** \Rightarrow **Tour of Mathematica** \Rightarrow **Palettes and Buttons**.

Man kann das bestimmte Integral auch über die Tastatur eingeben. Um zu sehen wie, schaue man sich die InputForm an. Damit nichts berechnet wird, wende man außerdem **Hold** an:

$$\text{In}[7] := \text{Hold}[\text{InputForm}[\int_1^2 \frac{1}{1+x^2} dx]]$$

$$\text{Out}[7] = \text{Hold}[\text{Integrate}[1/(1+x^2), \{x, 1, 2\}]]$$

Der Befehl lautet also in InputForm:

$$\text{In}[8] := \text{Integrate}[1/(1+x^2), \{x, 1, 2\}]$$

$$\text{Out}[8] = -\frac{\pi}{4} + \text{ArcTan}[2]$$

■ Zahlen

■ Ganze Zahlen

Man kann in gewohnter Weise mit ganzen Zahlen rechnen. Das Malzeichen * kann man durch ein Blank ersetzen. Beispiel:

$$\text{In}[9] := (200 - 302) 123 - 1 + 2^{10}$$

$$\text{Out}[9] = -11523$$

Die Summe der ersten 10 natürlichen Zahlen:

$$\text{In}[10] := \sum_{k=1}^{10} k$$

$$\text{Out}[10] = 55$$

Die Summe der ersten 10 Quadratzahlen:

$$\text{In}[11] := \sum_{j=1}^{10} j^2$$

$$\text{Out}[11] = 385$$

Die Primfaktorzerlegung einer ganzen Zahl:

$$\text{In}[12] := \text{FactorInteger}[12121212]$$

$$\text{Out}[12] = \{\{2, 2\}, \{3, 1\}, \{73, 1\}, \{101, 1\}, \{137, 1\}\}$$

Mit dem Befehl **Head** erfährt man etwas über den Typ eines Ausdrucks. Der Kopf einer ganzen Zahl ist **Integer**.
Beispiel:

```
In[13]:= Head[1234501]
```

```
Out[13]= Integer
```

■ Rationale Zahlen

Gibt man einen Bruch ein, so wird seine gekürzte Form bestimmt. Beispiel:

```
In[14]:=  $\frac{24}{36}$ 
```

```
Out[14]=  $\frac{2}{3}$ 
```

Die Arithmetik mit rationalen Zahlen ergibt stets exakte Werte. Als Beispiel berechnen wir eine harmonische Zahl:

```
In[15]:=  $\sum_{k=1}^{10} \frac{1}{k}$ 
```

```
Out[15]=  $\frac{7381}{2520}$ 
```

Der Kopf eines Bruches, der nicht ganz ist, ist **Rational**. Beispiele:

```
In[16]:= Head[ $\frac{1}{2}$ ]
```

```
Out[16]= Rational
```

```
In[17]:= Head[ $\frac{4}{2}$ ]
```

```
Out[17]= Integer
```

■ Reelle Zahlen

Reelle Zahlen werden als Dezimalbrüche eingegeben. Natürlich kann man nur endliche Dezimalbrüche eingeben. Das Komma wird dabei als Punkt notiert. Auch eine ganze oder eine rationale Zahl kann als reelle Zahl dargestellt werden.
Beispiele:

```
In[18]:= {Head[2123], Head[2123.0], Head[2123.]}
```

```
Out[18]= {Integer, Real, Real}
```

```
In[19]:= Head[ $\frac{1}{2.}$ ]
```

```
Out[19]= Real
```

$\sqrt{2}$ wird als formaler Ausdruck **Power**[2, $\frac{1}{2}$] aufgefaßt und nicht als reelle Zahl. Demzufolge ergibt sich:


```
In[20]:= Head[ $\sqrt{2}$ ]
```

```
Out[20]= Power
```

Man kann aber mit dem Befehl **N** zu einem reellen Wert übergehen. Hier der Wert von $\sqrt{2}$ auf 12 Dezimalstellen genau (Die Ziffern vor dem Koma werden natürlich mitgezählt!):

```
In[21]:= N[ $\sqrt{2}$ , 12]
```

```
Out[21]= 1.41421356237
```

Eine andere Möglichkeit:

```
In[22]:=  $\sqrt{2.}$ 
```

```
Out[22]= 1.41421
```

Die Gaußklammer $[x]$ ist die größte ganze Zahl, die kleiner oder gleich x ist. Man kann auch **Floor** verwenden.

```
In[23]:= { $[123.4]$ , Floor[123.4], Floor[ $\sqrt{2}$ ], Floor[ $\frac{345}{57}$ ]}
```

```
Out[23]= {123, 123, 1, 6}
```

■ Komplexe Zahlen

Die letzte Stufe in der Hierarchie des Zahlensystems sind die komplexen Zahlen. Die Zahl $\sqrt{-1}$ wird in *Mathematica* mit dem festen Symbol **I** bezeichnet. Beispiele:

```
In[24]:= (3 + 4 I) (2 - I)
```

```
Out[24]= 10 + 5 I
```

```
In[25]:= Head[3 + 4 I]
```

```
Out[25]= Complex
```

```
In[26]:= Head[3. + 4. I]
```

```
Out[26]= Complex
```

■ Algebraische Terme

Ein Buchstabe oder ein Wort wird von *Mathematica* als ein Symbol oder eine Variable betrachtet. Aus solchen Symbolen kann man durch arithmetische Operationen, Verschachteln mit Klammern neue Ausdrücke bilden. Einige Beispiele:

```
In[27]:= Head[a]
```

```
Out[27]= Symbol
```

```
In[28]:= (a + b) (a + b) (a - b) (a - 3)
```

```
Out[28]= (-3 + a) (a - b) (a + b)2
```

In[29]:= **Expand**[%]

Out[29]= $-3a^3 + a^4 - 3a^2b + a^3b + 3ab^2 - a^2b^2 + 3b^3 - ab^3$

Mit % bezeichnet man den vorhergehenden Output. Man sollte dieses Zeichen möglichst nicht verwenden.

In[30]:= **Factor**[%]

Out[30]= $(-3 + a)(a - b)(a + b)^2$

Im folgenden Term tauchen drei Variable auf: **a**, **b** und **ab**. Dabei wird **ab** nicht etwa als Produkt von **a** und **b** gelesen, sondern als ein einziges Symbol mit dem Namen **ab**.

In[31]:= **a b² - ab²**

Out[31]= $-ab^2 + ab^2$

Das kann man mit dem Befehl **Variables** abfragen.

In[32]:= **Variables**[%]

Out[32]= {a, ab, b}

Die Namen von Variablen können beliebig lang sein, dürfen aber keine Blanks enthalten, weil ein Blank als Malzeichen gelesen wird. Ein weiteres Beispiel:

In[33]:= **Expand**[(**variablea + variableb**)²]

General::spell1 :
Possible spelling error: new symbol name "variableb" is similar to existing symbol "variablea".

Out[33]= $variablea^2 + 2variableavariabileb + variableb^2$

Mathematica hat eine Fehlermeldung ausgegeben, die darauf aufmerksam macht, daß wir uns eventuell verschrieben haben. Das ist hier nicht der Fall. Also können wir, ohne die Fehlermeldung zu beachten, fortfahren.

In der Palette **Algebraic Manipulation** findet man viele weitere Befehle zum Vereinfachen oder Umformen algebraischer Ausdrücke. Hier illustrieren wir noch die Wirkung von **Simplify**:

In[34]:= **Simplify** $\left[\frac{x^3 - y^3}{x - y}\right]$

Out[34]= $x^2 + xy + y^2$

In[35]:= **Simplify** $\left[\frac{x^4 - y^4}{x - y}\right]$

Out[35]= $x^3 + x^2y + xy^2 + y^3$

Beim nächsten Beispiel ist der Term nach dem Kürzen zu lang, deshalb bleibt der Ausdruck unverändert.

In[36]:= **Simplify** $\left[\frac{x^5 - y^5}{x - y}\right]$

Out[36]= $\frac{x^5 - y^5}{x - y}$

Manchmal bringt Ausmultiplizieren eine Vereinfachung. Beispiel:

```
In[37]:= Simplify[(1 + x + x^2) (1 - x + x^2)]
```

```
Out[37]= 1 + x^2 + x^4
```

oft aber auch nicht:

```
In[38]:= Simplify[(1 + 2 x + 6 x^2) (3 - x + x^2)]
```

```
Out[38]= (3 - x + x^2) (1 + 2 x + 6 x^2)
```

Manchmal liefert umgekehrt Faktorisieren eine Vereinfachung. Beispiel:

```
In[39]:= Simplify[(1 + 2 x + x^2) (9 - 6 x + x^2)]
```

```
Out[39]= (-3 + x)^2 (1 + x)^2
```

Ein letztes Beispiel:

```
In[40]:= (1 - x) Sum[x^k, {k, 0, 7}]
```

```
Out[40]= (1 - x) (1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7)
```

```
In[41]:= Simplify[%]
```

```
Out[41]= 1 - x^8
```

■ Hilfe

Will man Informationen zu einem Befehl, etwa **Expand** haben, so tippt man **?Expand** oder **??Expand**. Kennt man den Namen des Befehls, nach dem man sucht nicht genau, so kann man auch den Stern * verwenden. Es werden dann alle dazu passenden Befehle angezeigt. Einige Beispiele:

```
In[42]:= ?*pand
```

```
ComplexExpand LogicalExpand
Expand PowerExpand
FunctionExpand TrigExpand
```

```
In[43]:= ?Si*
```

```
Sign
Signature
SignPadding
Simplify
Sin
SingleLetterItalics
SingularityDepth
SingularValues
Sinh
SinhIntegral
SinIntegral
SixJSymbol
```

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

```
Exp
Expand
Expandable
ExpandAll
ExpandDenominator
ExpandNumerator
ExpIntegralE
ExpIntegralEi
Exponent
ExponentFunction
ExponentStep
Expression
ExpressionPacket
ExpToTrig
```

```
In[45]:= ?? 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.
```

```
Attributes[Expand] = {Protected}
```

```
Options[Expand] = {Modulus -> 0, Trig -> False}
```

■ Optionen

Viele *Mathematica*-Befehle besitzen Optionen, die verschiedene Werte annehmen können. Eine Option wird festgelegt nach dem Schema (den Pfeil entnimmt der Palette Basis Input oder man kombiniert ihn aus den Zeichen - und >) **Optionsname**→**Optionswert**.

