

B1B

Back in Business L^AT_EX

Thomas Zenger
mail@thomas-zenger.de
23.11.2016

Inhaltsverzeichnis

1	Vorwort	8
1.1	B1B - Was ist das?	8
1.2	Motivation	8
1.3	Literatur und Grundlagen	8
2	Allgemeines	9
2.1	Präambel	9
2.2	Befehle	9
2.3	Umgebung	9
2.4	Key-Value-Interface	9
2.5	Kommentare	9
2.6	Leerzeichen	10
3	Dokumentaufbau	11
3.1	minimaler Dokumentenaufbau	11
3.2	Deklaration Dokumentenklasse	11
3.3	Standard-Dokumentenklassen	11
3.4	Gliederungsebenen	12
3.5	Optionen der Standardklassen	12
4	Pakete	13
4.1	Laden der Pakete	13
4.2	Wichtige Pakete	13
4.2.1	babel	13
4.2.2	color	13
4.2.3	graphicx	14
4.2.4	inputenc	15
4.2.5	listings	15
4.2.6	longtable	15
4.2.7	multicol	16
5	Titelseite	17
6	Dokumentenunterteilung	18
6.1	Kapitel	18
6.2	Inhaltsverzeichnis	18
6.3	Unterteilung in mehrere Dateien	19
7	Schrift	20
7.1	Schriftfamilie	20
7.2	Schriftgröße	20

7.3	Schriftstil & Hervorhebung	21
8	Umbruch	22
8.1	Blocksatz	22
8.2	Trennungshilfen	22
8.3	Zeilenumbruch & Absatz	22
8.4	Seitenumbruch	23
9	Abstand	25
10	Ausrichtung	26
10.1	Zentrieren	26
10.2	Flatterrand	26
11	Zitate	27
12	Verweise und Bemerkungen	28
12.1	Fußnoten	28
12.2	Randbemerkung	28
12.3	Marken	28
12.4	Querverweise	29
13	Listen	30
13.1	einfache Listen	30
13.2	Numerierte Listen	30
13.3	Beschreibung	31
14	Tabellen	32
14.1	Tabulator - <code>tabbing</code>	32
14.2	Tabellen - <code>tabular</code>	32
15	Quellcode Darstellung	34
16	Boxen	35
16.1	Konzept	35
16.2	Arten	35
16.3	Rahmen	36
16.4	Parbox	36
16.5	Linien & Balken	37
16.6	Verschieben von Boxen	37
17	Gleitobjekte	38
17.1	Abbildungen	38
17.2	Tabellen	38
17.3	<code>minipage</code>	39
18	Eigene Befehle	40

19 Verzeichnisse	41
19.0.1 Anhang	41
20 Steuerzeichen & Sonderzeichen	42
20.1 Griechische Buchstaben	42
20.2 Interpunktions-, Satzzeichen und Klammern	44
20.3 Sonderzeichnen & Symbole	45
21 Changelog	46

Abbildungsverzeichnis

11.1 Verse (Beispiel)	27
13.1 Listen Beispiele	31
16.1 Parbox Beispiele	37

Tabellenverzeichnis

3.1	Gliederungsebenen der Klassen	12
5.1	Titelseite (Befehle)	17
6.1	Kapitel & Inhaltsverzeichnis (Befehle)	18
8.1	Trennungsbefehle	22
8.2	Umbruch (Befehle)	23
8.3	Automatischer Seitenumbruch	23
8.4	Manueller Seitenumbruch	24
12.1	Verweise (Befehle)	29
14.1	tabular-Beispiel	33
19.1	Verzeichnisse	41
20.1	Auswahl an griech. Klein- & Großbuchstaben	43
20.2	Auswahl Satzzeichen usw.	44
20.3	Symbole	45

Listings

2.1	Umgebung	9
3.1	Minimaler Dokumentenaufbau	11
4.1	Babel, globale Optionen	13
4.2	Color, Farbdefinition	13
4.3	Listings Beispiele	15
4.4	Mulicol-Umgebung	16
7.1	Emph-Hervorhebung	21
11.1	Quote (Befehle)	27
12.1	Fußnoten (Befehle)	28
12.2	Randbemerkung (Befehle)	28
13.1	itemize-Liste	30
13.2	enumerate-Liste	30
13.3	itemize-Liste	31
14.1	tabbing-Beispiel	32
14.2	tabular-Beispiel	33
15.1	Verbatim (Befehle)	34
16.1	Rahmenbeispiele	36
16.2	Parbox Befehle	36
17.1	Beispiele Abbildung & Tabelle	38
17.2	minipage-Umgebung	39
18.1	Befehl mit Parametern	40

1 Vorwort

1.1 B1B - Was ist das?

Dieses Skript soll als Nachschlagewerk und Cheat Sheet dienen und eine Gedächtnisstütze zu einem bereits vertieften Thema bieten. Es soll keine Fachliteratur ersetzen und dient auch nicht zum Erlernen der Thematik, da auf ausführliche Erklärungen größtenteils verzichtet wird.

1.2 Motivation

Diese Idee zu diesem Kompendium entstand während meines Informatikstudiums. Da bereits erlernte Techniken, wie z.B. Programmier- oder Scriptsprachen, mit dem Fortschreiten des Studiums in den Hintergrund traten, zu einem späteren Zeitpunkt jedoch wieder benötigt wurden, war es unerlässlich sich diese wieder ins Gedächtnis zu rufen. Aus diesem Grund entstand dieses Werk als eine Art “erweiterte Zusammenfassung”.

1.3 Literatur und Grundlagen

Folgende Werke fanden bei der Erstellung dieses Dokuments Beachtung. An dieser Stelle soll ausdrücklich erwähnt werden, dass sich diese Arbeit nicht als Plagiat oder Kopie genannter Literatur verstanden werden soll, sondern als private Lernhilfe und Zusammenfassung.

- M. Öchsner, A. Öchsner (2015) - Das Textverarbeitungssystem LaTeX
- T. Oetiker (2005) - The Not So Short Introduction to L^AT_EX 2 _ε
- M. Jürgens, T. Feuerstack (2013) - L^AT_EX - Eine Einführung und ein bisschen mehr...
- M. Jürgens (2011) - L^AT_EX - Fortgeschrittene Anwendung
- <https://ctan.org>
- www.dante.de

2 Allgemeines

2.1 Präambel

Den Teil zwischen der Festlegung der Dokumentenklasse und dem Beginn des Dokuments nennt man Präambel. In der Präambel werden Pakete geladen und eigene Befehle definiert.

2.2 Befehle

Grundsätzlich beginnen $\text{\LaTeX}$ -Befehle mit einem Backslash ($\backslash$) gefolgt vom eigentlich Namen des Befehls. Beendet wird der Befehl durch ein Leerzeichen oder Sonderzeichen. Einigen Befehlen können Parametern übergeben werden. Verpflichtende Parameter werden üblicherweise in geschweiften Klammern übergeben ($\backslash\text{befehlsname}\{\text{parameter}\}$). Manche Befehle erwarten die Parameterangabe in eckigen Klammern. Diese Parameter sind optional. Sollten Befehle im Fließtext verwendet werden, kann mit abschließenden geschweiften Klammern ein Leerzeichen erzeugt werden.

