

Einführung in L^AT_EX

Version 1.01

Sascha Frank

SS 05

Zusammenfassung

Diese Einführung orientiert sich an den von mir im SS 05 gehaltenen L^AT_EX Kursen. Der Aufbau weicht von der Themenreihenfolge innerhalb der Kurse leicht ab. Ziel ist es einen ersten groben Überblick zu erhalten und ist mehr für den Einstieg als für's Nachschlagen gedacht. Im Laufe der Versionen wird sich der Inhalt und die Menge sicher noch mehr als einmal ändern.

Inhaltsverzeichnis

1	Wie funktioniert L^AT_EX ?	3
2	L^AT_EX bei uns im Pool	3
3	Basis L^AT_EX-Befehle	4
3.1	Dokumentenklassen	4
3.2	Anfang und Ende	5
3.3	Packages	5
3.3.1	Deutsche Sonderzeichen	5
3.3.2	Pakete die man häufig braucht	6
3.4	Besondere Zeichen	6
3.5	Gliederungsbefehle	7
3.6	Titelseite	8
4	Listen	8
4.1	Unnummerierte Listen	8
4.2	Nummerierte Listen	9
5	Schrift und Text	9
5.1	Schriftgröße	9
5.2	Schriftarten	10
5.3	Text	10
5.3.1	Textformatierung	10
5.3.2	Verbatim & Fußnoten	11
6	Zeilen- und Seitenwechsel	12
7	Mathematik	13
7.1	Mathe im Text	13
7.2	Text im Mathemodus	13
7.3	Umgebungen	14
7.4	Mehrzeilige Gleichungssysteme	14
8	Tabellen	15
8.1	Fußnoten innerhalb einer Tabelle	17
9	Zähler	17

1 Wie funktioniert L^AT_EX ?

1. Den Text mit einem beliebigen Editor erstellen z.B. test.tex
2. mit L^AT_EX übersetzen und eine DVI-Datei erzeugen → test.dvi
 - wenn man Verzeichnisse, Verweise usw. nutzt muß man es mindestens 2 machen
 - bei übersetzen werden noch andere Dateien erzeugt
3. das Ergebnis mit einem DVI-Anzeigeprogramm anschauen
4. in eine PostScript oder eine PDF umwandeln → test.ps oder test.pdf

2 L^AT_EX bei uns im Pool

Im Pool stehen mehrere Editoren zur Verfügung um damit zu texten. Ich für meinen Teil bevorzuge dabei den XEmacs. Dort sind u.a. auch DVI-Viewer (xdvi), GhostView (gs) für PostScript(*.ps) Dokumente, acroread bzw. xpdf für das Betrachten von PDF-Dateien (*.pdf), verfügbar.

Ablauf:

1. mit XEmacs einen Quelltext erzeugen
2. Mit der Dateiendung .tex abspeichern
3. xterm öffnen (ggf. in das Verzeichnis wechseln in die .tex Datei liegt)
4. latex Datei.tex eingeben, dann wird eine DVI-Datei erzeugt
5. Sollen auch noch Verzeichnisse, Verweise usw. erzeugt werden: das ganze nochmal – mit der Zeit macht man es schon von alleine zweimal
6. Bei Fehlermeldungen: In der angegebenen Zeile nach dem Fehler suchen
Fehlermeldungen sind am Anfang sehr verwirrend, insbesondere wenn der eigentliche Fehler an einer ganz anderen Stelle, als der genannten ist.
7. mit xdvi Datei.dvi das Ergebnis anschauen
8. Bis das gewünschte Ergebnis erzielt ist: Goto 4.
9. Wenn alles so aussieht wie man es will, mit dvips Datei.dvi eine ps Datei erzeugen

10. Die ps Datei kann mit gv Datei.ps betrachtet und ausgedruckt werden

Alternativ mit pdflatex:

1. mit XEmacs einen Quelltext erzeugen
2. Mit der Dateiendung .tex abspeichern
3. xterm öffnen (ggf. in das Verzeichnis wechseln in die .tex Datei liegt)
4. pdflatex Datei.tex machen, auch hier muß man es bei Verzeichnissen usw. zweimal laufen lassen
5. das Ergebniss mit xpdf Datei.pdf oder acroread Datei.pdf anschauen und drucken

3 Basis L^AT_EX-Befehle

3.1 Dokumentenklassen

Die jeweiligen Klassen sind eine Art “Dokumentenvorlage” und wird zu Beginn des Dokuments festgelegt. Sie bestimmt einen Teil des Aussehen, welche Befehle man nutzen kann usw..

