

# L<sup>A</sup>T<sub>E</sub>X

[www.abbyter.de](http://www.abbyter.de)

Diesen und viele andere Workshops, daneben Tipps, Software und ein Lexikon finden  
Sie auf [www.abbyter.de](http://www.abbyter.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
<b>2</b>	<b>Vorraussetzungen</b>	<b>5</b>
2.1	Woher bekomme ich L <sup>A</sup> T <sub>E</sub> X ?	5
2.2	L <sup>A</sup> T <sub>E</sub> X-Distribution	5
2.3	Editoren für L <sup>A</sup> T <sub>E</sub> X	5
<b>3</b>	<b>L<sup>A</sup>T<sub>E</sub>X-Grundlagen</b>	<b>6</b>
3.1	Aufbau eines L <sup>A</sup> T <sub>E</sub> X-Kommandos	6
3.2	Umgebungen	6
3.3	Vorspann	6
3.4	Zusatzpakete	7
3.5	Titel	8
<b>4</b>	<b>Formatierungen</b>	<b>9</b>
4.1	Strukturierung von Texten	9
4.2	Gestaltung des Schriftbilds	10
<b>5</b>	<b>Symbole</b>	<b>11</b>
5.1	Symbole und die L <sup>A</sup> T <sub>E</sub> X-Syntax	11
<b>6</b>	<b>Bilder</b>	<b>12</b>
6.1	Bilder einbinden	12
6.2	figure-Umgebung	12
6.3	includegraphics	13
6.4	minipage	13
<b>7</b>	<b>fortgeschrittene Funktionen</b>	<b>14</b>
7.1	wissenschaftliche Texte	14
7.2	Inhaltsverzeichnis	14
7.3	Querverweise	15
7.4	Fußnoten	15
7.5	Literaturverzeichnis	15
7.6	Stichwortverzeichnis	16
7.7	umfangreiche Texte zerteilen	16

<b>8 Formelsatz</b>	<b>18</b>
8.1 Formeln einfügen . . . . .	18
8.2 Konstruktion mathematischer Formeln . . . . .	18
8.3 Matrizen . . . . .	19
8.4 Mathematische Sonderzeichen . . . . .	20
8.5 griechische und kalligraphische Buchstaben . . . . .	20
<b>9 Layoutsteuerung</b>	<b>22</b>
9.1 Steuerung des Layouts . . . . .	22
9.2 Trennungen . . . . .	22
9.3 Wortzwischenräume und horizontale Leerräume . . . . .	22
9.4 Zeilenumbruch und vertikale Leerräume . . . . .	23
9.5 Fester Seitenumbruch . . . . .	23
9.6 Eigene Kopfzeile . . . . .	23
9.7 Globale Layouteinstellungen . . . . .	24
<b>10 Abschluss</b>	<b>26</b>

# 1 Einführung

Wer bei LaTeX an etwas erotisches denkt, der ist hier definitiv falsch! So mancher Anfänger wird sogar (vielleicht nicht ganz ohne Grund) behaupten LaTeX wäre alles andere als erotisch und schön. Sondern hässlich und schwer.

Kurzum, bei LaTeX handelt es sich um ein Programm zum Layouten von wissenschaftlichen Texten, aber auch Büchern. LaTeX zeichnet sich dabei besonders durch seinen herausragenden Formelsatz von Formeln aus, das dürfte ein Grund sein, warum er bei der Erstellung von Skripten für die Universität unverzichtbar ist. Daneben bietet LaTeX auch noch eine überragende Satzqualität, diese macht sich in dem automatischen Zeichenausgleich bemerkbar, der z.B. eine e näher an ein V rückt (z.B. bei Vektor). Außerdem ist LaTeX völlig kostenlos. Es wird ständig von einer Vielzahl von Entwicklern weiterentwickelt.

LaTeX ist jedoch alles andere als einfach zu bedienen! Wer sich jedoch in die komplizierte Bedienung einarbeitet, wird mit einem sehr flexiblen und überragend arbeiteten Layoutsystem belohnt, dass fast alle anderen Programme dieser Gattung ohne Mühe um Weiten hinter sich lässt.

LaTeX ist dabei eigentlich nur ein Makropaket für das Satzprogramm TeX. TeX ist jedoch noch komplizierter zu handhaben als LaTeX.

## 2 Voraussetzungen

### 2.1 Woher bekomme ich $\LaTeX$ ?

Da stelle ich die Vor- und Nachteile von  $\LaTeX$  heraus, sage aber noch gar nicht, wie man sich selbst ein Bild machen kann...

Wer Linux hat, der sollte in den Installationsdialog gehen (bei SuSe **YaST/Software installieren**) und dort die  $\LaTeX$ -Pakete auswählen und die Abhängigkeiten checken und schon hat man  $\LaTeX$  auf dem Rechner installiert. Wer Windows hat, der muss sich eine  $\LaTeX$ -Distribution herunterladen. Diese umfasst die notwendigen Dateien, die für die Benutzung von LaTeX notwendig sind. Empfehlenswert ist die kostenlose MiKTeX-Distribution.

### 2.2 $\LaTeX$ -Distribution

Erst einmal: LaTeX ist kein Textverarbeitungsprogramm, sondern ein Satzprogramm! Der Text muss erst mit einem beliebigen Editor geschrieben werden, dabei sind besonders zu formatierende Stellen mit speziellen  $\LaTeX$ -Befehlen zu kennzeichnen, z.B. schreibt man mit folgendem Kommando ein Wort **fett**: `\textbf{fettes Wort}`. Anschließend wird das Ganze von  $\LaTeX$  interpretiert und heraus kommt eine dvi-Datei, die man in eine pdf-Datei umwandeln kann und fertig ist der exzellent gelayoutete Text. Umständlich aber lohnend.

### 2.3 Editoren für $\LaTeX$

Wie ich eben erwähnt habe, ist  $\LaTeX$  nur ein Satzsystem und kein Editor. Man braucht also einen Editor. Man könnte jeden x-beliebigen verwenden, manche sind jedoch für  $\LaTeX$  geeigneter als andere. Es gibt nämlich Editoren, die einem beim Erstellen der LaTeX-Datei unter die Arme greifen. Sei es durch Syntaxhervorhebung oder sogar vorgefertigte LaTeX-Textbausteine, die man per Knopfdruck komfortabel an die richtige Stelle einfügen kann.

Bei Linux muss man mal wieder nur den Installationsdialog bemühen, denn mit LYX oder kile gibt es sehr komfortablen Editoren für LaTeX unter Linux. Suchen Sie einfach im Installationsdialog danach!

Unter Windows muss man auf externe Software zurückgreifen. Der TeXnicCenter harmonisiert gut mit der MiKTeX-Distribution.

## 3 L<sup>A</sup>T<sub>E</sub>X-Grundlagen

### 3.1 Aufbau eines L<sup>A</sup>T<sub>E</sub>X-Kommandos

Wie ich schon erwähnt habe, ist L<sup>A</sup>T<sub>E</sub>X ein Satzsystem. Man gibt also in einen Editor den Text ein, Formatierungen werden dann mit speziellen L<sup>A</sup>T<sub>E</sub>X-Kommandos vorgenommen. Genau um diese speziellen L<sup>A</sup>T<sub>E</sub>X-Kommandos soll es ab nun gehen.

