

Kurze Einführung in \LaTeX

zusammengefasst während des \LaTeX -Kurses an der HTW Saar

Hong-Phuc Bui

Sommersemester 2015

Inhaltsverzeichnis

1	Grundbegriffe	4	6.3	Aufzählung	15
1.1	L ^A T _E X-Klasse	4	6.4	Flexible Listen mit enumitem	16
1.2	Befehle	5	6.5	Fußnote und Randnotizen	17
1.3	Umgebung	5	7	Graphik in L^AT_EX	18
1.4	Kommentar	5	7.1	Bilder neben Bilder	19
1.5	Sonderzeichen	5	7.2	Text um die Bilder	20
2	Die Titelei	6	8	Tabelle in L^AT_EX	22
3	Logische Aufteilung des Dokuments	8	8.1	Grundlage	22
4	Maß und Einheiten in Typographie	9	8.2	Ästhetische Tabelle	23
4.1	Einheiten	9	9	Einheitliche Formatierung von Text- Elementen	24
4.2	Gängige horizontale Abstände	9	10	Silbentrennung	25
4.3	Weitere Abstand-Befehlen in Text- Modus	9	11	Mathematische Formeln	26
4.4	Dynamische Abstand	10	11.1	Display-Modus	26
5	Text-Formatierung	11	11.2	Brüche	27
5.1	Schriftstil	11	11.3	Indizes und Exponenten	28
5.2	Schriftgröße	12	11.4	Wurzeln	28
6	Weitere Text-Strukturen	13	11.5	Operatoren	28
6.1	Beschreibung	13	11.6	Matrizen	28
6.2	Nummerierung	14	11.7	Integral und Summe	30

Typographie konventionell

Monospaced Schriften sind \LaTeX -Code, den man in die \LaTeX -Datei eingeben muss. Der Teil innerhalb eines $\langle \rangle$ -Paares ist der Platzhalter. Man muss statt $\langle \text{platzhalter} \rangle$ die richtigen Werte setzen. Diese Werten werden im Kontext erklärt.

Die Nummer am Rand eines Listing sind die Zeilennummer. Man darf sie nicht in der \LaTeX -Datei eingeben. Sie werden zur Referenzierung im Text verwendet.

1 Grundbegriffe

Der unterstehende \LaTeX -Code-Auszug ist ein Beispiel für einen minimalen deutschen Artikel.

```
1 \documentclass[paper=a4]{scrartcl}
2 \usepackage{fontspec}
3 \usepackage[ngerman]{babel}
4
5 \begin{document}
6
7 Test: deutsche Buchstaben
8 und Quote \glqq{}ä Ä ö Ö ü Ü \grqq{}.
9 \end{document}
```

Anhand dieses Beispiel werden die Grundbegriffe erklärt.

1.1 \LaTeX -Klasse

Ein \LaTeX -Dokument wird immer zu einer Klasse zugeordnet. Der Befehl

```
\documentclass[<option>]{<classname>}
```

definiert die Klasse des Dokuments. Ein \LaTeX -Dokument fängt in der Regel mit diesem Befehl an. Die Optionen in der eckigen Klammer steuern das Verhalten der Klasse. Sie können weg lassen, dann können auch die eckigen Klammer wegfallen. Die Table 1.1 zeigt einige Klassen.

EU-Konform (KOMA-Klass)	US-Konform	Type des Dokument
scrartcl	article	kurze Artikel bis gegen 20 Seiten
scrreprt	report	klein Booklet von 20 Seiten bis 100 Seiten
scrbook	book	Bücher, lange Druckwerk ab 70 Seiten

Tab. 1.1: *Dokument-Klassen in \LaTeX*

Es ist zu empfehlen, die EU-Konform Klassen statt der US-Konform-Klassen zu verwenden, auch wenn man in Englisch schreiben will. Diese Klassen bieten vielen Optionen um die Gestaltung des Dokument zu kontrollieren. Z.B Peter möchte eine Ausarbeitung für das Thema „*Der rekursive Satz*“ schreiben. Die Ausarbeitung ist nicht so lang, etwa 15 A4 Seiten. Er möchte die Ausarbeitung zweiseitig ausdrucken. So kann er die Klasse `scrartcl` wie folgt wählen:

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

Die vollständige Liste von Optionen mit vielen Erklärungen kann man in (Kohm und Jeans-Uwe-Morawski 2012) nachlesen.

1.2 Befehle

Der gesamte Aufbau eines Dokumentes wird in \LaTeX mittels Befehlen gesteuert. Befehle haben allgemein die Syntax $\backslash\langle\text{name}\rangle[\langle\text{option}\rangle]\{\langle\text{argument}\rangle\}$. Das obige Listing hat fünf Befehle.

Mit dem Befehl $\backslash\text{documentclass}[\langle\text{Optionen}\rangle]\{\text{Klassen}\}$ wird –wie oben beschrieben– der allgemeine Dokumentenaufbau (z.B. A4 doppelseitig mit passender Nummerierung etc.) gewählt.

Über den Befehl $\backslash\text{usepackage}\{\langle\text{paketname}\rangle\}$ kann sich sich zahlreiche weitere Einstellungen laden. Der Befehl $\backslash\text{usepackage}\{\text{fontspec}\}$ lädt das Paket `fontspec`. Dieses Paket ermöglicht es die Umlauten direkt aus Tastatur einzugeben.

Wir betrachten den nächsten Befehl $\backslash\text{usepackage}[\text{ngerman}]\{\text{babel}\}$ in der dritten Zeile. Der Name des Befehls heißt `usepackage`. Der Befehl hat eine Option, nämlich `[ngerman]`, welche die Sprache des Dokuments als Deutsch definiert. Der Befehl hat genau ein Argument: `\{babel\}`. Der Befehl bindet das Paket `babel` in das Dokument ein, und setzt damit die Sprache des Dokuments auf Deutsch.

Die Befehlen $\backslash\text{glqq}\{\}$ und $\backslash\text{grqq}\{\}$ erzeugen die deutschen Anführungszeichen. Diese Befehlen werden von dem Paket `babel` zur Verfügung gestellt, welches zuvor geladen wurde.

1.3 Umgebung

Umgebungen werden mittels $\backslash\text{begin}\{\langle\text{Umgebungsname}\rangle\}$ und $\backslash\text{end}\{\langle\text{Umgebungsname}\rangle\}$ definiert. Jede \LaTeX -Dokument hat die Umgebung `document`. Der Inhalt des Dokuments befindet sich in dieser Umgebung. Umgebungen können verschachtelt werden.

