

L^AT_EX-Skript

Ein Überblick über die Grundlagen

Marvin Ködding

7. April 2025

Inhaltsverzeichnis

1	Grundlagen	4
1.1	Allgemeines	4
1.1.1	Bestandteile eines L ^A T _E X-Dokuments	5
1.1.2	Vor- und Nachteile	5
1.2	Syntax	6
1.3	Aufbau der Datei	6
1.4	Abschnitte	8
1.5	Zeichensatzbefehle	8
1.6	Schriftgrößen	8
1.7	Abstände	9
1.8	Fußnoten	10
1.9	enumerate und itemize	10
1.10	Quelltext	10
1.10.1	Codebeispiel	10
1.10.2	Codebeispiel mit Zeilennummer	11
1.10.3	Zeilenumbrüche	11
1.11	Tabellen	12
1.12	Mehrere Spalten	12
1.13	Wichtige Pakete	13
1.14	Sonderzeichen	14
1.15	Zitationen	14
1.16	Übungen	18
2	Mathematischer Formelsatz	19
2.1	Griechische Buchstaben	19
2.2	Relationen	19
2.3	Operationen	20
2.4	Hoch- und Tiefstellen	20
2.5	Brüche	20
2.6	Wurzeln	20
2.7	Linien	20
2.8	Summen und Produkte	21

2.9	Binomialkoeffizienten	21
2.10	Integrale	21
2.11	Größere Klammern	22
2.12	Matrix	22
2.12.1	array	22
2.12.2	pmatrix	22
2.12.3	Andere Matrizen	23
2.12.4	Kleine Matrizen	24
2.13	Mathematische Schriftarten	25
2.14	Ausrichten von Gleichungen	25
2.14.1	Unnummeriert	25
2.14.2	Nummeriert	25
2.15	Text	26
2.16	Leerzeichen	26
2.17	Limes, Sinus, Cosinus	26
2.18	Weitere Symbole	26
2.19	Übungen	27
3	L^AT_EX anpassen	28
3.1	Eigene Befehle	28
3.2	Aufteilen des Dokuments	28
3.3	DeclareMathOperator	29
3.4	Theoreme	29
3.5	Zähler	30
3.6	Geometry	32
3.7	Übungen	32
4	Präsentationen	33
4.1	Folien erstellen	33
4.2	Einblendeanimation	33
4.3	Titelfolie	34
4.4	Layout	34

Changelog

- **02.09.2022** — Erste Veröffentlichung
- **08.09.2022** — Abschnitt [1.13](#) Wichtige Pakete hinzugefügt
- **07.11.2022** — Abschnitt [3.1](#) überarbeitet und ausführlicher formuliert
- **09.01.2023** — Kleinere Fehler und Ungenauigkeiten behoben
- **06.03.2025** — Abschnitt [1.15](#) hinzugefügt

- **24.03.2025** — Abschnitt [1.6](#) und [1.12](#) hinzugefügt

Kapitel 1

Grundlagen

1.1 Allgemeines

- \LaTeX ist keine Textverarbeitung (wie Word), sondern ein Textsatzsystem, d.h. es nimmt eine bestehende Datei (Quelldatei) und wandelt sie gemäß bestimmten Vorgaben in ein druckfähiges Format (*Ausgabeformat*) um. Das ist sehr oft PDF, aber es gibt auch PS (*Postscript*), DVI oder HTML.
- Die Vorgaben sind teils in der Quelldatei in Form von Befehlen vorgegeben, teils im Programm selbst implementiert. Beispielsweise trennt \LaTeX automatisch Wörter, die über das Zeilenende hinausgehen, gemäß internen Regeln. Falls \LaTeX die Wörter nicht kennt, kann man diese mit `\-` trennen.
- \LaTeX ist ideal zum Schreiben längerer wissenschaftlicher Arbeiten insb. für mathematische und technische Themen (Mathematischer Formelsatz)
- Insb. ist \LaTeX kein Texteditor. Durch die strikte Trennung von Layout und Inhalt kann man sich voll und ganz auf das Verfassen des Textes konzentrieren.
- UNIX-Gedanke: Lieber viele kleine Werkzeuge, die man kombinieren kann als ein großes Werkzeug. Man benötigt zur Erstellung von Quelldateien einen Editor, den man frei wählen kann. Das könnte theoretisch auch der Windows-Texteditor sein, es gibt aber extra \TeX -Editoren, die speziell für \LaTeX ausgelegt sind. (Wir werden i.d.R. Overleaf verwenden)
- In den Quelldateien wird normalerweise nicht gesagt, wo sich Zeilenumbrüche befinden, wie lang eine Seite ist etc. Dies wird alles von \LaTeX entschieden, das immerhin speziell für diesen Zweck entwickelt wurde. Ebenso die Seitenzahlen, Abschnittsnummern, Querbezüge, Inhaltsverzeichnis, Nummerierung der Fußnoten. Auch welche Textteile z.B. fett gedruckt werden sollen bestimmt zumeist \LaTeX der Nutzer gibt lediglich an, welche Funktion der jeweilige Text erfüllt (z.B. Abschnittsüberschrift).

Diese Einstellungen lassen sich aber bei Bedarf alle neu setzen, wenn auch evtl. mit viel Aufwand.