Generell beginnen alle L<sup>A</sup>T<sub>E</sub>X-Kommandos mit `\`, dann folgt der Befehlsname. Manche Befehle haben so genannte Parameter, sie erhalten als Zusatzinformationen, damit sie das tun was erwünscht ist. Parameter stehen in geschweiften Klammern, optionale Parameter werden in eckige Klammern geschrieben. Die Wirksamkeit von Kommandos lässt sich ebenfalls mit geschweiften Klammern regulieren.

### 3.2 Umgebungen

Neben den Befehlen gibt es auch noch so genannte Umgebungen. Diese Umgebungen bestehen aus zwei Teilen, der erste Teil leitet die Umgebung ein und ist folgendermaßen aufgebaut: `\begin{Umgebungsname}`, der zweite Teil beendet die Umgebung und ist folgendermaßen aufgebaut `\end{Umgebungsname}` Kommentare werden mit `%` eingeleitet.

### 3.3 Vorspann

L<sup>A</sup>T<sub>E</sub>X-Texte beginnen mit dem Kommando `\documentclass [optionen] {typ}` L<sup>A</sup>T<sub>E</sub>X kennt vier wichtige Texttypen: `book`, `report`, `article` und `letter`. Die ersten drei Typen sind ähnlich. `article` kennt im Gegensatz zu `book` und `report` keine Kapitel, die kleinste Gliederungseinheit ist dort ein Abschnitt (`section`). In `book` werden alle Seiten automatisch mit Kopfzeilen ausgestattet, in denen neben der Seitennummer auch der Name des aktuellen Kapitels (bei geraden Seiten) und der Name des aktuellen Abschnitts (ungerade Seiten) angegeben wird. Der Texttyp `letter` kann zum Verfassen von Briefen verwendet werden.

Für deutsche Texte sind die Dokumenttypen `scrbook`, `scrreprt`, `scartcl` und `scrlettr2` besser geeignet, da diese besser auf das deutsche DIN-A4-Format angepasst sind und die Überschriften darüber hinaus auch noch in Sans-Serif-Schriften (wie z.B. Arial) ausgeführt werden. Für Sie gelten die gleichen Regeln wie für `book`, `report`, `article` und `letter`. Diese Dokumenttypen sind Bestandteil des KOMA-Paketes; meist ist es installiert, wenn nicht leitet MiKTeX die Nachinstallation des Pakets ein. Übrigens kann man auch weitere Dokumenttypen einbinden; Verlage und Universitäten haben hin und wieder eigene angepasste Vorlagen.

Neben den Texttypen gibt es - wie in dem allgemeinen Kommando oben zu sehen ist - auch noch Optionen, die für das ganze Dokument als Standardwerte gelten:

<b>L<sup>A</sup>T<sub>E</sub>X-Syntax</b>	<b>Wirkung</b>
<code>11pt</code>	Standardschriftgröße 11 (statt 10) Punkt
<code>12pt</code>	Standardschriftgröße 12 (statt 10) Punkt
<code>a4paper</code>	DIN-A4-Format (statt US-Letter)
<code>twoside</code>	Unterscheidung gerade / ungerade Seite (Standard bei book)
<code>twocolumn</code>	zweispaltiger Text

### 3.4 Zusatzpakete

Neben den eben genannten Möglichkeiten das Aussehen des Dokuments zu beeinflussen gibt es auch noch die Möglichkeit über Zusatzpakete L<sup>A</sup>T<sub>E</sub>X-Zusatzfunktionen zu nutzen. Diese Zusatzpakete werden mit dem Kommando `\usepackage [optionen] {name}` geladen.

Im Zusammenhang mit den Zusatzpaketen ist aber noch eines zu beachten: Es ist wichtig, wo die Zusatzpakete eingebunden werden. Alle Zusatzpakete (es können mehrere geladen werden) müssen zwischen dem Beginn des LaTeX-Dokuments `\documentclass {Dokumententyp}` und der `document`-Umgebung die mit `\begin{document}` eingeleitet wird, stehen! Dieser Bereich wird Vorspann oder Präambel genannt. Folgende wichtige Zusatzpakete stehen zur Verfügung:

<code>\usepackage {[...]}</code>	<b>Nutzen</b>
<code>babel</code>	mehrsprachiges Dokument
<code>color</code>	Farben nutzen
<code>fancyhdr</code>	Gestaltung der Kopf- und Fußzeile
<code>german</code>	deutsche Trennungen, Überschriften, usw.
<code>ngerman</code>	wie german, aber neue Trennregeln
<code>UKenglish</code>	britisches Englisch (Trennungen, Überschriften, usw.)
<code>french</code>	französisch (Trennungen, Überschriften, usw.)
<code>graphicx</code>	Einbindung von Grafikdateien
<code>hyperref</code>	PDF- und HTML-Funktionen
<code>makeidx</code>	Stichwortverzeichnis
<code>\usepackage[latin1]{inputenc}</code>	deutscher Zeichensatz

Will man deutsche Texte schreiben, so muss man auch das deutsche Sprachpaket einbinden, sonst sind alle Überschriften und die Trennregeln in der Standardsprache english eingestellt.

## 3.5 Titel

Wollen Sie, dass im Dokument der Titel und der Autor erscheinen, so müssen in der Vorspann zwei Kommandos eingeben, damit L<sup>A</sup>T<sub>E</sub>X auch weiß, was denn als Titel gewählt werden soll. Diese beiden Kommandos sind:

```
\title{Titel des Dokuments}  
\author{Autor des Dokuments}
```

Mit dem Kommando `\maketitle` können Sie innerhalb des Dokuments den Titel, den Autor und das Datum einblenden lassen.



# 4 Formatierungen

## 4.1 Strukturierung von Texten

Der eigentliche Text wird mit dem Kommando `\begin{document}` eingeleitet und mit `\end{document}` abgeschlossen. Innerhalb stehen dann einige Kommandos zur Verfügung, mit denen der Text strukturiert werden kann:

<b>L<sup>A</sup>T<sub>E</sub>X-Syntax</b>	<b>Wirkung</b>
<code>\part{Überschrift}</code>	Teil (nur <code>book</code> und <code>report</code> )
<code>\chapter{Überschrift}</code>	Kapitel (nur <code>book</code> und <code>report</code> )
<code>\section{Überschrift}</code>	Abschnitt
<code>\subsection{Überschrift}</code>	Unterabschnitt
<code>\subsubsection{Überschrift}</code>	Unterunterabschnitt

L<sup>A</sup>T<sub>E</sub>X kümmert sich selbstständig um die Nummerierung der Kapitel und Abschnitte. Gleichzeitig wählt L<sup>A</sup>T<sub>E</sub>X automatisch geeignete Schriftarten und Textabstände für die Überschriften. Bei dem Dokumenttyp `book` werden die Texte des `\chapter`- und `\section`-Kommandos auch beim Erstellen der Kopfzeilen berücksichtigt. Wenn die Kommandos in der Form `\section [kurzfassung] {vollständig}` verwendet werden, wird die Kurzfassung für die Kopfzeile und das Inhaltsverzeichnis, die vollständig Variante dagegen unmittelbar im Text verwendet.