Dabei ist zu beachten das jedes Dokument genau eine Dokumentenklasse einbindet. Es gibt vier Standardklassen:

- article für Aufsätze
- report wenn es etwas mehr wird
- book für Bücher
- letter, hier ist i.d.R. das amerikanische Format gemeint

Daneben gibt es zusätzliche Klassen:

- Für Vorträge: beamer
- Für dt. Briefe: dinbrief

Die Klassen können zudem noch Optionen besitzen:

- 12 pt für die Schriftgröße
- oneside für einseitigen Druck
- a4paper für das Papierformat

3.2 Anfang und Ende

Das eigentliche Dokument beginnt mit:

```
\begin{document}
```

und endet mit:

```
\end{document}
```

Läßt man diese Befehle weg oder vergißt sie, funktioniert das compilieren nicht.

Beispiel Nr.1 :

```
\documentclass[12pt,oneside,a4paper]{article}  
\begin{document}
```

Beispiel f\"ur ein sehr kleines \LaTeX-Dokument

```
\end{document}
```

3.3 Packages

Pakete stellen zusätzliche Eigenschaften oder Makros(=Befehle) zur Verfügung bzw. ändern bestehende. Die Pakete werden vor dem Dokumentenbeginn aufgerufen mit:

```
\usepackage{paketname}
```

Auch bei Paketen kann man Optionen nutzen, das sieht dann so aus:

```
\usepackage[paketooption]{paketname}
```

3.3.1 Deutsche Sonderzeichen

Um deutsche Sonderzeichen wie ä, ö, ü und ß verwenden zu können, bindet man vor dem Dokumentenbeginn das Paket `german` bzw. `ngerman` ein.

Beispiel Nr.2 :

```
\documentclass[12pt,twoside,a4paper]{article}  
\usepackage{german}  
\begin{document}
```

Dann funktioniert das auch mit dt. Sonderzeichen wie:

```
\"a, \"o, \"u und \ss .\  
\end{document}
```

Durch die Nutzung des Paketes `german` wird aus `tabel of contents`, dann ein *Inhaltsverzeichnis*.

Es gibt aber auch die Möglichkeit Sonderzeichen direkt zu tippen, was aber bei uns im Pool aufgrund der amerikanischen Tastaturen nicht funktioniert, aber an einer deutschen Tastatur.

```
\usepackage[german]{babel}  
\usepackage[latin1]{inputenc}
```

3.3.2 Pakete die man häufig braucht

Häufig genutzte Pakete sind

- `(n)german` deutsche Anpassung
- `amsmath` für mathematische Symbole
- `graphicx` um Grafiken einzubinden
- `color` für Farben

Unter <http://www.ctan.org/> stehen eine Menge von Paketen zum Download zur Verfügung.

3.4 Besondere Zeichen

<code>\</code>	Escape-Zeichen: maskiert Sonderzeichen. Leitet Kommandos ein.
<code>{ }</code>	umschließen Argumente, bilden Textblöcke, ...
<code>%</code>	Kommentarzeichen: Der Rest der Zeile wird ignoriert
<code>\$</code>	umschließpaarweise mathematische Formel im Text
<code>^ _</code>	Hoch- und Tiefstellung im Mathemodus
<code>&</code>	je nach Kontext - Tabulator o.ä.
<code>~</code>	Geschütztes Leerzeichen.

3.5 Gliederungsbefehle

Ein Dokument kann durch Überschriften und Unter-Überschriften gegliedert werden. Dabei stehen die folgenden Befehle zur Verfügung:

- `\chapter{...}` Kapitel nicht in article
- `\section` Abschnitt
- `\subsection` Unterabschnitt
- `\paragraph` Absatz
- `\subparagraph` Unterabsatz
- `\appendix` Anhang

Das Inhaltsverzeichnis wird durch \LaTeX selbst mit dem Befehl `\tableofcontents` erstellt.¹ Die einzelnen Überschriften werden in den Text getippt.

```
\chapter{Einleitung}
\section{Was ist \LaTeX ?}
bisschen Text
\subsection{\LaTeX f\"ur Linux }
noch mehr Text
\subsection{\LaTeX f\"ur Windows}
\chapter{Projekt}
neues Kapitel
....
\appendix
```

¹Hierbei muß man zweimal compilieren

3.6 Titelseite

In \LaTeX gibt es die Möglichkeit automatisch eine Titelseite erzeugen zu lassen. Dabei werden vor Dokumentenbeginn die folgenden Befehle festgelegt:

- `\title{...}` Titel des Dokument
- `\author{...}` Autor des Dokument
- `\date{...}` Datum
 - `\date{\today}` heutiges Datum
 - `\date{6.12.2004}` festgelegtes Datum
 - `\date{ }` kein Datum

Der Befehl `\maketitle` erzeugt dann die Titelseite. Die Klassen `report` und `book` machen jeweils eine extra Seite, ohne Seitennummer, für die Titelseite. Die Klasse `article` macht keine neue Seite für die Titelseite, und hat zudem noch eine Seitenzahl.