2.3 Umgebung

Ein zentrales Element in $\text{\LaTeX}$ sind Umgebungen (engl. Environment). Mit Umgebungen lassen sich Effekte auf einen bestimmten Textbereich begrenzen oder spezielle Elemente umschließen.

```
1 \begin{umgebungsname}  
2 Inhalt  
3 \end{umgebungsname}
```

Listing 2.1: Umgebung

2.4 Key-Value-Interface

Mit dem Graphics Bundle von David Carlisle wurde das *keyval*-Package eingeführt. Dieses Interface regelt die Strukturierung von Parametern. Jeder Parameter besteht aus einem Key und einem zugehörigem Wert (Value). Viele Pakete benutzen dieses Interface und ermöglichen damit die Übergabe von “key=value“-Paaren.

2.5 Kommentare

$\text{\LaTeX}$ kennt als Standard nur den einzeiligen Kommentar. Dieser wird mit dem Prozentzeichen (%) eingeleitet. Mehrzeilige Kommentare können in $\text{\LaTeX}$ mit mehreren einzeiligen Kommentaren realisiert werden.

2.6 Leerzeichen

$\text{\LaTeX}$ behandelt mehrere Leerzeichen wie ein Leerzeichen. Sollte dies nicht gewünscht sein, muss mit entsprechenden $\text{\LaTeX}$ -Befehlen nachgeholfen werden. Mehrere Leerzeilen werden ebenfalls nach obigen Prinzip behandelt.

3 Dokumentenaufbau

3.1 minimaler Dokumentenaufbau

`\documentclass`

Legt fest in welcher Art und Weise ein Dokument erzeugt wird. Hier festgelegte Eigenschaften gelten für das gesamte Dokument.

`\begin{document}`

Mit diesem Befehl wird die Dokumentenumgebung eingeleitet. Es folgen der eigentliche Text des Dokumentes und die Formatierungsbefehle.

`\end{document}`

Beendet die Dokumentenumgebung. $\text{\LaTeX}$ wird alle nachfolgenden Eingaben ignorieren.

```
1 \documentclass{article}
2 \begin{document}
3     Hallo minimaler Aufbau!
4 \end{document}
```

Listing 3.1: Minimaler Dokumentenaufbau

3.2 Deklaration Dokumentenklasse

`\documentclass[<optionen>][<klasse>]`

Es muss genau eine Dokumentenklasse angegeben werden. Optionen dürfen mehrere angegeben werden. Diese sind durch Kommata zu trennen.

3.3 Standard-Dokumentenklassen

In der Standardinstallation sind bereits einige Grunddokumententypen enthalten. Dokumententypen des Koma-Script-Pakets passen die Dokumentenklassen an die Bedürfnisse der deutschen/europäischen Nutzern an.

Klassen

<code>article</code>	Dokumentklasse für kurze, technische Artikel
<code>book</code>	Dokumentklasse für das Schreiben von Büchern
<code>report</code>	Dokumentklasse für längere technische Artikel
	Zwischenstufe zwischen <i>article</i> und <i>book</i>
<code>letter</code>	Dokumentklasse zum Schreiben von Briefen

<code>scrartcl</code>	KOMA-Script Äquivalent zu <i>article</i>
<code>scrbook</code>	KOMA-Script Äquivalent zu <i>book</i>
<code>srcrprt</code>	KOMA-Script Äquivalent zu <i>report</i>
<code>scrlttr2</code>	KOMA-Script Äquivalent zu <i>letter</i>
<code>beamer</code>	Dokumentklasse für Präsentationen
<code>moderncv</code>	Lebensläufe

3.4 Gliederungsebenen

article	book	report	letter	Level
	<code>\chapter</code>	<code>\chapter</code>		0
<code>\section</code>	<code>\section</code>	<code>\section</code>	<code>\signature</code>	1
<code>\subsection</code>	<code>\subsection</code>	<code>\subsection</code>	<code>\address</code>	2
<code>\subsubsection</code>	<code>\subsubsection</code>	<code>\subsubsection</code>	<code>\opening</code>	3
<code>\paragraph</code>	<code>\paragraph</code>	<code>\paragraph</code>	<code>\closing</code>	4
<code>\subparagraph</code>	<code>\subparagraph</code>	<code>\subparagraph</code>		5

Tabelle 3.1: Gliederungsebenen der Klassen

3.5 Optionen der Standardklassen

Einige Beispiele für Optionen. Genauer ist in den Klassendokumentationen nachzulesen.

Papiergröße	letterpaper, a4paper
Titelblatt	titlepage, notitlepage
Schriftgröße	10pt, 11pt, 12pt
Papierausrichtung	portrait, landscape
Drucklayout	oneside, twoside
Spaltenlayout	onecolumn, twocolumn
Math. Gleichungen	fleqn, leqno

4 Pakete

4.1 Laden der Pakete

`\usepackage[<optionen>]{<paket>}`

Mit dem `usepackage`-Befehl können Pakete zur Verwendung geladen werden. Pakete stellen zusätzliche Funktionalität bereit. Es können auch mehrere Pakete gleichzeitig geladen werden. Allerdings ist hierbei eine Parameterübergabe nicht mehr möglich bzw. gelten die übergebenen Parameter für alle Pakete.

4.2 Wichtige Pakete

4.2.1 babel

`\usepackage[<optionen>]{babel}`

Babel ist ein Sprachpaket für $\text{\LaTeX}$. Es gibt an in welchen Sprachen ein Dokument geschrieben ist. Damit andere Pakete auch Zugriff auf die Funktionalität von *Babel* haben ist es ratsam die Sprachen als *globale Optionen*. Wird mehr als eine Sprache angegeben, kann mit der *main*-Option die Hauptsprache explizit gesetzt werden. Mit dem Makro `\selectlanguage{<sprache>}` können zwischen den Sprachen gewechselt werden. Ebenfalls kann dies mittels einer Umgebung geschehen. Mehr hierzu in der Dokumentation von *Babel*¹.

```
1 \documentclass[ngerman]{book}
2 \usepackage{babel}
```

Listing 4.1: Babel, globale Optionen

4.2.2 color

`\usepackage{color}`

Das Paket *Color* erlaubt es Farben im Dokument zu benutzen. Es existieren bereits vordefinierte Farbwerte wie `black`, `white`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow`. Sollten andere Farbwerte gewünscht werden, müssen diese zuerst definiert werden. Mehr hierzu in der Dokumentation von *Babel*².

```
1 \definecolor{<name>}{<model>}{<spezifikation>}
2 %Beispiel
3 \definecolor{light-blue}{rgb}{0.8,0.85,1}
```