Falls der Text mit einem Appendix ausgestattet werden soll, wird dieser mit `\appendix` eingeleitet. Innerhalb des Anhangs können wieder `\chapter`, `\section`, usw. verwendet werden, wobei die Kapitel nun mit A, B, C nummeriert werden.

Wenn man fast immer die selben Dokumenttypen mit den gleichen Einstellungen verwendet, so lohnt es sich von diesem Rahmen ein Template zu erstellen, das man nur noch individuell anpassen muss. Das könnte z.B. so aussehen:

```
\documentclass [a4paper, 11pt]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\begin{document}
% eigentlicher Text
\end{document}
```

Damit kann man schnell einen Artikel in deutscher Sprache mit neuen Trennregeln auf DIN-A4-Papier mit 11 Punkt Standardgröße schreiben, will man daraus ein Buch machen, so muss man lediglich die erste Zeile anpassen.

## 4.2 Gestaltung des Schriftbilds

Bei den üblichen Textverarbeitungsprogrammen wie Word sind Sie es gewohnt, dass Sie beliebige Schriftarten wählen können. In  $\text{\LaTeX}$  ist das anders, da gibt es normalerweise nur drei Schriftarten, die Standardschrift, die Schrift Typewriter für Programm listings und die Schrift Sans Serif für Hervorhebungen und Überschriften. Man kann aber mit Zusatzpaketen wichtige Schriften nachrüsten. Zur Auswahl stehen:

<code>\usepackage {[...]}</code>	Schrift	wirkt auf
<code>mathptmx</code>	Times New Roman	Normaltext & Formeln
<code>courier</code>	Courier	Typewriter-Schrift
<code>\usepackage[scaled]{helvet}</code>	Helvetica	Sans-Serif-Schrift

Sonstige Formatierungen sind wie gewohnt möglich, folgende Kommandos formatieren den Text wie gewünscht:

$\text{\LaTeX}$ -Syntax	Stil
<code>\textrm{text}</code>	Standard (Roman)
<code>\textsf{text}</code>	Sans Serif
<code>\texttt{text}</code>	Typewriter
<code>\textbf{text}</code>	<b>fett</b>
<code>\textmd{text}</code>	normal
<code>\textit{text}</code>	<i>kursiv</i>
<code>\textsl{text}</code>	<i>geneigt</i>
<code>\textsc{text}</code>	KAPITÄLCHEN
<code>\textup{text}</code>	normal
<code>\emph{text}</code>	<i>kursiv</i>

`\emph` wechselt das Schriftattribut zwischen kursiv und normal. Innerhalb einer kursiven Schrift liefert `\emph` daher eine normale Schrift.

Auch bei den Schriftgrößen gibt es einen Unterschied zwischen  $\text{\LaTeX}$  und dem Rest der Textverarbeitungswelt. Während man bei Word beliebig mit der Schriftgröße herum spielen kann, kann man bei LaTeX nur eine Standardschriftgröße (zwischen 10 und 12 Punkt) einstellen, von der ausgehend von LaTeX eine Schriftgröße berechnet wird. Das

startet mit `\tiny` bei winzig klein, und endet bei dem **sehr großen** `\Huge`, zur Auswahl stehen: `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`.

Um Text <sup>hoch</sup> zu stellen, verwenden Sie `\textsuperscript{text}`. In mathematischen Formeln gilt dagegen die Schreibweise  $\$a \wedge \{b\}$$  für  $a^b$  bzw.  $\$a_{\{b\}}$$  für  $a_b$ .

# 5 Symbole

## 5.1 Symbole und die $\LaTeX$ -Syntax

Wie Sie schon bemerkt haben, werden eine Vielzahl an Zeichen von  $\LaTeX$  für Kommandos verwendet, diese Zeichen können Sie nicht ohne besondere Vorkehrungen im Fließtext nutzen. Solche Zeichen sind: % für Kommentare, ~ für feste Leerzeichen, { und } zum Klammern von Textbereichen, \$ zum Einleiten einer mathematischen Formel, \_ tiefstellen (im mathematischen Modus), ^ hochstellen (im mathematischen Modus). So werden Sonderzeichen in  $\LaTeX$  erzeugt:

$\LaTeX$ -Syntax	Zeichen	$\LaTeX$ -Syntax	Zeichen
--	-	<code>\textbar</code>	
---	—	<code>\char34</code>	”
<code>\_</code>	-	<code>\backslash</code>	\
<code>\#</code>	#	<code>{\tt&lt;}</code>	<
<code>\\$</code>	\$	<code>{\tt&gt;}</code>	>
<code>\&amp;</code>	&	<code>\textasciitilde</code>	~
<code>\%</code>	%	<code>\pounds</code>	£
<code>\{</code>	{	<code>\copyright</code>	©
<code>\}</code>	}	<code>\wedge</code>	^
<code>\dots{}</code>	...	<code>\S{}</code>	§
<code>\</code>	halbes Leerzeichen, Umbruch verboten	”~	Bindestrich, Zeilen- umbruch verboten

# 6 Bilder

## 6.1 Bilder einbinden

Ein Bild sagt mehr als tausend Worte. Also ein guter Grund in  $\LaTeX$  Bilder zu integrieren. Das Einbinden von Bildern in  $\LaTeX$  erfordert zunächst die Einbindung einer neuen Dokumentklasse. Binden Sie also das Paket `graphicx` am Anfang des Dokuments ein (`\usepackage{graphicx}`).

Die Einbindung geht in zwei Schritten von statten. Erst wird eine `figure`-Umgebung erstellt, innerhalb dieser Umgebung wird dann das Bild mit `\includegraphics` in das Dokument eingebunden. Die `figure`-Umgebung ist für die Platzierung des Bildes und die Beschriftung verantwortlich; `\includegraphics` bindet das Bild letztendlich ein. So sieht das Einbinden einer jpg-Grafik an einem Beispiel aus:

```
\begin{figure}[Ausrichtung]
\centering
\includegraphics[Skalierungsoptionen]{bild.jpg}
\caption{Ein schönes Bild}
\label{fig:Bild1}
\end{figure}
```

Die Option *Ausrichtung* wird im folgenden Abschnitt `figure`-Umgebung beschrieben und die Skalierungsoptionen im Abschnitt `includegraphics`.

## 6.2 figure-Umgebung

Nun wollen wir aber noch einmal genauer auf die `figure`-Umgebung eingehen: Die `figure`-Umgebung hat einen optionalen Parameter der bestimmt, wo das Bild im Text platziert wird. Man kann wählen zwischen: `h` (an dieser Stelle), `t` (am Beginn der laufenden Seite), `b` (am Ende der laufenden Seite), `p` (mehrere Abbildungen auf einer eigenen Seite zusammenfassen). Die letzten drei Parametermöglichkeiten können auch kombiniert werden. Wenn  $\LaTeX$  dennoch das Bild an einer anderen Stelle platziert, kann man die Platzierung an einer bestimmten Stelle erzwingen, in dem man als Parameter z.B. `[h!]` eingibt, also den Platzierungswunsch mit dem Ausrufezeichen betont.

