

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

## Umfangreiche Dokumente

Sebastian Blänsdorf

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

4. Dezember 2019

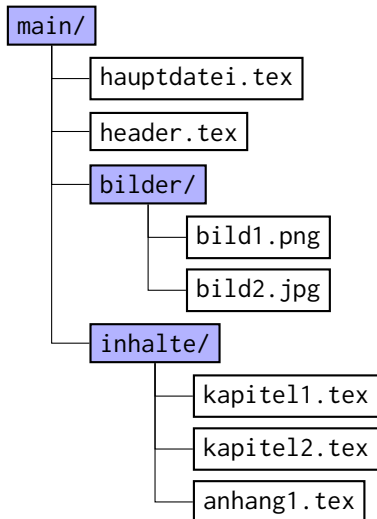
- 1 Projekte mit vielen Dateien
- 2 Header
- 3 Vor dem Inhalt
  - Titelei
  - Verzeichnisse (TOC, LOF, LOT)
- 4 Im Inhalt
  - Fußnoten, Randbemerkungen
  - Zitate
  - Verweise
  - Links
- 5 Nach dem Inhalt
  - Bibliografie
  - Code
  - Index
- 6 Alternative Klassen

- Schmutztitel
- Titel
- Verzeichnisse
- Gliederung
- Kopf-/Fußzeilen
- Fußnoten, Randbemerkungen
- Formeln
- Abbildungen, Tabellen etc.
- Verweise
- Programmcode
- Anhang
- Bibliografie
- Indizes

- Nachteil von  $\text{T}_{\text{E}}\text{X}$ : lange Dokumente werden unübersichtlich

- Nachteil von  $\text{T}_{\text{E}}\text{X}$ : lange Dokumente werden unübersichtlich
- Vorteil von  $\text{T}_{\text{E}}\text{X}$ : Teile des Dokumentes können in externe Dateien ausgelagert werden
- geschickte Aufteilung und Verwaltung eines Dokumentes möglich

- eine Hauptdatei als leeres Gerüst
- eine header-Datei (evtl. weitere Datei(en) für spezielle Befehlsdefinitionen)
- Inhalte in einem Unterordner
- Abbildungen und sonstige Materialien in weiteren Unterordnern



- `\input` und `\include` fügen externe Dateien am angegebenen Ort ein
- T<sub>E</sub>X „springt“ aus dem aktuellen Dokument, liest woanders, und springt wieder zurück

- `\input` und `\include` fügen externe Dateien am angegebenen Ort ein
- T<sub>E</sub>X „springt“ aus dem aktuellen Dokument, liest woanders, und springt wieder zurück
- T<sub>E</sub>X-Version: `\input` liest den Code einfach ein, als gehöre er ins Hauptdokument
- L<sup>A</sup>T<sub>E</sub>X-Version: `\include` erstellt eigene `.aux`-Datei (sinnvoll, wenn `.aux` benötigt)
- `\includeonly{a.tex,b.tex}` in der Präambel lässt nur die angegebenen Dateien für `\include` zu
- `\excludeonly{b.tex,c.tex}` lässt die angegebenen Dateien für `\include` *nicht* zu (benötigt Paket `excludeonly`)



- nach Aufteilung muss immer das Hauptdokument kompiliert werden
- ⇒ ständiges Wechseln zwischen Dokumenten

- nach Aufteilung muss immer das Hauptdokument kompiliert werden
- ⇒ ständiges Wechseln zwischen Dokumenten
- gute Editoren nehmen die Arbeit ab:
    - Definition von Hauptdokumenten möglich
    - Kompiliert automatisch das zugehörige Hauptdokument

- nach Aufteilung muss immer das Hauptdokument kompiliert werden
- ⇒ ständiges Wechseln zwischen Dokumenten
- gute Editoren nehmen die Arbeit ab:
    - Definition von Hauptdokumenten möglich
    - Kompiliert automatisch das zugehörige Hauptdokument

