

# Ausarbeitungen und Präsentationen mit L<sup>A</sup>T<sub>E</sub>X

Tobias Röhms (roehm@in.tum.de)

24. Oktober 2005

## Inhaltsverzeichnis

<b>1</b>	<b>L<sup>A</sup>T<sub>E</sub>X Grundlagen</b>	<b>2</b>
1.1	Was ist L <sup>A</sup> T <sub>E</sub> X? . . . . .	2
1.2	Geschichte von L <sup>A</sup> T <sub>E</sub> X . . . . .	2
1.3	Philosophie oder WYSIWYG vs. logisches Design . . . . .	3
<b>2</b>	<b>L<sup>A</sup>T<sub>E</sub>X Start: Von der Idee zum Ausdruck</b>	<b>3</b>
2.1	Eine einfache Eingabedatei . . . . .	3
2.2	Verarbeitung der Eingabedatei . . . . .	4
<b>3</b>	<b>L<sup>A</sup>T<sub>E</sub>X Syntax</b>	<b>5</b>
3.1	Befehle . . . . .	5
3.2	Umgebungen . . . . .	6
<b>4</b>	<b>L<sup>A</sup>T<sub>E</sub>X Sprachelemente</b>	<b>6</b>
4.1	Titel (-seite) . . . . .	6
4.2	Gliederung/ Inhaltsverzeichnis . . . . .	7
4.3	Formatierung . . . . .	7
4.4	Formeln . . . . .	8
4.5	nützliche Befehle . . . . .	9
4.6	L <sup>A</sup> T <sub>E</sub> X und die deutsche Sprache . . . . .	11
<b>5</b>	<b>Präsentationen mit L<sup>A</sup>T<sub>E</sub>X Beamer</b>	<b>11</b>
<b>6</b>	<b>Anhang</b>	<b>12</b>

# 1 $\LaTeX$ Grundlagen

## 1.1 Was ist $\LaTeX$ ?

$\LaTeX$  ist ein System zum Setzen von Dokumenten und eignet sich vor allem für große Dokumente (z. B. Bücher, Diplomarbeiten, ...), ist aber prinzipiell für jede Art von Veröffentlichung geeignet.

Typische Ausgabeformate sind .ps- Dateien (zum Drucken) und .pdf- Dateien (digitale Verbreitung über Internet etc.).

Es gibt viele Zusatzpakete für  $\LaTeX$ , die den Funktionsumfang erweitern, so z. B. für Mathematiker (Formelsatz) oder Musiker (Notensatz).

$\LaTeX$  ist in Wissenschaft und Industrie weitverbreitet und das Standardformat für wissenschaftliche Ausarbeitungen (v. a. in Verbindung mit Formeln).

## 1.2 Geschichte von $\LaTeX$

### Geschichte von $\TeX$

Die Geschichte von  $\LaTeX$  beginnt mit der Geschichte von  $\TeX$ .

Der Autor von  $\TeX$ , Donald E. Knuth (Informatik- Professor in Stanford), begann in den 70er Jahren die Buchreihe "The Art of Computer Programming" zu schreiben. Da er unzufrieden mit der Druckqualität seiner Bücher war, begann er 1977 damit, ein neues Textsatzsystem zu entwickeln:  $\TeX$  (zunächst dachte er, er würde dafür ein halbes Jahr brauchen, er brauchte dann aber fast 10 Jahre).

Eine der wesentlichen Eigenschaften von  $\TeX$  ist, dass es sicherstellt, dass ein Dokument immer genau gleich gesetzt wird - egal auf welcher Architektur, Betriebssystem, Drucker, ... (wenn ein neuer Drucker auf den Markt kommt muss nur ein Treiber geschrieben werden, damit  $\TeX$  und der Drucker kommunizieren können).

Allerdings stellt  $\TeX$  nur elementare Grundfunktionen bereit, so dass zur komfortablen Benutzung Makros benötigt werden - hier kommt  $\LaTeX$  ins Spiel.

### Geschichte von $\LaTeX$

$\LaTeX$  ist eines von vielen, aber das beliebteste Makropaket für  $\TeX$  (der Name steht für **L**amports **T** $\TeX$ ).