Einige Beispiele:

1. In den folgenden Beispielen werden in der binomischen Formel die Binomialkoeffizienten modulo 2, 3 und schließlich modulo 5 reduziert.

```
In[46]:= Expand[(x + y)^5, Modulus -> 2]
```

```
Out[46]= x^5 + x^4 y + x y^4 + y^5
```

```
In[47]:= Expand[(x + y)^5, Modulus -> 3]
```

```
Out[47]= x^5 + 2 x^4 y + x^3 y^2 + x^2 y^3 + 2 x y^4 + y^5
```

```
In[48]:= Expand[(x + y)^5, Modulus -> 5]
```

```
Out[48]= x^5 + y^5
```

Der Befehl **Plot** besitzt eine Fülle von Optionen. Welche und mit welchen aktuellen Werten, kann man mit dem Befehl **Options** feststellen.

```
In[49]:= Options[Plot]
```

```
Out[49]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None, AxesOrigin -> Automatic,
  AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic, Compiled -> True,
  DefaultColor -> Automatic, Epilog -> {}, 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, Ticks -> Automatic, DefaultFont -> $DefaultFont,
  DisplayFunction -> $DisplayFunction, FormatType -> $FormatType, TextStyle -> $TextStyle}
```

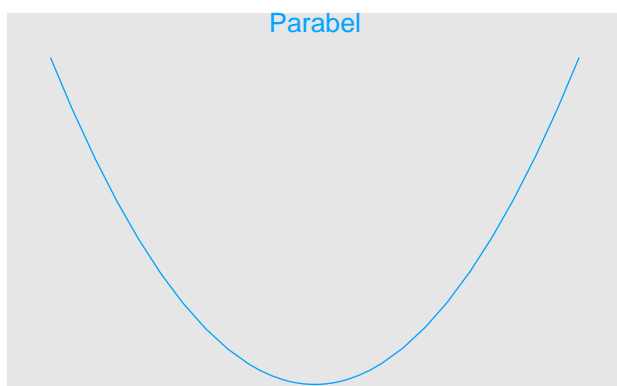
Man kann die Werte ändern. Hier einige Beispiele:

1. Die Normalparabel in der Farbe **Hue[0.56]**. Die Farbtafeln zu **Hue** findet man im *Mathematica*-Buch unter G.2.43.

Zum Aufruf `Plot[x2, {x, -1.2, 1.2}]` haben wir die Optionen

Axes, DefaultColor, Background, PlotLabel, TextStyle mit neuen Werten hinzugefügt. Etwas schwierig ist es allerdings, herauszubekommen, welche Werte für eine gegebene Option möglich sind.

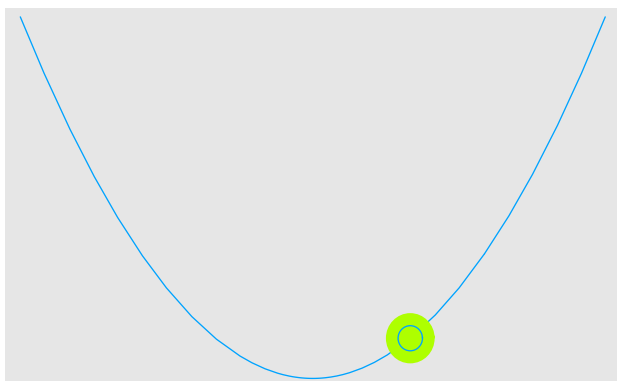
```
In[50]:= Plot[x2, {x, -1.2, 1.2}, Axes -> False,
  DefaultColor -> Hue[0.56], Background -> GrayLevel[0.9], PlotLabel -> "Parabel",
  TextStyle -> {FontFamily -> Helvetica, FontSize -> 10}]
```



```
Out[50]= - Graphics -
```

2. Die Normalparabel mit einem zusätzlich durch **Epilog**→**Grafikobjekt** eingefügten Grafikobjekt, hier: eine Kreisscheibe und eine Kreislinie. Man beachte, daß der Defaultwert der Option **Epilog** für den Befehl **Plot** leer ist. Die Optionen **Epilog** und **Prolog** sind offensichtlich sehr nützlich. Mehr darüber später.

```
In[51]:= Plot[x2, {x, -1.2, 1.2}, Axes → False, DefaultColor → Hue[0.56],
  Background → GrayLevel[0.9], Epilog → {{Hue[0.22], Disk[{0.4, 0.42}, 0.1]},
  {Hue[0.56], Circle[{0.4, 0.42}, 0.05]}}
```



```
Out[51]= - Graphics -
```

Über die verwendeten Befehle **GrayLevel**, **Hue**, **Disk**, **Circle** informiere man sich im Help Browser.

3. Mit dem Befehl **Factor** kann man zum Beispiel Polynome in irreduzible Faktoren zerlegen. Bekanntlich kommt es dabei auch auf den Grundkörper an. Mit der Option **Extension** kann man vom Körper \mathbb{Q} der rationalen Zahlen zu einem geeigneten Erweiterungskörper übergehen.

```
In[52]:= Options[Factor]
```

```
Out[52]= {Extension → None, GaussianIntegers → False, Modulus → 0, Trig → False}
```

```
In[53]:= Factor[x2 - 2]
```

```
Out[53]= -2 + x2
```

```
In[54]:= Factor[x2 - 2, Extension →  $\sqrt{2}$ ]
```

```
Out[54]= -( $\sqrt{2}$  - x) ( $\sqrt{2}$  + x)
```

```
In[55]:= Factor[x4 - 2, Extension → {I  $\sqrt{2}$ ,  $\sqrt[4]{2}$ }]
```

```
Out[55]= -(21/4 - x) (21/4 - I x) (21/4 + I x) (21/4 + x)
```

■ Namen und Zuweisungen

Einem Symbol kann man mit Hilfe eines einfachen Gleichheitszeichen einen Wert zuweisen. Der Wert kann eine Zahl, ein neues Symbol oder ein komplizierter Ausdruck sein. Beispiele:

```
In[56]:= a = 23
```

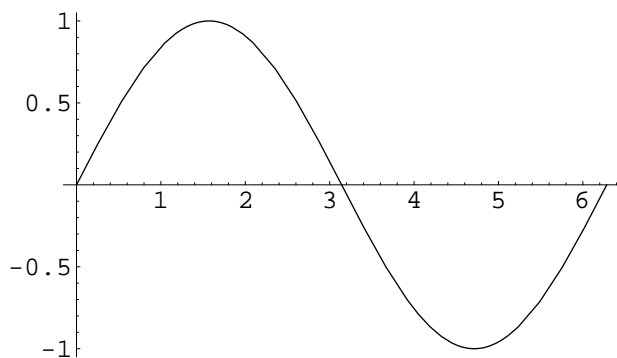
```
Out[56]= 23
```

Denselben Effekt hat

```
In[57]:= Set[a, 23]
```

```
Out[57]= 23
```

```
In[58]:= b = Plot[Sin[x], {x, 0, 2 π}]
```



```
Out[58]= - Graphics -
```

```
In[59]:= c = f[x] - x2 - 4 + y
```

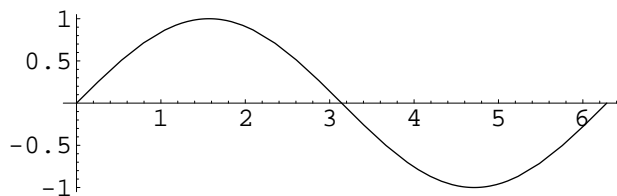
```
Out[59]= -4 - x2 + y + f[x]
```

Wenn man nun **a**, **b** oder **c** im weiteren Verlauf benutzt, werden diese durch ihre zugewiesenen Werte ersetzt. Beispielsweise erhält man:

```
In[60]:= a2 - 1
```

```
Out[60]= 528
```

```
In[61]:= Show[b, AspectRatio → Automatic]
```



```
Out[61]= - Graphics -
```

```
In[62]:= (c - y)2 // Expand
```

```
Out[62]= 16 + 8 x2 + x4 - 8 f[x] - 2 x2 f[x] + f[x]2
```

Man kann natürlich die Zuweisungen löschen oder durch neue Zuweisungen ändern. Durch **a = .** oder **Clear[a]** wird ein eventuell vorhandener Wert von **a** gelöscht.

```
In[63]:= Clear[a]
```

```
In[64]:= a
```

```
Out[64]= a
```

Man betrachte die folgende Kette von Befehlen und versuche ihre Wirkungsweise zu verstehen.

```
In[65]:= c
```

```
Out[65]= -4 - x2 + y + f[x]
```

```
In[66]:= x = π
```

```
Out[66]= π
```

```
In[67]:= f = Sin
```

```
Out[67]= Sin
```

```
In[68]:= f[x]
```

```
Out[68]= 0
```

```
In[69]:= c
```

```
Out[69]= -4 - π2 + y
```

```
In[70]:= x = .; f = .; c
```

```
Out[70]= -4 - x2 + y + f[x]
```

Es ist ratsam Wertzuweisungen aufzuheben, wenn die Werte nicht mehr gebraucht werden. So vermeidet man Fehler im späteren Verlauf einer *Mathematica*-Sitzung. Beispielsweise erhält man mit den jetztigen Werten von **b**, **c** ein merkwürdiges Ergebnis:

```
In[71]:= d = (b + c)2
```

```
Out[71]= (-4 - x2 + y + f[x] + (- Graphics -))2
```

Wir löschen die Werte:

```
In[72]:= Clear[a, b, c, f, d]
```

Wir erfragen die bisher in der *Mathematica*-Sitzung eingeführten Namen. Sie haben den Kontext **Global**.

```
In[73]:= Names["Global`*"]
```

```
Out[73]= {a, ab, b, c, d, f, Helvetica, j, k, variablea, variableb, x, y}
```

Wir entfernen alle Namen:

```
In[74]:= Map[Remove, Names["Global`*"]];
```

Jetzt gibt es keine Namen mehr:

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

```
Out[75]= {}
```


■ Gleichungen

Eine Gleichung ist eine Aussageform der Bauart **linkeseite==rechteseite**. Das Gleichheitszeichen in einer Gleichung besteht also aus zwei nebeneinanderstehenden Gleichheitszeichen **==**. Sind konkrete Werte, etwa Zahlen, gegeben, so besitzt eine Gleichung einen der beiden Wahrheitswerte **True** oder **False**. Einige Beispiele:

```
In[76]:= 2 == 3
```

```
Out[76]= False
```

