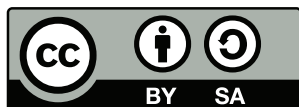


Ein Einstieg in \LaTeX

Prof. Dr. Christian Baun
Frankfurt University of Applied Sciences
(1971-2014: Fachhochschule Frankfurt am Main)
Nibelungenplatz 1
60318 Frankfurt am Main
`christianbaun@fb2.fra-uas.de`

Version 0.7

26. September 2018



Dieses Werk und der zugehörige \LaTeX -Quelltext inklusive der Bilder sind unter der Webseite <https://github.com/christianbaun/einstieginlatex> online verfügbar.

Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz.
<https://creativecommons.org/licenses/by-sa/3.0/de/>

Inhaltsverzeichnis

Vorwort	xiii
1 Einleitung	1
1.1 Aufbau dieses Dokuments	1
1.2 Zusammenhang von T _E X/L ^A T _E X und historischer Hintergrund	2
1.3 Installation von L ^A T _E X und Möglichkeiten damit zu arbeiten	3
1.4 Das allererste Beispiel	4
2 Befehle, Gruppen, Umgebungen und Zeichen	9
2.1 Befehle	9
2.2 Gruppen	9
2.3 Umgebungen	10
2.4 Maßangaben mit absoluten und relativen Maßeinheiten	11
2.4.1 Anwendungsbeispiele	12
2.4.2 Elastische Maße	12
2.5 Sonderzeichen	13
2.5.1 Besondere Zeichen	13
2.5.2 Einige häufig verwendete Zeichen bzw. Symbole	14
2.5.3 Punkte und Striche	16
2.5.4 Anführungszeichen	17
2.5.5 Akzente	17
2.5.6 Ligaturen	17
3 Dokumentklassen und Seitenstile	21
3.1 Dokumentklassen	21
3.1.1 Klassenoptionen	22
3.2 Seitenstile	24
3.3 Definition der Kopfzeile mit <code>markright</code> und <code>markboth</code>	25
3.4 Definition der Kopfzeile mit <code>fancyhdr</code>	27
3.5 Seitennummerierung	30
3.6 Wichtige Abstände	31
3.6.1 Zeilenabstand	31

3.6.2	Absatzabstand	32
3.6.3	Einrückung der ersten Zeile eines Absatzes	32
3.7	Parameter zur Seitendeklaration	33
3.8	Mehrspaltenlayout	35
3.9	Querformat	36
3.10	Befehle und Umgebungen zur Gliederung	37
3.10.1	Titelei	37
3.10.2	Die Zusammenfassung – das Abstract	38
3.10.3	Gliederungsbefehle	39
3.10.4	Tiefe der Nummerierung bei der Gliederung ändern	41
3.10.5	Anfangswert der Nummerierung ändern	41
3.10.6	Zusätzliche Untergliederung bei Büchern	42
3.10.7	Anhang	42
3.11	Inhaltsverzeichnis	43
3.11.1	Manuell Einträge in das Inhaltsverzeichnis einfügen	44
3.12	Abbildungs- und Tabellenverzeichnis	44
3.13	Querverweise	45
3.14	Index	45
4	Formatierungshilfen	47
4.1	Abstände von Wörtern und Zeichen	47
4.1.1	Abstände in Sätzen	47
4.1.2	Horizontale Abstände	48
4.1.3	Vertikale Abstände	49
4.1.4	Leerzeichen in Verbindung mit Satzzeichen – Frenchspacing	50
4.1.5	Leerzeichen darstellen	51
4.1.6	Zeilen mit Zwischenräumen auffüllen	51
4.1.7	Zwischenräume mit Zeichen auffüllen	52
4.1.8	Grenzen des Textfeldes einhalten	53
4.1.9	Trennungsregeln	54
4.2	Absätze zentrieren, nach einer Seite ausrichten oder einrücken	55
4.2.1	Zentrierter Text	55
4.2.2	Linksbündiger Text	56
4.2.3	Rechtsbündiger Text	56
4.2.4	Beidseitig eingerückter Text	56
4.3	Zeilenumbrüche	57
4.4	Seitenumbrüche	58

5	Texthervorhebungen	59
5.1	Texte hervorheben	59
5.2	Schriftgrößen	60
5.3	Schriftfamilien und Schriftschnitte	60
5.4	Aufzählungen	62
5.5	Fußnoten	64
5.6	Marginalien	64
5.7	Unformatierter Text	65
5.8	Internetadressen	66
5.9	Boxen um Text und Bilder zeichnen	66
6	Tabellen	71
6.1	Die Umgebung <code>tabular</code>	71
6.2	Die Umgebung <code>tabularx</code>	75
6.3	Farbige Tabellen	78
6.4	Felder zusammenfassen	81
6.5	Horizontale Linien	84
7	Bilder einfügen	85
7.1	Pfade der Bilddateien definieren	87
8	Mathematischer Formelsatz	89
8.1	Eingebettete Formeln (Textformeln)	89
8.2	Abgesetzte Formeln	90
8.3	Grundlegende mathematische Konstrukte	92
8.3.1	Brüche	92
8.3.2	Indizes und Exponenten	92
8.3.3	Wurzeln	93
8.3.4	Summen und Integrale	94
8.3.5	Informationen über oder unter Zeichen positionieren	95
8.4	Mathematische Zeichen und Symbole	96
8.4.1	Griechische Buchstaben im Mathematikmodus	96
8.4.2	Kalligrafische Buchstaben	98
8.4.3	Operationssymbole	98
8.4.4	Vergleichssymbole	99
8.4.5	Pfeile und Zeiger	100
8.4.6	Sonstige mathematische Zeichen	101
8.4.7	Mathematische Funktionen	103
8.4.8	Akzente in Formeln	104
8.4.9	Über- und Unterstreichungen	106

8.4.10	Pfeile über und unter Formeln	107
8.4.11	Die Größe von Klammern anpassen	107
8.4.12	Matrizen und Felder	108
8.4.13	Mengensymbole	110
8.4.14	Normaler Text in Formeln	110
8.4.15	Mathematische Inhalte einrahmen	111
8.4.16	Ein paar Formeln zum üben...	111
9	Briefe setzen	115
9.1	Briefe mit der Dokumentklasse <code>g-brief2</code>	115
9.2	Briefe mit der Dokumentklasse <code>scrlttr2</code>	119
10	Sonderschriften	123
10.1	Altdeutsche Schriften	123
10.2	Barcodes	123
10.3	QR-Codes	125
10.4	Besondere Zeichen	125
10.5	Kalligrafie	128
11	Literaturverzeichnis	131
11.1	Manueller Satz des Literaturverzeichnisses	131
11.2	Automatisierter Satz des Literaturverzeichnisses	132
11.3	Funktionsweise von Bib _T E _X	135
11.4	Einträgen in Literaturdatenbanken und deren Felder	135
11.5	Das Aussehen des Literaturverzeichnisses anpassen	137
11.6	Die Literaturdatenbank mit der L ^A T _E X-Quelldatei erzeugen	137
12	Präsentationsfolien	139
12.1	Titelfolie	141
12.2	Automatischer Satz eines Inhaltsverzeichnisses	143
12.3	Inhalte auf Präsentationsfolien setzen	143
12.4	Mehrspaltige Präsentationsfolien setzen	145
	Literaturverzeichnis	149
	Verzeichnis der Befehle, Umgebungen, Dokumentklassen und Optionen	151
	Stichwortverzeichnis	157

Abbildungsverzeichnis

1.1	Donald Knuth	3
1.2	Resultat von Listing 1.1	7
2.1	Resultat von Listing 2.1	10
3.1	Aufteilung von <code>fancyheadings</code> bei geraden Seiten	28
3.2	Aufteilung von <code>fancyheadings</code> bei ungeraden Seiten	28
3.3	Voreingestelltes Seitenlayout bei <code>fancyhdr</code>	29
3.4	Layout ungerader Seiten im aktuellen Dokument	34
4.1	Auswirkungen von <code>\nonfrenchspacing</code> und <code>\frenchspacing</code>	51
4.2	Resultat von Listing 4.1	53
5.1	Minipages mit Rahmen realisiert die Umgebung <code>boxedminipage</code>	69
7.1	Abbildungen skalieren die Optionen <code>height</code> oder alternativ <code>width</code>	85
7.2	Abbildungen drehen ermöglicht die Option <code>angle</code>	86
8.1	Der Satz von Brüchen geschieht mit dem Befehl <code>\frac</code>	92
8.2	Der Satz von Wurzeln geschieht mit dem Befehl <code>\sqrt</code>	93
9.1	Resultat von Listing 9.1	118
9.2	Resultat von Listing 9.2	121
11.1	Resultat von Listing 11.1	132
11.2	Resultat von Listing 11.3	138
12.1	Resultat von Listing 12.1	140
12.2	Resultat von Listing 12.2	142
12.3	Resultat von Listing 12.3	145
12.4	Resultat von Listing 12.4	147

Tabellenverzeichnis

2.1	Absolute und relative Maßeinheiten	11
2.2	Besondere Zeichen	14
2.3	Zeichen (Symbole)	14
2.4	Befehle zum komfortablen Satz der T _E X/L ^A T _E X-Schriftzüge	15
2.5	Zeichen aus dem Erweiterungspaket <code>textcomp</code>	15
2.6	Horizontale Striche	16
2.7	Fortsetzungspunkte mit L ^A T _E X	17
2.8	Anführungszeichen	18
2.9	Akzente	18
2.10	Spezialzeichen verschiedener europäischer Sprachen	18
2.11	Ligaturen	19
3.1	Papierformate	24
3.2	Seitenstile	25
3.3	Standardmäßiger Inhalt der Kopfzeile	27
3.4	Seitennummerierungsstile	30
3.5	Längenbefehle zur Beeinflussung des Layouts	35
3.6	Gliederungsebenen und der Zähler <code>tocdepth</code>	39
3.7	Auswirkungen von <code>(n)german.sty</code> auf die Gliederungselemente	40
4.1	Beispiele für Abstände in Sätzen	48
5.1	Befehle zur Definition der Schriftgröße	61
5.2	Befehle zur Definition der Schrift mit Text als Argument	61
5.3	Befehle zur Definition der Schrift und die entsprechenden Umgebungen	62
5.4	Längenbefehle für Boxen	68
6.1	Mögliche Einträge im Parameter <i>Spalten</i>	72
6.2	Eine einfache Tabelle	73
6.3	Vertikale und horizontale Linien trennen die Zellen voneinander ab	73
6.4	Ohne vertikale Trennlinien wird die Tabelle optisch <i>leichter</i>	74
6.5	Identischen Inhalt in jeder Zeile zwischen zwei Spalten einfügen	75

6.6	Jede Spalte erhält den gleichen Anteil an der verfügbaren Tabellenbreite	75
6.7	Spalten mit automatischer Breite bei <code>tabularx</code>	76
6.8	Kombination der Spaltenspezifikationen von <code>tabular</code> und <code>tabularx</code>	77
6.9	Übersicht über die unterstützten Farbmodelle	78
6.10	Einfärben ganzer Zeilen einer Tabelle	79
6.11	Einfärben ganzer Spalten einer Tabelle	80
6.12	Einfärben einzelner Felder einer Tabelle	80
6.13	Spaltenübergreifendes Zusammenfassen von Feldern einer Tabelle	82
6.14	Änderung der Ausrichtung des Inhalts in einem einzelnen Feld	82
6.15	Das zeilenübergreifende Zusammenfassen von Feldern einer Tabelle ermöglicht der Befehl <code>multirow</code>	83
6.16	Horizontale Linien mit fast beliebiger Breite in Tabellen ermöglicht der Befehl <code>cline</code>	84
7.1	Beispiele für relative und absolute Pfade	87
8.1	Größenübersicht verschiedener mathematischer Zeichen	95
8.2	Griechische Großbuchstaben	97
8.3	Griechische Kleinbuchstaben	97
8.4	Variierte griechische Kleinbuchstaben	98
8.5	Kalligrafische Zeichen	98
8.6	Binäre Operationssymbole	99
8.7	Vergleichssymbole	100
8.8	Negierte Vergleichssymbole	101
8.9	Zusätzliche Vergleichssymbole	102
8.10	Pfeile	102
8.11	Pfeile und Zeiger	103
8.12	Negierte Pfeile	103
8.13	Sonstige mathematische Zeichen	104
8.14	Mathematische Funktionen	105
8.15	Akzente in Formeln	105
8.16	Doppelakzente in Formeln	105
8.17	Dreifache bzw. vierfache Punktakzente	106
8.18	Über- und Unterstreichungen in Formeln	106
8.19	Manuelle Größenanpassung der Klammern (Teil 1)	108
8.20	Manuelle Größenanpassung der Klammern (Teil 2)	108
8.21	Befehle um unterschiedliche Klammersymbole zu erzeugen	109
8.22	Befehle um Auslassungspunkte zu setzen	109
8.23	Befehle um Mengensymbole zu setzen	110

10.1 Einzelne Zeichen, als Barcodes gesetzt	124
10.2 Zeichen aus dem Font ZapfDingbats	126
11.1 Auswahl an Stilen für Literaturverzeichnisse	133

Vorwort

Wer sich schon einige Zeit mit dem Textsatz und L^AT_EX beschäftigt, und zum ersten Mal auf dieses Werk stößt, der wird sich vielleicht folgende berechtigte Frage stellen:

„Warum hat sich der Autor die Mühe gemacht dieses Werk zu schreiben?“

In der Tat sind in den letzten Jahrzehnten zahlreiche exzellente Bücher in deutscher [1, 2, 3] und englischer [4, 5] Sprache zu diesem Thema erschienen und wer in den gängigen Suchmaschinen nach einer guten und freien Einführung zum Thema sucht, der wird an vielen Stellen [6, 7, 8, 9, 10, 11] fündig.

Es ist auch nicht das Ziel, das dieses Werk einmal eine möglichst umfangreiche Dokumentation zum Textsatz mit L^AT_EX wird, das andere Werke bzgl. des Umfangs in den Schatten stellt.

Die Motivation hinter der Entstehung dieses Werks ergab sich in erster Linie daraus, das es nur wenige einführende Bücher zu L^AT_EX gibt, deren Quelltext frei ist und deren Lizenz so viele Freiheiten bietet, wie die für dieses Werk verwendete Lizenz.

Den Quellcode des vorliegenden Dokuments finden Sie unter dieser Webadresse:
<https://github.com/christianbaun/einstieginlatex>

Das Werk ist lizenziert unter der Lizenz Creative Commons mit den Einschränkungen „Namensnennung“ und „Weitergabe unter gleichen Bedingungen“ in der Version 3.0 für Deutschland [12].

Wenn Sie Fehler oder Ungenauigkeiten in diesem Dokument finden oder Verbesserungsvorschläge haben, dann zögern Sie bitte nicht mir zu schreiben.

Frankfurt am Main

Prof. Dr. Christian Baun
Oktober 2018

1 Einleitung

L^AT_EX ist ein sehr flexibles und von Rechnerarchitekturen und Betriebssystemen unabhängiges Textsatzsystem. Es ist nicht zu vergleichen mit einem WYSIWYG-Textverarbeitungsprogramm, wie beispielsweise OpenOffice Writer oder MS Word. Die Abkürzung WYSIWYG steht für „What You See Is What You Get“. Das bedeutet, das das gedruckte Dokument der Darstellung auf dem Bildschirm entspricht.

Im Gegensatz zu einem Textverarbeitungsprogramm schreibt ein Autor der L^AT_EX verwendet, entweder mit einem beliebigen Editor oder mit einer speziellen für das Erstellen von L^AT_EX-Dokumenten konzipierten Anwendung den Text zusammen mit Formatanweisungen in eine Textdatei. Das Textsatzsystem L^AT_EX setzt diese Dokumente dann automatisch. Der Autor definiert also den Inhalt (den Text) und die Eckpunkte des Layouts (den Satz). Die Fähigkeiten von L^AT_EX gehen deutlich über die eines gewöhnlichen Textverarbeitungsprogramms hinaus. L^AT_EX realisiert also kein WYSIWYG, sondern folgt dem Prinzip WYSIWYM. Dieses steht für „What You See Is What You Mean“ – der Benutzer umschreibt also das im Ausdruck gewünschte Ergebnis.

Mit L^AT_EX ist es möglich, Dokumente aller Art in Buchdruckqualität zu erstellen. Dabei kann es sich unter anderem um Artikel, (wissenschaftliche) Ausarbeitungen, Briefe, Bücher, Dokumentationen, Präsentationsfolien oder Zeitschriften handeln.

Dieses Dokument wurde mit dem Ziel erstellt, Sie dabei zu Unterstützten, einen möglichst einfachen Einstieg in das Textsatzsystem L^AT_EX zu bekommen.

1.1 Aufbau dieses Dokuments

Dieses Kapitel enthält Informationen zum technischen Hintergrund und zur Arbeitsweise von L^AT_EX. Zudem erfahren Sie hier anhand erster Beispiele, wie eine L^AT_EX-Quelldatei aufgebaut ist und wie Sie von der Quelldatei zum fertigen Dokument kommen.

In Kapitel 2 lernen Sie grundlegendes Wissen über Befehle und Gruppen, sowie Maßangaben und Sonderzeichen unter L^AT_EX kennen.

Die Themen zur Grundlegenden Definition des Layouts von Dokumenten behandelt Kapitel 3. Zu den in diesem Kapitel relevanten Themen gehören Dokumentklassen und Seitenstile, Kopfzeilen und Seitennummern, Mehrspaltenlayout, Titelei und Gliederungsbefehle, Inhaltsverzeichnis und Stichwortverzeichnis sowie Querverweise.

Grundlegende Formatierungshilfen enthält Kapitel 4. Dabei handelt es sich um Befehle und Umgebungen um Abstände einzufügen, um die Inhalte von Absätzen auszurichten, sowie um Zeilen- und Seitenumbrüche anzuweisen.

Kapitel 5 stellt Befehle und Umgebungen vor, die in der Lage sind Texthervorhebungen zu realisieren. Dieses Themengebiet umfasst die Definition der Schriftart und -größe sowie Aufzählungen, Fußnoten, Marginalien und Boxen um Text und Bilder.

Mit dem Satz von Tabellen beschäftigt sich Kapitel 6.

Das Einfügen von Bildern in Dokumenten beschreibt Kapitel 7.

Kapitel 8 thematisiert den mathematischen Formelsatz, der eine der großen Stärken von L^AT_EX ist.

Eine Einführung in das Schreiben von Briefen enthält Kapitel 9.

Verschiedene Möglichkeiten, ein Literaturverzeichnis zu realisieren, stellt Kapitel 11 vor.

Eine komfortable Möglichkeit, um optisch ansprechende Präsentationsfolien zu erstellen, ist die Verwendung der Dokumentklasse `beamer`, die Kapitel 12 vorstellt.

1.2 Zusammenhang von T_EX/L^AT_EX und historischer Hintergrund

Der Urvater des Textsatzsystems T_EX ist der amerikanische Mathematiker und Informatiker Donald E. Knuth [13]. Dieser erlebte den im Laufe der 1970er Jahre, wie immer mehr Druckereien ihre Arbeit vom traditionellen Hand- auf digitalen Computersatz umstellten. Die Ergebnisse des neuen Verfahrens hatten in den ersten Jahren nichts mehr mit der Schönheit und Eleganz des traditionellen Buchdrucks zu tun. Knuth wollte für seine mehrbändige Abhandlung *The Art of Computer Programming* aber einen optisch einwandfreien Drucksatz und begann ein Programm schreiben, das einen menschlichen Setzer, der mit Bleiletern arbeitet, simuliert.

Abbildung 1.1: Donald Knuth¹

Dieses Projekt namens TeX schloss Knuth 1986 ab – also noch lange vor der ersten Version von Linux im Jahre 1992. Das X steht dabei für den im lateinischen Alphabet nicht vorhandenen griechischen Buchstaben Chi.

Da die Auszeichnungssprache von TeX an manchen Stellen nicht sehr intuitiv ist und einen recht eingeschränkten Funktionsumfang bietet, entwickelte der Mathematiker Leslie Lamport [14] ab 1982 L^ATeX. Dieses enthält neben TeX selbst auch unzählige Makropakete, die dazu da sind, den Autoren die Arbeit an Ihren Dokumenten zu vereinfachen und im Gegensatz zu TeX, dessen Entwicklung abgeschlossen ist, wird L^ATeX stetig weiterentwickelt.

1.3 Installation von L^ATeX und Möglichkeiten damit zu arbeiten

L^ATeX besteht aus dem eigentlichen L^ATeX-Compiler und einer Vielzahl an Kommandozeilenwerkzeugen, Erweiterungspaketen, Schriften und Dokumentation. Die manuelle Installation dieser Komponenten wäre ein aufwändiges Unterfangen. Aus diesem Grund existieren sogenannte L^ATeX-Distributionen, die die Installation von L^ATeX zu erleichtern.

Die bekanntesten aktiv gepflegten L^ATeX-Distributionen sind TeX Live [15] MiKTeX [16] und MacTeX [17]. Praktisch alle Linux-Distributionen enthalten Pakete einer L^ATeX-Distributionen von Haus aus. Während TeX Live für alle gängigen UNIX-Betriebssysteme und Windows verfügbar ist, liegt der Fokus von MiKTeX auf Windows und der Fokus von MacTeX auf Mac OS X.

Wenn eine Linux-Distribution L^ATeX bei der Installation des Betriebssystems nicht schon automatisch mitinstalliert hat, ist das über deren Paketverwaltung möglich.

¹Bildquelle: Wikimedia (CC-BY-SA 2.0)

Andere Betriebssystemen wie Windows und Mac OS erfordern einen manuellen Bezug der Installationspakete und deren Installation.

Die Arbeit mit \LaTeX kann auf unterschiedliche Art und Weise erfolgen.

- Eine Möglichkeit ist die Eingabe der Quelldateien mit einem beliebigen Texteditor. Handelt es sich um ein sehr einfaches Programm – im Extremfall könnte es auch Notepad unter MS Windows oder etwas vergleichbares sein – sind noch eine Shell zum Start des \LaTeX -Compilers und ein PDF-Anzeigeprogramm zur Ergebniskontrolle nötig. Wer diese Form der Arbeit mit \LaTeX mit drei Fenstern bevorzugt, der greift idealerweise auf einen Editor mit Komfortmerkmalen wie Syntaxhervorhebung zurück. Einige Texteditor wie zum Beispiel Kate [18] bieten die Möglichkeit, eine Shell in die Oberfläche einzubinden. Das kann die Organisation des Desktop erleichtern.
- Eine weitere Möglichkeit der Arbeit mit \LaTeX ist die Nutzung einer grafischen integrierten Entwicklungsumgebung (IDE = *Integrated Development Environment*) wie zum Beispiel \TeX studio [19], Texmaker [20], \TeX works [21] oder Eclipse [22] zusammen mit der Erweiterung TeXlipse [23]. Diese Werkzeuge bieten eingebettete Anzeigeprogramme und Schaltflächen, um die gängigen Interaktionen mit dem \LaTeX -Compiler und den beteiligten Werkzeugen mit der Maus anzuweisen. Zudem bieten integrierte Entwicklungsumgebung Schaltflächen um \LaTeX -Befehle und Sonderzeichen mit der Maus auszuwählen und in den Quelltext einzufügen.
- Eine weitere Gruppe von Werkzeugen, die besonders hinsichtlich Portabilität und Zusammenarbeit mit anderen Autoren exzellente Möglichkeiten bietet, sind die sogenannten Online- \LaTeX -Editoren, die eigentlich Online-IDEs sind. Beispiele für solche Lösungen sind Authorea [24], Overleaf [25] und ShareLaTeX [26]. Alle diese Dienste bieten deren Nutzung innerhalb gewisser Ressourcenkontingente kostenfrei an. Weitere Speicherressourcen und einzelne Funktionen erfordern in der Regel einen kostenpflichtigen Zugang. Weitere Nachteile solcher Werkzeuge sind die Tatsachen, dass die Software dieser Dienste üblicherweise nicht quelloffen ist und die eignen Dokumente liegen auf fremden Servern, was je nach Inhalt der Dokumente datenschutzrechtlich problematisch sein kann.