**T<sub>E</sub>Xworks** Setzen von *magic comments*:

**T<sub>E</sub>Xshop** %!<sub>E</sub>TEX<sub>E</sub>root<sub>E</sub>=<sub>E</sub>\(Hauptdokument)

**T<sub>E</sub>Xstudio**

```
% !TEX root = ../Masterarbeit.tex
% !TEX program = lualatex
% !TEX encoding = utf8
% !TEX spellcheck = de_DE
```

viele IDEs Festlegen einer „Projekt-Hauptdatei“

# Beispiel-Hauptdokument

```
\input{header}

\includeonly{chapter1}
\excludeonly{anhang} % erfordert Paket excludeonly!

\begin{document}
  \include{chapter1}
  \include{chapter2}
  ...
  \appendix
  \include{anhang}
\end{document}
```

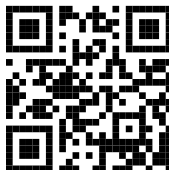
⇒ Nur chapter1 wird hier gesetzt, anhang explizit nie.

- Satzspiegel
- Schriften (Brottschrift, Überschriften)
- Formatierung von Formeln
- ...
- alles, was vor `\begin{document}` steht

# Titelei

- enthält alles bis zur ersten Inhaltsseite
- enthält Autor, Titel, etc.
- mit KOMA: Dokumentoption `titlepage=true/false` setzt eigene Seiten oder einen Titelkopf
- Umgebung `\begin{titlepage}` setzt eine frei gestaltbare Titelseite
- Befehl `\maketitle` setzt vordefinierte Titelei
- Angaben von `\title`, `\author`, `\extratitle` etc. nötig und möglich

In Overleaf ausprobieren:



<http://qn3.de/tex0701>

# Titelbefehle im KOMA-Bundle

```
\documentclass{scrbook}
\begin{document}
  \titlehead{\Large Universität Schlauenheim}
  \subject{Masterarbeit}
  \title{Risikomanagement in Zeiten von Social Media}
  \subtitle{Design interaktiver Apps für Banken und
    Versicherungen}
  \author{cand.\,stup. Uli Ungenau}
  \date{30. Februar 2017}
  \publishers{Betreut durch Prof.\,Dr.\,rer.\,stup. Naseweis}
  \dedication{Für meine Mama.}

  \maketitle
\end{document}
```

## `\maketitle` (in der Beamer-Klasse)

```
\title{Risikomanagement in Zeiten von Social Media}  
\subtitle{Design interaktiver Apps für Banken und  
  Versicherungen}  
\author{cand.\,stup. Uli Ungenau}  
\date{30. Februar 2017}  
  
\maketitle
```

# Risikomanagement in Zeiten von Social Media

## Design interaktiver Apps für Banken und Versicherungen

cand. stup. Uli Ungenau

30. Februar 2017



# abstract

- Umgebung abstract existiert für eine kurze Zusammenfassung des Dokuments
- mehrere Abstracts möglich (z. B. englisch / deutsch etc.)

```
\begin{abstract}
  Hier kommt eine kurze Zusammenfassung
  des Inhalts \dots
\end{abstract}

Und hier fängt das eigentlich Dokument
an
\dots
```

## Zusammenfassung

Hier kommt eine kurze  
Zusammenfassung des Inhalts ...

Und hier fängt das eigentlich Dokument an  
...

- Verzeichnisse fassen strukturierte Elemente zusammen
- prinzipiell kann alles in ein eigenes Verzeichnis aufgenommen werden
- übliche Verzeichnisse:
  - Inhaltsverzeichnis `\tableofcontents`
  - Abbildungsverzeichnis `\listoffigures`
  - Tabellenverzeichnis `\listoftables`
- Aufnahme der Verzeichnisse ins Inhaltsverzeichnis: Dokumentenoption  
`toc=toc`
- möglich: Codeverzeichnis, Beispielverzeichnis, ...

