

L^AT_EX für Blöde

Thomas Dettbarn*

17.06.2k

Inhaltsverzeichnis

1 Fangen wir an!	4
1.1 Langweilige Theorie	4
1.2 Mein erstes Dokument	4
1.3 Wie kann ich jetzt was sehen?	5
1.4 Wie kann ich das jetzt ausdrucken?	5
1.5 Und was heißt das jetzt?	6
2 Machen wir mal was richtiges...	8
2.1 Zeilen- und Seitenumbrüche	8
2.2 Titelseiten	8
2.3 Überschriften	9
2.3.1 Ein Inhaltsverzeichnis	10
2.4 Aufzählungen	10
2.4.1 Mit Punkten	10
2.4.2 Mit Zahlen	11
2.4.3 Mit was anderem	11
2.5 Tabellen	12
2.5.1 Und mit Strichen dazwischen	12
2.6 Mathematik	13
3 Und Bilder?	16
3.1 Funktionsgraphen etc.	16
3.1.1 Gerade Linien/Krumme Linien	17
3.1.2 Text dazu	18
3.2 Gemalte Bilder	19
A Die wichtigsten Befehle	22
B Mathematische Formelzeichen	24
C Ein Brief	26

*tdettbar@techfak.uni-bielefeld.de

Hallo Du!

Ja, genau Du. Wie ich sehe liest Du gerade diese kleine Anleitung, daraus schließe ich mal, daß Du \LaTeX lernen willst. Warum auch nicht? Sieht man sich heutzutage mal auf dem Markt für wissenschaftliche Literatur um, so wird man mit einiger Übung feststellen, daß die meisten Bücher in diesem Satzsystem verfaßt worden sind. Auch wenn es heutzutage schon solche „moderneren“ „Textverarbeitungssysteme“ wie Winword oder WordPerfect gibt, so sind diese doch im Vergleich mit \LaTeX ein wenig mühselig zu bedienen. Ihr einziger Vorteil gegenüber \LaTeX ist die WYSIWYG-Funktion, bei der man genau sehen kann, wie der Text im Endeffekt aussehen wird. Versuche doch mal so eine komplexe Formel wie zum Beispiel

$$\int_{i=0}^{2\pi} \frac{1}{2r - x^2} \cdot e^{-\frac{x}{4}} dx$$

zu setzen. Inklusive einiger Tippfehler habe ich dafür eben gerade mal 28 Sekunden benötigt. Die Formel hat zwar keinen Sinn, aber garantiert wirst Du mit Winword ein paar Minuten dahinter sitzen, auch wenn Du schon einige Erfahrung hast. Und wird sie auch so schön aussehen?

Weiterhin hat Winword das Problem, daß es einfach nicht fehlerfrei ist, und daß sich in jedem Dokument, welches man von einem Kumpel bekommt, ein Makrovirus eingebaut sein könnte. \LaTeX hingegen beruht auf \TeX , einem sehr viel älteren Satzsystem, dessen Vater Leslie Lamport ist. Es ist erwiesenermaßen derartig fehlerlos, daß Leslie für jeden Fehler, den man findet bereit ist, einen Haufen Geld zu bezahlen. Stelle Dir mal vor, Steve Ballmer würde das mit Winword zulassen...

Der Grund dafür, warum ich diese Anleitung geschrieben habe ist der, daß ich unzufrieden bin mit der Literatur zu \LaTeX die ich bis jetzt gefunden habe. Und gibt man einmal im Internet den Suchbegriff `latex` ein, so wird man doch wohl eher einige Seiten mit Erotik-Angeboten finden, als das was man wirklich sucht¹.

Dieses Buch ist für wirklich Blöde gedacht, die überhaupt ersteinmal anfangen wollen \LaTeX zu lernen. Für tiefer gehendes Verständnis sei hier auf die im Anhang angegebene Literatur verwiesen. Wenn Du schon welche davon hast, dann lies Dir hier nur die Kapitel 2.2 und 2.3 durch und Du wirst glücklich.

Ansonsten: Viel Spaß!

¹Okay, keine Ahnung wie Du drauf bist...

1 Fangen wir an!

1.1 Langweilige Theorie

\LaTeX ist kein Programm, welches besonders Benutzerfreundlich ist. Sehr viel eher ist es eine Art Programmiersprache, bei der Du den Text mit einem Editor Deiner Wahl schreiben kannst².

Verglichen mit einer Textverarbeitung gibt es hier einen SEHR wichtigen Unterschied: Während man beispielsweise in der Textverarbeitung einen Teil des Textes markieren kann und anschließend mit der Maus auf Fett klickt um eben diesen Teil fett zu drucken, muß man einen Befehl eingeben, der das Fettdrucken einschaltet. Der Bereich, der fettgedruckt werden soll, wird dann geklammert³. Befehle fangen immer mit einem `\`, einem Backslash an, der macht dem \TeX -Kompiler hinterher deutlich, was ein Befehl ist, und was nicht. Zum Beispiel das fettdrucken:

Ich schreibe in den Editor einfach Dies `{\bf hier ist}` ein Text um eine solche Ausgabe zu erhalten: Dies **hier ist** ein Text.

Sehr schön ist hier zu erkennen, daß der fettgedruckte Bereich durch `{` und `}` geklammert ist. Und der Befehl (hier `\bf`) beginnt mit einem Backslash und besteht aus Buchstaben. Das Ende eines Befehles wird dem Kompiler durch das Leerzeichen verraten. Die Sache mit den Befehlen ist zu Anfang ein wenig umständlicher als die intuitivere Mausbedienung, aber hinterher hast Du Deine wichtigsten Befehle im Kopf, und Du wirst merken, daß Du Deine Texte sehr viel schneller schreiben kannst als andere, weil Du nicht ständig hin- und herwechseln mußst.

Einige Befehle erhalten noch einige zusätzliche Parameter, die werden dann auch in geschweifte Klammern geschrieben, also z.B. `\documentclass{article}`.

Übrigends: Die Schreibweise des Befehles ist nicht beliebig. `\bf` ist ein ganz anderer Befehl als z.B. `\Bf`.