1.4 Das allererste Beispiel

Listing 1.1 enthält ein einfaches Beispiel für ein mit \LaTeX gesetztes Dokument mit einigen typischen Befehlen. Wenn Sie das Beispiel nicht aus dem Github-Verzeichnis, das

auch die aktuellste Version dieses Dokuments enthält, kopieren können oder möchten, dann schreiben Sie Quellcode von Listing 1.1 mit einem Texteditor Ihrer Wahl in eine neue Textdatei mit der Dateierdung `.tex`. Der Vorspann, auch Präambel genannt, bestimmt den Aufbau des Dokuments und dort werden Erweiterungspakete importiert. In der Präambel sind neben der Dokumentklasse auch die Standardschriftgröße, das Seitenformat (einseitige oder doppelseitige Ausgabe), die Papiergröße sowie die Textbreite und -höhe definiert.

```
1 \documentclass[a4paper,12pt]{article}
2 \usepackage{ngerman}
3 \usepackage[T1]{fontenc}
4 \usepackage[utf8]{inputenc}
5
6 % Erste Zeile eines Absatzes nicht einrücken
7 \setlength{\parindent}{0pt}
8 % Absatzabstand
9 \setlength{\parskip}{2ex}
10 % Unbedingt die Seitenränder einhalten
11 \sloppy
12
13 \begin{document}
14 \huge
15 Ihr erstes \LaTeX-Dokument.
16 \normalsize
17
18 Hier ist noch \textit{nicht} so viel zu sehen,\,
19 außer \underline{\small wenig} \textbf{Text}.
20 \end{document}
```

Listing 1.1: Das allererste Beispiel

LaTeX-Befehle beginnen stets mit einem Backslash (`\`) und können je nach Befehl mehrere Argumente haben. Bei den meisten Befehlen sind verschiedene Argumente zwingend nötig und weitere optional möglich. Die obligatorischen Argumente stehen in der Regel zwischen geschweiften oder in einigen Fällen zwischen runden Klammern, optionale Argumente sind immer in eckigen Klammern.

Der Befehl `\documentclass` in Zeile 1 definiert die verwendete Dokumentklasse und gehört zwingend zu jedem LaTeX-Dokument. Das Beispiel in Listing 1.1 verwendet die Dokumentklasse `article`, das Papierformat DIN-A4 und eine Schriftgröße von 12 Punkten.

Das Kommando `\usepackage` bindet Erweiterungspakete ein. In Listing 1.1 ist in Zeile 2 das Paket `ngerman` eingebunden, das die Silbentrennung nach den neuen deutschen Rechtschreibregeln festlegt und Kapitel- und Abschnittsüberschriften übersetzt. Sind alternativ die alten Rechtschreibregeln gewünscht, muss alternativ das Erweiterungspaket `german` eingebunden sein.

In Zeile 3 wird die T1-Zeichensatzkodierung aus dem Erweiterungspaket `fontenc` in das Dokument eingebunden. Diese Zeichensatzkodierung trennt Worte mit Umlauten und `ß` korrekt. In Zeile 4 ist festgelegt, dass das Dokument die Zeichensatzkodierung UTF-8 (Unicode) verwendet und nicht mehr das zuvor gebräuchliche ISO 8859-1 (Latin-1) oder ISO 8859-15 (Latin-9).

In den Zeilen 7, 9 und 11 befinden sich Befehle, um einzelne Aspekte des Layouts des fertigen Dokuments zu definieren, wie beispielsweise den Absatzabstand und die Toleranz bei Wortabständen.

Kommentare beginnen mit einem Prozentzeichen (%) und bewirken, dass der LaTeX-Compiler alle Inhalte bis zum Zeilenende ignoriert.

Der Inhalt des Dokuments befindet sich innerhalb der Umgebung `document`. Diese beginnt mit `\begin{document}` in Zeile 13 und endet mit `\end{document}` in der letzten Zeile. Sollten sich nach diesem Befehl noch weitere Zeilen in einer Quelldatei befinden, ignoriert der LaTeX-Compiler diese. Die Schriftart und Schriftgröße des Textes in den Zeilen 14 bis 19 wird mit diversen Formatierungsbefehlen beeinflusst.

Zeilenumbrüche kann ein Autor mit `\\` (siehe Zeile 17) oder alternativ mit dem Befehl `\newline` erzwingen. Mit solchen manuellen Eingriffen in das Layout sollte man nach Möglichkeit sehr sparsam umgehen, da sie dem Prinzip der Trennung von Layout und Inhalt widersprechen.

Um mit Hilfe von LaTeX und der Quelldatei von Listing 1.1 eine PDF-Datei zu erzeugen, rufen Sie auf der Kommandozeile den LaTeX-Compiler mit dem Dateinamen der Quelldatei als Kommandozeilenargument auf:

```
$ pdflatex listing1.tex
```

Die Ausgabe des Kommandos liefert einen detaillierten Bericht über die Dokumenterzeugung. Enthält die Quelldatei Fehler, benennt der LaTeX-Compiler die entsprechenden Stellen mittels Fehlermeldungen und Warnungen. Im Erfolgsfall erzeugt der Aufruf des Compilers drei Dateien im aktuellen Verzeichnis:

- `listing1.aux` – Eine Hilfsdatei, die Querverweise enthält, so dass beispielsweise Fußnoten und Referenzen korrekt nummeriert werden
- `\listing1.log` – Das Protokoll des letzten LaTeX-Aufrufs
- `\listing1.pdf` – Die Ausgabe des LaTeX-Dokuments im Portable Document Format (pdf)

Das erzeugte Dokument (siehe Abbildung 1.2) in der `pdf`-Datei lässt sich mit entsprechenden Betrachtern, wie sie alle aktuellen Betriebssysteme enthalten, anzeigen und ausdrucken.

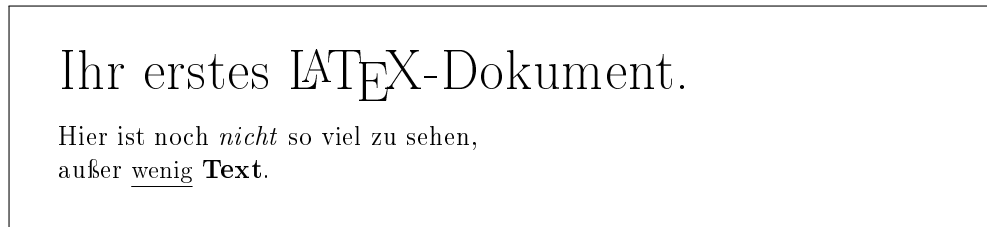


Abbildung 1.2: Resultat von Listing 1.1

2 Befehle, Gruppen, Umgebungen und Zeichen

In diesem Kapitel erfahren Sie, was in der Auszeichnungssprache von \LaTeX Befehle und Umgebungen sind, und wie Sie mit diesen arbeiten können. Zudem enthält dieses Kapitel einige wichtige Sonderzeichen und präsentiert eine Auswahl grundlegender Befehle und Umgebungen anhand von Beispielen.

2.1 Befehle

Wie auf Seite 5 beschrieben, leitet der Backslash (\backslash), der sogenannte *Escape Character* \LaTeX -Befehle ein. Befehle können mehrere Argumente haben. Zwingende Argumente stehen meist in geschweiften Klammern $\{\dots\}$, in manchen Fällen auch in runden (\dots) . Optionale Argumente sind in eckigen Klammern $[\dots]$ angegeben.

$\backslash\text{befehl}[\text{optionale Argumente}]\{\text{zwingende Argumente}\}$

In den allermeisten Fällen Befehle mit Kleinbuchstaben geschrieben. Dabei ist zu beachten, dass \LaTeX *case sensitive* arbeitet. Es unterscheidet also bei den Namen von Befehlen zwischen Groß- und Kleinschreibung. Zudem dürfen die Namen von Befehlen nur Buchstaben enthalten.

Jede Zeile kann mehrere Befehle enthalten und die Reichweite eines Befehls beginnt ab der Stelle im Quelltext, wo er auftritt, bis zum Ende der lokalen Gruppe.

2.2 Gruppen

Eine Gruppe ist ein Textbereich, der von geschweiften Klammern eingeschlossen ist. Der Inhalt einer Gruppe beginnt also nach $\{$ und reicht bis zur nächsten schließenden geschweiften Klammer $\}$.

Gruppen werden häufig verwendet, um einen Textabschnitt in einer anderen Schriftart oder Schriftgröße zu setzen. Dafür wird vor den Text der Gruppe der gewünschte Befehl geschrieben. Dieser Befehl hat dann auch nur innerhalb der geschweiften Klammern Gültigkeit. Einige Beispiele für Gruppen enthält Listing 2.1. Das Ergebnis dieses Beispiels zeigt Abbildung 2.1.

```

1 \documentclass[a4paper,12pt]{article}
2 \usepackage{ngerman}
3 \usepackage[T1]{fontenc}
4 \usepackage[utf8]{inputenc}
5
6 % Erste Zeile eines Absatzes nicht einrücken
7 \setlength{\parindent}{0pt}
8 % Absatzabstand
9 \setlength{\parskip}{2ex}
10
11 \begin{document}
12 Ein {\bfseries Wort} schreiben wir {\Huge groß}.
13
14 Einen anderen {\scshape Textbereich} schreiben
15 wir {\tiny ganz klein}.
16
17 Hin und wieder ist eine Schrift mit {\ttfamily fester
18 Zeichenbreite} sinnvoll, um Dinge {\em hervorzuheben}.
19 \end{document}

```

Listing 2.1: Ein Beispiel zu Gruppen

Ein **Wort** schreiben wir groß.

Einen anderen TEXTBEREICH schreiben wir ganz klein.

Hin und wieder ist eine Schrift mit `fester` Zeichenbreite sinnvoll, um Dinge *hervorzuheben*.

Abbildung 2.1: Resultat von Listing 2.1

2.3 Umgebungen

Umgebungen beginnen mit dem Befehl `\begin{Umgebung}` und enden mit dem Befehl `\end{Umgebung}`.

Zwischen dem Anfang und dem Ende einer Umgebung können Texte und auch Befehle stehen. Allerdings sind nicht alle Befehle innerhalb aller Umgebungen zulässig. Ein

Beispiel dafür ist der Befehl `\verb` zum Satz unformatierter Texte mit fester Zeichenbreite. Dessen Verwendung ist bei zahlreichen Umgebungen nicht zulässig.

Es ist auch möglich, Umgebungen miteinander zu verschalten. Das ist beispielsweise nötig, um Aufzählungen mit mehreren Ebenen (siehe Abschnitt 5.4) zu realisieren.

2.4 Maßangaben mit absoluten und relativen Maßeinheiten

Maßeinheiten und feste sowie elastische Maße ermöglichen es, das Layout von Dokumenten zu definieren.

Maßangaben bestehen aus mindestens einer Dezimalzahl (mit oder ohne Vorzeichen) und aus einer Maßeinheit, die direkt (ohne Leerzeichen) auf die Zahl folgt. Eine Übersicht über die unterstützten Maßeinheiten zeigt Tabelle 2.1. Dezimalzahlen dürfen bei \LaTeX mit einem Komma oder alternativ mit einem Dezimalpunkt geschrieben werden, um die Nachkommastellen kenntlich zu machen. Somit sind beispielsweise `3.5cm` und `3,5cm`, sowie `3cm` erlaubte Angaben.

\LaTeX kennt neun absolute und zwei relative Maßeinheiten. Die relativen Maßeinheiten `em` und `ex` orientieren sich an der aktuell verwendeten Schriftart.

Tabelle 2.1: Absolute und relative Maßeinheiten [27]

Abkürzung	Bedeutung	Umrechnungen	
<code>pt</code>	„Point“ (Punkt)	1 pt = 0,013837 in	1 pt = 0,0351 cm
<code>pc</code>	Pica	1 pc = 12 pt	1 pc = 0,422 cm
<code>in</code>	„Inch“ (Zoll)	1 in = 72,27 pt	1 in = 2.54 cm
<code>mm</code>	Millimeter	1 mm = 2,845 pt	
<code>cm</code>	Zentimeter	1 cm = 28,45 pt	1 cm = 10 mm
<code>bp</code>	Big Point	72 bp = 1 in	1 bp = 0,03528 cm
<code>dd</code>	Didot Punkt	1157 dd = 1238 pt	1 dd = 0,03761 cm
<code>cc</code>	Cicero	1 cc = 12 dd	1 cc = 0,45132 cm
<code>sp</code>	Scaled Point	65536 sp = 1 pt	1 sp < 0,6 * 10 ⁻⁶ cm
<code>em</code>	Geviert	Relatives Maß. Ist ca. die Breite von „M“	
<code>ex</code>		Relatives Maß. Ist ca. die Höhe von „x“	

Das *Geviert* und ist eine typografische Maßeinheit aus der Zeit des Bleisatzes mit beweglichen Lettern. Ein Geviert bezeichnet dabei die Schriftgröße (Kegelhöhe). Bei

einer Schriftgröße von 12 Punkten hat ein Geviert dementsprechend eine Länge von 12 Punkten [28].

Bei der Arbeit mit \LaTeX kommt es häufig vor, dass Absatzeinzüge in Geviert angegeben sind (meist `1em` oder `2em`). Auch die konkreten Längen der unterschiedlichen horizontalen Striche im Text – Geviertstrich, Halbgeviertstrich (Gedankenstrich) und Viertelgeviertstrich (Bindestrich) hängen von der Seitenlänge des Gevierts und damit von der Schriftgröße ab [29].

Während das relative Maß `em` meist für horizontalen Maße verwendet wird, eignet sich `ex` besonders für vertikale Maße.

In der Literatur zu \LaTeX findet sich auch immer wieder die Aussage, dass `em` etwas der Breite des Großbuchstabens „M“ und `ex` in etwa der Höhe des Buchstabens „x“ im aktuellen Font entspricht. [1, 27]

2.4.1 Anwendungsbeispiele

Mit den in diesem Abschnitt beschriebenen Maßeinheiten werden unter anderem Längenbefehlen, die das Dokumentenlayout definieren, Werte zugewiesen

Um beispielsweise die Höhe und Breite des Textes festzulegen, müssen den Längenbefehlen `\textheight` und `\textwidth` mit dem Befehl `\setlength` die gewünschten Werte zugewiesen werden.

```
\setlength{\Längenbefehl}{Maßangabe}
```

Soll beispielsweise ein Dokument eine Texthöhe von 18 cm und eine Textbreite von 13,5 cm haben, muss dieses mit den folgenden beiden Zeilen in der Präambel der `.tex`-Quelldatei definiert werden.

```
\setlength{\textwidth}{13.5cm}
\setlength{\textheight}{18cm}
```

Weitere häufige Anwendungsbeispiele, wo die Maßangaben eine Rolle spielen, sind Befehle um horizontale oder vertikale Abstände zu definieren, wie zum Beispiel `\hspace` und `\vspace`, sowie Abbildungen und Tabellen.

2.4.2 Elastische Maße

Der Unterschied zwischen festen und elastischen Maßen ist, das elastische Maße um festzulegende Werte gestreckt und gestaucht werden können. Die Definition eines elastischen Maßes erfolgt mit drei Werten [1]:

- Sollwert
- Streckwert – der Wert, um den maximal gestreckt werden darf
- Stauchwert – der Wert, um den maximal gestaucht werden darf

Die Syntax für ein elastisches Maß ist:

```
\setlength{\Befehl}{Sollwert plus Streckwert minus Stauchwert}
```

Das folgende Beispiel zeigt ein Anwendungsbeispiel. Dieses definiert den Zeilenabstand zwischen Absätzen auf elastische Art und Weise.

```
\setlength{\parskip}{2ex plus 0.3ex minus 0.5ex}
```

Konkret ist hier definiert, dass der Abstand zwischen zwei Absätzen zweimal so hoch sein soll, wie der Buchstabe „x“ im aktuellen Zeichensatz hoch ist. Der Abstand darf bei Bedarf bis auf den 2,3-fachen Abstand gedehnt oder auf den 1,5-fachen Abstand gestaucht werden.

2.5 Sonderzeichen

Sonderzeichen sind unter L^AT_EX einerseits Zeichen, die eine besondere Funktion haben (beispielsweise & und \$), andererseits sind Sonderzeichen aber auch Zeichen, für die auf einer normalen Tastatur keine Taste existiert (zum Beispiel ©, † und ‡).

Buchstaben mit Akzenten sind im Grunde auch so etwas wie Sonderzeichen und Anführungszeichen aller Art sowie Punkte und Striche der verschiedensten Form kann man auch als Sonderzeichen bezeichnen.

2.5.1 Besondere Zeichen

L^AT_EX kennt eine große Zahl besonderer Zeichen, die auch Sonderzeichen heißen. Dieser Abschnitt präsentiert eine Auswahl an vergleichsweise häufig verwendeten besonderen Zeichen. Diese sind:

\ \$ & % # _ ~ ^ { }

Diese besonderen Zeichen haben bei der Texteingabe eine spezielle Bedeutung (siehe Tabelle 2.2) und die meisten von ihnen können durch das Voranstellen eines Backslashes (\) dargestellt werden.

Tabelle 2.2: Besondere Zeichen

Zeichen	Bedeutung des Zeichens	Befehl
\	Mit einem Backslash beginnt ein Befehl	<code>\textbackslash</code>
\$	Schaltet in den Mathematikmodus um	<code>\\$</code>
&	Markiert die Grenzen der Tabellenspalten	<code>\&</code>
%	Alles bis zum Zeilenende wird ignoriert	<code>\%</code>
#	Makroparameterzeichen	<code>\#</code>
_	Definiert im Mathematikmodus Indizes	<code>\textunderscore</code> oder <code>_</code>
~	Fügt einen untrennbaren Zwischenraum ein	<code>\textasciitilde</code>
^	Definiert im Mathematikmodus Exponenten	<code>\textasciicircum</code>
{	Definiert den Anfang einer Gruppe	<code>\{</code>
}	Definiert das Ende einer Gruppe	<code>\}</code>

2.5.2 Einige häufig verwendete Zeichen bzw. Symbole

\LaTeX bietet eine große Anzahl von Zeichen, die auch Symbole heißen. Tabelle 2.3 präsentiert einige der häufiger verwendeten.

Tabelle 2.3: Zeichen (Symbole)

Symbol	Bezeichnung	Schreibweise
©	Copyright	<code>\copyright</code>
®	Registered	<code>\textregistered</code>
™	Trademark	<code>\texttrademark</code>
§	Juristischer Paragraph	<code>\S</code>
¶	Paragraph	<code>\P</code>
€	Euro	<code>\euro</code>
£	Pfund Sterling	<code>\pounds</code>
¥	Yen	<code>\textyen</code>
†	Dagger	<code>\dag</code>
‡	Doppel-Dagger	<code>\ddag</code>
•	Aufzählungspunkt	<code>\textbullet</code>
@	Klammeraffe	<code>@</code>

Auch für die in diesem Dokument schon mehrfach verwendeten Schriftzüge von \TeX und \LaTeX gibt es entsprechende Befehle. Diese enthält Tabelle 2.4.

Eine sehr große Zahl weiter Zeichen enthält das Erweiterungspaket `textcomp.sty`, das mit dem Befehl `\usepackage{textcomp}` in der Präambel eines \LaTeX -Quelltextes

Tabelle 2.4: Befehle zum komfortablen Satz der T_EX/L^AT_EX-Schriftzüge

Symbol	Bezeichnung	Befehl
T _E X	Schriftzug von T _E X	<code>\TeX</code>
L ^A T _E X	Schriftzug von L ^A T _E X	<code>\LaTeX</code>
L ^A T _E X 2 _ε	Schriftzug der seit 1994 aktuellen Version von L ^A T _E X	<code>\LaTeXe</code>

eingebunden. Eine Auswahl der Befehle im Paket `textcomp` enthält Tabelle 2.5.

Tabelle 2.5: Zeichen aus dem Erweiterungspaket `textcomp`

Befehl	Zeichen	Befehl	Zeichen
<code>\textdollaroldstyle</code>	\$	<code>\textcentoldstyle</code>	¢
<code>\textdollar</code>	\$	<code>\textcent</code>	¢
<code>\textzerooldstyle</code>	0	<code>\textoneoldstyle</code>	1
<code>\texttwooldstyle</code>	2	<code>\textthreeoldstyle</code>	3
<code>\textfouroldstyle</code>	4	<code>\textfiveoldstyle</code>	5
<code>\textsixoldstyle</code>	6	<code>\textsevenoldstyle</code>	7
<code>\texteightoldstyle</code>	8	<code>\textnineoldstyle</code>	9
<code>\texttimes</code>	×	<code>\textonesuperior</code>	¹
<code>\texttwosuperior</code>	²	<code>\textthreesuperior</code>	³
<code>\textonequarter</code>	$\frac{1}{4}$	<code>\textonehalf</code>	$\frac{1}{2}$
<code>\textthreequarters</code>	$\frac{3}{4}$	<code>\textsurd</code>	√
<code>\textperthousand</code>	‰		
<code>\textrightarrow</code>	→	<code>\textleftarrow</code>	←
<code>\textuparrow</code>	↑	<code>\textdownarrow</code>	↓
<code>\textdagger</code>	†	<code>\textdaggerdbl</code>	‡
<code>\textohm</code>	Ω	<code>\textmho</code>	℧
<code>\textlbrackdbl</code>	⌈	<code>\textrbrackdbl</code>	⌋
<code>\textcopyright</code>	©	<code>\textcopyrightleft</code>	©
<code>\textregistered</code>	®	<code>\textcircledP</code>	®
<code>\texttrademark</code>	™	<code>\textservicemark</code>	SM
<code>\textnumero</code>	№	<code>\textestimated</code>	€
<code>\textdegree</code>	°	<code>\textcelsius</code>	°C
<code>\textbullet</code>	•	<code>\textopenbullet</code>	◦

<code>\textasteriskcentered</code>	*	<code>\textmusicalnote</code>	♪
<code>\textordfeminine</code>	ª	<code>\textordmasculine</code>	º
<code>\textlquill</code>	{	<code>\textrquill</code>	}
<code>\textbardbl</code>		<code>\textbrokenbar</code>	
<code>\textthreequartersemdash</code>	—	<code>\texttwelveudash</code>	—
<code>\texttimes</code>	×	<code>\textdiv</code>	÷
<code>\textminus</code>	—	<code>\textdblhyphen</code>	=
<code>\textdiscount</code>	%	<code>\textfractionsolidus</code>	/
<code>\textlangle</code>	<	<code>\textrangle</code>	>
<code>\textparagraph</code>	¶	<code>\textpilcrow</code>	¶
<code>\textreferencemark</code>	※	<code>\textcurrency</code>	¤
<code>\textsection</code>	§	<code>\textbigcircle</code>	○
<code>\textquotestraightdblbase</code>	„	<code>\textquotestraightbase</code>	‚
<code>\textquotesingle</code>	'	<code>\texttildelow</code>	~
<code>\textasciibreve</code>	˘	<code>\textasciicaron</code>	ˇ
<code>\textperiodcentered</code>	·	<code>\textasciidieresis</code>	¨
<code>\textlnot</code>	¬	<code>\textdblhyphenchar</code>	=
<code>\textpm</code>	±	<code>\textasciimacron</code>	—
<code>\textgravedbl</code>	“	<code>\textacutedbl</code>	”
<code>\textasciigrave</code>	`	<code>\textasciacute</code>	´

2.5.3 Punkte und Striche

Tabelle 2.6 zeigt eine Übersicht über die Befehle von L^AT_EX, um unterschiedliche Arten von horizontalen Strichen zu setzen.

Tabelle 2.6: Horizontale Striche

Symbol	Bezeichnung	Befehl
-	Trenn-/Bindestrich	- oder <code>\texthyphenchar</code>
—	Gedankenstrich	-- oder <code>\textendash</code>
—	längerer Gedankenstrich	--- oder <code>\textemdash</code>

Der Trennstich (-), der auch Viertelgeviertstrich heißt, wird als Trennzeichen beim Trennen von Wörtern, oder als *Bindestrich* in zusammengesetzten Begriffen benutzt. Er wird wie ein ganz gewöhnlicher horizontaler Strich (-) geschrieben.

Der Gedankenstrich (—), der auch Halbgeviertstrich heißt, wird oft bei Strecken- oder Zeitangaben benutzt. Er wird mit zwei Strichen (--) oder alternativ mit dem Befehl `\textendash` erzeugt.

Primär im englischen Sprachraum wird in Dokumenten ein längerer Gedankenstrich (—), der sogenannte Geviertstrich, verwendet. Dieser wird mit drei Strichen (---) oder dem Befehl `\textemdash` erzeugt.

Tabelle 2.7: Fortsetzungspunkte mit L^AT_EX

Symbol	Bezeichnung	Befehl
.	gewöhnlicher Punkt	.
...	drei gewöhnliche Punkte	...
...	Fortsetzungspunkte	<code>\dots</code>

Auf den ersten Blick erscheint es kurios, das L^AT_EX mit `\dots` einen Befehl bietet, um drei aufeinander folgende Punkte, so genannte Fortsetzungspunkte, zu setzen. Der Grund, warum dieser Befehl sehr sinnvoll ist, ist der Abstand zwischen den Fortsetzungspunkten. Schreibt ein Autor einfach drei Punkte hintereinander, ist der Abstand zwischen den Punkten recht klein. Der Befehl `\dots` erzeugt ein optisch besseres Ergebnis. Tabelle 2.7 zeigt den Unterschied.

2.5.4 Anführungszeichen

L^AT_EX ermöglicht den Satz verschiedener Anführungszeichen. Eine Übersicht präsentiert Tabelle 2.8.

