

# Latex – Eine Einführung

Sascha Lüdecke

1. November. 1998

## Inhaltsverzeichnis

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Einleitung</b>                                | <b>1</b> |
| 1.1      | Konzepte von $\text{\LaTeX}$ . . . . .           | 2        |
| 1.2      | Von $\text{\LaTeX}$ zum Papier . . . . .         | 2        |
| <b>2</b> | <b>Mein erstes Dokument</b>                      | <b>3</b> |
| <b>3</b> | <b>Meistgebrauchte Elemente</b>                  | <b>4</b> |
| 3.1      | Kapitel und Unterkapitel . . . . .               | 4        |
| 3.2      | Inhaltsverzeichnis und Querverweise . . . . .    | 4        |
| 3.3      | Schriftänderungen . . . . .                      | 5        |
| 3.3.1    | Schriftgrößen . . . . .                          | 5        |
| 3.3.2    | Schrifthervorhebungen . . . . .                  | 5        |
| 3.4      | Aufzählungen . . . . .                           | 5        |
| 3.5      | Tabellen . . . . .                               | 6        |
| 3.6      | Bilder . . . . .                                 | 7        |
| 3.7      | Vorformatierter Text (z.B. Quellcodes) . . . . . | 7        |
| <b>4</b> | <b>Literaturverzeichnisse</b>                    | <b>8</b> |
| <b>5</b> | <b>Epilog</b>                                    | <b>8</b> |
| <b>6</b> | <b>Weiterführende Literaturhinweise</b>          | <b>9</b> |

## 1 Einleitung

Mit Verbreitung der Computer wurde es möglich, Texte damit zu verarbeiten. Es entstand eine Vielzahl von Programmen, einige davon als Speziallösung für große Verlage, Druck- und Setzereien. Eines davon ist das Satzprogramm  $\text{\TeX}$  (Tau Epsilon Chi) von Donald E. Knuth. Er begann bereits Mitte der 70er Jahre mit der Entwicklung und konnte 1978 eine funktionierende Version bereitstellen, stabil war  $\text{\TeX}$  dann 1982.

T<sub>E</sub>X wurde als public domain veröffentlicht. Dies und der gut dokumentierte Quellcode führte zu der heute weiten Verbreitung, es ist inzwischen auf so gut wie jedem Betriebssystem verfügbar.

Mit T<sub>E</sub>X kann man nahezu alle Aufgaben eines Setzer lösen, unter anderem der Satz komplexer Formeln und Tabellen. Dazu sind jedoch große Fachkenntnisse im Satz und Druck sowie in Programmiersprachen erforderlich, denn T<sub>E</sub>X kennt ca. 900 Befehle, davon 300 Basisbefehle. Hinzu kommen noch notwendige Kenntnisse im Layout um dem Text ein ansprechendes Erscheinungsbild und gute Lesbarkeit zu verleihen.

Damit wäre T<sub>E</sub>X für den Privatmann sicher nicht geeignet, sondern nur einem kleinen Kreis vorbehalten. Leslie Lamport hat mit L<sup>A</sup>T<sub>E</sub>X ein umfangreiches Werkzeug entwickelt, mit dem der Benutzer nur noch die logische Struktur des Dokumentes, wie Kapitel, Absatz, Aufzählung etc., angeben muß. Diese wird dann in T<sub>E</sub>X-Befehle umgewandelt und weiterverarbeitet. L<sup>A</sup>T<sub>E</sub>X stellt zudem eine Reihe von professionellen Layouts zur Verfügung. Mit diesen wird eine professionelle grafische Gestaltung des Textes gewährleistet, ohne das man sich um Schriftgrößen, Abstände etc. kümmern muß.

## 1.1 Konzepte von L<sup>A</sup>T<sub>E</sub>X

*Textverarbeitungen* wie Word, AmiPro oder Framemaker sind WYSIWYG-Programme. Der Benutzer bearbeitet gleichzeitig den Inhalt und das Layout seines Textes und sieht auf dem Bildschirm, was später auf Papier gedruckt wird. Der Benutzer überzeugt sich sofort von der korrekten Erscheinungsweise seines Textes und kann diese ggf. gleich ändern.