1.2 Mein erstes Dokument

Jetzt, da Du theoretischen Grundzüge der Sprache kennst kannst Du ja gleich mal Deinen Texteditor starten und folgenden Text abtippen:

```
\documentclass{article}           % Reicht in den meisten Fällen
\usepackage{german, amssymb}
\usepackage[latin1]{inputenc}
\begin{document}
Hallo Welt.\!
\LaTeX rult!
\end{document}
```

²Ich persönlich benutze vi, aber das ist den meisten Leuten ein wenig zu masochistisch...

³Wer schon einmal HTML programmiert hat hat bestimmt keine Probleme mit dieser Denkweise

Einfach eintippen! Das speicherst Du jetzt unter dem Namen `eins.tex` ab. Dieses kurze Dokument hier ist das, was ich in den meisten anderen L^AT_EX-Anleitungen bisher vermißt habe. Es enthält die grundlegendsten Befehle, die jeder braucht. Bis auf das `Hallo Welt.\` natürlich. Was die einzelnen Befehl im einzelnen bedeuten verrate ich später noch. Ist erstmal nicht wichtig.

1.3 Wie kann ich jetzt was sehen?

Hast Du das eingetippt? Brav! Ich hoffe, Du hast Zugriff auf einen Unix-Rechner. Dann kannst Du jetzt nämlich mal die folgenden Befehle in einer Konsole eintippen⁴:

```
> latex eins.tex
> latex eins.tex
> xdvi eins.dvi
```

Gewöhne Dir am besten gleich an, daß Du zweimal hintereinander den Text mit z.B. `latex eins.tex` durchkompilierst. Das wird später wichtig, wenn wir zu den Inhaltsverzeichnissen kommen.

Hast Du bis hierhin keine Fehler gemacht, dann wird jetzt ein Fenster aufgehen, in dem steht einfach nur

Hallo Welt
L^AT_EXrult.

Und unten in der Mitte eine Seitenzahl. Rechts sind noch so ein paar Buttons zum draufklicken. Das ist richtig. Mehr hast Du ja noch gar nicht geschrieben. Versuche jetzt doch einfach mal mit dem Editor ein wenig mehr Text zwischen das `\begin{document}` und `\end{document}` zu schreiben (speichern nicht vergessen), kompiliere den Text durch und achte darauf was sich ändert.

Du kannst jetzt L^AT_EX.

1.4 Wie kann ich das jetzt ausdrucken?

Willst Du das bißchen Text wirklich schon ausdrucken? Na gut, wenn es sein muß... Dazu mußt Du zunächst einmal aus der `.dvi`-Datei eine `.ps` Datei machen. Das geht, indem Du in der Konsole eintippst:

```
> dvips eins.dvi -f >eins.ps
```

Diese `.ps`-Dateien kannst Du nun einfach mit

```
> lpr eins.ps
```

⁴Natürlich NICHT in dem Editor...

ausdrucken, wenn Du Deinen Drucker richtig eingerichtet hast. Das sollte hoffentlich nicht das Problem sein, welches Du hast. Manchmal muß allerdings auch ein Druckername mit angegeben werden, das geht indem du z.B. eintippst:

```
> lpr -Phrzhp eins.ps
```

Wenn Du das absolut nicht hinkriegst, allerdings unter Windoof auf Deinem Rechner immer drucken konntest, dann kannst Du einfach mit

```
> ps2pdf eins.ps
```

eine .pdf Datei daraus machen, die Du dann dort mit dem Acrobat Reader ausdrucken kannst.

Übrigends: Die Tatsache, daß die Eingabe und die graphische Ausgabe des Dokumentes streng voneinander getrennt sind sorgt dafür, daß es keinen Unterschied macht, ob Du einen kurzen Text, oder einen langen mit ganz vielen Bildern schreibst, weil Du die (rechenintensive) graphische Ausgabe erst dann benutzt, wenn Du sie wirklich benötigst. Und keine hüpfende Büroklammer nervt Dich und stört Deinen Arbeitsfluß.

1.5 Und was heißt das jetzt?

Erinnere Dich noch mal an das Dokument von eben:

```
1. \documentclass{article}
2. % Reicht in den meisten Fällen
3. \usepackage{german, amssymb}
4. \usepackage[latin1]{inputenc}
5. \begin{document}
6. Hallo Welt.\\
7. \LaTeX rult!
8. \end{document}
```

Da drinne stehen einige Befehle, und ein bißchen Text. Die Befehle bedeuten jetzt im Einzelnen:

1. `\documentclass{article}`
Der `\documentclass`-Befehl sagt dem Compiler was für eine Art von Text Du überhaupt schreiben willst. Mögliche Dokumentenarten wären auch noch `letter`, `book`, `report` und vieles andere. `article` reicht Dir allerdings für den Anfang vollkommen. Glaub es mir...
2. `% Reicht in den meisten Fällen`
`LATEX` ist eine Programmiersprache. Und in den meisten Programmiersprachen kann man natürlich in seinem Quellcode Anmerkungen hinschreiben, damit man sich später daran erinnert, was man nun genau gemeint hat. Das Prozentzeichen `%` erzählt dem `LATEX`-Compiler nun, daß dahinter nichts interessantes mehr steht.

3. `\usepackage{german, amssymb}`

Zusätzlich zu den (ganz ganz vielen) Befehlen, die \LaTeX von sich aus als Programmiersprache schon zur Verfügung stellt kann man noch einige Pakete benutzen, die einem noch mehr geben. In dieser Zeile werden jetzt erstmal zwei Pakete benutzt:

- das `german`-Paket, welches dafür sorgt, daß zum Beispiel dafür sorgt, daß das Inhaltsverzeichnis auch so heißt, und nicht mit „Table of contents“ betitelt wird, und
- das `amssymb`-Paket, was Dir noch viel mehr mathematische Formelzeichen gibt, die Du später bestimmt brauchen kannst.

4. `\usepackage[latin1]{inputenc}`

\LaTeX kommt aus Amiland. Und deswegen kann es von sich aus auch nicht solche Buchstaben wie ä, ö, ü, ß, sondern macht irgendeinen Schrott damit, wenn man dieses Paket nicht benutzt.

5. `\begin{document}`

Hiermit gibst Du an, wo der Text von Deinem Dokument anfängt. Alles was davor steht sind einige Voreinstellungssachen, die braucht der \LaTeX -Kompiler.

