

Einführung in wissenschaftliches Schreiben: 2 L^AT_EX

Éva Mújdricza-Maydt
mujdricza@cl.uni-heidelberg.de

Institut für Computerlinguistik, Universität Heidelberg

Sommersemester 2017

Vielen Dank an alle Doktoranden, die mir freundlicherweise Materialien für diese Folien zur Verfügung gestellt haben.

Teil I

Grundlagen

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

TEX

ist ein Textsatzsystem zur Erzeugung von typographisch hochwertigen Dokumenten, hauptsächlich im naturwissenschaftlichen Bereich

LATEX

ist eine Sammlung von arbeitserleichternden Makros für TEX

Prinzipien

Unterschiede zu Textverarbeitungsprogrammen wie beispielsweise Microsoft Word:

- \LaTeX ist eine Auszeichnungssprache wie HTML (*markup language*), keine Programmoberfläche \rightarrow kein WYSIWYG
- \TeX - bzw. \LaTeX -Dateien sind *plain text*, das heißt sie bestehen lediglich aus Buchstaben, Zahlen und Sonderzeichen und können mit jedem Texteditor geschrieben werden, wobei es einige gibt, die speziell auf \LaTeX ausgerichtet sind (z.B. TeXstudio, TeXShop)
- Aus $.\text{tex}$ -Dateien können jedoch eine Vielzahl von Formaten generiert werden: PS, PDF, HTML, DVI, TXT, ...

Motivation: Warum \LaTeX (nicht) verwenden?

Einige der Nachteile von \LaTeX

- geringere Benutzerfreundlichkeit, steilere Lernkurve
- Layouts lassen sich nur geringfügig variieren, größere Abweichungen sind nur mit viel Aufwand möglich (Definition einer eigenen Dokumentklasse)

Motivation: Warum \LaTeX (nicht) verwenden?

Einige der Vorteile von \LaTeX

- Produziert standardmäßig typographisch hochwertige Dokumente, unter anderem Artikel, Briefe, Bücher...
- Querverweise, Nummerierungen, Inhaltsverzeichnis, Bibliographie etc. können automatisch erzeugt und aktualisiert werden
- \LaTeX -Dokumente sind plattformunabhängig
- Auch lange, aufwändige Dokumente lassen sich problemlos und mit vergleichsweise geringem Ressourcenverbrauch bewältigen
- \LaTeX ist kostenlos, hat exzellente Dokumentation
- ...

Grundgerüst einer L^AT_EX-Datei

1. `\documentclass{article}`
2. `\usepackage[ngerman]{babel}`
3. `\title{Die Henne und das Ei}`
4. `\author{Charles Dickens}`
5. `\date{23. August 2010}`

6. `\begin{document}`
7. `\maketitle`
8. `[...]`
9. `\end{document}`

- 1 – 5 Die sogenannte **Präambel**, Definition der **Dokumentklasse**, Angabe von Metadaten und Einbindung von **Paketen**
- 7 – 8 Hier wird der eigentliche Inhalt des Dokuments geschrieben
- 6, 9 Definiert eine **Umgebung**, hier des Dokuments `document`

L^AT_EX-Zeichen

Textzeichen

a...z A...Z 0...9

. : ; , ? ! ' ' () [] - / * @ + =

Steuerzeichen

\$ & % # _ { } ~ ^ " \ | < >

Beispiel

```
\usepackage[ngerman]{babel}
```

- L^AT_EX-Befehle beginnen mit einem *backslash* \
- worauf der Name des Befehls folgt, hier `usepackage`
- optionale Parameter werden in eckigen Klammern [] übergeben, hier `ngerman`
- notwendige Parameter in geschweiften Klammern {}, hier `babel`

Dokumentklasse

```
\documentclass{...}
```

- die Dokumentklasse kennzeichnet das Format für das gesamte Dokument¹
- Typische Klassen: book, report, article, letter, beamer, ...
- Typische Optionen: a4paper, 10pt, 11pt, 12pt, oneside, twoside, landscape ...
- Beispiel:

```
\documentclass[a4paper,11pt,landscape]{article}
```

¹diese Folien wurden mit der Dokumentklasse beamer erstellt

Dokumentklassen

- Die Dokumentklassen `book`, `article` und `report` sind im Hinblick auf US-amerikanischen Textsatzkonventionen und Papierformaten entstanden
- Ein flexibleres, alternatives Layout bieten die KOMA-Script-Klassen `scrartcl`, `scrreprt`, `scrbook`

Präambel

```
\documentclass{...}  
\usepackage[ngerman]{babel}  
\title{Die Henne und das Ei}  
\author{Charles Chickens}  
\date{19. April 2017}  
\begin{document}  
\maketitle
```

- die Präambel ist der Bereich zwischen der Angabe der Dokumentklasse und der Dokumentumgebung
- es können Erweiterungspakete geladen werden (`\usepackage`)
- man kann Titel, Autor und Datum des Dokuments festlegen (mit `\maketitle` lässt sich daraus eine Titelseite erzeugen)
- ...

demo - ein erstes L^AT_EX-Dokument

- 1 Einführung
- 2 Formatierungen**
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

Textformatierung

Formatierungen wie Hervorheben oder Unterstreichen werden in \LaTeX durch Befehle vorgenommen

- `\textbf{fetter Text}` **fetter Text**
- `\textit{kursiver Text}` *kursiver Text* (italics)
- `\texttt{feste Laufweite}` feste Laufweite (typewriter)
- `\underline{unterstrichen}` unterstrichen

Schriftgrößen

`\Huge`

`\huge`

`\LARGE`

`\Large`

`\large`

`\normalsize`

`\small`

`\footnotesize`

`\scriptsize`

`\tiny`

In \LaTeX gibt es verschiedene Arten von Befehlen:

- **Befehl + Option**,
z. B. `\textbf{fetter Text}` oder
- **Schalter**, z. B. `\Large`. Schalter haben so lange Gültigkeit bis sie wieder aufgehoben werden, hier durch `\normalsize`.