`glq` steht übrigens für *german-left-quote* (deutsches linkes Anführungszeichen) und `grq` steht für *german-right-quote* (deutsches rechtes Anführungszeichen).

2.5.5 Akzente

L^AT_EX ermöglicht den Satz verschiedener Akzente (siehe Tabelle 2.9).

Zudem ermöglicht L^AT_EX den Satz mehrerer Spezialzeichen, wie sie in einigen europäischen Sprachen gängig sind. Eine Übersicht enthält Tabelle 2.10.

2.5.6 Ligaturen

Eine Ligatur ist eine Verbindung von zwei oder drei Buchstaben zu einem einzigen Zeichen. Anders ausgedrückt, sind Ligaturen zu einem einzelnen Zeichen zusammen-

Tabelle 2.8: Anführungszeichen

Anführungszeichen	Kurzform	Befehl
”Doublequotes”	\dq	\dq oder \doublequotedbl
„deutsche Anführungszeichen“	"‘	\glqq oder \quotedblbase
	"’	\grqq oder \quotedbrbase
‚Halbe deutsche‘		\glq oder \quotedsinglbase
		\grq oder \quotedsingrbase
»französische«	">	\frqq oder \guillemontright
	"<	\flqq oder \guillemonleft
›Halbe französische‹		\frq oder \guilemonright
		\flq oder \guilemontleft
“anglikanische”	“‘	\textquotedblleft
	”’	\textquotedblright
‘halbe anglikanische’	‘	\textquoteleft
	’	\textquoteright

Tabelle 2.9: Akzente

Akzent	Eingabe	Akzent	Eingabe	Akzent	Eingabe
ò	\‘o	ó	\.o	õ	\H{o}
ó	\’o	ö	\u{o}	ø	\c{o}
ô	\^o	õ	\v{o}	ö	\r{o}
õ	\~o	ø	\d{o}	ô	\t{oo}
ō	\=o	⓪	\b{o}	⓪	\textcircled{o}

Tabelle 2.10: Spezialzeichen verschiedener europäischer Sprachen

Ausgabe	Bezeichnung	Eingabe
œ, Œ	Französische Ligatur	\oe, \OE
æ, Æ	Lateinische, skandinavische Ligatur	\ae, \AE
å, Å	Skandinavisches A mit Kreis	\aa, \AA
ø, Ø	Skandinavisches O mit Strich	\o, \O
ł, Ł	Polnisches L mit Strich	\l, \L
ß	Deutsches Es-Zet	\ss

gerückte Buchstaben (siehe Tabelle 2.11). Im Einzelnen handelt es sich dabei um die Buchstabenkombinationen **ff**, **fi**, **fl**, **ffi** und **ffl**. Diese werden von L^AT_EX wie folgt gesetzt: ff, fi, fl, ffi, ffl.

Streng genommen sind auch der Gedankenstrich und der verlängerte Gedankenstrich (siehe Seite 16) Ligaturen, da Sie aus zwei bzw. drei einzelnen Zeichen zu einem zusammengefasst werden.

Tabelle 2.11: Ligaturen

Buchstabenkombination	Ligatur	unterdrückte Ligatur
ff	ff	ff
fi	fi	fi
fl	fl	fl
ffi	ffi	ffi
ffl	ffl	ffl

Wenn \LaTeX eine der vorgestellten Buchstabenkombinationen nicht automatisch als Ligatur setzen soll, muss der Autor dieses im Dokument unterdrücken. Anwendungsbeispiele, wo es sinnvoll sein kann, eine Ligatur zu unterdrücken, ist wenn Wörter falsch getrennt werden.

Eine Ligatur kann unterdrückt werden, indem zwischen jedes Buchstabenpaar, das zur möglichen Ligatur gehören, ein Paar geschweiffter Klammern gesetzt wird. So wird aus `f{ }f{ }l` dann `ffl` und nicht `ffl`. Eine alternative Möglichkeit ist, zwischen jedes Buchstabenpaar, das zur möglichen Ligatur gehört, einen Backslash und einen Slash zu setzen `\./`.

3 Dokumentklassen und Seitenstile

Die in diesem Kapitel beschriebenen Dokumentklassen und Seitenstile definieren die grundlegenden Parameter des Layouts eines Dokuments. Zudem präsentiert dieses Kapitel die benötigten Befehle und Umgebungen, um Dokumente sinnvoll zu gliedern.

3.1 Dokumentklassen

Der erste Befehl, der in der Präambel einer \LaTeX -Quelldatei (`.tex`-Datei) steht, ist der Befehl `\documentclass`. Dieser definiert im Parameter **Dokumentklasse** die Dokumentklasse für das gesamte Dokument.

`\documentclass[Optionen]{Dokumentklasse}`

Mehr als eine Dokumentklasse pro Dokument ist unzulässig. Moderne \LaTeX -Distributionen bringen eine große Anzahl an Dokumentklassen mit. Weil dieses Dokument aus Platzgründen nicht alle beschreiben kann, liegt an dieser Stelle der Fokus auf denjenigen Dokumentklassen, die seit Jahrzehnten sehr populär sind, und die in der Literatur und in der Praxis häufig eingesetzt werden. Zusätzlich berücksichtigt diese Dokumentation einige der neueren KOMA-Klassen [30], die sich speziell den Gepflogenheiten des deutschen Sprachraums widmen.

Im Detail handelt es sich bei den hier vorgestellten traditionellen Dokumentklassen um **article**, **book**, **proc**, **report**, **letter** und **slides**.

Die Dokumentklasse **article** eignet sich gut für kleinere bis mittelgroße Dokumente. Texte, die mit dieser Dokumentklasse realisiert werden, können in Abschnitte (*Sections*), Unterabschnitte (*Subsections*) usw. untergliedert werden. Kapitel (*Chapter*) kennt diese Klasse aber leider nicht.

Für längere Texte, die in Kapitel, Abschnitte, Unterabschnitte usw. unterteilen werden sollen, eignet sich u.a. die Dokumentklasse **report**. Für Bücher oder sonstige umfangreiche Dokumente, die in Kapitel, Abschnitte und Unterabschnitte untergliedert sein sollen, eignet sich die Dokumentklasse **book**. Bei den Dokumentklassen **book** und **report** beginnen Kapitel auch immer mit einer neuen Seite.

Speziell zur Erstellung von Sitzungsprotokollen eignet sich die Dokumentklasse `proc`.

Die Dokumentklasse `letter` ist eine auf US-amerikanische Briefe zugeschnittene Klasse. Für Briefe nach deutschem Standard sind andere Klassen wie `dinbrief`, `g-brief` oder das modernere `g-brief2` besser geeignet (siehe Kapitel 9).

Die Dokumentklasse `slides` eignet sich für Präsentationsfolien. Eine modernere Alternative ist `beamer` (siehe Kapitel 12).

Bei den moderneren KOMA-Dokumentklassen handelt es sich um `scrartcl` (als Ersatz für `article`), `scrreprt` (als Ersatz für `report`), `scrbook` (als Ersatz für `book`) und `scrlettr2` (als Ersatz für `letter`).

3.1.1 Klassenoptionen

Der Befehl `\documentclass` bietet die Möglichkeit, Optionen in eckigen Klammern anzugeben. Mit diesen wird das grundlegende Layout eines Dokuments angepasst.

In den Optionen ist es unter anderem möglich, die Schriftgröße der Basisschrift anzupassen, die Anzahl der Textspalten pro Seite und die Papiergröße sowie die Ausrichtung zu definieren.

Von den in diesem Abschnitt beschriebenen Optionen muss keine zwingend angegeben werden. Sie sind optional. Sind mehrere Optionen angegeben, müssen diese mit Kommas voneinander getrennt sein, wie im folgenden Beispiel zu sehen ist:

```
\documentclass[12pt,twoside,a4paper]{report}
```

10pt, 11pt oder 12pt

Diese drei Dokumentklassenoptionen definieren die Schriftgröße fest. Ist keine der drei Alternativen angegeben, wird \LaTeX standardmäßig die Schriftgröße 10pt wählen.

onecolumn oder twocolumn

Diese beiden Optionen ermöglichen die Definition der Seitenformatierung. Zur Auswahl stehen einspaltiges und zweispaltiges Seitenlayout. Standardmäßig verwendet \LaTeX ein einspaltiges Seitenlayout.

oneside oder twoside

Hiermit kann die Seitenformatierung auf ein- oder doppelseitige Ausgabe festgelegt werden.

Bei doppelseitiger Ausgabe sind die linken Seitenränder von geraden und ungeraden Seiten unterschiedlich groß, so dass beim doppelseitigen Ausdruck die Textfelder übereinstimmen. Bei einseitiger Ausgabe sind die Seitenränder von geraden und ungeraden Seiten gleich groß.

Standardmäßig ist bei den Dokumentklassen **article**, **report** und **letter** die einseitige Ausgabe voreingestellt. Bei der Dokumentklasse **book** hingegen ist doppelseitige Ausgabe voreingestellt.

titlepage oder notitlepage

Ob eine eigene Titelseite generiert wird oder ob der Titel horizontal oben zentriert auf der ersten Dokumentseite gesetzt ist, hängt von der verwendeten Dokumentklasse ab.

Bei den Dokumentklassen **book** und **report** wird eine separate Titelseite erzeugt. Bei der Dokumentklasse **article** ist das nicht der Fall. Mit den Optionen **titlepage** und **notitlepage** können Autoren dieses Verhalten ändern.

draft oder final

Bei der Option **draft** werden Zeilen, bei denen L^AT_EX den Zeilenumbruch nicht korrekt realisieren konnte, das heißt Zeilen, die ein wenig über den Rand des Textfeldes hinausragen, mit einem fetten schwarzen Balken am Rand gekennzeichnet. Dieser Randbalken entfällt bei der Option **final**, die die Standardeinstellung ist.

leqno

Ist diese Dokumentklassenoption angegeben, werden die Nummern von mathematischen Formeln nicht rechtsbündig, wie sonst automatisch üblich, sondern linksbündig gesetzt.

fleqn

Mit dieser Option werden abgesetzte mathematische Formeln nicht zentriert, sondern linksbündig gesetzt.

a4paper, a5paper, b5paper, letterpaper, legalpaper und executivepaper

Diese Dokumentklassenoptionen definieren die Papiergröße. Standardmäßig verwendet \LaTeX die Option **letterpaper** als Standard-Papierformat. Da dieses US-amerikanische Papierformat in Mitteleuropa sehr unüblich ist, sollten Autoren im europäischen Raum mit der Option **a4paper** auf DIN-A4 umzustellen. Eine Übersicht über die Dimensionen der unterschiedlichen Papierformate enthält Tabelle 3.1.

Tabelle 3.1: Papierformate

Dokumentklassenoption	Größe [mm]			Größe [Zoll]		
a4paper	297	x	210	11,7	x	8,3
a5paper	210	x	148	8,3	x	5,8
b5paper	250	x	176	9,8	x	6,9
letterpaper	279,4	x	215,9	11	x	8,5
legalpaper	355,6	x	215,9	14	x	8,5
executivepaper	266,7	x	184,1	10,5	x	7,25

landscape

Mit der Option **landscape** wird das Dokument im Querformat gesetzt.

openright oder openany

Bei der Dokumentklasse **book** beginnt ein neues Kapitel immer auf einer rechten (ungeraden) Seite. Wenn ein vorhergegangenes Kapitel auf einer ungeraden Seite endet, dann wird eine leere, gerade Seite eingefügt. Mit der Option **openany** wird dieses Verhalten von \LaTeX unterdrückt. Dann kann ein neues Kapitel auch auf einer geraden Seite beginnen.

Bei der Dokumentklasse **report** kann ein Kapitel standardmäßig auf jeder Seite beginnen. Mit der Option **openright** können Autoren festlegen, dass Kapitel (genau wie bei der Klasse **book**) nur auf ungeraden, rechten Seiten beginnen dürfen.

3.2 Seitenstile

Der grundsätzliche Aufbau einer Seite heißt Seitenstil. Das Festlegen des Seitenstils geschieht mit dem Befehl `\pagestyle` in der Präambel der `.tex`-Quelldatei. Der gewählte Seitenstil gilt dann für das gesamte Dokument.

`\pagestyle{Seitenstil}`

\LaTeX bietet einige vorgefertigte Seitenstile (siehe Tabelle 3.2). Der Seitenstil `plain` ist die Standardauswahl von \LaTeX , wenn kein anderer Seitenstil mit `\pagestyle` ausgewählt ist.

Tabelle 3.2: Seitenstile

Seitenstil	Aussehen
<code>empty</code>	Die Kopf- und Fußzeilen bleiben leer und es wird keine Seitennummer gesetzt. Nur das Textfeld ist zu sehen.
<code>plain</code>	Die Kopfzeile bleibt leer. In der Fußzeile wird die Seitennummer zentriert ausgegeben.
<code>headings</code>	Die Kopfzeile jeder Seite enthält die aktuelle Seitenzahl und eine Überschrift. Welche Überschrift das genau ist, hängt von der verwendeten Dokumentklasse ab. Für gewöhnlich handelt es sich dabei um die aktuelle Kapitel- oder Abschnittsüberschrift. Die Fußzeile bleibt leer. Die einzige Ausnahme ist die jeweils erste Seite eines neuen Kapitels.
<code>myheadings</code>	Der einzige Unterschied zum Seitenstil <code>headings</code> ist, dass die Kopfzeile nicht automatisch gefüllt wird, sondern dass der Autor den Inhalt der Kopfzeile mit den Befehlen <code>\markright</code> oder <code>\markboth</code> festlegen muss.

Es ist auch möglich, den Seitenstil einer einzelnen Seite festzulegen. Dieses geschieht nicht in der Präambel der `.tex`-Quelldatei, sondern im Text des Dokuments. Der Befehl `\thispagestyle` legt den Seitenstil der aktuellen Seite fest – an der Stelle im Quelltext, an der der Befehl auftritt.

`\thispagestyle{Seitenstil}`

3.3 Definition der Kopfzeile mit `markright` und `markboth`

Ist ein Dokument mit den Seitenstilen `headings` oder `myheadings` gesetzt, können Autoren mit Hilfe der Befehle `\markright` und `markboth` die Inhalte der Kopfzeilen im Dokument definieren und damit die Standardeinstellung (siehe Tabelle 3.3) von \LaTeX zu überschreiben. Die Wirkung der beiden Befehle `\markright` und `\markboth` setzt erst ab der zweiten Seite des Dokuments ein.

Der Befehl `\markright` eignet sich besonders dann, wenn das Dokument einseitig gesetzt wird (Dokumentklassenoption `oneside`). Jede bedruckte Seite gilt dann als rechte Seite.

```
\markright{rechter Kopf}
```

Für doppelseitig gesetzte Dokumente (Dokumentklassenoption `twoside`) eignet sich `\markboth`. Hier können Autoren die Kopfzeilen für linke und rechte Seiten unterschiedlich definieren.

```
\markboth{linker Kopf}{rechter Kopf}
```

Seiten mit *gerader* Seitennummer sind in diesem Fall *linke* Seiten und Seiten mit *ungerader* Seitennummer sind *rechte* Seiten.

Zudem wird beim Befehl `\markboth` die Seitennummer jeder Seite *linksbündig* auf jede Seite mit *gerader* Seitennummer gesetzt und *rechtsbündig* auf jede Seite mit *ungerader* Seitennummer.

Was bei `\markboth` und `\markright` voreingestellt ist, also welche Informationen die Kopfzeilen enthalten, hängt von der verwendeten Dokumentklasse ab (siehe Tabelle 3.3).

Bei doppelseitigem Druckstil und den Bearbeitungsklassen `book` oder `report` wird von `\markboth` automatisch die Überschrift des aktuellen Kapitels (`\chapter`) in die Kopfzeilen der linken Seiten gesetzt, und die Kopfzeilen der rechten Seiten enthalten die Überschriften der Abschnitte (`\section`).

```
\markboth{\chapter}{\section}
```

Bei einseitigem Druckstil und den Bearbeitungsklassen `book` oder `article` wird von `\markright` standardmäßig die Überschrift des aktuellen Kapitels (`\chapter`) in die Kopfzeile der rechten Seiten gesetzt.

```
\markright{\chapter}
```

Bei doppelseitigem Druckstil und der Bearbeitungsklasse `article` wird von `\markboth` die Überschrift des aktuellen Abschnitts (`\section`) in die Kopfzeilen der linken Seiten gesetzt und die Überschrift des aktuellen Unterabschnitts (`\subsection`) wird in die Kopfzeilen der rechten Seiten geschrieben.

```
\markboth{\section}{\subsection}
```


Bei einseitigem Druckstil und `article` wird von `\markright` standardmäßig die Überschrift des aktuellen Abschnitts (`\section`) in die Kopfzeile der rechten Seiten gesetzt.

```
\markright{\section}
```

Tabelle 3.3: Standardmäßiger Inhalt der Kopfzeile [1]

Stil	Befehl	Dokumentklasse	
		book, report	article
doppelseitig	<code>\markboth{linker Kopf}</code>	<code>\chapter</code>	<code>\section</code>
	<code>\markboth{rechter Kopf}</code>	<code>\section</code>	<code>\subsection</code>
einseitig	<code>\markright</code>	<code>\chapter</code>	<code>\section</code>

Alternativ zur automatischen Befüllung der Kopfzeilen mit den Überschriften der Kapitel, Abschnitten und Unterabschnitten, ist es auch möglich, einen beliebigen Text zu definieren, der auf allen Seiten in der Kopfzeile gesetzt wird.

Wenn beispielsweise bei einem zweiseitigen Dokument mit der Dokumentklasse `book` auf jeder linken Seite das Wort *Masterthesis* und auf jeder rechten Seite die aktuelle Kapitelüberschrift stehen soll, kann dieses mit den folgenden Befehlen realisiert werden:

```
\pagestyle{headings}
\markright{Masterthesis}{\chapter}
```

Befinden sich auf einer Seite mehrere `\section`- oder `\subsection`-Befehle, fügt \LaTeX standardmäßig auf *linken* Seiten die jeweils *letzte* und auf *rechten* Seiten die jeweils *erste* Überschrift in die Kopfzeile ein.

3.4 Definition der Kopfzeile mit `fancyhdr`

Umfangreichere Möglichkeiten zur Definition der Kopfzeile und der Fußzeile bietet das Erweiterungspaket `fancyhdr`. Dieses definiert Befehle für den Seitenstil `fancy`. Bei `fancyhdr` sind, wie in den Abbildungen 3.1 und 3.2 zu sehen ist, die Kopf- bzw. Fußzeile dreigeteilt (in Links-Mitte-Rechts).

<i>LK-gerade</i>	<i>MK-gerade</i>	<i>RK-gerade</i>
Gerade Seiten		
<i>LF-gerade</i>	<i>MF-gerade</i>	<i>RF-gerade</i>

Abbildung 3.1: Aufteilung von `fancyheadings` bei geraden Seiten [2]

<i>LK-ungerade</i>	<i>MK-ungerade</i>	<i>RK-ungerade</i>
Ungerade Seiten		
<i>LF-ungerade</i>	<i>MF-ungerade</i>	<i>RF-ungerade</i>

Abbildung 3.2: Aufteilung von `fancyheadings` bei ungeraden Seiten [2]

Um `fancyhdr` zu verwenden, muss das Erweiterungspaket in der Präambel der \LaTeX -Quelldatei mit dem Befehl `\usepackage{fancyheadings}` eingebunden und der Seitenstil `fancy` mit dem Befehl `\pagestyle{fancy}` ausgewählt sein.

```
\pagestyle{fancy}
```

Die Definition des Inhalts der in den Abbildungen 3.1 und 3.2 dargestellten Felder der Kopf- bzw. Fußzeilen auf den gerade und ungeraden Seiten geschieht mit den Befehlen `\lhead`, `\chead`, `\rhead`, `\lfoot`, `\cfoot` und `\rfoot`.

```
\lhead[LK-gerade]{LK-ungerade}
\chead[MK-gerade]{MK-ungerade}
\rhead[RK-gerade]{RK-ungerade}
\lfoot[LF-gerade]{LF-ungerade}
\cfoot[MF-gerade]{MF-ungerade}
\rfoot[RF-gerade]{RF-ungerade}
```

Diese Befehle müssen in der Präambel der \LaTeX -Quelldatei, also vor dem `\begin{document}` stehen. Wie in den Abbildungen 3.1 und 3.2 dargestellt, werden die

Informationen in den Feldern *LK* und *LF* linksbündig und in den Feldern *RK* und *RF* rechtsbündig gesetzt. Die Einträge in den Feldern *MK* und *MF* erscheinen zentriert.

Das Aussehen der Linen unter der Kopfzeile und über der Fußzeile wird mit den Befehlen `\headrulewidth` und `\footrulewidth` beeinflusst. Auch diese beiden Befehle müssen in der Präambel der \LaTeX -Quelldatei stehen.

```
\renewcommand{\headrulewidth}{Wert}
\renewcommand{\footrulewidth}{Wert}
```

Sollen die Linen unter der Kopfzeile und über der Fußzeile gar nicht erscheinen, so kann die Strichbreite einfach auf einen Wert `0pt` gesetzt werden.

Die Höhe der Kopfzeile kann mit dem Längenbefehl `\headheight` beeinflusst werden. So ist es möglich mehrzeilige Kopfzeilen zu setzen.

```
\setlength{\headheight}{Wert}
```

Abbildung 3.3 zeigt das voreingestellte Seitenlayout des Erweiterungspaketes `fancyhdr`.

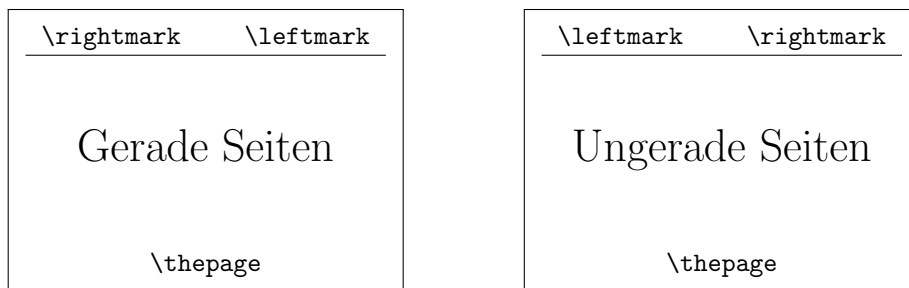


Abbildung 3.3: Voreingestelltes Seitenlayout bei `fancyhdr` [2]

Der Befehl `\thepage` fügt an der Stelle, an der er eingefügt ist, auf jeder Seite die entsprechende Seitenzahl ein.

Der Befehl `\leftmark` fügt bei den Dokumentklassen `book` und `report` standardmäßig die Überschrift des aktuellen Kapitels (`chapter`) auf der jeweiligen Seite ein. `\rightmark` fügt standardmäßig die Überschrift des aktuellen Abschnitts (`section`) auf der jeweiligen Seite ein.

Bei Dokumentklassen, die über keine Kapitel verfügen, wie zum Beispiel `article`, fügt `\leftmark` die Überschrift des aktuellen Abschnitts und `\rightmark` die Überschrift des aktuellen Unterabschnitts ein.

3.5 Seitennummerierung

\LaTeX nummeriert automatisch die Seiten eines Dokuments durch. Der voreingestellte Stil der Seitennummerierung ist dabei `arabic`, also die Nummerierung mit arabischen Ziffern.

Der Befehl `\pagenumbering` ermöglicht die Definition des Seitennummerierungsstils.

```
\pagenumbering{Stil}
```

Weitere, von \LaTeX unterstützte Stile zur Seitennummerierung (siehe Tabelle 3.4) sind die Nummerierung mit kleinen (`roman`) und großen (`Roman`) römischen Ziffern oder mit fortlaufenden Klein- oder Großbuchstaben von a-z (`alph`) bzw. A-Z (`Alph`).

Tabelle 3.4: Seitennummerierungsstile

Stil	Ergebnis
<code>arabic</code>	Nummerierung mit arabischen Ziffern
<code>roman</code>	Nummerierung in kleinen römischen Ziffern
<code>Roman</code>	Nummerierung in großen römischen Ziffern
<code>alph</code>	Nummerierung in Kleinbuchstaben (a-z)
<code>Alph</code>	Nummerierung in Großbuchstaben (A-Z)

Bei den Seitennummerierungsstilen `alph` und `Alph` entsprechen die Buchstaben a-z bzw. A-Z den Zählerwerten 1-26 des Zählers `page` für die Seitenzahlen. Das bedeutet, auch wenn die Seitenzahlen in Buchstaben angezeigt werden, rechnet \LaTeX intern ganz normal mit Zahlen weiter.

Eine Änderung des Seitennummerierungsstils mit `\pagenumbering` setzt den \LaTeX -internen Zähler der Seitennummern `page` automatisch auf den Wert 1 zurück.