6. `Hallo Welt.\`

Die erste Textzeile. Ich denke zu dem `Hallo Welt.` muß ich nichts sagen. Interessant sind in dieser Zeile die beiden Backslashes, `\`, damit sagst Du dem \LaTeX -Kompiler nämlich wo er einen Zeilenumbruch hinsetzen soll. Mache mal aus der einen Zeile zwei Stück:

```
Hallo
Welt.\
```

Kompilier Dir das Dokument jetzt noch einmal durch und sieh es Dir an. Kein Unterschied zu vorhin. Wenn Du schon einmal HTML programmiert hast, wirst Du keine Probleme mit dieser Denkweise haben.

7. `\LaTeX rult`

Tut es wirklich! Diese Zeile zeigt Dir einfach nur, daß es auch Befehle gibt, die einfach nur was hinschreiben. `\LaTeX` schreibt einfach nur \LaTeX .

8. `\end{document}`

Du hast schon `\begin{document}` gesagt. Also muß Du auch `\end{document}` sagen. Logisch. Oder nicht? Alles was danach kommt ist für Deinen \LaTeX -Kompiler uninteressant.

So, nun hast Du erstmal genug für heute gelernt, mach mal eine Pause.

2 Machen wir mal was richtiges...

Ich nehme mal an, Du hast die Kapitel 2.2 und 2.3 gelesen und verstanden. Das heißt, Du solltest jetzt \LaTeX -Dokumente schreiben und angucken können.

2.1 Zeilen- und Seitenumbrüche

Irgendwie sieht ein Text ziemlich doof aus, wenn der nur aus einem einzigen Absatz besteht, ich denke da stimmst Du mir zu.

Um eine neue Zeile zu beginnen fügst Du einfach an deren Ende zwei Backslashes `\` ein, das sieht das so aus. Drückst Du nach den Backslashes ZWEIMAL Enter,

dann sieht das so aus. In der Ausgabe hast Du einen korrekten Absatz geschaffen, und die Zeile dahinter ist lustig eingerückt. Statt den Backslashes kannst Du auch `\par`

eingeben, mit diesem Resultat. Die nächste Zeile wird einfach eingerückt, dazwischen ist allerdings keine Zeile frei.

Wenn Du eine ganz neue Seite anfangen willst, dann benötigst Du den Befehl `\newpage`. Den zeige ich Dir hier jetzt aber NICHT.

2.2 Titelseiten

Jedes gute Dokument sollte auch eine Titelseite haben. Damit man auch weiß, was in dem ganzen Stapel Papier drinnesteht, der da auf dem Schreibtisch liegt. Und damit diese Titelseite auch schön aussieht, gibt es bei \LaTeX vier tolle Befehle, die einem dabei helfen: `\title`, `\author`, `\date` und `\maketitle`. Diese Befehle sollte man tunlichst auf der ersten Seite benutzen. In dieser kurzen Anleitung habe ich die so benutzt:

```
...
\begin{document}
\title{\LaTeX für Blöde}
\author{Thomas Dettbarn}
\date{17.06.2k}
\maketitle
...
\end{document}
```

Das sah dann so aus:



Natürlich heißt Du nicht Thomas Dettbarn⁵, Dein Text heißt anderes, und auch der 17.06.2k ist schon ein wenig her. Ich schätze aber mal, daß Du weißt was Du willst.

2.3 Überschriften

Guck mal, genau über dieser Zeile ist eine Überschrift. Und davor steht eine Nummer, die Dir sagt, in welchem Kapitel Du gerade beim Lesen bist.

Überschriften für ein Kapitel gehen mit `\section{Überschrift}`. Der formatiert die dann automatisch durch, gibt denen vernünftige Nummern usw. Und: In das Inhaltsverzeichnis setzt der die auch gleich automatisch ein. Komplett mit Seitenzahl und so.

Zusätzlich für Kapitel gibt es noch Unterkapitel, `\subsection{Überschrift}` und Unter-Unterkapitel, `\subsubsection{Überschrift}`, die werden dann jeweils ein bisschen kleiner geschrieben.

```
\section{Das ist ein Kapitel}
\subsection{Das ist ein Unterkapitel}
\subsubsection{Das ist ein Unter-Unterkapitel}
Das hier ist normaler Text.
Wenn Sie das hier lesen können ist Ihre Brille optimal.
\subsubsection{Das hier auch}
\subsection{Das ist noch ein Unterkapitel}
\section{Das ist noch ein Kapitel}
```

Sieht echt toll aus:

1 Das ist ein Kapitel

1.1 Das ist ein Unterkapitel

1.1.1 Das ist ein Unter-Unterkapitel

Das hier ist normaler Text. Wenn Sie das hier lesen können ist Ihre Brille optimal.

1.1.2 Das hier auch

1.2 Das ist noch ein Unterkapitel

2. Das ist noch ein Kapitel

⁵falls doch, schreibt mir! Ich wollte immer schonmal wissen, ob noch irgendjemand so einen bescheuerten Namen hat wie ich.

2.3.1 Ein Inhaltsverzeichnis

Wie schon erwähnt werden die Kapitelüberschriften automatisch ins Inhaltsverzeichnis aufgenommen. Und selbiges erscheint an der Stelle, wo Du `\tableofcontents` hingeschrieben hast. Ich persönlich packe es immer auf die Titelseite.

```
...
\begin{document}
\title{\LaTeX für Blöde}
\author{Thomas Dettbarn}
\date{17.06.2k}
\maketitle
\tableofcontents
\newpage
...
```

Übrigends: Wenn das zu lang wird, das Inhaltsverzeichnis, dann brauchst Du den Befehl `\setcounter{page}{3}` nach dem `\newpage`.

2.4 Aufzählungen

2.4.1 Mit Punkten

Du willst sowas machen?

- Hierhin schreibe ich gaaaaaanz tollen Text. Der wird richtig gut werden, das weiss ich.
- Hierhin auch.
- Dieser Text ist aber besser.

Kein Problem. Geht so:

```
\begin{itemize}
\item{Hierhin schreibe ich gaaaaaanz tollen Text.\\
Der wird richtig gut werden, das weiss ich.}
\item{Hierhin auch.}
\item{Dieser Text ist aber besser.}
\end{itemize}
```

Zunächst einmal sagst du mit `\begin{itemize}`, daß Du eine Aufzählung machen willst. Die mußt Du irgendwann mit `\end{itemize}` beenden. Und in den Klammern vom `\item`-Befehl steht drinne, was hinter dem Punkt stehen soll. Das kann richtig viel werden. Wie Du siehst auch komplette Absätze.

