

Verfassen wissenschaftlicher Texte mit  
L<sup>A</sup>T<sub>E</sub>X

# Übersicht

Umfrage

Einführung

Grundlagen

Text erstellen

Error

Aufzählungen

Struktur

Formelsatz

Mathe-Umgebungen

Advanced

Zahlen und Einheiten

Chemische Formeln

Gleitumgebungen

Tabellen

Fußnoten

Literaturverzeichnis

Fortgeschritten

Links

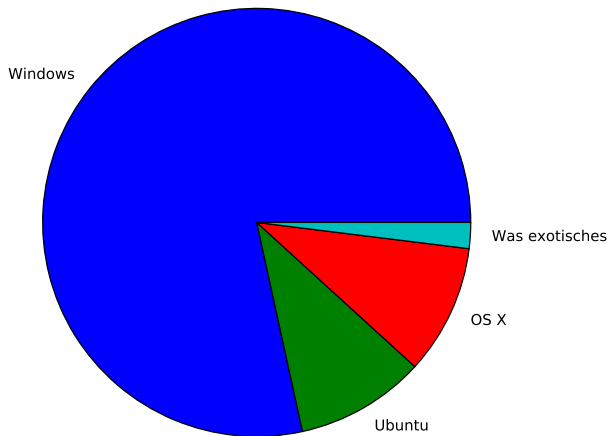
Makros

Tricks

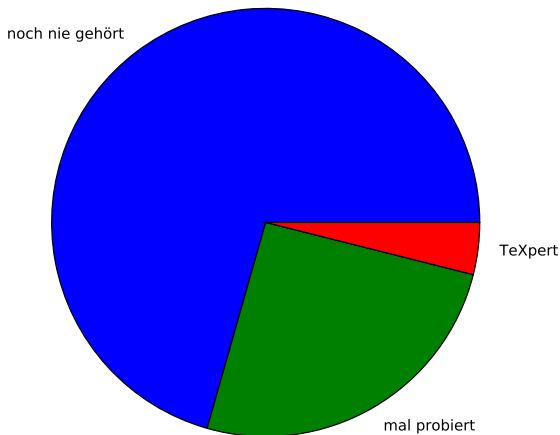
Ausblick

# Umfrage

## Betriebssystem



## Erfahrung mit LaTeX



# Einführung

## Was ist L<sup>A</sup>T<sub>E</sub>X?

- *Programmiersprache* zum Setzen von Text
- Markup  $\Rightarrow$  kein What-You-See-Is-What-You-Get
- L<sup>A</sup>T<sub>E</sub>X-Code  $\rightarrow$  Compiler  $\rightarrow$  Ausgabedokument (meist PDF)
- Open-Source, große Erweiterungsmöglichkeit (Pakete)
- Standard-Werkzeug in der Wissenschaft

## Warum L<sup>A</sup>T<sub>E</sub>X?

- Hervorragender Text- und Formelsatz
- Automatisierte Erstellung von Inhalts- und Literaturverzeichnis
- T<sub>E</sub>X-Dateien sind reine Text-Dateien
  - ⇒ Gut für Versionskontrolle geeignet
- Sehr gute Vorlagen für wissenschaftliches Arbeiten



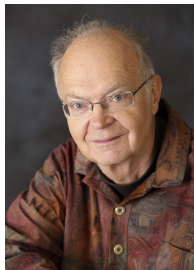
## Warum L<sup>A</sup>T<sub>E</sub>X?

- Ausgezeichnete Dokumentation
- Erweiterbar durch zahlreiche und mächtige Pakete
- Auf allen geläufigen Betriebssystemen verfügbar
- Ausgabe direkt als PDF mit Hyperlinks

## Geschichte

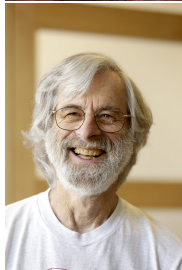
### T<sub>E</sub>X:

- Geschrieben von Donald E. Knuth 1978, um sein Buch „The Art of Computer Programming“ zu setzen
- Auf Aussprache achten!
- Version (2014): 3.14159265  $\rightarrow \pi$
- Viele Erweiterungen:  $\epsilon$ -T<sub>E</sub>X, pdfT<sub>E</sub>X, X<sub>Y</sub>T<sub>E</sub>X, LuaT<sub>E</sub>X



### L<sup>A</sup>T<sub>E</sub>X:

- Geschrieben von Leslie Lamport 1984
- Version (1994): L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\epsilon$</sub>
- L<sup>A</sup>T<sub>E</sub>X3 seit Anfang der Neunziger in Arbeit...