Um den Startwert der Seitennummerierung auf einen anderen Wert als 1 zu setzen, muss der Zähler `page` auf einen höheren Wert gesetzt werden.

```
\setcounter{page}{Nummer}
```

Sollen beispielsweise das Vorwort und das Inhaltsverzeichnis eines Dokuments mit (großen) römischen Ziffern und das eigentliche Dokument mit arabischen Ziffern durchnummeriert werden, kann dieses auf folgende Art und Weise realisiert werden: Nach dem `\begin{document}` wird mit `\pagenumbering{Roman}` der Seitennummerierungsstil auf große römische Ziffern festgelegt und direkt nach dem ersten `\chapter`-Befehl erfolgt mit `\pagenumbering{arabic}` ein Wechsel auf arabische Ziffern. Die Struktur des Dokuments ist also so oder so ähnlich:

```
<Präambel der LaTeX-Quelldatei>
\begin{document}
\pagenumbering{Roman}
...
\tableofcontents
...
\chapter{Kapitel 1}
\pagenumbering{arabic}
...
```

3.6 Wichtige Abstände

In diesem Abschnitt ist die Anpassung wichtiger Abstände im Dokument beschrieben. Bei diesen handelt es sich um den Zeilenabstand, den Absatzabstand und die Tiefe der Einrückung der ersten Zeile eines Absatzes.

3.6.1 Zeilenabstand

Der Abstand zwischen zwei aufeinander folgenden Zeilen heißt Zeilenabstand. Diesen definiert der Wert in der Variablen `\baselineskip`. Der \LaTeX -Compiler legt den Zeilenabstand automatisch fest und berücksichtigt dabei die aktuelle Schriftgröße.

Um den Zeilenabstand zu ändern, sollte nicht `\baselineskip` einen neuen Wert zugewiesen bekommen. Gründe dafür sind:

- Der Wert für `\baselineskip` kann nicht in der Präambel definiert werden.
- Bei jeder Änderung der Schriftgröße wird eine selbst vorgenommene Änderung des Zeilenabstandes überschrieben, da eine Änderung der Schriftgröße auch eine Änderung des Zeilenabstandes von \LaTeX nach sich zieht.
- Wird `\baselineskip` innerhalb eines Absatzes geändert, gilt das rückwirkend für den gesamten Absatz.
- Wird innerhalb eines Absatzes mehrmals mit `\baselineskip` der Zeilenabstand geändert, gilt nur die letzte Änderung im Absatz, da diese alle anderen Änderungen rückwirkend überschreibt.

Besser als eine Änderung des Werts von `\baselineskip` ist es, den Längenbefehl `\baselinestretch` zu verwenden. Dieser enthält einen Faktor, mit dem der normale Zeilenabstand (also `\baselineskip`) multipliziert wird. Standardmäßig hat `\baselinestretch` den Wert 1. Die Zuweisung eines neuen Werts geschieht folgenderweise:

```
\renewcommand{\baselinestretch}{Faktor}
```

Wird der Wert des Längenbefehls `\baselinestretch` in der Präambel geändert, gilt diese Änderung für das gesamte Dokument.

Wird der Wert von `\baselinestretch` außerhalb der Präambel geändert, dann gilt die Änderung des Zeilenabstandes erst nach dem nächsten Schriftgrößenwechsel. Darum macht es Sinn, direkt nach dem Festlegen des neuen Zeilenabstandes einen Schriftgrößenbefehl zu bringen. Im einfachsten Fall, wenn die Schriftgröße gleich bleiben soll, geschieht das mit dem Befehl `\normalsize`.

3.6.2 Absatzabstand

Der Abstand, den der \LaTeX -Compiler zwischen zwei Absätzen einfügt, hängt vom Wert des Längenbefehls `\parskip` ab.

Es ist ratsam, für die Definition des Absatzabstands die Maßeinheit `ex` (siehe Seite 11) zu verwenden, da diese sich an der aktuellen Zeichengröße orientiert und den Absatzabstand als elastisches Maß (siehe Seite 12) zu definieren, damit \LaTeX das Entstehen von *Hurenkindern* und *Schusterjungen* verhindert. Mit diesen Begriffen bezeichnet man den Makel beim Textsatz, dass eine Seite mit der ersten Zeile eines Absatzes endet (so genannte *Schusterjungen*) oder aber eine neue Seite mit der letzten Zeile eines Absatzes beginnt (so genannte *Hurenkinder*).

Die folgende Zeile beispielsweise setzt den Absatzabstand auf `1.5ex` plus/minus `0.5ex`:

```
\setlength{\parskip}{1.5ex plus 0.5ex minus 0.5ex}
```

3.6.3 Einrückung der ersten Zeile eines Absatzes

Bei englischsprachigen Dokumenten ist es üblich, keinen zusätzlichen Freiraum zwischen Absätzen einzufügen, und dafür (außer beim ersten Absatz) immer die erste Zeile eines Absatzes um einen festen Abstand (z.B. einen halben Zentimeter) einzurücken.

Auch \LaTeX nimmt diese Einrückung standardmäßig vor. Wie es aussieht, wenn immer die erste Zeile eines Absatzes eingerückt wird, zeigt dieser Abschnitt. Dabei ist die erste Zeile immer um 1.5em eingerückt. Der Betrag, um den \LaTeX die erste Zeile eines jeden Absatzes nach rechts einrückt, ist im Längenbefehl `\parindent` gespeichert.

Das Einrückverhalten von \LaTeX können Autoren anpassen, indem sie mit dem Befehl `\setlength` dem Längenbefehl `\parindent` einen anderen Wert zuweisen. Die Syntax ist dabei folgende:

`\setlength{\parindent}{Betrag}`

Um das automatische Einrücken zu unterbinden, genügt es also, dem Längenbefehl `\parindent` in der Präambel der \LaTeX -Quelldatei den Wert `0em` zuzuweisen.

Um nur das Einrückverhalten des nächsten Absatzes zu beeinflussen, existiert der Befehl `\noindent`. Dieser verhindert das Einrücken des nächsten Absatzes, hat aber auch nur für den nächsten Absatz Gültigkeit.

`\noindent`

Zudem existiert der Befehl `\indent`, der das Einrücken nur des nächsten Absatzes anweist.

`\indent`

3.7 Parameter zur Seitendeklaration

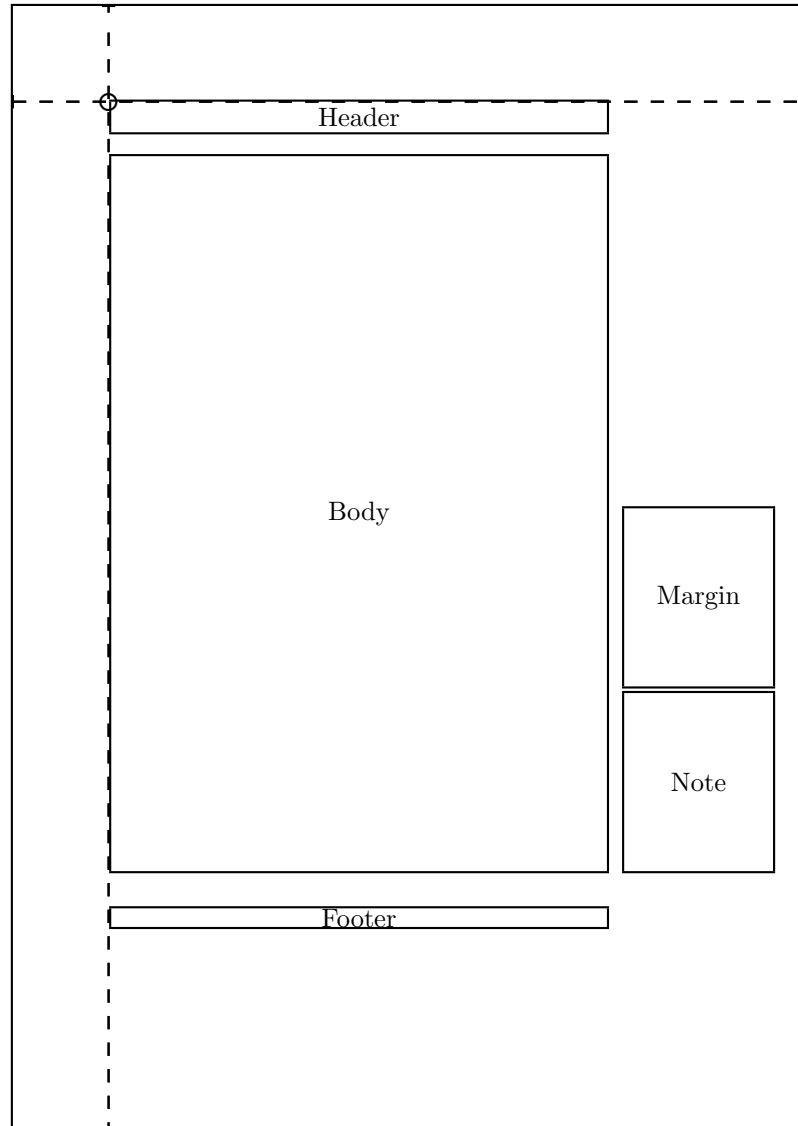
Dieser Abschnitt beschreibt einige Längenbefehle (siehe Tabelle 3.5), die die geometrischen Dimensionen des Layouts beeinflussen. Diese Befehle sind nur dann für Autoren relevant, wenn sie von den Standardwerten, die von der verwendeten Dokumentklasse abhängen, abweichen wollen.

Jede Seite besteht aus einer Kopfzeile (*Head*), einem Textkörper (*Body*) und einer Fußzeile (*Foot*).

Das Zuweisen neuer Werte zu den Längenbefehlen in Tabelle 3.5 erfolgt mit dem Befehl `\setlength` (siehe Seite 32).

Das Erweiterungspaket `layouts.sty` ermöglicht die Ausgabe einer übersichtlichen Darstellung des aktuellen Seitenlayouts. Um eine Darstellung, wie in Abbildung 3.4 zu auszugeben, lassen Sie mit dem Befehl `\currentpage` die aktuellen Werte der Längenbefehle auslesen, legen mit dem Befehl `\oddpagelayouttrue` oder alternativ mit dem Befehl `\oddpagelayoutfalse` fest, ob Sie das Layout einer ungeraden oder einer geraden Seiten ausgeben wollen und setzen die Übersicht mit dem Befehl `\pagedesign`.

The circle is at 1 inch from the top and left of the page. Dashed lines represent (`\hoffset + 1 inch`) and (`\voffset + 1 inch`) from the top and left of the page.



Lengths are to the nearest pt.

page height = 845pt	page width = 598pt
<code>\hoffset</code> = 0pt	<code>\voffset</code> = 0pt
<code>\oddsidemargin</code> = 2pt	<code>\topmargin</code> = 0pt
<code>\headheight</code> = 23pt	<code>\headsep</code> = 18pt
<code>\textheight</code> = 538pt	<code>\textwidth</code> = 373pt
<code>\footskip</code> = 42pt	<code>\marginparsep</code> = 13pt
<code>\marginparpush</code> = 5pt	<code>\columnsep</code> = 10pt
<code>\columnseprule</code> = 0.0pt	

Abbildung 3.4: Layout ungerader Seiten im aktuellen Dokument

Tabelle 3.5: Längenbefehle zur Beeinflussung des Layouts [2]

Befehl	Bedeutung
<code>\columnsep</code>	Abstand der Spalten bei Mehrspaltensatz
<code>\columnseprule</code>	Breite der Trennline, die die Spalten beim Mehrspaltensatz voneinander trennt (normalerweise <code>0pt</code> , also unsichtbar)
<code>\columnwidth</code>	Spaltenbreite beim Mehrspaltensatz. Hängt von <code>\textwidth</code> und <code>\columnsep</code> ab
<code>\evensidemargin</code>	Zusätzlicher linker Rand auf geraden Seiten (bei zweiseitigem Satz)
<code>\footskip</code>	Vertikale Abstand zwischen der Grundlinie der letzten Textzeile und der Grundlinie der Fußzeile
<code>\headheight</code>	Höhe der Kopfzeile
<code>\headsep</code>	Vertikale Abstand zwischen Kopfzeile und Textkörper
<code>\linewidth</code>	Breite der aktuellen Zeile
<code>\marginparpush</code>	Vertikale Mindestabstand zwischen zwei aufeinander folgenden Marginalien
<code>\marginparsep</code>	Horizontaler Abstand zwischen Textkörper und Marginalien
<code>\marginparwidth</code>	Breite der Marginalien
<code>\oddsidemargin</code>	Zusätzlicher linker Rand auf ungeraden Seiten (bei zweiseitigem Satz)
<code>\paperheight</code>	Höhe des Papiers
<code>\paperwidth</code>	Breite des Papiers
<code>\topmargin</code>	Abstand vom oberen Rand des Papiers bis zum oberen Rand der Kopfzeile
<code>\textheight</code>	Höhe des Textkörpers ohne Kopf- und Fußzeile
<code>\textwidth</code>	Breite des Textkörpers

3.8 Mehrspaltenlayout

Soll ein Dokument ab einer bestimmten Stelle zweiseitig gesetzt werden, muss dafür an der gewünschten Stelle der Befehl `\twocolumn` aufgerufen werden.

An der Stelle, wo der Befehl aufgerufen wird, beginnt \LaTeX eine neue Seite mit zweispaltigem Layout. Ist eine Überschrift gewünscht, die über beiden Spalten steht, wird diese als optionales Argument in eckigen Klammern hinter dem Befehl `\twocolumn` angegeben.

`\twocolumn[Überschrift über beide Spalten]`

Eine Eigenschaft von `\twocolumn` ist, dass es immer dafür sorgt, dass ganze Seiten zweispaltig gesetzt werden. Dabei wird immer zuerst die linke Spalte und dann die rechte Spalte gefüllt. So kann es zu einer sehr unausgewogenen Verteilung des Textes kommen.

Soll wieder auf ein einspaltiges Layout umgeschaltete werden, geschieht dieses mit dem Befehl `\onecolumn`.

`\onecolumn`

3.9 Querformat

Beim Setzen von Text ist es üblich, dass die längere Papierseite die vertikale ist und die kürzere die horizontale. Dieses Konzept heißt *Hochformat* oder *Portrait*. Für die allermeisten Arten von Dokumenten ist dieses das übliche Vorgehen.

Für einige Anwendungen, zum Beispiel wenn Aushänge mit breiten Bildern oder Tabellen gesetzt werden sollen, empfiehlt es sich allerdings die längere Papierseite als die Horizontale zu verwenden. In diesem Fall spricht man vom *Querformat*.

Um Seiten im Querformat zu setzen, ist das Makropaket `portland` nützlich. Dieses definiert die Befehle `\landscape` und `\portrait` sowie die beiden Umgebungen:

- `\begin{portrait}...\end{portrait}` für das Hochformat und
- `\begin{landscape}...\end{landscape}` für das Querformat

Der Befehl `\portrait` setzt das Seitenlayout auf die ursprünglichen Werte des Hochformats, so wie sie zu Beginn eines Dokuments nach dem `\begin{document}` vorgegeben sind. Beim Befehl `\landscape` werden die horizontalen und vertikalen Werte vertauscht. Das Textfeld belegt also im Landscape-Modus die gleiche Fläche und die gleichen Koordinaten wie im Portrait-Modus.

An der Stelle im Quelltext, wo der Befehl `\portrait` oder alternativ der Befehl `\landscape` aufgerufen wird, beginnt der \LaTeX -Compiler eine neue Seite, die dann entweder im Hoch- oder Querformat erscheint. Die beiden Befehle `\portrait` und `\landscape` gelten aber immer nur für die nächste Seite. Sollen mehrere aufeinander folgende Seiten im Querformat gesetzt werden, ist es bequemer, die Umgebung `\landscape` zu nutzen, als für jede Seite den Befehl `\landscape` aufzurufen.

3.10 Befehle und Umgebungen zur Gliederung

Die Gliederung von Dokumenten unterstützt deren Lesbarkeit. Dieser Abschnitt stellt Befehle und Umgebungen zur Realisierung der Titelseite, der Zusammenfassung (Abstract), von Kapiteln, Abschnitten und Unterabschnitten, sowie Anhängen vor.

3.10.1 Titelei

Eine Titelseite kann unter L^AT_EX entweder komplett manuell definiert, oder komfortabel mit Hilfe weniger Befehle generiert werden. Die zweite Alternative geht deutlich schneller, bietet dafür aber auch weniger Freiheiten als die erste Alternative.

Soll die Titelseite automatisch generiert werden, geschieht dieses mit Hilfe der Befehle `\maketitle`, `\title`, `\author`, `\date` und `\thanks`.

Der Befehl `\title` dient zur Definition des Dokumenttitels.

```
\title{Text}
```

Mit dem Befehl `\author` können Name und Adresse des Autors oder auch mehrerer Autoren definiert werden.

```
\author{Text}
```

Gibt es mehrere Autoren, so ist es möglich, deren Namen mit dem Befehl `\and` zu verknüpfen. Sollen beispielsweise die Namen und Adressen von zwei Autoren auf dem Titel erscheinen, ist eine Lösung wie die folgende denkbar:

```
\author{erster Autor \\Adresse des \\ersten Autors \and  
zweiter Autor \\Adressse des \\zweiten Autors}
```

Auf der Titelseite würde dieses Beispiel zu folgendem Ergebnis führen:

erster Autor	zweiter Autor
Adresse des	Adresse des
ersten Autors	zweiten Autors

Der Befehl `\date` setzt einen zentrierten Textblock unterhalb der Angaben zu den Autoren. Dieser Textblock kann ein Datum oder sonst eine Information enthalten.

```
\date{Datum}
```

Eine einfache Möglichkeit, das aktuelle Datum einzufügen, ist die Verwendung der Befehlskombination `\date{\today}`, denn der Befehl `\today` fügt an der Stelle, an der er aufgerufen wird, das aktuelle Datum ein.

Fehlt der Befehl `\date`, fügt L^AT_EX automatisch das aktuelle Datum auf die Titelseite ein. Ist keine Datumsangabe auf der Titelseite gewünscht, sollte der Befehl `\date` ohne Argumente, also mit leeren geschweiften Klammern im Quelltext stehen. Das unterdrückt die automatische Datumsangabe durch L^AT_EX.

Der Befehl `\thanks` setzt auf der Titelseite eine einzeilige Fußnote.

```
\thanks{Text}
```

Eine häufige Anwendung von `\thanks` ist, wenn mehrere Autoren an einem Dokument gearbeitet haben und weitere Informationen zu deren Beitrag zum Dokument, zu deren Erreichbarkeit oder eventuell zu deren Arbeitgebern auf der Titelseite erwähnt sein sollen.

Der Befehl `\maketitle`, der die Titelseite nach den gemachten Vorgaben setzt, sollte sich im Quelltext nach den Befehlen `\title`, `\author`, `\date`, `\thanks` und nach dem `\begin{document}` befinden.

```
\maketitle
```

3.10.2 Die Zusammenfassung – das Abstract

Als *Abstract* bezeichnet man einen (kurzen) Text, der den Inhalt des Dokuments zusammenfasst. Es steht abhängig von der Dokumentklasse auf der Titelseite, beidseitig eingerückt unter dem Titel oder auf einer eigenen Seite nach der Titelseite.

Um solche Zusammenfassungen unter L^AT_EX zu setzen, existiert die Umgebung `abstract`.

```
\begin{abstract}...\end{abstract}
```

In der Dokumentklasse `book` ist die Umgebung `abstract` nicht definiert. Eine Zusammenfassung am Anfang eines Buches wäre auch sehr ungewöhnlich. Wenn, dann finden sich solche Zusammenfassungen auf dem Cover eines Buches oder in Buchbesprechungen, und diese werden nicht in der Klasse `book` erstellt [1].

Die Dokumentenklasse `article` hat standardmäßig keine extra Seite für die Titelseite. Aus diesem Grund setzt der L^AT_EX-Compiler bei `article` die Zusammenfassung auf der ersten Seite unter der Überschrift. Wurde allerdings die Option `titlepage` bei der Deklaration der Dokumentklasse verwendet, gibt es doch eine eigene Seite für

die Titelseite und die Zusammenfassung erscheint auf der Titelseite unter dem Titel, beidseitig eingerückt und in einer kleinen Schriftgröße.

Bei der Dokumentklasse `report` erscheint die Zusammenfassung auf einer separaten Seite ohne Seitennummer nach der Titelseite. Die Schriftgröße der Zusammenfassung entspricht der Standardgröße. Eine beidseitige Einrückung wie bei der Dokumentklasse `article` erfolgt nicht.

3.10.3 Gliederungsbefehle

\LaTeX enthält für die fortlaufende Untergliederung von Dokumenten in Teile, Kapitel, Abschnitte, Paragraphen usw. entsprechende Befehle. Tabelle 3.6 enthält eine Zusammenstellung der unter \LaTeX verfügbaren Gliederungsbefehle und der zugehörigen Werte des Zählers `tocdepth`.

Tabelle 3.6: Gliederungsebenen und der Zähler `tocdepth` [31]

Gliederungsebene	Wert
<code>\part</code>	-1 bei <code>book</code> und <code>report</code> , 0 bei <code>article</code>
<code>\chapter</code>	0
<code>\section</code>	1
<code>\subsection</code>	2
<code>\subsubsection</code>	3
<code>\paragraph</code>	4
<code>\subparagraph</code>	5

Alle diese Befehle zur Untergliederung von Dokumenten, mit Ausnahme von `\part` (*Teil*), bauen eine Gliederungshierarchie auf. Bei den Dokumentklassen `book` und `report` beginnt die Gliederung mit *Kapiteln* (`\chapter`). Die Kapitel werden untergliedert in *Abschnitte* (`\section`) und diese wiederum in *Unterabschnitte* (`\subsection`) und so weiter [1].

Bei den Dokumentklassen `article` und `proc` beginnt die Gliederung erst mit *Abschnitten* (`\section`). *Kapitel* (`\chapter`) stehen hier nicht zur Verfügung.

Die Überschrift wird den Gliederungsbefehlen aus Tabelle 3.6 in geschweiften Klammern übergeben. Zudem kann als optionales Argument in eckigen Klammern eine Kurzform der Überschrift definiert werden. Diese wird anstatt der langen Überschrift im Inhaltsverzeichnis und in der Kopfzeile verwendet.

<code>\Gliederungsbefehl[Kurzform]{Überschrift}</code>
--

Bei den Dokumentklassen `book` und `report` erhält die Gliederungsüberschrift eines Kapitels (`\chapter`) eine einstellige Nummer, die eines *Abschnitts* (`\section`) eine zweistellige und die Überschrift eines *Unterabschnitts* (`\subsection`) eine dreistellige Nummer. Gliederungsbefehle, die in der Hierarchie tiefer als *Unterabschnitte* stehen, erhalten keine Nummer. Wenn die Nummer einer Gliederungsüberschrift mehr als eine Ziffer enthält, dann werden die Ziffern von L^AT_EX jeweils mit einem Punkt voneinander getrennt.

Bei den Dokumentklassen `article` und `proc` erhalten die *Abschnitte* (`\section`) eine einstellige und die *Unterabschnitte* (`\subsection`) eine zweistellige Nummer.

Der Gliederungsbefehl `\part` hat eine Sonderposition unter den Gliederungsbefehlen inne. Er beeinflusst nicht die Nummerierung der anderen Gliederungsbefehle.

Wird mit dem Befehl `\chapter` ein neues Kapitel begonnen, beendet L^AT_EX automatisch die aktuelle Seite und beginnt eine neue. Diese enthält ganz oben im Textfeld eine Zeile „**Chapter x**“. Das **x** steht für die Nummer des Kapitels. Direkt darunter befindet sich die Überschrift des Kapitels. Ist in einem Dokument das Erweiterungspaket `german.sty` oder alternativ `ngerman.sty` eingebunden, setzt der L^AT_EX-Compiler anstatt **Chapter x** das deutsche **Kapitel x**.

Tabelle 3.7: Auswirkungen von (n)`german.sty` auf die Gliederungselemente

Befehl	Standard	mit (n) <code>german.sty</code>
<code>\prefacename</code>	Preface	Vorwort
<code>\refname</code>	References	Literatur
<code>\abstractname</code>	Abstract	Zusammenfassung
<code>\bibname</code>	Bibliography	Literaturverzeichnis
<code>\chaptername</code>	Chapter	Kapitel
<code>\appendixname</code>	Appendix	Anhang
<code>\contentsname</code>	Contents	Inhaltsverzeichnis
<code>\listfigurename</code>	List of Figures	Abbildungsverzeichnis
<code>\listtablename</code>	List of Tables	Tabellenverzeichnis
<code>\indexname</code>	Index	Index
<code>\figurename</code>	Figure	Abbildung
<code>\tablename</code>	Table	Tabelle
<code>\partname</code>	Part	Teil
<code>\pagename</code>	Page	Seite

Tabelle 3.7 enthält eine Übersicht, wie das Erweiterungspaket (n)`german.sty` die Bezeichnungen und Überschriften der Gliederungselemente verändert.