2.4.2 Mit Zahlen

Wie, Du magst keine Punkte? Du willst Zahlen? Also so was:

- | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none">1. Blablابلabla. Mir ist jetzt einfach mal nichts eingefallen.2. Wer das liest ist doof.3. Dettus was here... |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Auch ganz einfach. Tippe noch mal das Beispiel von eben ab, und schreibe statt `\begin{itemize}... \end{itemize}` einfach `\begin{enumerate}... \end{enumerate}`. Laß die Zeilen mit dem `\item` aber so wie sie sind. Okay, ist dann natürlich ein anderer Text dahinter, aber ich denke, der Effekt ist wohl klar.

2.4.3 Mit was anderem

Wenn Du vorhast so eine Art Lexikon zu schreiben, also sowas zu machen:

- | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Computer Ein furchtbar teures Gerät, was nie macht, was es sollte.</p> <p>Linux Ein Betriebssystem, was nichts kostet.</p> <p>Windows Ein teures Betriebssystem, bei dem es Computerviren gibt.</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Dann mußt Du sowas eintippen:

```
\begin{description}
\item[Computer]{Ein furchtbar teures Gerät, was nie macht, was es sollte.}
\item[Linux]{Ein Betriebssystem, was nichts kostet.}
\item[Windows]{Ein teures Betriebssystem, bei dem es Computerviren gibt.}
\end{description}
```

Statt `\begin{itemize}... \end{itemize}` nimmst Du `\begin{description}` bzw. `\end{description}`. Und zu den `\item` Befehlen schreibst Du in die eckigen Klammern das, was fett gedruckt werden soll.

2.5 Tabellen

Du willst Tabellen, also so was wie das hier:

Artikel	Anzahl	Preis
Klavier	3	3,50 DM
Toaster	5	17,30 DM
Ferrari	2	1000,00 DM

Na ja, die Preise sind Wunschdenken. Ist ja auch egal. Die Tabelle hat auf jeden Fall diesen Befehlssatz:

```
\begin{tabular}{lcr}

Artikel&Anzahl&Preis \\
Klavier&3&3,50 DM\\
Toaster&5&17,30 DM\\
Ferrari&2&1000,00 DM\\

\end{tabular}
```

Du siehst dort oben ein `\begin{tabular}` und unten ein `\end{tabular}`. Hinter `\begin{tabular}` stehen drei Buchstaben in den Klammern, nämlich `{lcr}`. Wenn Du genau hinguckst wirst Du merken, daß die Tabelle drei Spalten hat. Bei der ersten fangen alle Zeichen auf der linken Seite an, bei der letzten alle auf der rechten.

Merkst Du was?

Die drei Buchstaben stehen dafür, wie der Text in den Spalten ausgerichtet sein soll. Also `l`=linksbündig, `r`=rechtsbündig und `c`=zentriert(center). Probiere mal `\begin{tabular}{llr}` oder `\begin{tabular}{ccc}` aus und sieh Dir an, was sich ändert.

Mehr Buchstaben in den Klammern bedeuten mehr Spalten. Also läßt demnach `\begin{tabular}{rclrclrrlc}` eine 11-spaltige Tabelle beginnen.

Weiterhin siehst Du den Text aus der Tabelle auch in dem Befehlssatz darunter. Mit dem `&`-Zeichen weiß \LaTeX daß eine neue Spalte anfängt. Und mit den zwei Backslashes am Schluß `\\` daß eine Zeile endet. Am thinkink is very easy, da?

2.5.1 Und mit Strichen dazwischen

Irgendwo sieht die Tabelle doch doof aus, oder nicht? Woher soll ein Leser denn jetzt wissen, wo die Überschrift aufhört und die interessanten Informationen anfangen? Besser wäre so eine Tabelle:

Artikel	Anzahl	Preis
Klavier	3	3,50 DM
Toaster	5	17,30 DM
Ferrari	2	1000,00 DM

Auch gar kein Problem. Schreibe hinter die beiden Backslashes `\` einfach noch zusätzlich den Befehl `\hline`. Dann steht in Deinem Code hinterher sowas hier: `\\\hline`, drei Backslashes hintereinander. Stimmt schon so. Sieht ein wenig seltsam aus. Ist aber egal.

Und wenn Du lieber solche Striche hast

Artikel	Anzahl	Preis
Klavier	3	3,50 DM
Toaster	5	17,30 DM
Ferrari	2	1000,00 DM

Dann liegt der Trick darin, daß Du zu den drei Buchstaben bei `\begin{tabular}{lcr}` noch solche `|` Pipe-Zeichen eingeben musst. Das ist das Zeichen, was fast so aussieht wie ein Doppelpunkt der geschmolzen ist. Liegt meistens zusammen mit `<` `>` auf einer Taste.

Also für die Tabelle oben sieht die erste Zeile so aus: `\begin{tabular}{|l|c|r|}`. Jeweils da, wo ein Strich in der Tabelle hin soll malst Du so ein `|` dazu.

Kann man natürlich auch kombinieren:

Artikel	Anzahl	Preis
Klavier	3	3,50 DM
Toaster	5	17,30 DM
Ferrari	2	1000,00 DM

Der komplette Befehlssatz dafür sieht so aus:

```
\begin{tabular}{|l|c|r|}\hline
```

```
Artikel&Anzahl&Preis \\\hline
```

```
Klavier&3&3,50 DM\\\hline
```

```
Toaster&5&17,30 DM\\\hline
```

```
Ferrari&2&1000,00 DM\\\hline
```

```
\end{tabular}
```

2.6 Mathematik

In der Einleitung habe ich ja schon eine Formel hingeschrieben:

$$\int_{i=0}^{2\pi} \frac{1}{2r - x^2} \cdot e^{-\frac{x}{4}} dx$$

Und hier erkläre ich Dir nun, wie so was funktioniert.

Prinzipiell gibt es zwei Möglichkeiten, um in einem Text eine Formel unterzubringen, einmal so, daß er im laufenden Text steht, also etwa so:

...Und damit wissen wir, daß $a_2 = \frac{1}{2} \cdot e^{17}$ ist...

Oder explizit als z.B. eine längere Umformung:

$$\begin{aligned}
 a^2 - a^2 &= a^2 - a^2 \\
 (a - a) \cdot (a + a) &= a \cdot (a - a) \\
 a + a &= a \\
 2a &= a \\
 2 &= 1
 \end{aligned}$$

(Wo liegt der Fehler?)