Listing 4.2: Color, Farbdefinition

¹<http://ftp.uni-erlangen.de/ctan/macros/latex/required/babel/base/babel.pdf>

²<http://vesta.informatik.rwth-aachen.de/ftp/pub/mirror/ctan/macros/latex/required/graphics/grfguide.pdf>

Um Text farblich hervorzuheben gibt es folgende Möglichkeiten:

Befehl	Erklärung
<code>\color{<name>}</code>	Färbt Text bis zum Ende der Umgebung
<code>\textcolor{<name>}{<text>}</code>	Färbt angegebenen Text
<code>\pagecolor</code>	Setzt Hintergrundf. für alle folg. Seiten
<code>\nopagecolor</code>	Setzt weiße Hintergrundfarbe
<code>\colorbox{<name>}{<text>}</code>	Setzt farbige Box um Text
<code>\fcolorbox{<name2>}{<name1>}{<text>}</code>	Setzt farbige Box mit Rahmen um Text

4.2.3 graphicx

`\usepackage[<optionen>]{graphicx}`

Mit dem *Graphics*-Paket ist es möglich externe Grafiken einzubinden. Um einen schnellen Kompiliervorgang zu erreichen können einige Optionen gesetzt werden.

Option	Erklärung
<code>draft</code>	Grafiken werden nicht geladen
<code>final</code>	Gegenteil von <code>draft</code>
<code>hiderotate</code>	Verberge gedrehten Text
<code>hidescale</code>	Verberge skalierten Text
<code>demo</code>	Ersetzt Grafik durch Rechteck

Rotieren

`\rotatebox[<key val list>]{<angle>}{<text>}`

Ohne gesetzte Optionen rotiert der Text immer bezüglich des Standard Referenz Punkt der Box.

Skalieren

`\scalebox{<h-scale>}[<v-scale>]{<text>}`

Skaliert den angegebenen Text. Falls kein vertikaler Wert explizit gesetzt wurde, ist dieser gleich dem horizontalen Wert.

`\reflectbox{<text>}`

Äquivalent zu `\scalebox{-1}[1]{<text>}`

Grafik einbinden

`\includegraphics*[<key val list>]{<file>}`

Option	Erklärung
<code>height</code>	Grafik wird in der Höhe skaliert
<code>width</code>	Grafik wird in der Breite skaliert
<code>angle</code>	Grafik wird gedreht

Weitere Optionen können der offiziellen Dokumentation des *Graphicx*-Pakets³ entnommen werden.

4.2.4 inputenc

`\usepackage[<encoding name>]{inputenc}`

Das Paket *Inputenc* erlaubt dem Nutzer die Zeichenkodierung des Dokumentes festzulegen.

Unterstützte Kodierung

ascii
latinX
utf8

Erklärung

ASCII Code range 32-127
ISO Latin Encoding (X für 1 bis 10)
Unicode UTF-8 Encoding

4.2.5 listings

`\usepackage[<optionen>]{listings}`

Mit dem Paket *Listings* ist es möglich nicht formatierten Text dem Dokument hinzuzufügen. Der Unterschied zur *Verbatim*-Umgebung ist, dass das Hauptaugenmerk auf dem Hinzufügen von Source Code liegt.

```

1 %Listings Umgebung
2 \begin{lstlisting}[captionBeschreibung]
3 SourceCode
4 \end{lstlisting}
5
6 %SourceCode aus einer Datei importieren
7 \lstinputlisting{source_filename.py}
8
9 %SourceCode aus einer Datei importieren mit Angabe der Sprache
10 \lstinputlisting[language=Phyton]{source_filename.py}
11
12 %Nur angegebene Zeilen des SourceCodes darstellen
13 \lstinputlisting[language=Phyton, firstline= 10, lastline=15]{source_
    filename.py}
14
15 %Listings werden mit Zeilennummern und einfachen Rahmen dargestellt
16 \lstset{numbers=left, frame=single}

```

Listing 4.3: Listings Beispiele

Mit dem Befehl `\lstset{}` können globale Parameter gesetzt werden. Für eine komplette Liste aller möglichen Optionen kann auf die offizielle Dokumentation⁴ zurückgegriffen werden.

4.2.6 longtable

`\usepackage{longtable}`

Mit Hilfe des *longtable*-Paketes ist es möglich Tabellen zu erstellen, welche durch $\text{\LaTeX}$ auch umgebrochen werden können. So ist es möglich mehrseite Tabellen zu erstellen. Es

³<http://vesta.informatik.rwth-aachen.de/ftp/pub/mirror/ctan/macros/latex/required/graphics/grfguide.pdf>

⁴<http://ftp.gwdg.de/pub/ctan/macros/latex/contrib/listings/listings.pdf>

benutzt den selben `counter` wie die `table`-Umgebung und wird ebenfalls im Tabellenverzeichnis aufgelistet. Damit $\text{\LaTeX}$ die Tabellen korrekt darstellt sollte der Compiler mehrfach drüberlaufen. Die Definition der Tabelle erfolgt analog zur `tabular`-Umgebung.

Befehl	Erklärung
<code>\endhead</code>	Head, auf jeder Seite
<code>\endfirsthead</code>	Head, nur für die 1. Seite
<code>\endfoot</code>	Footer, jeder Seite
<code>\endlastfoot</code>	Footer, der letzten Seite
<code>\caption{text}</code>	Beschreibung
<code>\caption*{text}</code>	Beschreibung, non lot
<code>\setlongtables</code>	Obsolete
<code>\begin{longtable}[pos]</code>	c, l, r erlaubt
<code>\pagebreak</code>	alle Seitenumbruch Befehle erlaubt

4.2.7 multicols

`\usepackage{multicols}`

$\text{\LaTeX}$ erlaubt bereits das Wechseln des Layouts mithilfe von `\twocolumn` und `\onecolumn`. Dies ist allerdings auch immer mit dem Beginn einer neuen Seite verbunden. Anders ist dies mit dem Paket *multicols* möglich. Die Spalten dürfen allerdings keine Gleitobjekte enthalten.

```

1 %Umgebung
2 \begin{multicols}{spaltenanzahl}[Titel]
3     text
4 \end{multicols}
5
6 %Umgebung ohne Balancing
7 \begin{multicols*}
8
9 %Spaltenabstand
10 \columnsep{breite}
11
12 %Strichbreite
13 \columnseprule{breite}
14
15 %Manueller Spaltenumbruch
16 \columnbreak

```

Listing 4.4: Mulicol-Umgebung

5 Titelseite

L^AT_EX stellt Methoden zur Verfügung eine Titelseite für das Dokument zu erstellen. Im Top Matter (direkt nach `\begin{document}`) können Informationen hierzu bereitgestellt werden.

<code>\title{titel}</code>	Setzt Titel
<code>\date{datum}</code>	Setzt Datum Leeres Argument, falls kein Datum gewünscht
<code>\today</code>	Aktuelles Datum
<code>\author{}</code>	Setzt Autor
<code>\and</code>	mehrere Autoren
<code>\thanks{}</code>	Danksagung, Fußnote
<code>\maketitle</code>	Erstellt Titelseite

Tabelle 5.1: Titelseite (Befehle)

Mit Hilfe der Umgebung `titlepage` kann man eigene Titelseiten erstellen. Dabei muss allerdings das komplette Layout selbst angepasst werden. Das Paket *titling* ist zum Anpassen des `\maketitle`-Befehls geeignet.