Es wurde ursprünglich von Leslie Lamport im Jahr 1984 entwickelt.

Die erste Version, die weltweit und in größerem Umfang genutzt wurde, war die Version 2.09.

Die aktuelle Version ist  $\LaTeX$  2<sub>ε</sub> (second edition), die von einem Team ent-

wickelt wurde.

### 1.3 Philosophie oder WYSIWYG vs. logisches Design

L<sup>A</sup>T<sub>E</sub>X unterscheidet im Gegensatz zu WYSIWYG- Programmen (Bsp. Office Textverarbeitung) zwischen Inhalt und Formatierung.

Das bedeutet konkret, dass wenn z. B. eine Überschrift “Einleitung“ angegeben wird, die Überschrift dann nicht zugleich mit dem Format gespeichert wird (also z. B. Schriftgröße 20 pt, Schriftart Arial, fett), sondern der Text zunächst nur als Überschrift gekennzeichnet wird. In einer anderen Datei (in den Paketen oder .sty- Dateien) wird dann das Format zentral festgelegt.

Vorteile des logischen Designs:

- Konzentration auf den Inhalt
- Formatierung/ Setzen übernehmen Experten
- Format - Anpassungen nur an einer Stelle im Dokument
- konsistente Formatierung

## 2 L<sup>A</sup>T<sub>E</sub>X Start: Von der Idee zum Ausdruck

### 2.1 Eine einfache Eingabedatei

L<sup>A</sup>T<sub>E</sub>X verarbeitet sogenannte Eingabedateien, welches ganz normale Textdateien sind, die mit einem beliebigen Texteditor erstellt werden können und die Endung .tex haben.

Dabei geht L<sup>A</sup>T<sub>E</sub>X nach dem logischen Design vor und unterscheidet zwischen den logischen Konstrukten Wort und Absatz.

Regeln für die Eingabedatei:

- Ein Leerzeichen (oder Zeilenende) beenden ein Wort. Die Anzahl der Leerzeichen zwischen zwei Worten ist L<sup>A</sup>T<sub>E</sub>X egal, es hat die Worte erkannt und setzt diese entsprechend den Formatdefinitionen in den Paketen.
- Eine Leerzeile beendet einen Absatz. Auch hier ist die Anzahl der Leerzeilen zwischen zwei Absätzen gleichgültig.
- L<sup>A</sup>T<sub>E</sub>X - Befehle beginnen mit einem “ \ “
- Kommentare beginnen mit einem “ % “ (der Rest der Zeile wird ignoriert)

## Ein einfaches Beispiel

```
\documentclass{article}
\usepackage{german}
\begin{document}
Hier kommt der Text hin
\end{document}
```

Eine  $\LaTeX$  - Eingabedatei ist in zwei Teile unterteilt: die Präambel (zentrale Definitionen, Einstellungen etc.) und den Body (Text). Der Body befindet sich zwischen den beiden Befehlen `\begin{document}` und `\end{document}`, alles was davor steht ist die Präambel.

`\documentclass{article}` legt die Dokumentenklasse fest.

In der Dokumentenklasse befinden sich die zentralen Formatdefinitionen für das gesamte Dokument.

In  $\LaTeX$  gibt es standardmäßig folgende Dokumentenklassen: `article` (für kurze Texte), `report` (für Texte mittlerer Länge), `book` (für Bücher), `slides` (für Folien) und `letter` (für Briefe). Die Dokumentenklasse ist obligatorisch und muss in jeder  $\LaTeX$  Eingabedatei vorhanden sein.

Mit dem Befehl `\usepackage{Paketname}` können zusätzliche Pakete und damit zusätzliche Funktionen geladen werden, im Beispiel wird das `german`-Paket für die deutsche Sprache geladen.

Wie oben bereits erwähnt begrenzen die beiden Befehle `\begin{document}` und `\end{document}` den Dokumententext. Diese beiden Befehle sind obligatorisch, sonst beschwert sich der  $\LaTeX$  - "Compiler".