Machen wir ersteinmal die erste Art und Weise. Die ist die einfachste. Schreibe einfach mal in Deinen Quellcode die Zeile `$a_2 = \frac{1}{2} \cdot e^{17}$` . Mit dem `$` Dollarzeichen sagst Du wo Deine Formel anfängt, und wo sie aufhört. Ist wie eine Klammer, nur anders.

`a_2` produziert ein a_2 . Das was in den Klammern steht wird zum Index.

`\frac{1}{2}` produziert ein $\frac{1}{2}$ Jetzt kannst Du Brüche.

`\cdot` produziert ein \cdot Ein Multiplikationszeichen.

`e^{17}` produziert ein e^{17} Das was in den Klammern steht wird zum Exponenten

Probier doch jetzt mal selber so eine Formel: $a_{i+1}^2 = 3 + e^{\frac{3}{4}}$. Ein wenig trickreich, weil Du einen Bruch als Exponenten nehmen mußt, aber das schaffst Du schon.

Die zweite Möglichkeit darunter funktioniert ähnlich wie die Tabellen. Der Befehlssatz dafür war der hier:

```

\begin{eqnarray*}
a^2 - a^2 &=& a^2 - a^2 \\
(a - a) \cdot (a + a) &=& a \cdot (a - a) \\
a + a &=& a \\
2a &=& a \\
2 &=& 1
\end{eqnarray*}

```

Das ist im Prinzip eine Tabelle mit automatisch drei Spalten. Nur statt `\begin{tabular}` sagst Du `\begin{eqnarray*}`. Und in die mittlere Spalte habe ich hier einfach nur das Gleichheitszeichen geschrieben. Daß zwei Spalten mit `&` getrennt werden muß ich Dir hoffentlich nicht sagen...

Übrigends: Wenn Du eingibst `\begin{eqnarray}... \end{eqnarray}`, also ohne Sternchen, dann werden die Zeilen durchnummeriert:

$$\begin{array}{rcl}
 a^2 - a^2 & = & a^2 - a^2 \quad (1) \\
 (a - a) \cdot (a + a) & = & a \cdot (a - a) \quad (2) \\
 a + a & = & a \quad (3) \\
 2a & = & a \quad (4) \\
 2 & = & 1 \quad (5)
 \end{array}$$

Wichtig sind jetzt auch noch Summenzeichen. Also z.B. $\sum_{n+1}^i i^2$. Das Summenzeichen hat ja auch eine obere und untere Grenze, fast so, wie jede Variable einen Index und einen Exponenten haben kann.

Der Befehl fürs Summenzeichen ist `\sum`. Ganz einfach. Gibst Du aber jetzt ein `\sum^{i}_{n+1} i^2`, dann sieht das doof aus, nämlich so: $\sum_{n+1}^i i^2$.

VIEL besser ist da diese Befehlszeile: `\sum\limits^{i}_{n+1} i^2`. Hinter dem Befehl für `\sum` kommt noch zusätzlich der Befehl `\limits`. Der arbeitet auch zusammen mit dem Befehl `\int` für das Integralzeichen, `\lim` für den Limes und noch ein paar anderen.

Probiere doch jetzt mal mit Hilfe der `\begin{eqnarray*}... \end{eqnarray*}`-Umgebung und der Befehlsübersicht für mathematische Symbole im Anhang diese Formeln zu `TeX`:

$$\begin{array}{rcl}
 \int_{i=0}^{2\pi} \frac{1}{2r - x^2} \cdot e^{-\frac{x}{4}} dx & = & 7 \\
 \lim_{h \rightarrow \infty} \frac{1}{h} & = & 0 \\
 \prod_{i=j+1}^n \sqrt[3]{i+6} & = & 9
 \end{array}$$

Und noch ein paar eigene. Dann hast Du schon wieder ersteinmal genug gelernt. Mach noch eine Pause. Du weißt schon genug um Deine Mathehausaufgaben mit `LaTeX` zu machen.

3 Und Bilder?

Gehen natürlich auch.

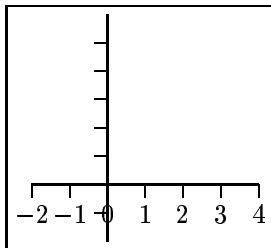
3.1 Funktionsgraphen etc.

Fangen wir zunächst mal mit Funktionsgraphen an. Wenn Du RICHTIG gut bist, dann kannst Du mit den Befehlen, die Du hier gleich lernen wirst natürlich auch Vektorgraphiken selbst programmieren, aber darauf hast Du nicht immer Lust.

Kommen wir jetzt mal zu den wichtigsten Sachen:

Als allererstes mußt Du das Pakete `PiCTEX` benutzen. Das geht, indem Du `\usepackage{pictex}` zu den anderen Paketen schreibst. Ganz oben in der Datei, die Du im Texteditor offen hast.

Ich male Dir jetzt erstmal ein Koordinatensystem:



Und die Befehle dazu sehen dann so aus:

1. `\beginpicture`
2. `\setcoordinatesystem units <5mm,7.5mm>`
3. `\setplotarea x from -2 to 4, y from -1 to 3`
4. `\axis bottom shiftedto y=0 ticks numbered from -2 to 4 by 1 /`
5. `\axis left shiftedto x=0 ticks from -0.5 to 2.5 by 0.5 /`
6. `\endpicture`

Merkst Du was? Die Befehle sehen ein wenig anders aus als gewohnt. Keine Ahnung woran das liegt. Ist ja auch egal. Die Befehle bedeuten im einzelnen:

1. `\beginpicture`
Heißt auf Deutsch „beginne ein Bild“. Noch Fragen? Ja?! Okay. An dieser Stelle wird der `LATEX`-Kompiler angewiesen daß die nun folgenden Befehle ein Bild malen sollen.
2. `\setcoordinatesystem units <5mm,7.5mm>`
Wenn Du mit dem `PiCTEX`-Paket ein Bild malst, dann verbindest Du im Prinzip zwei Punkte in der Ebene miteinander. Und mit dem `5mm` sagst Du wie weit diese Punkte nach links und rechts voneinander entfernt sein

sollen. Die 7.5mm sind die Entsprechung für oben und unten.