Der Befehl `\maketitle` darf erst hinter `\begin{document}` stehen.

```
\begin{document}
```

```
\maketitle
```

4 Listen

Mit \LaTeX lassen sich normale und nummerierte Listen erzeugen.

4.1 Unnummerierte Listen

```
\begin{itemize}
\item Ein Stichpunkt
\item Noch ein Stichpunkt
\end{itemize}
```

- Ein Stichpunkt
- Noch ein Stichpunkt

4.2 Numerierte Listen

```
\begin{enumerate}
\item Ein Stichpunkt
\item Noch ein Stichpunkt
\end{enumerate}
```

1. Ein Stichpunkt
2. Noch ein Stichpunkt

\LaTeX macht automatisch eine neue Zeile in der die Liste beginnt.

5 Schrift und Text

5.1 Schriftgröße

Von \LaTeX wird als Standardeinstellung eine Serifenschrift der Größe 10 pt benutzt. Wie schon bei den Dokumentklassen erwähnt kann man die Größe als Option ändern.

Spätere Änderungen im Dokument beziehen sich dann auf diese Einstellung. Mit folgenden Befehlen kann man innerhalb eines Dokumentes die Größe ändern:

normale Größe	normale Größe
<code>\tiny{winzig}</code>	winzig
<code>\small{klein}</code>	klein
<code>\large{groß}</code>	groß
<code>\Large{bisschen größer}</code>	bisschen größer
<code>\huge{riesig}</code>	riesig

`\Huge{Riesig}`

Riesig

Alternativ gibt es noch diese Variante:

normale Größe	normale Größe
<code>{\tiny winzig}</code>	winzig
<code>{\small klein}</code>	klein
<code>{\large groß}</code>	groß
<code>{\Large bisschen größer}</code>	bisschen größer
<code>{\huge riesig}</code>	riesig

`{\Huge Riesig}`

Riesig

5.2 Schriftarten

Neben der Größe läßt sich auch die Art der Schrift verändern:

(Anmerkung: Ich benutze als Schrift times, d.h. meine “normale” Schrift ist nicht die Standard normale Schrift.)

normale Schrift	normale Schrift
<code>\textbf{Fettdruck}</code>	Fettdruck
<code>\textit{Kursiv}</code>	<i>Kursiv</i>
<code>\textrm{Serifenschrift}</code>	Serifenschrift das rm steht für Roman
<code>\emph{Hervorgehoben}</code>	<i>Hervorgehoben</i>
<code>\texttt{Maschienenschrift}</code>	Maschienschrift

Mit solchen Änderungen sollten in Texten sparsam umgegangen werden.

5.3 Text

5.3.1 Textformatierung

```
\begin{flushleft}
Text kann links stehen
\end{flushleft}
```

Text kann links stehen

```
\begin{flushright}
Er kann aber auch rechts stehen.
\end{flushright}
```

Er kann aber auch rechts stehen.

```
\begin{center}
Oder einfach in der Mitte.
\end{center}
\end{flushright}
```

Oder einfach in der Mitte.

5.3.2 Verbatim & Fußnoten

Die Verbatimumgebung wird genutzt um Zeichen darzustellen, die innerhalb von \LaTeX geschützt sind bzw. die so ausgegeben werden sollen wie sie in der `.tex` Datei stehen. Ohne Verbatim wäre es nicht möglich mit \LaTeX Dokumente zu erstellen die \LaTeX Befehle zeigen, da der Compiler versucht diese Befehle auszuwerten, was nicht gewollt ist. Es eignet sich für die Darstellung von Sourcecode. Dabei verwendet ich zusätzlich noch die `small` Umgebung da der Text innerhalb von Verbatim im Vergleich zum restlichen Text zu groß wirkt.

```
\begin{small}  
\begin{verbatim}  
\LaTeX  
\end{verbatim}  
\end{small}
```

```
\LaTeX
```

Fußnoten

Wieder etwas sinnfreier²`\footnote{bezogen auf unser Problem}` Text.

Wieder etwas sinnfreier² Text.

Eine Fußnote³ läßt sich durch `\footnote{Text}` in den Text schreiben. Fußnoten werden in \LaTeX durchgehend mitgezählt.