Hervorhebungen und Fußnoten

- *Emphatischer Text*² lässt sich mittels `\emph{Emphatischer Text}` erzeugen
- Eine Fußnote erzeugt man mittels `\footnote{d.h. hervorgehobener Text}`

²d.h. hervorgehobener Text

Sonderzeichen

- Steuerzeichen von \LaTeX müssen umgangen werden, wenn sie als Zeichen im Text vorkommen sollen
 - Die meisten Zeichen werden mit \backslash vor dem Zeichen geschrieben, z.B. $\&$, $\$$, $\%$ \rightarrow $\&$, $\$$, $\%$
 - Manchmal hat auch \backslash Zeichen eine Sonderbedeutung, z.B. \backslash , hier brauchen wir den Befehl \backslashtextbackslash
- Für Sonderzeichen, die nicht im Inventar von \LaTeX sind, werden auch Befehle verwendet, z.B. \backslashtexteuro für $\text{\text{€}}$ ³
- eine Übersicht gibt es hier: http://en.wikibooks.org/wiki/LaTeX/Special_Characters

³erfordert $\backslash\text{usepackage}\{\text{textcomp}\}$

Abstände und Umbrüche

- Der Abstand nach „.“ wird automatisch vergrößert. Das kann mit `.~` oder `.\` abgestellt werden. `~` verhindert zusätzlich einen Zeilenumbruch
- Einen manuellen Zeilenumbruch erzeugt man mit `\\`
- Größerer Abstand zwischen Zeilen: `\bigskip`
- Die Befehle `\hspace{<Länge>}` und `\vspace{<Höhe>}` erzeugen horizontalen und vertikalen Abstand.
Beispielangaben: `0.5cm` oder `1in`

Es empfiehlt sich, sparsam mit eigenen Abstandsdefinitionen umzugehen, da \LaTeX (meistens) von selbst gut leserlichen und stimmig angeordneten Output erzeugt.

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen**
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

Dokumentumgebung

```
\begin{document}
```

Der eigentliche Inhalt eines Dokuments steht in der Dokumentumgebung. Beliebig viele Leerzeichen werden als `\emph{ein}` Leerzeichen behandelt.

Einen neuen Absatz erhält man mittels einer Leerzeile. Kommentare beginnen mit einem % blöden % oder besser: einem lehrreichen Beispiel Prozentzeichen (`\%`).

```
\end{document}
```

Der eigentliche Inhalt eines Dokuments steht in der Dokumentumgebung. Beliebig viele Leerzeichen werden als *ein* Leerzeichen behandelt.

Einen neuen Absatz erhält man mittels einer Leerzeile. Kommentare beginnen mit einem Prozentzeichen (%).

L^AT_EX-Umgebungen

Beispiel

```
\begin{itemize}
\item \LaTeX-Umgebungen definieren Bereiche, in denen
    besondere Regeln für die Formatierung gelten
\item ...
\end{itemize}
```

- L^AT_EX-Umgebungen definieren Bereiche, in denen besondere Regeln für die Formatierung gelten
- eine Umgebung beginnt mit `\begin{umgebungsname}` und endet mit `\end{umgebungsname}` (hier: `itemize`)
- in manchen Umgebungen stehen zusätzliche Befehle zur Verfügung (hier: `\item`)

Wichtige Umgebungen – Listen

① nummerierte Listen

② Aufzählungslisten

③ Beschreibungslisten

- nummerierte Listen

- Aufzählungslisten

- Beschreibungslisten

`enumerate` nummerierte Listen

`itemize` Aufzählungslisten

`description` Beschreibungslisten

```
\begin{enumerate}
\item nummerierte Listen
\item Aufzählungslisten
\item Beschreibungslisten
\end{enumerate}
```

```
\begin{itemize}
\item nummerierte Listen
\item Aufzählungslisten
\item Beschreibungslisten
\end{itemize}
```

```
\begin{description}
\item[enumerate] nummerierte Listen
\item[itemize] Aufzählungslisten
\item[description] Beschreibungslisten
\end{description}
```

Wichtige Umgebungen – Tabellen

```
\begin{table}
  \begin{tabular}{|r|l|}
    \hline
    7C0 & hexadezimal \\
    3700 & oktal \\
    11111000000 & bin\"ar \\
    \hline\hline
    1984 & dezimal \\
    \hline
  \end{tabular}
\caption{Neunzehnhundertvierundachtzig}
\end{table}
```

7C0	hexadezimal
3700	oktal
11111000000	binär
1984	dezimal

Tabelle: Neunzehnhundertvierundachtzig

Wichtige Umgebungen – Abbildungen

```
\usepackage{graphicx} % in der Präambel
```

```
\begin{figure}
```

```
\includegraphics[scale=0.15]{fig/Buchdrucker-1568.png}
```

```
\caption{Buchdruck im 16. Jahrhundert}
```

```
\end{figure}
```

