

Kapitel I

Grundlagen der Arbeit mit Mathematica

In diesem Kapitel wollen wir grundlegende Fragen zur Benutzung von Mathematica klären. Dazu wollen wir in Abschnitt 2 ein wenig in Mathematica „hineinschnuppern“ und erste Erfahrungen sammeln, um dann in Abschnitt 3 die Programmstruktur und den Dateityp des Notebooks (so heißen die Mathematica-Dateien) etwas detaillierter zu erkunden. Datenstrukturen sollen dann in Abschnitt 4 behandelt werden, und Abschnitt 5 beschäftigt sich mit elementaren mathematischen Aspekten, wie etwa dem Funktionsbegriff. In Abschnitt 6 werden wir uns schließlich mit den grafischen Fähigkeiten von Mathematica auseinandersetzen.

2 Erste Schritte in Mathematica

Gleich beim ersten Kontakt mit Mathematica werden Nutzer ab Version 10 einige Änderungen gegenüber früheren Versionen bemerken. Zunächst einmal wird man von einem Startbildschirm wie in Abbildung 1 begrüßt, den man durch Entfernen des Häkchens links unten aber auch unterbinden kann. Entscheidet man sich dann z. B. dafür, ein neues Dokument zu erzeugen, dann erscheint die übliche Arbeitsoberfläche aus Abbildung 2. Wobei es auch hier eine Neuerung gibt: Gegenüber früheren Versionen ist die Hauptmenüleiste, über die sich das Programm steuern lässt, nun nicht mehr „freischwebend“, sondern direkt in das Fenster des (noch leeren) *Notebooks* integriert. Ein Notebook ist hierbei der wichtigste Dateityp in Mathematica, der nach Abspeichern die Endung `.nb` erhält; mit diesem und weiteren Dateitypen werden wir uns später insbesondere in Abschnitt 3.4 beschäftigen. Alles Weitere erscheint aber erst einmal vertraut, insbesondere, dass Mathematica auch weiterhin keine geschlossene Oberfläche aufweist, sondern sich aus vielen Teilfenstern zusammensetzt. Auf der rechten Seite etwa ist die Standard-*Eingabepalette* (*Basic Input*) zu sehen, die man über das Menü *Palettes*, Eintrag *Other* → *Basic Math Input* aktiviert (auch für zukünftige Sitzungen, bis man die Palette wieder wegklickt). Diese Eingabepalette erlaubt es, durch simples Anklicken mit der linken Maustaste mathematische Objekte, wie etwa leere Brüche oder Wurzeln, anzulegen, die dann entsprechend zu füllen sind. Eine weitere Neuerung mit Version 10 wird sichtbar, wenn das letzte Notebook geschlossen wurde: Durch diese Aktion wird Mathematica nämlich nicht beendet, sondern es erscheint der Abschlussbildschirm aus Abbildung 3. Hier könnte man nun neue Notebooks öffnen, oder nach Klick auf *Quit* das Programm Mathematica jetzt tatsächlich beenden. Wenn dieser zusätzliche Klick stört, so kann man den Abschlussbildschirm durch Setzen eines Häkchens bei *Always exit . . .* dauerhaft unterdrücken.



Abbildung 1: Der Startbildschirm von Mathematica, Version 10.