²bezogen auf unser Problem

³Aber innerhalb von Tabellen macht man das anders

6 Zeilen- und Seitenwechsel

In `\LaTeX` sieht `\` nur das Trennen von einzelnen Wörtern, dabei spielt es keine Rolle ob es ein Leerzeichen oder `f` sind.

In `\TeX` sieht `\` nur das Trennen von einzelnen Wörtern, dabei spielt es keine Rolle ob es ein Leerzeichen oder fünf sind.

Wenn man einen Zeilenwechsel haben will muß man das extra angeben. Neben Leerzeichen und Zeilenwechsel gibt es Absätze.

Eine Leerzeile führt zu einem neuen Absatz.

Wenn man einen Zeilenwechsel haben will muß man das extra angeben. Neben Leerzeichen und Zeilenwechsel, gibt es Absätze.

Eine Leerzeile führt zu einem neuen Absatz.

Durch den Befehl `\noindent` wird das Einrücken des Absatzes verhindert.

In `\TeX` gibt es die folgenden Befehle mit denen Abstände, Zeilen- und Seitenbruch erzwungen werden können:

Befehl	Wirkung
<code>\ !</code>	Ein Leerzeichen
<code>\,</code>	schmalere Abstand (etwas breiter wie ein Leerzeichen)
<code>\newline</code> bzw. <code>\\</code>	Zeilenwechsel
Leerzeile	neuer Absatz (ggf. mit Einrückung) und Zeilenwechsel
<code>\newpage</code>	Seitenwechsel

Innerhalb der Mathematikumgebung gibt es noch zusätzliche Abstandsbefehle:

<code>\,</code>	ca. ein Leerzeichen
<code>\;</code>	schmalere Abstand (etwas breiter wie ein Leerzeichen)
<code>\quad</code>	ca. zwei Leerzeichen
<code>\qquad</code>	ca. drei Leerzeichen

7 Mathematik

Der Bereich Mathematik und mathematische Formel ist einer der Gründe warum ich fast ausschließlich mit \LaTeX meine Dokumente erstelle.

7.1 Mathe im Text

Innerhalb von Text wird durch ein $\$$ die Matheumgebung aufgemacht und durch ein weiteres $\$$ Zeichen geschlossen.

\dots α und β sind die ersten zwei \dots

\dots α und β sind die ersten zwei \dots

7.2 Text im Mathemodus

Innerhalb des Mathemodus werden Leerzeichen und Zeilenwechsel ignoriert. Die Buchstaben werden zusammengeschoben.

```
Seien  $a, b \in \mathbb{R}$ ,  
dann gilt  $(a+b)^2 = a^2 + 2ab + b^2$   $\text{\texttrm}$   
 $\text{\texttrm}$   
Besser  $\text{\texttrm}$   
 $\text{\texttrm}$   
Seien  $a, b \in \mathbb{R}$ ,  
 $\text{\texttrm{dann gilt}}$ ,  $(a+b)^2 = a^2 + 2ab + b^2$  $\text{\texttrm}$ 
```

Seien $a, b \in \mathbb{R}$, dann gilt $(a + b)^2 = a^2 + 2ab + b^2$

Besser

Seien $a, b \in \mathbb{R}$, dann gilt $(a + b)^2 = a^2 + 2ab + b^2$

Der Befehl $\text{\texttrm{}}$ bewirkt, daß der Text in diesem Fall als times roman dargestellt wird. Der Befehl $\text{\text{}}$ reicht für die Darstellung. Da ich times roman benütze, nehme ich halt $\text{\texttrm{}}$.

Es empfiehlt sich vor und nach $\text{\texttrm{...}}$ einen Abstand einzufügen.

7.3 Umgebungen

Bei Texten die zum überwiegenden Teil aus Formeln bestehen, gibt es spezielle Umgebungen. Hier sehen die Formeln dann auch noch ein Stück schöner aus als innerhalb der $\$$ -Umgebung.

```
 $\sum_{i=1}^n i = \frac{n(n+1)}{2}$   $\$$ 
```

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

```
\begin{displaymath}
\sum_{i=1}^n i = \frac{n(n+1)}{2}
\end{displaymath}
```