Ist bei einem Gliederungsbefehl weder eine Nummerierung noch ein Eintrag im Inhaltsverzeichnis gewollt, muss ein Stern `*` zwischen Gliederungsbefehl und dem Argument mit der Überschrift stehen.

```
\Gliederungsbefehl*{Überschrift}
```

Die Angabe einer verkürzten Überschrift in eckigen Klammern ist hierbei nicht möglich. Es würde auch keinen Sinn machen, hier eine verkürzte Überschrift anzugeben, da diese eh nie abgedruckt würde.

Bei jedem Aufruf eines Gliederungsbefehls, wird der dazugehörige Zähler um eins erhöht. Wird ein Gliederungsbefehl der `*`-Form aufrufen, wird der Zähler nicht verändert.

3.10.4 Tiefe der Nummerierung bei der Gliederung ändern

Bei der Dokumentklasse `article` ist die Tiefe, bis zu der Gliederungsbefehle durchnummeriert werden können, gleich drei, und bei den Dokumentklassen `book` und `report` ist die Schranke gleich zwei.

Das bedeutet, dass `book` und `report` standardmäßig bis einschließlich `\subsection` und `article` bis einschließlich `\subsubsection` durchnummerieren.

Die Schranke realisiert unter L^AT_EX der Zähler `secnumdepth` und eine Änderung seines Werts ermöglicht der Befehl `\setcounter`.

```
\setcounter{secnumdepth}{Zahl}
```

Würde beispielsweise in einem Dokument mit der Dokumentklasse `book` der Wert `secnumdepth` auf 4 gesetzt, würde der L^AT_EX-Compiler die Gliederungsbefehle bis einschließlich `\paragraph` durchnummerieren. Mögliche Werte für `secnumdepth` sind bei den Dokumentklassen `book` und `report` `-1, 0, 1, ..., 5`. Bei der Dokumentklasse `article` ist der Wertebereich `0, 1, ..., 5`.

3.10.5 Anfangswert der Nummerierung ändern

Die Nummerierung jeder Ebene der Gliederungshierarchie beginnt mit dem Wert 1. Soll der Zähler eines bestimmten Gliederungsbefehls nicht mit 1 beginnen, muss dem Zähler mit dem Befehl `\setcounter` ein anderer Wert zugewiesen werden.

```
\setcounter{Gliederungsname}{Zahl}
```

Dieser Befehl legt den Startwert des Zählers des angegebenen Gliederungsbefehls fest. *Gliederungsname* steht in der Syntax für einen Gliederungsbefehl, allerdings ohne den üblichen Backslash `\` am Anfang des Befehls.

Der Befehl `\setcounter{section}{5}` zum Beispiel setzt den Zähler von `\section` auf den Wert 5. Beim nächsten Aufruf von `\section` wird der Zähler um eins erhöht, und damit bekommt der nächste Abschnitt (`\section`) die Nummer 6 und der folgende Abschnitt die Nummer 7 und so weiter.

Ein Anwendungsbeispiel, wo die manuelle Definition des Zählers eines Gliederungsbefehls hilfreich ist, sind große Dokumente, die in mehrere unabhängige L^AT_EX-Dateien unterteilt sind, und die zusammen wie ein großes Dokument aussehen sollen.

3.10.6 Zusätzliche Untergliederung bei Büchern

Die Dokumentklasse `book` ermöglicht noch eine weitere Möglichkeit der Untergliederung. Es handelt sich hierbei um eine Untergliederung in einen Buchvorspann (u.a. Vorwort), Buchhauptteil (Kapitel und Anhänge) und Buchnachspann (u.a. Literaturverzeichnis und Index). Den Beginn jedes dieser drei Teile markieren die Befehle `\frontmatter`, `\mainmatter` und `\backmatter`.

Der Dokumentteil, der nach dem Befehl `\frontmatter` folgt, erhält von L^AT_EX kleine römische Zahlen für die Seitennummerierung. Gliederungsbefehle in diesem Teil erhalten keine Nummerierung.

```
\frontmatter
```

Im Hauptteil des Buches, der nach dem Befehl `\mainmatter` folgt, erfolgt die Seitennummerierung mit arabischen Ziffern. Der Zähler für die Seitenzahlen wird automatisch zurückgesetzt. Das heißt, dass die erste Seite des Hauptteils eine arabische 1 als Seitenzahl erhält. Gliederungsbefehle in diesem Teil erhalten eine fortlaufende Nummerierung.

```
\mainmatter
```

Im Nachspann des Buches, der nach dem Befehl `\backmatter` folgt, ist die Gliederungsnummerierung wie im Buchvorspann abgeschaltet. Die Seitenzahlen werden aber wie im Hauptteil in arabischen Ziffern geschrieben und der Zähler wird nicht zurückgesetzt.

```
\backmatter
```

3.10.7 Anhang

Zur Realisierung des Anhangs existiert der Befehl `\appendix`.

```
\appendix
```


An der Stelle im Quelltext, wo dieser Befehl aufgerufen wird, setzt \LaTeX bei den Dokumentklassen `book` und `report` den Zähler der Kapitel (`\chapter`) auf null zurück.

Bei der Dokumentklasse `article`, bei der ja keine Kapitel (`\chapter`) existieren, wird der Zähler der Abschnitte (`\chapter`) auf null zurückgesetzt.

Im Anhang nummeriert der \LaTeX -Compiler die Kapitel, bzw. bei `article` die Abschnitte, nicht mehr mit arabischen Ziffern, sondern mit großen Buchstaben. Dabei steht `A` für 1 und `Z` für 26. Außerdem wird das Wort **Chapter** bzw. **Kapitel** (bei `(n)german.sty`) durch das Wort **Appendix** bzw. **Anhang** ersetzt.

Alle in der Hierarchie darunter gelegenen Gliederungsbefehle werden ganz normal mit arabischen Ziffern durchnummeriert. So könnte ein Unterabschnitt im Anhang eines Dokuments mit der Dokumentklasse `book` beispielsweise die Nummer **D.5.3** tragen.

3.11 Inhaltsverzeichnis

Das Inhaltsverzeichnis (*Table of Contents*) enthält die Abschnittsüberschriften mit den zugehörigen Seitennummern, die der \LaTeX -Compiler beim Durchlauf in der `.toc`-Datei ablegt. Diese Datei wird, wenn der Befehl `\tableofcontents` aufgerufen wird und vorausgesetzt sie existiert, bei der Übersetzung eines Dokuments eingelesen und als Inhaltsverzeichnis formatiert auf einer neuen Seite ausgegeben.

`\tableofcontents`

Wegen des Konzepts von \LaTeX schlagen sich Änderungen in der Struktur (Gliederung) des Dokuments sich erst nach zwei Durchläufen des \LaTeX -Compilers im Inhaltsverzeichnis nieder.

Der Grund dafür ist, dass die (aktuellsten) Informationen über das Inhaltsverzeichnis erst nach einer kompletten Abarbeitung des Dokuments vorliegen können. Da aber das Inhaltsverzeichnis normalerweise am Anfang des Dokuments liegt und nicht am Ende, muss bei Änderungen am Inhaltsverzeichnis der \LaTeX -Compiler zwei mal das Dokument durchlaufen. Beim ersten Durchlauf durch das Dokument weist der Befehl `\tableofcontents` den \LaTeX -Compiler an, eine Datei mit dem Stammnamen des Dokuments und der Endung `.toc` anzulegen. In diese Datei schreibt der Compiler die verwendete Sprache, Gliederungsbefehle und ihre zugehörigen Überschriften sowie Seitenzahlen. Diese Datei wird während jeder Abarbeitung des Dokuments befüllt bzw. aktualisiert.

Typische Einträge in der `.toc`-Datei sehen wie folgt aus:

```
\contentsline{Gliederungsname}{Überschrift}{Seitennummer}
```

Gliederungsname steht für den Gliederungsbefehl, der verwendet wurde, also zum Beispiel `\chapter`, `\section` oder `\subsection`. *Überschrift* steht für die Gliederungsüberschrift bzw. für die optionale Kurzform der Überschrift, die in eckigen Klammern zwischen Gliederungsbefehl und Gliederungsüberschrift angeben sein kann. Die *Seitennummer*, der letzte Eintrag einer jeden Zeile, steht für die zugehörige Seitennummer im Dokument.

3.11.1 Manuell Einträge in das Inhaltsverzeichnis einfügen

Einige Situationen erfordern es, eigene Eintragungen in das Inhaltsverzeichnis zu setzen. Denkbare Szenarien sind der Einsatz von Gliederungsbefehlen der *-Form oder wenn ein Gliederungsbefehl (z.B. Literaturverzeichnis oder Index) verwendet wird, der keinen automatischen Eintrag im Inhaltsverzeichnis erzeugt.

Der Befehl `\addcontentsline` ermöglicht das manuelle Einfügen von Einträgen in das Inhaltsverzeichnis.

```
\addcontentsline{toc}{Gliederungsname}{Text}
```

Gliederungsname ist der Name des gewünschten Gliederungsbefehls ohne den für einen Befehl typischen, vorgestellten Backslash `\`. Dieser *Gliederungsname* ist für die Einrückung des Eintrags im Inhaltsverzeichnis verantwortlich. Bei der Dokumentklasse `book` ist ein `chapter`-Eintrag überhaupt nicht eingerückt. Ein `section`-Eintrag hingegen schon, und ein `subsection`-Eintrag noch mehr.

Text ist der neue Eintrag im Inhaltsverzeichnis, der vor der Seitennummer stehen soll.

3.12 Abbildungs- und Tabellenverzeichnis

Bei umfangreichen Dokumenten sind ein Abbildungsverzeichnis und ein Tabellenverzeichnis häufig sinnvoll.

Der Befehl `\listoffigures` fügt an der Stelle im Quelltext, an der er aufgerufen wird, auf einer neuen Seite ein Abbildungsverzeichnis ein.

```
\listoffigures
```

Das Tabellenverzeichnis fügt der Befehl `\listoftables` an der aufgerufenen Stelle auf einer neuen Seite ein.

`\listoftables`

Eintragungen in diese Verzeichnisse erfolgen automatisch immer dann, wenn der Befehl `\caption` aufgerufen wird. Dieser Befehl wird bei den Umgebungen **figure** und **table** angewendet, um die Bildunter- bzw. Tabellenunterschrift zu setzen.

3.13 Querverweise

Mit dem Befehl `\label` werden in Dokumenten unsichtbare Markierungen eingefügt, auf die mit den Befehlen `\ref` und `\pageref` referenziert werden kann.

`\label{Markierung}`
`\pageref{Markierung}`
`\ref{Markierung}`

Die *Markierung*, die dem Befehl `\label` als Argument übergeben wird, kann eine fast beliebige Folge von Buchstaben, Zahlen und Zeichen aller Art sein. Auf die Verwendung der Sonderzeichen aus Tabelle 2.2 sollte hier in jedem Fall verzichtet werden.

Mit dem Befehl `\pageref` ist es möglich, auf eine Seitenzahl zu referenzieren. An der Position, an der der Befehl `\pageref` auftritt, erscheint die Seitennummer, an der sich die *Markierung* befindet, auf die die `\pageref` referenziert.

Ein weiterer Befehl, um Querverweise zu erzeugen, ist der Befehl `\ref`. Mit diesem ist es möglich auf eine Gleichungs-, Gliederungs-, Tabellen- oder Bildnummer zu referenzieren. Was konkret referenziert wird, hängt davon ab, ob sich der Befehl `\label` in einer mathematischen Umgebung wie zum Beispiel `math` oder `displaymath` befindet, oder in nach einem Gliederungsbefehl (z.B. `\chapter`, `\section` oder `\subsection`), oder in einer Tabellenumgebung (z.B. `table`) oder in der Umgebung **figure** zum Satz einer Abbildung, etc. Weitere sinnvolle Anwendungen, wie zum Beispiel die Referenzierung auf Fußnoten sind denkbar.

3.14 Index

Die Erzeugung eines Indexregisters (Stichwortverzeichnis) geschieht weitgehend automatisiert. Hierfür muss das Erweiterungspaket `makeidx` mit dem Befehl `\usepackage{makeidx}` in der Präambel der `.tex`-Quelldatei eingebunden sein. Zudem müssen im Quelltext des Dokuments an den entsprechenden Stellen die Einträge für das Indexregister mit dem Befehl `\index` definiert sein.

`\index{Indexeintrag}`

Zusätzlich muss in der Präambel des Dokuments der Befehl `\makeindex` stehen und an der Stelle im Dokument, an der das Indexregister erscheinen soll, ist der Befehl `\printindex` nötig.

`\makeindex`
`\printindex`

Die Schritte zur Erstellung des Stichwortverzeichnisses sind wie folgt:

Nachdem die Quelldatei vom \LaTeX -Compiler abgearbeitet wurde...

1) `pdflatex dateiname.tex`

erzeugt der \LaTeX -Compiler eine Datei `dateiname.idx`, die die Einträge für das Indexregister enthält. Anschließend muss ein Durchlauf des Programms `makeindex` folgen:

2) `makeindex dateiname.idx`

Das Programm `makeindex` legt eine Datei `dateiname.ind` an, die ebenfalls alle Einträge für das Indexregister enthält, aber im Gegensatz zu `dateiname.idx` sind die Einträge hier alphabetisch sortiert und die Seitenzahlen beigefügt.

Ein abschließender Durchlauf des \LaTeX -Compilers sorgt dafür, dass das alphabetisch korrekt sortierte Indexregister an der Stelle im Dokument eingefügt wird, wo der Befehl `\printindex` eingefügt ist.

3) `pdflatex dateiname.tex`

4 Formatierungshilfen

Dieses Kapitel beschreibt Formatierungshilfen und verschiedene Möglichkeiten, um das Layout von Dokumenten zu beeinflussen. Konkret stellt dieses Kapitel Befehle vor, um Absätze auszurichten, Zeilen umzubrechen, Abstände zwischen Wörtern und Zeichen einzufügen, sowie Leerräume aller Art zu erzeugen.

4.1 Abstände von Wörtern und Zeichen

Der \LaTeX -Compiler ermöglicht es auf mehrere Arten Abstände zwischen Wörtern und Zeichen zu realisieren. Die Möglichkeiten reichen von gewöhnlichen Leerzeichen und Abständen nach Satzzeichen bis zu Leerräumen aller Art.

4.1.1 Abstände in Sätzen

\LaTeX interpretiert einen Punkt, genau wie ein Ausrufezeichen, ein Fragezeichen oder einen Doppelpunkt, der nach einem Kleinbuchstaben oder einer Ziffer kommt, als das Ende eines Satzes. Nach dem Punkt fügt \LaTeX automatisch einen Zwischenraum ein damit der neue Satz nicht direkt nach dem Punkt folgt.

Kommt innerhalb eines Satzes ein Punkt vor, ist der vergleichsweise große Zwischenraum, der am Ende eines Satzes eingefügt wird, meist unerwünscht. Beispiele hierfür sind: »Dr. Meier«, »i. Allg.« oder »u. a.«.

In einem solchen Fall kann mit einer Tilde (\sim) oder mit dem Befehl \backslash_\square ein Wortabstand eingefügt werden, der kleiner ist als der übliche Abstand zwischen Sätzen. \backslash_\square steht an dieser Stelle für einen Backslash, direkt gefolgt von einem Leerzeichen.

\sim

\backslash_\square

Der Abstand, den \sim und \backslash_\square realisieren, ist identisch. Der einzige Unterschied zwischen beiden Vorgehensweisen ist, dass bei der Verwendung von \sim an der Stelle, an der dieses

Zeichen eingesetzt wird, ein Zeilenumbruch vermieden wird, was besonders bei Namen sinnvoll ist.

Bei Abkürzungen empfiehlt sich der Einsatz eines Backslash und eines Kommas `\`, um einen kleinen Zwischenraum einzufügen, bei dem ebenfalls kein Zeilenumbruch möglich ist.

Tabelle 4.1 enthält einige Beispiele, bei denen ein Abstand, wie er zwischen Sätzen vorkommt, zu groß wäre.

Tabelle 4.1: Beispiele für Abstände in Sätzen

Beispiel	Quelltext
Prof. Dr. Meier	<code>Prof.~Dr.~Meier</code>
i. Allg.	<code>i.\,Allg.</code>
u. a.	<code>u.\,a.</code>
1. Stock	<code>1.~Stock</code>
25 m	<code>25~m</code>
z. B.	<code>z.\,B.</code>
Dt. Bundesbahn	<code>Dt.\,Bundesbahn</code>
Dipl. Inf. (FH)	<code>Dipl.\,Inf.\,(FH)</code>

4.1.2 Horizontale Abstände

Das manuelle Einfügen fester Abstände kann helfen, rasch ein Problem mit dem Layout zu lösen. Dennoch sollten Autoren dieses nur in Ausnahmefällen tun, weil es dem Prinzip der Trennung von Layout und Inhalt widerspricht. Zudem besteht die Gefahr, dass sich beim weiteren Einfügen von Texten, Bildern oder Tabellen im Dokument diese festen Abstände verschieben und dadurch den optischen Eindruck negativ beeinflussen.

Ein horizontaler Abstand kann unter anderem mit dem Befehl `\hspace` eingefügt werden.

`\hspace{Abstand}`

Der Befehl erfordert die Angabe des einzufügenden Abstands inklusive einer Maßeinheit (siehe Abschnitt 2.4) in geschweiften Klammern.

Bei einem Zeilenumbruch oder ganz am Anfang einer Zeile wird `\hspace` vom L^AT_EX-Compiler ignoriert.

Eine Variante des Befehls `\hspace` ist der Befehl `\hspace*`. Dieser fügt im Gegensatz zu `\hspace` den Zwischenraum auch dann ein, wenn an der Stelle des Befehlsaufrufs

gerade ein Zeilenumbruch stattfindet oder wenn er am Anfang einer Zeile aufgerufen wird. Einfach ausgedrückt: Die *-Form von `\hspace` erzwingt den gewünschten, horizontalen Zwischenraum auf alle Fälle.

<code>\hspace*{Abstand}</code>

Da jeweils ein Leerzeichen vor und nach den Befehlen `\hspace` und `\hspace*` von \LaTeX mitgezählt wird, ist es für das Ergebnis nicht unerheblich, ob diese Befehle direkt nach einem Wort aufgerufen werden, oder ob sich noch ein Leerzeichen zwischen dem Wort und dem Befehl befindet.

<code>1cm\hspace{1cm}einfügen \newline</code>	1cm	einfügen
<code>1cm \hspace{1cm}einfügen \newline</code>	1cm	einfügen
<code>1cm \hspace{1cm} einfügen</code>	1cm	einfügen

Es ist auch möglich, den Befehlen `\hspace` und `\hspace*` negative Werte zu übergeben. Dann wird \LaTeX die aktuelle Position um die angegebene Länge zurücksetzen. In so einem Fall ist es allerdings möglich, dass Text oder andere Inhalte überschrieben werden.

Einen Text <code>\hspace{-1cm}</code> überschreiben	Einen Text überschreiben
---	-------------------------------------

Weitere Befehle, die horizontale Abstände einfügen, sind `\enskip`, `\quad` und `\qquad`.

Wie groß der konkrete Abstand ist, der von diesen Befehlen eingefügt wird, hängt von der Länge der relativen Maßeinheit `em` (siehe Abschnitt 2.4), also von der aktuell verwendeten Schriftart ab.

Der Befehl `\enskip` fügt einen horizontalen Abstand von einem halben `em` ein, während die Befehle `\quad` und `\qquad` Abstände von einem bzw. zwei `em` einfügen.

<code>\enskip</code>
<code>\quad</code>
<code>\qquad</code>

Zudem gibt es noch die bereits in den Abschnitten 4.1.1 und 4.1.6 gezeigten Möglichkeiten zum Einfügen kleiner Wort- bzw. Zeichenabstände mit `~`, `\,`, `_` und `\hfill`.

4.1.3 Vertikale Abstände

Ein vertikaler Abstand kann unter anderem mit dem Befehl `\vspace` eingefügt werden. Der Befehl erfordert die Angabe des einzufügenden Abstands inklusive einer Maßeinheit (siehe Abschnitt 2.4) in geschweiften Klammern.

`\vspace{Abstand}`

Eine Variante des Befehls `\vspace` ist der Befehl `\vspace*`. Dieser fügt im Gegensatz zu `\vspace` den Zwischenraum auch ein, wenn an der Stelle des Befehlsaufrufs gerade ein Seitenumbruch stattfindet. Einfach ausgedrückt: Die *-Form von `\vspace` erzwingt den gewünschten, vertikalen Zwischenraum auf alle Fälle.

`\vspace*{Abstand}`

Beim Aufruf von `\vspace` oder `\vspace*` innerhalb eines Absatzes wird der L^AT_EX-Compiler die aktuelle Zeile fertig mit Text befüllen und erst danach den vertikalen Zwischenraum einfügen.

Es ist auch möglich, einen negativen Abstand einzufügen, und somit einen Bereich im Dokument *nach oben zu rücken*. Allerdings besteht auch hier – genau wie bei `\hspace*` – das Risiko, Inhalte zu überschreiben.

Weitere Befehle zum einfügen vertikaler Abstände, auf die aus Platzgründen an dieser Stelle nicht weiter eingegangen wird, sind `\smallskip`, `\medskip` und `\bigskip`, sowie `\smallbreak`, `\medbreak` und `\bigbreak`. Bei diesen hängt der konkrete Abstand von der verwendeten Dokumentklasse (siehe Abschnitt 3.1) ab.

4.1.4 Leerzeichen in Verbindung mit Satzzeichen – Frenchspacing

Einige Textverarbeitungsprogramme fügen nach einem Satzzeichen einen freien Zwischenraum ein, der größer ist als der normale Wortabstand. Diesen zusätzlichen Zwischenraum fügt auch L^AT_EX ein, wenn es dazu mit dem Befehl `\nonfrenchspacing` angewiesen wird.

`\nonfrenchspacing`

Die Wirksamkeit des Befehls erstreckt sich von seinem Aufruf (der an jeder beliebigen Stelle eines Dokuments sein kann) bis zum Ende des Dokuments oder bis zu einem Aufruf des Befehls `\frenchspacing`. Dieser Befehl sorgt dafür, dass alle Wortabstände innerhalb einer Zeile gleich sind.

`\frenchspacing`

Einen Vergleich der Auswirkungen der beiden Befehle zeigt Abbildung 4.1. Das Erweiterungspaket `(n)german.sty` verwendet standardmäßig `\frenchspacing`.

Mit `\nonfrenchspacing`

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac. Vestibulum quis pellentesque diam. Aliquam erat volutpat. Nunc gravida vestibulum lacus eu feugiat. Phasellus sit amet tellus nibh. Phasellus lobortis quam vel enim efficitur vestibulum. Duis eleifend eros a erat vehicula porta. Nam.


Mit `\frenchspacing`

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac. Vestibulum quis pellentesque diam. Aliquam erat volutpat. Nunc gravida vestibulum lacus eu feugiat. Phasellus sit amet tellus nibh. Phasellus lobortis quam vel enim efficitur vestibulum. Duis eleifend eros a erat vehicula porta. Nam.


Abbildung 4.1: Auswirkungen von `\nonfrenchspacing` und `\frenchspacing`

4.1.5 Leerzeichen darstellen

Die Eingabe eines Leerzeichens mit geschieht einfach durch die direkte Eingabe mit der Leertaste. Stehen im Quelltext mehrere Leerzeichen hintereinander, wird der L^AT_EX-Compiler dieses ignorieren und nur ein einziges Leerzeichen ausgeben.

Ein Leerzeichen kann auch mit Hilfe des Befehls `\textvisiblespace` gesetzt werden. Der Befehl setzt folgendes Zeichen: „“. So etwas ist z. B. bei Dokumentationen hilfreich.

`\textvisiblespace`

Eine alternative Möglichkeit, um ein Leerzeichen zu setzen, ist der Befehl `\space`. Der erzeugt auch ein richtiges Leerzeichen, so wie dieses hier: „“

`\space`

4.1.6 Zeilen mit Zwischenräumen auffüllen

Der Befehl `\hfill` erzeugt einen Zwischenraum, der an der Stelle des Aufrufs so viel freie Fläche erzeugt, dass die Zeile links und rechts bündig abschließt. Der Befehl ist eine Abkürzung für `\hspace{\fill}` (siehe Abschnitt 4.1.2).

`\hfill`

Diese Zeile:

Anfang\hfill Ende

hat den folgenden Effekt:

Anfang Ende

Wird innerhalb einer Zeile mehrmals `\hfill` aufgerufen, dann wird der einzufügende Zwischenraum entsprechend geteilt, so dass jedes `\hfill` ein gleich großes Stück erhält.

Diese Zeile:

```
Anfang\hfill Mitte\hfill Ende
```

hat den folgenden Effekt:

Anfang Mitte Ende

Es ist auch möglich, mehrere `\hfill` zu einem Zwischenraum zu kombinieren. Jedes `\hfill` einer Zeile bekommt seinen Anteil am möglichen Zwischenraum.