Abbildung: Schematischer L^AT_EX-Code für untere Abbildung



Abbildung: Buchdruck im 16. Jahrhundert

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus**
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

Mathematikmodus I

- Um mathematische Formeln zu setzen, muss man in den **Mathemodus** wechseln
- Im Mathemodus funktioniert vieles anders, und manche Zeichen haben eine andere Bedeutung
- Mathemodus wird mit `$` und `$` oder `\begin{math}` und `\end{math}` begrenzt
- Formeln bzw. Gleichungen werden auf eigene Zeilen gesetzt mit `\begin{displaymath}` bzw. `\begin{equation}`

$$a^2 + b^2 = c^2$$

$$a^2 + b^2 = c^2$$

$$c = \pm\sqrt{a^2 + b^2}$$

$$c = \pm\sqrt{a^2 + b^2}$$

$$\sum_{i=1}^n i = \frac{n \cdot (n+1)}{2}$$

$$\sum_{i=1}^n i = \frac{n \cdot (n+1)}{2}$$

Mathematikmodus II

L^AT_EX bietet enorm viele Möglichkeiten Mathematik aufzuschreiben. . .

Eine graphische Übersicht gibt es auch hier:

<http://www.codecogs.com/latex/eqneditor.php>

$$\sum_1^n, \prod_1^n, \bigcup_{a \in A}$$

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, M = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

$$y = \frac{a}{b+1} = \frac{a}{b+1}$$

$$f(x) = \int_1^\infty \frac{1}{x^2} dx = 1$$

a)	$y =$	c	$(konstant)$	} Polynome
b)	$y =$	$cx + d$	$(linear)$	
c)	$y =$	$bx^2 + cx + d$	$(quadr.)$	
d)	$y =$	$ax^3 + bx^2 + cx + d$	$(kubisch)$	

Mathemodus – Demo

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung**
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

Kapitel und Überschriften

- Welche Typen von Überschriften es gibt, hängt von der Dokumentklasse ab
- bei `article` sind das:
 - `\section{...}`
 - `\subsection{...}`
 - `\subsubsection{...}`
- Der Befehl `\tableofcontents` erzeugt automatisch ein Inhaltsverzeichnis

Abbildungen & Tabellen

- Abbildungen und Tabellen können wie auch `equation` automatisch mit einer Nummerierung versehen werden, hierfür dienen die Umgebungen `figure` bzw. `table`
- Diesen Umgebungen kann man mittels `\caption{...}` eine Bild- oder eine Tabellenbeschreibung hinzufügen (sie erscheint unterhalb der Abbildung/Tabelle)
- Abbildungs- und Tabellenverzeichnis können mit `\listoffigures` bzw. `\listoftables` erzeugt werden

Floating-Umgebungen

- Bei `table` und `figure` handelt es sich um *floating*-Umgebungen
- sie werden von \LaTeX automatisch positioniert, erscheinen also nicht notwendigerweise an der Stelle, an der sie im Quelltext stehen
- Man kann dieses Verhalten jedoch beeinflussen, indem man am Ende des jeweiligen `begin`-Befehls eine oder mehrere der folgenden Optionen anhängt:
 - `h` here (Selbe Position wie im Quellcode)
 - `t` top (Anfang der nächsten Seite)
 - `b` bottom (am Seitenende)
 - `p` page (auf eigener Seite)

Biespiel:

```
\begin{figure}[h]
```

Querverweise

- Man kann verschiedenen Umgebungen ein `label` hinzufügen, mit dem diese Umgebung referenziert werden kann

```
\begin{figure}
  \includegraphics{bilddatei}
  \caption{Ein schönes Bild}
  \label{abb:schoenesb}
```

```
\end{figure}
```

```
[...]
```

```
Abbildung \ref{abb:schoenesb} ist ein schönes Bild.
```

- Das Label darf erst nach der `\caption` stehen, da es sich sonst nicht auf die Abbinungsnummer bezieht.

Strukturierung – Demo

Strukturierung der Datei

Manchmal ist es sinnvoll die `.tex`-Datei in mehrere Dateien aufzuteilen, hierfür sieht \LaTeX zwei Befehle vor:

- `\input{datei}`, Inhalt von `datei` wird einfach in die aufrufende Datei übernommen
- `\include{datei}`, wie oben nur mit einem davor eingefügten Seitenumbruch

```
\section{Teil 1}  
\input{teil1}
```

```
\section{Teil 2}  
\input{teil2}
```

Abbildung: Strukturierung mit `input`

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete**
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I

Wichtige Pakete

`\usepackage [option] {paketname}`

`babel` Sprachspezifische Einstellungen (Optionen: `german`, `ngerman`, ...)

`inputenc` Zeichenkodierung (Optionen: `utf8`, `latin1`, ...)

`url` URLs und Links

`graphicx` Einbinden von Bilddateien (`jpg`, `png`, `pdf`, ...)

`tikz` Grafiken (z.B. Bäume, Diagramme, etc.)

`listings` Programmcode

`AVM` Attribut-Wert-Matrizen

`Covington` (linguistische) Beispiele

`tree-dvips` Bäume, Linien, etc.