Dasselbe leistet (**==** ist das Infixzeichen für den Operator **Equal**)

```
In[77]:= Equal[2, 3]
```

```
Out[77]= False
```

```
In[78]:= 2 + 2 == 4
```

```
Out[78]= True
```

```
In[79]:= Sum[1/n^2, {n, 1, Infinity}] == Pi^2/6
```

```
Out[79]= True
```

Kommen unbestimmte Symbole vor, so wird die Gleichung nicht ausgewertet. Beispiele:

```
In[80]:= x + 2 == 3
```

```
Out[80]= 2 + x == 3
```

```
In[81]:= gleichungssystem = {3 x + 4 y == 3, 6 x + 9 y == -4}
```

```
Out[81]= {3 x + 4 y == 3, 6 x + 9 y == -4}
```

Mit dem Befehl **Solve** kann man Gleichungen und Gleichungssysteme lösen. **Solve[g1, t]** löst die Gleichung **g1** nach **t** auf. **g1** darf auch eine Liste von Gleichungen und **t** darf eine Liste von Variablen sein. Für unsere beiden Beispiele ergibt sich:

```
In[82]:= Solve[x + 2 == 3, x]
```

```
Out[82]= {{x -> 1}}
```

```
In[83]:= Solve[gleichungssystem, {x, y}]
```

```
Out[83]= {{x -> 43/3, y -> -10}}
```

Die Lösungen werden in der Form von Einsetzungsregeln angegeben. Das ist auf den ersten Blick vielleicht etwas aufwendig, erweist sich aber als sehr praktisch.

Man kann die Lösungen durch Einsetzen mit Hilfe des Befehls **/.** (**ReplaceAll**) erhalten:

```
In[84]:= x /. x -> z
```

```
Out[84]= z
```

```
In[85]:= x /. Solve[x + 2 == 3, x]
```

```
Out[85]= {1}
```

```
In[86]:= {x, y} /. Solve[gleichungssystem, {x, y}]
```

```
Out[86]= {{43/3, -10}}
```

Natürlich gibt es kompliziertere Gleichungen, die man nicht symbolisch auflösen kann! Einige Beispiele mögen dies illustrieren:

1. Wenn nicht nur Polynome in der Gleichung vorkommen, klappt die Berechnung in der Regel nicht:

```
In[87]:= Solve[Sin[x] - x^2 + Sqrt[x] == 2, x]
```

```
Solve::tdep : The equations appear to involve transcendental functions of the variables in an essentially non-algebraic way.
```

```
Out[87]= Solve[Sqrt[x] - x^2 + Sin[x] == 2, x]
```

Manchmal wenigstens teilweise:

```
In[88]:= Solve[Sin[x] == Cos[x], x]
```

```
Solve::ifun : Inverse functions are being used by Solve, so some solutions may not be found.
```

```
Out[88]= {{x -> -3 Pi/4}, {x -> Pi/4}}
```

2. Bekanntlich kann man Polynomgleichungen vom Grad größer als vier im allgemeinen nicht durch Wurzeln lösen:

```
In[89]:= Solve[x^5 + 2 x^3 - x == 1, x]
```

```
Out[89]= {{x -> Root[-1 - #1 + 2 #1^3 + #1^5 &, 1]}, {x -> Root[-1 - #1 + 2 #1^3 + #1^5 &, 2]},
  {x -> Root[-1 - #1 + 2 #1^3 + #1^5 &, 3]}, {x -> Root[-1 - #1 + 2 #1^3 + #1^5 &, 4]},
  {x -> Root[-1 - #1 + 2 #1^3 + #1^5 &, 5]}}
```

In speziellen Fällen geht es doch:

```
In[90]:= Solve[-1 + 5 x - 10 x^2 + 10 x^3 - 5 x^4 + x^5 == 1, x]
```

```
Out[90]= {{x -> 2}, {x -> 3/4 - sqrt(5)/4 - 1/2 I sqrt(5/2 - sqrt(5)/2)}, {x -> 3/4 - sqrt(5)/4 + 1/2 I sqrt(5/2 - sqrt(5)/2)},
  {x -> 3/4 + sqrt(5)/4 - 1/2 I sqrt(5/2 + sqrt(5)/2)}, {x -> 3/4 + sqrt(5)/4 + 1/2 I sqrt(5/2 + sqrt(5)/2)}}
```

Für Polynome vom Grad ≤ 4 gibt es stets Lösungen durch Radikale:

`In[91]:= Solve[x4 - x2 + 12 x - 36 == 0, x]`

`Out[91]=` $\left\{ \{x \rightarrow -3\}, \{x \rightarrow 2\}, \left\{ x \rightarrow \frac{1}{2} (1 - \sqrt{23}) \right\}, \left\{ x \rightarrow \frac{1}{2} (1 + \sqrt{23}) \right\} \right\}$

Sie können allerdings sehr kompliziert sein. Ersetzt man etwa in obiger Gleichung $12x$ durch $10x$, so sind die Lösungen sehr komplizierte Ausdrücke. Wir geben die erste der vier Lösungen an:

`In[92]:= a = x /. Solve[x4 - x2 + 10 x - 36 == 0, x][[1]]`

`Out[92]=`
$$-\frac{1}{2} \sqrt{\left(\frac{2}{3} - \frac{431}{3 (53 + 30 \sqrt{88962})^{1/3}} + \frac{1}{3} (53 + 30 \sqrt{88962})^{1/3} \right) - \frac{1}{2} \sqrt{\left(\frac{4}{3} + \frac{431}{3 (53 + 30 \sqrt{88962})^{1/3}} - \frac{1}{3} (53 + 30 \sqrt{88962})^{1/3} + 20 \sqrt{\left(\sqrt{\left(\frac{2}{3} - \frac{431}{3 (53 + 30 \sqrt{88962})^{1/3}} + \frac{1}{3} (53 + 30 \sqrt{88962})^{1/3} \right)} \right)} \right)}$$

Man kann das Ergebnis auch in traditioneller Form aufschreiben:

`In[93]:= TraditionalForm[a]`

$$-\frac{1}{2} \sqrt{\frac{2}{3} - \frac{431}{3 \sqrt[3]{53 + 30 \sqrt{88962}}} + \frac{1}{3} \sqrt[3]{53 + 30 \sqrt{88962}}} - \frac{1}{2} \sqrt{\left(\frac{4}{3} + \frac{431}{3 \sqrt[3]{53 + 30 \sqrt{88962}}} - \frac{1}{3} \sqrt[3]{53 + 30 \sqrt{88962}} + \frac{20}{\sqrt{\frac{2}{3} - \frac{431}{3 \sqrt[3]{53 + 30 \sqrt{88962}}} + \frac{1}{3} \sqrt[3]{53 + 30 \sqrt{88962}}}} \right)}$$

Die Lösung ist zwar durch Wurzeln beschrieben, man sieht aber nicht annähernd, welchen numerischen Wert sie hat. Doch dafür haben wir ja den Befehl **N**. Es ergibt sich:

`In[94]:= N[a]`

`Out[94]=` -2.93217

`In[95]:= Clear[a]`

Mit dem Befehl **NSolve** kann man stets numerische Lösungen bestimmen

`In[96]:= NSolve[x5 + 2 x3 - x == 1, x]`

`Out[96]=` $\left\{ \{x \rightarrow -0.509941 - 0.452053 I\}, \{x \rightarrow -0.509941 + 0.452053 I\}, \{x \rightarrow 0.0707466 - 1.56412 I\}, \{x \rightarrow 0.0707466 + 1.56412 I\}, \{x \rightarrow 0.878389\} \right\}$

■ Eingebaute Funktionen

Mathematica kennt alle wichtigen mathematischen Funktionen. Wir begnügen uns mit einigen Beispielen. Die vollständige Liste und ausführliche Informationen erhält man im Help Browser unter **Built-in Functions**.

Zunächst laden wir mit dem Befehl **Get** oder kurz **<<** die Pakete **Graphics`Colors`** und **Graphics`Legend`**.

```
In[97]:= ?Get
```

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

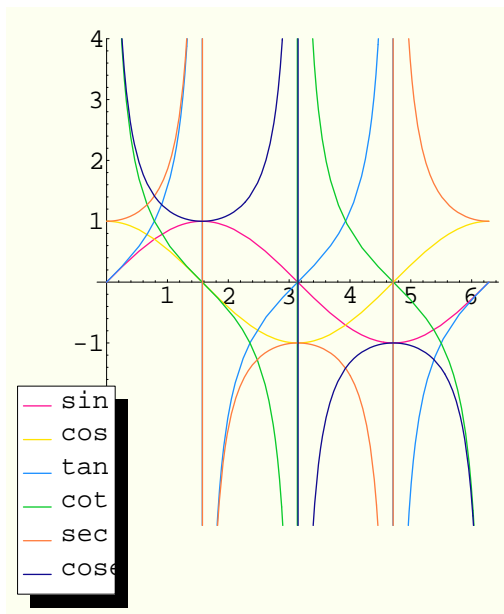
```
In[98]:= <<Graphics`Colors`
```

```
In[99]:= <<Graphics`Legend`
```

Jetzt hat man die Möglichkeit eine Grafik mit einer Legende zu versehen. Mehr zur Struktur des Befehls Plot findet man im Abschnitt Grafik mit Plot

Die trigonometrischen Funktionen:

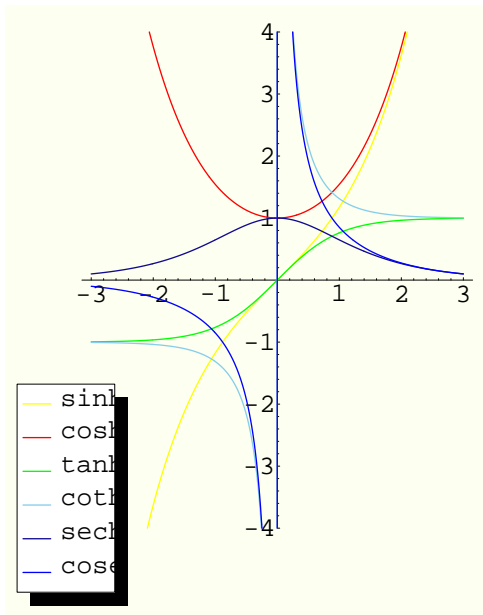
```
In[100]:= Plot[{Sin[x], Cos[x], Tan[x], Cot[x], Sec[x], Csc[x]},  
  {x, 0, 2 Pi}, AspectRatio -> Automatic, PlotRange -> {-4, 4},  
  PlotStyle -> {DeepPink, CadmiumLemon, DodgerBlue, PermanentGreen, Oak, Ultramarine},  
  Background -> Titanium, PlotLegend -> {"sin", "cos", "tan", "cot", "sec", "cosec"}]
```



```
Out[100]= - Graphics -
```