1.1.1 Bestandteile eines L^AT_EX-Dokuments

L^AT_EX liest und erzeugt mehrere Typen von Dateien

- Eingabe- und Quelldatei `.tex` — Reine Textdatei, die mit einem Texteditor erstellt wird
- Definitionsdateien `.cls`, `.sty` — Struktur und Layout
- Metrikangaben für Zeichensatz `.tfm`
- Ausgabedatei `.pdf`, `.dvi` — Ausgabedateien, entweder im pdf oder dvi Format
- Protokolldatei `.log`
- Ausgabedatei für Querverweise `.aux`
- Ausgabedatei für Inhaltsverzeichnis `.toc`
- Ausgabedatei für Abbildungsverzeichnis `.lof`
- Ausgabedatei für Tafelverzeichnis `.lot`
- Indizierte Dateien `.idx` — Zur Weiterverarbeitung mit `makeindex`

1.1.2 Vor- und Nachteile

Vorteile

- Professionell gestaltete Layouts stehen zur Verfügung. Dadurch wird Buchdruckqualität erreicht
- Sehr gutes Setzen von mathematischen Formeln
- Anwender muss sich nur um den Inhalt kümmern, drucktechnische Gestaltung kann separat definiert werden.
- Fußnoten, Literaturangaben, Inhaltsverzeichnisse, Tabellen, Zeichnungen, numerierte Abschnitte u. v. m. können ohne großen Aufwand erstellt werden.
- Sehr kleine Eingabedateien im Vergleich zu anderen Textverarbeitungsprogrammen.

Nachteile

- Für Einsteiger am Anfang etwas „kryptisch“
- „Kompilieren“ von Text ungewohnt, da L^AT_EX nicht WYSIWYG ist

- Erfordert etwas Einarbeitungszeit
- Abweichen von vorgefertigten Layouts ist arbeitsintensiv und erfordert meist sehr gute T_EX bzw L^AT_EX Kenntnisse

1.2 Syntax

- L^AT_EX basiert auf T_EX und erweitert dies um einige bequeme Funktionen. Dadurch existieren aber die T_EX sowie die L^AT_EX Syntaxen, die sich manchmal sehr ähnlich sind
- Befehle starten generell mit einem Backslash `\`. Befehle bestehen dann aus mehreren Buchstaben (der Befehl wird durch ein Leerzeichen, ein Sonderzeichen oder eine Ziffer beendet) oder aus genau einem Sonderzeichen oder einer Ziffer.
- Manche Befehle verlangen dann noch nach Parametern. Verpflichtende Parameter stehen dabei in geschwungenen Klammern und optionale Parameter in eckigen Klammern. Z.B. `\rightarrow`, `\frac{2}{3}`, `\sqrt[3]{x}`.
- Manche Befehle wirken wie Schalter `\em` Dieser Text ist abgehoben. Um diese Wirkung zu begrenzen, kann man Gruppen definieren, indem man diesen Text in geschwungene Klammern schreibt. Dabei enden alle Schalter am Ende dieser Gruppe. `{\em Das ist hervorgehoben}` und das nicht mehr
- Eine weitere Variante sind sogenannte Umgebungen, die durch ein `\begin{...}` beginnen und mit `\end{...}` enden. Beispielsweise:

```
\begin{center}
  Dieser Text ist zentriert.
\end{center}
```

- Alte Befehle aus T_EX nehmen Parameter ohne Klammern entgegen. `\topmargin4mm`
- Kommentare beginnen mit `%` und gelten dann für den Rest der Zeile. Dies kann für Notizen des Autors oder zur übersichtlicheren Gestaltung der Quelldatei verwendet werden.
- L^AT_EX-Dokumente werden normalerweise von oben nach unten und von links nach rechts gelesen

1.3 Aufbau der Datei

- Eine L^AT_EX-Datei besteht aus einem Vorspann und einem Textteil
- Der erste Befehl des Vorspanns ist i.d.R. `\documentclass[optionen]{dokumentklasse}`

- Der Textteil beginnt mit `\begin{document}` und endet mit `\end{document}`. Dort steht dann der „Inhalt“ des Dokuments. Eingaben nach diesem Bereich werden ignoriert.
- Eine Quelldatei könnte folgendermaßen aussehen:

```
\documentclass[11pt, a4paper]{article}
\usepackage[german]{babel} %für Umlaute
\title{\LaTeX-Kurs}
\author{Marvin Koedding}
\begin{document}
  \maketitle
  Hallo Welt
\end{document}
```

Dokumentenklassen

- Es stehen u.a. die folgenden Dokumentenklassen zur Verfügung:
 - article — Für Artikel in wissenschaftlichen Zeitschriften, Praktikums- und Seminararbeiten, Anträge, Gutachten, kürzere Berichte
 - report — Für längere Berichte mit mehreren Kapiteln, Diplomarbeiten, Dissertationen, Skripte, etc
 - book — Für Bücher mit mehreren Abschnitten (und Kapiteln)
 - proc — Für Konferenzbände
 - letter, dinbrief, g-brief — Für Briefe
 - slides, seminar, prosper, beamer — Für Folien
- Zusätzlich stehen meistens folgende Optionen zur Verfügung:
 - 10pt, 11pt, 12pt — Für 10, 11, 12 Punkte hohe Schrift
 - a4paper — Für DIN A4-Papier
 - fleqn — Für linksbündige (statt zentrierte) mathematische Gleichungen
 - leqno — Für Gleichungsnummern links statt rechts
 - notitlepage/titlepage — Für (k)eine Titelseite
 - onecolumn/twocolumn — Für ein-/zweispaltigen Druck
 - oneside/twoside — Für ein-/zweiseitigen Druck

1.4 Abschnitte

Wir haben die Möglichkeit, ein Dokument in Abschnitte zu gliedern (Wir sind z.B. gerade im Abschnitt „Abschnitte“). Dafür bietet L^AT_EX die folgenden Typen:

- `part` (nur für Bücher)
- `chapter` (nur für Bücher)
- `section`
- `subsection`
- `subsubsection`
- `paragraph`
- `subparagraph`

Einen neuen Abschnitt kennzeichnet man mit z.B. `\section{Name des Abschnitts}` bzw. `\subsection{Name des Unterabschnittes}`. Man kann mit `\tableofcontents` ein Inhaltsverzeichnis erstellen lassen (s. Anfang des Dokuments).

Will man einen Abschnitt nicht nummerieren und nicht im Inhaltsverzeichnis sehen, dann kann man dies mit `\section*{Name}` bzw. `\subsection*{Name}` erreichen.