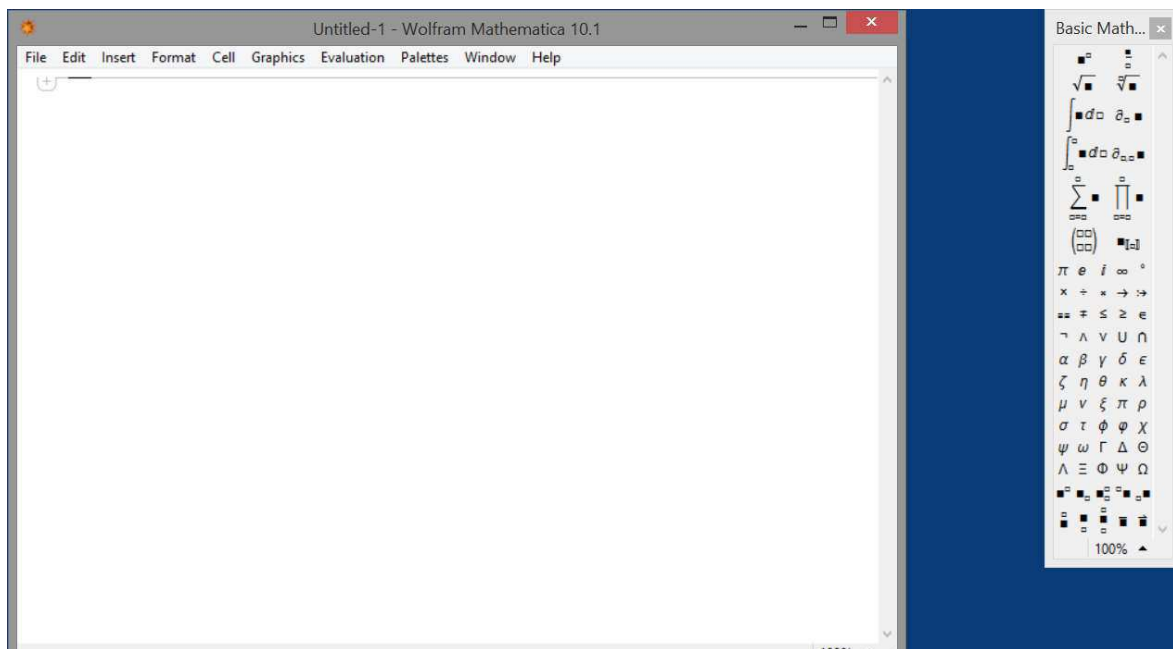


Abbildung 2: Die Arbeitsoberfläche von Mathematica, Version 10.

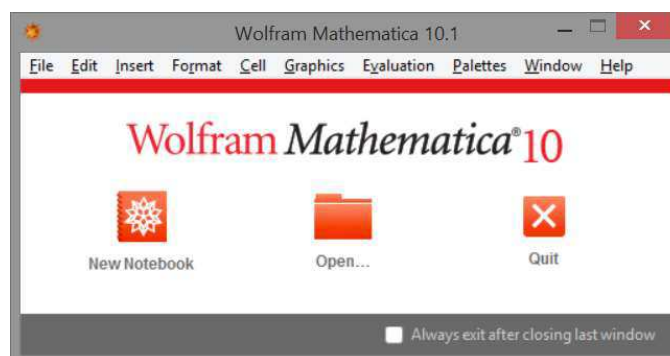


Abbildung 3: Der Abschlussbildschirm von Mathematica, Version 10.

Versuchen wir als Allererstes die Rechnung $3^2 + 1$ einzugeben, diese sollte den Wert 10 ergeben. Dazu klicken wir auf das Potenzsymbol in der Eingabepalette links oben, was zur Folge hat, dass im bis dato leeren Notebook eine *Zelle* angelegt wird, symbolisiert durch die Klammer auf der ganz rechten Seite, in der sich ein Platzhalter für Basis und Exponent befindet. Zu Beginn sollte dabei die Basis markiert sein, wie dies in der folgenden Bilderfolge ganz links zu sehen ist:

\square^{\square}] 3^{\square}] 3^{\square}] $3^{2\blacksquare}$] $3^{2\blacksquare}$] $3^{2\blacksquare} + 1$]

Nun durchlaufen wir Schritt für Schritt diese Folge. Wir tippen die Zahl „3“ ein und markieren anschließend das Feld für die Potenz. Dies erreichen wir, indem wir dieses Feld entweder mit der Maus anklicken oder die Tabulatortaste betätigen. Solange sich ein Objekt mit leeren Platzhaltern in der Zelle befindet, kann man generell mit dieser Taste von Feld zu Feld springen. Wir befinden uns jetzt also beim dritten Bild der Bilderfolge. Nun tippen wir die Zahl „2“ ein, und, da damit die Potenz abgeschlossen ist, bewegen wir den Cursor mit der Pfeiltaste \rightarrow nach rechts, so dass wir wieder auf der Ebene der „3“ sind (vorletztes Bild). Nun können wir noch „+1“ eintippen, dann ist die gewünschte Rechnung fertig eingegeben.