Bei dem *Formatierungsprogramm* L<sup>A</sup>T<sub>E</sub>X ist dieser Prozeß in zwei Teile gespalten. Zuerst schreibt der Benutzer den Text mit seiner logischen Struktur in einem beliebigen Editor nieder und erzeugt dann die gedruckte Version. Gefällt ihm diese nicht, muß er den Text anpassen und wieder mit dem Formatierungsprogramm bearbeiten.

Auf den ersten Blick sind Textverarbeitungen idealer. Sie ersetzen die Schreibmaschine und übertreffen diese in ihrer Leistungsfähigkeit um Längen. Formatierungsprogramme ersetzen den traditionellen Setzer und spielen ihre Leistungsfähigkeit bei komplexen Strukturen, wie Formeln oder Tabellen aus. Ebenso kann man z.B. mit der Änderung eines einzelnen Befehls einen Text zweispaltig ausdrucken oder den gesamten Stil von Buch auf Artikel umstellen.

Intern formatiert L<sup>A</sup>T<sub>E</sub>X den Text auf Basis von Blöcken. Wie der Text aus Buchstaben, Wörtern und Absätzen besteht, werden diese auch aufgebaut. Mehrere Buchstabenblöcke, Größe je nach Höhe und Breite des Buchstabens, bilden einen Wortblock, mehrere Wortblöcke bilden einen Absatzblock usw. L<sup>A</sup>T<sub>E</sub>X versucht nun diese Blöcke ausgewogen auf einer Seite zu verteilen. Um eine gleichmäßige Füllung einer Seite zu erzielen, sind manche Abstände, wie z.B. zwischen Worten, Zeilen und Absätzen, dehnbar.

## 1.2 Von L<sup>A</sup>T<sub>E</sub>X zum Papier

Wie gesagt, arbeitet man mit L<sup>A</sup>T<sub>E</sub>X zweistufig. Nach dem Schreiben des Textes wird daraus die gedruckte Version erzeugt. Konkret wird aus einem T<sub>E</sub>X-File (`meintext.tex`) jedoch erst ein DVI-File (device independent) (`meintext.dvi`). Diese Datei enthält den

formatierten Text und Informationen über verwendete Schriftarten. Man kann sie anzeigen lassen (Preview des Textes mit `xdvi meintext.dvi`) oder mit einem Druckertreiber für einen Drucker vorbereiten (`dvips meintext.dvi` erzeugt `meintext.ps`).

Kurz:

```
emacs meintext.tex           // Text schreiben
latex meintext.tex           // meintext.tex --> meintext.dvi
xdvi meintext.dvi            // (evtl.) Preview des Drucks
dvips meintext.dvi           // meintext.dvi --> meintext.ps
```

Verwendet man im Text Querverweise oder ein Inhaltsverzeichnis, muß man  $\LaTeX$  zweimal ausführen. Beim ersten Lauf werden die Seitenzahlen etc. bestimmt und beim zweiten Lauf eingesetzt.

## 2 Mein erstes Dokument

Nach all der schönen Einleitung stellt sich die Frage, wie ein Dokument in  $\LaTeX$  wirklich aussieht. Es besteht aus drei Teilen:

- Einer Stilangabe (Dokumentklasse = Artikel, Buch, Bericht, Folie ...),
- einem Kopf (Metaangaben wie Titel, Autor, verwendete Pakete ...)
- und dem eigentlichen Text.

All diese Angaben macht man mit einem  $\LaTeX$ -Befehl, der mit dem Zeichen „\“ (Backslash) eingeleitet wird, doch zunächst ein kurzes Dokument:

```
\documentclass{article}
\title{Mein erstes Dokument}
\author{Ich}
\usepackage{german}
\begin{document}
\maketitle
```

```
Hallo Welt !
```

```
N"achster Absatz.
\end{document}
```

Das Dokument ist offensichtlich ein Artikel (`\documentclass{article}`) mit einem Titel und Autor (`\title{...} \author{...}`), der für Umlaute das Paket Deutsch (`\usepackage{german}`) verwendet. Der eigentliche Text ist: „`\begin{document} TEXT \end{document}`“, hier zwei Absätze.

