

Einführung in L^AT_EX

Jannik Arndt

5. Dezember 2014

Inhaltsverzeichnis

1	Einführung	2
1.1	Geschichte	2
1.2	Prinzip	2
2	Programme	2
2.1	Windows	2
2.2	Mac OS X	2
2.3	Unix	2
3	Grundstrukturen eines Dokuments	3
3.1	documentclass	3
3.2	usepackage	3
3.3	title	4
3.4	Sonstige	4
4	Unterteilung, Aufzählungen, etc.	5
4.1	Kapitel	5
4.2	Normale Aufzählung: itemize	5
4.3	Nummerierte Aufzählung: enumerate	5
4.4	Beschreibende Aufzählung: description	6
4.5	Typographie	6
4.6	Schriftfarbe	6
5	Grafiken	7
5.1	Includegraphics	7
5.2	Die Figure-Umgebung	7
6	Mathe-Modus	8
6.1	Inline-Umgebung	8
6.2	Abgesetzte Formeln	8
6.3	Befehle	9
6.3.1	Text in Formeln	9
6.3.2	Brüche	9
6.3.3	Indizes und Potenzen	9
6.3.4	Wurzeln	10
6.3.5	Summen und Produkte	10
6.3.6	Logische Operatoren	10
6.3.7	Fallunterscheidungen	10
6.3.8	Sonderzeichen	10
7	Tabellen	11
8	Hilfeseiten im Internet	11
9	Anhang	12

6. Auflage, licensed under [CC-BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/), 2009-2014, [Jannik Arndt](https://www.jannik-arndt.de/)

1 Einführung

1.1 Geschichte

\TeX [$\tau\epsilon\chi$] wurde zwischen 1977 und '86 von Donald E. Knuth entwickelt, weil dieser mit dem Drucksatz seiner eigenen Bücher unzufrieden war. Besonderheiten sind dabei die Genauigkeit und Vielfalt des mathematischen Formelsatzes ("I wanted to design something that would be still usable in 100 years."), sowie ein komplexer Algorithmus, der den bestmöglichen Zeilenumbruch berechnet (später z. B. von Adobe InDesign übernommen). Außerdem ist \TeX ein Vorreiter in der Entwicklung der Worttrennungsalgorithmen und vermeidet Hurenkinder und Schusterjungen (vereinsamte Zeilen am Ende bzw. Anfang einer Seite, dies wird von MS Word z. B. erst seit Version 2003 unterstützt). \TeX läuft auf allen Plattformen und liefert immer dieselben Resultate.

\TeX wurde Anfang der 80er, also noch während der Entwicklungsphase, von Leslie Lamport um sehr viele Makros ergänzt und dann als \LaTeX (für Lamport \TeX) veröffentlicht.

1.2 Prinzip

\LaTeX ist kein WYSIWYG-Editor, d.h. man sieht nicht direkt das Resultat, sondern arbeitet in einer einfachen Text-Datei, die beim Setzen interpretiert wird. Sämtliche Formatierung wird durch Befehle und Makros übernommen, d.h. Überschriften, Aufzählungen, Strukturen, etc. werden im Text durch Schlüsselwörter gekennzeichnet.

Besonders komfortabel ist (wenn man eingearbeitet ist) die Erstellung von komplexen mathematischen Formeln. Alle Befehle sind sehr klar und mathematisch strukturiert, die meisten sehr leicht zu erlernen, da sie durch das entsprechende englische Wort erzeugt werden.

Es gibt auch grafische Editoren, von denen ich aber stark abrate.

Dank der Formatierung durch Befehle erhält das Dokument eine logische Struktur, die nicht nur für ein einheitliches Aussehen sorgt, sondern auch z. B. die automatische Erstellung eines Inhaltsverzeichnisses mit nur einem Befehl zulässt.

2 Programme

Die Basisinstallation von \LaTeX ist immer kostenlos, die meisten Editoren auch.

2.1 Windows

Unter Windows dient MikTeX (miktex.org) als Basisinstallation. Eine passende grafische Oberfläche dazu bietet das plattformübergreifende Programm [TexMaker](#). Alternativen sind TeXnicCenter (texniccenter.org) oder TeXworks ([Google Code](#)).

2.2 Mac OS X

Die Basisinstallation für den Mac findet man als [MacTeX](#).