## Dieser Kurs

- In L<sup>A</sup>T<sub>E</sub>X gibt es immer viele Möglichkeiten, ein Ziel zu erreichen
- Wir zeigen einen modernen Ansatz
- Wir erklären, warum wir diesen Ansatz gewählt haben
- Weitere Ansätze werden an manchen Stellen kurz erwähnt

## Begriffe

**T<sub>E</sub>X-Engine** Implementierung von T<sub>E</sub>X, wird als Programm ausgeführt  
**T<sub>E</sub>X-Format** Paket, welches standardmäßig geladen wird, z.B. L<sup>A</sup>T<sub>E</sub>X

Eine Kombination davon ist oft ein neues Programm.

Beispiel: `dvilualatex` = LuaT<sub>E</sub>X + L<sup>A</sup>T<sub>E</sub>X + DVI-Output (statt PDF)

# Grundlagen

## Das Dokument

Diese drei Zeilen braucht jedes L<sup>A</sup>T<sub>E</sub>X-Dokument:

### Code

```
\documentclass[optionen]{klasse}  
  % Präambel  
  % .  
  % .  
\begin{document}  
  % Inhalt des Dokuments  
  % .  
  % .  
\end{document}
```

### \documentclass

Dokumentenklasse=Vorlage wählen,  
mit Optionen anpassen.

### Präambel

Globale Optionen und zusätzliche  
Pakete.

### document-Umgebung

Inhalt des Dokuments.

## Syntax: Befehle

L<sup>A</sup>T<sub>E</sub>X-Befehle beginnen stets mit einem \ (Backslash).

Obligatorische Argumente stehen in { }, optionale Argumente stehen in [ ].

### Syntax

```
\befehl[optional]{obligatorisch}
\befehl*[optional]{obligatorisch}
```

\* ruft häufig eine Alternativform des Befehls auf.

### Code

```
\documentclass[paper=a4]{scrartcl}

\frac{1}{2}

% Kommentar
```

### Erklärung

Dokumentenklasse scrartcl,  
Papierformat DIN A4.  
Zwei oder mehr Pflichtargumente,  
z.B. der Bruch.  
%-Zeichen für Kommentare.

## Syntax: Umgebungen

- Einstellungen für Bereich des Dokuments
- extrem vielseitig
- Können ggfs. auch Optionen übergeben bekommen

### Syntax

```
\begin{Umgebung*}[optional]{  
    obligatorisch}  
% .  
% .  
\end{Umgebung*}
```

### Beispiel

```
\begin{flushright}  
% .  
\end{flushright}
```



## Syntax: Umgebungen

Umgebungen können weitere Umgebungen enthalten, diese müssen aber in dieser Umgebung wieder geschlossen werden.

Geht:

```
\begin{document}
  \begin{flushright}
    % .
  \end{flushright}
\end{document}
```

Geht nicht:

```
\begin{itemize}
  \begin{enumerate}
\end{itemize}
  \end{enumerate}
```

## Standardpakete

Die hier aufgezählten Pakete sollten immer geladen werden, da sie wesentliche Funktionen bieten und wichtige Einstellungen vornehmen.

Paket	Funktion
<code>\usepackage{fixltx2e}</code> <code>\usepackage[aux]{rerunfilecheck}</code>	L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> korrigieren. Warnung, falls nochmal kompiliert werden muss.
<code>\usepackage{polyglossia}</code> <code>\setmainlanguage{german}</code> <code>\usepackage{fontspec}</code> % mehr Pakete hier	Deutsche Spracheinstellungen.  Für Fonteinstellungen
<code>\usepackage[unicode]{hyperref}</code>  <code>\usepackage{bookmark}</code>	Für Hyperlinks (z.B. Inhaltsverzeichnis → Kapitel). Erweiterte Bookmarks im PDF.

Die Reihenfolge ist manchmal wichtig, z.B. damit Pakete die Spracheinstellung kennen.

## KOMA-Script-Klassen

Doku: KOMA-Skript

- `scrartcl`, `scrreprt` und `scrbook`
- Sehr gute Vorlagen
- Schnell global mit Klassenoptionen anpassbar

Fürs Praktikum empfohlenene Klasse

```
\documentclass[...]{scrartcl}
```

## Fonteinstellungen

Standardeinstellung sind die Latin-Modern-Fonts.

### Latin Modern

```
\usepackage{fontspec}
```

### Alternativ: Tex Gyre

```
\usepackage{fontspec}  
\setmainfont{Tex Gyre Pagella}  
\setsansfont{Tex Gyre Heros}  
\setmonofont{Tex Gyre Cursor}
```

- Jede System-Schriftart kann genutzt
- Das ist i.A. nicht sinnvoll: *Hallo Welt in Comic Sans MS!*
- Schriften müssen zueinander passen
- Schriften müssen alle benötigten Sonderzeichen enthalten
- Bei Änderung auch Mathefont anpassen → später.