## 2.2 Verarbeitung der Eingabedatei

Bei Linux ist  $\LaTeX$ /  $\TeX$  in der Regel mit dabei (ansonsten muss es als Paket nachinstalliert werden). Unter Windows ist eine  $\TeX$  Distribution notwendig (mehr dazu siehe unter Links).

Ist  $\LaTeX$  installiert, so kann die `.tex`- Datei mit dem Kommandozeilenbefehl `"latex datei.tex"` bzw. `"pdflatex datei.tex"` "kompiliert" werden (oder über entsprechende Funktionen einer Editorumgebung, auch hier wird i. d. R. eine Konsole automatisch geöffnet).

Dabei generiert  $\LaTeX$  einige Dateien:

<code>.pdf</code>	pdf- Datei (bei Verwendung von "pdflatex")
<code>.dvi</code>	(device independent) Ausgabe in geräteunabhängiger Form (bei Verwendung von "latex") um ein endgültiges Ergebnis zu erhalten muss diese Datei nochmals bearbeitet werden

.log	Verarbeitungs- Logdatei (Fehler, Meldungen)
.aux	interne Hilfsdatei
.toc	Zwischenspeicher für Inhaltsverzeichnis
...	...

## **L<sup>A</sup>T<sub>E</sub>X Fehlermeldungen**

Tritt während der Verarbeitung ein Fehler auf, so wird auf der Konsole eine Fehlermeldung ausgegeben (und auch in die log- Datei geschrieben).

Meist handelt es sich um einfache Fehler wie Vertipper oder obligatorische Teile der Eingabedatei wurden vergessen (z. B. Dokumentenklasse).

Solche Fehler sind leicht zu korrigieren, da L<sup>A</sup>T<sub>E</sub>X meist die Zeilennummer mit ausgibt, in der der Fehler aufgetreten ist.

Schwieriger ist es, T<sub>E</sub>X - Fehler zu korrigieren, da diese keine Zeilennummer ausgeben und oft etwas kryptisch und unverständlich sind.

Deshalb empfiehlt es sich - vor allem an Anfang - L<sup>A</sup>T<sub>E</sub>X öfters laufen zu lassen um so den Fehler besser lokalisieren zu können.

## **3 L<sup>A</sup>T<sub>E</sub>X Syntax**

L<sup>A</sup>T<sub>E</sub>X unterscheidet grundsätzlich zwischne Groß- und Kleinschreibung.

Es gibt mehrere Arten von L<sup>A</sup>T<sub>E</sub>X- Anweisungen, die beiden wichtigsten sind Befehle und Umgebungen.

### **3.1 Befehle**

Es gibt zwei Arten von L<sup>A</sup>T<sub>E</sub>X- Befehlen: textgenerierende Befehle und “Konfigurations-befehle“.

Textgenerierende Befehle veranlassen L<sup>A</sup>T<sub>E</sub>X einen bestimmten Text in einer (durch den Befehl festgelegten) Art zu setzen.

“Konfigurationsbefehle“ ändern die aktuellen Einstellungen.

allgemeine Syntax der Befehle:

```
\befehlsname [optionen]{parameter}
```

Ein Befehl beginnt mit einem \, gefolgt von dem Befehlsnamen. Dann folgen - in eckigen Klammern - die optionalen Parameter und dann - in geschweiften Klammern - die obligatorischen Parameter.

Die Anzahl sowohl der optionalen als auch der obligatorischen Parameter sind vom Befehl abhängig.

Beispiele:

`\today` gibt das aktuelle Datum aus (textgenerierend)  
`\documentclass` lädt entsprechende Dokumentklasse (Konfiguration)  
`\textbf{text}` setzt Text fett (textgenerierend)

## 3.2 Umgebungen

Im Gegensatz zu Befehlen, welche i. d. R. die Formatierung von Textparametern festlegen, definieren Umgebungen die Formatierung oder log. Einordnung von ganzen Textblöcken.

Die Einstellungen, die eine Umgebung definiert, gelten für den Textblock, der von den Befehlen `\begin{umgebungsname}` und `\end{umgebungsname}` eingeschlossen wird.