Als Programme können ebenfalls [TexMaker](#) oder aber TexShop (bei der Basisinstallation mit dabei) genutzt werden. Ein sehr viel mächtigerer Editor ist [TextMate](#).

2.3 Unix

Auf den meisten Unix-Systemen ist \LaTeX bereits vorinstalliert, KDE und Gnome enthalten meist auch Editoren. Für Ubuntu bietet das [Wiki der deutschen Ubuntu-User](#) Informationen.

3 Grundstrukturen eines Dokuments

Alle Befehle in LaTeX beginnen mit einem Backslash `\`. Darauf folgt meist ein Parameter in geschweiften Klammern `{...}` und manchmal optionale Parameter in eckigen Klammern `[...]`:

```
\befehl[opt. param]{param}.
```

Zum Beispiel ist `\sqrt{2} = \sqrt{2}`, aber `\sqrt[3]{2} = \sqrt[3]{2}`.

Ein LaTeX-Dokument besteht aus zwei Teilen: In der **Präambel** werden die `documentclass` definiert, Pakete geladen, Titel und Autor festgelegt, das Inhaltsverzeichnis erstellt, usw. Darauf folgt das eigentliche Dokument. Ein kurzes Beispiel:

```
\documentclass[a4paper,10pt]{article}
  \usepackage[ngerman]{babel}
  \usepackage[T1]{fontenc}
  \usepackage[utf8]{inputenc}
  \usepackage{amsmath}
  \title{Mein Dokument}
  \author{Max Mustermann}
  \date{\today}
\begin{document}
  Hallo Welt.
\end{document}
```

3.1 documentclass

Die Struktur eines LaTeX-Dokumentes wird über seine `documentclass` bestimmt. Optional kann man z. B. als Papierformat A4 angeben. Die `documentclass` dieses Skripts ist

```
\documentclass[a4paper,10pt]{article}
```

Eine gute Basis bilden die folgenden vier Vorlagen:

`article` Für kürzere Texte, Übungszettel, etc.

`report` Für größere Texte, unterstützt die Unterteilung in `chapter`

`book` Für größere Texte und mit `chapter`, beeinflusst außerdem die Seitenränder

`letter` Für Briefe, am besten mit Vorlage benutzen

Es gibt darüber hinaus noch viele Vorlagen aus dem KOMA-Skript sowie Vorlagen für Präsentationen, Grafiken, etc. Diese sind für den Einstieg aber nicht wichtig.

3.2 usepackage

Kodierung Damit z. B. Umlaute unterstützt werden, kann man in der Präambel des Dokuments die Kodierung festlegen. Dies sollte zur besseren Zusammenarbeit immer utf-8 sein:

```
\usepackage[utf8]{inputenc}
```

Es gibt außerdem `latin1` (Windows), `macce` und `applemac` (Mac).

Lokalisierung Damit das Inhaltsverzeichnis auf deutsch erscheint und die Worttrennung nach neuer deutscher Rechtschreibung erfolgt, muss folgendes Paket eingebunden werden:

```
\usepackage[ngerman]{babel}
```

amsmath Um mathematische Formeln etc. darzustellen, muss das von der American Mathematical Society entwickelte Paket `amsmath` eingebunden werden. Häufig ist das `amssymb`-Paket eine gute Ergänzung. Mehr zum Mathemodus weiter unten.

```
\usepackage{amsmath,amssymb}
```

3.3 title

Um LaTeX einen Titel oder eine ganze Titelseite (abhängig von der `documentclass`) generieren zu lassen, definiert man in der Präambel folgendes:

```
\title{Mein Dokument}  
\author{Max Mustermann}  
\date{29. Mai 2009}
```

Im Dokument selbst erzeugt man dann mit `\maketitle` den Titel.

3.4 Sonstige

Kommentare Mit `%` wird ein einzeliger Kommentar erzeugt.

Leerzeichen Mehrere Leerzeichen hintereinander (whitespace) werden ignoriert. Abstände lassen sich über “\ ”, “\,”, “\.”, “\;” und “\:” regulieren. Größere Abstände erhält man mit `\quad` oder (doppelt) `\qquad`.

Absätze Eine freie Zeile erzeugt einen neuen Absatz mit (meistens) eingerückter erster Zeile. Einen einfachen Zeilenumbruch erzeugt man mit `\par`, mit der Ergänzung `\smallskip`, `\medskip` oder `\bigskip` kann man den Abstand beeinflussen: `\par\bigskip`. Falls ein automatischer Einzug erfolgt, kann man diesem mit `\noindent` verhindern.

