

# Einführung in das Textsatzsystem $\text{\LaTeX}$

## Einführung und grundlegende Bedienung

Sebastian Blänsdorf

[blaensdorf@stud.uni-heidelberg.de](mailto:blaensdorf@stud.uni-heidelberg.de)

16. Oktober 2019

### Anmeldung

- Anmeldung im MÜSLI obligatorisch
- Verwaltung der Zetelpunkte über MÜSLI

### Anmeldung

- Anmeldung im MÜSLI obligatorisch
- Verwaltung der Zetelpunkte über MÜSLI

### Materialien

- Übungszettel und Vorlesungsfolien stehen auf der Vorlesungshomepage zum Download

### Prüfungsmodalitäten

- Bearbeiten der wöchentlichen Übungszettel
- Erreichen von mindestens 50% der möglichen Gesamtpunktzahl
- Übungspunkte ergeben Scheinnote (Notentabelle auf Vorlesungshomepage)
- 2 ECTS-Punkte für Übergreifende Kompetenz

- Übungszettel auf <http://latexkurs.github.io/exercises/>
- Ausgabe Mittwoch nach der Vorlesung

- Übungszettel auf <http://latexkurs.github.io/exercises/>
- Ausgabe Mittwoch nach der Vorlesung

### Abgabe

Je nach Aufgabenstellung per E-Mail oder ausgedruckt

**E-Mail** [abgabe@latexkurs.de](mailto:abgabe@latexkurs.de)

**Betreff** LaTeX-Abgabe: Musterfrau, Mustermann

**Dateinamen** uebung2.1\_musterfrau\_mustermann.tex

- Abgabe per E-Mail bis Mittwoch 16:00
- Abgabe der Ausdrücke Mittwoch in der Vorlesung oder in Zettelkästen in INF 205 (1. OG)
- Abgabe in Dreiergruppen möglich (ausgenommen erstes Blatt)

- 1 Einführung und grundlegende Bedienung
- 2 allgemeine Formatierung und Pakete
- 3 Mathematiksatz I
- 4 Gleitumgebungen, Tabellen
- 5 Mathematiksatz II
- 6 Grafiken, Abbildungen, TikZ
- 7 Diagramme
- 8 umfangreiche Dokumente
- 9 Bibliographien, mehrsprachiger Satz
- 10 Präsentationen
- 11 Briefe, Lebensläufe
- 12 komplexe Makros und Befehle

## Nutzung von $\LaTeX$

- *Wie* erreiche ich, was ich haben will?
- Wie funktionieren Syntax und Semantik?

## Verstehen von $\LaTeX$

- *Was* passiert, wenn ich auf den Knopf drücke?
- Was sind zugrundeliegende Paradigmen?

## Typographie

- *Warum* macht  $\LaTeX$  manche Dinge so und nicht anders?
- Auf welche Details sollte ich achten?



Häufig gehörte Aussage: „Typographie ist doch Geschmackssache. Ich mach das so, wie es schön aussieht!“

Häufig gehörte Aussage: „Typographie ist doch Geschmackssache. Ich mach das so, wie es schön aussieht!“

“ *Das Selbermachen ist längst üblich, die Ergebnisse oft fragwürdig, weil Laien-Typografen nicht sehen, was nicht stimmt und nicht wissen können, worauf es ankommt. So gewöhnt man sich an falsche und schlechte Typografie. [...] Jetzt könnte der Einwand kommen, Typografie sei doch Geschmackssache. Wenn es um Dekoration ginge, könnte man das Argument vielleicht gelten lassen, da es aber bei Typografie in erster Linie um Information geht, können Fehler nicht nur stören, sondern sogar Schaden anrichten.*

HPW, FF

”

# The Name of the Game

- Programm  $\text{T}_{\text{E}}\text{X}$  (Seit 1977)  
Geschrieben von Donald E. Knuth für sein Buch  
„The Art of Computer Programming“.  
„ $\text{T}_{\text{E}}\text{X}$ “ von griechisch τέχνη

# The Name of the Game

- Programm  $\text{T}_{\text{E}}\text{X}$  (Seit 1977)
- Makropaket  $\text{plainT}_{\text{E}}\text{X}$   
Macht  $\text{T}_{\text{E}}\text{X}$  für normale Nutzer bedienbar.

# The Name of the Game

- Programm  $\text{T}_{\text{E}}\text{X}$  (Seit 1977)
- Makropaket  $\text{plainT}_{\text{E}}\text{X}$
- großes Makropaket  $\text{LaT}_{\text{E}}\text{X}$  (Anfänge 1980er)  
Von Leslie Lamport: „Lamport’s  $\text{T}_{\text{E}}\text{X}$ “.  
Viele Vereinfachungen für den normalen Anwender.