Die Hyperbelfunktinen:

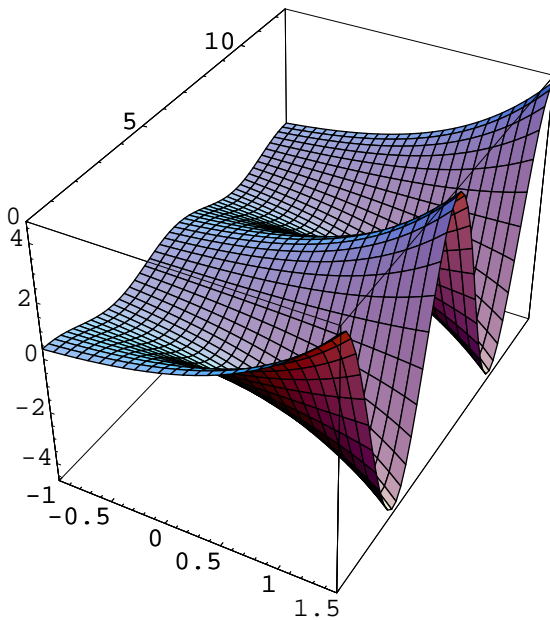
```
In[101]:= Plot[{Sinh[x], Cosh[x], Tanh[x], Coth[x], Sech[x], Csch[x]},
  {x, -3, 3}, AspectRatio -> Automatic, PlotRange -> {-4, 4},
  PlotStyle -> {Yellow, Red, Green, SkyBlue, Ultramarine, Blue}, Background -> Titanium,
  PlotLegend -> {"sinh", "cosh", "tanh", "coth", "sech", "cosech"}, PlotPoints -> 300]
```



Out[101]= - Graphics -

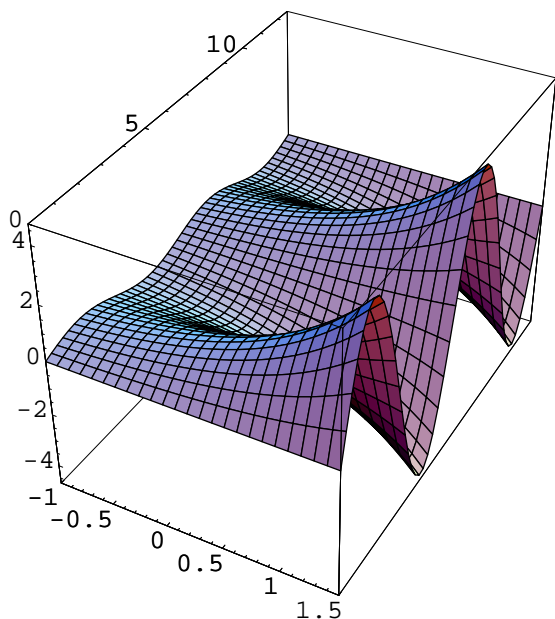
Die komplexe Exponentialfunktion (Es werden Real- und Imaginärteil dargestellt):

```
In[102]:= Plot3D[Re[Exp[x + I y]], {x, -1, 1.5}, {y, 0, 4 Pi}, BoxRatios -> {1, 1.5, 1},
  PlotPoints -> {25, 50}]
```



Out[102]= - SurfaceGraphics -

```
In[103]:= Plot3D[Im[Exp[x + I y]], {x, -1, 1.5}, {y, 0, 4 Pi}, BoxRatios -> {1, 1.5, 1},  
PlotPoints -> {25, 50}]
```



```
Out[103]= - SurfaceGraphics -
```

Der natürliche Logarithmus ist die Stammfunktion der Funktion $\frac{1}{x}$.

```
In[104]:= Log' [x]
```

```
Out[104]=  $\frac{1}{x}$ 
```

Wir definieren das n -te Taylorpolynom der Funktion $\log(x + 1)$:

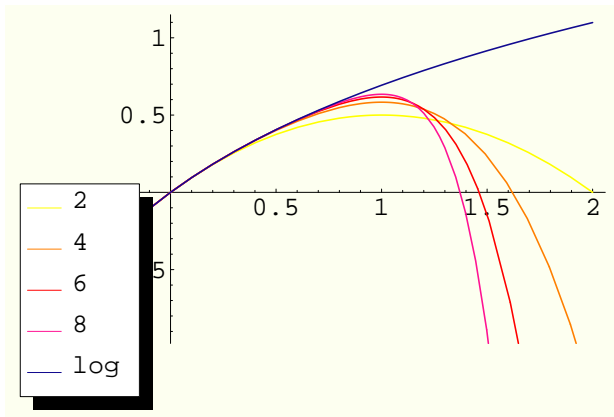
```
In[105]:= taylorpolynom[n_] := Normal[Series[Log[x + 1], {x, 0, n}]]
```

Ein Beispiel:

```
In[106]:= taylorpolynom[10]
```

```
Out[106]=  $x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \frac{x^6}{6} + \frac{x^7}{7} - \frac{x^8}{8} + \frac{x^9}{9} - \frac{x^{10}}{10}$ 
```

```
In[107]:= Plot[Evaluate[Append[Table[taylorpolynom[2 n], {n, 1, 4}], Log[x + 1]],
  {x, -0.4, 2}, PlotStyle -> First[
    Outer[List, {Thickness[0.003]}, {Yellow, Orange, Red, DeepPink, Ultramarine}]],
  Background -> Titanium, PlotLegend -> {"2", "4", "6", "8", "log"}]
```



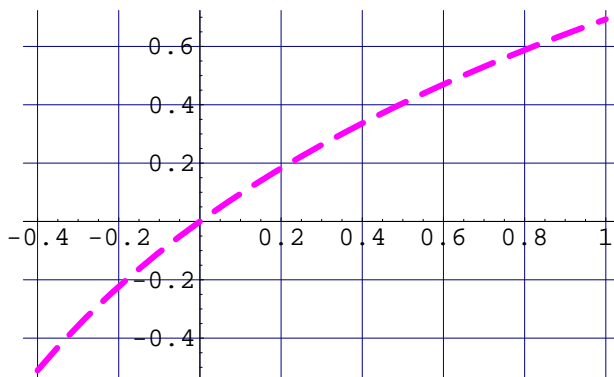
Out[107]= - Graphics -

Am Rande sei bemerkt (mehr über Listenprogrammierung folgt später): Der Befehl zur Beschreibung des Plotstils, der oben auftritt, geht so:

```
In[108]:= First[Outer[List, {a}, {b, c, d}]]
```

Out[108]= {{a, b}, {a, c}, {a, d}}

```
In[109]:= Plot[Log[x + 1], {x, -0.4, 1},
  PlotStyle -> {Thickness[0.01], Magenta, Dashing[{0.05, 0.03]}], GridLines -> Automatic]
```



Out[109]= - Graphics -

Die Namen der eingebauten *Mathematica*-Funktionen beginnen stets mit einem Großbuchstaben und sind so ausführlich, daß man in der Regel ihre Bedeutung am Namen ablesen kann. Ist ein Name zusammengesetzt, so werden die einzelnen Teile groß geschrieben, z. B. **GridLines**, **PlotPoints**, **NSolve**, aber **Background**, **Epilog**. Der Wert einer Funktion mit dem Namen *f* an einer Stelle *x* wird stets mit *f[x]* bezeichnet. Es müssen eckige Klammern benutzt werden! Es kann auch *f[x, y, option -> a]* und ähnliches vorkommen.

■ Benutzerdefinierte Funktionen

Es gibt verschiedene Möglichkeiten eigene Funktionen zu definieren. Die folgenden Beispiele definieren alle die Funktion $f(x) = x^2 + 1$.

1. Diese Methode entspricht der in der Mathematik üblichen Methode Funktionen zu definieren am ehesten. Um deutlich zu machen, daß x eine Variable ist, muß die Schreibweise $\mathbf{x_}$ benutzt werden. (Das Zeichen $\mathbf{:=}$ ist die Infixdarstellung der *Mathematica*-Funktion **SetDelayed** und bedeutet, daß die Zuweisung bei jedem Aufruf neu erfolgt. Mehr dazu später.)

```
In[110]:= f[x_] := x2 + 1
```

Setzt man nun irgendetwas ein, so wird die Vorschrift ausgeführt:

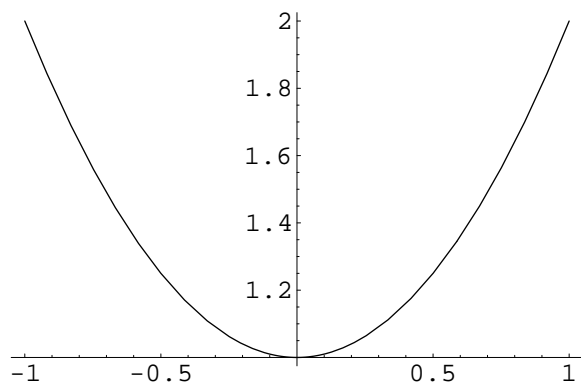
```
In[111]:= f[2]
```

```
Out[111]= 5
```

```
In[112]:= f[wert]
```

```
Out[112]= 1 + wert2
```

```
In[113]:= Plot[f[x], {x, -1, 1}];
```



2. Man kann auch die eingebaute *Mathematica*-Funktion **Function** benutzen:

```
In[114]:= g = Function[x, x2 + 1]
```

```
Out[114]= Function[x, x2 + 1]
```

Damit ist **g** diejenige Funktion, die einem Wert **x** den Funktionswert $x^2 + 1$ zuordnet. Beispiel:

```
In[115]:= g[2]
```

```
Out[115]= 5
```

Wir sehen, daß **f** und **g** dasselbe leisten:

```
In[116]:= a
```

```
Out[116]= a
```

```
In[117]:= f[a] == g[a]
```

```
Out[117]= True
```