1.5 Zeichensatzbefehle

Das Aussehen eines einzelnen Zeichens ist abhängig von der verwendeten Schriftart. Eine Umstellung kann u.a. mit folgenden Befehlen erfolgen:

Ein roman Text. Hier kann ein Text mit Schreibmaschine stehen. Ein serifenloser Text. *Kursive* oder *geneigte* Schrift. Schöne Texte für HINWEISE oder **Warungen**. *Hervorgehoben* und unterstrichen geht auch.

Ein `\textrm{roman}` Text.
Hier kann ein `\texttt{Text mit Schreibmaschine}` stehen.
`\textsf{Ein serifenloser}` Text.
`\textit{Kursive}` oder `\textsl{geneigte}` Schrift.
Schöne Texte für `\textsc{Hinweise}` oder
`\textbf{Warungen}`. `\emph{Hervorgehoben}`
und `\underline{unterstrichen}` geht auch.

1.6 Schriftgrößen

- `\tiny` – Hallo Welt!

- `\scriptsize` – Hallo Welt!
- `\footnotesize` – Hallo Welt!
- `\small` – Hallo Welt!
- `\normalsize` – Hallo Welt!
- `\large` – Hallo Welt!
- `\Large` – Hallo Welt!
- `\LARGE` – Hallo Welt!
- `\huge` – Hallo Welt!
- `\Huge` – Hallo Welt!

1.7 Abstände

Horizontale Abstände:

- `\quad` — A B
- `\qquad` — A B
- `\;` — A B
- `\,` — A B
- `\hspace{4cm}`
- `\hspace*{4cm}`
- `ABCD\hfill EFGH` — ABCD

EFHG

Vertikale Abstände:

- `\smallskip`
- `\medskip`
- `\bigskip`
- `\vspace{4mm}`
- `\vspace*{4mm}`

1.8 Fußnoten

Fußnoten¹ werden automatisch nummeriert und am unteren Ende der Seite gedruckt. Fußnoten`\footnote{Das ist eine Fußnote}` werden ...

1.9 enumerate und itemize

- Erster Punkt
- Zweiter Punkt

```
\begin{itemize}
  \item Erster Punkt
  \item Zweiter Punkt
\end{itemize}
```

1. Erster Punkt
2. Zweiter Punkt

```
\begin{enumerate}
  \item Erster Punkt
  \item Zweiter Punkt
\end{enumerate}
```

1.10 Quelltext

Quelltext kann man sehr einfach mit `minted` oder `listings` erstellen. Ich verwende immer `minted`, weswegen das hier ins Skript kommt. Wer sich `listings` mal anschauen möchte, kann sich gerne auf Youtube oder im Handbuch darüber informieren.

1.10.1 Codebeispiel

```
\begin{minted}{java}
System.out.println("Hello World!");
int i = 0;
int j = 1;
System.out.println((i == j) ? "ja" : "nein");
\end{minted}

System.out.println("Hello World!");
int i = 0;
```

¹Das ist eine Fußnote

```
int j = 1;
System.out.println((i == j) ? "ja" : "nein");
```

1.10.2 Codebeispiel mit Zeilennummer

```
\begin{minted}[linenos]{java}
System.out.println("Hello World!");
int i = 0;
int j = 1;
System.out.println((i == j) ? "ja" : "nein");
\end{minted}
```

```
1 System.out.println("Hello World!");
2 int i = 0;
3 int j = 1;
4 System.out.println((i == j) ? "ja" : "nein");
```

1.10.3 Zeilenumbrüche

Wenn man Programmcode mit `minted` schreibt, so wird der Text normalerweise nicht umgebrochen.

```
\begin{minted}[linenos]{java}
System.out.println("Hello World! Das ist eine weitere Zeile, in der unser Text
steht. Der Text ist etwas zu lang");
\end{minted}
```

```
1 System.out.println("Hello World! Das ist eine weitere Zeile, in der unser Text steht.
```

Wollen wir den Programmcode umbrechen, so kann man dies mit dem optionalen Parameter `breaklines` umsetzen.

```
\begin{minted}[breaklines, linenos]{java}
System.out.println("Hello World! Das ist eine weitere Zeile, in der unser Text
steht. Der Text ist etwas zu lang");
\end{minted}
```

```
1 System.out.println("Hello World! Das ist eine weitere Zeile, in der unser
  ↳ Text steht. Der Text ist etwas zu lang");
```

Wenn wir in einer Zeile einen Befehl, wie zum Beispiel `System.out.println()`; darstellen wollen, können wir das mit `\mintinline{java}{System.out.println();}` umsetzen.

1.11 Tabellen

Die `tabular`-Umgebung dient zum Setzen von Tabellen. Die Tabellenbreite wird dabei von \LaTeX bestimmt. Die Parameter des Tabularbefehls bestimmt dabei die Ausrichtung der Spalte:

<code>l</code>	linksbündig
<code>c</code>	zentriert
<code>r</code>	rechtsbündig
<code>p{breite}</code>	Spalte mit fester Breite
<code> </code>	senkrechter Strich

```
\begin{tabular}{r|l}
\hline
\texttt{l} & linksbündig \\
\texttt{c} & zentriert \\
\texttt{r} & rechtsbündig \\
\texttt{p\{breite\}} & Spalte mit fester Breite \\
\texttt{\$} & senkrechter Strich \\
\hline
\end{tabular}
```

1.12 Mehrere Spalten

Es gibt mehrere Möglichkeiten, Texte in mehreren Spalten anzeigen zu lassen. Die gängigsten sind `multicol` und `minipage`.

Dabei verteilt `multicol` den Text auf mehrere Spalten. So erzeugt

```
\begin{multicols}{3}
\lipsum[1]
\end{multicols}
```

die Ausgabe

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senec-	tus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Ae-	nean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
---	--	---

Mit `minipage` kann der Text manuell auf mehrere Spalten aufgeteilt werden. Hierbei müssen die Spalten auch nicht gleichmäßig befüllt werden.