Multilinguales L^AT_EX

L^AT_EX ist ursprünglich auf amerikanisches Englisch ausgerichtet. Folgende Probleme ergeben sich, wenn man Text in anderen Sprachen schreiben will:

- ① Umgang mit unterschiedlichen orthographischen Regeln, z.B. Silbentrennung am Zeilenende
- ② Umgang mit anderen typographischen Konventionen (z.B. Anführungszeichen)
- ③ Kapitelüberschriften etc. in der Sprache des Dokuments
- ④ Sprachliche Zeichen, die nicht im Standardzeichensatz von L^AT_EX enthalten sind (z.B. Umlaute)

Für diese Probleme wurde das Paket `babel` entwickelt

Das Paket babel

- babel bietet Sprachunterstützung für viele, hauptsächlich europäische Sprachen (siehe Liste in Dokumentation)
- Das Paket lädt automatisch die Trennungsregeln für die gewünschte Sprache und erzeugt entsprechende Kapitelüberschriften
- Einbinden des Pakets: z.B. `\usepackage[ngerman]{babel}` für neue deutsche Rechtschreibung (`german` für alte Rechtschreibung)
- In der Präambel können auch mehrere Sprachen angegeben werden. Umschalten kann man mit `\selectlanguage{language}`

Umlaute und Anführungszeichen

Anführungszeichen Deutsche Anführungszeichen („ “) können mit `\glqq \grqq` oder mit `"‘ ”’` erzeugt werden.

Umlaute Sehr umständliche Eingabe ohne `babel` :

`\"a` → ä

Umlaute mit `babel`:

`"a` → ä

scharfes s Unter `babel` mit `\ss`

Kodierungen I

Die Kodierung betrifft sowohl die Eingabe- als auch die Ausgabedatei.

1. Eingabekodierung

- `\usepackage[<encoding>]{inputenc}`: gibt die Kodierung der `.tex`-Datei an.
- Wird, wie hier, `utf8` als Kodierung verwendet, können auch im Quelltext Sonderzeichen (z.B. Umlaute) verwendet werden
- Das ist komfortabler, funktioniert aber nur, wenn die Kodierung vom System unterstützt wird

Kodierungen II

2. Ausgabekodierung

- `\usepackage[T1]{fontenc}` gibt die Kodierung des Ausgabefonts an
- T1 lädt beispielsweise Schriften, die über die verwendeten Umlaute/Akzente für die meisten europäischen Sprachen verfügen.
- Für andere Schriftsysteme müssen andere Kodierungen geladen werden

Quelltext I

- Text in fester Laufweite lässt sich mit `\texttt{...}` oder `{\tt ...}` erzeugen
- Innerhalb der `verbatim`-Umgebung werden alle Zeichen wörtlich interpretiert (auch \LaTeX -Steuerzeichen). Sie kann auf 2 Arten aufgerufen werden
 - ① `\verb|...|` wenn der Text innerhalb einer Zeile steht
 - ② `\begin{verbatim} ...`
- Eine besonders schöne Möglichkeit bietet das Paket `listings`, damit sind u.A. automatische Formatierung und Zeilennummerierungen möglich

Paket listings

```
1  /* Ein Kommentar */
2  #include <iostream>
3
4  int main (int argc, char * const argv[]) {
5      // und noch einer!
6      std::cout << "Hello, \uWorld!\n";
7      return 0;
8  }
```

Die Darstellung kann mittels `\lstset{}` sehr fein eingestellt werden, das Paket bietet auch vordefinierte Einstellungen für verschiedene Programmiersprachen (im Beispiel C++)

Weitere Pakete finden

- Die meisten Pakete gibt es über das CTAN⁴
<http://www.ctan.org>
- latex4ling listet eine Reihe von Paketen, die man als Linguist braucht
<http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/>
- ams-latex bietet viele Erweiterungen zum Setzen mathematischer Texte
<http://www.ams.org/tex/amslatex.html>

⁴the Comprehensive T_EX Archive Network

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX**
- 8 Zusammenfassung Teil I

Von .tex zu .pdf

Um aus einer \LaTeX -Datei ein druckbares Dokument zu erzeugen, gibt es im Wesentlichen zwei Möglichkeiten:

Kommandozeile:

- `pdflatex meinedatei.tex`
→ `meinedatei.pdf`

spezieller Editor oder Plugin:

- TeXstudio
- Texmaker
- Eclipse using TeXlipse
- ...

Wo bekommen wir diese Programme her? Das ist je nach Betriebssystem sehr unterschiedlich:

- Windows: MiKTeX, <http://miktex.org/>
- Mac: MacTeX, <http://www.tug.org/mactex/>
- Linux: TeX Live⁵, <http://tug.org/texlive/>

Ist eine dieser Distributionen installiert, stehen auf der Kommandozeile die vorher genannten Befehle zur Verfügung bzw. fähige Editoren können benutzt werden.

⁵Am besten über eine Paketverwaltung installieren

Fehlermeldungen

Beim Übersetzen von `.tex`-Dateien gibt \LaTeX oft Fehlermeldungen aus. Folgende Fehlermeldungen können u.a. auftreten:

`warnings` z.B. Fehlende Fonts, oft kein gravierendes Problem

`badboxes` z.B. `Overfull \hbox`, zeigt Probleme bei Zeilenumbrüchen an

`errors` Beginnen mit `!`, treten auf, wenn \LaTeX etwas nicht interpretieren oder ausführen kann

\LaTeX generiert nicht nur das Ausgabe-PDF, sondern auch eine `.log`-Datei, in der steht, welche Schritte ausgeführt wurden und wo Fehlermeldungen aufgetreten sind.