Übrigends: Es ist sinnvoll die Bilder zunächst einmal mit großen Abständen zu malen und sie hinterher zu verkleinern. Also erstmal mit so etwas wie `\setcoordinatesystem units <25mm,25mm>` anfangen, dann siehst Du einige Details einfach besser. Wenn Du das Bild dann fertig hast verkleinerst Du es mit `\setcoordinatesystem units <2.5mm,2.5mm>` auf ein zehntel.

Da \LaTeX intern mit Maßeinheiten arbeitet, die in ihrer Genauigkeit der Wellenlänge des Lichtes entsprechen ist es egal.

3. `\setplotarea x from -2 to 4, y from -1 to 3`

Hiermit gibst Du an, von wo bis wo Du Dein Bild malen willst. Von x - und y -Koordinaten hast Du doch bestimmt schon mal was in Mathe gehört, oder nicht? Gut. Hier siehst Du übrigens, daß x von -2 bis 4 reicht. Also insgesamt $6 \cdot 5mm = 30mm$ breit wird also das Bild werden. Meß doch mal nach...

4. `\axis bottom shiftedto y=0 ticks numbered from -2 to 4 by 1 /`

- Das hier malt die x -Achse, die sich normalerweise am Boden (`bottom`) befindet.
- Sie wird soweit angehoben, daß Sie durch $y = 0$ geht. (`shiftedto y=0`)
- Auf Ihr sollen Striche (`ticks`) gleichen Abstands stehen.
- Die Striche sollen durchnummeriert (`numbered`) werden.
- Die Striche sollen von -2 (`from -2`) bis +4 (`to 4`) reichen und im Abstand von 1 (`by 1`) auftauchen.

ACHTUNG: Der / ist wichtig!

5. `\axis left shiftedto x=0 ticks from -0.5 to 2.5 by 0.5 /`

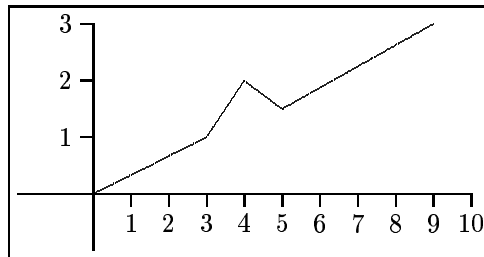
Genauso wie eben, nur daß die y -Achse nicht durchnummeriert wird. (`numbered` fehlt.)

6. `\endpicture`

Und die Befehle, die danach kommen gehören nicht mehr zu dem Bild.

3.1.1 Gerade Linien/Krumme Linien

Super! Nun hast Du ein Koordinatensystem auf dem Bildschirm. Und was macht man mit einem Koordinatensystem? Richtig. Man zeichnet einen Funktionsgraphen hinein. Guck Dir mal das Bild hier an:



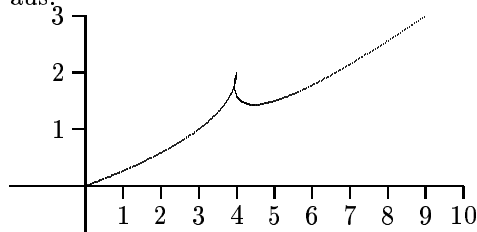
Dieser Funktionsgraph geht durch die Punkte $(0;0)$, $(3;1)$, $(4;2)$, $(5;1,5)$, $(9;3)$. Ja, wirklich. Die Befehle, die Du dafür benutzen mußt sind diese hier:

```
\beginpicture
\setcoordinatesystem units <5mm,7.5mm>
\setplotarea x from -2 to 10, y from -1 to 3
\axis bottom shiftedto y=0 ticks numbered from 1 to 10 by 1 /
\axis left shiftedto x=0 ticks numbered from 1 to 3 by 1 /

\setlinear
\plot 0 0 3 1 4 2 5 1.5 9 3 /
\endpicture
```

Du siehst, da sind nur zwei neue Befehle dazugekommen, die Du noch nicht kanntest. `\setlinear` und `\plot`. Hinter dem `\plot` Befehl stehen die Punkte, durch die der Funktionsgraph verlaufen soll. Und ganz am Ende ein /, ein WICHTIGER Schrägstrich.

`\setlinear` sagt \TeX einfach nur, daß die Punkte gerade verbunden werden sollen. Willst Du einen quadratischen Funktionsgraphen mit „runden“ Verbindungen, ist `\setquadratic` der Befehl, den Du einsetzen mußt. Das sieht so aus:



Bißchen wild, oder? Dann laß den Graphen doch einfach mal durch die Punkte $(-1;1)$, $(0;0)$, $(1;1)$, $(2;4)$, $(3;9)$ laufen und Du hast eine Normalparabel. WICHTIG: Wenn Du `\setquadratic` angegeben hast muß der Graph IMMER durch eine ungerade Anzahl von Punkten laufen!

3.1.2 Text dazu

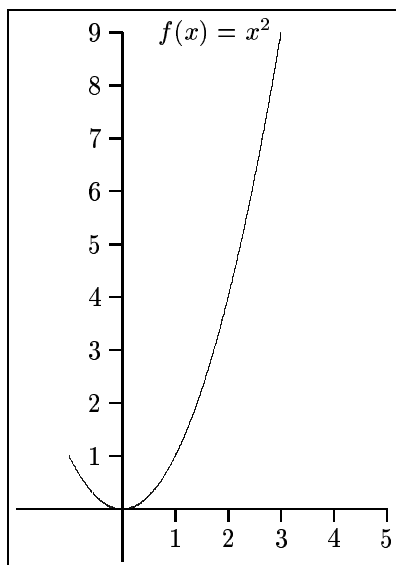
Super, nicht wahr? Und jetzt willst Du Deinen Funktionsgraphen noch Namen geben, also z.B. so was wie $f(x)$, $g(x)$ oder sowas.

Ist auch ganz einfach: Probiere bei der Normalparabel, die Du im Absatz vorher gemalt hast mal den Befehl `\put {\$f(x)=x^2\$}` at 2 9 aus. OHNE den

Schrägstrich am Ende.

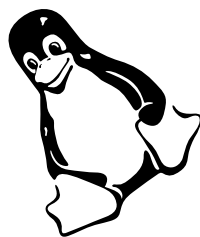
L^AT_EX wird Dir links neben die Parabel nun ein $f(x) = x^2$ schreiben. Genau genommen zentriert er den Text um den Punkt (2;9) herum. Wenn der Text dabei über den Funktionsgraphen rübermalen sollte muß Du einfach die x -Koordinate dieses Punktes verkleinern, also etwa `\put $\{f(x)=x^2\}$ at 1 9` oder `\put $\{f(x)=x^2\}$ at 1.75 9` oder was weiß ich...