Um diese Rechnung nun auszuführen, betätigen wir die Tastenkombination $\square + \leftarrow$, d. h. wir halten die Umschalt-/Shifttaste \square gedrückt und drücken zusätzlich auf die Eingabe-/Enter-taste \leftarrow . Mit dieser Tastenkombination löst man stets die Berechnung einer *Input-Zelle* aus, dafür muss sich der Cursor irgendwo in dieser Zelle befinden bzw. die Zellklammer muss markiert sein. Nun passiert zweierlei: Da es sich um die allererste Rechnung seit Programmstart handelt, vergeht eine Gedenksekunde, in welcher der sog. *Kernel* startet. Der Kernel ist ein von der Benutzeroberfläche getrennt laufendes Programm, welches einzig und allein für die Berechnungen zuständig ist, und für den Benutzer unsichtbar bleibt. Wie wir später in Abschnitt 3.5 besprechen werden, hat diese strikte Trennung einige praktische Vorteile.

Das zweite Ereignis, welches eintritt, findet im Notebook statt. Dort wird die von uns erstellte Zelle mit der Beschriftung `In[1]` versehen, da es eine Input-Zelle ist. Ferner wird eine neue Zelle angelegt, in welcher das Resultat „10“ ausgegeben wird, die *Output-Zelle* `Out[1]`. Da beide Zellen zusammengehören, wird um beide herum eine weitere Zellklammer angelegt, so dass diese beiden Elementarzellen zu einer übergeordneten Zelle zusammengeschlossen sind. Es entsteht eine Zellhierarchie. Im Notebook ist nun Folgendes zu lesen:

```
In[1]:= 32+1
Out[1]= 10
```

Wir erkennen, dass nun im Notebook, über die ganze Fensterbreite hinweg, eine waagrechte Linie gezogen ist, unterhalb der Ausgabe. Diese Linie zeigt uns die Position des Cursors an, er ist nun unterhalb der Zelle `Out[1]`. Um eine weitere Rechnung einzugeben, tippen wir einfach los: An jener Stelle des Notebooks, wo sich die waagrechte Linie befand, wird eine neue Zelle angelegt. Um eine bestehende Zelle zu manipulieren, müsste man direkt in diese hineinklicken. Nach möglicher Änderung könnte man diese erneut via $\square + \leftarrow$ berechnen.

Tatsächlich werden Nutzer ab Version 9 nach Eingabe von $\square + \leftarrow$ mehr als nur die Zellen `In[1]` und `Out[1]` sehen, bei diesen erscheint im Anschluss zusätzlich noch die *Vorschlagsleiste* („Suggestions Bar“) wie in Abbildung 4, welche u. a. mögliche nächste Rechenschritte anbietet; die Besprechung dieses wie auch weiterer Unterstützungssysteme von Mathematica verschieben wir auf Abschnitt 3.3.

Wir aber wollen unterhalb von `Out[1]` eine neue Rechnung anlegen, etwa $4 \cdot 5$, was 20 ergeben sollte. Also tippen wir `4*5` ein und lösen aus, als Resultat erscheint 20. An Stelle des expliziten

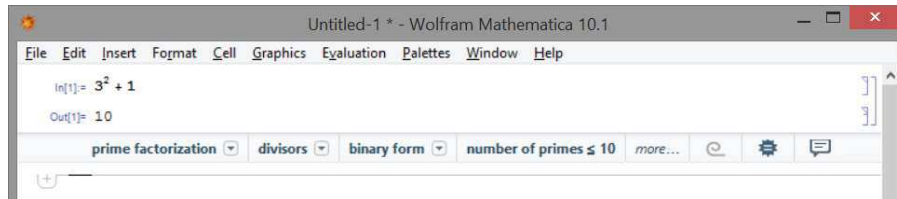


Abbildung 4: Die Vorschlagsleiste von Mathematica, seit Version 9.

Befehl	Beschreibung
$a + b$ bzw. $a - b$	Summe bzw. Differenz von a und b .
$a * b$ bzw. $a b$	Produkt von a und b .
a / b bzw. $\frac{a}{b}$	Quotient $\frac{a}{b}$, Letzteres via Eingabepalette oder Strg + / .
$a ^ b$ bzw. $a ^ b$	Potenz a^b , Letzteres via Eingabepalette oder Strg + ^ .
$a == b$	Prüft Gleichheit von a und b .
$a != b$	Prüft Ungleichheit von a und b .
$a > b$ bzw. $a >= b$	Prüft Ungleichung $a > b$ (größer) bzw. $a \geq b$ (größer gleich).
$a < b$ bzw. $a <= b$	Prüft Ungleichung $a < b$ (kleiner) bzw. $a \leq b$ (kleiner gleich).