1.4 Kommentar

Kommentaren in \LaTeX fangen mit einem `%`-Zeichen an und enden am Ende der Zeile. Sie werden von \LaTeX ignoriert.

1.5 Sonderzeichen

Diese Zeichen haben besondere Bedeutungen in \LaTeX -Source und können nicht direkt eingegeben werden. Sie sind:

Name	Ausgabe	Eingabe	Name	Ausgabe	Eingabe
Unterstrich	_	_	kaufmanisches Und	&	\&
Backslash	\	\textbackslash	Raute	#	\#
linke spize Klammer	<	\textless	Prozentzeichen	%	\%
rechte spize Klammer	>	\textgreater	Tilde	~	\textasciitilde
Geschweifte Klammern	{}	\{ \}	Dollar	\$	\\$

Tab. 1.2: Sonderzeichen

2 Die Titelei

Die klassischen Klassen `article`, `report` und `book` stellen drei Befehle zur Verfügung, um die Titelei eines Dokumentes zu definieren:

`\author{<Name>}` definiert den Autor-Name des Dokument.

`\title{<Titel>}` definiert den Titel des Dokuments.

`\date{<Datum>}` definiert das Datum des Dokuments.

Um die Titelei auszugeben muss man den Befehl `\maketitle` in der Umgebung `document` schreiben. Beim Datum kann man den Befehl `\today` verwenden um das aktuelle Datum zu setzen.

Die KOMA-Klassen bieten außerdem noch weitere Befehle an:

`\titlehead{<Titel>}` Header-Zeile der Titel-Seite, hier kann man z.B die Fakultät und Hochschule schreiben, etwa Fakultät für Ingenieurwissenschaften HTW Saar.

`\subject{<Titel>}` Art des Dokumentes, etwa Diplomarbeit, Masterarbeit, etc.

`\title{<Titel>}` Titel des Dokuments.

`\subtitle{<Titel>}` Untertitel des Dokuments.

`\author{<Name>}` Autor des Dokuments.

`\date{<Datum>}` Datum des Dokuments.

`\dedication{<Text>}` die Danksagung, oder auch die Widmung des Dokuments.

`\publishers{<Verlage>}` der Verlage, für Abschlußarbeiten kann alternativ an dieser Stelle aber auch der Betreuer angegeben werden.

Für zweiseitige Dokument kann man noch die Befehle `\uppertitleback{<Text>}` und `\lowertitleback{<Text>}` benutzen, um die Texte auf die Rückseite der Titel-Seite zu setzen.

Man kann auch Befehle zur Text-Formatierung (Siehe Kapitel Text-Formatierung auf der Seite 11) in den Argumenten dieser Befehle verwenden um den Text selbst zu gestalten.

Das unterstehenden Listing zeigt eine Titel-Seite einer Masterarbeit.

```
1 \documentclass[a4paper,BCOR=1cm]{scrreprt}
2 \usepackage{fontspec}
3 \usepackage[ngerman]{babel}
4
5 \titlehead{HTW Saar \hfill Fakultät für Ingenieurwissenschaften}
6 \subject{Masterarbeit}
7
8 \title{Ein Beispiel Title}
9 \subtitle{ein Beispiel Untertitel}
```

```

10 \author{Max Mustermann}
11 \date{30.\,September\,2013}
12
13 \publishers{
14     \begin{tabular}{r@{\,:\,}\l}
15 Erster Betreuer & Dr.~Prof.~Vorname Name \\
16 Zweiter Betreuer & Dr.~Prof.~Vorname Name \\
17     \end{tabular}
18 }
19
20 \begin{document}
21
22 \maketitle
23
24 \end{document}

```

Das Ergebnis wird in Abb. 2.1 gezeigt.

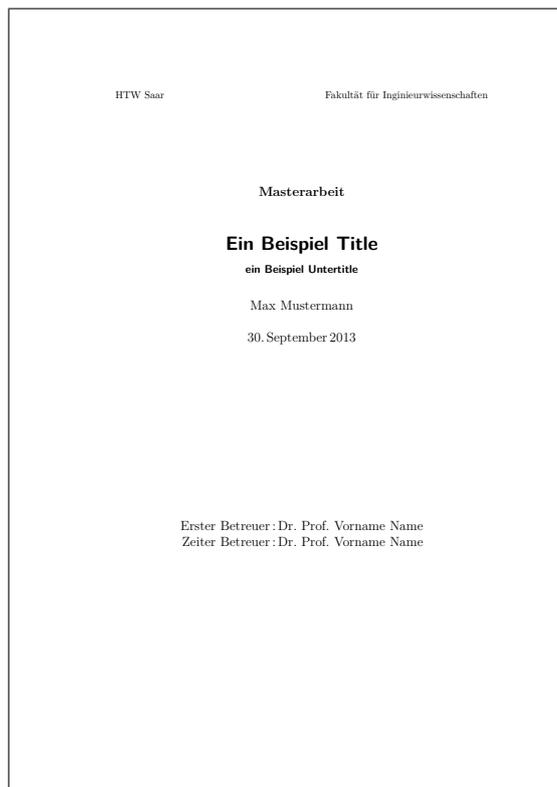


Abb. 2.1: *Titel-Seite einer Masterarbeit*

3 Logische Aufteilung des Dokuments

Die Tabelle 3.1 zeigt die Befehle um das Dokument zu strukturieren.

Der Befehl `\part{}` steht nur in der Klasse `book` bzw. `scrbook` zur Verfügung. Der Befehl `\chapter{}` steht nur in der Klassen `book` und `report` bzw. `scrbook` und `scrrept` zur Verfügung. \LaTeX nummeriert die Gliederungen automatisch. In einem Dokument soll man allerdings die Gliederung nicht tiefer als drei Ebenen nummerieren.

Mit dem Befehl `\setcounter{secnumdepth}{<Nummer>}` kann man die Tiefe der Nummerierung steuern. Das Argument `Nummer` ist der Wert in der Spalte Ebene-Nummer in der Tabelle 3.1. Z. B. mit dem Befehl `\setcounter{secnumdepth}{5}` lässt man \LaTeX auch Paragraphen nummerieren.

Um das Inhaltsverzeichnis auszudrucken verwendet man den Befehl `\tableofcontents`. In der Regel schreibt man diesen Befehl direkt nach dem Befehl `\maketitle`.