In zweispaltigen Texten kann man mit der Verwendung von `figure*` statt `figure` die Verteilung über beide Spalten einstellen. Die Grafik erstreckt sich dann über beide Spalten. Man kann jedoch dann die Parameteroption `h` nicht mehr nutzen.

Die Beschriftung der Grafik wird mit `\caption` vorgenommen.  $\LaTeX$  stellt dann der

Beschreibung **Abbildung  $n$** : voran, wobei  $n$  eine fortlaufende Nummer aller eingefügten Bilder im Kapitel oder Artikel ist. Wenn innerhalb der `\caption`-Anweisung eine `\label`-Anweisung verwendet wird, so kann man mit Querverweisen auf die Grafik hinweisen. Der Beschriftungstext wird automatisch zentriert. Ist der Text umfangreicher und übersteigt die Breite des Bildes, so ist der Einsatz der `\parbox` sinnvoll, die die Textlänge auf die Breite des Bildes beschränkt. Die `\parbox`-Anweisung mit Beschriftung ist folgendermaßen aufgebaut:

```
\parbox{7cm}{\caption{...}}
```

## 6.3 includegraphics

Das Kommando `\includegraphics` bietet einige optionale Optionen an, mit denen sich das Bild skalieren und rotieren lässt. Folgende Optionen stehen zur Auswahl:

<b>L<sup>A</sup>T<sub>E</sub>X</b> -Syntax	<b>Wirkung</b>
<code>width=<math>n</math></code>	bestimmt die Breite der Grafik
<code>height=<math>n</math></code>	bestimmt die Höhe der Grafik
<code>scale=<math>n</math></code>	gibt an, wie stark die Grafik skaliert werden soll
<code>angle=<math>n</math></code>	gibt an, um wie viel Grad die Grafik gedreht werden soll

Zur Größenangabe ist nur die Angabe einer Option nötig, die Grafik wird dann automatisch proportional skaliert. Wollen Sie haben, dass die Grafik eingerahmt, bzw. zentriert wird, müssen Sie dies explizit angeben. Mit `\centerline[...]` wird das Bild zentriert und mit `\fbox[...]` eingerahmt.

## 6.4 minipage

Wenn mehrere Abbildungen nebeneinander platziert werden sollen, so müssen wie im folgenden Beispiel innerhalb der `figure`-Umgebung mehrere `minipages` platziert werden:

```
\begin{figure} [h]
\begin{minipage} [t] {6cm}
\centerline{\includegraphics[width=6cm]{bild1.jpg}}
\caption{\label{bild1} \textsl{Auf diesem Bild ist der unsortierte Binärbaum
zu sehen}}
\end{minipage}
\hfill
\begin{minipage} [t] {6cm}
\setlength{\fboxsep}{0mm} % kein Abstand zur Umrahmung
\centerline{\fbox{\includegraphics[width=6cm]{bild2.jpg}}}
\caption{\label{bild2} \textsl{Auf diesem Bild wurde er sortiert.}}
\end{minipage}
\end{figure}}
```

# 7 fortgeschrittene Funktionen

## 7.1 wissenschaftliche Texte

Seine Stärken kann  $\text{\LaTeX}$  erst richtig bei wissenschaftlichen Texten ausspielen, wo man Inhalts-, Abbildungs-, Literatur-, und Stichwortverzeichnisse benötigt, Querverweise machen und Fußnoten einfügen können muss. Dies alles soll natürlich am besten ohne viel Arbeit geschehen, am besten automatisch.

Na ja, ganz ohne Zutun des Benutzers kann es  $\text{\LaTeX}$  zwar noch nicht, aber die Lösung ist meist nur ein Kommando entfernt. Wie Sie im folgenden sehen werden, ist das Anlegen und Erstellen von Verzeichnissen, Querverweisen und Fußnoten erstaunlich schnell und einfach zu bewerkstelligen.

## 7.2 Inhaltsverzeichnis

Das Inhaltsverzeichnis kann mit dem Kommando `\tableofcontents` an jeder beliebigen Stelle in den Text eingefügt werden, normalerweise wird das Inhaltsverzeichnis jedoch am Anfang des Textes oder nach dem Vorwort platziert. Zur Erstellung des Inhaltsverzeichnisses verarbeitet  $\text{\LaTeX}$  die Datei `name.toc`. In diese Datei trägt  $\text{\LaTeX}$  automatisch bei jedem Durchlauf mit dem Kommando `\tableofcontents` alle Informationen für das Inhaltsverzeichnis ein (Kapitel-, Abschnitts- und Teilabschnittsnamen zusammen mit ihren Seitennummern). Diese Vorgehensweise hat jedoch zur Folge, dass  $\text{\LaTeX}$  im ungünstigsten Fall dreimal ausgeführt werden muss, bis das Inhaltsverzeichnis korrekt ist!

Beim ersten Mal existiert `name.toc` noch nicht, es kann also noch kein Inhaltsverzeichnis angelegt werden,  $\text{\LaTeX}$  muss erst alle Informationen für das Inhaltsverzeichnis sammeln. Bei dem zweiten Durchlauf existiert zwar die Datei `name.toc`, jedoch stimmen nun die Seitennummern des Inhaltsverzeichnisses alle nicht mehr, das Inhaltsverzeichnis beansprucht schließlich auch einige Seiten und diese Seiten wurden beim ersten Durchlauf nicht berücksichtigt. Die Seitenzahlen werden also jetzt aktualisiert. Erst beim dritten Durchlauf stimmen die Seitennummern im Inhaltsverzeichnis mit den tatsächlichen Seitennummern überein.

In das Inhaltsverzeichnis werden normalerweise die Texte aller `\part-`, `\chapter-`, `\section-`, `\subsection-` und `\subsubsection-`Kommandos aufgenommen. Die Anzahl der Gliederungsebenen kann aber mit `\setcounter{topdepth}{n}` vermindert werden.

Die Formatierung des Inhaltsverzeichnisses erfolgt automatisch.  $\text{\LaTeX}$  wählt dabei passende Schriftgrößen, rückt untergeordnete Einträge ein und füllt den Zeilenfreiraum zwischen dem Eintrag und der Seitennummer mit Punkten.

## 7.3 Querverweise

Querverweise weisen auf eine andere Stelle in dem Dokument, z.B. auf eine Abbildung oder einen anderen Abschnitt. Wer diese Daten nicht von Hand ändern will (eigentlich geht das bei komplexen Dokumenten auch nicht!), der setzt einen Querverweis auf diese Stelle. In diesem Zusammenhang gibt es drei wichtige  $\LaTeX$ -Befehle: `\label{verweis-name}`, `\ref{verweis-name}` und `\pageref{verweis-name}`. `\label` kennzeichnet die Stelle, auf den der Querverweis verweisen soll - also z.B. die Abbildung - mit einem Namen; auf diesen Namen kann man dann mit `\ref` und `\pageref` zugreifen. `\ref` gibt die Abschnittsnummer (oder Nummer des Bildes oder der Formel) des Verweises mit dem genannten Namen zurück. `\pageref` gibt die Seite auf der der Verweis mit dem angegebenen Namen steht zurück.