Beispiele:

`equation` Formel  
`verbatim` gibt Text aus ohne zu interpretieren

## 4 L<sup>A</sup>T<sub>E</sub>X Sprachelemente

### 4.1 Titel (-seite)

Mit L<sup>A</sup>T<sub>E</sub>X kann ganz einfach ein Titel bzw. eine Titelseite generiert werden. Bevor der Titel erstellt werden kann, muss man L<sup>A</sup>T<sub>E</sub>X mitteilen, welche Daten es im Titel anzeigen soll:

`\title{Titel}` Dokumententitel  
`\author{Autor}` Autor(en) (mehrere mit `\and` trennen)  
`\date{Datum}` Datum (wenn nicht spezifiziert wird das aktuelle verwendet)

Hat man die "Titeldaten" spezifiziert, so kann mit dem Befehl `\maketitle` der Titel erstellt werden.

Der Titel wird an der Stelle im Text gesetzt, die durch den Befehlsaufruf festgelegt wird.

Das Aussehen des Titels wird in der Dokumentenklasse definiert.

So erstellt die Dokumentenklasse `report` beispielsweise eine extra Seite für den Titel, während die Dokumentenklasse `article` den Titel nur in einem Bereich auf einer Seite anzeigt und keine Extra- Seite verwendet.

## 4.2 Gliederung/ Inhaltsverzeichnis

Mehrere Befehle stehen zur Gliederung von Dokumenten in  $\text{\LaTeX}$  zur Verfügung: `part`, `chapter`, `section`, `subsection`, `subsubsection`

Die Syntax der Gliederungsbefehle ist immer dieselbe: `\chapter{Kapitel 1}`.

Ein Gliederungsbefehl hat dabei immer zwei Auswirkungen:

- Der Text (im Beispiel “Kapitel 1“) wird entsprechend den Definitionen in der Dokumentenklasse gesetzt
- $\text{\LaTeX}$  “merkt“ sich die Tatsache, dass an dieser Stelle ein Kapitel mit Überschrift “Kapitel 1“ beginnt und verwendet diese Information z. B. bei der Generierung eines Inhaltsverzeichnisses.

Im Zusammenhang mit den Gliederungsbefehlen gibt es einige Besonderheiten:

- In der Dokumentenklasse `article` gibt es kein “`chapter`“ (damit Dokumente der Klasse `article` als eigenes Chapter in einem Dokument der Klasse `report` eingefügt werden können)
- “`part`“ dient der Gliederung von sehr langen Dokumenten (Bücher etc.) und hat auf die Nummerierung der anderen Gliederungsarten keine Auswirkung (die Nummerierung von z. B. Kapiteln startet nach einem `part`- Befehl nicht neu)
- Damit das Inhaltsverzeichnis aktualisiert wird muss  $\text{\LaTeX}$  zweimal laufen (1. Lauf: Infos in `.toc`- Datei schreiben, 2. Lauf: Verzeichnis aufgrund der aktualisierten `.toc`- Datei erstellen)

Wenn man die Gliederung seines Dokuments konsequent mit den Gliederungsbefehlen durchzieht, so kann mit dem Befehl `\tableofcontents` ganz einfach ein Inhaltsverzeichnis an einer beliebigen Stelle im Dokument erzeugt werden.

## 4.3 Formatierung

In  $\text{\LaTeX}$  gibt es auch Formatierungsbefehle, die  $\text{\LaTeX}$  anweisen, einen Text z. B. fett zu setzen.

Aufgrund der  $\text{\LaTeX}$  Philosophie (Stichwort `log. Design`) sollte Formatierung allerdings nur in selbstdefinierten Befehlen vorkommen (Bsp: Befehl oder Umgebung für Computerprogramme definieren und dort zentral Formatierung festlegen), da sonst das logische Design nicht verwirklicht wird und man dann gleich eine WYSIWYG- Textbearbeitung benutzen kann.

Mit dem `\emph{text}`- Befehl (für emphasize) kann L<sup>A</sup>T<sub>E</sub>X mitgeteilt werden, dass "text" hervorgehoben werden soll (die Formatierung legt die Dokumentenklasse fest).