In Overleaf ausprobieren:



<http://qn3.de/tex0701>

zusätzlicher Text, der nicht ins Hauptdokument / in den Textfluss passt

- Fußnoten `\footnote{}`
- gleitende Randnotiz `\marginpar`
- Randbemerkung (Paket `marginnote`) `\marginnote`

Paket `footmisc` bietet vielfältige Möglichkeiten Aussehen von Fußnoten anzupassen

Es gibt eigene Umgebungen für Zitate:

- `quote` für kurze Zitate
- `quotation` für längere Zitate
- `verse` für Gedichte

Das Paket `csquotes` passt Feinheiten von Anführungszeichen für den nicht-englischen Satz an.

```
\begin{quote}  
  alea iacta est \hfill\textit{Caesar}  
\end{quote}
```


- Elemente können mittels `\label{}` bezeichnet werden
- mögliche Elemente sind Überschriften (sections etc.), `table`, `figure`, Formeln, ...
- Referenzierung mit `\ref{}`
- Pakete liefern vielfältige Referenzierungsmöglichkeiten:  
`fancyref`, `varioref`, `cleveref`
- geschicktes Benennen:  
`\label{fig:Haus}`  $\Rightarrow$  Pakete können erkennen, dass es eine Abbildung ist

# Links im Dokument

## hyperref

- Paket `hyperref` macht Verweise im PDF anklickbar
- `\ref` und `\cite` wird automatisch verlinkt
- URLs können mit `\url{\langle URL \rangle}` angegeben werden
- benannte Links mit `\href{\langle URL \rangle}{\langle angezeigter Text \rangle}`

```
\url{http://xkcd.com}\  
\href{mailto:mo@uni-hd.de}{\huge\Letter}
```

```
http://xkcd.com  

```


# Links im Dokument

## hyperref

- Paket `hyperref` macht Verweise im PDF anklickbar
- `\ref` und `\cite` wird automatisch verlinkt
- URLs können mit `\url{\langle URL \rangle}` angegeben werden
- benannte Links mit `\href{\langle URL \rangle}{\langle angezeigter Text \rangle}`

Um Probleme zu vermeiden `hyperref` eher als letztes Paket laden!

```
\url{http://xkcd.com}\  
\href{mailto:mo@uni-hd.de}{\huge\Letter}
```

```
http://xkcd.com  

```

- Befehl `\appendix` schaltet auf Anhang um
- Nummerierung startet neu  
(abhängig von Dokumentenklasse A, B, C, ...)
- Abschnitte im Anhang wie gewohnt mit `\chapter`, `\section`, etc.

```
\appendix
```



- Bibliografie enthält Liste verwendeter Quellen und ggf. weiterführende Literatur.
- je nach Fachbereich unterschiedliche Zitierstile
- (grobes) Aussehen der Bibliografie wird von Dokumentenklasse bestimmt.
- bestimmte Syntax zum Setzen der Bibliografie:
  - Umbelegung `\begin{thebibliography}{\langle Anzahl \rangle}`
  - Aufzählung der Werke mittels `\bibitem{\langle Key \rangle} \langle Text \rangle`
  - Zitieren eines Werks mit `\cite{\langle Key(s) \rangle}` oder `\cite[\langle Seite \rangle]{\langle Key \rangle}`

```
\begin{thebibliography}{9}  
  \bibitem{frankfurt05} Harry G. Frankfurt:  
    \textit{On Bullshit}, Princeton University Press,  
    Princeton, New Jersey, 2005.  
\end{thebibliography}
```

```
\begin{thebibliography}{9}  
  \bibitem{frankfurt05} Harry G. Frankfurt:  
    \textit{On Bullshit}, Princeton University Press,  
    Princeton, New Jersey, 2005.  
\end{thebibliography}
```

- manuelles Erstellen (und Sortieren) der Bibliografie ist sehr umständlich
- Einträge nicht sinnvoll wiederverwendbar

```
\begin{thebibliography}{9}  
  \bibitem{frankfurt05} Harry G. Frankfurt:  
    \textit{On Bullshit}, Princeton University Press,  
    Princeton, New Jersey, 2005.  
\end{thebibliography}
```

- manuelles Erstellen (und Sortieren) der Bibliografie ist sehr umständlich
  - Einträge nicht sinnvoll wiederverwendbar
- ⇒ Programm Bib<sub>T</sub>E<sub>X</sub> übernimmt Sortierung und Verwaltung der Einträge (siehe Vorlesung nach den Ferien)