4 Unterteilung, Aufzählungen, etc.

Einer der großen Vorteile von LaTeX ist, dass der Autor sich lediglich um den Inhalt kümmert und diesen deklariert. LaTeX übernimmt die Formatierung der `documentclass` entsprechend.

4.1 Kapitel

Die Grundstrukturen eines Dokuments sind `chapter` (nicht bei `article`), `part`, `section`, `subsection` und `subsubsection` sowie `paragraph` und `subparagraph`. Die Überschrift des jeweiligen Absatz steht nachgestellt in geschweiften Klammern.

In eckigen Klammern kann man optional einen kürzeren Titel angeben, der im Inhaltsverzeichnis angezeigt wird: `\section[Kurztitel]{Mein voller Titel}`

Um die automatische Nummerierung der (Unter-)Kapitel bei den `section`-Befehlen zu verhindern, kann man den Befehl mit einem `*` versehen: `\subsection*{Mein Titel ohne Nummer}`.

Der Befehl `\setcounter{section}{ZAHL}` ermöglicht es, die Nummerierung anzupassen, wenn z.B. ein Kapitel am Anfang fehlt.

4.2 Normale Aufzählung: `itemize`

Um eine Aufzählung mit Punkten zu erstellen, gibt es folgenden Befehl:

```
\begin{itemize}
\item Punkt 1
\item Punkt 2
\end{itemize}
```

Dies sieht wie folgt aus:

- Punkt 1
- Punkt 2

4.3 Nummerierte Aufzählung: `enumerate`

Für nummerierte Aufzählungen bietet sich `enumerate` an:

```
\begin{enumerate}
\item Punkt 1
\item Punkt 2
\end{enumerate}
```

Dies sieht wie folgt aus:

1. Punkt 1
2. Punkt 2

Die Art der Nummerierung kann durch einen optionalen Parameter verändert werden. Dieser besteht aus dem Aufzählungstyp (“1” für Zahlen, “a” oder “A” für Buchstaben, “i” oder “I” für römische Zahlen) und “Dekoration”, wie Klammern, Punkte und Ähnlichem:

```
\begin{enumerate}[(a)]
```

4.4 Beschreibende Aufzählung: description

Soll am Anfang der Zeile ein fett gedrucktes Schlagwort stehen, aber die nächste Zeile eingerückt sein (im Vergleich zu `paragraph`), kann man eine Beschreibung wählen:

```
\begin{description}
  \item [{Schlagwort 1}] Die Beschreibung des Schlagwortes Nummer eins ist zwar lang,
    aber man kann sie immer noch sehr gut zuordnen, weil sie in der nächsten Zeile
    (und allen darauf folgenden Zeilen) eingerückt ist.
  \item [{Schlagwort 2}] Dies gilt natürlich auch für alle anderen.
\end{description}
```

Dies sieht wie folgt aus:

Schlagwort 1 Die Beschreibung des Schlagwortes Nummer eins ist zwar lang, aber man kann sie immer noch sehr gut zuordnen, weil sie in der nächsten Zeile (und allen darauf folgenden Zeilen) eingerückt ist.

Schlagwort 2 Dies gilt natürlich auch für alle anderen.

4.5 Typographie

Um Text hervorzuheben, gibt es verschiedene Möglichkeiten:

Hervorgehoben: `\emph{Mein \emph{eigener} Text}` *Mein eigener Text*

Kursiv: `\textit{Mein Text}` *Mein Text*

Geschnitten: `\textsl{Mein Text}` *Mein Text*

Roman: `\roman{Mein Text}` *Mein Text*

Fett: `\textbf{Mein Text}` **Mein Text**

Kapitälchen: `\textsc{Mein Text}` **MEIN TEXT**

Medium: `\textmd{Mein Text}` **Mein Text**

Ohne Serifen: `\textsf{Mein Text}` **Mein Text**

Kaligraphisch: `\cal{A B C}` *ABC* (nur in der Matheumgebung)

Code: `\texttt{Mein Text}` **Mein Text**

Code (ignoriert alle Zeichen): `\verb"Mein Text"` **Mein Text**
(alternativ `\begin{verbatim}...\end{verbatim}`)

4.6 Schriftfarbe

Text kann mit dem Befehl `\textcolor{FARBE}{TEXT}` bunt geschrieben werden, zum Beispiel:

```
\textcolor{blue}{roter Text}: roter Text.
```

5 Grafiken

Grafiken in einem textbasierten Editor zu platzieren, scheint zunächst ein wenig mühselig, man bekommt jedoch schnell ein gutes Gefühl dafür.