Mit dem Befehl `\setcounter{tocdepth}{<Nummer>}` kann man die Tiefe der eingetragenen Gliederung in dem Inhaltsverzeichnis steuern. Er funktioniert ähnlich wie der Befehl `\setcounter{secnumdepth}{<Nummer>}`.

In kurzen Artikeln kann man das Inhaltsverzeichnis zweispaltig setzen. Dazu braucht man das Paket `multitoc`. Konkret muss man den Befehl `\usepackage[toc]{multitoc}` in den Vorspann schreiben.

Die Formatierung von Einträgen im Inhaltsverzeichnis sowie Titel von Kapiteln und Abschnitten wird in der Kapitel ?? erklärt.

Befehl	Ebene	Ebene-Nummer
<code>\part{}</code>	Teil	0
<code>\chapter{}</code>	Kapitel	1
<code>\section{}</code>	Abschnitt	2
<code>\subsection{}</code>	Unterabschnitt	3
<code>\subsubsection{}</code>	Unter-Unterabschnitt	4
<code>\paragraph{}</code>		5
<code>\subparagraph{}</code>		6

Tab. 3.1: Befehle zur Gliederung

4 Maß und Einheiten in Typographie

4.1 Einheiten

Die Tabelle 4.1 zeigt die absolute Einheiten in Typographie. Sie werden um Abstände zu definieren. Diese Einheiten heißen absolute, da sie nicht von Schriftgröße abhängig sind.

pt	Punkt; 1 pt = 0,351 mm	cm	Zentimeter; 2,54 cm = 1 in
pc	Pica; 1 pc = 12 pt	mm	Millimeter; 10 mm = 1 cm
in	Inch; 1 in = 72,27 pt	dd	Didot-Punkt; 1 dd = 0,376 mm
bp	Big Point; 72 bp = 1 in	cc	Cicero; 1 cc = 12 dd
sp	Skaliertpunkt; 65536 sp = 1 pt		

Tab. 4.1: Maß in Typographie, (Berndt 2008)

Neben absolute Einheiten gibt es noch die relativen Einheiten, die von Schriftgröße abhängig sind, in Typographie. Sie sind ex und em. Ein ex ist die Höhe eines kleinen „x“ in der aktuellen Schriftart, ein em ist die Breite eines großen „M“ in der aktuellen Schriftart.

4.2 Gängige horizontale Abstände

- Name Abkürzung
M. Mustermann M. Mustermann
- Abkürzungen des Wortes
bzw. \ kann man... bzw. kann man
- Abstände und keine neue Zeileumbruch
Prof.~Dr.~ Allerwissener Prof. Dr. Allerwissener
- Einheit, Datum
3 \,cm 3 cm
1. \,Oktober~2013 1. Oktober 2013

4.3 Weitere Abstand-Befehlen in Text-Modus

`\hspace{3cm}` erzeugt ein Abstand von 3 cm: | `|\hspace` und `\hspace` haben die „Stern“-Form: `\vspace*` und `\hspace*` Ihre Wirkung werden Sie in der nächsten Übung kennen.[TODO: ergänzen]

4 Maß und Einheiten in Typographie

<code>\enspace</code>	0,5 em
<code>\quad</code>	Ein horizontaler Abstand von 1 em
<code>\qquad</code>	Ein horizontaler Abstand von 2 em
<code>\smallskip</code>	Ein vertikaler Abstand von 3.0pt plus 1.0pt minus 1.0pt
<code>\medskip</code>	Ein vertikaler Abstand von 6.0pt plus 2.0pt minus 2.0pt
<code>\bigskip</code>	Ein vertikaler Abstand von 12.0pt plus 4.0pt minus 4.0pt
<code>\hspace{<len>}</code>	Horizontale Abstand der Länge <len>
<code>\vspace{<len>}</code>	Vertikale Abstand der Länge <len>

4.4 Dynamische Abstand

`\hfill` füllt eine horizontale Abstand bis zum Ende der Zeil. `\hrulefill` Ebenso, aber mit einer Gerade. `\dotfill` Ebenso, aber mit Punkten. `\vfill` füllt

ein vertikalen Abstand bis zum Ende der Seite.

5 Text-Formatierung

In \LaTeX -Dokument kann man mittels der Befehle in der Tabellen 5.1 und 5.2 den Text formatieren. Diese Befehle kann man ineinander verschachteln, um die gewünschte Formatierung zu erhalten. Zum Beispiel um einen kursiven fett gedruckten Text groß zu formatieren schreibt man:

```
{\Large\textit{\textbf{Formatierter Text}}}
```

Das Ergebnis ist: ***Formatierter Text***.

Der Befehl `\emph{<Text>}` hebt den Text hervor, indem er den Text kursiv ausdrückt. Verschachtelt man diesen Befehl, alterniert er den Effekt. Beispiel:

```
Normaler Text \emph{hervorgehobener Text \emph{wieder hervorgehobenen Text}  
Normaler Text} Normaler Text
```

erzeugt „Normaler Text *hervorgehobener Text* wieder hervorgehobenen Text *Normaler Text* Normaler Text“.

5.1 Schriftstil

Mit Argument	unärer Befehl	Schrift
<code>\textrm{text}</code>	<code>{\rmfamily text}</code>	Antiqua / Roman
<code>\textsf{text}</code>	<code>{\sffamily text}</code>	Grotesk / Sans
<code>\texttt{text}</code>	<code>{\ttfamily text}</code>	Schreibmaschine
<code>\textbf{text}</code>	<code>{\bfseries text}</code>	fett, breit laufend
<code>\textmd{text}</code>	<code>{\mdseries text}</code>	normale laufend
<code>\textup{text}</code>	<code>{\upshape text}</code>	aufrecht
<code>\textsl{text}</code>	<code>{\slshape text}</code>	<i>geneigt</i>
<code>\textit{text}</code>	<code>{\itshape text}</code>	<i>krursiv</i>
<code>\textsc{text}</code>	<code>{\scshape text}</code>	KAPITÄLCHEN
<code>\textnormal{text}</code>	<code>{\normalfont}</code>	Brotschrift

Tab. 5.1: Schriftstil, (Marco Daniel u. a. 2012)

5.2 Schriftgröße

Befehl	Schrift
<code>{\tiny winzig kleine Schrift}</code>	winzig kleine Schrift
<code>{\scriptsize sehr klein}</code>	sehr klein
<code>{\footnotesize klein wie Fußnote}</code>	klein wie Fußnote
<code>{\small kleine Schrift}</code>	kleine Schrift
<code>{\normalsize normale Schrift}</code>	normale Schrift
<code>{\large große Schrift}</code>	große Schrift
<code>{\Large größere Schrift}</code>	größere Schrift
<code>{\LARGE sehr große Schrift}</code>	sehr groß
<code>{\huge riesig groß}</code>	riesig groß
<code>{\Huge gigantisch}</code>	gigantisch

Tab. 5.2: *Schriftgröße, (Marco Daniel u. a. 2012)*

6 Weitere Text-Strukturen

Neben der Gliederung bietet \LaTeX noch einige Text-Strukturen um Text zu gestalten. Man braucht keine zusätzlichen Paketen um diese Text-Strukturen zu benutzen.