Hast Du es korrekt gemacht, so sieht das so aus:



3.2 Gemalte Bilder

Isser nicht süß?



Und wie habe ich das nun gemacht? Nun, zunächst einmal muß mit `\usepackage{epsfig}` im Kopf das `epsfig`-Paket eingebunden werden. Danach kannst Du irgendwo im Text ein paar Zeilen wie diese hier eingeben:

```

1. \begin{center}
2. \resizebox*{0.3\textwidth}{0.2\textheight}{
3.   \rotatebox{35}{
4.     \includegraphics{penguin.eps}
5.   }
6. }
7. \end{center}

```

1. `\begin{center}`
Dieser Befehl sorgt einfach nur dafür, daß der nachfolgende Text in der Mitte des Blattes dargestellt wird.
2. `\resizebox*{0.3\textwidth}{0.2\textheight}{`
Mit diesem Befehl machst Du das Bild größer und kleiner. In diesem Fall wäre es auf 30% der Seitenbreite (`0.3\textwidth`) und 20% (`0.2\textheight`) verkleinert.
Du könntest hier auch explizite Maße wie z.B. `\resizebox*{10cm}{5cm}` angeben. Aber das wäre ja langweilig...
Was Dir vielleicht auffällt ist die Tatsache, daß am Ende dieser Zeile eine Klammer aufgeht. Stimmt! und in Zeile 6. geht sie wieder zu.
3. `\rotatebox{35}{`
Dieser Befehl hier dreht das Bild. In diesem Fall um 35° Grad. Noch Fragen?
4. `\includegraphics{penguin.eps}`
Schlußendlich lädt dieser Befehl das eigentliche Bild. In dieser Klammer muß einfach nur der Dateiname des Bildes stehen. Der VOLLSTÄNDIGE! Mit Endung.
Das einzige Problem hier ist, daß sich das Bild entweder im `.ps` oder `.eps` Format befinden muß. Du kannst also nicht so einfach ein `.jpg` Bild von Deinem Sexualpartner einscannen und auf ein Dokument packen.
Glücklicherweise gibt es unter Linux einige Programme, die aus einem `.jpg` Bild eine `.ps` Datei machen. Zum Beispiel GIMP.
5. `}`
Klammer zu. Von Zeile 3.
6. `}`
Klammer zu. Von Zeile 2.
7. `\end{center}`
Ab dieser Zeile wird der nachfolgende Text nicht mehr zentriert geschrieben. Das sollte ja auch nur für das Bild gelten.

Prima! Nun kannst Du schon die wichtigsten Sachen mit L^AT_EX machen. Ich bin fertig.

Ich denke, daß Du, wenn Du das hier wirklich gelesen hast nun den Großteil

Deiner Texte schreiben kannst. Und das ohne, daß Du Angst vor Computerviren, Programmierfehlern oder hüpfenden Büroklammern haben mußt, die Dir Deinen ohnehin schon langsamen Rechner noch langsamer machen.
Ansonsten hoffe ich, daß Du zumindest ein bißchen Spaß gehabt hast. Na komm... SOOOOO scheiße waren meine Witze doch auch nicht.
Auf den nächsten paar Seiten gebe ich Dir noch eine kurze Zusammenfassung der wichtigsten Befehle. Am Besten ist es WIRKLICH, wenn Du dieses Dokument einmal ausdrückst, nachdem Du mit
> psnup -n 2 latex_fuer_bloede.ps tempo.ps
zwei Seiten auf eine gedruckt hast. Dann kannst Du nämlich die Seiten richtig schön vor Deine Nase hängen.

Tschüß!

A Die wichtigsten Befehle

Hier noch einmal für Dich die Kurzzusammenfassung der wichtigsten Befehle, die Du garantiert irgendwann brauchen kannst.

Kompilieren/Ausdrucken

Kompilieren:

```
> latex Datei.tex
```

```
> latex Datei.tex
```

```
> xdvi Datei.dvi
```

Ausdrucken:

```
> dvips Datei.dvi -f >Datei.ps
```

```
> psnup -n 2 Datei.ps tempo.ps (Spart Papier...)
```

```
> lpr tempo.ps oder >lpr Datei.ps
```

Für jedes Dokument

```
\documentclass{article}
\usepackage{german, amssymb}
\usepackage[latin1]{inputenc}
\begin{document}
Text... Bilder...
\end{document}
```

Cooler Titelseite...

```
\documentclass{article}
\usepackage{german, amssymb}
\usepackage[latin1]{inputenc}
\begin{document}
\title{\LaTeX $ $ für Blöde}
\author{Thomas Dettbarn}
\date{17.06.2k}
\maketitle
\tableofcontents
\newpage
Text... Bilder...\\
\end{document}
```

Überschriften

```
\section{Kapitel}
\subsection{Unterkapitel}
\subsubsection{Unterunterkapitel}
```

Tabellen

```
\begin{tabular}{rcl}
Zeile1Spalte1&Zeile1Spalte2&Zeile1Spalte3\\
Zeile2Spalte1&Zeile2Spalte2&Zeile2Spalte3\\
Zeile3Spalte1&Zeile3Spalte2&Zeile3Spalte3\\
\end{tabular}
```

Grafiken

```
...
\usepackage{pictex}
...
\beginpicture
\setcoordinatesystem units <5mm,5mm>
\setplotarea x from 0 to 10, y from 0 to 10
\axis bottom shiftedto x=0 ticks numbered from 0 to 10 by 0.1 /
\axis bottom shiftedto y=0 ticks numbered from 0 to 10 by 0.2 /
\setquadratic
\setlinear
\plot 0 0 0 1 1.5 2 2 1 0 0 2 0 2 1 0 1 2 0 /
\put {Das ist das Haus vom Nikolaus} at 7.5 1
\endpicture

\begin{center}
\resizebox*{0.3\textwidth}{0.2\textheight}{
  \rotatebox{35}{
    \includegraphics{penguin.eps}
  }
}
\end{center}
```

Mathe

```
\begin{eqnarray*}
a_{i-1}^2&=&5
\end{eqnarray*}
```

Hochstellen:	a^2	macht a^2 .
Tiefstellen:	a_2	macht a_2 .
Brüche:	$\frac{1}{a^2}$	macht $\frac{1}{a^2}$
Summen:	$\sum\limits_{i=1}^n$	macht $\sum_{i=1}^n$
Wurzeln:	$\sqrt[n]{27}$	macht $\sqrt[n]{27}$.