5.1 Includegraphics

Ein Bild anzeigen ist einfach, man benötigt das Package `\usepackage{graphicx}` und den Befehl `\includegraphics[Optionen]{Pfad/Dateiname}`.

Als Optionen stehen u.a. zur Verfügung:

- `angle` - die Drehung der Grafik im Uhrzeigersinn,
- `scale` - skaliert um einen bestimmten Faktor
- `height` und `width` - skaliert auf eine gegebene Größe, hier sind Werte wie “10cm”, “.5cm”, “`\textwidth`” (für die gesamte Textbreite), “`0.4\textwidth`” und ähnliche möglich

Unterstützt werden in der Regel die Dateitypen JPG, PNG, PDF, EPS und Vektorgrafiken. Ein Befehl sieht also zum Beispiel so aus:

```
\includegraphics[width=0.7\textwidth, angle=30]{Bild.png}
```

5.2 Die Figure-Umgebung

Um anzugeben, *wo* eine Grafik landen soll, ob sie einen Rahmen haben soll, oder eine Bildunterschrift, und wie man sie referenziert, setzt man sie meistens in eine `figure`-Umgebung:

```
\begin{figure}[Positionierung]
  \centering
  \includegraphics[Optionen]{Pfad/Dateiname}
  \caption{Bildunterschrift}
  \label{Referenz-Name}
\end{figure}
```

Die Umgebung entscheidet zunächst selbst, wo die Grafik am Besten passt. Man kann jedoch als *Positionierung* über folgende Buchstaben angeben, wo man sie am liebsten hätte:

- `h`: “here”, also etwa da, wo der Befehl im Quelltext auch ist
- `t / b`: “top”, oben auf der Seite bzw. “bottom”, unten auf der Seite
- `p`: auf eine extra Seite (evtl. mit anderen Grafiken zusammen)

Mit einem Ausrufezeichen hinter dem Buchstaben kann man seinem Wunsch mehr Nachdruck verleihen. Außerdem empfiehlt es sich, mehrere Buchstaben anzugeben, um die Alternativen festzulegen, falls der “Erstwunsch” nicht gut aussehen würde (die Grafik z. B. nicht mehr auf die Seite passt).

Der Befehl `centering` zentriert die Grafik. Möchte man den Text um die Grafik herum laufen lassen, muss man die Umgebung `wrapfigure` benutzen:

```
\begin{wrapfigure}[Zeilenhöhe]{Ausrichtung (l, r, L, R)}{Breite}
```

Die Bildunterschrift wird zusammen mit einem “Abbildung 1: ...” unter das Bild gesetzt, wobei die Zahl automatisch hochgezählt wird.

Der Referenz-Name ist sinnvoll, um im Text nicht fest “Abbildung 1” zu schreiben, sondern mit `\ref{Referenz-Name}` auf die Grafik zu verweisen. LaTeX fügt automatisch die richtige Nummer ein.

6 Mathe-Modus

6.1 Inline-Umgebung

Um mathematische Symbole, Variablen oder kurze Formeln im Text darzustellen, bietet sich die Inline-Umgebung an. Sie beginnt und endet mit einem \$, oder aber mit `\(...\)`. Buchstaben (Variablen) werden so anders dargestellt und sind leichter zu erkennen (z. B. n : n). Für griechische Buchstaben wird ebenfalls eine Mathe-Umgebung benötigt: `\varepsilon < 0` wird zu $\varepsilon < 0$.

Besonders größere Formeln werden in Inline-Umgebungen anders dargestellt, z. B. wird

`\sum_{1}^{n+1}(\frac{1}{n}+\sqrt{2^n})` zu $\sum_1^{n+1}(\frac{1}{n} + \sqrt{2^n})$ im Gegensatz zur abgesetzten Umgebung:

$$\sum_1^{n+1} \left(\frac{1}{n} + \sqrt{2^n} \right)$$

(Die Klammern wurden nun als `\bigg(... \bigg)` gesetzt, damit sie dem Bruch entsprechend groß sind.)