- für kurze Sequenzen: `\verb~\befehl~`
- für längere Sequenzen: `\begin{verbatim} \befehle \end{verbatim}`
- beide bieten \*-Version für Anzeigen von Leerzeichen:
- Paket `listings` kann rudimentäre Syntaxhervorhebung für viele Programmiersprachen
- Paket `minted` nutzt externen Parser für komplexe Syntaxhervorhebung
- für Setzen von  $\LaTeX$ -Beispielcode: Paket `showexpl`

- Indexerstellung ist immens aufwändiges Unterfangen:
- sämtliche (sinnvollen!) Erscheinungen von Namen / Ereignissen / Sachthemen müssen registriert werden  
nicht jede Nennung eines Namens soll im Index erwähnt werden!
- sinnvolle Seitenangabe: 1, 2–4, 17

- dank logischer Struktur leichte Erstellung in  $\text{T}_{\text{E}}\text{X}$ :
- Definieren von Befehlen erleichtert die Eingabe:  
`\euler` statt `Euler` `\index{Euler}`
- mit  $\text{\LaTeX}$  dreistufiger Prozess:
  - im  $\text{\LaTeX}$ -Lauf wird Hilfsdatei erstellt
  - Verarbeitung mittels Programm `makeindex` (Sortierung, Seitenangaben etc.)  
moderne Variante: `xindy`
  - Einbettung im nächsten  $\text{\LaTeX}$ -Lauf

## im Dokument

---

```
\usepackage{makeidx}
\makeindex %% VOR \begin{document}!!

\index{Stichwort} %% IM Dokument!

\printindex %% druckt das Verzeichnis hier
```

---

## in der Kommandozeile

Aufruf von  
\$ makeindex hauptdocument  
im Ordner des Hauptdokumentes



`multind` ermöglicht Erstellung mehrerer Indizes – Unterscheidung mit zusätzlichem Attribut:

### im Dokument

```
\usepackage{multind}
\makeindex{stichwoerter}
\makeindex{Personen}
\index{Stichwoerter}{Stichwort}
\index{Personen}{Euler}
\printindex{stichwoerter}{Index der Stichwörter}
\printindex{personen}{Personenverzeichnis}
```

### in der Kommandozeile

```
$ makeindex personen
$ makeindex stichwoerter
```

- Paket `xeindex` verwendet XeTeX-Interns, um automatisch Indizes zu erstellen
- `xesearch` durchsucht dabei (mittels XeTeX-Befehlen!) selbst das Dokument
- gefundene Einträge werden Indiziert

- Paket `xeindex` verwendet  $X_{\text{E}}\text{TeX}$ -Interns, um automatisch Indizes zu erstellen
  - `xesearch` durchsucht dabei (mittels  $X_{\text{E}}\text{TeX}$ -Befehlen!) selbst das Dokument
  - gefundene Einträge werden Indiziert
- + extrem leichtes Erstellen von Indizes beliebiger Größe
- Sinnhaftigkeit fragwürdig – nicht jede Nennung eines Begriffes sollte indiziert werden, sonst ist der Index nutzlos.  
Der Leser sollte nur die wichtigsten Einträge finden.

- verwendet intern `makeidx`, daher sind `\makeindex`, `\printindex` und `\index` weiter verfügbar
- zu suchende Einträge:

## IndexList

```
\IndexList * { name }{ list of entries }
```

`*` ⇒ case insensitive

`name` ⇒ beliebiger Name für die Liste (mehrere möglich)

`list of entries` ⇒ katze, hund?, maus

`hund?` ⇒ findet auch hundehütte

- Klasse `memoir` bietet gute Alternative für anspruchsvollen Buchsatz.
  - viele vordefinierte Stile.
  - extrem anpassbar durch viele Optionen
- Klasse `classicthesis` imitiert den Stil von Robert Bringhursts „The Elements of Typographic Style“ für Abschlussarbeiten.
  - Hilfreiche Vorlagen erleichtern den Einstieg.
  - typografisch sehr ansprechend