6 Dokumentenunterteilung

6.1 Kapitel

Erstellte Überschriften werden von $\text{\LaTeX}$ automatisch durchnummeriert und hervorgehoben. Folgende Befehle können zur logischen Aufteilung des Dokuments genutzt werden:

- `\chapter` (NUR *BOOK* UND *REPORT*)
- `\section`
- `\subsection`
- `\subsubsection`
- `\paragraph`
- `\subparagraph`

6.2 Inhaltsverzeichnis

Überschriften werden automatisch ins Inhaltsverzeichnis übernommen. Um dieses aktuell zu halten ist ein zweifaches Kompilieren des Dokuments von Nöten. Mit `tocdepth` wird die Tiefe des Verzeichnisses festgelegt (1 entspricht der obersten Ebene). Desweiteren kann die Darstellung der Überschriften im Inhaltsverzeichnis beeinflusst werden. Der Inhalt des Inhaltsverzeichnis wird in eine externe Datei ausgelagert (*.toc*).

<code>\tableofcontents</code>	Erstellt Inhaltsverzeichnis
<code>\setcounter{tocdepth}{Wert}</code>	Setzt Tiefe
<code>\befehl{Überschrift}</code>	Überschrift ins Inhaltsverzeichnis
<code>\befehl*{Überschrift}</code>	Überschrift nicht ins Inhaltsverzeichnis
<code>\befehl[Alter]{Überschrift}</code>	Alter ins Inhaltsverzeichnis

Tabelle 6.1: Kapitel & Inhaltsverzeichnis (Befehle)

6.3 Unterteilung in mehrere Dateien

Es ist möglich ein Dokument aus mehreren einzelnen Dateien zusammenzusetzen. Die eingefügte Datei wird als normaler $\text{\LaTeX}$ -Quellcode interpretiert. Die eingefügten Dateien dürfen keine Dokumentendeklaration enthalten. Im Gegensatz zum `\include`-Befehl, darf der `\input`-Befehl verschachtelt werden. Die Dateiendung sollte angegeben werden, da $\text{\LaTeX}$ sonst erst nach einer Datei ohne Endung sucht, bevor *.tex* angehängt wird.

Verwendung `\input(pfad/datei.tex)`

7 Schrift

L^AT_EX unterstützt die Verwendung von unterschiedlicher Schriftarten. Die Voreinstellungen sind 10 Punkte Schriftgröße (1 Punkt entspricht ca. 0,35mm) und Roman Schrift (mit Serifen).

7.1 Schriftfamilie

In L^AT_EX stehen drei verschiedene Schriftfamilien zur Verfügung.

Proportionalschrift	Nichtproportionalschrift
• Serif • Sans Serif	• Type Writer

Diese können mit folgenden Befehlen deklariert werden. Welche Schrift dabei gewählt wird hängt von den Standardeinstellungen (Installation, Dokumentenklasse, Präambel) ab.

<code>\textrm{text}</code>	Schrift mit Serifen	<code>\rmfamily</code>
<code>\textsf{text}</code>	Schrift ohne Serifen	<code>\sffamily</code>
<code>\texttt{text}</code>	Schreibmaschinenschrift	<code>\ttfamily</code>

7.2 Schriftgröße

In L^AT_EX spezifizieren folgende Befehle die Schriftgröße. Dies erfolgt relativ zur Grundschriftgröße. Eine Änderung der Schriftgröße kann auch mit Hilfe einer Umgebung erfolgen.

<code>\tiny</code>	Schriftgröße
<code>\scriptsize</code>	Schriftgröße
<code>\footnotesize</code>	Schriftgröße
<code>\small</code>	Schriftgröße
<code>\normalsize</code>	Schriftgröße
<code>\large</code>	Schriftgröße
<code>\Large</code>	Schriftgröße
<code>\LARGE</code>	Schriftgröße

`\huge` Schriftgröße
`\Huge` Schriftgröße

7.3 Schriftstil & Hervorhebung

Außer der Schriftfamilie hat ein Font die Attribute Schriftstärke (*series*) und Schriftform (*shapes*). Die Befehle zur Änderung der Schriftstärke beeinflussen die Stärke der Linien und die Laufweite (*weight & width*). Bei Änderung der Schriftform wird die Ausrichtung der Buchstaben geändert.

Series

`\textbf{text}` **Fette Schrift**

`\textmd{text}` Normale Schrift

Shapes

`\textup{text}` Aufrechte Schrift (upright)

`\textsl{text}` *Schräge Schrift (slanted)*

`\textit{text}` *kursive Schrift (italic)*

`\textsc{text}` SCHRIFT IN KAPITÄLCHEN (SMALLCAPS)

`\textnormal{text}` Grundschrift des Dokuments

Man kann L^AT_EX auch die Standardhervorhebung überlassen. L^AT_EX ermittelt dann selbstständig welche Hervorhebungsform verwendet wird. Hierzu wird der `\emph`-Befehl verwendet.

```
1 \textit{Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
2 diam \emph{nonumy} eirmod invidunt ut labore et dolore magna aliquyam.}
3
4 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
5 invidunt ut labore et dolore magna aliquyam.
```

Listing 7.1: Emph-Hervorhebung

8 Umbruch

8.1 Blocksatz

Der Standard-Satz in $\text{\LaTeX}$ ist der Blocksatz. Hierbei wird der Rand ausgeglichen, Umbrüche werden gesetzt und eine automatische Aufteilung der Wörter wird vorgenommen. Falls nötig erfolgt auch die Trennung eines Wortes.

8.2 Trennungshilfen

$\text{\LaTeX}$ übernimmt normalerweise automatisch die Trennung von Wörtern. Bei unbekannten Wörtern oder Fremdwörtern kann das allerdings zu fehlerhafter Trennung führen. Durch das Setzen von Trennhilfen kann man $\text{\LaTeX}$ mitteilen, an welcher Stelle ein Wort getrennt werden soll oder eine Trennung explizit untersagen.

<code>\hyphenation{trenn-ung}</code>	Wort wird an angegebener Stelle getrennt Deklaration in der Präambel
<code>\-</code>	Trennung erfolgt nur an dieser Stelle (einmalig)
<code>''-</code>	Trennung darf auch an anderen Stellen erfolgen (Paket <i>babel</i>)
<code>\mbox{wort}</code>	Wort wird nicht getrennt Es erfolgt keine Anpassung der Wortabstände
<code>~</code>	Es erfolgt keine Trennung (hartes Blank)

Tabelle 8.1: Trennungsbefehle

8.3 Zeilenumbruch & Absatz

$\text{\LaTeX}$ organisiert den Text in Absätze welche vertikal auf den Seiten platziert werden. Folgende Befehle dienen nicht dazu das Design zu verbessern. Hierzu gibt es Möglichkeiten die Abstände anzupassen.