B Mathematische Formelzeichen

Buchstaben

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>
o	<code>o</code>	π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>
ω	<code>\omega</code>						

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Mengensymbole

Man kann den Befehl `\mathbb{A}` benutzen, was folgende Ausgabe liefert: \mathbb{A} .
 Für andere Buchstaben gilt dies analog: $\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$.
 Oder man gibt ein: `I\hspace{-0.39em}N`, was so aussieht: \mathbb{N} .

Binäre Operationssymbole

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\bullet	<code>\bullet</code>	\square	<code>\square</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	\diamond	<code>\diamond</code>	\diamond	<code>\diamond</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\triangle	<code>\triangle</code>
\cdot	<code>\cdot</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	∇	<code>\nabla</code>
$*$	<code>\ast</code>	\vee	<code>\vee</code>	\triangleleft	<code>\triangleleft</code>	\triangleleft	<code>\triangleleft</code>
\star	<code>\star</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\triangleright</code>	\triangleright	<code>\triangleright</code>
\dagger	<code>\dagger</code>	\setminus	<code>\setminus</code>	\oslash	<code>\oslash</code>	\oplus	<code>\oplus</code>
\ddagger	<code>\ddagger</code>	\wr	<code>\wr</code>	\odot	<code>\odot</code>	\ominus	<code>\ominus</code>
\amalg	<code>\amalg</code>	\parallel	<code>\parallel</code>	\mid	<code>\mid</code>	\otimes	<code>\otimes</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\neq	<code>\neq</code>	\sim	<code>\sim</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\asymp	<code>\asymp</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\smile	<code>\smile</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\equiv	<code>\equiv</code>	\frown	<code>\frown</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\propto	<code>\propto</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\prec	<code>\prec</code>	\succ	<code>\succ</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>
\models	<code>\models</code>	\perp	<code>\perp</code>	\notin	<code>\notin</code>		

Pfeile

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto	<code>\leadsto</code>		

Andere mathematische Symbole

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	\square	<code>\Box</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\diamond	<code>\Diamond</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\triangle	<code>\triangle</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\clubsuit	<code>\clubsuit</code>
ℓ	<code>\ell</code>	∂	<code>\partial</code>	\natural	<code>\natural</code>	\diamondsuit	<code>\diamondsuit</code>
\wp	<code>\wp</code>	\top	<code>\top</code>	\sharp	<code>\sharp</code>	\heartsuit	<code>\heartsuit</code>
\Re	<code>\Re</code>	\perp	<code>\perp</code>	\parallel	<code>\parallel</code>	\spadesuit	<code>\spadesuit</code>
\Im	<code>\Im</code>	\vdash	<code>\vdash</code>	\angle	<code>\angle</code>	\Join	<code>\Join</code>
Υ	<code>\Upsilon</code>	\dashv	<code>\dashv</code>	\backslash	<code>\backslash</code>	∞	<code>\infty</code>

\sum	<code>\sum</code>	\bigcap	<code>\bigcap</code>	\odot	<code>\bigodot</code>
\int	<code>\int</code>	\bigcup	<code>\bigcup</code>	\otimes	<code>\bigotimes</code>
\oint	<code>\oint</code>	\bigsqcup	<code>\bigsqcup</code>	\oplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigvee	<code>\bigvee</code>	\uplus	<code>\biguplus</code>
\coprod	<code>\coprod</code>	\bigwedge	<code>\bigwedge</code>	\lim	<code>\lim</code>
$\frac{1}{2}$	<code>\frac{1}{2}</code>	$\sqrt[27]{}$	<code>\sqrt[n]{27}</code>	a_i^n	<code>a^*_i</code>

Noch ein paar Beispiele

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \left(\begin{array}{l} a \\ b \\ c \\ d \end{array} \right)$$

$$f(x,y) = \begin{cases} 1 & : x = y \\ 0 & : \text{sonst} \end{cases}$$

$$f(x,y) = \left\{ \begin{array}{l} 1 \\ 0 \end{array} \right. \text{ : } x=y \text{ \& } 0 \text{ \& } \text{sonst} \end{array} \right.$$

C Ein Brief

```
\documentclass[a4paper]{letter}
\usepackage{german}
\renewcommand*{\ccname}{}           %% Verteiler
\renewcommand*{\enclname}{}        %% Anlagen
\renewcommand*{\headtoname}{Bill Gates\\
                                   Microsoftcampus 666\\
                                   90210 Redmont}           %% An
\renewcommand*{\pagename}{}        %% Seite
%% Absender f"ur alle folgenden letter-Umgebungen
\name{}
\address{Dagobert Duck\\Geldspeicher 1\\ 90210 Entenhausen}
\date{3. Mai 2k}
\pagestyle{headings}               %% Ab Seite 2 mit An, Datum Seite im Kopf
\begin{document}
\begin{letter}{\headtoname}
\opening{Sehr geehrte Damen und Herren,}

In a world without Walls and Fences... Who needs Windows and Gates?
\closing{Mit freundlichen Grüßen}
\end{letter}
\end{document}
```

Folgende Bücher haben mir zum Thema \LaTeX besonders gut gefallen:

Literatur

- [1] Helmut Kopka.
LaTeX Band 1 – Einführung.
Addison-Wesley Co., 2. Auflage
ISBN: 3-8273-1025-3

- [2] Helmut Kopka.
LaTeX Band 2 – Ergänzungen- Mit einer Einführung in META-FONT.
Addison-Wesley Co., 2. Auflage
ISBN: 3-8273-1229-9

- [3] Matthias Kalle Dalheimer.
LaTeX kurz & gut
O'REILLY
ISBN: 3-89721-204-8

- [4] Richard Petersen.
Linux Programmer's Reference
Osborne
ISBN: 0-07-882587-3

- [5] ILLIAD.
USER FRIENDLY the comic strip
O'REILLY
ISBN: 1-56592-673-0

- [6] ILLIAD.
EVIL GENIUSES in a Nutshell
O'REILLY
ISBN: 1-565922-861-X