Der \LaTeX -Code

```
\begin{minipage}{0.2\textwidth}
  Marvin Ködding\\scriptsize MAT06\\PH Heidelberg
\end{minipage}
\begin{minipage}{.6\textwidth}
  \centering
  -- Blatt 01 --\\textbf{Grundlagen zu \LaTeX}
\end{minipage}
\begin{minipage}{.2\textwidth}
  \flushright SoSe 2025\\scriptsize \today
\end{minipage}
```

erzeugt folgende Ausgabe.

Marvin Ködding
MAT06
PH Heidelberg

– Blatt 01 –
Grundlagen zu \LaTeX

SoSe 2025
7. April 2025

1.13 Wichtige Pakete

Dies sind Pakete, die man installiert haben könnte, um sich das Leben mit \LaTeX einfacher zu machen. Diese werden hier aber nicht behandelt, sondern der Leser ist angehalten, sich selbstständig über diese Pakete zu informieren.

- `inputenc` — Wird für die direkte Eingabe von Umlauten gebraucht
- `fontenc` — Trennung von Umlauten
- `lmodern` — Erstellt ein moderneres Schriftbild
- `babel` — Worttrennung nach anderen Sprachen (german - alte deutsche Rechtschreibung, ngerman - neue deutsche Rechtschreibung)

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

- `graphicx` — Einbinden von Graphiken und Bildern
Einbinden mit `\includegraphics[scale=1]{pfad zum bild}`
- `tikz` — Zum zeichnen von Graphiken und Diagrammen. Eine gute Anleitung findet man [hier](#)
- `tabularx` — Tabellen mit automatischem Zeilenumbruch
- `longtable` — Ermöglicht überlange Tabellen
- `geometry` — Vereinfacht das Ändern der Seitenränder und Papierformate

- `fancyvrb` — Erweiterung der Verbatimumgebung, zum Beispiel Schriftart und Schriftgröße
- `hyperref` — Erstellt [Verweise](#) innerhalb und nach außerhalb eines PDF Dokumentes
- `pdfpages` — Einbinden von PDF Dateien zum Beispiel weitere bereits fertige PDFs in ein neues PDF einfügen.
- `amsmath`, `amssymb`, `amsfont` — Zusätzliche Elemente für mathematische Formeln
- `minted` — Für Syntax-Highlighting von Quellcode

1.14 Sonderzeichen

In \LaTeX werden Zeichen wie bspw. Anführungszeichen als Steuersymbole verwendet. So wird `"a` zu `ä` ausgewertet oder `"s` zu `ß`. Wenn man nun Anführungszeichen verwenden möchte, so schreibt man entweder `'Test'`, um `"Test"` zu erhalten oder verwendet die deutschen Anführungszeichen `\glqq{}Test\grqq{}`, die zu `„Test“` ausgewertet werden.

1.15 Zitationen

Um mit \LaTeX zu zitieren, verwendet man in der Regel ein Literaturverwaltungssystem wie `BibTeX` oder `BibLaTeX`. Hier ist ein einfaches Beispiel, wie man im Text zitiert und das Literaturverzeichnis hinzufügt:

1. Erstelle eine `.bib`-Datei: Diese Datei enthält alle benötigten Literaturangaben im `BibTeX`-Format.

In dieser `.bib`-Datei ist die zu zitierende Literatur zusammengetragen. Jeder einzelne Eintrag wird nach folgendem Schema eingetragen:

Zunächst wird mit `@` ein Eintrag gestartet, gefolgt von der Art des Dokuments.

Es gibt insgesamt 14 Arten von Literatur:

- **article**: Ein wissenschaftlicher Artikel, der in einer Zeitschrift oder einem Magazin veröffentlicht wurde.
- **book**: Ein eigenständiges Buch, das ein Thema umfassend und systematisch behandelt.
- **booklet**: Eine kleinere, oft informellere Publikation ohne feste ISBN, die kompakte Informationen bietet.
- **conference**: Ein Beitrag, der auf einer Konferenz vorgestellt und in den zugehörigen Schriften veröffentlicht wurde.

- **inbook**: Ein Kapitel oder Abschnitt innerhalb eines Buches, das als eigenständiger Beitrag zitiert werden kann.
- **incollection**: Ein Beitrag in einem Sammelband, der von einem Herausgeber zusammengestellt wurde.
- **inproceedings**: Ein Konferenzbeitrag, der in den veröffentlichten Tagungsbänden einer Konferenz enthalten ist.
- **manual**: Eine Anleitung oder ein Handbuch, das detaillierte Informationen zur Bedienung oder Anwendung liefert.
- **masterthesis**: Eine wissenschaftliche Abschlussarbeit auf Master-Niveau, in der eigenständige Forschungsergebnisse präsentiert werden.
- **misc**: Eine sonstige Quelle, die keiner der anderen spezifischen Kategorien zugeordnet werden kann.
- **phdthesis**: Eine Doktorarbeit, die umfangreiche, eigenständige Forschungsergebnisse dokumentiert.
- **proceedings**: Eine Sammlung von Konferenzbeiträgen, die gemeinsam als Buch veröffentlicht werden.
- **techreport**: Ein technischer Bericht, der detaillierte Ergebnisse und Informationen aus der Forschung oder Technik präsentiert.
- **unpublished**: Eine zitierwürdige Quelle, die noch nicht offiziell veröffentlicht wurde.

Hierauf folgt ein Schlüssel, der den Eintrag identifiziert. Danach werden die restlichen Daten des Dokuments aufgelistet.