6.1 Beschreibung

Wenn man einige Begriffe im einem Zusammenhang kurz beschreiben möchte, kann man die `description`-Umgebung benutzen. Der unterstehende Code zeigt einen Anwendungsfall dafür.

```
1 \documentclass{scrartcl}
2 \usepackage{fontspec}
3 \pagestyle{empty}
4 \begin{document}
5 \begin{description}
6   \item[Tiere] sind Lebewesen, die sich bewegen können.
7     \begin{description}
8       \item [Fisch] sind Tiere, die im Wasser leben.
9       \item [Amphibien] sind Tiere, die im Wasser und auf dem Land leben.
10    \end{description}
11   \item[Pflanzen] sind Lebewesen, die  $\text{CO}_2$  in  $\text{O}_2$ 
12     in  $\text{O}_2$  umwandeln können.
13 \end{description}
14 \end{document}
```

Das Ergebnis wird in der Abbildung 6.1 gezeigt.

Tiere sind Lebewesen, die sich bewegen können.

Fisch sind Tiere, die im Wasser leben.

Amphibien sind Tiere, die im Wasser und auf dem Land leben.

Pflanzen sind Lebewesen, die CO_2 in O_2 umwandeln können.

Abb. 6.1: *Die description-Umgebung*

Man kann diese Umgebung ineinander verschachteln, das Ergebnis sieht aber nicht schön. Wenn die Beschreibung größer ist, sie also über einen Absatz ausstreckt, verwendet man statt der `description`-Umgebung die `labeling`-Umgebung. Die `labeling`-Umgebung ist ein wenig komplexer als die `description`-Umgebung.

Das folgende Beispiel zeigt die Anwendung der `labeling`-Umgebung:

```
1 \begin{labeling}[~:]{Pflanzen}
2   \item[Tiere] sind nach biologischem Verständnis eukaryotische
3     Lebewesen, die ihre Energie nicht durch Photosynthese gewinnen und
4     Sauerstoff zur Atmung benötigen, aber keine Pilze sind.
```

6 Weitere Text-Strukturen

```
5 \begin{labeling}{Amphibien}
6   \item [Fische] (Pisces) (von lateinisch piscis „Fisch) sind
7     aquatisch lebende Wirbeltiere, die mit Kiemen atmen. In der
8     zoologischen Systematik bilden die Fische keine natürliche Einheit
9     (Monophylon), \ldots
10  \item [Amphibien] Die Amphibien oder Lurche (Amphibia) sind die
11    stammesgeschichtlich älteste Klasse der Landwirbeltiere (Tetrapoda).
12    Viele Arten verbringen zunächst ein Larvenstadium im Wasser und gehen
13    nach einer Metamorphose zum Leben an Land über. \ldots
14 \end{labeling}
15 \item[Pflanzen] Die Pflanzen (Plantae) bilden ein eigenes Reich innerhalb der
16   Domäne der Lebewesen mit Zellkern und Zellmembran, also der Domäne
17   Eukaryoten. Nach heutigen Schätzungen existieren auf der Erde zwischen
18   rund 320.000 und 500.000 Pflanzenarten, \ldots
19 \end{labeling}
20 \end{document}
```

Das Ergebnis wird in der Abbildung 6.2 gezeigt.

- Tiere : sind nach biologischem Verständnis eukaryotische Lebewesen, die ihre Energie nicht durch Photosynthese gewinnen und Sauerstoff zur Atmung benötigen, aber keine Pilze sind.
- Fische (Pisces) (von lateinisch piscis „Fisch“) sind aquatisch lebende Wirbeltiere, die mit Kiemen atmen. In der zoologischen Systematik bilden die Fische keine natürliche Einheit (Monophylon),...
- Amphibien Die Amphibien oder Lurche (Amphibia) sind die stammesgeschichtlich älteste Klasse der Landwirbeltiere (Tetrapoda). Viele Arten verbringen zunächst ein Larvenstadium im Wasser und gehen nach einer Metamorphose zum Leben an Land über. ...
- Pflanzen : Die Pflanzen (Plantae) bilden ein eigenes Reich innerhalb der Domäne der Lebewesen mit Zellkern und Zellmembran, also der Domäne Eukaryoten. Nach heutigen Schätzungen existieren auf der Erde zwischen rund 320.000 und 500.000 Pflanzenarten,...

Abb. 6.2: Die `labeling`-Umgebung

Wir betrachten den Befehl `\begin{labeling}[~:]{Pflanzen}`. Das erste Argument `{labeling}` ist die Umgebung. Das zweite Argument `[~:]` stellt die Trennung zwischen dem Begriff und seiner Definition. Hier ist also einer geschützter Leerraum (`~`) gefolgt von einem Doppelpunkt als Trennung angegeben. Das dritte Argument `{Pflanzen}` ist der längsten Begriff in der aktuellen `labeling`-Umgebung. An ihm erfolgt die Ausrichtung in der Darstellung.

6.2 Nummerierung

Eine Nummerierung wird mit der Umgebung `enumerate` gesetzt. Jeder Punkt wird nummeriert. Der unterstehende Code zeigt ein Beispiel dafür.

```
1 \begin{enumerate}% Stufe 1
2   \item Punkt 1
```

```

3     \begin{enumerate}% Stufe 2
4     \item Unterpunkt 1
5         \begin{enumerate}
6             \item Unterpunkt, Ebene 2
7             \item Unterpunkt, Ebene 2
8         \end{enumerate}
9     \item Unterpunkt 1
10 \end{enumerate}
11 \item Punkt 2
12     \begin{enumerate}
13         \item Unterpunkt 1
14         \item
15     \end{enumerate}
16 \item Punkt 3
17 \end{enumerate}
18 \end{document}

```

Das Ergebnis wird in der Abbildung 6.3 gezeigt.