## Gerüst

```
\documentclass{scrartcl}

\usepackage{fixltx2e}
\usepackage[aux]{rerunfilecheck}
\usepackage{polyglossia}
\setmainlanguage{german}

\usepackage{fontspec}
% mehr Pakete hier

\usepackage[unicode,pdfusetitle]{hyperref}
\usepackage{bookmark}
% Einstellungen hier, z.B. Fonts

\begin{document}
  % Text hier
\end{document}
```

## Das Ausgabedokument erstellen

Es gibt verschiedene L<sup>A</sup>T<sub>E</sub>X-Kompiler, die verschiedene Ausgabeformate erzeugen können. Der modernste Compiler, der PDF-Dateien erstellt, ist **lualatex**.

### L<sup>A</sup>T<sub>E</sub>X-Dokument kompilieren

Terminal öffnen:

```
lualatex MeinDokument.tex
```

### Vorsicht!

- Es muss fast immer mindestens zweimal kompiliert werden.
- Es werden diverse Hilfs- und Logdateien erzeugt.
- Die Input-Dokumente müssen UTF-8 codiert sein.

## texdoc

L<sup>A</sup>T<sub>E</sub>X und (fast) alle Pakete sind hervorragend dokumentiert. Die Dokumentation wird automatisch mitinstalliert.

### Dokumentation zu einem Paket

```
texdoc paket
```

Dabei ist *paket* ein Suchstring.

### Nach Dokumentation suchen

```
texdoc -l name
```

Es ist wichtig zu lernen, Dokumentationen zu lesen. Probiert es an den oben genannten Paketen aus.

Alternativ kann man das Paket bei Google suchen, dann findet man auch die Dokumentation auf CTAN.

Text erstellen



## Text schreiben

### Beispiel

```
% Präambel
\begin{document}
  Hallo, Welt!

  Dies ist ein dummer Beispieltext.
  Er soll zeigen, dass LaTeX sich nicht um
  Zeilenumbrüche im Code      oder      zuviele
  Leerzeichen kümmert.

  Ein Absatz wird mit einer leeren Code-Zeile
  markiert.
\end{document}
```

## Konventionen für Text

- Höchstens ein Satz pro Code-Zeile
- Absätze werden durch eine Leerzeile markiert
- Im Fließtext sollten keine Umbrüche mit `\\` erzwungen werden

### Sonderzeichen

Viele Sonderzeichen sind L<sup>A</sup>T<sub>E</sub>X-Steuerzeichen. Damit diese im Text genutzt werden können, muss meist ein `\` vorangestellt oder ein Befehl genutzt werden.

#### Code

```
% \& \_ \textbackslash \$ \{ \}
```

#### Ergebnis

```
% & _ \ $ { }
```

## Textauszeichnung

Änderungen der Schrifteigenschaften sind mit diesen Befehlen möglich:

### Code

```
\textit{kursiv} \emph{kursiv}
\textbf{fett}
\textbf{\textit{fett-kursiv}}
\textrm{Serifen-Schrift}
\texttt{Mono-Schrift}
\textsf{Sans-Serif-Schrift}
\textsc{Kapitälchen}
```

### Ergebnis

*kursiv kursiv*  
**fett**  
***fett-kursiv***  
 Serifen-Schrift  
 Mono-Schrift  
 Sans-Serif-Schrift  
 KAPITÄLCHEN

Diese Befehle sollten sehr selten benutzt werden, semantischer Markup ist besser.

## Schriftgrößen

Gelten immer für den aktuellen Block, z.B. in einer Umgebung oder zwischen { }

### Code

```
{\tiny tiny}
{\small small}
{\normalsize normal}
{\large large}
{\huge huge}
```

### Ergebnis

tiny small normal large huge

### Alle Größen

```
\tiny, \scriptsize, \footnotesize, \small, \normalsize, \large,
\Large, \LARGE, \huge, \Huge
```

Auch diese Befehle sollten nur über semantischen Markup benutzt werden.

## Inhalt auslagern

### Code

```
\input{header.tex}  
\begin{document}  
  \input{Teil1.tex}  
  \input{Teil2.tex}  
  % .  
\end{document}
```

- Verschachtelung möglich
- Zur Aufteilung größerer Dokumente (z.B. diese Präsentation)
- Für häufig wiederverwendeten L<sup>A</sup>T<sub>E</sub>X-Code (header, Erläuterungen zu Fehlerrechnung, ...)
- Für per Skript erzeugte Tabelleninhalte

## Anführungszeichen

Die richtigen Anführungszeichen, wo die Satzzeichen hingehören und vieles mehr hängt von der Sprache ab. So macht man es richtig:

### Benötigte Pakete

```
\usepackage[autostyle]{csquotes} % nach polyglossia
\setotherlanguages{english, french} % andere Sprachen laden.
```

### Code

```
foo \enquote{bar} baz
\enquote{foo \enquote{bar} baz}
\textenglish{\enquote{foo}}
\textfrench{\enquote{foo}}
\textcquote{root}{foo}
```

### Ergebnis

```
foo „bar“ baz
„foo ,bar‘ baz“
“foo”
« foo »
„foo“ [1]
```

# Error

## Errors

- Alles kaputt. Was nun?
- Fehlermeldungen anfangs (und teils auch später) etwas kryptisch.