Fehlermeldungen

Trennungsfehler

- Manchmal hat \LaTeX Probleme mit der Trennung von Wörtern, was zu badboxes führen kann
- Trennungsvorgaben lassen sich auch manuell machen:
- `\-` in einem Wort markiert mögliche Trennungspunkte
`Do\nau\dampf\schiff\fahrts...`
- In der Präambel können Trennungsvorgaben mit `\hyphenation{}` angegeben werden
`\hyphenation{Do-nau-dampf-schiff-fahrts...}`

Fehlermeldungen

Fehlermeldungen verstehen

- Errors führen dazu, dass \LaTeX die Fehlerbenachrichtigung anzeigt und den Übersetzungsprozess anhält. Es wird keine Ausgabedatei generiert.
- Oft enthält die Fehlermeldung eine Zeilenangabe, die helfen kann, den Fehler zu finden
- Häufig auftretende Fehler sind:
 - Tippfehler
 - Fehlende `}` oder unvollständige Umgebung
 - Fehlender Mathemodus
 - Paket nicht geladen oder nicht installiert

Umgang mit Fehlermeldungen

Fehlermeldungen vermeiden/Troubleshooting

- Highlight-Modus des Editors verwenden, dieser zeigt z.B. fehlende Klammern
- Viele Editoren bieten automatische Vervollständigung an
- Große Dateien vermeiden, lieber ein Master-Dokument verwenden und einzelne Kapitel mit `\input` einfügen
- `pdflatex` während des Schreibprozesses oft laufen lassen, das vermeidet langwierige Fehlersuche am Ende

- 1 Einführung
- 2 Formatierungen
- 3 Umgebungen
- 4 Mathematikmodus
- 5 Strukturierung
- 6 Spezielle Pakete
- 7 Arbeiten mit \LaTeX
- 8 Zusammenfassung Teil I**

L^AT_EX – Zusammenfassung I

- L^AT_EX ist eine Auszeichnungssprache für das Textsatzsystem T_EX
- L^AT_EX-Dateien sind *plain text* und werden als .tex-Dateien abgespeichert. Daraus können dann verschiedene Formate generiert werden
- In der ersten Zeile eines L^AT_EX-Dokuments muss eine Dokumentklasse angegeben werden
- zusätzliche Pakete, Metadaten sowie neu definierte Befehle können in der Präambel angegeben werden
- Der Inhalt des Dokuments befindet sich in der Dokumentumgebung
- Textformatierung erfolgt mit Hilfe von Befehlen
- Für besonders formatierte Bereiche (Tabellen, Listen etc.) verwendet man weitere Umgebungen innerhalb des Dokuments

L^AT_EX – Zusammenfassung II

- Mathematische Formeln erfordern den Mathemodus
- Jede Dokumentklasse stellt eine bestimmte Kapitel-/Abschnittstruktur zur Verfügung, anhand dieser Struktur kann automatisch ein Inhaltsverzeichnis generiert werden
- Gibt man einem Abschnitt, einer Tabelle oder einer Abbildung ein `\label{}`, so kann an anderen Stellen mit `\ref{}` wieder darauf verwiesen werden
- zusätzliche Pakete bieten zusätzliche Optionen. Sie werden in der Präambel mit `\usepackage[options]{Paketname}` geladen.
- Pakete kann man von CTAN herunterladen

Teil II

Literaturverwaltung und Präsentationen

9 Bibliographie mit $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$

10 Präsentationen mit beamer

11 Zusammenfassung Teil II

Bibliographie

- Eine Bibliographie listet, für gewöhnlich am Ende des Dokuments, die zitierte Literatur auf
- In \LaTeX wird mittels dem Befehl `\cite{key}` zitiert⁶
- Um eine Bibliographie zu erstellen gibt es nun zwei Möglichkeiten
 - ① Mit der Umgebung `bibliography`, Einträge müssen manuell angelegt und formatiert werden
 - ② Oder mit `BIB \TeX`

⁶Dieser bietet auch ein paar Optionen bzw. Varianten

- BIB_TE_X verkompliziert die Erstellung der Ausgabedatei ein wenig, vereinfacht aber den Umgang mit Literatur ungemein
- Angaben zur Literatur werden in eine „Datenbank“ eingetragen, hier ein Eintrag:

```
@mastersthesis{Frank:90,          CITE-KEY
  author = {Anette Frank},      ATTRIBUTE VALUE
  title  = {Eine LFG-Analyse zur Kongruenz des ...},
  school = {Stuttgart University},
  year   = {1990}
}
```

```
@type{cite-key,  
  attribute = value,  
  ...  
}
```

TYPE: Typ der Publikation, dies bestimmt die Angaben, die zu einem Werk gemacht werden können/müssen

CITE-KEY: Zeichenkette⁷, über die mit dem Befehl `\cite{CITE-KEY}` zugegriffen werden kann, er muss *eindeutig* sein

ATTRIBUTE VALUE: *Attribut-Wert-Paare*, die das zitierte Dokument beschreiben und aus denen der Eintrag generiert wird, Beispiele: `author`, `year`, `title`, ...