3. Man kann schließlich ohne Bezug auf Variablenamen die Funktion als sogenannte **reine Funktion** einführen. # stellt in *Mathematica* dabei den Platzhalter für eine Variable dar. Das Zeichen & signalisiert, daß der vorstehende Term als Funktion zu lesen ist.

```
In[118]:= h = (#^2 + 1) &
```

```
Out[118]= #1^2 + 1 &
```

```
In[119]:= h[a]
```

```
Out[119]= 1 + a^2
```

4. Man kann auch zunächst dem formalen Ausdruck $1 + t^2$ einen Namen geben, etwa **p**. **p** ist dann sozusagen ein Polynom in der Unbestimmten **t**.

```
In[120]:= p = t^2 + 1
```

```
Out[120]= 1 + t^2
```

Jetzt ergibt, wie man vermuten wird, der Ausdruck **p[2]** nicht den Wert 5, sondern:

```
In[121]:= p[2]
```

```
Out[121]= (1 + t^2)[2]
```

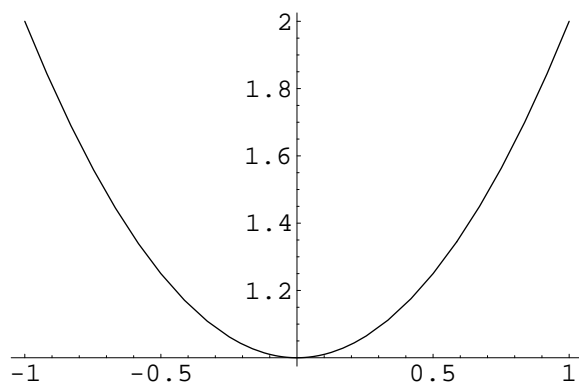
Man muß hier anders vorgehen. Einsetzen von Werten erfolgt jetzt durch Anwenden der Regel **t** → **x**. Beispiel:

```
In[122]:= p /. t -> 2
```

```
Out[122]= 5
```

Man beachte, daß man **p** plotten kann. Was ist der Unterschied zu dem entsprechenden Befehl für **f**?

```
In[123]:= Plot[p, {t, -1, 1}]
```



```
Out[123]= - Graphics -
```

Verwandeln wir nun **p** in eine Funktion:

```
In[124]:= q[x_] := p /. t -> x
```

Beispiele:

```
In[125]:= {q[2], q[wert]}
```

```
Out[125]= {5, 1 + wert2}
```

```
In[126]:= ?q
```

```
Global`q
```

```
q[x_] := p /. t -> x
```

Wir löschen die Definition von **q**:

```
In[127]:= Clear[q]
```

Jetzt wird nichts mehr berechnet:

```
In[128]:= {q[2], q[wert]}
```

```
Out[128]= {q[2], q[wert]}
```

Ab und an aufräumen:

```
In[129]:= Names["Global`*"]
```

```
Out[129]= {a, b, c, d, f, g, gleichungssystem, h, n, p, q, t, taylorpolynom, wert, x, y, z}
```

```
In[130]:= ToExpression[%]
```

```
Out[130]= {a, b, c, d, f, Function[x, x2 + 1], {3 x + 4 y == 3, 6 x + 9 y == -4}, #12 + 1 &, n,  
1 + t2, q, t, taylorpolynom, wert, x, y, z}
```

```
In[131]:= Map[Remove, Names["Global`*"]];
```

■ Listen

Eine *Mathematica*-Liste der Länge n ist ein geordnetes n -tupel irgendwelcher *Mathematica*-Ausdrücke: Anstelle der in der Mathematik üblichen runden Klammern werden in *Mathematica* geschweifte Klammern verwendet. Beispiele:

```
In[132]:= A = {a, b, c}
```

```
Out[132]= {a, b, c}
```

```
In[133]:= B = {b, c, a}
```

```
Out[133]= {b, c, a}
```

Es kommt auf die Reihenfolge an:

```
In[134]:= a = 1; b = 2; c = 3; A == B
```

```
Out[134]= False
```

```
In[135]:= a = .; b = .; c = .;
```

Wichtig ist die Funktion **Part**. Beispiele:

```
In[136]:= Part[A, 2]
```

```
Out[136]= b
```

Dasselbe erhält man mit:

```
In[137]:= A[[2]]
```

```
Out[137]= b
```

Weitere Beispiele mit verschachtelten Listen:

```
In[138]:= Part[{{x, y}, {a, b, c}}, 1]
```

```
Out[138]= {x, y}
```

```
In[139]:= Part[{{x, y}, {a, b, c}}, 1, 2]
```

```
Out[139]= y
```

oder gleichbedeutend:

```
In[140]:= {{x, y}, {a, b, c}}[[1, 2]]
```

```
Out[140]= y
```

```
In[141]:= {{x, y}, {a, b, c}}[[2, 2]]
```

```
Out[141]= b
```

```
In[142]:= M = {{{a}, {{c, b}, x}, y, t}}, r}
```

```
Out[142]= {{{a}, {{c, b}, x}, y, t}}, r}
```

```
In[143]:= M[[1, 1, 2]]
```

```
Out[143]= {{c, b}, x}
```

Man kann die Baumstruktur von **M** anzeigen:

```
In[144]:= TreeForm[M]
```

```
Out[144]//TreeForm=
  List[ |
    List[ |
      List[ |
        List[a]
      List[ |
        List[c, b]
      ], x]
    ], y, t]
  ], r]
```

```
In[145]:= M[[1, 1, 1]]
```

```
Out[145]= {a}
```

```
In[146]:= Clear[M]
```

Viele Funktionen kann man auch auf Listen anwenden, z. B. die Sinusfunktion:

```
In[147]:= Attributes[Sin]
```

```
Out[147]= {Listable, NumericFunction, Protected}
```

```
In[148]:= Sin[A]
```

```
Out[148]= {Sin[a], Sin[b], Sin[c]}
```

Das ermöglicht es, auf sehr einfache Weise eine Wertetabelle von **Sin** zu berechnen. Mit dem Befehl **Table** stellt man zunächst eine Liste der gewünschten x -Werte her. Darauf wendet man **Sin** an. Mit dem Befehl **Transpose** kann man dann aus zwei Listen mit jeweils n Elementen eine Liste aus n Paaren machen.

Die Liste der Zahlen von 0 bis 6 in Abständen 0.5:

```
In[149]:= tabelle = Table[x, {x, 0, 6, 0.5}]
```

```
Out[149]= {0, 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5, 6.}
```

```
In[150]:= wertetabelle = Transpose[{tabelle, Sin[tabelle]}];
```

Mit dem Befehl **TableForm** wird die Tabelle auch wirklich in Tabellenform angezeigt:

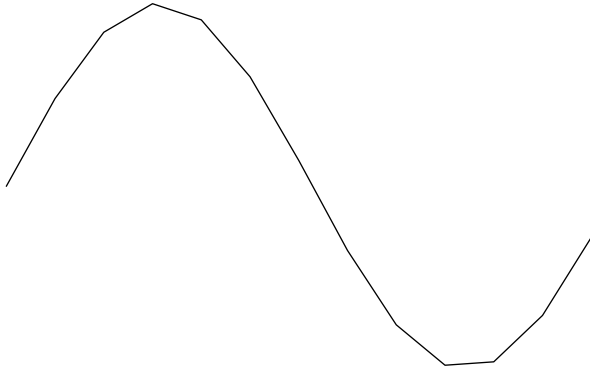
```
In[151]:= TableForm[wertetabelle]
```

```
Out[151]//TableForm=
```

0	0
0.5	0.479426
1.	0.841471
1.5	0.997495
2.	0.909297
2.5	0.598472
3.	0.14112
3.5	-0.350783
4.	-0.756802
4.5	-0.97753
5.	-0.958924
5.5	-0.70554
6.	-0.279415

Man kann diese Wertetabelle auch plotten:

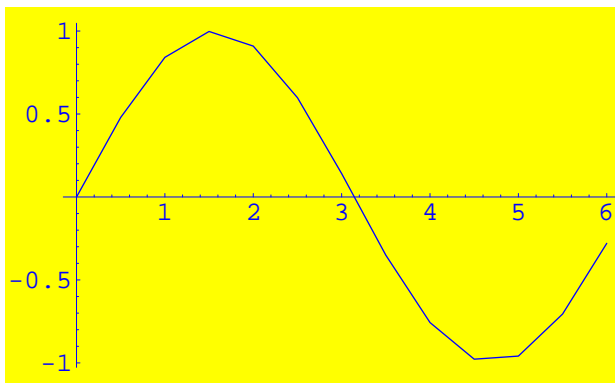
```
In[152]:= Show[Graphics[Line[wertetabelle]]]
```



```
Out[152]= - Graphics -
```

Was ist passiert? Die Punkte der Wertetabelle sind durch Strecken miteinander verbunden worden. Dazu diente der Befehl **Line**. Mit **Show[Graphics[■]]** wurde schließlich das Grafikobjekt **■ = Line[wertetabelle]** gezeichnet. Jetzt folgt dasselbe noch einmal mit Achsenkreuz und gelben Hintergrund:

```
In[153]:= Show[Graphics[Line[wertetabelle], Axes → True, Background → RGBColor[1, 1, 0]]]
```

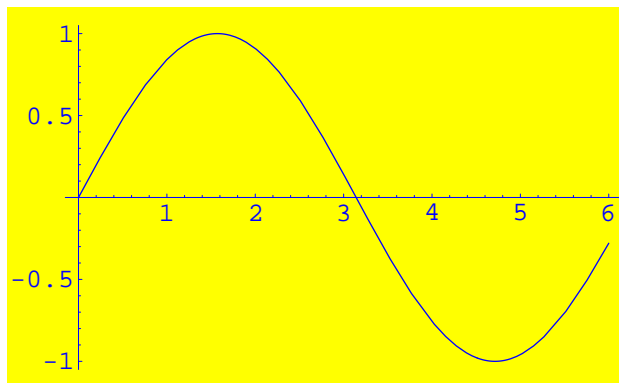


```
Out[153]= - Graphics -
```

Das Baumuster des letzten Befehls ist **f[g[h[x], a→b, c→d[r, g, t]]]**.