Ein Beispiel für einen solchen Eintrag:

```
@incollection{fischer.spannagel.2012,
author = "Fischer, Maike and Spannagel, Christian",
title = "Lernen mit Vorlesungsvideos in der umgedrehten Mathematikvorlesung",
year = 2012,
booktitle = "DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.",
publisher = "Gesellschaft für Informatik e.V.",
address = "Bonn",
isbn = "978-3-88579-601-5",
pages = "225--236",
}
```

2. Im Hauptdokument: Binde das Paket im Präambel mit `\usepackage{biblatex}` ein. Hier sind einige sinnvolle Parameter im Beispiel aufgetragen. Verwende den Befehl `\printbibliography` am Ende deines Dokuments, um das Literaturverzeichnis einzufügen.

3. Zum Zitieren im Text: Verwende `\cite{schlüssel}`, wobei 'schlüssel' der Referenzname aus deiner .bib-Datei ist.

```
\usepackage[  
  backend=biber,  
  style=alphabetic,  
  sorting=ynt  
{biblatex}  
\addbibresource{lit.bib}
```

Text mit Zitat `\cite{fischer.spannagel.2012}`.

```
\printbibliography
```

Text mit Zitat [FS12].

Literatur

- [FS12] Maïke Fischer und Christian Spannagel. „Lernen mit Vorlesungsvideos in der umgedrehten Mathematikvorlesung“. In: *DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e. V.* Bonn: Gesellschaft für Informatik e.V., 2012, S. 225–236. ISBN: 978-3-88579-601-5.

1.16 Übungen

1. Erstelle ein Dokument mit Titel, Autor, mehreren Abschnitten, etwas Text und einer Fußnote
2. Erstelle eine (nummerierte, unnummerierte) Liste mit 3 Punkten. Bau nun unter dem 2. Punkt eine weitere (nummerierte, unnummerierte) Liste ein und untersuche das Verhältnis von „Oberliste“ und „Unterliste“
3. Setze die folgende Tabelle:

NUMMER	TEXT	<i>Buchstabe</i>
1	links	x
2	gerade	y
3	mitte	z

4. Setze den folgenden Text mit Python-Syntaxhighlighting:

```
b = 10
if b > 20:
    print("b > 20")
print("b =", b)
```

Kapitel 2

Mathematischer Formelsatz

Mathematische Formeln werden in Dollarzeichen eingeschlossen: $f(x)=x$ ergibt $f(x) = x$. Eine abgesetzte Formel wird mit doppelten Dollarzeichen eingeschlossen: $f(x)=x$ ergibt

$$f(x) = x$$

Für mathematische Formeln empfehle ich das Paket `amsmath`.

Alles aus den Unterkapiteln wird nur im Formelsatz gesetzt. D.h. z.B. für griechische Buchstaben muss man den Text in Dollarzeichen einschließen.

2.1 Griechische Buchstaben

- α
- β

Eine Liste aller griechischen Buchstaben ist [hier](#) zu finden.

2.2 Relationen

- $a \leq b$
- $a \geq b$
- $A \subset B$
- $A \subseteq B$
- $a \mid b$

Eine Liste aller wichtigen Relationen ist [hier](#) zu finden.

2.3 Operationen

- $\$a\cdot b\$$ — $a \cdot b$
- $\$a\times b\$$ — $a \times b$

Eine Liste aller wichtigen Operationen ist [hier](#) zu finden.

2.4 Hoch- und Tiefstellen

- $\$x^2\$$ — x^2
- $\$x^23\$$ — x^23
- $\$x_1\$$ — x_1
- $\$x_{12}\$$ — x_{12}
- $\$x_{12}\$$ — x_{12}
- $\$x^{23}\$$ — x^{23}
- $\$x_1^2\$$ — x_1^2
- $\$x_{12}^{23}\$$ — x_{12}^{23}

2.5 Brüche

Brüche werden mit dem Befehl $\frac{a}{b}$ erstellt. Dabei ist a der Zähler und b der Nenner. D.h. $\frac{3x+1}{4x}$ ergibt $\frac{3x+1}{4x}$

2.6 Wurzeln

Die Größe des Wurzelzeichens legt L^AT_EX automatisch fest.

- $\sqrt{x^2+y^2}$
- $\sqrt[3]{x^2}$
- $\sqrt{x^2+\sqrt{y}}$

2.7 Linien

- $\underline{x+y}$
- $\underline{x+y}$

- $\overline{x+y} = x+y$
- $\underbrace{x+\cdots+x}_{=z} = x+\cdots+x$
- $\overbrace{x+\cdots+x}^{=z} = x+\cdots+x$

2.8 Summen und Produkte

$\sum_{i=1}^n i = \frac{n(n+1)}{2}$ wird mit $\text{\texttt{\$}\sum_{i=1}^n i = \frac{n(n+1)}{2}\text{\texttt{\$}}}$ erstellt. Wenn man dies in doppelte Dollarzeichen schreibt, erhält man

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

Man kann auch Texte aus „inline-Umgebungen“ als „display“ darstellen, indem man der Formel `\displaystyle` voranstellt. Also

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

Das Produkt funktioniert analog. $\prod_{i=1}^{10} 3i$ bzw

$$\prod_{i=1}^{10} 3i$$

Erstellt wird das mit $\text{\texttt{\$}\prod_{i=1}^{10} 3i\text{\texttt{\$}}}$ bzw $\text{\texttt{\$}\prod_{i=1}^{10} 3i\text{\texttt{\$}}}$

2.9 Binomialkoeffizienten

$$\text{\texttt{\$}\{n \choose k\}\text{\texttt{\$}}}$$

$$\binom{n}{k}$$

2.10 Integrale

$$\text{\texttt{\$}\int_a^b f(x) \text{\texttt{\text{d}}}x\text{\texttt{\$}}}$$

$$\int_a^b f(x)dx$$

2.11 Größere Klammern

`$$\int\limits_a^b f(x)$$` erstellt folgende Formel:

$$\left(\int_a^b f(x)\right)$$

Die Klammern passen nicht zur Größe der Formel. Dafür gibt es `\left(` und `\right)`. Diese passen sich an die Größe der Formel an. So ergibt

`$$\left(\int\limits_a^b f(x)\right)$$` dann

$$\left(\int_a^b f(x)\right)$$

Ansonsten kann man mit `$$\big(f(x)\big)$$` die Größe von Klammern verändern.