Tabelle 2: Elementare Rechen- und Vergleichsoperatoren.

Multiplikationszeichens „*“ hätten wir auch einfach ein Leerzeichen eintippen können, dieses wird als Multiplikation interpretiert². Seit Version 6 ersetzt Mathematica übrigens ein solches Leerzeichen *zwischen Zahlen* durch das Multiplikationszeichen \times . Probieren wir dies aus:

```
In[2]:= 4*5
Out[2]= 20

In[3]:= 4 5
Out[3]= 20
```

Wie wir in Abschnitt 3.5, insbesondere Bemerkung 3.5.1, noch besprechen werden, nummeriert Mathematica die Ein- und Ausgaben fortlaufend durch, allerdings werden diese Nummern nicht mit abgespeichert. Da sie i. Allg. ohne weitere Bedeutung sind und eher der eigenen Orientierung dienen, werden wir sie im Folgenden aus Platzgründen meist weglassen.

Dass beide Rechnungen zum gleichen Resultat führen, hätten wir auch per

```
4 5==4*5           True
```

nachprüfen können, wogegen folgende Ungleichung korrekterweise als falsch erkannt wird:

```
4 5>20             False
```

Man beachte, dass man durch `==` auf Gleichheit prüft, wogegen `=` allein eine Wertzuweisung darstellt. Weitere elementare Rechen- und Vergleichsoperatoren sind in Tabelle 2 zusammengefasst.

²Achtung! Die Eingabe `x y` mit Leerzeichen wird als $x \cdot y$ interpretiert, wogegen `xy` als *ein* Symbol namens „xy“ verstanden wird. Ein vergessenes Leerzeichen ist eine beliebte Fehlerquelle!

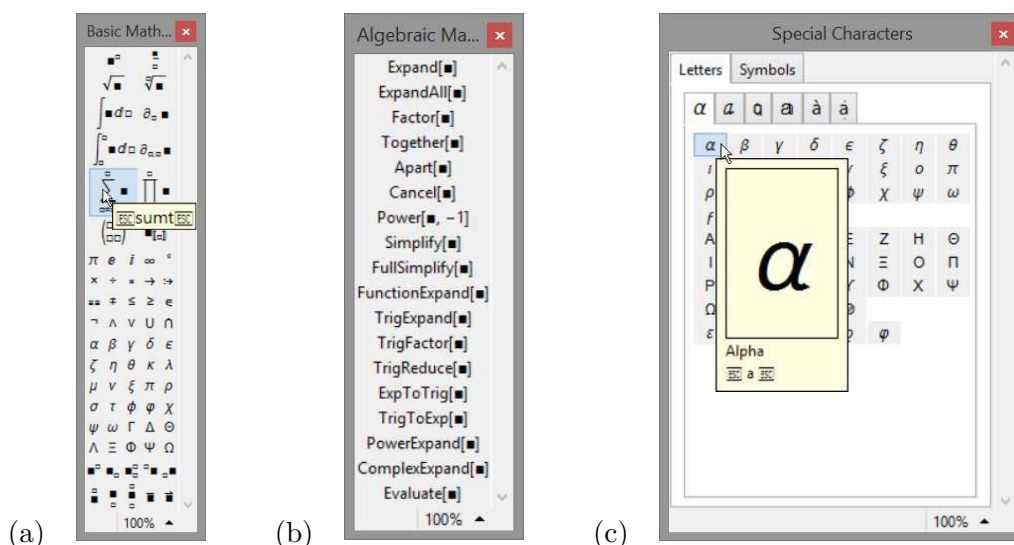


Abbildung 5: Basic Input, Algebraic Manipulation und Special Characters.

Wenden wir uns nochmal der Standard-Eingabepalette zu. Wie in Abbildung 5 (a) zu sehen, ist die Standard-Eingabepalette viergeteilt. Im obersten Teil befinden sich mathematische Ausdrücke wie Brüche, Summen, Integrale, mit deren Hilfe man Formeln grafisch eingeben kann, in der gleichen Form, wie man sie auch per Hand auf Papier notieren würde. Wir werden in späteren Abschnitten, je nach Thema, auf die eine oder andere Schaltfläche und deren Tastenkürzel zu sprechen kommen. Im zweiten Block kann man Konstanten wie die Kreiszahl π (=Pi) oder mathematische Operatoren per Klick auswählen, im dritten Block sind griechische Buchstaben aufgelistet. Diese können, genau wie solche des lateinischen Alphabets, zur Eingabe von Formeln verwendet werden. So macht Mathematica keinen Unterschied zwischen³