<code>\\[vAbstand]</code>	Zeilenumbruch, zwangsweise Kein Absatzende)
<code>*[vAbstand]</code>	Verhindert einen Zeilenumbruch
<code>\newline</code>	Erzeugt einen Zeilenumbruch Kein Absatzende
<code>\linebreak[Priorität04]</code>	möglicher Zeilenumbruch Zeile vor Umbruch bis zum Rand gestreckt.
<code>\nolinebreak[Priorität04]</code>	möglichst keinen Zeilenumbruch
Leerzeile	Neuer Absatz
<code>\par</code>	Neuer Absatz

Tabelle 8.2: Umbruch (Befehle)

8.4 Seitenumbruch

Wie schon erwähnt kümmert sich $\text{\LaTeX}$ selbstständig um etwaige Umbrüche von Zeilen. Selbiges trifft ebenfalls auf Seitenumbrüche zu. Auch hier kann man dies mit den entsprechenden Befehlen manipulieren.

Automatische Steuerung

<code>\flushbottom</code>	Vertikaler Ausgleich an, alle Seiten gleich lang
<code>\raggedbottom</code>	Vertikaler Ausgleich aus

Tabelle 8.3: Automatischer Seitenumbruch

Manuelle Steuerung

<code>\newpage</code>	Seitenumbruch, zwangsweise Spaltenumbruch, twocolumn-Dokumenten
<code>\clearpage</code>	Seitenumbruch, zwangsweise Floating Umgebungen werden eingefügt
<code>\pagebreak[Priorität04]</code>	möglicher Seitenumbruch Seite vor Umbruch bis zum Rand gestreckt
<code>\nopagebreak[Priorität04]</code>	möglichst keinen Seitenumbruch

Tabelle 8.4: Manueller Seitenumbruch

9 Abstand

In der Präambel kann der, durch die Dokumentenklasse festgelegte, Abstand mit Hilfe des Befehls `\linespread{Abstand}` verändert werden. Zum Ändern des Zeilenabstands sollte allerdings auf das Paket `setspace` gesetzt werden.

Horizontale Abstände kann man mit den Befehlen `hspace{länge}` und `hspace*{länge}` erzeugt werden. Der Befehl `hspace*` setzt den Abstand auch am Beginn/Ende einer Zeile. Mit `\hfill` wird vorgegebener Platz ausgefüllt.

Vertikale Abstände werden genauso wie Horizontale abgehandelt. Hierzu dienen die Befehle `\vspace{länge}`, `\vspace*{länge}` und `\vfill`.

10 Ausrichtung

L^AT_EX setzt den Text automatisch in Blocksatz. Dies führt dazu, dass der rechte und der linke Rand bündig sind. Die Ausrichtung kann allerdings mit entsprechenden Befehlen manipuliert werden.

10.1 Zentrieren

Zum Zentrieren von Text stehen der Befehl `\centering` (für *floating*-Umgebungen geeignet) oder die Umgebung `center` (fügt zusätzlichen vertikalen Abstand ein) zur Verfügung. Der T_EX-Befehl `\centerline` sollte in L^AT_EX vermieden werden, da er z.B. inkompatibel mit dem *color*-Paket ist und zu unerwarteten Effekten führen kann.

10.2 Flatterrand

Linksbündiger Rand

Einen rechten Flatterrand kann mit Hilfe der `flushleft`-Umgebung erstellen. Soll nur eine einzelne Zeile linksbündig gesetzt werden, kann der Befehl `\raggedright` genutzt werden.

Rechtsbündiger Rand

Einen linken Flatterrand kann mit Hilfe der `flushright`-Umgebung erstellen. Soll nur eine einzelne Zeile rechtsbündig gesetzt werden, kann der Befehl `\raggedleft` genutzt werden.

11 Zitate

In L^AT_EX gibt es mehrere Umgebungen mit der Zitate dargestellt werden können. Die Umgebungen erzeugen einen neuen Absatz, bei welchen der Text beidseitig eingerückt wird.

```
1
2 %Absatz
3 \begin{quote}
4     Zitat
5 \end{quote}
6
7 %Absatz mit Erstzeileneinzug
8 \begin{quotation}
9     Zitat
10 \end{quotation}
```

Listing 11.1: Quote (Befehle)

Eine Besonderheit stellt die **verse**-Umgebung dar. Der Text wird von ihr in der Form eines Gedicht- oder Liedtextes dargestellt.

Seven deadly sins
Seven ways to win
Seven holy paths to hell
And your trip begins

Seven downward slopes
Seven bloodied hopes
Seven are your burning fires
Seven your desires. . .
(Iron Maiden: Moonchild)

Abbildung 11.1: Verse (Beispiel)

12 Verweise und Bemerkungen

12.1 Fußnoten

L^AT_EX stellt einen Befehl bereit, der automatisch Fußnoten erzeugt, nummeriert und den Text der Fußnote am Ende der Seite darstellt. Optional kann ein anderes Symbol zur Nummerierung vergeben werden. Mit `\footnotemark` kann eine Fußnote ohne Text gesetzt werden (z.B. um auf eine bestehende Footmark per Label zu verweisen). Per `\footnotetext` kann eine Fußnote gesetzt werden, welche nicht im Text referenziert wird.

```
1 \footnote[nummer]{text}
2
3 \footnotemark[]
4
5 \footnotetext[]{}

```

Listing 12.1: Fußnoten (Befehle)

12.2 Randbemerkung

L^AT_EX unterstützt das Setzen von Marginalien (Randbemerkungen). Hierbei wird der angegebene Text auf Höhe des Befehls an den Rand gesetzt.

```
1 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
2 diam nonumy eirmod\marginpar{Anmerkung} tempor invidunt ut
3 labore et dolore magna aliquyam erat, sed diam voluptua.

```

Listing 12.2: Randbemerkung (Befehle)

12.3 Marken

In L^AT_EX kann mit Hilfe des Befehls `\label{markenname}` eine Marke gesetzt werden. Zur Benennung sollten nur ASCII Zeichen verwendet werden, um Fehler beim kompilieren zu vermeiden. Markennamen müssen innerhalb eines Dokuments eindeutig sein.

12.4 Querverweise

Sind Marken gesetzt worden können diese aufgerufen werden. Für die Bezugnahme stehen verschieden Befehle zur Verfügung. Querverweise werden in einer externen Datei gespeichert (*.aux*). Mit Hilfe von Paketen kann weitere Funktionalität nachgeladen werden.

<code>\ref{markenname}</code>	Kapitelnummer
<code>\pageref{markenname}</code>	Seitennummer
<code>\href{ziel}{text}</code>	Link ins Internet
<code>\hypertarget{name}{text}</code>	legt Ziel fest
<code>\hyperlink{ziel}{text}</code>	springt Ziel an

Tabelle 12.1: Verweise (Befehle)

13 Listen

In $\text{\LaTeX}$ gibt es mehrere Möglichkeiten unterschiedliche Listen zu realisieren. Hierzu gibt es verschiedene Umgebungen.