Diese Zeile:

```
Eins\hfill Zwei\hfill\hfill Drei\hfill\hfill\hfill Ende
```

hat den folgenden Effekt:

Eins Zwei Drei Ende

4.1.7 Zwischenräume mit Zeichen auffüllen

Während `\hfill` leere Zwischenräume erzeugt, füllen die Befehle `\hrulefill` und `\dotfill` Zwischenräume mit einer durchgezogenen bzw. mit einer gepunkteten Linie.

Ein Aufruf des Befehls `\hrulefill` füllt den gesamten Raum von der Stelle, an der der Befehl aufgerufen wurde, bis zum Ende der aktuellen Zeile mit einem tiefen Strich.

Das sieht so aus:_____

```
\hrulefill
```

Die Arbeitsweise des Befehls `\dotfill` ist identisch zu `\hrulefill`, allerdings füllt `\dotfill` den Raum bis zum Ende der aktuellen Zeile mit einer gepunkteten Linie.

Das sieht so aus:.....

```
\dotfill
```

Mit diesen einfachen Möglichkeiten, durchgezogene und gepunktete Linien zu erzeugen und Abstände zu beeinflussen, ist es zum Beispiel möglich, Formulare zum Ausfüllen oder Umfragebögen zu erstellen. Der Quelltext von Listing 4.1 zeigt exemplarisch einen Ausschnitt aus einem Formular mit typischen Feldern. Das Ergebnis des Beispiels zeigt Abbildung 4.2.

```

1 \documentclass[a4paper,12pt]{scrartcl}
2 \usepackage{ngerman}
3 \usepackage{pifont}
4 \usepackage[T1]{fontenc}
5 \usepackage[utf8]{inputenc}
6 \setlength{\parindent}{0pt}
7 \setlength{\parskip}{5.0ex plus 0.5ex minus 0.5ex}
8 \thispagestyle{empty}
9
10 \begin{document}
11 \centerline{\textbf{\LARGE Anmeldung}}
12
13 \ding{111} Ich bestätige die Richtigkeit meiner Angaben.
14
15 Nachname: \dotfill\quad Vorname: \dotfill
16
17 \ding{37} Telefon: \hrulefill /\hrulefill\hrulefill\quad
18 \ding{41} E-Mail: \hrulefill\hrulefill\hrulefill
19
20 Datum: \dotfill\quad Unterschrift: \dotfill
21 \end{document}

```

Listing 4.1: Eine sinnvolles Beispiel zu `\dotfill` und `\hrulefill`

Anmeldung

☐ Ich bestätige die Richtigkeit meiner Angaben.

Nachname: Vorname:

☎ Telefon: _____ ☒ E-Mail: _____

Datum: Unterschrift:

Abbildung 4.2: Resultat von Listing 4.1

4.1.8 Grenzen des Textfeldes einhalten

Je nach dem Layout des Dokuments und dem zu setzenden Text kann es passieren, dass der \LaTeX -Compiler einzelne Wörter über das Textfeld herausragen lässt oder Wortabstände sehr groß erscheinen. In diesem Zusammenhang sind auch Warnungen des \LaTeX -Compilers wie `Overfull \hbox...` und `Underfull \hbox` zu verstehen.

Diese Warnungen können auch im Zusammenhang mit übergroßen oder zu geringen Abständen zwischen Absätzen aufkommen. In diesem Fall enthält die Warnung nicht ein `\hbox`, sondern ein `\vbox`.

Ein Überschreiten des Textfeldes verhindert ein Befehl `\sloppy` in der Präambel des Dokuments.

```
\sloppy
```

Dieser Befehl sorgt dafür, das L^AT_EX bereit ist, Wortabstände wenn nötig weiter zu dehnen als das üblich der Fall ist.

Dem Autor dieser Zeilen ist bewusst, das der Befehl `\sloppy` nicht nur Fans hat, sondern auch an einigen Stellen von seiner Verwendung abgeraten wird [32]. Diese Ablehnung teilt der Autor dieses Werkes aber explizit nicht.

4.1.9 Trennungsregeln

Bei Verwendung der Erweiterungspakete `german` oder `ngerman` hält sich der L^AT_EX-Compiler an die deutschen Trennungsregeln. Gerade bei Fremdwörtern kann es jedoch vorkommen, das Autoren aktiv in die Worttrennung eingreifen müssen.

Eine Möglichkeit ist es, in betreffenden Wörtern mit Hilfe von `\-` manuell Trennungshilfen dort zu definieren, wo Sie ein Wort getrennt sehen möchten. In der Praxis kann so etwas wie folgt aussehen:

```
... kommt es zu Durch\-\schnitts\-\verschiebungen ...
```

Diese Trennungshilfen werden nur dann an der konkreten Stelle vom L^AT_EX-Compiler berücksichtigt, wenn das Wort am Ende einer Zeile steht und getrennt werden muss. Ansonsten sind im fertigen Dokument die im Quelltext eingefügten Trennungshilfen zu keiner Zeit erkennbar.

Kommen ein oder mehrere Wörter mit aus Sicht des L^AT_EX-Compilers unklarer Worttrennung mehrfach in einem Dokument vor, ist es sinnvoll in der Präambel des Dokuments mit dem Befehl `\hyphenation` Trennungsregeln für diese Wörter für das gesamte Dokument zu definieren.

```
\hyphenation{Wort_1 Wort_2 ... Wort_n}
```

Die Wörter sind durch Leerzeichen voneinander abgegrenzt. Um die Lesbarkeit des Quelltextes an dieser Stelle zu verbessern, kann auch jedes Wort mit einer neuen Zeile beginnen. In den Wörtern sind die Stellen, an denen Sie dem L^AT_EX-Compiler Worttrennungen erlauben, mit Bindestrichen markiert. Ein Beispiel könnte wie folgt aussehen:

```
\hyphenation{
DHCP-Dis-cover
Durch-schnitts-ver-schiebung
Pegel-wechsel
Signal-pegel
Maxi-mum
Topo-lo-gie
}
```

4.2 Absätze zentrieren, nach einer Seite ausrichten oder einrücken

Standardmäßig setzt der L^AT_EX-Compiler Texte im *Blocksatz*. Diese Layoutform ist die am besten lesbare und optisch ansprechendste. Zudem existiert die Möglichkeit, Text linksbündig, rechtsbündig, zentriert oder beidseitig eingerückt zu setzen. Text, der nur nach einer Seite bündig ist, heißt auch *Flattersatz*.

Linksbündiger Text ist ähnlich gut lesbar wie Blocksatz, aber optisch weniger ansprechend. Rechtsbündige Texte sind in einigen Kulturkreisen der Standard, im deutschen Sprachraum aber nur selten anzutreffen. Das zentrieren (also mittiges setzen) von Text ist ein häufig angewandtes Stilmittel, um etwas hervorzuheben (z. B. eine Überschrift).

Zur Realisierung von zentriertem, linksbündigem, rechtsbündigem oder beidseitig eingerücktem Text existieren jeweils Umgebungen und Befehle.

4.2.1 Zentrierter Text

Mit der Umgebung `center` werden beliebige Inhalte (Text, Abbildungen, Tabellen, etc.) zentriert gesetzt. Alternativ existiert der Befehl `\centering`, der den nachfolgenden Text zentriert setzt.

Der nebenstehende zentriert gesetzte Text wurde mit der Umgebung `center` erzeugt.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac. Vestibulum quis pellentesque diam.

Aliquam erat volutpat. Nunc gravida vestibulum lacus eu feugiat. Phasellus sit amet tellus nibh. Phasellus lobortis quam vel enim efficitur vestibulum. Duis eleifend eros a erat vehicula porta. Nam.

```
\begin{center}...\end{center}
\centering
```

4.2.2 Linksbündiger Text

Inhalte linksbündig im Flattersatz setzen die Umgebung `flushleft` und alternativ der Befehl `\raggedleft`.

Der nebenstehende linksbündig gesetzte Text wurde mit der Umgebung `flushleft` erzeugt.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac. Vestibulum quis pellentesque diam. Aliquam erat volutpat. Nunc gravida vestibulum lacus eu feugiat. Phasellus sit amet tellus nibh. Phasellus lobortis quam vel enim efficitur vestibulum. Duis eleifend eros a erat vehicula porta. Nam.

```
\begin{flushleft}...\end{flushleft}
\raggedleft
```

4.2.3 Rechtsbündiger Text

Inhalte rechtsbündig im Flattersatz setzen die Umgebung `flushright` und alternativ der Befehl `\raggedright`.

Der nebenstehende rechtsbündig gesetzte Text wurde mit der Umgebung `flushright` erzeugt.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac. Vestibulum quis pellentesque diam. Aliquam erat volutpat. Nunc gravida vestibulum lacus eu feugiat. Phasellus sit amet tellus nibh. Phasellus lobortis quam vel enim efficitur vestibulum. Duis eleifend eros a erat vehicula porta. Nam.

```
\begin{flushright}...\end{flushright}
\raggedright
```

4.2.4 Beidseitig eingerückter Text

Besonders bei Zitaten ist es optisch ansprechend, wenn der Text beidseitig gleich weit eingerückt ist, um ihn dadurch hervorzuheben. Zu diesem Zweck existierenden die beiden Umgebungen `quote` und alternativ `quotation`.

Der folgende beidseitig gleich weit eingerückt gesetzte Text wurde mit der Umgebung `quote` erzeugt.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

 Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac.

 Vestibulum quis pellentesque diam. Aliquam erat volutpat.

Die Umgebung `quote` fügt automatisch zwischen Absätzen einen vertikalen Abstand ein und rückt die erste Zeile eines jeden Absatzes nicht ein.

```
\begin{quote}...\end{quote}
```

Das Verhalten von `quote` entspricht damit dem Standard im deutschen Sprachraum.

Der folgende beidseitig gleich weit eingerückt gesetzte Text wurde mit der Umgebung `quotation` erzeugt.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

 Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac.

 Vestibulum quis pellentesque diam. Aliquam erat volutpat.

Die Umgebung `quotation` rückt die erste Zeile eines jeden Absatzes ein und verzichtet dafür auf den vertikalen Abstand zwischen Absätzen.

```
\begin{quotation}...\end{quotation}
```

Das Verhalten von `quotation` entspricht damit dem Standard im anglo-amerikanischen Sprachraum.

4.3 Zeilenumbrüche

Ein Zeilenumbruch kann an beliebiger Stelle im Text mit dem Befehl `\` realisiert werden. Zudem ist es möglich, mit dem optionalen Parameter *Abstand* den vertikalen Zwischenraum zur nächsten Zeile anzugeben. Wenn durch den Parameter *Abstand* ein Seitenumbruch zustande kommt, dann wird der Inhalt von *Abstand* vom L^AT_EX-Compiler ignoriert und die nächste Seite beginnt mit der nächsten Textzeile.

```
\[Abstand]
```

Eine Variante des Befehls `\` ist der Befehl `*`. Dieser verhindert, dass nach dem Zeilenumbruch ein Seitenwechsel vor der nächsten Textzeile erscheint.

```
\*[Abstand]
```

Enthält ein Dokument beispielsweise den Befehl `\[1cm]`, dann wird der \LaTeX -Compiler einen Zeilenumbruch mit einem vertikalen Zeilenabstand von 1 cm einfügen. Kommt durch diesen Befehl ein Seitenumbruch zustande, dann wird der \LaTeX -Compiler auf der neuen Seite die dem Befehl vorangegangene Zeile setzen, dann den 1 cm großen Abstand einfügen und danach die nächste Zeile.

Eine weitere Variante ist der Befehl `\newline`, dessen Verhalten mit dem des Befehls `\` identisch ist. Mit `\newline` ist es allerdings nicht möglich, den Zwischenraum zur nächsten Zeile zu definieren.

`\newline`

4.4 Seitenumbrüche

Genau wie beim Zeilenumbruch wird auch der Seitenumbruch vom \LaTeX -Compiler selbst vorgenommen. Er versucht eventuell vorhandene Gleitobjekte wie Bilder und Tabellen an der geeignetsten Position zu setzen, damit es nicht zu den so genannten *Hurenkindern* oder *Schusterjungen* kommt (siehe Abschnitt 3.6.2),

In seltenen Fällen kann es aber sinnvoll sein, einen Seitenumbruch an einer bestimmten Stelle anzuweisen.

Der Befehl `\newpage` beendet bei einem einspaltigen Seitenlayout die aktuelle Seite und bei einem zweispaltigen Layout die aktuelle Spalte.

`\newpage`

Das manuelle Umbrechen der aktuelle Seite bei eine zweispaltigen Seitenlayout (Klassensoption `twocolumn` oder Befehl `\twocolumn`) geschieht mit dem Befehl `\clearpage`.

`\clearpage`

Bei einem doppelseitigem Layout (Klassensoption `twoside`) geschieht das manuelle Umbrechen der aktuelle Seite mit dem Befehl `\cleardoublepage`. Dieser garantiert, dass die nächste Seite, auf der sich Inhalte befinden, eine ungerade (linke) Seite ist.

`\cleardoublepage`

In bestimmten Situationen kann es hilfreich sein, einen Seitenumbruch an einer bestimmten Stelle zu untersagen. Zu diesem Zweck existiert der Befehl `\nopagebreak`.

`\nopagebreak`

5 Texthervorhebungen

Texthervorhebungen ermöglichen es, bestimmte Textstellen von anderen abzugrenzen oder hervorzuheben. Konkret beschreibt dieses Kapitel unter anderem, Möglichkeiten um Schriftarten und Schriftgröße zu verändern, Texte zu unterstreichen und die Lesbarkeit von Texten mit Aufzählungen zu verbessern.

5.1 Texte hervorheben

Das *Hervorheben* kurzer Textstellen (einzelner Wörter oder Sätze) geschieht mit dem Befehl `\emph` oder alternativ mit der Umgebung `em`. Beide heben den betreffenden Text durch einen Wechsel des Schriftschnitts hervor.

```
\emph{Text}  
\begin{em} Text \end{em}
```

Bei einer senkrechten Grundschrift stellt `\emph` den *hervorzuhebenden Text* mit einer *kursiven Schrift* dar. Bei einer kursiven Grundschriftart ist es genau umgekehrt. Hier wird der hervorgehobene Text mit einer senkrechten Schriftart dargestellt.

Der Wechsel des Schriftschnitts sieht weniger aufdringlich aus als das klassische Unterstreichen mit dem Befehl `\underline`.

```
\underline{Text}
```

Ein Nachteil des Befehls `\underline` ist, dass er keinen Zeilenumbruch zulässt, weil er den kompletten Inhalt des Arguments *Text* in einer horizontalen Box unterbringt. Darum eignet sich `\underline` in erster Linie zur Hervorhebung einzelner Zeichen oder Wörter.

Weitere Befehle, um Text hervorzuheben, enthält das Erweiterungspaket `ulem.sty`. Wird dieses in der Präambel eines Dokuments mit dem Befehl `\usepackage{ulem}` eingebunden, steht eine Reihe weiterer Befehle zur Verfügung.

Dieses sind die Befehle `\uline` zum Unterstreichen, `\uuline` zum Doppeltunterstreichen, `\uwave` zum Unterstreichen mit Wellen, `\sout` zum

Durchstreichen, `\xout` zum Ausstreichen, `\dashuline` zum Unterstreichen mit kurzen Strichen und `\dotuline` zum Unterstreichen mit einer Linie aus Punkten

```
\uline{Text}
\uuline{Text}
\uwave{Text}
\sout{Text}
\xout{Text}
\dashuline{Text}
\dotuline{Text}
```

Da diese Befehle aus dem Erweiterungspaket `ulem.sty` jedes Wort in einer eigenen Box verpacken, sind innerhalb des Arguments *Text* Zeilenumbrüche möglich.

Eine Eigenheit des Erweiterungspakets `ulem.sty` ist, dass es das Verhalten des Befehls `\emph` dahingehend ändert, dass der L^AT_EX-Compiler mit `\emph` betonte Wörter nicht mehr kursiv, sondern unterstrichen setzt. Mit dem Befehl `\normalem` nach dem `\begin{document}` kann das *normale* Verhalten des Befehls `\emph` wiederhergestellt werden.

5.2 Schriftgrößen

Zur Definition der Schriftgröße existiert eine Reihe von Befehlen (siehe Tabelle 5.1) Diese Befehle gelten ab der Position, wo sie im Quelltext aufgerufen werden. Die so ausgewählte Schriftgröße bleibt so lange aktuell, bis das Ende der Gruppe oder das Ende des Dokuments erreicht ist oder bis durch einen Befehl eine neue Schriftgröße festlegt wird.

5.3 Schriftfamilien und Schriftschnitte

Tabelle 5.2 zeigt Befehle zur Definition der Schrift, die den betreffenden Text als Argument übergeben bekommen.

Während die Befehle aus Tabelle 5.2 mit dem Text als Argument sich eher für kürzere Texte eignen, sind die Deklarationsbefehle aus Tabelle 5.3 zum Umschalten der Schrift für längere Texte besser geeignet.

Die Befehle in Tabelle 5.3 schalten die Schrift an der Stelle ihres Auftretens im L^AT_EX-Quelltext um. Die Wirkung dieser Befehle ist durch die aktuelle Gruppe begrenzt. Ein

Tabelle 5.1: Befehle zur Definition der Schriftgröße

Befehl	Beispiel
<code>\tiny</code>	ABCDEFGH0123456789
<code>\scriptsize</code>	ABCDEFGH0123456789
<code>\footnotesize</code>	ABCDEFGH0123456789
<code>\small</code>	ABCDEFGH0123456789
<code>\normalsize</code>	ABCDEFGH0123456789
<code>\large</code>	ABCDEFGH0123456789
<code>\Large</code>	ABCDEFGH0123456789
<code>\LARGE</code>	ABCDEFGH0123456789
<code>\huge</code>	ABCDEFGH0123456789
<code>\Huge</code>	ABCDEFGH0123456789

Tabelle 5.2: Befehle zur Definition der Schrift mit Text als Argument

Befehl	Beschreibung
<code>\textrm{Text}</code>	schaltet auf die Schriftart Roman
<code>\texttt{Text}</code>	schaltet auf eine Schreibmaschinenschrift (Typewriter)
<code>\textsf{Text}</code>	schaltet auf die Sans-Serif-Schriftfamilie
<code>\textsc{Text}</code>	SCHALTET AUF KAPITÄLCHEN-SCHRIFT (SMALL CAPS)
<code>\textsl{Text}</code>	<i>schaltet auf geneigte Roman-Schrift (Slanted)</i>
<code>\textit{Text}</code>	<i>schaltet auf die Schrift Italic</i>
<code>\textup{Text}</code>	schaltet auf senkrechten Schriftschnitt
<code>\textmd{Text}</code>	schaltet auf normale Breite und Strichstärke (normale Schrift)
<code>\textbf{Text}</code>	schaltet auf fette Schrift

Aufruf eines dieser Befehle in der Präambel eines Dokuments beeinflusst das gesamte Dokument.

Nach eine Änderung der Schrift kann mit dem Befehl `\textnormal` jederzeit auf die alte Schrift zurückgegriffen werden, die zu Beginn des Dokuments gültig war.

`\textnormal{Text}`

Die Umgebungen in Tabelle 5.3 eignen sich besonders dann, wenn für einzelne Abschnitte eines Dokuments die Schrift angepasst werden soll.

Tabelle 5.3: Befehle zur Definition der Schrift und die entsprechenden Umgebungen

Befehl	Umgebung			Beispiel
<code>\rmfamily</code>	<code>\begin{rmfamily}</code>	<i>Text</i>	<code>\end{rmfamily}</code>	Roman
<code>\ttfamily</code>	<code>\begin{ttfamily}</code>	<i>Text</i>	<code>\end{ttfamily}</code>	Typewriter
<code>\sffamily</code>	<code>\begin{sffamily}</code>	<i>Text</i>	<code>\end{sffamily}</code>	Sans Serif
<code>\scshape</code>	<code>\begin{scshape}</code>	<i>Text</i>	<code>\end{scshape}</code>	SMALL CAPS
<code>\slshape</code>	<code>\begin{slshape}</code>	<i>Text</i>	<code>\end{slshape}</code>	<i>Slanted</i>
<code>\itshape</code>	<code>\begin{itshape}</code>	<i>Text</i>	<code>\end{itshape}</code>	<i>Italic</i>
<code>\upshape</code>	<code>\begin{upshape}</code>	<i>Text</i>	<code>\end{upshape}</code>	Senkrechte Schrift
<code>\mdseries</code>	<code>\begin{mdseries}</code>	<i>Text</i>	<code>\end{mdseries}</code>	Normale Schrift
<code>\bfseries</code>	<code>\begin{bfseries}</code>	<i>Text</i>	<code>\end{bfseries}</code>	Fette Schrift

5.4 Aufzählungen

Das Setzen von Aufzählungen geschieht mit den Umgebungen `itemize`, `enumerate` und `description`. Alle diese Umgebungen rücken den aufgezählten Text ein wenig ein und versehen ihn am Anfang mit einer Markierung (einem Zeichen).

```

\begin{itemize} ... \end{itemize}
\begin{enumerate} ... \end{enumerate}
\begin{description} ... \end{description}

```

Während `itemize` jedes neue Element der Aufzählung mit einem schwarzen, ausgefüllten Punkt • kennzeichnet, nummeriert `enumerate` die Elemente durch. `description` hingegen hebt einen vom Autor festzulegenden Text am Anfang des Elements hervor.

Bei allen diesen drei Umgebungen wird jeder neue Element der Aufzählung durch den Befehl `\item` gekennzeichnet. Der Text jedes Elements – man könnte auch Aufzählungspunktes sagen – darf beliebig lang sein, und kann auch aus mehreren Absätzen bestehen.

Alle in diesem Abschnitt beschriebenen Umgebungen können verschachtelt werden. Mehr als vier Ebenen sind aber nicht zulässig.

Bei der Umgebung `itemize` ist für jede Ebene ein Markierungszeichen voreingestellt. Die folgende Übersicht zeigt, welche das im Einzelnen sind.

```

\begin{itemize}
\item In der ersten Ebene...
\begin{itemize}
\item In der zweiten Ebene
ist es ein...
\begin{itemize}
\item In der dritten Ebene
ist das Markierungszeichen...
\begin{itemize}
\item In der vierten Ebene
ist es ein...
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}

```

- In der ersten Ebene von `itemize` ist das Markierungszeichen ein sogenannter `\textbullet`.
 - In der zweiten Ebene ist es ein fett geschriebener Trennstrich `{\normalfont\bfseries \textendash}`.
 - * In der dritten Ebene ist das Markierungszeichen ein `\textasteriskcentered`.
 - In der vierten Ebene ist es ein `\textperiodcentered`.

Aufzählungen realisiert die Umgebung `enumerate`. Auch bei `enumerate` ist für jede Ebene eine andere Art von Markierungszeichen voreingestellt und die Nummerierung der Aufzählungspunkte startet in jeder Ebene neu beim Wert 1.

```

\begin{enumerate}
\item In der ersten Ebene...
\item arabischen Ziffern...
\begin{enumerate}
\item Die zweite Ebene...
\item Buchstaben und...
\begin{enumerate}
\item Die dritte Ebene...
\item römische Ziffern.
\begin{enumerate}
\item Die vierte Ebene nutzt
großen Buchstaben.
\end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}

```

1. In der ersten Ebene wird mit
2. arabischen Ziffern durchnummeriert.
 - a) Die zweite Ebene verwendet kleine
 - b) Buchstaben und schließende Klammern.
 - i. Die dritte Ebene verwendet
 - ii. römische Ziffern.
 - A. Die vierte Ebene nutzt
 - B. großen Buchstaben.
 - iii. Beim Verschachteln gehen die Zählerstände nicht verloren.

Die Umgebung `description` ist ideal um Begriffe (Schlagwort) zu beschreiben oder Teilnehmerlisten zu realisieren. Diese Umgebung unterscheidet sich in einigen Punkten

von `itemize` und `enumerate`. Bei `description` wird zwischen dem Befehl `\item` und der Beschreibung das Schlagwort in eckigen Klammern angegeben. Die Ausgabe des Schlagworts erfolgt im Fettdruck.

```
\begin{description}
\item[Bit] Kleinstmögliche...
\item[Byte] Gruppe von 8...
\item[Nibble] Gruppe von 4...
\item[Oktett] siehe Byte...
\item[Unicode] Mehrbyte....
\end{description}
```

Bit Kleinstmögliche Informationseinheit. Zwei mögliche Zustände

Byte Gruppe von 8 Bits

Nibble Gruppe von 4 Bits bzw. ein Halbbyte

Oktett siehe Byte

Unicode Mehrbyte-Zeichenkodierung

5.5 Fußnoten

Das Erzeugen von Fußnoten geschieht mit dem Befehl `\footnote`.

```
\footnote{Text in der Fußnote}
```

Der Befehl `\footnote` folgt im Quelltext immer direkt – also ohne Leerzeichen – nach dem Wort, an das die Markierung der Fußnote angehängt sein soll. Der Text der Fußnote wird vom \LaTeX -Compiler in einer kleineren Schrift von der Größe `footnotesize` geschrieben. Standardmäßig rückt der \LaTeX -Compiler die erste Zeile einer Fußnote einen halben Zentimeter ein.