Möglich wäre: `$$\bigl(\Bigl(\biggl(\Biggl(`, also `((((` bzw.

`$$\bigr)\Bigr)\biggr)\Biggr)$$`, also `))))`

2.12 Matrix

2.12.1 array

`$$\left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right)$$`

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2.12.2 pmatrix

erfordert das `amsmath`-Paket

```

$$
\begin{pmatrix}
a & b & c & d \\
b & a & c+d & c-d \\
0 & 0 & a+b & a-b \\
0 & 0 & ab & cd
\end{pmatrix}

```

```
\end{pmatrix}
$$
```

$$\begin{pmatrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{pmatrix}$$

2.12.3 Andere Matrizen

Eckige Klammern – bmatrix

```
$$
\begin{bmatrix}
a & b & c & d \\
b & a & c+d & c-d \\
0 & 0 & a+b & a-b \\
0 & 0 & ab & cd
\end{bmatrix}
\end{bmatrix}
$$
```

$$\begin{bmatrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{bmatrix}$$

Geschwungene Klammern – Bmatrix

```
$$
\begin{Bmatrix}
a & b & c & d \\
b & a & c+d & c-d \\
0 & 0 & a+b & a-b \\
0 & 0 & ab & cd
\end{Bmatrix}
\end{Bmatrix}
$$
```

$$\begin{Bmatrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{Bmatrix}$$

Einfache Striche – vmatrix

```
$$
\begin{vmatrix}
a & b & c & d \\
b & a & c+d & c-d \\
0 & 0 & a+b & a-b \\
0 & 0 & ab & cd
\end{vmatrix}
$$
```

$$\begin{vmatrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{vmatrix}$$

Doppelte Striche – Vmatrix

```
$$
\begin{Vmatrix}
a & b & c & d \\
b & a & c+d & c-d \\
0 & 0 & a+b & a-b \\
0 & 0 & ab & cd
\end{Vmatrix}
$$
```

$$\left\| \begin{matrix} a & b & c & d \\ b & a & c+d & c-d \\ 0 & 0 & a+b & a-b \\ 0 & 0 & ab & cd \end{matrix} \right\|$$

2.12.4 Kleine Matrizen

Wenn wir versuchen, inline Matrizen zu verwenden

```
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix},
```

sieht es ziemlich groß aus. Stattdessen können wir auch

```
\big(\begin{smallmatrix}
a & b \\
c & d
\end{smallmatrix}\big)
```

`\end{smallmatrix}\big)$`
verwenden.

Wenn wir versuchen, inline Matrizen zu verwenden $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, sieht es ziemlich groß aus. Stattdessen können wir auch $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ verwenden.

2.13 Mathematische Schriftarten

Will man die reellen Zahlen darstellen, so verwendet man das Symbol \mathbb{R} . Diese Symbole erhält man mit `\mathbb{R}`. Äquivalent funktioniert das natürlich für alle Buchstaben. Es gibt noch andere Schriftarten:

- `\mathcal{P}` — \mathcal{P}
- `\mathscr{P}`¹ — \mathscr{P}
- `\mathfrak{P}` — \mathfrak{P}

2.14 Ausrichten von Gleichungen

Wollen wir mehrere Gleichungen zueinander ausrichten, so können wir das Paket `align` verwenden. Zwischen `align` und `align*` besteht der Unterschied, dass ein Stern die Nummerierung unterdrückt. Mit `&` können die Formeln so ausgerichtet werden, dass alle `&` untereinanderstehen.

2.14.1 Unnummeriert

```
\begin{align*}
a+b &= c+d \\
x &= w+y+z \\
m+n+o+q &= q
\end{align*}
```

$$\begin{aligned} a + b &= c + d \\ x &= w + y + z \\ m + n + o + q &= q \end{aligned}$$

2.14.2 Nummeriert

```
\begin{align}
```

¹Erfordert das Paket `mathrsfs`

```

a+b &= c+d \\
x &= w+y+z \\
m+n+o+q &= q
\end{align}

```

$$a + b = c + d \quad (2.1)$$

$$x = w + y + z \quad (2.2)$$

$$m + n + o + q = q \quad (2.3)$$

2.15 Text

Wenn man versucht, im Mathemodus Text zu schreiben, dann merkt man, dass z.B. `$Das ist ein Text$` ergibt dann *DasisteinText*. Das kann man mit `Das ist ein Text` umgehen.

2.16 Leerzeichen

L^AT_EX verzichtet, wie oben gesehen, auf Leerzeichen im Mathe-Mode. Leerzeichen können mit `$f(x)\ =\ x$` ($f(x) = x$) erzwungen werden.

2.17 Limes, Sinus, Cosinus

`$sin(x)$` ($\sin(x)$) ist nicht korrekt, da vordefinierte Funktionen wie sin, cos, ln, log, ... nicht kursiv geschrieben werden. Dafür bietet L^AT_EX schon für einige Funktionen die Möglichkeit, diese durch z.B. `\sin(x)` ($\sin(x)$) korrekt zu setzen.

- `$\cos(x)$` — $\cos(x)$
- `$\ln(x)$` — $\ln(x)$
- `$\lim_{x \rightarrow 0} \frac{x^2}{4x}$` — $\lim_{x \rightarrow 0} \frac{x^2}{4x}$
- `$$\lim_{x \rightarrow \infty} \frac{x^2}{4x}$$` —

$$\lim_{x \rightarrow \infty} \frac{x^2}{4x}$$

2.18 Weitere Symbole

Weitere Symbole findet man [hier](#).