Werte werden mit `"` oder `{}` eingeschlossen und können mit \LaTeX -Befehlen⁸ formatiert werden

⁷will heißen: beliebige Abfolge von Zeichen (aber ohne `{` und `!`)

⁸auch Mathemodus

BIB_TE_X – Typen von Literatur

Es gibt sehr viele verschiedene Arten von Publikationen, diese werden auch unterschiedlich zitiert und benötigen teilweise andere Angaben, hier eine kleine Auswahl:

`article` author, title, journal, year

`book` author/editor, title, publisher, year

`incollection` author, title, booktitle, year

`inproceedings` author, title, booktitle, year

`thesis` author, title, school, year

BIB_TE_X – Zitier- und Bibliographiestile

- Es gibt viele Arten wie die Einträge und die Zitationen formatiert werden können, *Standard*: Zitationen numerisch [1], [2], [3], ...
- Das Format der Literaturverweise und der Bibliographie wird durch ein Stylefile definiert. Viele Stylefiles (Endung .bst) werden standardmäßig mit der L^AT_EX-Distribution mitgeliefert
- Das gewünschte Stylefile wird mit dem Befehl `\bibliographystyle{}` aufgerufen.
- Es MUSS immer ein Stylefile angegeben werden
- Man kann auch eigene Stylefiles definieren

- Mit dem Paket `natbib` bekommt man per default Literaturverweise im Format *Autor, Jahr*
- `natbib` einbinden

```
\usepackage[round]{natbib}
\bibliographystyle{plainnat}
```

- In `natbib` wird nicht der BIB_TE_X Befehl `\cite` verwendet, sondern z. B.

`\citep` Autor und Jahr in Klammern: (Frank, 1994)

`\citet` Nur das Jahr in Klammern: Goecke (1997)

- Es bietet zudem noch viele weitere Zitationsmöglichkeiten und Formatierungsoptionen an (siehe Dokumentation)

BIB_TE_X – Technisches

- Die Literaturdatenbank muss in einer `.bib`-Datei angelegt werden
- Im \LaTeX Dokument wird das Literaturverzeichnis mit dem Befehl `\bibliography{Name der .bib-Datei}` aufgerufen
- Tatsächlich mit `\cite` zitierte Einträge landen im Literaturverzeichnis⁹
- `\bibliographystyle{}` legt den Stil fest, wird innerhalb des Dokuments aufgerufen
- `natbib` sorgt für mehr Optionen
- Gesetzt wird ein Dokument nun so:

```
pdflatex dokument; bibtex dokument; pdflatex dokument;  
pdflatex dokument
```

⁹Wird vor dem Aufrufen des Literaturverzeichnisses der Befehl `\nocite{*}` eingefügt, dann werden alle Einträge aus der Literaturdatenbank im Literaturverzeichnis angegeben

BIB_TE_X – Demo

- Einträge in der Literaturdatenbank manuell zu erstellen kann zäh und fehleranfällig sein
- Viele Online-Ressourcen bieten die Möglichkeit, bibliographische Hinweise im BIB_TE_X-Format zu exportieren
- z. B. HEIDI (Online Katalog der UB), ACM Digital Library, Google Scholar¹⁰
- Eine weitere Hilfe zur Literaturverwaltung bieten BIB_TE_X Editoren an, auch hier gibt es verschiedene Optionen, zum Beispiel:
 - JabRef
 - KBibTex
 - BibDesk

¹⁰Anzeige im BIB_TE_X-Format muss erst unter Scholar-Einstellungen aktiviert werden.

9 Bibliographie mit $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$

10 Präsentationen mit beamer

11 Zusammenfassung Teil II

Folien mit \LaTeX /beamer

- Die beamer-Klasse dient dazu Folienpräsentationen in \LaTeX zu erstellen
- Bei neueren \LaTeX Distributionen ist sie standardmäßig dabei
- sonst kann das Paket auch unter <https://bitbucket.org/rivanvx/beamer/wiki/Home> heruntergeladen werden
- Eine gute Dokumentation findet sich im mitgelieferten Beamer User Guide

Ein paar Hinweise zu Präsentationen

- Daumenregel für die Zeitabschätzung: Nicht mehr als eine Folie pro Minute einplanen / 2 Minuten pro Folie
- Auch Präsentationen brauchen eine **sinnvolle Gliederung**
- Einzelne Folien sollten eine **gedankliche Einheit** bilden
- Folien möglichst **übersichtlich** gestalten
 - wichtige Informationen hervorheben (Fettdruck, Farbe)
 - kleine Fonts, Fußnoten und komplizierte Formeln vermeiden
 - kurze Sätze verwenden
- **Fließtext vermeiden**, möglichst oft Stichpunkte verwenden
- Visualisierungen verwenden, wenn möglich

Titelseite

- Metadaten für die Titelseite werden in der Präambel angegeben
- Zusätzlich zu `author`, `date`, `title` gibt es auch `institute`, `subtitle` und weitere
- Als Optionen zu `author`, `title` etc. kann man Kurzversionen der jeweiligen Angabe definieren. Diese erscheinen z.B. im Header/Footer

```
\author[JD]{John Doe}
```

Struktur

- Gliederung mit `\section` etc. wie bei anderen \LaTeX -Dokumenten
- Die Gliederung gibt nur die inhaltliche Struktur der Präsentation wieder
- Aufteilung in einzelne Folien erfolgt mit der `frame`-Umgebung

```
\begin{frame}{Folientitel}  
Folieninhalt  
\end{frame}
```

Inhaltsverzeichnis

- Das Inhaltsverzeichnis wird wie gehabt mit `\tableofcontents` aufgerufen
- Fügt man den folgenden Code in die Präambel ein, so wird vor jeder section der entspr. Gliederungspunkt angezeigt

```
\AtBeginSection[]{\begin{frame}\tableofcontents[currentsection]\end{frame}}
```