6.2 Abgesetzte Formeln

Es gibt verschiedene Umgebungen für verschiedene Anwendungsbereiche:

- `\[...]`

Dies ist die einfachste Möglichkeit, eine Formel darzustellen. Sie wird wie bei `equation*` zentriert und ohne Nummerierung dargestellt, Zeilenumbrüche sind nicht möglich.

$$\sum_{k=0}^n \binom{n}{k} \cdot a^k \cdot b^{n-k}$$

- `\begin{equation} ... \end{equation}`

Die Gleichung wird zentriert dargestellt und am rechten Rand durchnummeriert. Die Nummerierung kann durch ein Sternchen (`\begin{equation*}...`) unterdrückt werden. Zeilenumbrüche werden nicht unterstützt.

$$\sum_{k=0}^n \binom{n}{k} \cdot a^k \cdot b^{n-k} \tag{1}$$

- `\begin{gather} ... \\ ... \end{gather}`

Die `gather`-Umgebung erlaubt mehrere Zeilen, die mit `\\` voneinander getrennt werden und unabhängig voneinander zentriert werden.

$$(a+b)^{n+1} = \sum_{k=0}^{n+1} \binom{n}{k-1} a^k \cdot b^{n+1-k} + \sum_{k=0}^n \binom{n}{k} a^k \cdot b^{n+1-k} \tag{2}$$

$$= \sum_{k=0}^{n+1} \binom{n+1}{k} \cdot a^k \cdot b^{n+1-k} \tag{3}$$

- `\begin{align} ... & & \\ ... & & \end{align}`

Diese Umgebung erlaubt es, mehrere Zeilen durch `\\` zu erzeugen und diese mit einem oder mehreren `&` pro Zeile anzuordnen. Standardmäßig werden die Zeilen nummeriert, dies lässt sich aber unterdrücken (`\begin{align*}`). Mit dem Befehl `\nonumber` (oder auch `\notag`) in einer Zeile lässt sich für diese die Nummerierung unterdrücken.

(hier: `...b^{n+1-k} \\ \nonumber & = \sum...`)

$$(a+b)^{n+1} = \sum_{k=0}^{n+1} \binom{n}{k-1} a^k \cdot b^{n+1-k} + \sum_{k=0}^n \binom{n}{k} a^k \cdot b^{n+1-k} \tag{4}$$
$$= \sum_{k=0}^{n+1} \binom{n+1}{k} \cdot a^k \cdot b^{n+1-k}$$

Die Nummerierung kann mit dem Befehl `\numberwithin{equation}{section}` in der Präambel angepasst werden, so dass z. B. die Kapitelnummer vor die Nummerierung gesetzt wird. So wäre die letzte Gleichung mit (5.5) oder, wenn man `subsection` als Auflösung wählt, (5.2.5) beziffert.

Der Befehl `\tag{...}` lässt einen Namen frei definieren, die Nummerierung wird dann *nicht* hochgezählt.

Ergänzt man in der Präambel ein Attribut (`\usepackage[fleqn]{amsmath}`), werden alle Formeln statt zentriert linksbündig mit Einzug dargestellt.

6.3 Befehle

6.3.1 Text in Formeln

In Formelumgebungen werden alle Zeichen “mathematisch” interpretiert, d.h. Text wird kursiv und mit falschen Abständen angezeigt:

$$\lim_{n \rightarrow \infty} 1 + \frac{1}{n} = 1, \text{ weil } \lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

Dafür gibt es den Befehl `\text{...}`, der in einer mathematischen Umgebung Text normal darstellt:

$$\lim_{n \rightarrow \infty} 1 + \frac{1}{n} = 1, \text{ weil } \lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

Für längeren Text gibt es den Befehl `\intertext{...}`, der Text in einer neuen Zeile und linksbündig anzeigt, ohne, dass die Matheumgebung verlassen werden muss. Um z. B. Variablen im `intertext` anzuzeigen, bietet sich eine Inline-Umgebung an, also:

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt{(2n-1)(2n-1)}} = \sum_{n=1}^{\infty} \frac{1}{2n-1}$$

Majorantenkriterium: $|\sum_{n=N}^{\infty} a_n| \leq \sum_{n=N}^{\infty} b_n$

$$|a_n| = \left| \frac{1}{2n-1} \right| < \frac{1}{n} := b_n$$

Für mathematische Worte gibt es eigene Befehle, u.a. `\lim`, `\sin`, `\cos`, `\inf`, `\max`, `\limsup`. Die meisten LaTeX-Editoren haben eine Übersicht, allerdings ergibt sich der Befehl immer aus dem Namen.