13.1 einfache Listen

Mit der Umgebung `itemize` können einfache Listen erzeugt werden. Eine Verschachtelung ist möglich.

```
1 \begin{itemize}
2     \item
3     \item
4 \end{itemize}
```

Listing 13.1: `itemize`-Liste

13.2 Numerierte Listen

Mit der Umgebung `enumerate` können nummerierte Aufzählungen erzeugt werden. Eine Verschachtelung ist möglich.

```
1 \begin{enumerate}
2     \item
3     \item
4 \end{enumerate}
```

Listing 13.2: `enumerate`-Liste

13.3 Beschreibung

Die Umgebung `description` eignet sich für Beschreibungen.

```

1 \begin{description}
2     \item [text:] beschreibung
3     \item [text:] beschreibung
4 \end{description}

```

Listing 13.3: `itemize`-Liste

einfache Liste	nummerierte Liste	Beschreibung
• Punkt1	1. Punkt1	C: Beschreibung
• Punkt2	2. Punkt2	
• Punkt3	3. Punkt3	JAVA: Beschreibung
– Punkt3.1	a) Punkt3.1	
– Punkt3.2	b) Punkt3.2	PHP: Beschreibung
• Punkt4	4. Punkt4	HTML: Beschreibung

Abbildung 13.1: Listen Beispiele

14 Tabellen

14.1 Tabulator - tabbing

Innerhalb der `tabbing`-Umgebung ist es möglich Tabulatoren zu setzen. Tabulatoren werden mit `\=` gesetzt und können mit `\>}` angesprungen werden. Zeilen werden durch einen doppelten Backslash getrennt. Eine mit `\kill` beendete Zeile wird nicht ausgegeben, dennoch können aber in ihr gesetzte Tabulatoren angesprungen werden. Im Gegensatz zur `tabular`-Umgebung erlaubt `tabbing` einen Seitenumbruch.

```
1 \begin{tabbing}
2 xxxxxxxxxxxxxxxxxxx\=xxxxxxxxxxxxxxxx\=\kill
3 item1                \>color1          \>size1\\
4 item2                \>color2          \>size2\\
5 item3                \>color3          \>size3\\
6 item4                \>color4          \>size4\\
7 \end{tabbing}
```

Listing 14.1: tabbing-Beispiel

14.2 Tabellen - tabular

Richtige Tabellen können mit der `tabular`-Umgebung erzeugt werden. Tabellen können Parameter übergeben werden, die das Aussehen beeinflussen. Folgende Parameter können übergeben werden:

<code>l</code>	Spalte mit linksbündigem Text
<code>c</code>	Spalte mit zentriertem Text
<code>r</code>	Spalte mit rechtsbündigem Text
<code>p{breite}</code>	Spalte mit fixer Breite
<code> </code>	senkrechter Strich

Innerhalb der Tabelle erfolgt ein Spaltenwechsel mit `&`. Eine neue Zeile wird mit `\\` oder `\tabularnewline` erreicht. `\hline` zeichnet eine horizontale Linie. Um eine mehrspaltige Überschrift zu setzen dient der Befehl `\multicolumn{AnzahlSpalten}{Ausrichtung}{Text}`. Im Gegensatz zur `tabbing`-Umgebung setzt die `tabular`-Umgebung den Seitenumbruch nicht automatisch.

```

1 \begin{tabular}[h]{|l|c|l|p{8cm}|r|}
2 \hline
3 \multicolumn{4}{|c|}{Beispieltabelle}\\
4 \hline
5 Nr.      & Bezeichnung      & Prio      & Datum\\
6 1        & Art1              & 10        & 2010\\
7 2        & Art2              & 8         & 2012\\
8 3        & Art3              & 3         & 2016\\
9 \hline
10 \end{tabular}

```

Listing 14.2: tabular-Beispiel

Beispieltabelle			
Nr.	Bezeichnung	Prio	Datum
1	Art1	10	2010
2	Art2	8	2012
3	Art3	3	2016

Tabelle 14.1: tabular-Beispiel

Anmerkung

Falls innerhalb einer Tabellenumgebung eine Liste verwendet werden soll, muss diese in einer Zelle einer Spalte mit fester Breite sein oder innerhalb einer `minipage`. Ansonsten kommt es zu einem Compilerfehler.

15 Quellcode Darstellung

Selbstverständlich stellt L^AT_EX Methoden zur Verfügung um den eigenen Quellcode uninterpretiert darzustellen. Hierbei erfolgt keinerlei Formatierung und der Text wird in einer TypeWriter-Schrift dargestellt.

```
1
2 %Befehle werden nicht interpretiert
3 \begin{verbatim}
4     Quellcode
5 \end{verbatim}
6
7 %Leerzeichen werden ersetzt
8 \begin{verbatim*}
9     Quellcode
10 \end{verbatim*}
11
12 %Inline Darstellung, beliebige Zeichen zur Klammerung
13 \verb=Quellcode=
14
15 %Leerzeichen werden ersetzt
16 \verb**Quellcode+
```

Listing 15.1: Verbatim (Befehle)

16 Boxen

16.1 Konzept

Damit $\text{\LaTeX}$ ein sauberes Schriftbild erzeugen kann bedient es sich dem “Box”-Konzept. Jedes einzelne Zeichen wird in eine sogenannte Box gepackt. Die zu Wörtern zusammengefasste Zeichen werden ebenfalls in eine Box gepackt (jedes Wort eine Box). Die Wörterboxen werden in eine Zeilenbox gepackt, diese in Absatzboxen bis schlussendlich alles in eine Seitenbox gepackt wird. Abstände innerhalb dieser Boxen haben keine feste Größe. Die variablen Abstände werden von $\text{\LaTeX}$ als *rubber* bezeichnet. Ein “Gummi” der gestaucht und gedehnt werden kann.

16.2 Arten

Es gibt verschiedene Arten von Boxen.

LR-Box	Text wird von links nach rechts gesetzt kein Zeilenumbruch (Zeilenbox)
Parbox	Absatzbox mit Zeilenumbruch
Rule-Box	dient zum Linien & Balken zeichnen

16.3 Rahmen

Mit folgenden Befehlen können einzelne Wörter eingerahmt werden. Das Paket `fancybox` wird für die mit `*` gekennzeichneten Rahmen benötigt. Diese Rahmenbefehle erzeugen automatisch eine LR-Box.