1. Punkt 1
 - a) Unterpunkt 1
 - i. Unterpunkt, Ebene 2
 - ii. Unterpunkt, Ebene 2
 - b) Unterpunkt 1
2. Punkt 2
 - a) Unterpunkt 1
 - b)
3. Punkt 3

Abb. 6.3: Die enumerate-Umgebung

Die enumerate-Umgebung kann bis zu drei Ebenen in einander verschachtelt werden.

6.3 Aufzählung

Um eine Liste von Punkten ohne Nummerierung aufzuzählen, verwendet man die `itemize`-Umgebung. Der Beispiel-Code wird im untenstehenden Listing gezeigt.

```

1 \begin{itemize}% Stufe 1
2   \item Punkt 1
3     \begin{itemize}% Stufe 2
4       \item Unterpunkt 1
5       \item Unterpunkt 2
6     \end{itemize}
7   \item Punkt 2

```

6 Weitere Text-Strukturen

```
8 \begin{itemize}
9   \item Unterpunkt 1
10  \item Unterpunkt 2
11  \item Unterpunkt 3
12 \end{itemize}
13 \item Punkt 3
14 \end{itemize}
```

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

Abb. 6.4: Die enumerate-Umgebung

Das Ergebnis in die der Abbildung 6.4 gezeigt.

6.4 Flexible Listen mit enumitem

Um flexible Listen zu verwenden, muss man das Paket `enumitem` in Vorspann laden `\usepackage{enumitem}`.

Um die Standardaufzählungszeichen `•` durch z.B ein Dreieck zu ersetzen, setzt man die Liste wie folgt:

```
1 \begin{itemize}[label=${\triangleright}]
2   \item Punkt 1
3   \item Punkt 2
4 \end{itemize}
```

- ▷ Punkt 1
- ▷ Punkt 2

Abb. 6.5: Dreieck als Aufzählungszeichen

Das Ergebnis wird in der Abb. 6.5 dargestellt.

Wir können auch Listen mit einem Prefix setzen:

```
1 \begin{enumerate}[label=Punkt \roman*]
2   \item Das ist der Punkt Nummer 1 mit römische Zahl.
3   \item Und das ist der Punkt Nummer 2, auch mit römische Zahl.
4 \end{enumerate}
```

Punkt i Das ist der Punkt Nummer 1 mit römische Zahl.

Punkt ii Und das ist der Punkt Nummer 2, auch mit römische Zahl.

Abb. 6.6: *List mit Prefix*

Die Abb. 6.6 zeigt das Ergebnis.

Zu beachten bei der Option [`label=Punkt \roman*`] ist das Sternchen hinter dem Befehl `\roman`. Das Sternchen referenziert auf die aktuelle Ebene der Liste. Neben den Befehlen um die Aufzählung auszugeben kann man auch die Befehle in der Tabellen in dem Abschnitt 5 verwenden, um Text zu formatieren. Hier ist zu beachten, dass man die *unäre*-Form verwenden muss.

Der Befehl `\roman` erzeugt die Aufzählung als römische Zahl. Neben `\roman` kann man `\alph` für alphabetische Nummerierung und `\arabic` für arabische Nummerierung. Mehr Information über das Paket `enumitem` kann man in (Bezos 2009) finden.

6.5 Fußnote und Randnotizen

Fußnoten werden mit dem Befehl `\footnote{<Text>}` erzeugt. Bei einige Zitationsstilen werden die Referenzen als Fußnote geschrieben. Allerdings sie werden in \LaTeX automatisch gesetzt. Der mit dem Befehl ¹ gesetzt Text soll nur für richtige Fußnote verwendet.

Mit dem Befehl `\marginpar{<Text>}` kann man Randnotizen setzen (Siehe Rand).

Text

¹`\footnote{}`

7 Graphik in L^AT_EX

Man braucht das Paket `graphicx` in dem Vorspann zu laden und diversen Bild-Formaten in L^AT_EX einzubinden. L^AT_EX kann mit PDF, PNG, GIF, JPEG, JPG umgehen.

Um ein Bild in einem Dokument einzubinden verwendet man den Befehl `\includegraphics{<path/bildname>}`. Wobei `path` ist der relative Pfad zum L^AT_EX-Dokument. z.B Man hat in dem aktuellen Ordner den Unter-Ordner `img`. In diesen Ordner befindet sich das Bild `Lenna.png`, die man in dem Dokument einbinden möchte, dann schreibt man so: `\includegraphics{img/Lenna.png}` Um die Unter-Ordner zu trennen benutzt man das Symbol `/`.

Damit man auch die Abbildungen im Dokument referenzieren kann, muss man den Befehl `\includegraphics{}` in der Umgebung `figure` platzieren. Die Umgebung `figure` bietet einige Optionen um die Position der Abbildung zu platzieren:

- t Die Abbildung wird an dem oberen Kannten der Seite platziert.
- b Die Abbildung wird an dem unteren Kannten der Seite platziert.
- h L^AT_EX versucht, die Abbildung so nahe wie möglich an der Position zu platzieren, wo man in Quelltext sie platziert.
- p L^AT_EX platziert die Abbildung in einer separaten Seite.

Wenn man das Paket `float` im Vorspann lädt, kann man noch die Option `H` benutzen. Diese Option erzwingt L^AT_EX, die Abbildung genau an der Position zu platzieren, wo man sie auch in Quelltext schreibt.

Man kann mit dem Befehl `\caption` den Titel der Abbildung angeben. Weiter kann man *direkt* nach dem Befehl `\caption` einen `\label` vergeben, damit man später im Text die Abbildung referenzieren kann.

Damit das Bild auch zentriert gesetzt wird, benutzt man noch den Befehl `\centering`. Das gesamte Code sieht dann so aus:

```
1 \usepackage{graphicx}
2 %.....
3 \begin{document}
4 \begin{figure}
5 \centering
6 \includegraphics[scale=0.25]{img/Lenna.png}
7 \caption{Lena, ein sehr bekannte Bild in \textit{Image processing}}
8 \label{fig:lenna-one-img}
9 \end{figure}
10 %....
11 \end{document}
```

Das Ergebnis wird in der Abbildung 7.1 dargestellt.