$$\mathbf{Expand}[(a+b)^2] \qquad a^2+2 a b+b^2$$

$$\mathbf{Expand}[(\alpha+\beta)^2] \qquad \alpha^2+2 \alpha \beta+\beta^2$$

In beiden Fällen wird die erste binomische Formel korrekt angewendet. Griechische Buchstaben lassen sich auch leicht über die Tastatur eingeben, via $\boxed{\text{Esc}} \text{ Buchstabe } \boxed{\text{Esc}}$ (hintereinander eingeben, nicht gedrückt halten). So ergibt z. B. $\boxed{\text{Esc}} a \boxed{\text{Esc}}$ ein kleines Alpha, α , siehe auch Abbildung 5 (c). Die Schaltflächen des untersten Bereichs der Eingabepalette dienen schließlich der Eingabe von Indizes u. Ä. Der Leser sei an dieser Stelle dazu aufgefordert, ein wenig mit der Palette herumzuspielen.

Ferner gibt es neben der Standardpalette *Basic Input*, je nach Version, eine Reihe weiterer Paletten, die man über das Menü *Palettes* (seit Version 6) aufrufen kann. Darunter befinden sich etwa *Algebraic Manipulation* für die Termumformungen aus Abschnitt 5.1.1 oder *Special Characters* für Sonderzeichen, siehe die Abbildungen 5 (b) und (c). Eine Reihe neuer Paletten wurden mit Version 7 hinzugefügt, insbesondere sogenannte Assistenzpaletten wie in Abbildung 6, welche ein schnelleres Einlernen in Mathematica ermöglichen sollen. Auch beim Thema Paletten sei der Leser zu Experimentierfreude angeregt.

³Nutzer ab Version 9 werden feststellen, dass während der Eingabe von **Expand** eine Auswahlliste erscheint; mehr dazu in Abschnitt 3.3.

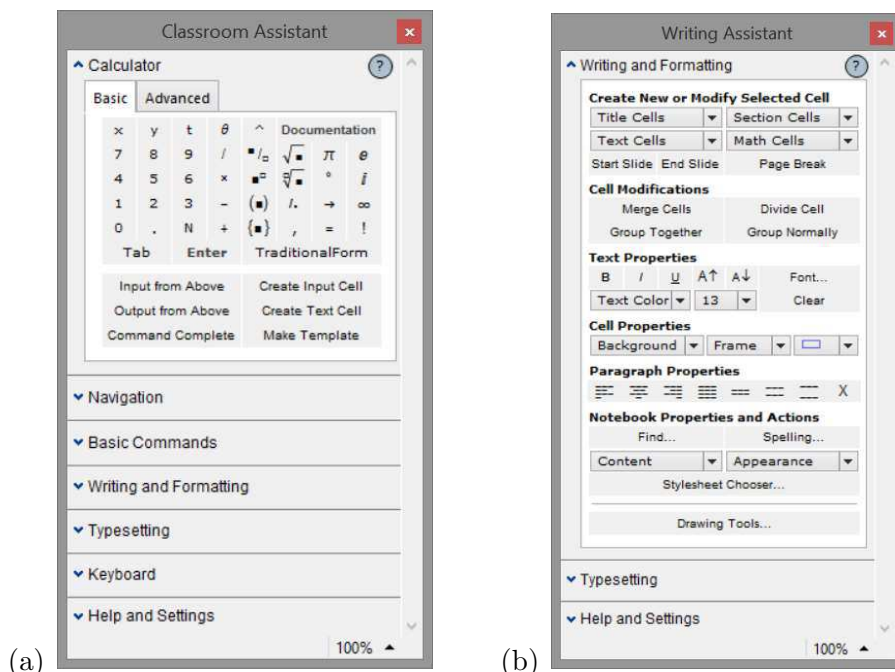
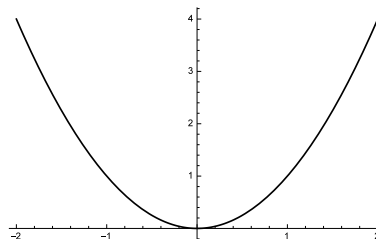


Abbildung 6: Assistenzpaletten: Classroom Assistant und Writing Assistant.