2.19 Übungen

1. Mach dir den Unterschied zwischen `\int_a^b` und `\int\limits_a^b` klar.

2. Setze die folgenden Formeln:

$$C_\alpha = \int_0^\alpha \Gamma(x) dx$$

3. Setze:

(a)

$$x \vee (y \wedge z) = (x \wedge y) \vee (x \wedge z)$$

(b)

$$(x \cdot (b + c)) \left((a \cdot b) + \frac{c}{x} \right) \text{ für } x \neq 0$$

4. (a) Setze den folgenden Text:

Mithilfe der **Mitternachtsformel** können wir die Lösungen der Gleichung $ax^2 + bx + c = 0$ bestimmen. Diese sind:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(b) Formuliere einen ähnlichen Text für die pq -Formel.

5. Setze den folgenden Text:

Den **Satz des Pythagoras** können wir in einem *rechtwinkligen* Dreieck mit den Kathetenlängen a und b und der Hypothenusenlänge c anwenden. Dann gilt:

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

6. Bestimme die Lösungen der Gleichung $3x^2 + 6x = 12$ schriftlich. Stelle deine Umformungen mit `\LaTeX` dar. Setze die Gleichheitszeichen untereinander. (Hinweis: `align*`)

Kapitel 3

L^AT_EX anpassen

3.1 Eigene Befehle

L^AT_EX kann neue Befehle definieren. Damit kannst du dir deine Arbeit vereinfachen, um wiederkehrende Befehle und Befehlsgruppen zu einem Befehl zusammenzufassen.

`\newcommand{\name}{definition}` definiert hierbei einen neuen Befehl. Wenn du nun irgendwo in deiner `.tex`-Datei `\name`, so ersetzt L^AT_EX `\name` durch das, was in `{definition}` steht.

Z.B. `\newcommand{\kurs}{\LaTeX-Kurs im Jahre 2025}`

Das ist der `\kurs` ergibt dann „Das ist der L^AT_EX-Kurs im Jahre 2025“

Ein Anwendungsbeispiel wäre hier die e-Funktion. Diese wird meistens mit `e^x` (e^x) gesetzt. Das ist aber falsch. Man muss die e-Funktion eigentlich `\mathrm{e}^x` (e^x) setzen, da hier `e` keine Variable ist, sondern eine feststehende Konstante. Genauso bei `\sin`, `\cos`, . . .

Nun könnte man anfangen, und in jeder Gleichung `\mathrm{e}^x` schreiben.

Stattdessen können wir uns auch eine Funktion `\ex{}` schreiben. Dafür möchten wir aber als Argument den Exponenten der e-Funktion mitgeben.

Dafür können wir `\newcommand{\ex}[1]{\mathrm{e}^{\#1}}` schreiben. Die `[1]` zeigt an, dass der Befehl `\ex` ein Argument entgegen nimmt. Dieses Argument können wir in unserer Definition des Befehls mit `#1` verwenden. In L^AT_EX kannst du Befehle mit maximal 9 Argumenten definieren.

3.2 Aufteilen des Dokuments

Man kann z.B. für größere Projekte das Dokument in mehrere Dateien aufteilen. Diese Dateien kann man dann mit `\input{file.tex}` oder `\include{file.tex}` wieder einbinden.

`\input{}` fügt dabei den Quelltext einfach nur ein, `\include{}` setzt dabei den Quelltext immer auf eine neue Seite.

3.3 DeclareMathOperator

Manchmal muss man eigene mathematische Funktionen definieren. Dies geschieht mit `\DeclareMathOperator{\befehl}{definition}`. Z.B. `\DeclareMathOperator{\rk}{rank}`

3.4 Theoreme

Ein Dokument sollte gegliedert sein. Deswegen bieten L^AT_EX die Möglichkeit, das eigene Dokument in Theoreme (Definitionen, Sätze, Lemmata, ...) zu gliedern. Dafür kann man entweder vordefinierte Theoreme verwenden oder eigene definieren.

```
\newenvironment{proof}{\bf Beweis }{\hfill$\square$}
\newtheorem{sz}{Satz}
\newtheorem{lem}[sz]{Lemma}
\newtheorem{df}{Definition}
...
\begin{lem}
Es gibt eine natürliche Zahl $n \neq 2$.
\end{lem}
\begin{sz}
Eine natürliche Zahl $n$ hat einen Nachfolger
\end{sz}
\begin{proof}
Folgt direkt aus Peano-Axiomen
\end{proof}
\begin{df}
Eine (affine) Gerade $g$ in $\mathbb{R}^n$ ist eine Menge der Form
$\{a + \lambda v \mid \lambda \in \mathbb{R}\}$
$= a + \mathbb{R}v$
wobei $v \neq 0$, $a \in \mathbb{R}^n$.
\end{df}
```

Lemma 1 *Es gibt eine natürliche Zahl $n \neq 2$.*

Satz 2 *Eine natürliche Zahl n hat einen Nachfolger*

Beweis Folgt direkt aus Peano-Axiomen □

Definition 1 *Eine (affine) Gerade g in \mathbb{R}^n ist eine Menge der Form*

$$\{a + \lambda v \mid \lambda \in \mathbb{R}\} = a + \mathbb{R}v$$

wobei $v \neq 0, a \in \mathbb{R}^n$.

Die proof-Umgebung haben wir selber als environment definiert. Dabei können wir als Parameter übergeben, was am Anfang und was am Ende einer Umgebung stehen soll. Dabei gilt

```
\newenvironment{\name}{Definition Anfang}{Definition Ende}.
```

3.5 Zähler

L^AT_EX verwendet Zähler, um Kapitelnummern, Seitenzahlen, etc. darzustellen. Diese Zähler können wir auch abgreifen. Wir können beispielsweise mit `\thepage` ausgeben, welchen Wert der Zähler *page* hat. Das wäre jetzt gerade 30.

... Das wäre jetzt gerade `\thepage`.