Abb. 7.1: Lenna, ein sehr bekannte Bild in Image processing

Die Positionierung von Bilder kann man mit zusätzlichen Paketen feiner gestalten. In dem nächsten Abschnitte werden wir diesen Paketen kennenlernen.

7.1 Bilder neben Bilder

Damit man in einer `figure`-Umgebung mehrere Bilder neben einander platzieren und referenzieren kann, braucht man die Paketen `subcaption` und `caption` im Vorspann zu laden. Wo man die Bilder platzieren möchte, kann man wie folgendes schreiben:

```

1 \begin{figure}
2   \begin{subfigure}{0.45\textwidth}
3     \includegraphics[scale=0.20]{img/Lenna.png}
4     \caption{Ohne Rotation}
5     \label{fig:lenna-ohne-rotation}
6   \end{subfigure}
7   \begin{subfigure}{0.45\textwidth}
8     \includegraphics[scale=0.20,angle=45]{img/Lenna.png}
9     \caption{Mit Rotation}
10    \label{fig:lenna-mit-rotation}
11  \end{subfigure}
12  \caption{Lenna}\label{fig:lenna}
13 \end{figure}

```

Das Ergebnis sieht so aus:

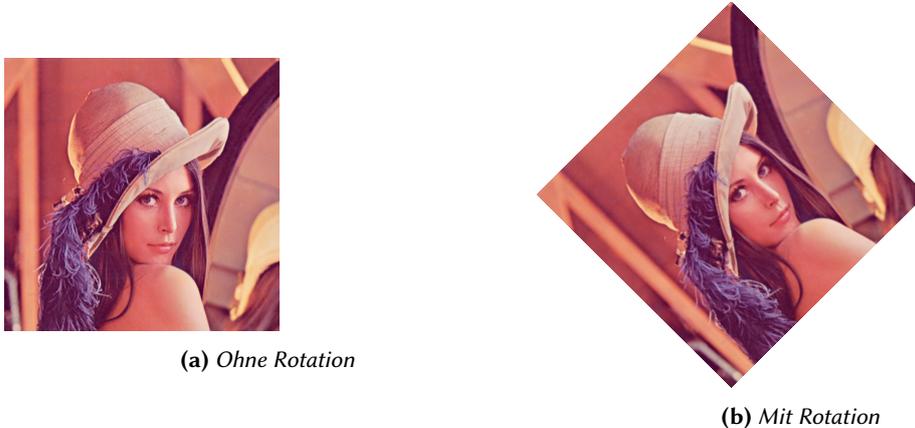


Abb. 7.2: Lenna

7.2 Text um die Bilder

Man kann in \LaTeX die Abbildung neben dem Text platzieren, indem man die Abbildung in der Umgebung `wrapfigure` statt der Umgebung `figure` schreibt. Das Paket `wrapfig` stellt diese Umgebung zur Verfügung. Man muss deshalb auch dieses Paket im Vorspann laden. Die Syntax der Umgebung `wrapfigure` sieht wie Folgendes aus:

```

1 \begin{wrapfigure}[<Zeile>]{<Position>}[<Abstand>]{<Breite>}
2 % Befehlen wie in der Umgebung "figure" kann man hier benutzen.
3 % Z.B \centering, \includegraphics, \caption, \label
4 %...
5 \end{wrapfigure}

```



Abb. 7.3: Der \LaTeX -Löwe

Die Umgebung benötigt zwei pflichten Argumenten und zwei optionalen Argumenten.

Zeile Anzahl der Zeile, die neben der Abbildung gesetzt werden.

Position Die Position der Abbildung. Möglichen Werten sind:

- l (left) Die Abbildung wird an der linken Seite vom Text gesetzt.
- r (right) Die Abbildung wird an der rechten Seite vom Text gesetzt.
- i (in) Die Abbildung wird an der innen Seite vom zweiseitig gesetzten Text gesetzt.
- o (out) Die Abbildung wird an der außen Seite vom zweiseitig gesetzten Text gesetzt.

Abstand Die willkürlichen Abstand zwischen Text und der Abbildung. Es muss eine Länge sein, D.h. Es muss eine Zahl gefolgt von einem Länge-Einheit wie cm, mm, pt, und so weiter.

Breite Die Breite, die für die Abbildung reserviert wird, unabhängig von der Größe der Abbildung.

Beispiel:

```

1 \begin{wrapfigure}[3]{r}[5pt]{4.5cm}
2 \centering

```

```
3 \includegraphics[scale=0.35]{img/ctan_lion_350x350.png}  
4 \caption{Der \LaTeX-Löwe}\label{fig:wrap-fig}  
5 \end{wrapfigure}
```

Das Ergebnis vom obigen Code wird in der Abbildung 7.3 gezeigt.

8 Tabelle in L^AT_EX

8.1 Grundlage

- table-Umgebung definiert Gleitobjekt.
- tabular-Umgebung definiert eigentliche Tabelle.
- Syntax: `\begin{tabular}{<Spaltendefinition>}`

Die Argument <Spaltendefinition> können ein oder mehrere diese Werten enthalten:

c	Center zentriert
l	Left linksbündig
r	Right rechtsbündig
p{Breite}	paragraph feste Breite, Zeilenumbrüche möglich
	vertikale Linie
@{Text}	Text in jeder Zeile (ohne Spaltenabstand)
*{n}{def}	wiederholt def n-mal.

In der Umgebung tabular kann man diesen Befehlen benutzen um die Zeilen und Spalten in der Tabelle abzugrenzen.

&	Spaltentrennung
\\	Zeilentrennung
\hline	Horizontale Linie (ganze Breite)
\cline{n-m}	Horizontale Line Spalten n bis m

Beispiel: Das folgende Code

```
1 \begin{table}[H]
2 \centering
3 \begin{tabular}{|cp{2cm}lr|}\hline
4 Zeitrierte Spalte & Lange Abschnitte mit Umbrüche (2\,cm Breite)
5 & Linksbündige Spalten
6 & Rechtsbündige Spalten \\ \hline
7 aaa & bbbbb
8 & ccccc
9 & dddd \\ \hline
10 \end{tabular}
```

```

11 \caption{Meine erste Tabelle}
12 \label{tab:mein-tab}
13 \end{table}

```

erzeugt diese Tabelle 8.1.