Bemerkung: (Dies kann man zunächst getrost überlesen) Auch der Befehl **Plot** erzeugt eine Liste von Koordinatenpaaren, allerdings werden die x -Werte auf raffinierte Weise gewählt, so daß ein glattes Bild entsteht. Wir wollen uns das einmal anschauen: Mit dem Befehl **InputForm** erhält man die gewünschte Information, die wir hier wegen seiner Länge nicht auf dem Bildschirm ausgeben (deshalb das Semikolon am Ende):

```
In[154]:= sinus = InputForm[Plot[Sin[x], {x, 0, 6}, Background → RGBColor[1, 1, 0]]];
```



Wir können einen Teil anschauen. Dazu ist der Befehl **Short** nützlich. Die Kurzinformation zu **sinus**:

```
In[155]:= Short[sinus, 25]
```

```
Out[155]//Short=
Graphics[{{Line[{{2.4999999999999999*^-7,
  2.49999999999999974*^-7}, {0.2434019494374947, 0.2410056872346213},
  {0.5088527991562421, 0.4871757126293376}, {0.7581561980312317,
  0.687583820773845}, {0.9979104466249685, 0.840340158094053},
  {1.12121430784254, 0.9006288310392944}, {1.2531155449374 ... 9499624130431747},
  {1.317477067661326, 0.9680858878848361}, {1.351389000372447, 0.9760266168566474},
  <<69>>, {5.26559505312316, -0.8508443701505079}, {5.51416303062295,
  -0.6954329879144458}, {5.753181857841487, -0.5055363173151401},
  {5.999999749999999, -0.279415738241489}}]}], {<<25>>}]
```

Das Symbol **sinus** hat jetzt einen komplizierten verschachtelten *Mathematica*-Ausdruck als Wert. An der Stelle (1,1,1,1,1) steht die Liste der berechneten Wertepaare. Wir setzen deshalb

```
In[156]:= wertetabelle2 = Part[sinus, 1, 1, 1, 1, 1];
```

Zurück zu unserem Beispiel. Es wurden 82 Wertepaare berechnet, wie man mit dem Befehl **Length** nachprüft:

```
In[157]:= Length[wertetabelle2]
```

```
Out[157]= 82
```

Wir lassen uns die ersten 6 Paare in Tabellenform anzeigen

```
In[158]:= wertetabelle2[1, 2, 3, 4, 5, 6] // TableForm
```

```
Out[158]//TableForm=
  2.5 × 10-7      2.5 × 10-7
  0.243402      0.241006
  0.508853      0.487176
  0.758156      0.687584
  0.99791       0.84034
  1.12121       0.900629
```

Den n -ten x -Wert der Wertetabelle bekommt man durch die folgende Definition:

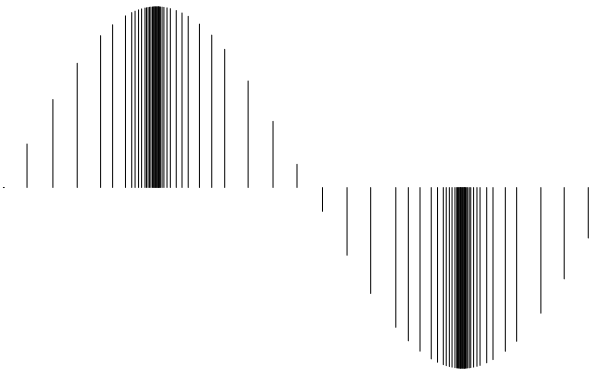
```
In[159]:= x[n_] := wertetabelle2[n, 1]
```

```
In[160]:= M = Table[x[n], {n, 1, 10}]
```

```
Out[160]= {2.5 × 10-7, 0.243402, 0.508853, 0.758156, 0.99791, 1.12121, 1.25312, 1.31748,  
1.35139, 1.38734}
```

Das folgende Bild zeigt wie die x -Werte der Sinuefunktion angepaßt sind.

```
In[161]:= Show[Graphics[  
Table[{Thickness[0.0001], Line[{{x[n], 0}, {x[n], Sin[x[n]]}}]}, {n, 1, 82}]]];
```



Man kann auch elementare Mengenlehre betreiben. In den Paletten findet man die Zeichen für die mengentheoretischen Operationen. Einige Beispiele:

```
In[162]:= Clear[a, b, c, d, e, f, g, h, x, y, z, u, v, w, r, s, t]
```

Wir definieren Mengen \mathcal{A} , \mathcal{B} , \mathcal{C} .

```
In[163]:= A = {a, b, c, d, e, f, g, h}
```

```
Out[163]= {a, b, c, d, e, f, g, h}
```

```
In[164]:= B = {e, f, g, h, x, y, z, u, v}
```

```
Out[164]= {e, f, g, h, x, y, z, u, v}
```

```
In[165]:= C = {b, c, d, z, u, v, w, r, s, t}
```

```
Out[165]= {b, c, d, z, u, v, w, r, s, t}
```

Der Durchschnitt:

```
In[166]:= A ∩ B
```

```
Out[166]= {e, f, g, h}
```

Die Vereinigung

```
In[167]:= A ∪ C
```

```
Out[167]= {a, b, c, d, e, f, g, h, r, s, t, u, v, w, z}
```

Ein dreifacher Durchschnitt

```
In[168]:=  $\mathcal{A} \cap \mathcal{B} \cap \mathcal{C}$ 
```

```
Out[168]= {}
```

Der Unterschied zwischen **Union** und **Join**.

```
In[169]:= Union[ $\mathcal{A}$ ,  $\mathcal{B}$ ]
```

```
Out[169]= {a, b, c, d, e, f, g, h, u, v, x, y, z}
```

```
In[170]:= Join[ $\mathcal{A}$ ,  $\mathcal{B}$ ]
```

```
Out[170]= {a, b, c, d, e, f, g, h, e, f, g, h, x, y, z, u, v}
```

```
In[171]:= Union[{d, a, a, b, b, c, c}]
```

```
Out[171]= {a, b, c, d}
```

Die Mengendifferenz.

```
In[172]:= Complement[ $\mathcal{A}$ ,  $\mathcal{B}$ ]
```

```
Out[172]= {a, b, c, d}
```

Aufgabe: Aus einer Menge von Zahlen zwischen 1 und 100 picke man die geraden Zahlen heraus. Es gibt verschiedene Lösungen. Zunächst wählen wir zufällig 40 natürliche Zahlen zwischen 1 und 100:

```
In[173]:= M = Table[Random[Integer, {1, 100}], {40}]
```

```
Out[173]= {74, 64, 75, 33, 18, 35, 4, 2, 41, 53, 30, 88, 77, 49, 10, 8, 67, 67, 87, 36, 7,  
90, 2, 58, 31, 68, 34, 21, 74, 29, 58, 99, 96, 30, 45, 86, 36, 63, 46, 13}
```

Wie entfernen die Wiederholungen und ordnen der Größe nach:

```
In[174]:= M = Union[M]
```

```
Out[174]= {2, 4, 7, 8, 10, 13, 18, 21, 29, 30, 31, 33, 34, 35, 36, 41, 45, 46, 49, 53, 58,  
63, 64, 67, 68, 74, 75, 77, 86, 87, 88, 90, 96, 99}
```

Aus der Menge **M** sollen nun die geraden Zahlen herausgesucht werden:

1. Lösung: Die Liste aller geraden Zahlen zwischen 1 und 100:

```
In[175]:= G = Table[2 x, {x, 1, 50}];
```

```
In[176]:= G  $\cap$  M
```

```
Out[176]= {2, 4, 8, 10, 18, 30, 34, 36, 46, 58, 64, 68, 74, 86, 88, 90, 96}
```

2. Lösung: Die Liste der geraden Zahlen können wir auch dadurch erhalten, daß wir die Liste **Range**[1, 50] mit 2 multiplizieren. Wenn man sich im Help Browser über den Befehl **Range** informiert, findet man eine noch einfachere Methode.

```
In[177]:= 2 Range[1, 50]  $\cap$  M
```

```
Out[177]= {2, 4, 8, 10, 18, 30, 34, 36, 46, 58, 64, 68, 74, 86, 88, 90, 96}
```

3. Lösung: Am elegantesten ist es wohl, die Eigenschaft "gerade" durch die Funktion `Mod[#, 2] == 0` auszudrücken und dann den *Mathematica*-Befehl `Select` anzuwenden.

```
In[178]:= Select[M, Mod[#, 2] == 0 &]
```

```
Out[178]= {2, 4, 8, 10, 18, 30, 34, 36, 46, 58, 64, 68, 74, 86, 88, 90, 96}
```

■ Vektoren und Matrizen

Eine **Matrix** ist die Liste ihrer Zeilen. Ein Beispiel:

```
In[179]:= M1 = {{1, 2, 3}, {3, 4, 5}}; MatrixForm[M1]
```

```
Out[179]//MatrixForm=
  ( 1 2 3 )
  ( 3 4 5 )
```

Man beachte: Mit M_1 kann man rechnen, mit `MatrixForm[M1]` nicht!

Man kann Matrizen über das Menü **Input**⇒**Create Matrix** eingeben. Mit der Maus und mit der Tabulator-Taste kann man sich in der Matrix bewegen. Man versuche es. Beispiel:

```
In[180]:= M2 = ( 1 2 4 )
              ( 6 4 9 )
              ( 1 0 5 )
```

```
Out[180]= {{1, 2, 4}, {6, 4, 9}, {1, 0, 5}}
```

In der ersten Zeile an der zweiten Stelle steht:

```
In[181]:= M2[[1, 2]]
```

```
Out[181]= 2
```

Das Matrizenprodukt wird durch einen Punkt "." (**Dot**) bezeichnet. Man beachte, daß man hier auf keinen Fall den Punkt weglassen darf!

```
In[182]:= M1 . M2
```

```
Out[182]= {{16, 10, 37}, {32, 22, 73}}
```

Ohne Punkt geht es hier nicht:

```
In[183]:= M1 M2
```

```
Thread::tdlen : Objects of unequal length in {{1, 2, 3}, {3, 4, 5}} <<1>> cannot be combined.
```

```
Out[183]= {{1, 2, 3}, {3, 4, 5}} {{1, 2, 4}, {6, 4, 9}, {1, 0, 5}}
```

Ein weiteres Beispiel:

```
In[184]:= M3 =  $\begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix}$ ; M2 . M3; MatrixForm[%]
```

```
Out[184]//MatrixForm=
  {{1, 2, 3}, {3, 4, 5}} {{1, 2, 4}, {6, 4, 9}, {1, 0, 5}}
```