L<sup>A</sup>T<sub>E</sub>X Befehle zur Formatierung (Auszug):

Format	Befehl
<i>kursiv</i>	<code>\textit{...}</code>
<b>fett</b>	<code>\textbf{...}</code>
Roman- Schrift	<code>\textrm{...}</code>
TypeWriter	<code>\texttt{...}</code>
Farbe	<code>\textcolor{red}{text}</code>

#### 4.4 Formeln

Eine der Stärken von L<sup>A</sup>T<sub>E</sub>X sind die Möglichkeiten zum Setzen von Formeln (unter anderem deshalb, weil es viele Pakete gibt, die sehr viele mathematischen Strukturen bereitstellen).

##### Formelumgebungen

Grundsätzlich gilt, dass eine Formel in einer Formelumgebung stehen muss. L<sup>A</sup>T<sub>E</sub>X kennt mehrere Formelumgebungen:

Umgebung	Kurzform	Erklärung
math	<code>\(...\)</code> <code>\$ ... \$</code>	normale Formel (im Text)
displaymath	<code>\[ ... \]</code>	abgesetzte Formel ohne Nummer
equation		abgesetzte Formel mit Nummer

##### Text in einer Formel

Eine Buchstabenfolge in einer Formelumgebung wird von L<sup>A</sup>T<sub>E</sub>X als Produkt interpretiert (Bsp:  $abc$ )

Um in eine Formel Text einzufügen kann man den `mbox`- Befehl (`\mbox{text}`) benutzen (Bsp:  $abc$ )

##### Symbole und griechische Buchstaben

In L<sup>A</sup>T<sub>E</sub>X gibt es für viele Symbole einen Befehl, mit dem man sie ganz einfach einfügen kann (diese Befehle sind nur in Formelumgebungen zulässig).

Beispiele:



<code>\alpha</code>	$\alpha$	griechische Kleinbuchstaben: <code>\name</code>
<code>\Omega</code>	$\Omega$	griechische Großbuchstaben: <code>\Name</code>
<code>\subset</code>	$\subset$	Teilmenge- Zeichen
<code>\Rightarrow</code>	$\Rightarrow$	“Daraus folgt“- Zeichen
<code>\infty</code>	$\infty$	unendlich
<code>\sum</code>	$\sum$	Summe
...	...	...

## einfache Strukturen

Einige Beispiele für relativ einfache Strukturen in Formeln:

- Brüche:  $\frac{1}{1+x}$  (`\frac{Zaehler}{Nenner}`)
- Wurzeln:  $\sqrt[n]{2}$  (`\sqrt[n]{ausdruck}`)
- Hochstellung:  $x^2$  (`x^{Exponent}`)
- Tiefstellung:  $x_2$  (`x_{Index}`)

Diese Elemente können auch verschachtelt werden:

$$x_t^{\sqrt{2}} \text{ (`x_{t}^{\sqrt{2}}`)}$$

## 4.5 nützliche Befehle

### unsichtbare Zeichen

L<sup>A</sup>T<sub>E</sub>X- Befehle zur Generierung von unsichtbaren Zeichen:

- Zeilenumbruch erzwingen: “`\`”
- Leerzeichen einfügen: “`\` ” (backslash leerzeichen)
- nicht trennbarer Wortzwischenraum: “`wort~wort`”

### Bilder einbinden

L<sup>A</sup>T<sub>E</sub>X kennt im Standard keine Bilder, allerdings können mit speziellen Paketen Anweisungen für T<sub>E</sub>X in die .dvi- Datei geschrieben werden, so dass T<sub>E</sub>X Bilder mitsetzt.

Um Bilder einzubinden kann das graphics- Paket (`\usepackage{graphics}`) verwendet werden.