6.3.2 Brüche

Der Befehl `\frac{Zähler}{Nenner}` (engl. fraction = Bruch) stellt Brüche dar. Diese sind beliebig verschachtelbar: `\frac{2^n \cdot \sqrt{\frac{1}{2}}}{\frac{2n^k+4}{\sqrt{2}}}`

$$\frac{2^n \cdot \sqrt{\frac{1}{2}}}{\frac{2n^k+4}{\sqrt{2}}}$$

6.3.3 Indizes und Potenzen

Die Befehle `_` und `^` stellen Text hoch oder tief. Diese beziehen sich immer auf das nächste Folgezeichen. Für mehr als ein Zeichen müssen diese in geschweifte Klammern gefasst werden (`{...}`).

$$2n = 2^n$$

$$n_{\{12\}} = n_{12}$$

$$\sum_{k=0}^n = \sum_{k=0}^n$$

$$\lim_{n \rightarrow \infty} = \lim_{n \rightarrow \infty}$$

$$a_{\{b_c\}} = a_{b_c}$$

6.3.4 Wurzeln

Wurzeln werden mit dem Befehl `\sqrt[Grad]{Radikand}` (sqrt für engl. Square Root = Quadratwurzel) dargestellt. Der Wert in eckigen Klammern ist optional und gibt den Grad an: `\sqrt{2}` und `\sqrt[3]{27}`

$$\sqrt{2} \quad \text{und} \quad \sqrt[3]{27} = 3$$

6.3.5 Summen und Produkte

Mit Hilfe der Hoch- und Tiefstellung werden auch Summen und Produkte dargestellt:

`\sum_{k=0}^n n^2` und `\prod_{k=1}^n n`

$$\sum_{k=0}^n n^2 \quad \text{und} \quad \prod_{k=1}^n n$$

Häufig findet man auch die Schreibweise `\sum \limits_{Anfangswert}^{Endwert}`. Diese ist bei angesetzten Formeln aber nicht zwingend nötig. In Inline-Umgebungen sorgt der Befehl dafür, dass die Grenzen nicht neben, sondern über bzw. unter dem Zeichen gesetzt werden.

6.3.6 Logische Operatoren

Logische Operatoren bestehen meist aus einem kleinen "l" und dem englischen Ausdruck:

`\neg = \lnot`, `\wedge = \land`, `\vee = \lor`.

6.3.7 Fallunterscheidungen

Für Fallunterscheidungen gibt es eine eigene Umgebung, die in einer Matheumgebung liegen kann:

`f(n) = \begin{cases} 0 & \text{für } n = 0 \\ 1 & \text{für } n = 1 \\ f(n-1) + f(n-2) & \text{für } n > 1 \end{cases}`

$$f(n) = \begin{cases} 0 & \text{für } n = 0 \\ 1 & \text{für } n = 1 \\ f(n-1) + f(n-2) & \text{für } n > 1 \end{cases}$$

6.3.8 Sonderzeichen

Zeichen	Befehl
∞	<code>\infty</code>
\rightarrow / \mapsto / \rightsquigarrow	<code>\to</code> / <code>\mapsto</code> / <code>\leadsto</code>
\leq / \geq	<code>\le</code> / <code>\ge</code>
\ll / \gg	<code>\ll</code> / <code>\gg</code>
\neq	<code>\neq</code>
\pm	<code>\pm</code>
\Leftrightarrow / \Rightarrow / \rightarrow	<code>\Leftrightarrow</code> / <code>\Rightarrow</code> / <code>\rightarrow</code>
\forall / \exists	<code>\forall</code> / <code>\exists</code>
\mathbb{N} / \mathbb{R}	<code>\mathbb{N}</code> / <code>\mathbb{R}</code>
ℓ	<code>\ell</code>

7 Tabellen

Tabellen sind (wie bei den meisten nicht-grafischen Editoren) in LaTeX etwas schwieriger und man sollte sich vor dem Erstellen der Tabelle Gedanken über ihre Struktur machen.

Der Befehl, um eine Tabelle zu definieren, ist

```
\begin{tabular}{FORMAT}
  TABELLENINHALT
\end{tabular}
```

Das Format bezieht sich zunächst auf die Spalten. Die Buchstaben `l`, `c` und `r` erzeugen dabei jeweils eine linksbündige, zentrierte oder rechtsbündige Spalte. Der Befehl `p{XXXcm}` erzeugt eine Par-Box mit Angabe der Breite. Senkrechte Trennstriche erzeugt man über ein `|` (pipe, auf Windows `Alt Gr + <`).