Wir können den Zähler auch bearbeiten, indem wir `\setcounter{page}{100}` oder `\addtocounter{page}{10}` verwenden.

Will man beispielsweise nach jedem Kapitel die Seitenzahl zurücksetzen, so kann man `\numberwithin{page}{chapter}` verwenden. Dann sieht die Seitenzahl folgendermaßen aus: *Kapielnummer.Seitenzahl* z.B. 3.4 als 4. Seite des 3. Kapitels.

Vordefinierte Zähler sind in L^AT_EX

- part
- chapter
- section
- subsection
- subsubsection
- paragraph
- subparagraph
- page
- figure
- table
- footnote
- mpfootnote
- equation
- enumi

- enumii
- enumiii
- enumiv

Man kann mit `\newcounter{zählername}` einen neuen Zähler definieren.

enumi - enumiv stellen die ersten 4 Ebenen der enumerate-Umgebung dar. Man kann so Z.B. mit

```
\begin{enumerate}
  \setcounter{enumi}{3}
  \item a
  \item b
  \item c
\end{enumerate}
```

4. a
5. b
6. c

erstellen.

Will man die Darstellungsform von Zählern verändern, kann man das mit den folgenden Befehlen umsetzen:

- `\alph` — Kleinbuchstaben
- `\Alph` — Großbuchstaben
- `\arabic` — Dezimalzahlen mit arabischen Ziffern
- `\roman` — Römisches Zahlensystem mit Kleinbuchstaben
- `\Roman` — Römisches Zahlensystem mit Großbuchstaben

Man kann mit `\renewcommand{\theenumi}{\roman{enumi}}` und

`\renewcommand{\labelenumi}{\textbf{\theenumi}}` neu setzen. Dann sieht die Aufzählung folgendermaßen aus:

- i) erstes
- ii) zweites
- iii) drittes

3.6 Geometry

Um die Ränder des Dokuments anzupassen, kann man das geometry-Paket verwenden. Z.B.:

```
\usepackage[left=3cm,right=3cm,top=2cm,bottom=3cm,includeheadfoot]{geometry}
```

erstellt die Seitenränder dieses Dokuments. Dabei sind 3cm Seitenrand links, rechts und unten und oben 2cm. Im laufenden Dokument kann man mit `\geometry`, `\newgeometry` und `\restoregeometry` das Layout nochmal anpassen.

3.7 Übungen

Aufgabe 1

(4 Punkte)

Schreibe einen Command `\newcommand{\aufgabe}[1]{...}`, der die Formatierung der Aufgaben auf diesem Übungsblatt ausgeben kann. Dabei soll die Nummerierung mithilfe eines Zählers realisiert werden, der bei jeder Aufgabe automatisch hochzählt. Das Argument soll die Anzahl der möglichen Punkte enthalten. D.h. `\aufgabe{2}` soll eine Aufgabe erstellen, die 2 Punkte bringt.

Aufgabe 2

(2 Punkte)

Schreibe einen Command, der eine 3×3 -Diagonalmatrix erstellt. So soll `\diagm{1}{2}{3}` die Matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

erstellen.

Aufgabe 3

(4 Punkte)

Teile eine Datei in 3 Teilbereiche auf, die in jeweils eigenen Dateien stehen. (Befehle, Vorwort, Hauptteil)

Dabei sollen die Seiten des Vorworts in großen römischen Zahlen nummeriert sein und die Seiten des Hauptteils sollen nach folgendem Muster nummeriert werden: `{Nummer der Section}. {Seitenzahl der Section in kleinen römischen Zahlen}`. So soll bspw. die 3. Seite der 2. Section `2.iii` heißen.

Kapitel 4

Präsentationen

Wenn wir mit eine Präsentation mit \LaTeX erstellen wollen, so verwenden wir dafür die Dokumentenklasse `\documentclass{beamer}`.

4.1 Folien erstellen

Folien erstellen wir mit

```
\begin{frame}{Titel der Folie}
  Inhalt der Folie
\end{frame}
```

4.2 Einblendeanimation

Da wir nur .pdf-Dateien erzeugen, haben wir natürlich keine richtige Animation. Wir werden ausschließlich neue Elemente erscheinen lassen. Wir haben dafür verschiedene Möglichkeiten.

1. `\pause` – Damit kann man das Laden der restlichen Folie pausieren und auf der nächsten Folie fortsetzen.
2. `\only<1-3>{Inhalt}` – Zeigt `Inhalt` nur auf den Folien 1 bis 3 an.
3. `\visible<1-3>{Inhalt}` – Blendet `Inhalt` nur auf den Folien 1 bis 3 ein.
4. In `itemize`- und `enumerate`-Umgebungen können wir statt `\item` auch `\item<1-3>` verwenden, um ein Item nur auf den Folien 1 bis 3 anzeigen zu lassen

4.3 Titelfolie

Wir legen mit

```
\title[Kurzer Titel]{Langer Titel,\ \ der sich auch über mehrere \ \ Zeilen erstrecken  
kann}  
\subtitle{Untertitel, falls vorhanden}  
\author[M.\ Ködding]{Marvin Ködding}  
\institute[JLU]{Justus-Liebig-Universität Gießen}  
\date{Datum oder Anlass}  
\begin{document}  
[... ggbf. \setlength und andere Einstellungen ...]  
\begin{frame}  
\titlepage  
\end{frame}  
  
\end{document}
```

ein Beispiel für die Titelfolie fest.

unseren Titel und den Autoren fest. Dann können wir mit `\frame{\titlepage}` die Titelfolie erstellen.

4.4 Layout

Für das Layout verwendet man in der Regel vorgefertigte Templates. Dafür verwenden wir den Befehl `\usetheme{theme}`. Wir haben etliche Vorlagen, aus denen wir einfach unser Wunschlayout auswählen können.

[Hier](#) findest du eine Liste an Themes.