# The Name of the Game

- Programm  $\text{T}_{\text{E}}\text{X}$  (Seit 1977)
- Makropaket  $\text{plainT}_{\text{E}}\text{X}$
- großes Makropaket  $\text{LaT}_{\text{E}}\text{X}$  (Anfänge 1980er)
- aktuelle, stabile Version:  $\text{LaT}_{\text{E}}\text{X } 2_{\varepsilon}$  (1994)  
„in einer  $\varepsilon$ -Umgebung von 2“...

# The Name of the Game

- Programm  $\text{T}_{\text{E}}\text{X}$  (Seit 1977)
- Makropaket  $\text{plainT}_{\text{E}}\text{X}$
- großes Makropaket  $\text{LaT}_{\text{E}}\text{X}$  (Anfänge 1980er)
- aktuelle, stabile Version:  $\text{LaT}_{\text{E}}\text{X } 2_{\epsilon}$  (1994)
- zukünftige Entwicklung:  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$   
noch nicht eigenständig verfügbar, aber als Paket `exp13` in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$

## Dafür ist L<sup>A</sup>T<sub>E</sub>X gut geeignet ...

- Alle Schriftstücke mit logischem Aufbau
  - Naturwissenschaftliche Arbeiten (hervorragender Mathesatz)
  - Geisteswissenschaftliche Arbeiten (hervorragende Mehrsprachigkeit, Bibliographieerstellung, Erstellung von Apparaten etc.)
  - Artikel, Diplomarbeiten, Dissertationen, ...
  - Buchreihen, Briefe
  - Präsentationen
- Viel „Missbrauch“ durch kreative Paketautoren



## Dafür ist L<sup>A</sup>T<sub>E</sub>X weniger gut geeignet ...

- Dokumente ohne logische Struktur
  - Präsentationen (bunt, drehend, blinkend, „durcheinander“)
  - Werbezetteln
  - Plakate
- Dokumente mit vielen uneinheitlichen Bildern, die frei bewegt werden

# Wie funktioniert T<sub>E</sub>X?

- WYSIWYM
- reine Textdateien
- keine versteckten Einstellungen
- Textauszeichnung durch besondere Befehle:
  - „Ich will einen Artikel schreiben!“
  - „Setze eine Überschrift!“
  - „Schreibe das folgende fett!“
  - „Setze eine Tabelle, die ...“

# Wie funktioniert T<sub>E</sub>X?

## Vorteile

- Stabilität und Portabilität
- geringe Dateigrößen
- Bearbeitung mit beliebigem Editor
- Textdateien *immer* lesbar
- Ausgabe überall gleich

## Nachteile

- Ergebnis nicht direkt sichtbar
- unintuitive Bedienung
- steile Lernkurve
- Bei Änderungen muss alles neu kompiliert werden
- komplizierte Layout-wünsche schwer realisierbar

Wie lässt sich Text von Befehlen unterscheiden?

Ansatz in *klassischen* Programmiersprachen:

```
print ( " Hallo Welt! " );
```

⇒ für ein Textsatzprogramm ungeeignet

- T<sub>E</sub>X ist eine Auszeichnungssprache (*markup language*)
- einzelne Zeichen haben besondere Bedeutung
- Backslash (\) dient als *escape character* und markiert den Anfang eines Befehls:  
`\chapter \section \author`

Einfachstes T<sub>E</sub>X-Dokument:

```
Hallo Welt! \bye
```

# Ein einfaches T<sub>E</sub>X-Dokument

- T<sub>E</sub>X ist eine Auszeichnungssprache (*markup language*)
- einzelne Zeichen haben besondere Bedeutung
- Backslash (\) dient als *escape character* und markiert den Anfang eines Befehls:  
`\chapter \section \author`

Einfachstes T<sub>E</sub>X-Dokument:

```
Hallo Welt! \bye
```

„tex dokument.tex“ erzeugt ein .dvi-Dokument und eine .log-Datei

- `\` *escape character*, Leitet Befehle ein
- `{}` *grouping character*, gruppieren zusammengehörende Zeichen  
z. B. Argumente `\textbf{fett}`
- `$` *math character*, startet und beendet Mathemodus
- `&` *tabbing character*, trennt Spalten in Tabellen
- `%` *comment character* Kommentiert den Rest der Zeile aus
- `^_~#` weitere Zeichen mit besonderer Bedeutung

```
\documentclass{minimal}  
\begin{document}  
Hallo Welt!  
\end{document}
```

Hallo Welt!