Overlays

Folieninhalte können nach und nach hinzugefügt werden, z.B. in Listenumgebungen

- einfache Möglichkeit: `\pause` bewirkt, dass der nachfolgende Text erst durch nochmaliges drücken der Pfeiltaste angezeigt wird
- Overlay Spezifikationen bieten noch komplexere Möglichkeiten
- Diese Spezifikationen werden in `<>` hinter einem Befehl angegeben

Overlay Spezifikationen - Beispiel

```
\textbf<1->{Immer Fett}  
\textbf<2>{nur auf der 2.  
Folie fett}  
\textbf<3->{ab der  
dritten Folie fett}
```

Abbildung: Overlay Spezifikationen

Immer Fett

nur auf der 2. Folie fett
ab der dritten Folie fett

```
\only<4>{erscheint nur auf  
der 4. Folie}  
\uncover<5>{ab der 5. Folie}  
\uncover<6>{ab der 6. Folie}
```

Abbildung: Spezielle Overlay-Befehle

Overlay Spezifikationen - Beispiel

```
\textbf<1->{Immer Fett}  
\textbf<2>{nur auf der 2.  
Folie fett}  
\textbf<3->{ab der  
dritten Folie fett}
```

Abbildung: Overlay Spezifikationen

Immer Fett
nur auf der 2. Folie fett
ab der dritten Folie fett

```
\only<4>{erscheint nur auf  
der 4. Folie}  
\uncover<5>{ab der 5. Folie}  
\uncover<6>{ab der 6. Folie}
```

Abbildung: Spezielle Overlay-Befehle

Overlay Spezifikationen - Beispiel

```
\textbf<1->{Immer Fett}  
\textbf<2>{nur auf der 2.  
Folie fett}  
\textbf<3->{ab der  
dritten Folie fett}
```

Abbildung: Overlay Spezifikationen

Immer Fett

nur auf der 2. Folie fett

ab der dritten Folie fett

```
\only<4>{erscheint nur auf  
der 4. Folie}  
\uncover<5>{ab der 5. Folie}  
\uncover<6>{ab der 6. Folie}
```

Abbildung: Spezielle Overlay-Befehle

Overlay Spezifikationen - Beispiel

```
\textbf<1->{Immer Fett}  
\textbf<2>{nur auf der 2.  
Folie fett}  
\textbf<3->{ab der  
dritten Folie fett}
```

Abbildung: Overlay Spezifikationen

Immer Fett

nur auf der 2. Folie fett

ab der dritten Folie fett

```
\only<4>{erscheint nur auf  
der 4. Folie}  
\uncover<5>{ab der 5. Folie}  
\uncover<6>{ab der 6. Folie}
```

Abbildung: Spezielle Overlay-Befehle

existiert nur auf der 4. Folie

Overlay Spezifikationen - Beispiel

```
\textbf<1->{Immer Fett}  
\textbf<2>{nur auf der 2.  
Folie fett}  
\textbf<3->{ab der  
dritten Folie fett}
```

Abbildung: Overlay Spezifikationen

Immer Fett

nur auf der 2. Folie fett

ab der dritten Folie fett

```
\only<4>{erscheint nur auf  
der 4. Folie}  
\uncover<5>{ab der 5. Folie}  
\uncover<6>{ab der 6. Folie}
```

Abbildung: Spezielle Overlay-Befehle

ab der 5. Folie

Overlay Spezifikationen - Beispiel

```
\textbf<1->{Immer Fett}  
\textbf<2>{nur auf der 2.  
Folie fett}  
\textbf<3->{ab der  
dritten Folie fett}
```

Abbildung: Overlay Spezifikationen

Immer Fett

nur auf der 2. Folie fett

ab der dritten Folie fett

```
\only<4>{erscheint nur auf  
der 4. Folie}  
\uncover<5>{ab der 5. Folie}  
\uncover<6>{ab der 6. Folie}
```

Abbildung: Spezielle Overlay-Befehle

ab der 5. Folie ab der 6. Folie

weitere Umgebungen - Spalten

```
\begin{columns}
```

```
\column{.5\textwidth}
```

mit der `{\tt columns}`-Umgebung werden Spalten erzeugt.

```
\column{.5\textwidth}
```

Jede neue Spalte wird mit `\verb|\column{<Breite>}` angegeben

```
\end{columns}
```

Abbildung: die `columns`-Umgebung

mit der `columns`-Umgebung
werden Spalten erzeugt.

Jede neue Spalte wird mit
`\column{<Breite>}` angegeben

weitere Umgebungen - Textblöcke

Textblock

- mit der `block`-Umgebung kann ein Textblock mit einer Überschrift vom restlichen Text abgesetzt werden
- die `verbatim`-Umgebung funktioniert auch in `beamer`, allerdings muss beim betreffenden frame die Option `fragile` angegeben sein
- zusätzlich gibt es die `semiverbatim`-Umgebung. In dieser Umgebung behalten `\`, `{` und `}` ihre Bedeutung, ansonsten wie `verbatim`.