<code>\fbox{text}</code>	einfacher Rahmen
<code>\framebox[breite]{text}</code>	Rahmen mit variabler Breite
<code>\shadowbox{text}</code>	Schattenbox*
<code>\doublebox{text}</code>	Doppelrahmen*
<code>\ovalbox{text}</code>	ovaler Rahmen*
<code>\Ovalbox{text}</code>	fetter, ovaler Rahmen*

```

1 \fbox{Iron Maiden}      \framebox[5cm]{Iron Maiden}      \framebox[1cm]{
   Iron Maiden}
2 \shadowbox{Iron Maiden} \doublebox{Iron Maiden} \ovalbox{Iron Maiden}
3 \Ovalbox{Iron Maiden}
4
5 %Beispiele
6 \fbox{Iron Maiden}      \framebox[5cm]{Iron Maiden}      \framebox[1cm]{
   Iron Maiden}
7 \shadowbox{Iron Maiden} \doublebox{Iron Maiden} \ovalbox{Iron Maiden}
8 \Ovalbox{Iron Maiden}

```

Listing 16.1: Rahmenbeispiele

16.4 Parbox

Da die oben aufgeführten LR-Boxen keinen Zeilenumbruch unterstützten, wird zu langer Text einfach über die Zeilenlänge hinaus geführt. Da dieses Verhalten meist nicht gewünscht ist, kann in solchen Fällen auf Par-Boxen zurückgegriffen werden. Par-Boxen beinhalten einen Absatz, indem auch Zeilenumbrüche möglich sind. ParBoxes können mit folgendem Befehl gesetzt werden: `\parbox[position]{breite}{text}`. Mit dem optionalen Parameter *position* kann die Position relativ zur Grundlinie beeinflusst werden (t, b, c ist möglich).

```

1 \parbox{3cm}{Woe to you, Oh Earth and Sea}
2 \parbox[t]{2cm}{Because he knows the time is short}
3 \parbox[c]{2cm}{Let him who hath understanding reckon the number of the
   beast}
4 \parbox[b]{2cm}{Its number is Six hundred and sixty six}

```

Listing 16.2: Parbox Befehle

Folgendes Beispiel veranschaulicht die obigen Befehle. Zur besseren Kenntlichmachung der Par-Boxen wurden diese mit dem `fbox`-Befehl umrahmt. Natürlich sind auch andere Rahmen möglich.

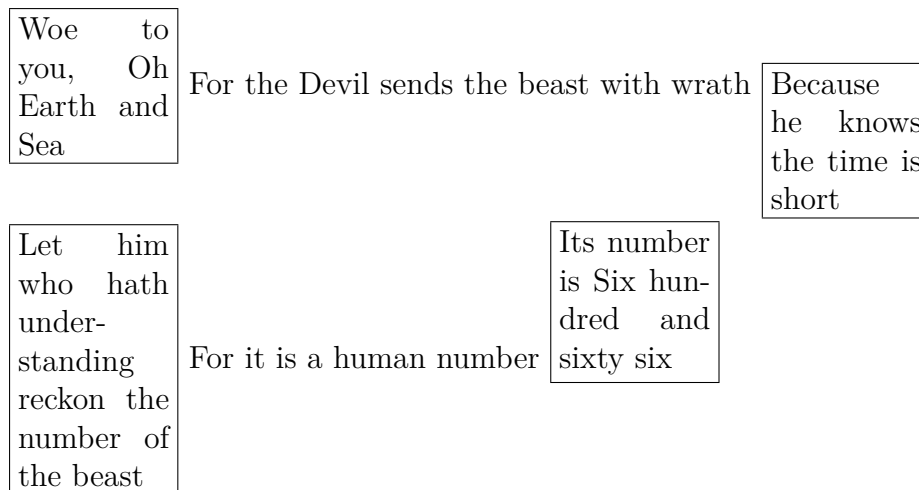


Abbildung 16.1: Parbox Beispiele

16.5 Linien & Balken

Der Befehl `\rule[position]{breite}{höhe}` dient zum Zeichnen von “schwarzen Rechtecken”. Größenangaben als Positionsparameter sind relativ zur Grundlinie zu sehen. Mit einer 0 cm breiten Box lassen sich z.B. Frame-Boxen vergrößern.

16.6 Verschieben von Boxen

Der Befehl `\raisebox{position}` lassen sich Boxen beliebig zur Grundlinie verschieben. Negative Werte setzen die Box unterhalb der Grundlinie.

17 Gleitobjekte

Gleitobjekte verhalten sich gemäß ihren Platzierungsregeln. Vereinfacht kann gesagt werden, dass sie so früh wie möglich platziert werden (aber nicht vor der Definition). Sollte das Seitenlayout eine Platzierung an dieser Stelle nicht zulassen, werden die Objekte verschoben. Dabei wird die Reihenfolge eingehalten. Mit dem Befehl `\clearpage` werden alle Gleitobjekte platziert und danach ein Seitenumbruch erzeugt.

17.1 Abbildungen

Die `figure`-Umgebung ist für Abbildungen gedacht. Bei zweispaltigem Textsatz kann mit dem Befehl `\begin{figure*}` erreicht werden, dass die Abbildung den Platz von zwei Spalten einnimmt. Mit `\caption{Beschreibung}` kann eine Bildunterschrift erstellt werden, welche im Abbildungsverzeichnis berücksichtigt wird. Außerdem ist es möglich ein Label zu setzen, um das Bild im Text zu referenzieren. $\text{\LaTeX}$ kann die Abbildung an verschiedenen Stellen positionieren. Hierzu können mehrere Parameter in eckigen Klammern übergeben werden.

<code>h</code>	here (hier, falls Platz)
<code>t</code>	top (Seitenanfang)
<code>b</code>	bottom (Seitenende)
<code>p</code>	page (eigene Seite)
<code>!</code>	bang (Beschränkungen ignorieren)
<code>H</code>	Genau hier

17.2 Tabellen

Alle genannten Eigenschaften der Abbildungen gelten auch für die Tabellen. Tabellen werden anstatt ins Abbildungsverzeichnis ins Tabellenverzeichnis übernommen und werden mit dem Befehl `\begin{table}` erzeugt.

```
1 \begin{figure}[H]
2     \includegraphics{...}
3     \caption{Beschreibung}
4 \end{figure}
5
6 \begin{table}[bt]
7     \begin{tabular}
8         %Tabelle
9     \end{tabular}
10    \caption{Beschreibung}
11 \end{table}
```

Listing 17.1: Beispiele Abbildung & Tabelle

17.3 minipage

Mit Hilfe einer `minipage` können Inhalte mit fester Breite in ein $\text{\LaTeX}$ Dokument eingebettet werden. Eine `minipage` darf keine Gleitobjekte und Randnotizen enthalten. Die *Äußere Position* legt die Ausrichtung zur Grundlinie fest (c, t, b). Die *Höhe* ignoriert die tatsächliche Höhe und streckt das Element auf die angegebene Größe. Die *Innere Position* richtet den Inhalt innerhalb der `minipage` aus (c, t, b). Die *Breite* muss immer angegeben werden.

```
1 \begin{minipage}[AUSPOSITION][HOHE][INNPOSITION]{BREITE}  
2     Beispieltext  
3 \end{minipage}
```