Syntax: `\includegraphics{bild.jpg}`  
(Bild liegt im gleichen Verzeichnis wie .tex- Datei)

Hier ein Beispiel (Zentrierung mit Hilfe der center- Umgebung):



### einfache Aufzählung

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

Manipulation der Symbole:

einzelnes Symbol: `\item[-] Punkt 1` (“-“ ist neues Aufzählungssymbol des aktuellen Punktes)

im ganzen Dokument: `\renewcommand{\labelitemi}{\phi}` (phi ist dokumentenweit neues Aufzählungssymbol der Ebene ii = 2. Ebene)

### nummerierte Aufzählung

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

Manipulation des Nummerierungsstils:

```
\renewcommand{\labelenumi}{\Roman{enumi}}
```

(römische Zahlen in Aufzählungen der Ebene i, i = 1. Ebene)

Stile: Roman, alph, Alph, arabic

### Sonstiges

- Fußnoten<sup>1</sup>: `\footnote{Fu"snote}`

---

<sup>1</sup>Fußnotentext

- Label setzen: `\label{labelname}` (Text, Formel, Bild)
- Label referenzieren: `\ref{labelname}` (als Label wird Nummer verwendet, z. B. bei Formel: 4)
- Labelseite referenzieren: `\pageref{labelname}` (als Label wird Seitennummer verwendet)

## 4.6 $\LaTeX$ und die deutsche Sprache

Da  $\LaTeX$  ursprünglich im englischen Sprachraum entstanden ist, muss es für die deutsche Sprache angepasst werden.

In der Regel reicht es, einfach das `german-` Paket (bzw. `ngerman-` Paket) einzubinden, das diese Anpassungen im Hintergrund vornimmt.

Deutsche Sonderzeichen müssen dann mit " markiert werden (Bsp: "a für ä, "s für ß, "' für ") (das ist eine Konvention allein für das `german-` Paket!).

Zwei wichtige Befehle zur Absatzgestaltung:

- Absatzeinzüge ausschalten  
`\setlength{\parindent}{0pt}`
- Abstand zwischen zwei Absätzen definieren  
`\setlength{\parskip}{5pt}`

## 5 Präsentationen mit $\LaTeX$ Beamer

$\LaTeX$  Beamer ist eine Klasse mit der einfach Präsentationen mit  $\LaTeX$  erstellt werden können.

Die Präsentationen können im pdf- Format gespeichert werden und dann aufgrund des Vorhandenseins des Adobe Acrobat Readers (unabhängig vom Betriebssystem) komfortabel auf fast jedem Rechner gehalten werden.

### Einbinden von $\LaTeX$ Beamer

```
Dokumentenklasse   beamer      \documentclass{beamer}
Template auswählen  Stadtname   \usetheme{Warsaw}
```

Es gibt viele Themes für  $\LaTeX$ - Beamer, die alle nach Städten benannt sind (Einzelheiten siehe Homepage von  $\LaTeX$ - Beamer, Link siehe unten).

Text, der auf Folien erscheinen soll, muss in einer `frame-` Umgebung stehen.

Beachtet man diese wenigen Konventionen kann  $\LaTeX$ - Beamer (fast) wie  $\LaTeX$  benutzt werden (insbesondere behalten die  $\LaTeX$ - Befehle ihre Bedeutung).

## 6 Anhang

### Ausblick - mehr $\LaTeX$

- Befehle und Umgebungen selbst definieren
- BibTeX - Tool zur Verwaltung von Quellen/ korrektes Zitieren
- MakeIndex - Tool zur automatischen Indexgenerierung
- weiter nützliche Pakete
  - Paket listings (für Sourcecode)  
<http://www.ctan.org/info?id=listings>
  - Pakete von AMS (American Mathematical Society - Pakete für Mathematiker)  
<http://www.ams.org/tex/amslatex.html>
  - ...
- ...

### Links und Quellen

- Leslie Lamport: Das  $\LaTeX$  Handbuch (Addison- Wesley)
- Wikipedia  
<http://www.wikipedia.de>
- $\LaTeX$  Project (Homepage der  $\LaTeX$ - Entwickler,  $\TeX$ - Distributionen  
<http://www.latex-project.org/>
- $\LaTeX$  Beamer Homepage <http://latex-beamer.sourceforge.net/>
- Paket listings (für Sourcecode)  
<http://www.ctan.org/info?id=listings>
- AMS American Mathematical Society (führender Autor von Mathe-Paketen)  
<http://www.ams.org/tex/amslatex.html>