Das Dokument enthält bereits die drei verschiedenen Arten von L<sup>A</sup>T<sub>E</sub>X-Befehlen: Die ohne Argument (`\maketitle` = drucke hier Titel, Autor und Datum des Dokumentes oder z.B. `\today` = Heutiges Datum oder `\large` = Text vergrößern) sind eher selten<sup>1</sup>.

Weitaus häufiger treten Befehle mit einem oder mehr Argumenten auf. Beispiele dafür sind `\author{...}`, `\textbf{...}` für fett gedruckten Text (Text Bold Face) oder `\footnote{...}` für eine Fußnote.

Die dritte Art sind Umgebungen, wie der eigentliche Text des Dokumentes. Stets mit `\begin{UMGEBUNG}` beginnend, enden sie immer mit `\end{UMGEBUNG}`. Eine Umgebung ist zum Beispiel der Text des Dokumentes, eine Aufzählung (`itemize` und `enumerate`) oder eine Tabelle (`tabular`). Man kann fast alle Umgebungen ineinander schachteln.

## 3 Meistgebrauchte Elemente

Nach den Konzepten und einem kurzen Dokument werde ich die Befehle und Umgebungen aufzählen, die in L<sup>A</sup>T<sub>E</sub>X am häufigsten verwendet werden.

### 3.1 Kapitel und Unterkapitel

Ein Kapitel wird in L<sup>A</sup>T<sub>E</sub>X mit dem Befehl „`\section{Kapitel über HEU}`“ eingeleitet. Es geht bis zum nächsten Befehl `\section{...}` oder sonst bis zum Ende des Textes. Mit `\subsection{...}` und `\subsubsection{...}` erzeugt man Unterkapitel.

### 3.2 Inhaltsverzeichnis und Querverweise

Für kleinere Artikel unter 20 bis 30 Seiten erzeugt man normalerweise kein eigenes Inhaltsverzeichnis. Bei größeren Dokumenten, die ja auch mehr Stoff beinhalten, ist es zur besseren Übersicht aber sehr hilfreich. Das Inhaltsverzeichnis wird aus den Kapitelangaben mit dem Befehl „`\tableofcontents`“ an der Stelle seines Auftretens erzeugt.

Ebenfalls bezieht man sich in längeren (aber auch in kürzeren) Texten oft auf andere Textstellen, Bilder oder Tabellen. Diese Stellen können sowohl vor als auch nach der aktuellen Position sein. In L<sup>A</sup>T<sub>E</sub>X setzt man dazu eine unsichtbare Markierung mit dem Befehl „`\label{NAME}`“, und verweist dann darauf. So wird „`\pageref{NAME}`“ durch die physikalische Seite des Labels ersetzt, „`\ref{NAME}`“ durch die Abschnitts-, Bild- oder Tabellenummer, je nachdem wo die Markierung gesetzt wurde. Also:

```
\section{Satz des Parmenides}\label{sec:parm}
Das Nichtseiende ist weder sagbar noch erkennbar.
...
```

Da es falsche Aussagen gibt, [...] mu"s man den Begriff des  
"Seins" (siehe Seite `\pageref{sec:parm}`, Abschnitt  
`\ref{sec:parm}`) genauer kl"aren.

---

<sup>1</sup>Anmerkung: Mit `{...}` kann man einen Textbereich festlegen. So wird in „`bla {\large foo bar} bla`“ der mittlere Teil groß, der Rest normal gedruckt: „`bla foo bar bla`“

Um über all die vielen Labels die Übersicht zu behalten, hat es sich eingebürgert, ein Prefix zu verwenden: `sec:` für Abschnitte, `fig:` für Bilder und `tab:` für Tabellen.