Läßt man den Punkt weg, so wird diesmal keine Fehlermeldung ausgegeben, da die Operation möglich ist, weil beide Listen dieselbe Länge haben. Es wird einfach komponentenweise multipliziert, solange dies möglich ist. Hier werden einfach alle Einträge von M_2 und M_3 komponentenweise multipliziert, was natürlich nicht das Matrizenprodukt ist! Es ist also bei der Matrizenmultiplikation unbedingt auf den Punkt zu achten.

```
In[185]:= TableForm[{{MatrixForm[M2], MatrixForm[M3], MatrixForm[M2 M3 ]}}]
```

```
Out[185]//TableForm=
   $\begin{pmatrix} 1 & 2 & 4 \\ 6 & 4 & 9 \\ 1 & 0 & 5 \end{pmatrix}$        $\begin{pmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{pmatrix}$        $\begin{pmatrix} a & 2b & 4c \\ 0 & 4d & 9e \\ 0 & 0 & 5f \end{pmatrix}$ 
```

```
In[186]:= M1 * = Transpose[M1]
```

```
Out[186]= {{1, 3}, {2, 4}, {3, 5}}
```

```
In[187]:= M2 M1 *
```

```
Thread::tdlen : Objects of unequal length in {1, 2, 4} {1, 3} cannot be combined.
Thread::tdlen : Objects of unequal length in {6, 4, 9} {2, 4} cannot be combined.
Thread::tdlen : Objects of unequal length in {1, 0, 5} {3, 5} cannot be combined.
General::stop : Further output of Thread::tdlen will be suppressed during this calculation.
```

```
Out[187]= {{1, 3} {1, 2, 4}, {2, 4} {6, 4, 9}, {3, 5} {1, 0, 5}}
```

```
In[188]:= M2 {x, y, z}
```

```
Out[188]= {{x, 2 x, 4 x}, {6 y, 4 y, 9 y}, {z, 0, 5 z}}
```

```
In[189]:= {x, y, z} M2
```

```
Out[189]= {{x, 2 x, 4 x}, {6 y, 4 y, 9 y}, {z, 0, 5 z}}
```

Ein **Vektor** ist die Liste seiner Koeffizienten. Es wird hier nicht zwischen Zeilen- und Spaltenvektor unterschieden! Man kann Vektoren und Matrizen miteinander multiplizieren. Das Ergebnis ist wieder ein Vektor. Man kann praktischerweise von links und rechts multiplizieren. Das ist ein Unterschied zur üblichen mathematischen Notation, an den man sich gewöhnen muß.

```
In[190]:= M2 . {x, y, z}
```

```
Out[190]= {x + 2 y + 4 z, 6 x + 4 y + 9 z, x + 5 z}
```

```
In[191]:= {x, y, z} . M2
```

```
Out[191]= {x + 6 y + z, 2 x + 4 y, 4 x + 9 y + 5 z}
```

Ein **Zeilenvektor** ist eine Matrix mit nur einer Zeile, d. h. eine $1 \times n$ -Matrix. Man achte auf das zusätzliche Klammerpaar!

```
In[192]:= w = {{x, y, z}}
```

```
Out[192]= {{x, y, z}}
```

Eine Verallgemeinerung des Befehls **Length** ist der Befehl **Dimensions**, mit dem man bequem den Typ einer Matrix bestimmen kann. (In *Mathematica* werden Matrizen als spezielle Tensoren betrachtet.)

```
In[193]:= Dimensions[w]
```

```
Out[193]= {1, 3}
```

```
In[194]:= Dimensions[{x, y, z}]
```

```
Out[194]= {3}
```

```
In[195]:= Dimensions[{{{x, y, z}}}]
```

```
Out[195]= {1, 1, 1, 3}
```

```
In[196]:= Dimensions[M2]
```

```
Out[196]= {3, 3}
```

Das Matrizenprodukt $M_2 \cdot w$ kann man nicht bilden, wohl aber $w \cdot M_2$.

```
In[197]:= M2 . w
```

```
Dot::dotsh : Tensors {{1, 2, 4}, {6, 4, 9}, {1, 0, 5}} and {{x, y, z}} have incompatible shapes.
```

```
Out[197]= {{1, 2, 4}, {6, 4, 9}, {1, 0, 5}} . {{x, y, z}}
```

```
In[198]:= w . M2
```

```
Out[198]= {{x + 6 y + z, 2 x + 4 y, 4 x + 9 y + 5 z}}
```

Ein **Spaltenvektor** ist eine Matrix mit Zeilen der Länge 1, also eine $n \times 1$ -Matrix.

```
In[199]:= v = Transpose[w]
```

```
Out[199]= {{x}, {y}, {z}}
```

```
In[200]:= Dimensions[v]
```

```
Out[200]= {3, 1}
```

```
In[201]:= M2 . v
```

```
Out[201]= {{x + 2 y + 4 z}, {6 x + 4 y + 9 z}, {x + 5 z}}
```

Mehr zur Matrizenrechnung in Abschnitt 3.

■ Grafik mit dem Befehl Plot

In den vorangegangenen Abschnitten haben wir schon einige Grafikbefehle benutzt. Grafik ist eine der Stärken von *Mathematica*. Es gibt eine Fülle von Möglichkeiten. Der Befehl **Plot** zeichnet, wie wir schon gesehen haben, Funktionsgraphen. Wir wollen eine Familie von Funktionen zeichnen und dabei einige der vielen Möglichkeiten kennenlernen.

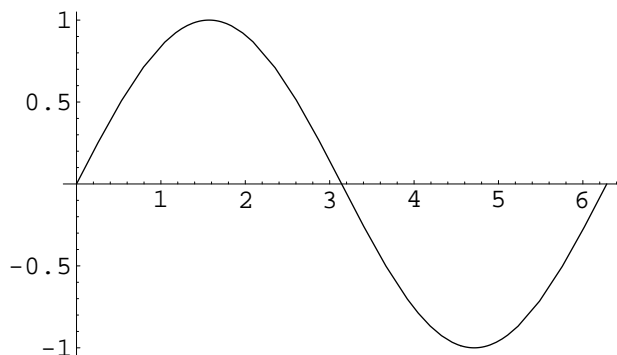
```
In[202]:= F = Table[Sin[ $\frac{k}{10} x$ ], {k, 1, 10}]
```

```
Out[202]= {Sin[ $\frac{x}{10}$ ], Sin[ $\frac{x}{5}$ ], Sin[ $\frac{3x}{10}$ ], Sin[ $\frac{2x}{5}$ ], Sin[ $\frac{x}{2}$ ], Sin[ $\frac{3x}{5}$ ], Sin[ $\frac{7x}{10}$ ],  
Sin[ $\frac{4x}{5}$ ], Sin[ $\frac{9x}{10}$ ], Sin[x]}
```

```
In[203]:= f = F[[10]]
```

```
Out[203]= Sin[x]
```

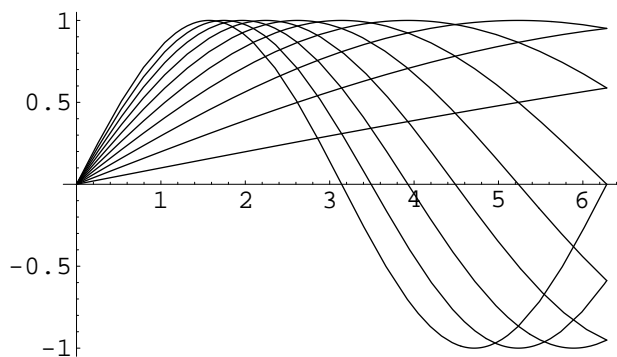
```
In[204]:= Plot[f, {x, 0, 2  $\pi$ ];
```



Das Semikolon bewirkt, daß nur geplottet wird, aber kein Output gezeigt wird.

Um die Liste **F** plotten zu können, muß man diese zunächst auswerten.

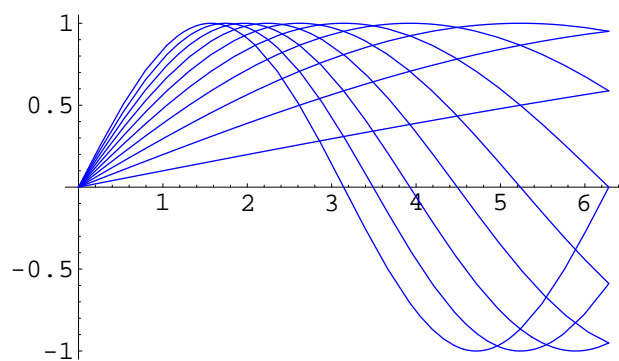
```
In[205]:= Plot[Evaluate[F], {x, 0, 2  $\pi$ ];
```



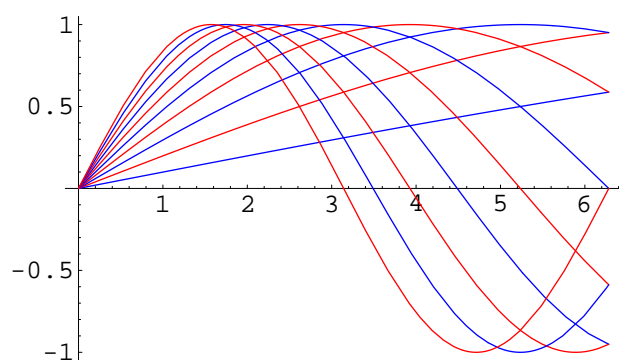
Wir verändern die Option **PlotStyle**. Man beachte jeweils die Wirkung.

```
In[206]:= Needs["Graphics`Colors`"]
```

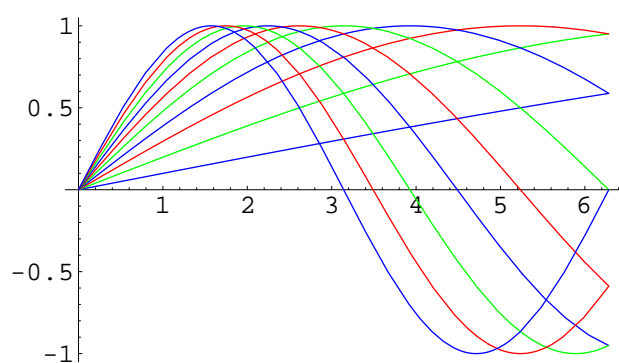
```
In[207]:= Plot[Evaluate[F], {x, 0, 2  $\pi$ }, PlotStyle -> {Blue}];
```



