

## Einführung $\LaTeX$

Referat im Rahmen des Proseminar *Unix-Tools*  
an der Technischen Universität München  
im Sommersemester 2005

Niklas Klügel  
kluegel@in.tum.de



## 1 Historischer Abriss

- 1 Historischer Abriss
- 2 Hintergrund und Features von  $\LaTeX$ 
  - Philosophie oder: warum  $\LaTeX$  anstelle von WYSIWYG?
  - Andere Features von  $\LaTeX$

- 1 Historischer Abriss
- 2 Hintergrund und Features von  $\LaTeX$ 
  - Philosophie oder: warum  $\LaTeX$  anstelle von WYSIWYG?
  - Andere Features von  $\LaTeX$
- 3 Grundlegendes
  - Befehlsformat
  - Texteingabe
  - Gruppen und Umgebungen
  - Grundsätzlicher Dokumentaufbau
  - Dokumentklassen
  - Aufzählungen/Listen
  - Dokumentaufteilung
  - Inhaltsverzeichnisse

- 1 Historischer Abriss
- 2 Hintergrund und Features von  $\LaTeX$ 
  - Philosophie oder: warum  $\LaTeX$  anstelle von WYSIWYG?
  - Andere Features von  $\LaTeX$
- 3 Grundlegendes
  - Befehlsformat
  - Texteingabe
  - Gruppen und Umgebungen
  - Grundsätzlicher Dokumentaufbau
  - Dokumentklassen
  - Aufzählungen/Listen
  - Dokumentaufteilung
  - Inhaltsverzeichnisse
- 4 Tools

- 1 Historischer Abriss
- 2 Hintergrund und Features von  $\LaTeX$ 
  - Philosophie oder: warum  $\LaTeX$  anstelle von WYSIWYG?
  - Andere Features von  $\LaTeX$
- 3 Grundlegendes
  - Befehlsformat
  - Texteingabe
  - Gruppen und Umgebungen
  - Grundsätzlicher Dokumentaufbau
  - Dokumentklassen
  - Aufzählungen/Listen
  - Dokumentaufteilung
  - Inhaltsverzeichnisse
- 4 Tools
- 5 Bibliographie

## Historischer Abriss

- 1 Ursprünglich basiert  $\LaTeX$  auf dem Textverarbeitungssystem  $\TeX$  ( $\tau\epsilon\chi$ ), welches 1977 von Donald Knuth in einem Zeitraum von 8 Jahren entwickelt wurde. Knuth vervollständigte das Textverarbeitungssystem mit seinem eigenen Fontgestaltungssystem METAFONT. Die Kombination aus beiden Komponenten sollten die Qualität und Exaktheit des Drucks in Wissenschaftlichen arbeiten revolutionieren.

## Historischer Abriss

- 1 Ursprünglich basiert  $\LaTeX$  auf dem Textverarbeitungssystem  $\TeX$  ( $\tau\epsilon\chi$ ), welches 1977 von Donald Knuth in einem Zeitraum von 8 Jahren entwickelt wurde. Knuth vervollständigte das Textverarbeitungssystem mit seinem eigenen Fontgestaltungssystem METAFONT. Die Kombination aus beiden Komponenten sollten die Qualität und Exaktheit des Drucks in Wissenschaftlichen arbeiten revolutionieren.
- 2  $\LaTeX$  verwendet  $\TeX$  als Fundament und erweitert es durch eigene Macros und weiterentwickelte Schriftsätze. Hauptentwickler war Leslie Lamport, der 1985  $\LaTeX$  in der Version 2.09 freigab.
- 3 Die aktuellste, fertige Version ist  $\LaTeX 2\epsilon$  und wurde 1993 veröffentlicht. Der Code der neusten Entwicklungsversion ( $\LaTeX 3$ ) wurde im Februar 2005 öffentlich gemacht. Diese entstand ohne die Beteiligung von Leslie Lamport.



## Philosophie oder: warum $\LaTeX$ anstelle von WYSIWYG?

- $\LaTeX$  ist eine Markup-Sprache; Dokumentsatzanweisungen werden in den fließenden Text eingeschoben.
- Das eigentliche  $\LaTeX$  Dokument ist eine Textdatei, die mit jedem beliebigen Editor editiert werden kann.
- $\LaTeX$  übernimmt die komplette Formatierung und den Dokumentsatz. Der Verfasser eines Textes sucht sich lediglich ein Textlayout aus und kann sich vollkommen auf das Schreiben des Textes konzentrieren.
- Oft sind komplette Dokumentenlayouts vorgegeben; die meisten wurden mehrere Jahre hinweg entwickelt, einige sind sogar ISO-zertifiziert.
- $\LaTeX$  ist eine logische Markupsprache: anstatt z.B. bei einer Überschrift explizit anzugeben, dass diese eine bestimmte Grösse haben soll, wird der entsprechende Textblock lediglich als Überschrift deklariert; die Repräsentation ist abhängig vom gewählten Layout.