### 3.3 Schriftänderungen

Es ist ein schlechter Stil, zu viele Schriftarten und Schriftgrößen auf einer Seite zu verwenden, da dies das Auge zu sehr anstrengt. Dennoch sind Änderungen der Schriftgröße oder Hervorhebungen manchmal notwendig.

#### 3.3.1 Schriftgrößen

Die Schriftgröße ändert man mit einem Befehl ohne Parameter. Die neue Größe gilt bis zur nächsten Änderung, dem Ende der aktuellen Umgebung oder bei z.B. `{\large ...}` bis zur schließenden Klammer:

|                                    |                     |
|------------------------------------|---------------------|
| <code>\tiny</code>                 | <code>\large</code> |
| <code>\scriptsize</code>           | <code>\Large</code> |
| <code>\footnotesize</code>         | <code>\LARGE</code> |
| <code>\small</code>                | <code>\huge</code>  |
| <code>\normalsize</code> (default) | <code>\Huge</code>  |

#### 3.3.2 Schrifthervorhebungen

Diese Befehle beziehen sich auf den in eingeschlossenen Text. Es gibt sie auch in einer parameterlosen Version analog zu den Schriftgrößen:

|                               |                          |   |
|-------------------------------|--------------------------|---|
| <code>\textrm{...}</code>     | <code>\rmfamily</code>   | Römisch                                       |
| <code>\textit{...}</code>     | <code>\itshape</code>    | Schräger Text (auch <code>\emph{...}</code> ) |
| <code>\textbf{...}</code>     | <code>\bfseries</code>   | Fetter Text                                   |
| <code>\textsc{...}</code>     | <code>\scshape</code>    | Text in Kapitälchen                           |
| <code>\texttt{...}</code>     | <code>\ttfamily</code>   | Schreibmaschinentext                          |
| <code>\textnormal{...}</code> | <code>\normalfont</code> | Standardfont im Dokument                      |

### 3.4 Aufzählungen

Es folgen nun Umgebungen. Die ersten sind die drei verschiedenen Listen: `itemize`, `enumerate` und `description`. Sie haben alle die Form:

```
\begin{AUFZAEHLUNG}
\item Erstens
\item Zweitens
\item Drittens
...
\end{AUFZAEHLUNG}
```

Um einzelne Punkte numeriert aufzuzählen, nimmt man `enumerate`, `itemize` für eine einfache Aufzählung. Möchte man dagegen Begriffe beschreiben, bietet sich die Umgebung `description` an. Sie unterscheidet sich dadurch, das man „`\item[BEGRIFF] TEXT`“ verwendet. Der Begriff wird dann fettgedruckt dem Text vorangestellt. All diese Aufzählungstypen kann man beliebig ineinander schachteln.

### 3.5 Tabellen

Tabellen in  $\text{\LaTeX}$  sind etwas umständlich zu realisieren. Man kann zwar alle möglichen Tabellen erzeugen, sollte sich deren Aufbau vorher jedoch gut überlegen so ist die Vertauschung von zwei Spalten mehr als trivial. Zum Glück gibt es jedoch Spreadsheets unter Unix, wie `sc` oder `teapot`, die direkt  $\text{\LaTeX}$ -Code erzeugen. Tabellen sind nicht Bestandteil des normalen Textes, sondern Gleitobjekte (*floats*). Das heißt, das  $\text{\LaTeX}$  diese so verschieben kann, das der restliche Text gut auf die Seite paßt und keine Lücken entstehen. So zum Beispiel an den Anfang, das Ende der aktuellen Seite oder auf die nächste Seite. Das Grundgerüst für eine Tabelle sieht so aus:

```
\begin{tabular}[PLATZIERUNG]{FORMAT}
  TABELLENINHALT
\end{tabular}
```

**Platzierung** Mit der Angabe der Platzierung kann man die Position der Tabelle im Text beeinflussen. Mögliche Werte sind „`h`“ für *here*, „`t`“ für *top*, „`b`“ für *bottom* und „`p`“ für eine spezielle Seite mit Tabellen und Abbildungen. Man kann die Werte beliebig kombinieren, Standard ist `htbp`.  $\text{\LaTeX}$  versucht die Tabelle in dieser Reihenfolge zu platzieren.