### Code

Ich begrüße euch mit einem `\enquote{Hallo Welt}`

```
(/usr/local/texlive/2014/texmf-dist/tex/generic/oberdiek/gettitlestring.sty))
(./test.out) (./test.out)
! Undefined control sequence.
l.16 Ich begrüße euch mit einem \enquote
                                {Hallo Welt}.
? █
```



## Errors

- Alles kaputt. Was nun?
- Fehlermeldungen anfangs (und teils auch später) etwas kryptisch.

### Code

Ich begrüße euch mit einem `\enquote{Hallo Welt}`

```
(/usr/local/texlive/2014/texmf-dist/tex/generic/oberdiek/gettitlestring.sty))
(./test.out) (./test.out)
! Undefined control sequence.
l.16 Ich begrüße euch mit einem \enquote
                                {Hallo Welt}.
? █
```

⇒ Vergessen csquotes zu laden.

## Lösungsstrategien

- Angegebene Zeile und vorherige Zeilen kontrollieren
- Teile des Codes auskommentieren um Ort des Fehlers einzugrenzen
- Google → [tex.stackexchange.com](https://tex.stackexchange.com)

# Aufzählungen

## Aufzählungen: Itemize

- L<sup>A</sup>T<sub>E</sub>X drei Umgebungen für Aufzählungen
- Unnummerierte Listen: `itemize`
- Verschachteln

### Code

```
\begin{itemize}
  \item Punkt 1
  \item Punkt 2
    \begin{itemize}
      \item Unterpunkt 1
      \item Unterpunkt 2
    \end{itemize}
  \item[→] Punkt 3
\end{itemize}
```

### Ergebnis

- Punkt 1
- Punkt 2
  - Unterpunkt 1
  - Unterpunkt 2
- Punkt 3

## Aufzählungen: Enumerate

Für nummerierte Listen wird `enumerate` genutzt.

### Code

```
\begin{enumerate}
  \item Punkt 1
  \item Punkt 2
    \begin{enumerate}
      \item Unterpunkt 1
      \item Unterpunkt 2
    \end{enumerate}
  \item Punkt 3
\end{enumerate}
```

### Ergebnis

- 1 Punkt 1
- 2 Punkt 2
  - 1 Unterpunkt 1
  - 2 Unterpunkt 2
- 3 Punkt 3

Anpassung der Listen mit dem Paket `enumitem`.

## Aufzählungen: Description

Zur Beschreibung von Stichwörtern wird `description` benutzt, dabei wird das Stichwort `\item` als optionales Argument übergeben.

### Code

```
\begin{description}
  \item[\LaTeX] gut
  \item[Word] böse
\end{description}
```

### Ergebnis

L<sup>A</sup>T<sub>E</sub>X gut  
Word böse

# Struktur

## Titelseite und Metadaten

L<sup>A</sup>T<sub>E</sub>X erstellt automatisch eine Titelei aus den Metadaten.

Mit der Klassenoption `titlepage=firstiscover` wird diese als eigene Seite gesetzt.

### Neue Klassenoption

```
\documentclass[... , titlepage=firstiscover, ...]{scrartcl}
```

### Empfehlung fürs Praktikum:

```
\title{101 Titel des Versuchs}
% Mehrere Autoren mit \and:
\author{Max Mustermann \and Maria Musterfrau}
\date{Durchführung: 26.09.2014, Abgabe: 29.09.2014}
```

### Titelseite generieren

```
\maketitle
```



## Gliederung

L<sup>A</sup>T<sub>E</sub>X bietet Befehle zum erstellen von Gliederungsebenen. Diese werden automatisch nummeriert und in entsprechend größerer und fatter Schrift gesetzt.

### Gliederungsebenen für `scrartcl`

```
\section{Überschrift}
\subsection{Überschrift}
\subsubsection{Überschrift}
\paragraph{Überschrift} % wird nicht nummeriert
\subparagraph{Überschrift} % wird nicht nummeriert
```

### Höhere Gliederungsebenen für `scrreprt` und `scrbook`

```
\part{Überschrift}
\chapter{Überschrift}
\section{Überschrift}
```

## Inhaltsverzeichnis

Aus den Gliederungselementen kann automatisch das Inhaltsverzeichnis erzeugt werden.

### Inhaltsverzeichnis generieren

```
\tableofcontents  
\newpage
```