## Philosophie oder: warum $\LaTeX$ anstelle von WYSIWYG?

- $\LaTeX$  ist eine Markup-Sprache; Dokumentsatzanweisungen werden in den fließenden Text eingeschoben.
- Das eigentliche  $\LaTeX$  Dokument ist eine Textdatei, die mit jedem beliebigen Editor editiert werden kann.
- $\LaTeX$  übernimmt die komplette Formatierung und den Dokumentsatz. Der Verfasser eines Textes sucht sich lediglich ein Textlayout aus und kann sich vollkommen auf das Schreiben des Textes konzentrieren.
- Oft sind komplette Dokumentenlayouts vorgegeben; die meisten wurden mehrere Jahre hinweg entwickelt, einige sind sogar ISO-zertifiziert.
- $\LaTeX$  ist eine logische Markupsprache: anstatt z.B. bei einer Überschrift explizit anzugeben, dass diese eine bestimmte Grösse haben soll, wird der entsprechende Textblock lediglich als Überschrift deklariert; die Repräsentation ist abhängig vom gewählten Layout.
- $\LaTeX$  basiert somit auf der logischen Trennung zwischen dem reinen Verfassen und dem Dokumentendesign.

## Andere Features von $\LaTeX$

- $\LaTeX$  hat sich in den letzten 10 Jahren kaum geändert. Es stellt kein Problem dar, alte Dokumente zu verarbeiten und zu editieren.
- Die Qualität des Font- und Layoutrenderers ist derartig überzeugend, dass selbst professionelle DTP (Desktop Publishing) Systeme auf  $\TeX$  oder  $\LaTeX$  basieren.
- $\LaTeX$  bietet eine Vielzahl an breitgefächerten Template-Layouts für verschiedene Anwendungszwecke.
- Vieles was die Organisation des Dokuments angeht (Inhaltsverzeichnis, Index, Referenzen, Nummerierungen, etc.), funktioniert komplett automatisiert.
- Durch vereinheitlichte Layouts und der Möglichkeit andere Datenquellen wie z.B. Bilder und Einzelteile des Dokuments extern einzubinden kann man ohne weiteres im Team an einem Dokument arbeiten. Die auf Plaintext encodierten Textfiles sind zudem auch CVS-freundlich.
- Der  $\LaTeX$  Textfile wird normalerweise durch den eigenen Compiler in das .dvi-Format (**D**evice**i**ndependent format) übersetzt. Danach besteht die Möglichkeit das Ergebnis in verschiedene andere Fileformate, z.B. PS, PDF oder RTF zu exportieren.
- $\LaTeX$  gibt es für die verschiedensten Betriebssysteme, nicht nur für UNIX-Derivate.

## Wirklich Grundlegendes...

'\,' leitet einen Befehl ein

'%' Rest der Zeile wird als Kommentar interpretiert

## Das Befehlsformat (mit Beispiel)

```
\pgfdeclareimage[width=3cm]{Lion}{/home/lodsb/lion}  
\pgfuseimage{Lion}
```

```
LATEXCMD    := CMDTAG CMDNAME [CMDARGS]  
CMDTAG      := '\'  
CMDNAME     := STRING  
CMDARGS     := [OPTCMDARG] CMDARG  
CMDARG      := [CMDARG] '{' STRING '}'  
OPTCMDARG   := '[' OPTSTRING ']'  
OPTSTRING   := [(OPTSTRING ',')] STRING
```

- Jeder  $\LaTeX$ -Befehl kann mehrere Argumente haben; zudem ist es möglich, dass Argumente optional sind
- Befehle können -ungleich anderen Sprachen- aus Sonderzeichen bestehen.
- $\LaTeX$  ist -ungleich manchen anderen Markup-Sprachen wie HTML-`cAsEsEnSiTiVe`