**Format** An dieser Stelle definiert man die einzelnen Spalten der Tabelle. Mögliche Werte sind:

|                      |  |
|----------------------|--|
| <code>c</code>       | Zentrierter Eintrag  |
| <code>l, r</code>    | Links- bzw. rechtsbündiger Eintrag   |
| <code>p{SIZE}</code> | Spalte der Breite <code>SIZE</code> , die einen Absatz enthält. Z.B. <code>p{3cm}</code> |
| <code> </code>       | Senkrechter Strich (keine Spalte!)   |

Dies sind die wesentlichen Grundelemente. Für mehr sei auf die weiterführende Literatur verwiesen.

**Tabelleninhalt** Hat man nun Position und Format der Tabelle eingestellt, folgt der eigentliche Inhalt. Er wird Zeile für Zeile eingegeben. Ein Spaltenwechsel wird mit „`&`“ (Kaufmannsund) erzeugt, eine horizontale Linie mit `\hline` und das Ende der Zeile mit „`\\`“ markiert.

Nun ein Beispiel für eine Tabelle:

```
\begin{tabular}[ht]{|l|c|}
  \hline
```

```

Lehrstuhl & Professoren\\
\hline\hline
LS2 & Wegener\\
LS8 & Morik\\
LS10 & Doberkat\\
\hline
\end{tabular}

```

Ergibt:

| Lehrstuhl | Professoren |
|-----------|-------------|
| LS2       | Wegener     |
| LS8       | Morik       |
| LS10      | Doberkat    |

### 3.6 Bilder

In jedem guten Dokument sollten auch ein paar Bilder vorhanden sein. Man kann in  $\text{\LaTeX}$  entweder selbst zeichnen (das ist mehr als unkomfortabel und für automatische Generation besser geeignet), oder externe Dateien einbinden. Das dafür am häufigsten verwendete Format ist EPS (Encapsulated Postscript), welches z.B. auch Word von Microsoft versteht. Dazu muß man das Paket `epsfig` verwenden, was durch Angabe von „`\usepackage{epsfig}`“ im Dokumentkopf geschieht. Ein Bild bindet man mit dem Befehl „`\epsfig{file=DATEINAME}`“ in den Text ein. Verwendet man noch die Umgebung `figure`, erhält das Bild eine Nummer und Beschriftung und wird wie eine Tabelle als Gleitobjekt behandelt:

```

\begin{figure}[PLATZIERUNG]
  \epsfig{file=DATEINAME}
  \caption{MEIN TOLLES BILD}
\end{figure}

```

Der Befehl „`\epsfig`“ kennt neben `file=` noch folgende Optionen:

```

scale=FAKTOR      Bild wird mit FAKTOR skaliert
height=, width=   Explizite Höhen- und Breitenangabe
angle=WINKEL      Bild wird um WINKEL Grad gedreht. Drehpunkt ist
                  die linke untere Ecke

```

### 3.7 Vorformatierter Text (z.B. Quellcodes)

Es kommt vor, das man seinen Text mit kleinen Programmsplittern oder anderen Beispielen in einer Programmiersprache würzen möchte.  $\text{\LaTeX}$  bietet dafür die Umgebung `verbatim` an. Der eingeschlossene Text wird in einem nichtproportionalen Font (Schreibmaschine) genau so abgedruckt, wie er in der  $\text{\TeX}$ -Datei steht.