Im zuletzt betrachteten Beispiel haben wir eine Fähigkeit von Mathematica kennengelernt, die es vom gewöhnlichen Taschenrechner unterscheidet, nämlich die zum *symbolischen Rechnen* („Buchstabenrechnen“). Das im Beispiel verwendete Kommando `Expand` dient dem Ausmultiplizieren von Formeln. Wir werden diese Thematik später in Abschnitt 5.1 vertiefen. An dieser Stelle soll ein anderer Punkt angeschnitten werden, nämlich das *Setzen von Klammern*. Im Wesentlichen gibt es drei Arten von Klammern: runde (`()`), eckige [`[]`] und geschweifte [`{}`]. Während man diese Klammern auf dem Papier völlig gleichberechtigt zur Strukturierung von Formeln einsetzen kann, darf man bei Mathematica hierfür *nur runde Klammern* verwenden! Die eckigen Klammern sind, wie oben bei `Expand`, den Mathematica-internen Kommandos vorbehalten, die geschweiften Klammern werden für Listen verwendet, siehe Abschnitt 4.2. Um trotzdem Übersichtlichkeit zu erlauben, passt Mathematica die Größe eingesetzter Klammern den Ausdrücken an, die sie umklammern. Ferner hilft eine Farbkodierung dabei, festzustellen, ob alle geöffneten Klammern auch wirklich geschlossen wurden, und umgekehrt: Eine einzelne Klammer, egal welcher Form, wird immer solange in einer Art Rosa/Violett dargestellt, bis ihr gleichartiges Gegenstück eingegeben wurde (mehr Details zu Mathematica's *Syntaxhervorhebung* folgen in Abschnitt 3.3). Tippen wir nun folgendes Kommando ein:

```
Plot[x2, {x,-2,2}]
```



Hierbei wird der Graph der Normalparabel x^2 im Bereich $x \in [-2; 2]$ gezeichnet. Wie wir später in Abschnitt 6 sehen werden, verfügt Mathematica über ein umfangreiches Repertoire an grafischen Methoden.

Wir haben bereits kennengelernt, dass man Berechnungen durch $\boxed{+}$ + $\boxed{\leftarrow}$ auslöst, dagegen wird durch alleiniges Drücken der Eingabetaste $\boxed{\leftarrow}$ einfach ein *Zeilenumbruch* in die Zeile eingefügt. Ein Zeilenumbruch wird kontextabhängig unterschiedlich interpretiert. Wenn für *Mathematica* klar ist, dass der bis zum Zeilenende eingegebene Ausdruck noch unvollständig ist, wird es den Zeilenumbruch ignorieren, d. h. man kann den Zeilenumbruch in solchen Fällen zum Strukturieren der Eingabe verwenden. Dies ist bei langen Befehlsketten auch durchaus zu empfehlen. Ist der Ausdruck am Zeilenende (gewollt oder ungewollt) dagegen abgeschlossen, so führt *Mathematica* den Ausdruck der Zeile aus, erzeugt auch eine entsprechende Ausgabe, und widmet sich dann der nächsten Zeile. Der Leser gebe mit Zeilenumbrüchen ein:

```
In[9]:=  2+2
         2+3
         2+4

Out[9]=  4

Out[10]= 5

Out[11]= 6
```

Hier wird für jede Zeile eine eigene Ausgabe erzeugt. Häufig ist man jedoch nicht daran interessiert, dass wirklich zu jeder Zeile eine Ausgabe erzeugt wird, etwa dann, wenn es sich nur um Nebenrechnungen handelt. Dann kann man die Eingaben durch ein *Semikolon* „;“ trennen: Für ein Kommando, das mit einem Semikolon beendet wird, wird die Ausgabe unterdrückt.

```
In[12]:= x=2+3;
         2 x

Out[12]= 10
```

Seit Version 6 gilt diese Ausgabeunterdrückung via Semikolon auch für Grafikkommandos. Der Leser probiere dies durch Ausführung von `Plot[x2, {x,-2,2}];` aus.

Zum Schluss, oder besser auch schon während der Arbeit, kann das Notebook abgespeichert werden. Dies erreicht man, wie von anderer Software her gewöhnt, über das Menü *File* → *Save*. Der eingegebene Dateiname wird automatisch mit der Endung `.nb` versehen.