## Leerzeichen und Zeilenumbrüche

mindestens eine Leerzeile

`'\par'`

Leerzeichen oder RETURN

`'\\'`

`'\ '`

Interpretation als Absatz

(Mehrere Leerzeilen haben keine Auswirkungen)

erzeugt explizit einen Absatz

wird als Wortende interpretiert

(Mehrere Leerzeichen haben keine Auswirkungen)

forciert einen Zeilenumbruch

erzeugt explizit ein Leerzeichen

## Trennungen

- **Trennung sollten in keinem Fall selbst vorgenommen werden, da dies komplett von Latex übernommen wird.** Man kann lediglich bestimmen, wie, oder ob ein Wort getrennt werden darf.
- Die für die verwendete Sprache entsprechenden Trennungsregeln werden in der Präambel als Argument für die Dokumentklasse angegeben.
- Möchte man dennoch eine Trennung *nur* an bestimmten Stellen zulassen, kann man die trennbaren Silben im Wort folgend kennzeichnen:  
'\-' Dies kann im fließenden Text geschehen.

Die Anweisung ist *lokal*, nur für das eben so deklarierte Wort gültig.

**Beispiel:** Donau\ -dampf\ -schiff

- Um Trennung *auch* an bestimmten Stellen zuzulassen, funktioniert folgendes:  
'"-' Dies kann im fließenden Text geschehen.

Die Anweisung ist wiederum nur *lokal* gültig.

**Beispiel:** Donau"-dampf"-schiff

'\hyphenation{'', '--' Dies muss in der Präambel des L<sup>A</sup>T<sub>E</sub>X-Dokuments festgelegt werden und ist *global* gültig

**Beispiel:** \hyphenation{Donau-dampf-schiff}

## Sonderzeichen

- Sonderzeichen sind in  $\LaTeX$  als eigenständige Befehle repräsentiert.
- Umlaute im Deutschen können entweder mit globalgültigen Befehlen oder abkürzenden Befehlen, die im Paket "german", oder "ngerman" enthalten sind, erzeugt werden.
- Ebenso wie die deutschen Umlaute können auch Sonderzeichen aus Fremdsprachen mittels  $\LaTeX$ -Befehl verwendet werden.

Zeichen	genereller $\LaTeX$ -Befehl	verkürzter $\LaTeX$ -Befehl
ä	<code>\{"a}</code>	<code>"a</code>
ö	<code>\{"o}</code>	<code>"o</code>
ü	<code>\{"u}</code>	<code>"u</code>
Ä	<code>\{"A}</code>	<code>"A</code>
Ö	<code>\{"O}</code>	<code>"O</code>
Ü	<code>\{"U}</code>	<code>"U</code>
ß	<code>\ss</code>	<code>"s</code>



## Gruppen (mit Beispiel2)

```
{\sf Dieser Text wird in SansSerif-Schriftart gerendert werden,  
wobei er ab {\bf hier fett gedruckt sein wird}; der Rest wird ab  
\huge hier sehr gro\ss{ } sein.}}
```

Dies ergibt den Output:

Dieser Text wird in SansSerif-Schriftart gerendert werden, wobei er ab **hier fett gedruckt sein wird**; der Rest wird ab **hier sehr groß sein**.

- Teilabschnitte des  $\LaTeX$ -Codes können gruppiert werden, indem man sie in geschweifte Klammern ('{' und '}') setzt.
- Die Auswirkung der Befehle ist lokal auf die Gruppe begrenzt in der sie enthalten sind.
- Ein Befehl kann in eine eigene Gruppe gefasst werden, damit man hinter ihm evtl. notwendige Leerzeichen auslassen kann (z.B. bei `sehr gro\ss{ } sein`).

## Umgebungen(mit Beispiel2)

- Große Abschnitte eines  $\LaTeX$ -Dokuments können in eine Umgebung gefasst werden.

```
\begin{name} % Beginn der Umgebung  
...  
\end{name} % Ende der Umgebung
```

- *name* bestimmt den Namen der Umgebung/des Befehls
- Umgebungen können auch (optionale) Parameter haben; diese werden wie bei normalen  $\LaTeX$ -Befehlen angegeben.
- Umgebungen gruppieren implizit, so ist der Wirkungsbereich von Befehlen lokal auf die Umgebung begrenzt.
- Wie bei den geschweiften Klammern bei Gruppen, müssen sich `\begin-` und `\end-` Teil einer Umgebung auf der gleichen Ebene befinden.
- Umgebungen haben die Eigenschaft, eigene, nur in ihr gültige Befehle bereitzustellen.
- $\LaTeX$  enthält eine Vielzahl an Umgebungen z.B. für Aufzählungen, Tabellen und Graphiken