Listing 17.2: minipage-Umgebung

18 Eigene Befehle

Um den Arbeitsablauf beim Schreiben zu vereinfachen gibt es in $\text{\LaTeX}$ die Möglichkeit eigene Kommandos zu definieren. So können z.B. oft zu tippende lange Befehle abgekürzt oder die Einheitlichkeit der Textelemente gesichert werden. Neben einfachen Befehlen ist es auch möglich Parameter (bis zu 9) zu übergeben.

```
1 %Befehl
2 \newcommand{\NAME}[ANZAHL][OPTIONAL]{DEFINITION}
3
4 %Befehl zum Ueberschreiben vorhandener Befehle
5 \renewcommand{\NAME}[ANZAHL][OPTIONAL]{DEFINITION}
6
7 %Beispielbefehl Definition
8 \newcommand{\format}[2][blue]{\textcolor{#1}{\textbf{#2}}}
9
10 %Beispielbefehl
11 \format{Dies} und \format[greed]{Das}
12
13 %Beispiel
14 Dies und Das
```

Listing 18.1: Befehl mit Parametern

19 Verzeichnisse

$\text{\LaTeX}$ ist in der Lage selbstständig Verzeichnisse zu erstellen. Hierzu dienen Auszeichnungen wie z.B. `\section{Überschrift}` oder auch gesetzte **Captions**. Erstellte Verzeichnisse werden in externe Dateien ausgelagert. Damit die Verzeichnisse korrekt sind sollte der $\text{\LaTeX}$ Kompiler am besten drei mal über das Dokument laufen. Auf das Erstellen eines Inhaltsverzeichnisses wird in Kapitel 6.2 genauer eingegangen.

Befehl	Art des Verzeichnisses	Datei
<code>\tableofcontents</code>	Inhaltsverzeichnis	.toc
<code>\listoftables</code>	Tabellenverzeichnis	.lot
<code>\listoffigures</code>	Abbildungsverzeichnis	.lof

Tabelle 19.1: Verzeichnisse

19.0.1 Anhang

Eine besondere Art des Verzeichnisses ist der Anhang. Dieser wird mit dem `\appendix` erstellt. Mit dem `\section`-Befehl lassen sich automatisch nummerierte Punkte erzeugen (nicht zu verwechseln mit dem Befehl für Überschriften). Die Art der Nummerierung hängt von der verwendeten Dokumentenklasse ab.

20 Steuerzeichen & Sonderzeichen

Um reservierte Steuerzeichen oder Sonderzeichen in $\text{\LaTeX}$ -Dokumenten darstellen zu können gibt es spezielle Befehle. $\text{\LaTeX}$ stellt hält eine sehr große Anzahl an Symbolen bereit. Eine Auflistung von über 14.000 Zeichen findet man in der Comprehensive $\text{\LaTeX}$ Symbol List ¹.

20.1 Griechische Buchstaben

Die griechischen Buchstaben werden normalerweise kursiv dargestellt, da der $\text{\LaTeX}$ -Interpreter davon ausgeht, dass es sich um mathematische Symbole handelt (deswegen müssen die Befehle auch mit $\$$ geklammert werden). Es gibt jedoch mehrere Möglichkeiten dies zu umgehen:

- Das Paket *Uppgreek* ermöglicht das gerade Darstellen der Buchstaben. Die Befehlsnamen erhalten das Präfix *up*.
- Das Paket *Textcomp* ermöglicht das Darstellen der Buchstaben im Fließtext. Die Befehlsnamen erhalten das Präfix *text*. Außerdem können mit dem Paket griechische Großbuchstaben realisiert werden.
- Für Einheiten sollte immer das Paket *Siunitx* verwendet werden.
- UTF-8 Kodierung ist im Fließtext möglich.
- Mit dem Paket *Textgreek* ist ein Wechseln in den Mathemodus nicht nötig.

¹<ftp://ftp.mpi-sb.mpg.de/pub/tex/mirror/ftp.dante.de/pub/tex/info/symbols/comprehensive/symbols-a4.pdf>

Bezeichnung	Befehl	Zeichen
<code>\textAlpha</code>	<code>\textalpha</code>	α
<code>\textBeta</code>	<code>\textbeta</code>	β
<code>\textGamma</code>	<code>\textgamma</code>	γ
<code>\textDelta</code>	<code>\textdelta</code>	δ
<code>\textEpsilon</code>	<code>\textepsilon</code>	ε
<code>\textZeta</code>	<code>\textzeta</code>	ζ
<code>\textEta</code>	<code>\texteta</code>	η
<code>\textTheta</code>	<code>\texttheta</code>	ϑ
<code>\textIota</code>	<code>\textiota</code>	ι
<code>\textKappa</code>	<code>\textkappa</code>	κ

Tabelle 20.1: Auswahl an griech. Klein- & Großbuchstaben

20.2 Interpunktions-, Satzzeichen und Klammern

Befehl	Befehl	Zeichen
Bindestrich	-	-
Halbgeviertstrich	--	—
Geviertstrich	---	---
Auslassungspunkte (horizontal)	\dots	...
Auslassungspunkte (vertikal)	\vdots	⋮
Anführungszeichen dt. links einfach	\glqq \glq	„ ,
Anführungszeichen dt. rechts einfach	\grqq \grq	“ ,
Anführungszeichen fr. links einfach	\flqq \flq	« <
Anführungszeichen fr. rechts einfach	\frqq \frq	» >
Geschweifte Klammern	\{ oder \}	{ oder }

Tabelle 20.2: Auswahl Satzzeichen usw.

20.3 Sonderzeichnen & Symbole

Befehl	Befehl	Zeichen
Backslash	<code>\textbackslash</code>	<code>\</code>
Dollarzeichen	<code>\\$</code>	<code>\$</code>
Copyright	<code>\copyright</code>	<code>©</code>
Trademark	<code>\texttrademark</code>	<code>™</code>
Registriert	<code>\textregistered</code>	<code>®</code>
Paragraph	<code>\S</code>	<code>§</code>
Kaufmanns-Und	<code>\&</code>	<code>&</code>
Raute	<code>\#</code>	<code>#</code>
Unterstrich	<code>_</code>	<code>—</code>
Prozentzeichen	<code>\%</code>	<code>%</code>
Pfeile	<code>\leftarrow</code> <code>\gets</code>	<code>←</code>
	<code>\rightarrow</code> <code>\to</code>	<code>→</code>
	<code>\longleftarrow</code>	<code>←</code>
	<code>\longrightarrow</code>	<code>→</code>
Doppelpfeile	<code>\Leftarrow</code>	<code>⇐</code>
	<code>\Rightarrow</code>	<code>⇒</code>
	<code>\Longleftarrow</code>	<code>⇐</code>
	<code>\Longrightarrow</code>	<code>⇒</code>

Tabelle 20.3: Symbole

Anmerkung: Um die Befehle der Pfeile verwenden zu können müssen diese im Mathe-Modus gesetzt sein (\$-Klammerung) oder das *textcomp* -Paket genutzt werden (text-Präfix).

21 Changelog

V1.0 - 2016.11.23

- Erstveröffentlichung