Zeitrierte Spalte	Lange Ab- schnitte mit Umbrüche (2 cm Breite)	Linksbündige Spalten	Rechtebündige Spalten
aaa	bbbb	cccc	dddd

Tab. 8.1: *Meine erste Tabelle*

3 €

4 €

5 €

8.2 Ästhetische Tabelle

Wir können die Gestaltung von Tabelle in \LaTeX verbessern, indem wir das Paket `booktabs` verwenden. Das Paket bittet einige neue Features:

- keine vertikale Linien
- keine doppelten Linien
- etwas mehr Abstand über dem Text
- führt neue Befehle ein:
 - `\toprule` Linien über dem Kopf
 - `\midrule` Linien in der Tabellen
 - `\bottomrule` Linien am Ende der Tabellen
 - `\cmidrule{n-m}` ersetzt `\cline`

Um lange Tabllen zu setzen können wir das Paket `longtable` benutzen. Das Dokument kann man in (Carlisle 2004) finden. Um eine lange Tabele auf mehrere Seiten zu setzen benutzt man die Umgebung `longtable` statt der Umgebung `tabular`. Man setzt ausserdem diese Umgebung nicht in der Umgebung `table`.

9 Einheitliche Formatierung von Text-Elementen

Siehe (Kohm und Jeans-Uwe-Morawski 2012) auf Seite 52, 53.

10 Silbentrennung

Das Paket `babel` kennt meist Silbentrennung. Es gibt aber Fälle, in denen man selbst die Trennung übernehmen muss. In diesen Einzelfällen kann man diese Notationen verwenden (Verweyen o.D):

- Bindestrich, der andere Trennungen unterdrückt:

Mess-Ergebnis ergibt Mess-
Ergebnis.

- "= Bindestrich, der andere Trennungen erlaubt:

Mess"=Ergebnis¹ ergibt Mess-Ergeb-
nis.

- "~ Bindestrich, an dem nicht getrennt werden darf:

Mess"~, Schätz"~ und andere Ergebnisse ergibt Mess-, Schätz- und andere Ergebnis-
se.

- \- Trennmöglichkeit, die andere Trennungen ausschließt:

Ur\-instinkt ergibt Ur-
instinkt.

- "- Trennmöglichkeit, die andere Trennungen nicht ausschließt, wenn \TeX mal was nicht findet: Am"-nestie (das ist übrigens inzwischen behoben).

- "" Trennmöglichkeit, bei der kein Trennstrich benötigt wird:

(Messer"~)""Gabel ergibt (Messer-)
Gabel.

Verwendet man ein Wort sehr häufig, kann man dieses Wort im Vorspann mit dem Befehl `\hyphenation{}` deklarieren (WikiBook 2013):

```
1 \hyphenation{Staats-ver-trag Stau-be-cken}
```

Diese Wörtern dürfen keine Umlauten bzw. ß, und Sonderzeichen enthalten. Man kann damit umgehen indem man einen Befehlen definieren:

```
1 \usepackage{xspace}
2 % ...
3 \begin{document}
4 \newcommand{\dreidStrahlung}{3D"=Strah-lung\xspace}
```

Das Ergebnis ist: `\dreidStrahlung` 3D-
Strah-
lung. Der selbst-definierte Befehl muss nach dem `\begin{document}` platziert werden.

¹Das kann man mal so machen, besser ist es aber, eine Trennung zu nah am Divis zu vermeiden: `Mess-Ergeb"-nis`

11 Mathematische Formeln

TeX kann schon sehr gut mathematische Formeln setzen. Dennoch muss man für einige Strukturen und Symbolen zusätzlichen Pakete laden.

Eines der bekanntesten Pakete ist das Paket `amsmath`. AMS steht für *American Mathematical Society*. Sein Nachfolger ist das Paket `mathtools`. Man kann beide benutzen. Das Paket `mathtools` lädt intern auch das Paket `amsmath`. Es gibt zwei *Modus* um mathematische Formel zu setzen:

inline-Modus ist für die Formel in Fließtext geeignet. Man setzt die Formel zwischen zwei `$`-Zeichen. Z.B. $\sqrt{9} = 3$ wird durch `$$\sqrt{9} = 3$` gesetzt.

display-Modus ist für Formeln, die in einem eigenen Absatz gesetzt werden und ggf. noch referenziert werden. Man setzt die Formel bei dem `display`-Modus in die Umgebung `align` ggf. `align*` oder `\[<Math. Formel>\]`. Die Umgebungen `displaymath`, `eqnarray`, `eqnarray*` sollte man nicht benutzen. Sie führen dazu, dass die Abstände inkonsistent sind.

11.1 Display-Modus

Die Umgebung `align` setzt die mathematische Formel in einem neuen Absatz und vergibt automatisch die Nummerierung der Formel. Die Umgebung `align*` dagegen vergibt keine Nummerierung. Wenn man das Paket `mathtools` benutzt, dann die `\[\]` ist die Abkürzung für `align*`, Beispiele:

Um die Formel zu nummerieren benutzt man die Umgebung `align`. Der Code

```
1 \begin{align}
2   a^2 &= b^2 + c^2 \\
3   \sin(x)^2 + \cos(x)^2 &= 1
4 \end{align}
```

erzeugt:

$$a^2 = b^2 + c^2$$
$$\sin(x)^2 + \cos(x)^2 = 1$$
$$a_1 + a_2 + \dots + a_n = \sum_{i=1}^n a_i$$

Wenn man die Gleichung referenzieren möchte, markiert man die Gleichung mit dem Befehl `\label` und kann sie dann später im Text mittels `\ref` referenzieren. Der Code

```
1 \begin{align}
2 \varphi(x) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x^2}
3 \label{eq:normalverteilung}
4 \end{align}
5
6 Die Dichtefunktion der Standardnormalverteilung wird in der Gleich~\ref{eq:normalverteilung}
7 dargestellt.
```

erzeugt

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \quad (11.1)$$

Die Dichtefunktion der Standardnormalverteilung wird in der Gleich 11.1 dargestellt.

Man kann mehrere Gleichungen in der Umgebung `align` setzen. Jede Gleichung bekommt eine Nummer. Mit dem Befehl `\notag` schaltet man die Nummerierung von individuellen Gleichung ab. Beispiel: Die Gleichungen

$$a = b \quad (11.2)$$

$$b = c$$

$$\Rightarrow a = c \quad (11.3)$$

werden durch diesen Code erzeugt:

```
1 \begin{align}
2     a &= b \ \label{eq:a-b} \\
3     b &= c \ \notag \\
4 \ \Rightarrow a &= c \ \label{eq:a-c}
5 \end{align}
```

Man kann ebenfalls die 11.2. Gleichung und die 11.3. Gleichung durch `\ref{eq:a-b}` und `\ref{eq:a-c}` referenzieren.