## Grundsätzlicher Dokumentaufbau: Beispiel2

```
%Praeambel, aequivalent zu <head>...<\head> in HTML
\documentclass[papiergroesse,schriftgroesse,etc]{typ}
\usepackage[packetname]{arg}
\title{dokumenttitel}
\author{verfasser}
\date{\today}
%Body
\begin{document}
\maketitle % optional
...
\end{document}
```

## Grundsätzlicher Dokumentaufbau: Beschreibung

- In der Präambel werden alle wichtigen Einstellungen festgelegt, unter anderem werden  $\LaTeX$  das Papierformat oder sprachspezifische Einstellungen mitgeteilt und zusätzliche Pakete aufgerufen, die  $\LaTeX$  um weitere Befehle erweitern.
- Der Grundlegende Character des Dokuments wird mit `\documentclass[papiergroesse,schriftgroesse,etc]{typ}` festgelegt. **Die Angabe einer Dokumentenklasse ist obligatorisch.**
- Als optionale Parameter können das Papierformat (z.B. a4paper), die Standartschrifgröße (z.B. 12pt, 11pt und 10pt) und zusätzliche Layouteigenschaften (z.B. headspine und smallheadings) angegeben werden. Werden sie weggelassen, so werden die Standarteinstellungen verwendet.
- Mit `\usepackage[packetname]{arg}` lassen sich vor dem Beginn des eigentlichen Dokuments zusätzliche Pakete laden (z.B. german, ngerman, graphicx). Diese können wie Bibliotheken in Programmiersprachen neue Befehle einführen oder überschreiben, um so zusätzliche Funktionalitäten bereitzustellen.
- `\title{dokumenttitel}`, `\author{verfasser}` und `\date{datum}` legen den Titel des Dokuments, den Namen der Autorin (des Autors) und das Datum fest. Wird kein Datum angegeben, geht  $\LaTeX$  vom aktuellen Datum (dem Datum der Kompilation) aus. Mit dem Befehl `\maketitle` im Hauptteil des Dokuments wird der Inhalt der drei Variablen layoutspezifisch ausgegeben.

## Dokumentklassen

- Grundsätzlich gibt es bei  $\LaTeX$  vier Dokumentklassen: `letter`, `article`, `report` und `book`. Es gibt hierfür auch Äquivalente -die sogenannten KOMA Script-Klassen- die die europäischen typographischen Gepflogenheiten besser berücksichtigen. Die Namen der KOMA Script-Klassen: `scrbook`, `scrreprt`, `scrartcl` und `scrlttr2`.

## Dokumentklassen

- Grundsätzlich gibt es bei  $\LaTeX$  vier Dokumentklassen: `letter`, `article`, `report` und `book`. Es gibt hierfür auch Äquivalente -die sogenannten KOMA Script-Klassen- die die europäischen typographischen Gepflogenheiten besser berücksichtigen. Die Namen der KOMA Script-Klassen: `scrbook`, `scrreprt`, `scrartcl` und `scrlttr2`.
- `letter` Erzeugt einen professionell aussehenden Brief, im Gegensatz zu den nachfolgenden Layouts heißt der Body des Letters nicht `document` sondern `letter`; zusätzlich erwartet diese Umgebung die Empfängeradresse als Argument.
- `article` unterstützt die Gliederung des Dokuments in mehrere Abschnitte (`sections`), diese können mit `\tableofcontents` angezeigt werden. Ausserdem ermöglicht er weitere Formatierungsmaßnahmen, die für wissenschaftliche Arbeiten notwendig sind (z.B. `abstract`).
- `report` ist ähnlich dem `article`, außer, dass er noch zusätzlich richtige Kapitel enthalten kann.
- `book` ist das komplexeste Standardlayout, es bietet die Funktionalität des `report` und führt zudem noch buchdruckspezifische Besonderheiten ein, wie z.B. unterschiedliches Layout für Seiten mit geraden und ungeraden Seitenzahlen.

## Aufzählungen/Listen

- $\LaTeX$  bietet drei verschiedene Standardlistentypen an: `description`, `enumerate` und `itemize`. Eine Liste ist als Umgebung definiert, Listeneinträge werden mittels `\item[beschreibung]` hinzugefügt; der nach dem Befehl stehende Text wird bis zum nächsten `\item`-Befehl dem Listeneintrag hinzugefügt.