Wie schon bei dem Inhaltsverzeichnis läuft auch bei den Querverweisen die Verarbeitung durch  $\LaTeX$  über eine externe Datei; die Datei `name.aux`. Diese muss erst erstellt werden, daher sind anfangs zwei Durchläufe notwendig bis die Verweise stimmen.

## 7.4 Fußnoten

Fußnoten werden mit `\footnote{text}` erstellt.  $\LaTeX$  fügt daraufhin an dieser Stelle eine hochgestellte Nummer als Fußnote ein und platziert den Fußnotentext an das Ende der laufenden Seite.

Fußnoten werden automatisch durchnummeriert, wobei die Nummerierung bei den Texttypen `book` und `report` in jedem Kapitel neu beginnt.

## 7.5 Literaturverzeichnis

Die Verwaltung des Literaturverzeichnisses erfolgt in zwei Schritten: Zuerst muss am Ende des Buchs (eben dort wo das Literaturverzeichnis erscheinen soll) eine Liste mit allen Einträgen erstellt werden. Anschließend kann im gesamten Text auf die Einträge dieses Verzeichnisses verweisen werden.

Der erste Schritt, also die Liste aller verwendeten Literatur, wird so erstellt:

```
\begin{thebibliography}{n}
\bibitem{marke1} Text
\bibitem{marke2} Text
\end{thebibliography}
```

Es wird eine Liste mit Einträgen in eckigen Klammern erzeugt.  $n$  ist dabei das Maß für die Einrückung der Zahlen, bei bis zu neun Einträgen wird hier 9 eingetragen, darüber und bis zu 99 wird 99 eingetragen, usw. Für Text wird der Titel, der Autor usw. eingetragen. Also z.B. *[1] Introduction to Algorithms, Second Edition; Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C.; MIT Press; 2003*

Der Text kann dabei beliebig formatiert werden.

Die Verweise auf das Literaturverzeichnis im laufenden Text wird mit `\cite{marke}`

vorgenommen. An dieser Stelle erscheint später der Eintrag, der zu `markel` gehört, in eckigen Klammern.

## 7.6 Stichwortverzeichnis

Das Anlegen eines Stichwortverzeichnisses ist etwas kompliziert mit  $\text{\LaTeX}$ Es sind mehrere Schritte notwendig, bis das Stichwortverzeichnis fertiggestellt ist. Als Erstes müssen Sie das Zusatzpaket `makeidx` laden und im Vorspann das Kommando `\makeindex` ausführen. Die Indexeinträge müssen im Text mit dem Kommando `\index{[...]}` markiert werden. Die Stichworteinträge lassen sich auch zusammenfassen, so wird der überbegriffliche Indexeintrag mit `\index{Ober-Eintrag}` wie gewohnt markiert, dazugehörige Einträge werden mit `\index{Ober-Eintrag!Unter-Eintrag}` markiert, also der Obereintrag nochmals genannt und mit einem Ausrufezeichen mit dem Untereintrag verbunden. Sollen Formeln nicht extra erscheinen, sondern an der alphabetisch richtigen Stelle, so muss diesen ein `@` vorangestellt werden. Also z.B. `\index{Pi@$\pi$}`, hier wird unter P einsortiert.

Schließlich muss an der Stelle im Text, an der das Stichwortverzeichnis erscheinen soll, das Kommando `\printindex` angegeben werden.

Beim ersten Durchlauf von  $\text{\LaTeX}$  wird die Datei `name.idx` erstellt, in dieser Datei stehen alle Indexeinträge mit dazugehörigen Seiten, auf denen Sie stehen. Diese Datei muss mit dem Programm `makeindex` bearbeitet werden, `makeindex` sortiert die Einträge alphabetisch (bzw. Symbole, Zahlen, und dann Klein- und Großbuchstaben alphabetisch) und formatiert sie. Unter Linux reicht der Aufruf des Kommandos `makeindex name.idx`; der TeXnicCenter startet `makeindex` automatisch. `makeindex` erstellt eine `.ind`-Datei, die sortiert und formatiert ist. Diese Datei wird beim nächsten Durchlauf von  $\text{\LaTeX}$  in ein Stichwortverzeichnis verwandelt wird.

Leider klappt die Unterstützung für deutsche Umlaute nicht. Sollten Sie also deutsche Umlaute in einem Stichwort haben, dann lohnt sich die Bearbeitung der Datei `name.idn`. Diese enthält die mit LaTeX-Syntax formatierten Einträge. Dort sollten Sie die Einträge an die richtige Stelle rücken und die Datei speichern. Dann sollten Sie  $\text{\LaTeX}$  nochmals ausführen, nun sollte das Stichwortverzeichnis korrekt sortiert sein.

Da es sich um  $\text{\LaTeX}$ -Syntax handelt, kann man auch noch Änderungen in der `idn`-Datei vornehmen, z.B. Seitenzahlen fett machen, usw.

## 7.7 umfangreiche Texte zerteilen

Wenn man ein Buch mit  $\text{\LaTeX}$  schreiben möchte, kann es sehr lohnend sein, den Text auf mehrere Dateien zu verteilen, die bei der Verarbeitung durch LaTeX wieder zusammengesetzt werden.

Es bietet sich an, ein Buch in eine Hauptdatei `buch.tex`, einer Datei `header.tex` die die globalen Dokumenteneinstellungen - den Vorspann - enthält, sowie eine Datei pro Kapitel. In der Hauptdatei `buch.tex` werden die einzelnen Dateien dann mit `\input{datei}` eingebunden. Die einzelnen Dateien müssen übrigens keinen Vorspann haben und auch



nicht mit einer `document`-Umgebung eingerahmt sein.  $\LaTeX$  bindet die Dateien da ein, wo sie mit `\input{[...]}` eingefügt wurden.

Die Erstellung von Verweisen, Inhaltsverzeichnissen und Stichwortverzeichnissen ist ohne Probleme in der Hauptdatei möglich und bezieht die Unterdateien automatisch ein. Die Zerlegung beschleunigt auch die Arbeit an dem Buch; man hat immer kleinere Dateien zu laden, außerdem kann man die einzelnen Kapitel schneller testen, wenn man nicht benötigte Kapitel temporär mit `%` auklammert.

Eine Haupt- und eine Header-Datei könnte z.B. so aussehen:

<b>buch.tex</b>	<b>header.tex</b>
<code>\input{header}</code>	<code>\documentclass{scrbook}</code>
<code>\makeindex</code>	<code>\usepackage{ngerman}</code>
<code>\begin{document}</code>	<code>\usepackage[latin1]{inputenc}</code>
<code>\tableofcontents</code>	
<code>\input{Vorwort}</code>	
<code>\input{kapitel1} \input{kapitel2}</code>	
<code>\inputprintindex</code>	
<code>\end{document}</code>	

# 8 Formelsatz

## 8.1 Formeln einfügen

Die wohl beeindruckendste Fähigkeit ist der unschlagbare Formelsatz von  $\text{\LaTeX}$ . Natürlich profitieren davon nicht nur Mathematiker, sondern auch alle anderen Berufsgruppen, die griechische Buchstaben, Brüche oder sonstige Symbole benötigen, die man nur umständlich in Word findet, oder nicht erzeugen kann.