Zwischen den Fußnoten einer Seite und dem gewöhnlichen Text des Dokuments fügt der \LaTeX -Compiler automatisch eine horizontale Linie ein, um die Fußnoten deutlich vom Text abzuheben.

Der Befehl `\footnote` darf nicht innerhalb mathematischer Umgebungen oder Tabellen aufgerufen werden. Es gibt aber verschiedene Tricks, um sich in einem solchen Fall zu behelfen. Eine Möglichkeit, um Fußnoten in von Tabellen zu realisieren, sind die Befehle `\footnotemark` zum Erzeugen einer einzelnen Fußnotenmarkierung und `\footnotetext` zum Erzeugen eines einzelnen Fußnotentextes für eine bestimmte Fußnotenmarkierung.

5.6 Marginalien

Randbemerkungen, die auch *Marginalien* heißen, sind ein mögliches Stilmittel für Ergänzungen und Erläuterungen.

```
\marginpar[Text einer linken Randnotiz]{Text einer rechten Randnotiz}
```

Bei einem Text mit zweispaltigem Layout wird die Randbemerkung an den am nächsten liegenden Rand gesetzt. Mit Hilfe der drei Befehle `\marginparwidth`, `\marginparsep` und `\marginparpush` ist es möglich, das Aussehen der Marginalien zu beeinflussen. Jeder der drei Befehle erfordert die Angabe einer Breite bzw. eines Abstands inklusive einer Maßeinheit (siehe Abschnitt 2.4) in geschweiften Klammern.

Der Befehl `\marginparwidth` definiert die Breite des Randnotizen-Bereichs fest.

```
\setlength{\marginparwidth}{Breite}
```

Der Befehl `\marginparsep` definiert den Abstand zwischen Textfeld und Randnotizen.

```
\setlength{\marginparsep}{Abstand}
```

Der Befehl `\marginparpush` definiert den Abstand zwischen zwei Randnotizen.

```
\setlength{\marginparpush}{Abstand}
```

Die erste Zeile des Textes der Marginalie wird von \LaTeX in der gleichen Zeile gesetzt, in der sich der Befehl `\marginpar` befindet.

5.7 Unformatierter Text

Die Möglichkeit, unformatierten Text in einem Dokument einzufügen, ist besonders für Dokumentationen und wissenschaftliche Publikationen im Bereich der Informatik hilfreich, denn Programmcode wird auf diese Art und Weise vom übrigen Text abgehoben. Auch in diesem Dokument sind Befehle und Umgebungen auf diese Art und Weise dargestellt.

Das Setzen von unformatiertem Text kann u.a. mit der Umgebung `verbatim` geschehen.

```
\begin{verbatim}Text\end{verbatim}
```

Text, der sich innerhalb dieser Umgebung befindet, wird vom \LaTeX -Compiler in `TypeWriter`, also `Schreibmaschinenschrift` gesetzt und nicht verändert oder interpretiert. Dieser Text kann somit alle Sonderzeichen von \LaTeX und beliebige Kombinationen von Leerzeichen und Zeilenumbrüchen enthalten.

Zu Beginn und Ende der Umgebung `verbatim` fügt der \LaTeX -Compiler einen vertikalen Leerraum ein und es wird zu Anfang und Ende der Umgebung eine neue Zeile begonnen.

Eine weitere Möglichkeit, unformatierten, maximal eine Zeile langen Text auszugeben ist der Befehl `\verb`. Direkt im Anschluss an den Befehl muss ein zur Abgrenzung

verwendetes Zeichen folgen. Dieses Begrenzungszeichen muss den unformatiert auszugebenden Text auch abschließen. In der folgenden Darstellung der Syntax von `\verb` dient das Zeichen `+` als Begrenzungszeichen.

```
\verb+Text+
```

Anstelle des Zeichens `+` könnte auch das Zeichen `!`, `?`, `#` oder `|` oder sonst (fast) ein beliebiges Zeichen als Begrenzungszeichen verwendet werden. Das Begrenzungszeichen darf sich aber nicht im darzustellenden Text befinden und die Begrenzungszeichen eines `\verb`-Befehls müssen sich in einer Zeile im Quelltext befinden.

Weder die Umgebung `verbatim`, noch Varianten davon wie der Befehl `\verb` dürfen in Argumenten von Befehlen oder innerhalb von Tabellen verwendet werden.

5.8 Internetadressen

Zum Satz von Internetadressen (URL) existiert das Erweiterungspaket `url`. Wird dieses mit dem Befehl `\usepackage{url}` in der Präambel der Quelldatei eingebunden, steht der gleichnamige Befehl `\url` zur Verfügung. Diesem wird die zu setzende Internetadresse in geschweiften Klammern als Parameter übergeben.

```
\url{Internatadresse}
```

Die Adresse setzt der \LaTeX -Compiler dann in einer Schrift mit fester Zeichenbreite und bricht die Adressen bei Bedarf optisch ansehnlich um. Zudem sind mit dem Befehl `\url` realisierte Internetadressen in der resultierenden PDF-Datei als Link nutzbar.

5.9 Boxen um Text und Bilder zeichnen

Boxen sind eine einfache Möglichkeit, um Textabschnitte oder Bilder hervorzuheben. Ein Beispiel für einen Befehl, der es ermöglicht, einen Kasten um einen zu umrahmenden Text zu setzen, ist `\fbox`.

```
\fbox{Text}
```

Ein Nachteil dieses Befehls ist, dass damit weder die Größe der Box, noch die Ausrichtung des Textes beeinflusst werden können.

Ausgefallenere Boxentypen bietet das Erweiterungspaket `fancybox`. Wird es mit dem Befehl `\usepackage{fancybox}` in der Präambel der `.tex`-Quelldatei eingebunden, stehen die Befehle `\doublebox`, `\ovalbox`, `\ovalbox` und `\shadowbox` zur Verfügung.

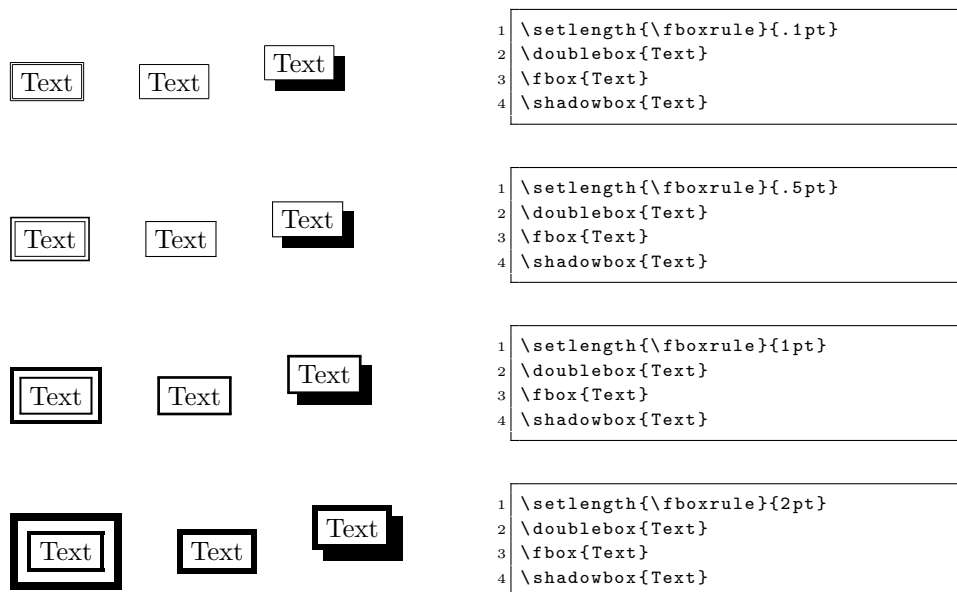


```
\doublebox{Text}
\ovalbox{Text}
\Ovalbox{Text}
\shadowbox{Text}
```

Zur manuellen Definition der Linienstärke bei den Boxentypen `doublebox`, `fbox` und `shadowbox` wird dem Längenbefehl `\fboxrule` mit dem Befehl `\setlength` ein Wert zugewiesen, der zum Standardwert unterschiedlich ist. Der Standardwert ist in der Datei `latex.ltx` definiert und hat den Wert `.4pt`.

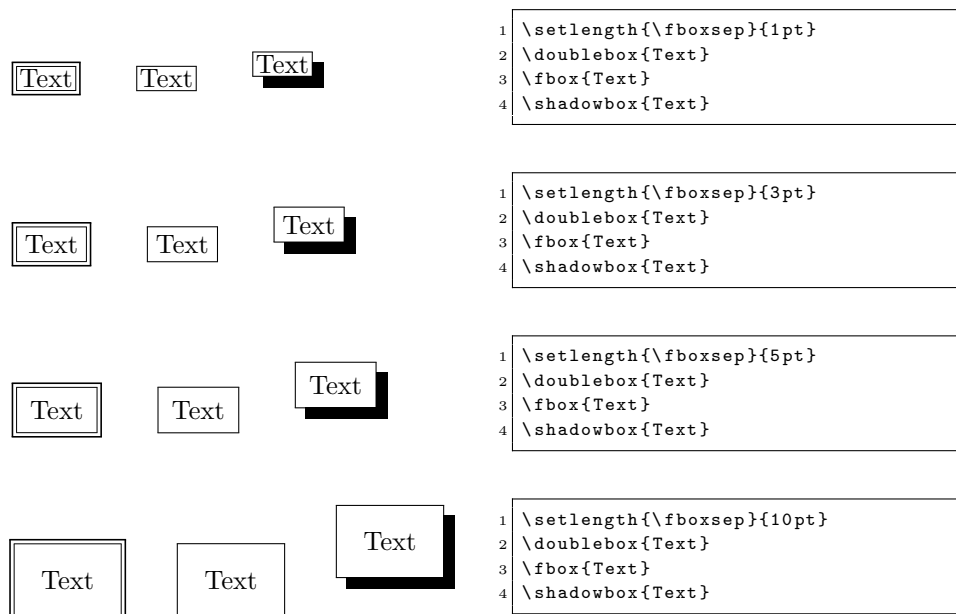
Bei einer `doublebox` ist die Linienstärke des äußeren Rahmens `1.5\fboxrule`, also das Anderthalbfache von `\fboxrule`, und die Linienstärke des inneren Rahmens ist `.75\fboxrule`.

Die folgenden Beispiele zeigen die Auswirkungen der Änderung des Längenbefehls `\fboxrule`:



Zur manuellen Definition des Abstands zwischen Inhalt und Rahmen bei den gezeigten Boxentypen wird dem Längenbefehl `\fboxsep` mit dem Befehl `\setlength` ein Wert zugewiesen, der zum Standardwert unterschiedlich ist. Der Standardwert ist in der Datei `latex.ltx` definiert und hat den Wert `3pt`.

Die folgenden Beispiele zeigen die Auswirkungen der Änderung des Längenbefehts `\fboxsep`:



Einige weitere Längenbefehle enthält Tabelle 5.4.

Tabelle 5.4: Längenbefehle für Boxen

Längenbefehl	Bedeutung	Standardwert
<code>\fboxrule</code>	Definiert die Linienstärke bei den Boxentypen <code>\doublebox</code> , <code>\fbox</code> und <code>\shadowbox</code>	.4pt
<code>\fboxsep</code>	Definiert den Abstand zwischen Inhalt und Rahmen	3tp
<code>\shadowsize</code>	Definiert die Breite des Schattens bei <code>\shadowbox</code>	4tp
<code>\thinlines</code>	Definiert die Linienstärke bei <code>\ovalbox</code>	.4pt
<code>\thicklines</code>	Definiert die Linienstärke bei <code>\Ovalbox</code>	.8pt

Eine komfortable Möglichkeit, mehrzeilige Inhalte mit einem Rahmen zu versehen, bietet das Erweiterungspaket `boxedminipage2e`. Wird es mit dem Befehl `\usepackage{boxedminipage2e}` in der Präambel der `.tex`-Quelldatei eingebunden, steht die Umgebung `boxedminipage` zur Verfügung. Mit dieser ist es möglich, `minipage`-Umgebungen zu erzeugen, die von einem Rahmen umgeben sind (siehe Abbildung 5.1).

```
\begin{boxedminipage}[Ausrichtung][Höhe][Position]{Breite}
Inhalt
\end{boxedminipage}
```

Die Argumente der Umgebung `boxedminipage` orientieren sich an denen der Umgebung `minipage`.

Das erste optionale Argument **Ausrichtung** definiert die Ausrichtung der Umgebung anhand der aktuellen Zeile. Es richtet die Minipage relativ zur aktuellen Grundlinie aus. Mögliche Werte sind **t** (top=oben), wobei die Minipage relativ zur aktuellen Grundlinie ausgerichtet wird, **c** (center=zentriert), wobei die Mitte der Minipage eine Linie mit der aktuellen Grundlinie bildet und **b** (bottom=unten), wobei die unterste Grundlinie innerhalb der Minipage eine Linie mit der aktuellen Grundlinie bildet. Der Standardwert ist **c**. [33]

Das optionale Argument **Höhe** definiert die Höhe als Absolutwert inklusive einer Maßeinheit (z.B. 5cm) als relativen Wert (z.B. .5\textheight). Wird der Wert dieses Arguments definiert, spielt die tatsächliche Höhe des Inhalts der Minipage für deren Dimensionierung keine Rolle.

Das optionale Argument **Position** definiert die Ausrichtung des Inhaltes. Mögliche Werte sind auch hier: **t**, **c** oder **b** und auch hier ist **c** der Standardwert.

Das einzige zwingend nötige Argument **Breite** definiert die Breite der Minipage. Auch dabei kann es sich um einen Absolutwert inklusive einer Maßeinheit oder einen relativen Wert handeln. Der L^AT_EX-Compiler bricht den Inhalt der Minipage entsprechend deren Breite um.

Sind nur zwei optionale Argumente angegeben, so sind die Werte von **Ausrichtung** und **Position** identisch. Ist nur ein optionales Argument angegeben, so entspricht der Wert von **Höhe** der Gesamthöhe des Inhalts der Minipage.

Das ist eine gewöhnliche `minipage`-Umgebung mit einem Rahmen, die mit der Umgebung `boxedminipage` aus dem Erweiterungspaket `boxedminipage2e` erzeugt wurde.

```
1 \begin{center}
2 \begin{boxedminipage}{6cm}
3 Das ist eine
4 ...
5 erzeugt wurde.
6 \end{boxedminipage}
7 \end{center}
```

Abbildung 5.1: Minipages mit Rahmen realisiert die Umgebung `boxedminipage`

6 Tabellen

L^AT_EX bietet zahlreiche Umgebungen zum Satz von Tabellen. Aus Platzgründen kann dieses Dokument aber nur eine Auswahl, nämlich die Umgebungen `tabular` und `tabularx` berücksichtigen.

6.1 Die Umgebung `tabular`

Die Umgebung `tabular` ist die am einfachsten zu verwendende Tabellen-Umgebung von L^AT_EX und gleichzeitig auch diejenige mit den wenigsten Layout-Möglichkeiten.

```
\begin{tabular}[Position]{Spalten}Tabelleninhalt\end{tabular}
```

In der ersten Zeile der Tabellendefinition, der sogenannten Tabellenpräambel (das ist die Zeile mit dem `\begin{tabular}`) ist die Anzahl der Spalten (*Spalten*) und die Ausrichtung der Tabelle (*Position*) definiert. Die Ausrichtung ist allerdings ein optionaler Parameter.

der Parameter *Spalten* definiert nicht nur die Anzahl der Spalten, sondern auch die Ausrichtung des Inhalts in den Spalten (siehe Tabelle 6.1). Hier ist festgelegt, ob der Inhalt in den Zellen der Tabelle linksbündig, rechtsbündig oder zentriert ausgerichtet ist. Zudem wird innerhalb dieses Parameters definiert, ob und wie viele vertikale Linien es in der Tabelle gibt, um die einzelnen Spalten voneinander abzugrenzen.

Der Parameter *Position* kann die Werte `t` oder `b` enthalten und definiert die vertikale Ausrichtung der Tabelle. Beim Wert `t` wird die oberste Zeile der Tabelle an der laufenden Umgebung ausgerichtet. Beim Wert `b` wird die Tabelle mit der untersten Zeile an der laufenden Umgebung ausgerichtet. Wird auf den Parameter *Position* verzichtet, dann wird die Tabelle in ihrer vertikalen Mitte auf die laufende Umgebung ausgerichtet.

Der Inhalt des Parameters *Spalten* hängt von folgenden Fragestellungen ab:

1. Wie viele Spalten enthält die Tabelle?
2. Wie soll der Inhalt in jeder Spalte ausgerichtet sein (links-, rechtsbündig oder zentriert)?

3. Sollen die Spalten durch vertikale Linien kenntlich gemacht sein, und wenn ja, welche Spalten und mit wie vielen Linien (eine oder zwei Linien)?

Für jede Spalte, deren Inhalt linksbündig ausgerichtet sein soll, enthält der Parameter *Spalten* ein mal den Buchstaben *l*. Für jede Spalte, deren Inhalt rechtsbündig ausgerichtet sein soll, wird ein *r* eingefügt, und für eine Spalte mit zentriertem Inhalt enthält der Parameter den Buchstaben *c*.

Es ist auch möglich, eine linksbündig formatierte Spalte mit einer definierten Breite zu erstellen. Dieses geschieht mit einem Eintrag *p{Breite}*. Eine Besonderheit dieses Spaltenformatierungseintrags ist, dass er die Realisierung mehrzeiliger Spalteneinträge ermöglicht. Übersteigt der Inhalt eines Feldes die definierte Spaltenbreite, wird dieser in mehrere Zeilen umbrochen. Mit den Spaltenformatierungseinträgen *l*, *r* und *c* sind nur einzeilige Felder möglich.

Ein weiterer Spaltenformatierungseintrag, ist **{Anzahl}{Spaltenform}*. Von der Stelle des Aufrufs im Parameter *Spalten* wird die in *Spaltenform* festgelegte Spaltenformatierung *Anzahl*-mal wiederholt.

Soll ein bestimmter Text oder ein sonstiger Inhalt in jeder Zeile der Tabelle zwischen zwei bestimmten Spalten erscheinen, kann dieses mit dem Spaltenformatierungseintrag *!{Text}* im Parameter *Spalten* angewiesen werden.

Tabelle 6.1: Mögliche Einträge im Parameter *Spalten*

Wert	Bedeutung
<i>l</i>	Der Inhalt der Spalte wird linksbündig gesetzt.
<i>r</i>	Der Inhalt der Spalte wird rechtsbündig gesetzt.
<i>c</i>	Der Inhalt der Spalte wird zentriert gesetzt.
<i>p{Breite}</i>	Die Spalte wird mit der Breite Breite gesetzt und der Inhalt wird, wenn er die definierte Spaltenbreite übersteigt, in mehrere Zeilen gebrochen.
<i>*{Anzahl}{Spaltenform}</i>	Die in <i>Spaltenform</i> stehende Spaltenformatierung wird <i>Anzahl</i> -mal wiederholt.
<i> </i>	Es wird ein senkrechter Strich gesetzt.
<i> </i>	Es wird ein senkrechter Doppelstrich gesetzt.
<i>!{Text}</i>	Der <i>Text</i> wird in jeder Zeile zwischen die beiden Spalten links und rechts davon gesetzt.

Senkrechte Striche oder Doppelstriche zum Kennzeichnen der Grenzen von Tabellen und/oder Spalten erzeugt man mit Hilfe von einzelnen oder doppelten senkrechten Strichen im Parameter *Spalten* (siehe Tabelle 6.1).

Der Befehl `\\` markiert das Ende jeder Zeile in einer Tabelle. Zwischen zwei Spalten befindet sich immer das Zeichen `&`. Auf diese Art und Weise sind die einzelnen Zellen der Tabellen voneinander abgegrenzt.

Tabelle 6.2: Eine einfache Tabelle

Linksbündig	Zentriert	Rechtsbündig
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

Tabelle 6.2 ist ein einfaches Beispiel für eine Tabelle mit drei Spalten. Der Quelltext von Tabelle 6.2 befindet sich in Listing 6.1. In der Spaltendeklaration der Tabellenpräambel ist definiert, dass die erste Spalte linksbündig (`l`), die zweite Spalte zentriert (`c`) und die dritte Spalte rechtsbündig (`r`) ist.

```

1 \begin{tabular}{lcr}
2 \textbf{Linksbündig} & \textbf{Zentriert} & \textbf{Rechtsbündig} \\
3 x1y1 & x2y1 & x3y1 \\
4 x1y2 & x2y2 & x3y2 \\
5 x1y3 & x2y3 & x3y3 \\
6 \end{tabular}

```

Listing 6.1: Eine einfache Tabelle

Zur besseren Übersichtlichkeit ist es in vielen Publikationen üblich die einzelnen Zellen in Tabellen durch horizontale und vertikale Linien voneinander abzugrenzen. Tabelle 6.3 erweitert das Beispiel aus Tabelle 6.2 um horizontale und vertikale Linien. Im Beispiel sind alle Zellen der Tabelle durch vertikale und horizontale Linien voneinander abgegrenzt. Der Quelltext von Tabelle 6.3 befindet sich in Listing 6.2.

Tabelle 6.3: Vertikale und horizontale Linien trennen die Zellen voneinander ab

Linksbündig	Zentriert	Rechtsbündig
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

```

1 \begin{tabular}{|l|c|r|}
2 \hline
3 \textbf{Linksbündig} & \textbf{Zentriert} & \textbf{Rechtsbündig} \\
4 \hline\hline
5 x1y1 & x2y1 & x3y1 \\
6 x1y2 & x2y2 & x3y2 \\
7 x1y3 & x2y3 & x3y3 \\
\end{tabular}

```

```

8 \hline
9 \end{tabular}

```

Listing 6.2: Vertikale und horizontale Linien grenzen die Zellen voneinander ab

In der Spaltendeklaration in Listing 6.2 sind die vertikalen Linien definiert. Horizontalen Linien werden mit Hilfe des Befehls `\hline` erzeugt.

Optisch eleganter ist es in vielen Fällen, auf vertikale Trennlinien zu verzichten und nur zur Begrenzung der eigentlichen Tabelle, sowie des Tabellenkopfs, horizontale Trennlinien zu ziehen. Ein Beispiel für eine solche Tabelle ist Tabelle 6.4. Der zugehörige Quelltext ist Listing 6.3.

Tabelle 6.4: Ohne vertikale Trennlinien wird die Tabelle optisch *leichter*

Linksbündig	Zentriert	Rechtsbündig
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

```

1 \begin{tabular}{lcr}
2 \hline
3 \textbf{Linksbündig} & \textbf{Zentriert} & \textbf{Rechtsbündig} \\
4 \hline
5 x1y1 & x2y1 & x3y1 \\
6 x1y2 & x2y2 & x3y2 \\
7 x1y3 & x2y3 & x3y3 \\
8 \hline
9 \end{tabular}

```

Listing 6.3: Ohne vertikale Trennlinien wird die Tabelle optisch *leichter*

Je nach gewünschter Struktur der Tabelle, ist es möglich, den Inhalt der Spaltendeklaration mit `*{Anzahl}{Spaltenform}` zu vereinfachen.

Damit wird die in *Spaltenform* definierte Spaltenform *Anzahl*-mal wiederholt. Enthält eine Tabelle beispielsweise in der Spaltendeklaration die Sequenz `|l|l|l|l|`, kann diese durch `*{4}{|l|}` verkürzt werden. Ein weiteres Beispiel ist die Sequenz `|l|r|l|r|l|r|`. Diese kann mit Hilfe des Eintrags `*{3}{|l|r|}` verkürzt werden.

Soll in jeder Zeile zwischen zwei bestimmten Spalten der gleiche Inhalt eingefügt werden, beispielsweise ein Gleichheitszeichen (=) oder ein Pfeil (z.B. \implies), kann dieses einfach mit dem Spaltenformatierungseintrag `!{Text}` angewiesen werden. Ein sinnvolles Beispiel dieses Spaltenformatierungseintrags zeigt Tabelle 6.5 mit einer Übersicht der binomische Formeln. Der zugehörige Quelltext ist Listing 6.4.