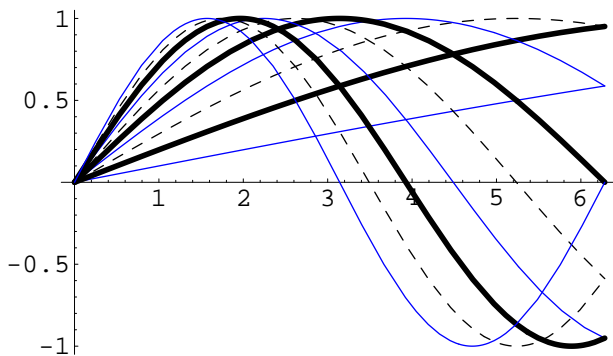
```
In[208]:= Plot[Evaluate[F], {x, 0, 2  $\pi$ }, PlotStyle -> {Blue, Red}];
```



```
In[209]:= Plot[Evaluate[F], {x, 0, 2  $\pi$ }, PlotStyle -> {Blue, Green, Red}];
```

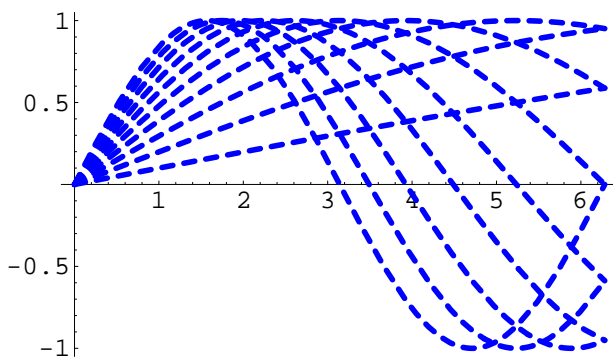


```
In[210]:= Plot[Evaluate[F], {x, 0, 2 π},
  PlotStyle → {Blue, Thickness[0.01], Dashing[{0.02, 0.02}]}];
```



Was ein zusätzliches Klammerpaar alles bewirken kann! Machen Sie sich die Struktur klar!

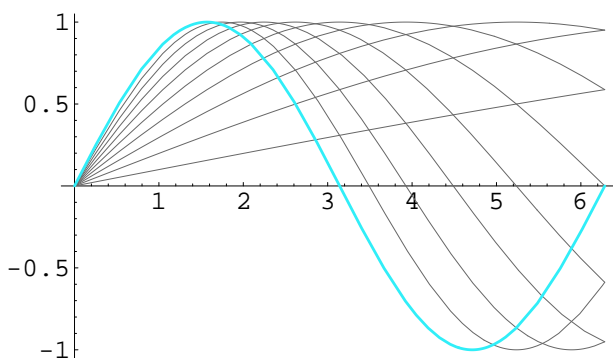
```
In[211]:= Plot[Evaluate[F], {x, 0, 2 π},
  PlotStyle → {{Blue, Thickness[0.01], Dashing[{0.02, 0.02}]}]}];
```



Mit dem **Colorselector** im Menü **Input** kann man beliebige Farben wählen

```
In[212]:= stil1 = {RGBColor[0.187503, 0.929702, 0.968765], Thickness[0.005]};
  stil2 = {GrayLevel[0.4], Thickness[0.001]};
```

```
In[213]:= Plot[Evaluate[F], {x, 0, 2 π},
  PlotStyle → {stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil1}];
```



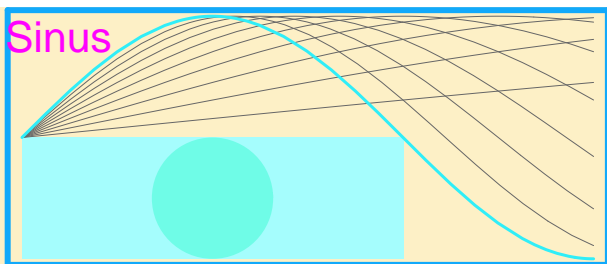
In[214]:= ? Text

Text[expr, coords] is a graphics primitive that represents text corresponding to the printed form of expr, centered at the point specified by coords.

In[215]:= ? FontForm

FontForm[expr, {"font", size}] specifies that expr should be printed in the specified font and size.

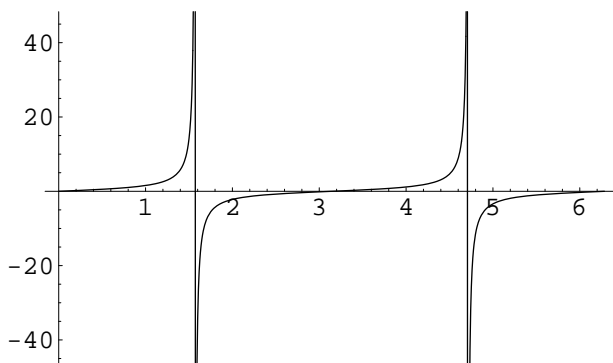
```
In[216]:= Plot[Evaluate[F], {x, 0,  $\frac{3}{2}\pi$ },
  PlotStyle -> {stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil2, stil1},
  AspectRatio -> Automatic, Frame -> True, Axes -> None, FrameTicks -> None,
  FrameStyle -> {RGBColor[0.0742199, 0.664073, 0.980484], Thickness[0.01]},
  Background -> RGBColor[0.992203, 0.937514, 0.777356],
  Prolog -> {Magenta, Text[FontForm["Sinus", {"Arial", 16}], {0.3, 0.8}]},
  {RGBColor[0.644541, 0.992203, 0.992203],
   Polygon[{{0, -1}, { $\pi$ , -1}, { $\pi$ , 0}, {0, 0}]}},
  {RGBColor[0.4336, 0.988296, 0.906264], Disk[ $\{\frac{\pi}{2}, -\frac{1}{2}\}, \frac{1}{2}\]}}$ 
```



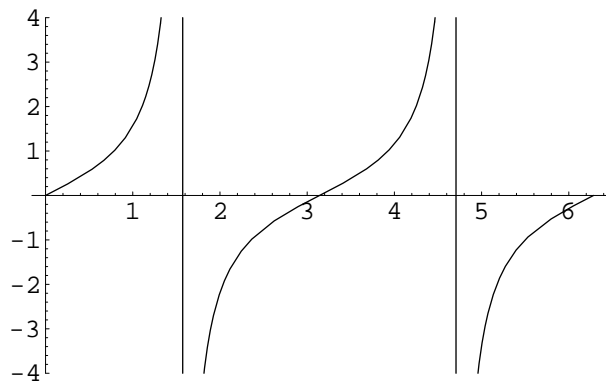
In[217]:= Clear[f, F]

Eine wichtige Option ist **PlotRange**. Beispiel: Der Tangens wird wegen seiner Pole nicht gut dargestellt. Deshalb ist es wichtig, die zugelassenen y-Werte anzugeben.

In[218]:= Plot[Tan[x], {x, 0, 2 π };

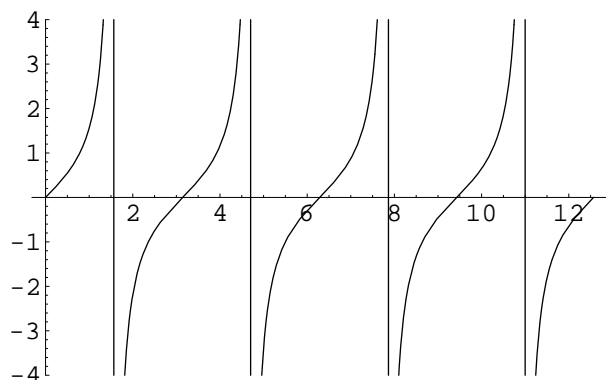


```
In[219]:= Plot[Tan[x], {x, 0, 2 π}, PlotRange → {-4, 4}]
```



```
Out[219]= - Graphics -
```

```
In[220]:= Plot[Tan[x], {x, 0, 4 π}, PlotRange → {-4, 4}];
```



■ Programmieren

■ Module

Mathematica ist eine vollwertige Programmiersprache. Hier wollen wir nur ein einfaches Beispiel betrachten. Ein Programm zur Berechnung des größten gemeinsamen Teilers zweier Zahlen a , b . Dazu verwendet man bekanntlich den euklidischen Algorithmus. Es ist sinnvoll, die im Verlauf des Programms benötigten Variablen als lokale Variable zu deklarieren. Das geht mit dem *Mathematica*-Befehl

```
Module[{lokale Variablen}, Programm]
```

Im folgenden Programm werden lokale Variable d , r deklariert. Am Anfang wird $d = a$, $r = b$ gesetzt. Solange $r \neq 0$ gilt, wird nun der Rest **Mod**[d , r] beim Teilen von d durch r berechnet und $d = r$, $r = \text{Mod}[d, r]$ gesetzt. Dazu benutzt man den *Mathematica*-Befehl

```
While[Bedingung, Anweisung].
```

Am Ende wird das Ergebnis d ausgegeben. Man beachte, daß die einzelnen Anweisungen nach der Variablendeklaration durch ein Semikolon (nicht durch ein Komma) zu trennen sind.

```
In[221]:= ggt[a_, b_] := Module[{d, r}, {d, r} = {a, b}; While[r ≠ 0, {d, r} = {r, Mod[d, r]}]; d]
```

```
In[222]:= ggt[778, 660]
```

```
Out[222]= 2
```

```
In[223]:= Remove[ggt]
```

■ Ein kleines Paket

Will man zu den vorhandenen *Mathematica*-Funktionen neue eigene Funktionen hinzufügen, so bietet es sich an, diese als *Mathematica*-Package zusammenzufassen. Hier unser obiges Beispiel als Package:

```
In[224]:= BeginPackage["GGT`"]
          ggT::usage = "ggT[a,b] ergibt den
                      größten gemeinsamen Teiler von a und b. Beispiel: ggT[6,4]=2"
          Begin["`Private`"]
          ggT[a_, b_] := Module[{d, r}, {d, r} = {a, b}; While[r ≠ 0, {d, r} = {r, Mod[d, r]}]; d]
          End[]
          EndPackage[]
```

```
Out[224]= GGT`
```

```
Out[225]= ggT[a,b] ergibt den größten gemeinsamen Teiler von a und b. Beispiel: ggT[6,4]=2
```

```
Out[226]= GGT`Private`
```

```
Out[228]= GGT`Private`
```

```
In[230]:= ?ggT
```

```
          ggT[a,b] ergibt den größten gemeinsamen Teiler von a und b. Beispiel: ggT[6,4]=2
```

```
In[231]:= ggT[12, 345]
```

```
Out[231]= 3
```