## Aufzählungen/Listen

- $\LaTeX$  bietet drei verschiedene Standardlistentypen an: `description`, `enumerate` und `itemize`. Eine Liste ist als Umgebung definiert, Listeneinträge werden mittels `\item[beschreibung]` hinzugefügt; der nach dem Befehl stehende Text wird bis zum nächsten `\item`-Befehl dem Listeneintrag hinzugefügt.
- Der `itemize`-Listentyp setzt vor jedem Listeneintrag lediglich einen Bullet, `description` verwendet das optionale Argument bei `\item[beschreibung]` und listet die einzelnen Einträge mit vorangehender Beschreibung auf.
- `enumerate` zählt die einzelnen Listeneinträge mit und stellt diese Zahl dem Eintrag voran.
- Selbstverständlich können Listen verschachtelt werden. Es gibt auch eine allgemeine Liste `list`, jedoch muss hier die Zählmethodik als Parameter mit angegeben werden.



## Dokumentaufteilung

- Bei langen Dokumenten ist es oft sinnvoll diese in verschiedene Abschnitte logisch zu gliedern. Hierfür gibt es je nach Abschnittsumfang entsprechende Befehle.
- Diese Gliederungsbefehle erledigen gleichzeitig die Nummerierung der Abschnitte, deren Überschrift, anderweitige Eingliederung in das Layout (z.B. Seitenüberschriften), sowie das Einfügen des Abschnitts in das Inhaltsverzeichnis.
- Folgende Gliederungsbefehle werden von den Layouts unherstützt:

	book, report	article
Teil	<code>\part[alttitel]{titel}</code>	<code>\part[alttitel]{titel}</code>
Kapitel	<code>\chapter[alttitel]{titel}</code>	-
Abschnitt	<code>\section[alttitel]{titel}</code>	<code>\section[alttitel]{titel}</code>
Unterabschnitt	<code>\subsection[alttitel]{titel}</code>	<code>\subsection[alttitel]{titel}</code>
Paragraph	<code>\paragraph[alttitel]{titel}</code>	<code>\paragraph[alttitel]{titel}</code>
Unterparagraph	<code>\paragraph[alttitel]{titel}</code>	<code>\subparagraph[alttitel]{titel}</code>

## Dokumentenaufteilung II

- Möchte man dem Abschnitt im Inhaltsverzeichnis einen anderen Namen geben als im fließenden Text, so kann man diesen als optionales Argument angeben.

```
\section[Titel im Inhaltsverzeichnis]{Titel}
```

- Möchte man die Nummerierung nicht anzeigen lassen funktioniert folgendes (nur bei `book` und `report`):

```
\section*{Titel}
```

## Inhaltsverzeichnisse

- Inhaltsverzeichnisse können überall im Dokument angezeigt werden:

```
\tableofcontents
```

- Da der  $\LaTeX$ -File linear abgearbeitet wird, muss man ihn mindestens zweimal compilieren, damit das Inhaltsverzeichnis korrekt ist.

## Tools

❶ Der  $\LaTeX$ -File muss im ASCII/Plaintext-Format gespeichert sein und sollte die Endung `.tex` haben.

❷ Das Kompilieren geschieht durch:

```
latex MeinDokument.tex
```

❸ Wenn alles geklappt hat, kann man sich den erhaltenen `.dvi`-File unter Unix mit X11 anschauen:

```
xdvi MeinDokument.dvi
```

❹ Man kann ihn aber auch in PostScript und PDF:

```
dvips MeinDokument.dvi -o MeinDokument.ps   PostScript  
dvi2pdf MeinDokument.dvi                    PDF
```

❺ Alternativ kann man sich auch eine  $\LaTeX$ -Editor-Umgebung installieren, diese übernimmt das Kompilieren und bietet meistens eine integrierte Befehlsreferenz mit automatisierter Befehlsvervollständigung:

<http://kile.sourceforge.net>

Kile (basierend auf KDE für Linux und Co)

<http://itexmac.sourceforge.net/>

iTeXMac (für MacOSX)

<http://www.winshell.de/>

WinShell (für Windows)



*LaTeX A document preparation system*

<http://www.latex-project.org/>  
1986-2005



*The (Not So) Short Introduction to LaTeX2e*

<http://ctan.tug.org/tex-archive/info/lshort/english/lshort.pdf>  
April 2004



*Wikipedia - LaTeX*

<http://en.wikipedia.org/wiki/LaTeX>  
2005



*Kochbuch für LaTeX*

<http://www.uni-giessen.de/hrz/tex/cookbook/cookbook.html>  
1994



*LaTeX Terminology*

<http://theoval.cmp.uea.ac.uk/~nlct/latex/csed/terms.html>  
2005



*Advanced LaTeX*

[http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex\\_advanced/latex\\_advanced.html](http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_advanced/latex_advanced.html)  
2004