Mathematische Formeln werden im mathematischen Modus erstellt, diesen mathematischen Modus kann man auf drei verschiedene Weisen aktivieren. Entweder man schreibt die Formel zwischen zwei  $\$$ -Zeilen, oder man schreibt die Formel zwischen  $\[$  und  $\]$  oder drittens, man schreibt die Formel in eine `equation`-Umgebung. Jede dieser Möglichkeiten hat ihre Berechtigung.

Mit der ersten Möglichkeit (zwischen  $\$$ -Zeilen) kann man die Formel in den Fließtext einbetten, mit der zweiten Möglichkeit (zwischen  $\[$  und  $\]$ ) kann man sie als eigenständige Formel zentriert, abgesetzt schreiben, bei der dritten Möglichkeit (`equation`-Umgebung) wird die Formel ebenfalls zentriert, abgesetzt geschrieben und zusätzlich noch nummeriert.

Es ist klar, dass die Darstellung im Fließtext anders aussieht, als wenn die Formel eigenständig, abgesetzt dargestellt wird. Wird die Formel mit den  $\$$ -Zeilen eingeleitet und beendet, so versucht  $\text{\LaTeX}$  sie möglichst platzsparend zu setzen. Dadurch gehen die Formeln im Text unter.

Wollen Sie, dass die Formeln linksbündig positioniert werden, so lohnt es sich das Paket `fleqn` einzubinden, soll die Nummerierung ebenfalls linksbündig erfolgen, so lohnt sich die Einbindung des Pakets `leqno`.

Innerhalb von Formeln wird Text kursiv dargestellt, Funktionsnamen werden jedoch in normaler Schrift dargestellt ( $\text{\LaTeX}$  durchsucht für seinen Funktionenfundus von 33 Funktionen), ist die gewünschte Funktion nicht darunter, oder soll ein anderes Zeichen in normaler Schrift dargestellt werden, so muss man den Text in  $\mbox{\{ \dots \}}$  schreiben.  $\text{\LaTeX}$  geht mit den Abständen in den Formeln recht sparsam um, wem das nicht gefällt, der kann den Abstand mit folgenden Kommandos vergrößern:  $\backslash$ , (kleiner Abstand),  $\backslash:$  (mittlerer Abstand) und  $\backslash;$  (großer Abstand)

## 8.2 Konstruktion mathematischer Formeln

Folgende mathematische Konstrukte lassen sich mit  $\text{\LaTeX}$  eindrucksvoll setzen:

<b>L<sup>A</sup>T<sub>E</sub>X-Syntax</b>	<b>Zeichen</b>	<b>L<sup>A</sup>T<sub>E</sub>X-Syntax</b>	<b>Zeichen</b>
<code>a^{\b}</code>	$a^b$	<code>\sum_{a}^{\b}c</code>	$\sum_a^b c$
<code>a_{\b}</code>	$a_b$	<code>\prod_{a}^{\b}c</code>	$\prod_a^b c$
<code>\frac{a}{b}</code>	$\frac{a}{b}$	<code>{a \choose b}</code>	$\binom{a}{b}$
<code>\sqrt{a}</code>	$\sqrt{a}$	<code>\overbrace{abc}^{\d}</code>	$\overbrace{abc}^d$
<code>\sqrt[n]{a}</code>	$\sqrt[n]{a}$	<code>\underbrace{abc}_{\d}</code>	$\underbrace{abc}_d$
<code>\int_{a}^{\b}c</code>	$\int_a^b c$	<code>\overline{abc}</code>	$\overline{abc}$
<code>\oint_{a}^{\b}c</code>	$\oint_a^b c$	<code>\underline{abc}</code>	$\underline{abc}$

All diese Kommandos können beliebig verschachtelt werden. Normalerweise kümmert sich L<sup>A</sup>T<sub>E</sub>X um die Skalierung des Textes, wollen Sie nichts desto trotz Einfluss auf die Größe haben, so können Sie dies mit den folgenden Kommandos tun:

<b>L<sup>A</sup>T<sub>E</sub>X-Syntax</b>	<b>Wirkung</b>
<code>\displaystyle</code>	normal
<code>\textstyle</code>	etwas kleiner
<code>\scriptstyle</code>	noch kleiner
<code>\scriptscriptstyle</code>	winzig

Klammern werden direkt mit ( ... ) oder [ ... ] gebildet, geschweifte Klammern in L<sup>A</sup>T<sub>E</sub>X werden so gebildet `\{ ... \}`. Die Vorgehensweise bei den geschweiften Klammern ist wegen ihrer Bedeutung in der L<sup>A</sup>T<sub>E</sub>X-Syntax notwendig.

Sollen die Klammern genau so hoch sein wie der Term, so sind die Kommandos `\left` und `\right` den eigentliche Klammern voran zu stellen, also so: `\left( [ ... ] \right)`. Diese Vorgehensweise lässt sich auch für Beträge und beliebige andere Konstrukte einsetzen.

## 8.3 Matrizen

Zur Darstellung von Matrizen wird die `array`-Umgebung verwendet. Allgemein ist die Syntax der Matrix folgendermaßen aufgebaut:

```
\begin{array}{ccc}
term1 & & term3\\
term4 & & term6\\
term7 & & term9
\end{array}
```

Die Syntax einer Matrix ist mit der einer Tabelle vergleichbar; auch bei der Matrix wird die Anzahl und die Ausrichtung durch die Angabe des Kürzels für die Ausrichtung definiert, Sprünge zum nächsten Term werden mit `&` vorgenommen und die Zeile mit `\\` beendet.

Die `array`-Umgebung erzeugt jedoch ungeklammerte Matrizen, sind Klammern erwünscht, so ist vor der Umgebung eine mit `\left(` zu setzen und mit `\right)` zu schließen.

Das sieht dann so aus:

$$\left( \begin{array}{ccc} term1 & term2 & term3 \\ term4 & term5 & term6 \\ term7 & term8 & term9 \end{array} \right)$$

## 8.4 Mathematische Sonderzeichen

In der Mathematik wimmelt es nur so von Sonderzeichen, die an allen Ecken und Enden eingesetzt werden.  $\LaTeX$  kann all diese Sonderzeichen in den Text einbinden, man muss nur das Kürzel kennen, dass das schöne Symbol erzeugt! Folgende wichtige Sonderzeichen werden mit folgenden Befehlen erzeugt:

$\LaTeX$ -Syntax	Zeichen	$\LaTeX$ -Syntax	Zeichen
<code>\infty</code>	$\infty$	<code>\rightarrow</code>	$\Rightarrow$
<code>\Re</code>	$\Re$	<code>\leftrightarrow</code>	$\Leftrightarrow$
<code>\Im</code>	$\Im$	<code>\bar{x}</code>	$\bar{x}$
<code>\forall</code>	$\forall$	<code>\tilde{x}</code>	$\tilde{x}$
<code>\exists</code>	$\exists$	<code>\equiv</code>	$\equiv$
<code>\cdot</code>	$\cdot$	<code>\leq</code>	$\leq$
<code>\pm</code>	$\pm$	<code>\geq</code>	$\geq$
<code>\vee</code>	$\vee$	<code>\sim</code>	$\sim$
<code>\wedge</code>	$\wedge$	<code>\approx</code>	$\approx$
<code>\neq</code>	$\neq$	<code>\cap</code>	$\cap$
<code>\subseteq</code>	$\subseteq$	<code>\cup</code>	$\cup$
<code>\subset</code>	$\subset$	<code>\mapsto</code>	$\mapsto$
<code>\notin</code>	$\notin$	<code>\rightarrow</code>	$\rightarrow$
<code>\in</code>	$\in$	<code>\simeq</code>	$\simeq$

## 8.5 griechische und kalligraphische Buchstaben

Griechische Buchstaben brauchen neben Mathematikern auch viele Ingenieure und Naturwissenschaftler, auch diese Buchstaben lassen sich ohne Probleme in  $\LaTeX$  erzeugen. Hier die Befehle, die die griechischen Buchstaben erzeugen (habe Sie mal alle hingeschrieben, den Namen kann man sich oft eh nicht merken):

$\LaTeX$ -Syntax	Zeichen	$\LaTeX$ -Syntax	Zeichen	$\LaTeX$ -Syntax	Zeichen
<code>\alpha</code>	$\alpha$	<code>\mu</code>	$\mu$	<code>\sigma</code>	$\sigma$
<code>\beta</code>	$\beta$	<code>\nu</code>	$\nu$	<code>\Sigma</code>	$\Sigma$
<code>\chi</code>	$\chi$	<code>\phi</code>	$\phi$	<code>\varsigma</code>	$\varsigma$
<code>\delta</code>	$\delta$	<code>\Phi</code>	$\Phi$	<code>\tau</code>	$\tau$
<code>\Delta</code>	$\Delta$	<code>\varphi</code>	$\varphi$	<code>\theta</code>	$\theta$
<code>\epsilon</code>	$\epsilon$	<code>\pi</code>	$\pi$	<code>\Theta</code>	$\Theta$
<code>\eta</code>	$\eta$	<code>\Pi</code>	$\Pi$	<code>\vartheta</code>	$\vartheta$
<code>\gamma</code>	$\gamma$	<code>\psi</code>	$\psi$	<code>\upsilon</code>	$\upsilon$
<code>\Gamma</code>	$\Gamma$	<code>\Psi</code>	$\Psi$	<code>\Upsilon</code>	$\Upsilon$
<code>\iota</code>	$\iota$	<code>\omega</code>	$\omega$	<code>\xi</code>	$\xi$
<code>\kappa</code>	$\kappa$	<code>\Omega</code>	$\Omega$	<code>\Xi</code>	$\Xi$
<code>\lambda</code>	$\lambda$	<code>\rho</code>	$\rho$	<code>\zeta</code>	$\zeta$
<code>\Lambda</code>	$\Lambda$	<code>\varrho</code>	$\varrho$		

Kalligraphische Buchstaben stehen nur als Großbuchstaben zur Verfügung. Zur Darstellung dieser Buchstaben müssen Sie mit `\cal` die Schriftart ändern:

`M(\cal B, f, \cal A)` erzeugt  $M(\mathcal{B}, f, \mathcal{A})$

# 9 Layoutsteuerung

## 9.1 Steuerung des Layouts

Eigentlich führt L<sup>A</sup>T<sub>E</sub>X Zeilen- und Seitenübrüche selbst durch, das Ergebnis ist manchmal jedoch nicht zufriedenstellend, so dass man doch korrigierend eingreifen möchte. Manchmal hat L<sup>A</sup>T<sub>E</sub>X auch Probleme mit dem Layout, es werden Warnungen in der Form *over- / underfull hbox/vbox* ausgegeben, der Grund dieser Warnungen sind in den meisten Fällen Wörter die sehr lange sind und für die LaTeX keine Trennmöglichkeiten sieht, so dass Löcher im Text entstehen.

Man kann aber, wie sollte es anders sein, auch manuell Änderungen am Layout vornehmen. Im Folgenden zeige ich wie man die Trennungen, die Wortzwischenräume und horizontalen Leerräume, Zeilenumbrüche und vertikale Leerräume, Seitenübrüche und Kopfzeilen des Dokuments beeinflussen kann.

## 9.2 Trennungen

Wir beginnen mit dem wichtigsten: den Trennungen; hier hat L<sup>A</sup>T<sub>E</sub>X manchmal Probleme, so dass wir helfend einschreiten wollen.

Damit L<sup>A</sup>T<sub>E</sub>X das Wort richtig trennen kann, können sie so genannte weiche Trennmöglichkeiten einfügen; solche Trennmöglichkeiten werden mit `\-` eingefügt. Bei anschließend sieht das dann so aus:

*an\–schlie\–ßend*

L<sup>A</sup>T<sub>E</sub>X wählt dann, je nach Platz der noch in der Zeile vorhanden ist, die geeignetste Trennmöglichkeit. Manchmal ist auch genau das Gegenteil gewünscht, Sie haben ein Wort, dass auf keinen Fall getrennt werden soll, oder so kurz ist, dass es sich eigentlich nicht lohnt zu trennen. Solche Wörter sollten Sie in `\mbox{Wort}` schreiben. Anschließend trennt L<sup>A</sup>T<sub>E</sub>X das Wort nicht mehr, sondern stellt es im Zweifelsfall in die nächste Zeile - zusammenhängend.

## 9.3 Wortzwischenräume und horizontale Leerräume

Wie schon bei den Trennungen, so ist es auch bei den Wortzwischenräumen so: L<sup>A</sup>T<sub>E</sub>X macht alles automatisch, manchmal muss man jedoch eingreifen, weil es erwünscht ist vom Standard abzuweichen.

Hinter einem Punkt wird von L<sup>A</sup>T<sub>E</sub>X ein etwas größerer Zwischenraum eingefügt, bei Abkürzungen ist dieser vergrößerte Wortzwischenraum jedoch nicht erwünscht! In solchen Fällen muss nach dem Punkt folgendes Kommando eingesetzt werden *backslash\_*

(`_` steht für Leerzeichen)

Zusätzlicher Leerraum zwischen zwei Wörtern kann durch `\quad`, `\qquad` oder `\hspace{abstand}` eingefügt werden. `\quad` fügt ein Leerzeichen der Größe eines Gedankenstrichs ein, `\qquad` fügt einen doppelt so großen Freiraum ein, `\hspace{abstand}` kann durch den eingegebenen `abstand` gesteuert werden (an Maßeinheiten stehen zur Verfügung: mm, cm, pt (Punkt = 0,353 mm), ex (Höhe eines x in der aktuellen Schriftart), em (Breite eines Gedankenstrichs in der aktuellen Schriftart))

Will man noch mehr Leerraum, so bietet sich `\hfill` an; `\hfill` fügt den gesamten zur Verfügung stehenden Freiraum einer Zeile an der aktuellen Position ein, also so viele bis die Zeile voll ist. Wird `\hfill` in einer Zeile mehrfach verwendet, verkleinert sich der eingefügte Raum entsprechend. Wenn `\hfill` am Ende einer Zeile verwendet wird, muss die Zeile mit `\hbox{ }` abgeschlossen werden. Die `\hbox{ }` ist somit die Grenze für das `\hfill`-Kommando. `\hfill` kann man also z.B. folgendermaßen einsetzen, um einen Text zentriert darzustellen:

```
\hfill mittig \hfill\hbox{ }
```

Soll der Freiraum statt mit Leerzeichen mit Punkten gefüllt werden, so ist das Kommando `\dotfill` zu verwenden. `\hrulefill` erzeugt bei gleichen Bedingungen eine durchgezogene Linie.