Aussehen der Präsentation

- Das Layout einer Präsentation wird durch **Themen** festgelegt
- beamer verwendet 5 verschiedene Arten von Themen:
 - `color theme` die verwendeten Farben
 - `font theme` die verwendeten Schriftarten
 - `inner theme` Elemente innerhalb von frames (Aufzählungen, Blockumgebungen etc.)
 - `outer theme` der äußere Rahmen (Kopf- und Fußzeile, Titel, Logo etc.)
 - `presentation theme` alle o.g. Aspekte
- Themen werden in der Präambel mit `\use< color, font, inner, outer >theme` festgelegt
- beamer liefert viele verschiedene Themen¹¹

¹¹http://deic.uab.es/~iblanes/beamer_gallery/

beamer-Demo

Handouts erstellen

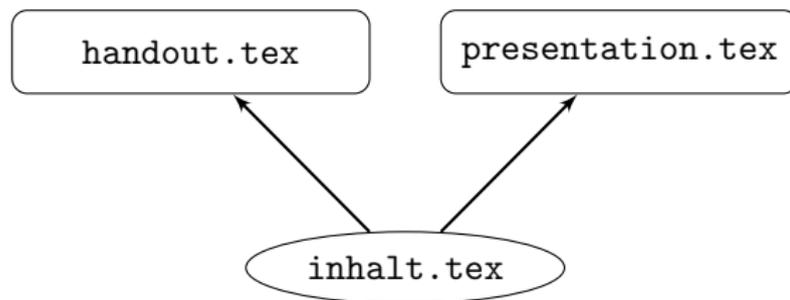
- Handouts helfen den Zuhörern der Präsentation zu folgen
- Zuhörer haben die Möglichkeit, sich Notizen zu einzelnen Punkten zu machen
- `beamer` bietet zwei Möglichkeiten, automatisch Handouts zu generieren
 - ① `\documentclass[handout]{beamer}` generiert eine Version der Folien ohne Overlays
 - ② Sparsamer und lesbarer ist eine `article`-Version der Folien

article-Handouts

- Das Handout als separates `.tex`-Dokument anlegen mit folgender Präambel

```
\documentclass{article}
\usepackage{beamerarticle}
```

- Am besten zwei `.tex` Dokumente mit der jeweiligen Dokumentklasse anlegen und den Inhalt mit `\include` einfügen.



- Soll Text oder ein Element nur auf den Folien oder nur im Handout angezeigt werden, markiert man diesen durch `\mode< presentation, article >`

9 Bibliographie mit $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$

10 Präsentationen mit beamer

11 Zusammenfassung Teil II

- BIB_TE_X bietet die Möglichkeit zur zentralen Literaturverwaltung
- Die Literatur muss in einer `.bib`-Datei angelegt werden, Literaturhinweise im Text werden mit `\cite{cite-key}` angegeben, die Literaturdatenbank wird mit `\bibliography{bibname}` eingebunden
- Literaturhinweise im Autor-Jahr-Format werden leichter mit dem Paket `natbib`

- Die `beamer`-Klasse wird für Folienpräsentationen verwendet
- Titelseite, Gliederung, Aufzählungen etc. in `beamer` funktionieren wie in anderen Dokumentklassen
- einzelne Folien werden durch die `frame`-Umgebung definiert
- Mit `Overlay`-Spezifikationen oder `\pause` können Elemente einer Folie nach und nach aufgedeckt werden
- Besondere Umgebungen: `block`, `columns` und `semiverbatim`
- Das Aussehen der Präsentation wird durch Themen festgelegt
- Handouts lassen sich entweder mit der Option `handout` oder auch im Fließtextformat mit dem Paket `beamerarticle` erzeugen

Hilfe zur Selbsthilfe I

Hilfen/Literatur zu Latex allgemein:

- Website der Tex User Group (Downloads, umfassende Linksammlung zu Dokumentationen und weiteren Usergroups, PracTeX Journal) <http://www.tug.org>
- Homepage (Links zu Dokumentation, FAQ) <http://www.latex-project.org>
- Comprehensive T_EX Archive Network (CTAN) (Download von Paketen inkl. Dokumentation) <http://www.ctan.org>
- Lshort (L^AT_EX-Einführung von Tobias Oetiker, in mehreren Sprachen erhältlich) <http://www.ctan.org/tex-archive/info/lshort/>
- The L^AT_EX wiki book (nutzergeneriertes Handbuch mit vielen hilfreichen Code-Beispielen) <http://en.wikibooks.org/wiki/LaTeX>

Hilfe zur Selbsthilfe II

- Befehlsreferenz
<http://www.weinelt.de/latex/>
- Symbolreferenz
<http://detexify.kirelabs.org>
- Latex Community (User Forum)
<http://www.latex-community.org>

Dokumentation zu einzelnen Paketen:

- $\text{BIB}_{\text{E}}\text{T}_{\text{X}}$ – kurze Einführung von Jürgen Fenn
Managing Citations and your Bibliography with $\text{BIB}_{\text{E}}\text{T}_{\text{X}}$ (2006)
<http://www.tug.org/pracjourn/2006-4/fenn/fenn.pdf>
- natbib Handbuch
über CTAN.org erhältlich.
Z.B. <http://sunsite.informatik.rwth-aachen.de/ftp/pub/mirror/ctan/macros/latex/contrib/natbib/natbib.pdf>

Hilfe zur Selbsthilfe III

- natbib Referenz
<http://merkel.zoneo.net/Latex/natbib.php>
- gnuplot
 - Dokumentation:
http://www.gnuplot.info/docs_4.4/gnuplot.pdf
 - gnuplot und L^AT_EX:
<http://www.gnuplot.info/files/tutorial.pdf>