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

So werden Formeln durchnummeriert dargestellt:

```
\begin{equation}
\sum_{i=1}^n i = \frac{n(n+1)}{2}
\end{equation}
```

$$\sum_{i=1}^n i = \frac{n(n+1)}{2} \tag{1}$$

7.4 Mehrzeilige Gleichungssysteme

Für den Fall das mehrere Gleichung untereinander geschrieben werden, gibt es die `eqnarray` Umgebung.

```
\begin{eqnarray}
\frac{1}{a} &=& \frac{\frac{1}{2} + \frac{1}{2}}{a} \\
\frac{1}{a} &=& \frac{0.5 + 0.5}{a}
\end{eqnarray}
```

$$\frac{1}{a} = \frac{\frac{1}{2} + \frac{1}{2}}{a} \tag{1}$$

$$\frac{1}{a} = \frac{0.5 + 0.5}{a} \tag{2}$$

Durch den Befehl `\nonumber` wird das Mitzählen einer Gleichung unterbunden:

```
\begin{eqnarray}
\frac{1}{a} &=& \frac{\frac{1}{2} + \frac{1}{2}}{a} \nonumber \\
\frac{1}{a} &=& \frac{0.5 + 0.5}{a} \\
\end{eqnarray}
```

$$\frac{1}{a} = \frac{\frac{1}{2} + \frac{1}{2}}{a}$$

$$\frac{1}{a} = \frac{0.5 + 0.5}{a} \tag{1}$$

Der Befehl `\setcounter{equation}{0}` setzt Stand des Zählers der Gleichung auf null zurück.

8 Tabellen

Wie ich als in meinen Kursen sage:” Über Tabellen gibt es bei \LaTeX so viel zu sagen, daß man ganze Bücher darüber schreiben kann, und es auch getan hat.” Kurzum wie der bisherige Teil der Einführung kann dies, was hier steht nur als Einstieg in das Thema betrachtet werden.

```
\begin{tabular}{|l|c|r|p{1.5 cm}|}
\hline
left & center & right & Breite \\
l & c & r & p \\
\hline
\end{tabular}
```

left	center	right	Breite
l	c	r	p

l linksbündig r rechtsbündig
c zentriert p feste Breite mit Zeilenumbruch
| senkrechte Linie \hline waagerechte Linie

Zu Beginn wird die Anzahl der Spalten und deren Ausrichtung festgelegt. Bei der Ausrichtung gibt es drei Möglichkeiten: links (l), mitte(c) oder rechts(r). Die Einträge der Spalten werden in der Tabelle dann durch ein & Zeichen voneinander getrennt. Zwischen den Spalten wird durch | und ||, eine bzw. zwei vertikale Linien gesetzt. Um horizontale Linien zwischen den Zeilen zu erzeugen gibt es den Befehl `\hline`.

Man kann noch mehr als nur gewöhnliche Tabellen wie diese:

```
\begin{tabular}{|c|c|c|}  
\hline  
A & B & C \\  
\hline  
1 & 2 & 3 \\  
\hline  
4 & 5 & 6 \\  
\hline  
\end{tabular}
```

A	B	C
1	2	3
4	5	6

in \LaTeX erzeugen.

Mit `\multicolumn{Spaltenzahl}{Ausrichtung}{Inhalt der Spalte}` gibt es die Möglichkeit, mehrere Spalten zu einer größeren Spalte zusammenzufassen.

```
\begin{tabular}{|c|c|c|l|r|}  
\hline  
\multicolumn{3}{|l|}{test} & A & B \\  
\hline  
1 & 2 & 3 & 4 & 5 \\  
\hline  
\end{tabular}
```

test			A	B
1	2	3	4	5

Hinweis: Fußnoten funktionieren innerhalb von Tabellen nicht wie innerhalb eines Textes.

8.1 Fußnoten innerhalb einer Tabelle

```
\usepackage{setspace}\usepackage{threeparttable}
\begin{threeparttable}
\begin{tabular}{|l|c|r|}
\hline
A & B & C \\
\hline
1 & 2 & 3 \tnote{1} \\
\hline
\end{tabular}
\begin{tablenotes}\footnotesize
\item[1] Prognose 2003
\end{tablenotes}
```

Der Befehl `\tnote{...}` “ersetzt `\footnote{...}`”

A	B	C
1	2	3 ¹

¹ Prognose
2003

9 Zähler

\LaTeX zählt bei viele Umgebungen und Befehlen mit. Typische Standardzähler sind neben `page`, die Überschriften(vgl. S. 7).

Befehl	Wirkung
<code>\newcounter{name}</code>	definiert einen neuen Zähler
<code>\setcounter{name}{neuer wert}</code>	weist einem Zähler einen neuen Wert zu
<code>\addtocounter{name}{wert}</code>	Addiert einen Wert zum Zähler dazu
<code>\stepcounter{name}</code>	Addiert eine 1 zum Wert des Zählers dazu
<code>\roman{name}</code>	zählt mit römischen Ziffern
<code>\arabic{name}</code>	zählt mit arabischen Ziffern
<code>\alph{name}</code>	zählt mit kleinen lateinischen Buchstaben