## 9.4 Zeilenumbruch und vertikale Leerräume

Normalerweise beginnt  $\text{\LaTeX}$  nur dann eine neue Zeile, wenn die aktuelle voll ist, ein vorzeitigen Zeilenumbruch lässt sich jedoch mit `\` erzwingen. Fügt man hinter `\` noch in eckigen Klammern einen Abstand ein (z.B. so `\[0.5cm]`), so wird ein erhöhter Abstand zur nächsten Zeile gelassen. Das Gleiche bewirkt das Kommando `\vspace{abstand}`. Verwendet man statt `\` das Kommando `\linebreak`, so führt  $\text{\LaTeX}$  einen Randausgleich durch, das heißt die Wortzwischenräume der aktuellen Zeile werden so vergrößert, dass die Zeile bis zum Ende gefüllt ist (also Blocksatz herrscht).

## 9.5 Fester Seitenumbruch

Einen festen Seitenumbruch kann man mit den drei Kommandos `\newpage`, `\pagebreak` und `\clearpage` erreicht werden. `\newpage` beginnt eine neue Seite / Spalte, der Rest der Seite bleibt leer; `\pagebreak` beginnt ebenfalls eine neue Seite / Spalte, es wird jedoch vertikaler Randausgleich durchgeführt, so dass größerer Absatzabstand herrscht (also vertikaler Blocksatz); mit `\clearpage` schließlich kann man eine neue Seite erzeugen, auch bei zweispaltigem Text.

## 9.6 Eigene Kopfzeile

Normalerweise erzeugt  $\text{\LaTeX}$  die Kopfzeilen selbst, bei dem Dokumententyp `book` sind diese bei gerade bzw. ungerade Seiten unterschiedlich (auf gerade Seiten kommt die Ka-

pitelüberschrift, auf die ungerade die Überschrift des Abschnitts). Will man da manuell eingreifen, so muss man das über das Kommando `\pagestyle` die Kopfzeile deaktivieren und mit `\markright` oder `\markboth` eine eigene Kopfzeile erstellen.

Mit `\pagestyle{empty}` kann man die Kopfzeile komplett abschalten, es werden also weder Seitenzahlen noch die vordefinierte Kopfzeile angezeigt. Mit `\pagestyle{plain}` wird die Kopfzeile deaktiviert, die Seitenzahl wird jedoch zentriert in der Fußzeile angezeigt. `\pagestyle{myheadings}` erlaubt eine eigene Kopfzeile, die mit `\markright` oder `\markboth` erstellt wird. `\thispagestyle` verändert die Kopfzeile nur für diese eine Seite.

Mit `\markright{Kopfzeile für ungerade Seiten}` oder `\markboth{Kopfzeile für gerade Seiten}` kann man - wie schon erwähnt - die Kopfzeile/n selbst erstellen. Die optische Darstellung der Kopfzeile lässt sich mit diesen Kommandos jedoch nicht beeinflussen. Will man die optische Darstellung beeinflussen, so muss man sich entweder eine eigene Style-Datei schreiben, oder einfacher das Paket `fancyhdr` einbinden.

## 9.7 Globale Layouteinstellungen

Bis jetzt haben wir immer nur das Layout für kleine Textauschnitte verändert, aber man kann mit  $\text{\LaTeX}$  auch das Layout des ganzen Textes verändern (bzw. des ganzen Textes ab dem ersten Aufruf des Kommandos, dass an jeder Stelle des Dokuments stehen kann).

$\text{\LaTeX}$ -Syntax	Wirkung
<code>\oddsidemarginmaß</code>	Abstand des linken Papierrandes für ungerade Seiten
<code>\evensidemarginmaß</code>	Abstand des linken Papierrandes für gerade Seiten
<code>\textwidthmaß</code>	Breite des bedruckbaren Bereichs
<code>\topmarginmaß</code>	Abstand zum oberen Papierrand
<code>\headheightmaß</code>	Höhe der Kopfzeile
<code>\headsepmaß</code>	Abstand von der Kopfzeile zum Text
<code>\textheightmaß</code>	Texthöhe (ohne Kopf- und Fußzeile)
<code>\columnsepmaß</code>	Spaltenabstand
<code>\columnseprulemaß</code>	Stärke der Linie zwischen den Spalten (Standard: 0)
<code>\footskipmaß</code>	Abstand zwischen Text und unterem Ende der Fußzeile
<code>\footheightmaß</code>	Höhe der Fußzeile

Standardmäßig fügt  $\text{\LaTeX}$  zwischen zwei Absätzen keinen Zwischenraum ein, dafür aber ab dem zweiten Absatz eine Einrückung. Wollen Sie diese Einrückung nicht, so können Sie diese mit dem Kommando `\parindentmaß` (in diesem Fall `\parindent0cm`) im Vorspann abschalten. Wollen Sie stattdessen lieber einen spürbaren Absatzabstand haben, so können Sie diesen mit dem Kommando `\parskipmaß` (z.B. `\parskip2ex`) im Vorspann aktivieren.

Wollen Sie den Zeilenabstand verändern, müssen Sie das Kommando `\renewcommand{\baselinestretchfaktor}` einsetzen. Der normale Zeilenabstand ist von der Schriftgröße abhängig, weshalb hier mit einem *faktor* gearbeitet wird.



$\LaTeX$  nummeriert in der Standardeinstellung automatisch alle Überschriften, dies führt mitunter zu skurrilen Abschnittsnummern wie 3.5.4.3.1; das hat nichts mehr mit Übersicht zu tun! Diesem Treiben kann man jedoch selbstverständlich mit Kommandos beileibe rücken:

<b><math>\LaTeX</math>-Syntax</b>	<b>Wirkung</b>
<code>\setcounter{secnumdepth}{n}</code>	$n+1$ Gliederungsebenen nummerieren
<code>\setcounter{tocdepth}{n}</code>	$n+1$ Gliederungsebenen ins Inhaltsverzeichnis übernehmen
<code>\setcounter{page}{n}</code>	verändert die laufende Seitenzahl
<code>\setcounter{chapter}{n}</code>	verändert die laufende Kapitelzahl
<code>\setcounter{section}{n}</code>	verändert die laufende Abschnittszahl

## 10 Abschluss

Die wichtigsten Befehle kennen Sie jetzt. Ein Blick in die Hilfe von  $\text{\LaTeX}$  oder die Hilfe des Editors wird sicherlich noch einige weitere (für Sie wichtige) Befehle zu Tage fördern. Probieren Sie im Zweifelsfall ein bisschen herum.