Dokumentenklassen legen grundlegende Eigenschaften des Dokuments fest:

- Layout
- Standardschriften
- Satzspiegel
- Gliederungsbefehle
- Aussehen von Verzeichnissen, Tabellen, Aufzählungen, ...

Eigenschaften sind durch Änderung von Optionen oder Laden von Paketen anpassbar.

## Standardklassen

article	(Kurze) Artikel
report	Reporte, Tagungsberichte
book	Bücher
letter	Briefe
minimal	Für Minimalbeispiele

## Spezialklassen

beamer	Für Präsentationen
tikzposter	Wissenschaftliche Poster

## KOMA-Script

scrartcl	Erweiterung von article
scrreprt	Erweiterung von report
scrbook	Erweiterung von book
scrlltr2	Sehr mächtige Briefklasse

- Gliederungen strukturieren Dokumente,
  - ermöglichen automatische Nummerierung, Eintragung in Verzeichnisse, Kolumnentitel etc.
  - Werden von der Dokumentenklasse definiert
  - Grundstruktur im Kernel festgelegt
- ⇒ bestimmte Elemente immer verfügbar

```
\part{Band I}
\chapter{Kapitel}
\section{Abschnitt}
\subsection{Unterabschnitt}
\subsubsection{Unterunterabschnitt}
\paragraph{Paragraph}
\subparagraph{Unterparagraph}
```

- Pakete bieten zusätzliche Funktionalität
- Arbeitserleichterungen
- Fehlerkorrekturen
- Einbinden in der Präambel mittels `\usepackage[option(en)]{paketname}`

```
\documentclass{article}
\usepackage{
  amsmath,
  hyperref,
}
\usepackage[left=2cm]{geometry}
```

# Grundbefehle

allgemein

<code>\textrm{Serifen}</code>	Serifen <i>Abcdxyz</i>
<code>\textit{kursiv}</code>	<i>kursiv</i> <i>Abcdxyz</i>
<code>\textsl{geneigt}</code>	<i>geneigt</i> <i>Abcdxyz</i>
<code>\textsf{serifenlos}</code>	serifenlos <i>Abcdxyz</i>
<code>\textbf{fett}</code>	<b>fett</b> <b>Abcdxyz</b>
<code>\texttt{Schreibmaschine}</code>	Schreibmaschine <i>Abcdxyz</i>
<code>\textsc{Kapitälchen}</code>	KAPITÄLCHEN <i>Abcdxyz</i>
<code>\emph{Hervorhebung}</code>	<i>Hervorhebung</i> <i>Abcdxyz</i>
<code>\\</code>	Zeilenende
<code>\par</code> oder Leerzeile	Absatzende
<code>\(E = \frac{p^2}{2m}\)</code>	Inline-Mathemodus: $E = \frac{p^2}{2m}$
<code>\[E = \frac{p^2}{2m}\]</code>	Display-Mathemodus: $E = \frac{p^2}{2m}$
<code>\tableofcontents</code>	Produziert Inhaltsverzeichnis
<code>\today</code>	aktuelles Datum

# Grundbefehle

## Schriftgrößen

<code>\tiny</code>	winzig
<code>\small</code>	klein
<code>\normalsize</code>	normal
<code>\large</code>	groß
<code>\Large</code>	größer
<code>\LARGE</code>	noch größer
<code>\huge</code>	riesig
<code>\Huge</code>	noch riesiger

## **Eingabe**

.tex      T<sub>E</sub>X-Datei mit Dokumententext

## **Ausgabe**

.pdf      pdfT<sub>E</sub>X-Ausgabe oder Umwandlung von (x)dvi

## **Eingabe**

.tex      T<sub>E</sub>X-Datei mit Dokumententext

## **Ausgabe**

.pdf      pdfT<sub>E</sub>X-Ausgabe oder Umwandlung von (x)dvi

## **Hilfsdateien (nur schreiben)**

.log      Log-Datei mit Informationen, Warnungen, Fehlern



## **Eingabe**

.tex      T<sub>E</sub>X-Datei mit Dokumententext

## **Ausgabe**

.pdf      pdfT<sub>E</sub>X-Ausgabe oder Umwandlung von (x)dvi

## **Hilfsdateien (nur schreiben)**

.log      Log-Datei mit Informationen, Warnungen, Fehlern

## **Hilfsdateien (schreiben und lesen)**

.aux      Hilfsdatei mit temporären Informationen

.toc      table of contents

.lof      list of figures

.synctex.gz      nötig für die SyncT<sub>E</sub>X-Funktion

⋮      ⋮

- Übungen setzten T<sub>E</sub>X Live 2018 voraus
- Installationsanleitung auf Vorlesungswebseite
- Editor



Happy T<sub>E</sub>Xing