Möchte man die Gleichungen linksbündig setzen, kann man das Paket `mathtools` mit der Option `[fleqn]` aufrufen. Die Abb. 11.1 zeigt den Effekt der Option.

Die Transitivität-Gesetz lautet:	
$a = b$	(1)
$b = c$	(2)
$\Rightarrow a = c$	(3)

Abb. 11.1: Linksbündige Gleichung

11.2 Brüche

Brüche werden durch `\frac{Zähler}{Nenner}` erzeugt. Sie können beliebig verschachtelt werden:

Im Inline-Modus wird `\frac{a}{b}` so dargestellt: $\frac{a}{b}$ während er im Display Modus wie in der „Gleichung“ 11.4 dargestellt wird:

$$\frac{a}{b} \quad (11.4)$$

Oft möchte man Brüche wie diese setzen: $\frac{1}{2}$. Dazu braucht man das Paket `nicefrac` und setzt die Bruch mit dem Befehl `\nicefrac{1}{2}`.

11.3 Indizes und Exponenten

Indizes werden durch `_` gesetzt: a_b ergibt a_b . Exponenten werden durch `^` gesetzt: a^b ergibt a^b . Die beiden Operator können kombiniert und verschachtelt werden:

- $a_{\{1\}}^1 + a_{\{2\}}^2 \rightarrow a_1^1 + a_2^2$
- $a_{\{i^1\}} + a_{\{i^2\}} + a_{\{i^3\}} \rightarrow a_{i^1} + a_{i^2} + a_{i^3}$

11.4 Wurzeln

Wurzeln werden durch `sqrt[<basis>]{argument}` gesetzt. Beispiel: $\sqrt[3]{8} = 2$. Bei langen Exponenten sieht es so aus: $\sqrt[\frac{1}{3}]{8} = 8^{\frac{1}{3}} = 8^{\frac{1}{3}}$

Möchtet man den Abstand zwischen 3 und dem Wurzel-Zeichen vergrößern, kann man die folgendermaßen tun: $\sqrt[\leftroot{-3}\uproot{3}]{\frac{1}{3}}{8} \rightarrow \sqrt[\frac{1}{3}]{8}$

leftroot ist der horizontale Abstand zwischen der Wurzel und dem Exponent, *von rechts nach links*.

uproot ist der vertikale Abstand zwischen der Wurzel und dem Exponent, *von unten nach oben*.

Die Argumente für diese Befehle sind ganzen Zahlen, die Einheit sind Point. Alternativ kann man den Befehl `\nicefrac` verwenden: $\sqrt[1/3]{8}$.

11.5 Operatoren

Operatoren sind bekannten Funktionen, die aufrecht gesetzt werden (müssen). Zum Beispiel: Die Funktion Sinus wird als $\sin(x)$ (`\sin(x)`) gesetzt, gängigsten Funktionen sind:

- Cosinus `\cos(x)` $\cos(x)$, Tangenz `\tan(x)` $\tan(x)$, Kotangenz `\cot(x)` $\cot(x)$
- Natürlicher Logarithmus `\ln(x)` $\ln(x)$, Logarithmus zur Basis n `\log_n(x)` $\log_n(x)$
- Real-Teil und Imaginär-Teil: `\Re(z)` $\Re(z)$ und `\Im(z)` $\Im(z)$

11.6 Matrizen

Matrizen werden folgendermaßen erstellt:

```

1 \begin{align}
2 A &= \left[
3   \begin{array}{ccc}
4     a & b & c \\
5     d & f & 2 \\
6     \sin(x) & \cos(x) & \log_{\frac{1}{3}}(x)
7   \end{array}
8   \right]
9 \end{align}

```


11 Mathematische Formeln

```
6 \end{matrix}
7 \right)
8 \end{align}
erzeugt
```

$$\begin{pmatrix} a & b & c \\ d & f & 2 \\ -1 & r & 3 \end{pmatrix} \quad (11.7)$$

Man kann auch die Arten der Klammern kombinieren.

11.7 Integral und Summe

```
1 \begin{align*}
2 \int_{-1}^1 \cos(x) \mathrm{d}x &= \int \\
3 \int \cos(x) dx &= \sin(x) + C \\
4 \end{align*}
erzeugt
```

$$\int_{-1}^1 \cos(x) dx =$$
$$\int \cos(x) dx = \sin(x) + C$$

Während

```
1 \begin{align*}
2 \sum_{i=0}^{\infty} i^2 &= ??? \\
3 \end{align*}
erzeugt
```

$$\sum_{i=0}^{\infty} i^2 = ???$$

Die Grenzen in der Summe- und Integral-Formel werden automatisch angepasst. Alternativ kann man den Befehl `\limits` benutzen:

```
1 \begin{align*}
2 \int\limits_{-1}^1 \cos(x) \mathrm{d}x &= \\
3 \end{align*}
erzeugt
```

$$\int_{-1}^1 \cos(x) dx =$$

Literaturverzeichnis

- Berndt, Tobias (2008). *LaTeX. Der typographische Einstieg*.
- Bezoz, Javier (2009). *Customizing lists with the enumitem package*. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/enumitem>.
- Carlisle, David (2004). *The longtable package*. URL: <http://www.ctan.org/pkg/longtable>.
- Kohm, Markus und Jeans-Uwe-Morawski (2012). *KOMA-Skript. ein wandelbares LaTeX_ε-Paket*. URL: <http://www.ctan.org/pkg/koma-script>.
- Marco Daniel u. a. (2012). *LaTeX 2_ε Kurzbeschreibung*. Es gibt auch eine englische Version. URL: <http://mirrors.ctan.org/info/lshort/german/l2kurz.pdf>.
- Verweyen, Georg. *Silbentrennung & Co*. URL: <http://homepage.ruhr-uni-bochum.de/georg.verweyen/silbentrennung.html>.
- WikiBook (2013). *LaTeXWörterbuch: Silbentrennung*. URL: http://de.wikibooks.org/wiki/LaTeX-W%C3%83%C2%B6rterbuch:_Silbentrennung#Globale_Silbentrennungsanweisung (besucht am 02. 12. 2013).