Der Tabelleninhalt wird zeilenweise geschrieben. Die Zellen werden dabei mit `&` getrennt, das Zeilenende mit `\\` markiert. Um einen waagerechten Strich zu erzeugen, schreibt man hinter das Zeilenende den Befehl `\hline` (horizontal line).

Eine Tabelle sieht also z. B. so aus:

Links	Mitte	Rechts	Par-Box
links-	zentrierte	rechts-	irgendwelcher mehrzeiliger Text, der automatisch umgebrochen wird.
bündige Werte	Texte	bündige Eingaben	

Der Quellcode dafür ist sehr knapp:

```
\begin{tabular}{l|c|r|p{3cm}}
  \textbf{Links} & Mitte & Rechts & Par-Box\\ \hline
  links- & zentrierte & rechts- & irgendwelcher mehrzeiliger Text, ... wird.\\
  bündige & & bündige & \\
  Werte & Texte & Eingaben & 
\end{tabular}
```

8 Hilfeseiten im Internet

Da LaTeX die wohl am meisten verbreitete Sprache für wissenschaftliche Texte ist, findet man bei Google i.d.R. auf den ersten Suchergebnissen schon eine Antwort auf Fragen zu Zeichen, Funktionen o.ä. Viele Universitäten veröffentlichen außerdem Übersichten und Schnellkurse.

Gute Hilfen, auf die ich bisher gestoßen bin, liste ich hier ohne Anspruch auf Vollständigkeit auf:

- [Detexify²](#) – Hier malt man ein gesuchtes Zeichen und kriegt prompt den Befehl dazu geliefert.
- [Wikipedia](#) – Eine gute Übersicht, welche Programme es gibt
- [WikiBooks](#) – Eine sehr umfassende Sammlung an Anleitungen zu fast allen Bereichen
- [Soft Skills Internetseiten](#) – Eine gute Linksammlung und Schrit-für-Schritt-Einführung mit Beispielen

Die folgenden beiden PDFs sollte man zusätzlich auch auf seinem Computer haben, denn sie sind eine perfekte Referenz für alles, was man je in Mathe braucht.

- [Mathematik mit LaTeX](#) – Eine sehr gute Übersicht
- [The Comprehensive LaTeX Symbol List](#) – 4947 Symbole und ihre Befehle

9 Anhang

Um einige Dinge auszuprobieren, ohne sich große Gedanken über die Präambel machen zu müssen, stelle ich hier meine Standard-Präambel zur Verfügung:

```
\documentclass[a4paper,10pt]{article}
  \usepackage[ngerman]{babel}
  \usepackage[T1]{fontenc}
  \usepackage[utf8]{inputenc}
  \usepackage{microtype}
  \usepackage{fixltx2e}
  \usepackage{graphicx}
  \usepackage[fleqn]{amsmath}
  \usepackage{amssymb,amsthm,amscd}
  \usepackage{color}
  \usepackage[a4paper, hdivide={2.5cm, *, 2cm}, vdivide={2.5cm, *, 2.5cm}]{geometry}
  \usepackage{multicol}
  \usepackage{enumerate}
  \usepackage[colorlinks=true, linkcolor=black, urlcolor=blue]{hyperref}
  \pagenumbering{arabic}

  \title{}
  \author{}
  \date{}

\begin{document}

  Hier kommt der Inhalt

\end{document}
```

Einige Definitionen, die einem bei der Arbeit mit Mathe helfen:

```
% Quantoren
  \newcommand{\A}{\: \forall \:}
  \newcommand{\E}{\: \exists \:}
% Zeichen
  \renewcommand{\*}{\, \cdot \,}
% Buchstaben
  \newcommand{\N}{\mathbb{N}}
  \newcommand{\R}{\mathbb{R}}
  \newcommand{\epsi}{\varepsilon}
% Funktionen
  \newcommand{\sgn}{\text{sgn}\,}
  \newcommand{\dx}{\, \mathrm{d}x}
  \newcommand{\lnx}{\, \mathrm{ln}x}
  \newcommand{\sinx}{\sin(x)}
  \newcommand{\cosx}{\cos(x)}
```