## 4 Literaturverzeichnisse

Ein gutes Dokument erfordert neben einer guten Idee oft auch eine gute Recherche in vorhandenen Veröffentlichungen. Damit der Leser diese nachvollziehen und sich selbst bei Bedarf vertiefen kann, gibt man die verwendeten Quellen in einem Literaturverzeichnis am Ende des Textes an und zitiert diese im Text geeignet. Natürlich gibt es in  $\text{\LaTeX}$  dafür entsprechende Befehle. Für das Literaturverzeichnis ist die Umgebung `thebibliography` gedacht. Innerhalb dieser werden einzelne Bücher oder Texte mit einem `\bibitem` Eintrag aufgelistet:

```
\begin{thebibliography}{MUSTER-MARKE}
\bibitem[MARKE]{BEZUG} EINTRAG
...
\end{thebibliography}
```

Die Angaben verhalten sich wie folgt: `MARKE` ist der Teil, der später im Text und auch im Literaturverzeichnis erscheint. `MUSTER-MARKE` ist ein Vorgabewert, der eigentlich nur so breit sein sollte, wie die größte verwandte `MARKE`. Damit wird die Einrücktiefe der Einträge im Literaturverzeichnis bestimmt. `BEZUG` ist der Text, der im Zitatbefehl verwandt wird. Er wird nicht gedruckt, sondern ist nur für den Autor gedacht. `EINTRAG` schließlich ist der eigentliche Eintrag. Man führt dort üblicherweise Titel, Autor und Erscheinungsjahr des Textes auf, Angaben, um den zitierten Text eindeutig zu identifizieren. Ein Beispiel ist:

```
\bibitem[Morik97]{MLskript} \emph{Skript Maschinelles Lernen},
  Prof. Katharina Morik, 1997
```

Im Text zitiert man dann mit dem Befehl „`\cite{BEZUG}`“, hier z.B.:

Die drei Lernaufgaben sind (aus `\cite{MLskript}`): `Begriffslernen, Regellernen und Funktionsapproximation`"

wird zu:

... (aus [Morik97]) ...

## 5 Epilog

$\text{\TeX}$  und  $\text{\LaTeX}$  sind sehr komplexe Programme, die wirklich viele Möglichkeiten bieten, in dieser Einführung habe ich die nur grundlegenden Dinge angesprochen. Was vielleicht noch fehlt, ist ein Abschnitt über Indexe, für einen Einstieg und kürzere Artikel sollten die Angaben hier allerdings reichen. Wer mehr wissen will, liest sicher gerne das Buch von Helmut Kopka (siehe 6.4), das ich als Grundlage für diese Einführung verwendet habe. Weitere Literatur und Quellen sind im letzten Kapitel zu finden.

Wer Fehler findet oder Verbesserungsvorschläge hat, erreicht mich unter folgender Adresse:

Sascha Lüdecke <sascha AT meta-x.de>

Mein Dank an Stefan Rüping für Durchsicht dieses Textes und konstruktive Vorschläge.

## 6 Weiterführende Literaturhinweise

Wer sich intensiver mit  $\LaTeX$  beschäftigen möchte oder hier nicht die gewünschte Hilfe findet, der möge folgende Quellen einsehen:

1. Homepage der „Deutschsprachige Anwendervereinigung  $\TeX$  e.V.“:  
<http://www.dante.de>  
Hier gibts  $\TeX$ , weitere Dokumentation und mehr Hilfe
2. Deutsche FAQ zu  $\TeX$   
<http://www.dante.de/faq/de-tex-faq>
3. Das  $\LaTeX$ -Kochbuch:  
<http://www.dante.de/TeX-Service-Paket/tex/cookbook/cookbook.html>  
Eine ausführlichere Einführung in  $\TeX$  (online Version) von einem Physiker der Uni Gießen, gibt es auch als offline Version.
4.  $\LaTeX$  - Einführung von Helmut Kopka, Band 1  
Addison Wesley, 1995, ISBN: 3-89319-664-1  
Eines der Standardwerke zu  $\TeX$ . Gut geeignet zum Selbstlernen und Nachschlagen, Grundlage für diese Einführung
5. Newsgroups zu Themen rund um  $\TeX$   
Deutsch: `de.comp.text.tex`  
Englisch: `comp.text.tex`