Tabelle 6.5: Identischen Inhalt in jeder Zeile zwischen zwei Spalten einfügen

$(a + b)^2 = a^2 + 2ab + b^2$
$(a - b)^2 = a^2 - 2ab + b^2$
$(a + b) * (a - b) = a^2 - b^2$

```
1 \begin{tabular}{r!{=}l}  
2 \hline  
3 \((a+b)^2\) & \((a^2+2ab+b^2)\) \\  
4 \((a-b)^2\) & \((a^2-2ab+b^2)\) \\  
5 \((a+b) * (a-b)\) & \((a^2-b^2)\) \\  
6 \hline  
7 \end{tabular}
```

Listing 6.4: Identischen Inhalt in jeder Zeile zwischen zwei Spalten einfügen

6.2 Die Umgebung tabularx

Außer der Umgebung `tabular` bietet \LaTeX zahlreiche weitere Umgebungen, um Tabellen zu setzen, die je nach konkretem Anwendungsfall hilfreich sein können. Eine dieser Umgebungen ist die Umgebung `tabularx`, denn diese ermöglicht es Tabellen mit definierbarer Breite und automatischer Spaltenbreite zu realisieren.

```
\begin{tabularx}{Breite}[Position]{Spalten} Zeilen \end{tabularx}
```

Um diese Umgebung zu verwenden, muss die Präambel der `.tex`-Quelldatei den Befehl `\usepackage{tabularx}` enthalten, um das Erweiterungspaket `tabularx` einzubinden.

Das Erweiterungspaket definiert u.a. die Spaltenspezifikation `X`. Diese realisiert eine Spalte mit linksbündigem Inhalt, deren Breite der \LaTeX -Compiler automatisch festlegt. Das Beispiel in Tabelle 6.6 zeigt die Funktionsweise. Jede Spalte erhält den gleichen Anteil an der verfügbaren Tabellenbreite. Der zugehörige Quelltext ist Listing 6.5.

Tabelle 6.6: Jede Spalte erhält den gleichen Anteil an der verfügbaren Tabellenbreite

Spalte 1	Spalte 2	Spalte 3
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

```
1 \begin{tabularx}{\textwidth}{|X|X|X|}
```

```

2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 x1y1 & x2y1 & x3y1 \\
6 x1y2 & x2y2 & x3y2 \\
7 x1y3 & x2y3 & x3y3 \\
8 \hline
9 \end{tabularx}

```

Listing 6.5: Eine einfache Tabelle mit der Umgebung `tabularx`

Wie das Beispiel in Tabelle 6.6 zeigt, sorgt die Spaltenspezifikation `X` dafür, dass jede Spalte den gleichen Anteil an der verfügbaren Tabellenbreite bekommt. Die Berechnung der Spaltenbreite erfolgt anhand folgender Formel:

$$\text{Spaltenbreite} = \frac{\text{Gesamtbreite der Tabelle}}{\text{Spaltenanzahl}}$$

Im Beispiel ist die Tabellenbreite identisch mit der Breite des Textfelds im Dokument. Alternativ könnte beispielsweise ein feste Breite unter Angabe eines Werts inklusive einer Maßeinheit (siehe Abschnitt 2.4) definiert sein.

Standardmäßig ermöglicht `tabularx` nur linksbündige Spalten mit automatischer Breite. Sollen auch Spalten mit rechtsbündigem oder zentriertem Inhalt mit automatischer Breite möglich sein, ermöglichen dieses die beiden folgenden Zeilen in der Präambel der `.tex`-Quelldatei.

```
\newcolumntype{Y}{>{\centering\arraybackslash}X}
```

```
\newcolumntype{Z}{>{\hfill\arraybackslash}X}
```

Die beiden obigen Zeilen definieren die beiden Spaltenspezifikationen `Y` für Spalten mit zentriertem Inhalt und `Z` für Spalten mit rechtsbündigem Inhalt. Das Beispiel in Tabelle 6.7 zeigt die praktische Anwendung der neu definierten Spaltenspezifikationen `Y` und `Z`. Der zugehörige Quelltext ist Listing 6.6.

Tabelle 6.7: Spalten mit automatischer Breite bei `tabularx`

Spalte 1	Spalte 2	Spalte 3
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

Für Tabelle 6.7 ist eine Tabellenbreite von 10 cm definiert. Die drei Spalten erhalten automatisch den gleichen Anteil an der verfügbaren Tabellenbreite.

```

1 \newcolumntype{Y}{>\centering\arraybackslash}X}
2 \newcolumntype{Z}{>\hfill\arraybackslash}X}
3
4 ...
5
6 \begin{tabularx}{.8\textwidth}{|X|Y|Z|}
7
8 Spalte 1 & Spalte 2 & Spalte 3 \\
9 \hline\hline
10 x1y1 & x2y1 & x3y1 \\
11 x1y2 & x2y2 & x3y2 \\
12 x1y3 & x2y3 & x3y3 \\
13 \hline
14 \end{tabularx}

```

Listing 6.6: Spalten mit automatischer Breite bei `tabularx`

Es ist auch möglich, in einer Tabelle, die mit `tabularx` gesetzt wird, die Spaltenspezifikationen `l`, `r` und `c` aus der Umgebung `tabular` zu verwenden. In diesem Fall orientiert sich die Spaltenbreite wie gehabt am Inhalt. Enthält die Tabelle auch eine oder mehr Spalten mit automatischer Breite, werden diese entsprechend gestreckt oder gestaucht, damit die im Parameter *Breite* definierte Gesamtbreite der Tabelle eingehalten wird. Tabelle 6.8 und der zugehörige Quelltext in Listing 6.7 zeigen dieses Verhalten.

Tabelle 6.8: Kombination der Spaltenspezifikationen von `tabular` und `tabularx`

Spalte 1	Spalte 2	Spalte 3
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

```

1 \begin{tabularx}{10cm}{|X|c|r|}
2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 x1y1 & x2y1 & x3y1 \\
6 x1y2 & x2y2 & x3y2 \\
7 x1y3 & x2y3 & x3y3 \\
8 \hline
9 \end{tabularx}

```

Listing 6.7: Kombination der Spaltenspezifikationen von `tabular` und `tabularx`

6.3 Farbige Tabellen

Zum Einfärben von Tabellen stehen nach dem Einbinden des Erweiterungspakets `colortbl.sty` zahlreiche Befehle zur Verfügung. Der Import des Erweiterungspakets geschieht mit dem Befehl `\usepackage{colortbl}` in der Präambel der `.tex`-Quelldatei. Zudem müssen noch die Erweiterungspakete `xcolor` und `array` in der Präambel eingebunden sein.

Der Befehl `\rowcolor` ermöglicht das Einfärben einzelner Zeilen.

`\rowcolor[Farbmodell]{Farbe}`

Das optionale Argument `Farbmodell` definiert das gewünschte Farbmodell. Mögliche Angaben sind `cmk`, `HTML`, `gray`, `rgb` und `RGB`. Das zwingend erforderliche Argument `Farbe` definiert eine Farbe innerhalb des gewählten Farbmodells. Eine Übersicht über die unterstützten Farbmodelle enthält Tabelle 6.9.

Tabelle 6.9: Übersicht über die unterstützten Farbmodelle

Farbmodell	Bedeutung
<code>cmk</code>	Bei diesem Farbmodell besteht jede Farbe aus vier Werten, stellvertretend für Cyan, Magenta, Gelb und Schwarz. Die vier Werte liegen im Zahlenraum zwischen 0 und 1 und sind durch Kommas voneinander abgetrennt.
<code>gray</code>	Die Farbangabe bei diesem Farbmodell besteht nur aus einem Wert zwischen 0 für schwarz und 1 für weiß.
<code>HTML</code>	Bei diesem Farbmodell besteht jede Farbe aus hexadezimalen Werten in der Form <code>RRGGBB</code> . Für jede der drei Grundfarben wird ein Wert (00-FF) angegeben. Die Funktionsweise ist identisch mit Farbangaben bei der Auszeichnungssprache HTML.
<code>rgb</code>	Dieses Farbmodell steht für <i>Rot-Grün-Blau</i> . Hier besteht jede Farbangabe aus drei durch Kommas abgetrennten Werten zwischen 0 und 1. Der jeweilige Wert steht für den Anteil an rotem, grünem und blauem Licht.
<code>RGB</code>	Auch dieses Farbmodell steht für <i>Rot-Grün-Blau</i> . Hier besteht jede Farbangabe aus drei durch Kommas abgetrennten Werten zwischen 0 und 255.

Ein Beispiel zum Befehl `\rowcolor` enthält Tabelle 6.10. In dieser wurden einzelne Zeilen mit dem Farbmodell `gray` eingefärbt. Der zugehörige Quelltext befindet sich in Listing 6.8.

Tabelle 6.10: Einfärben ganzer Zeilen einer Tabelle

Spalte 1	Spalte 2	Spalte 3
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3
x1y4	x2y4	x3y4
x1y5	x2y5	x3y5
x1y6	x2y6	x3y6
x1y7	x2y7	x3y7
x1y8	x2y8	x3y8
x1y9	x2y9	x3y9

```

1 \begin{tabular}{lcr}
2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 \rowcolor[gray]{0.9} x1y1 & x2y1 & x3y1 \\
6 \rowcolor[gray]{0.8} x1y2 & x2y2 & x3y2 \\
7 \rowcolor[gray]{0.7} x1y3 & x2y3 & x3y3 \\
8 \rowcolor[gray]{0.6} x1y4 & x2y4 & x3y4 \\
9 \rowcolor[gray]{0.5} x1y5 & x2y5 & x3y5 \\
10 \rowcolor[gray]{0.4} x1y6 & x2y6 & x3y6 \\
11 \rowcolor[gray]{0.3} x1y7 & x2y7 & x3y7 \\
12 \rowcolor[gray]{0.2} x1y8 & x2y8 & x3y8 \\
13 \rowcolor[gray]{0.1} x1y9 & x2y9 & x3y9 \\
14 \hline
15 \end{tabular}

```

Listing 6.8: Das Einfärben ganzer Tabellenzeilen ermöglicht der Befehl `rowcolor`

Das Einfärben von Spalten ermöglicht der Befehl `\columncolor`.

```
\columncolor[Farbmodell]{Farbe}[UeberhangL][UeberhangR]
```

Die Bedeutung der beiden Argumente *Farbmodell* und *Farbe* ist bereits vom Befehl `\rowcolor` bekannt. Die beiden optionalen Argumente *UeberhangL* und *UeberhangR* definieren den Überhang auf der linken bzw. rechten Seite. Mit *UeberhangL* und *UeberhangR* können Autoren also festlegen, wie weit die Färbung links und rechts über den Inhalt der Spalte hinausragt. Da diese beiden Argumente in der Praxis eher selten zum Einsatz kommen, werden Sie in diesem Werk nicht weiter betrachtet.

Ein Beispiel zum Befehl `\columncolor` enthält Tabelle 6.11. Der zugehörige Quelltext ist Listing 6.9. Auch in diesem Beispiel wurden einzelne Zeilen mit dem Farbmodell

gray eingefärbt. Der Befehl `\columncolor` wird in der Spaltendeklaration vor jeder einzufärbenden Spalte aufgerufen.

Tabelle 6.11: Einfärben ganzer Spalten einer Tabelle

Spalte 1	Spalte 2	Spalte 3	Spalte 4	Spalte 5
x1y1	x2y1	x3y1	x4y1	x5y1
x1y2	x2y2	x3y2	x4y2	x5y2
x1y3	x2y3	x3y3	x4y3	x5y3

```

1 \begin{tabular}{|>\columncolor[gray]{0.9}}c
2   |>\columncolor[gray]{0.7}}c
3   |>\columncolor[gray]{0.5}}c
4   |>\columncolor[gray]{0.3}}c
5   |>\columncolor[gray]{0.1}}c|}
6 \hline
7 Spalte 1 & Spalte 2 & Spalte 3 & Spalte 4 & Spalte 5 \\
8 \hline\hline
9 x1y1 & x2y1 & x3y1 & x4y1 & x5y1 \\
10 x1y2 & x2y2 & x3y2 & x4y2 & x5y2 \\
11 x1y3 & x2y3 & x3y3 & x4y3 & x5y3 \\
12 \hline
13 \end{tabular}

```

Listing 6.9: Das Einfärben ganzer Tabellenspalten ermöglicht der Befehl `columncolor`

Das Einfärben einzelner Felder einer Tabelle geschieht mit dem Befehl `\cellcolor`.

`\cellcolor[Farbmodell]{Farbe}`

Ein Beispiel zum Befehl `\cellcolor` enthält Tabelle 6.12. Der zugehörige Quelltext ist Listing 6.10.

Tabelle 6.12: Einfärben einzelner Felder einer Tabelle

Spalte 1	Spalte 2	Spalte 3
x1y1	x2y1	x3y1
x1y2	x2y2	x3y2
x1y3	x2y3	x3y3

```

1 \begin{tabular}{|c|c|c|}
2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 \cellcolor[gray]{0.9} x1y1 &
6 \cellcolor[gray]{0.8} x2y1 &
7 \cellcolor[gray]{0.7} x3y1 \\
8 \cellcolor[gray]{0.6} x1y2 &

```

```

9 \cellcolor[gray]{0.5} x2y2 &
10 \cellcolor[gray]{0.4} x3y2 \\
11 \cellcolor[gray]{0.3} x1y3 &
12 \cellcolor[gray]{0.2} x2y3 &
13 \cellcolor[gray]{0.1} x3y3 \\
14 \hline
15 \end{tabular}

```

Listing 6.10: Das Einfärben einzelner Tabellenfelder (Zellen) ermöglicht der Befehl `cellcolor`

Wird eine Farbe häufiger im Dokument verwendet, ist es sinnvoll, diese Farbe mit dem Befehl `\definecolor` in der Präambel des Quelltextes zu definieren und auf diese definierte Farbe immer wieder zuzugreifen. Dieses Vorgehen vereinfacht den Quelltext und eventuelle Änderungen an der Farbauswahl müssen später nur an einer einzigen Stelle vorgenommen werden. Beispielhaft für jedes der in Tabelle 6.9 vorgestellten Farbmodelle definieren die folgenden Befehle die Farbe Deep Sky Blue.

```

\definecolor{deepskyblue}{cmyk}{1,0.25,0,0}
\definecolor{deepskyblue}{HTML}{00BFFF}
\definecolor{deepskyblue}{rgb}{0.0,0.75,1.0}
\definecolor{deepskyblue}{RGB}{0,191,255}

```

Mit dem Farbmodell `gray` kann die Farbe Deep Sky Blue natürlich nicht dargestellt werden. Darum definiert der folgende Befehl exemplarisch ein helles Grau.

```

\definecolor{lightgray}{gray}{0.6}

```

Anschließend ist via `\rowcolor{deepskyblue}`, `\columncolor{deepskyblue}` oder `\cellcolor{deepskyblue}` die definierte Farbe nutzbar.

6.4 Felder zusammenfassen

Das Zusammenfassen von Feldern über die Grenzen von Spalten hinweg, geschieht mit dem Befehl `\multicolumn`.

```

\multicolumn{Anzahl}{Format}{Inhalt}

```

Um diesen Befehl zu verwenden, muss das gleichnamige Erweiterungspaket mit dem Befehl `\usepackage{multicol}` in der Präambel der `.tex`-Quelldatei eingebunden sein.

Der Parameter *Anzahl* definiert wie viele Felder zusammengefasst werden sollen und im Parameter *Format* wird mit den bekannten Positionszeichen *l*, *r* oder *c* gibt an, ob der Inhalt des neuen Felds linksbündig, rechtsbündig oder zentriert gesetzt wird. Auch vertikale Linien können hier mit *|* oder *||* angewiesen werden.

Ein Beispiel zum Befehl `\multicolumn` enthält Tabelle 6.13 und der zugehörige Quelltext befindet sich in Listing 6.11.

Tabelle 6.13: Spaltenübergreifendes Zusammenfassen von Feldern einer Tabelle

Oberste Zeile		
eins	zwei	drei
vier		fünf
sechs	sieben	

```

1 \begin{tabular}{|c|c|c|}
2 \hline
3 \multicolumn{3}{|c|}{Oberste Zeile} \\
4 \hline\hline
5 eins & zwei & drei \\
6 \hline
7 \multicolumn{2}{|l|}{vier} & fünf \\
8 \hline
9 sechs & \multicolumn{2}{r|}{sieben} \\
10 \hline
11 \end{tabular}

```

Listing 6.11: Der Befehl `\multicolumn` kann Felder über Spalten hinweg zusammenfassen

Hilfreich ist der Befehl `\multicolumn` auch dann, wenn nur die Ausrichtung des Inhalts eines Felds in einer Tabelle geändert werden soll. In diesem Fall erhält der Parameter *Anzahl* den Wert 1 und der Parameter *Format* das gewünschte Positionszeichen. Ein Beispiel zu diesem Vorgehen enthält Tabelle 6.14. Der zugehörige Quelltext ist Listing 6.12.

Tabelle 6.14: Änderung der Ausrichtung des Inhalts in einem einzelnen Feld

Hier ist die oberste Zeile		
links	links	links
links	rechts	links
links	links	links

```

1 \begin{tabular}{|p{2cm}|p{2cm}|p{2cm}|}
2 \hline

```



```

3 \multicolumn{3}{|c|}{Hier ist die oberste Zeile} \\
4 \hline\hline
5 links & links & links \\
6 \hline
7 links & \multicolumn{1}{r|}{\cellcolor[gray]{0.7} rechts} & links \\
8 \hline
9 links & links & links \\
10 \hline
11 \end{tabular}

```

Listing 6.12: Mit dem Befehl `\multicolumn` kann auch die Ausrichtung nur eines einzelnen Feldes angepasst werden

Das Zusammenfassen von Feldern über die Grenzen von Zeilen hinweg, geschieht mit dem Befehl `\multirow`.

```
\multirow{Anzahl}{Breite}{Inhalt}
```

Um diesen Befehl zu verwenden, muss das gleichnamige Erweiterungspaket mit dem Befehl `\usepackage{multirow}` in der Präambel der `.tex`-Quelldatei eingebunden sein.

Der Parameter *Anzahl* definiert wie viele Felder zusammenzufassen sind und *Breite* legt die Breite des zusammengefassten Felds fest. Hat dieser Parameter den Wert `*`, legt L^AT_EX die Breite automatisch anhand des Inhalts fest.

Ein Beispiel zum Befehl `\multirow` enthält Tabelle 6.15 und der zugehörige Quelltext befindet sich in Listing 6.13.

Tabelle 6.15: Das zeilenübergreifende Zusammenfassen von Feldern einer Tabelle ermöglicht der Befehl `\multirow`

Spalte 1	Spalte 2	Spalte 3
eins	zwei vier	drei fünf

```

1 \begin{tabular}{|c|c|c|}
2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 \multirow{2}{*}{eins} & zwei & drei \\
6 & vier & fünf \\
7 \hline
8 \end{tabular}

```

Listing 6.13: Mit dem Befehl `\multirow` können Felder zeilenübergreifend zusammengefasst werden

6.5 Horizontale Linien

Der Befehl `\hline`, mit dem einzelne oder doppelte horizontale Linien in Tabellen eingefügt werden, ist aus den Beispielen dieses Kapitel schon gut bekannt. Mit diesem Befehl ist es aber ausschließlich möglich, Linien über die komplette horizontale Breite einer Tabelle zu realisieren.

Mehr Flexibilität bietet der Befehl `\cline`, der in der nach seinem Aufruf folgenden Zeile eine horizontale Linie einfügt, die sich vom linken Rand der Spalte *SpalteL* bis zum rechten Rand der Spalte *SpalteR* erstreckt.

`\cline{SpalteL – SpalteR}`

Der mehrfache Aufruf von `\cline` direkt hintereinander ist zulässig. Soll beispielsweise unterhalb der gerade abgeschlossenen Zeile vom linken Rand der Spalte 1 bis zum rechten Rand der Spalte 3 und vom linken Rand der Spalte 5 bis zum rechten Rand der Spalte 9 eine horizontale Linie gesetzt werden, realisiert dieses die Befehlsfolge `\cline{1-3}\cline{5-9}`.

Ein Beispiel wo es sinnvoll sein kann, horizontale Linien nur durch einen Teil einer Tabelle laufen zu lassen, ist wenn Felder mit dem Befehl `\multirow` zusammengefasst werden. Ein Beispiel dafür zeigt Tabelle 6.16. Der zugehörige Quelltext befindet sich in Listing 6.14. Wäre in diesem Beispiel die Linie von der zweiten bis zur dritten Spalte anstatt mit dem Befehl `\cline{2-3}` mit `\hline` realisiert worden, hätte die horizontale Linie das zusammengefasste Feld in der ersten Spalte durchgeschnitten.

Tabelle 6.16: Horizontale Linien mit fast beliebiger Breite in Tabellen ermöglicht der Befehl `\cline`

Spalte 1	Spalte 2	Spalte 3
eins	zwei	drei
	vier	fünf

```

1 \begin{tabular}{|c|c|c|}
2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 \multirow{2}{*}{eins} & zwei & drei \\
6 \cline{2-3}
7 & vier & fünf \\
8 \hline
9 \end{tabular}

```

Listing 6.14: Horizontale Linien mit fast beliebiger Breite in Tabellen setzen

7 Bilder einfügen

Um Bilder in den Dateiformaten `.jpg`, `.png` oder `.pdf` in Dokumente einbinden zu können, muss mit das Erweiterungspaket `graphicx` mit dem Befehl `\usepackage{graphicx}` in der Präambel der `.tex`-Quelldatei eingebunden sein.

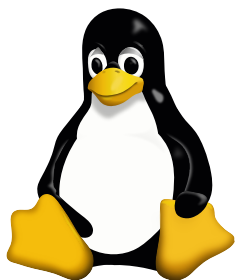
Das beste optische Ergebnis liefern Abbildungen, die als Vektorgrafiken vorliegen. Besonders bei selbst erstellten Diagrammen ist das problemlos möglich. Im Gegensatz zu Rastergrafiken ist das Skalieren von Vektorgrafiken ohne Qualitätsverlust stufenlos möglich. Dateiformate für Vektorgrafiken sind u.a. `eps`, `ps`, `pdf` und `svg`. Dateiformate für Rastergrafiken sind u.a. `bmp`, `gif`, `jpg` und `png`.

Das Einbindern der Bilder geschieht mit dem Befehl `\includegraphics`.

`\includegraphics[Optionen]{Dateiname}`

Optionen brauchen nicht zwingend angegeben werden. Allerdings kann es passieren, dass das Bild dann zu groß oder zu klein im fertigen Dokument erscheint. In den meisten Fällen ist es darum sinnvoll, zumindest mit `width=Breite` die Bildbreite oder mit `height=Höhe` die Bildhöhe festzulegen.

Das Beispiel in Abbildung 7.1 zeigt das Linux-Maskottchen Tux von Larry Ewing (Bildlizenz: CC0), skaliert auf eine Bildhöhe von 4 cm.



`1 \includegraphics[height=4cm]{Tux.pdf}`

Abbildung 7.1: Abbildungen skalieren die Optionen `height` oder alternativ `width`

Soll eine Abbildung genauso breit sein wie das Textfeld, können Autoren das

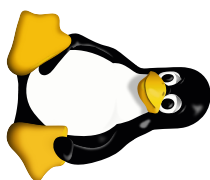
mit `width=\textwidth` realisieren. Das folgende Beispiel würde die Abbildung `bilddatei.pdf` so skalieren, dass die Bildbreite der halben Breite des Textes entspricht.

```
\includegraphics[width=.5\textwidth]{bilddatei.pdf}
```

Alternativ kann auch die Bildbreite fest inklusive einer Maßeinheit (siehe Abschnitt 2.4) definiert sein.

Das Drehen von Bildern ermöglicht der Parameter `angle=Winkel`.

Im Beispiel in Abbildung 7.2 ist das Linux-Maskottchen auf die Bildhöhe 3 cm skaliert und um 270 Grad gedreht.



```
1 \includegraphics[height=3cm,angle=270]{Tux.pdf}
```

Abbildung 7.2: Abbildungen drehen ermöglicht die Option `angle`

Meist ist es sinnvoll Abbildungen mit der Umgebung `figure` zu umschließen. Dadurch werden sie zu sogenannten Gleitobjekten. Vorteile von Gleitobjekten sind die automatisierte Positionierung sowie die Möglichkeit Bildunterschriften und Bezeichner zuweisen zu können. Ein mögliches Beispiel ist:

```
\begin{figure}[H]
\includegraphics[width=7cm]{Dateiname}
\caption{Hier ist die Bildunterschrift definiert}
\label{fig:eindeutiger_bezeichner}
\end{figure}
```

Ist einer Abbildung mit dem Befehl `\label{Name}` ein Bezeichner zugewiesen, kann an jeder beliebigen Stelle im Dokument mit dem Befehl `\ref{Name}` darauf verwiesen werden.

Mit dem Befehl `\caption{Bildunterschrift}` geschieht die Definition der Bildunterschrift.

7.1 Pfade der Bilddateien definieren

Beim Einbinden einer Abbildung mit dem Befehl `\includegraphics` kann der Pfad zur Bilddatei inklusive des Pfades relativ von der `.tex`-Quelldatei oder absolut von der Wurzel des Dateisystems angegeben sein. Einige Beispiele enthält Tabelle 7.1.

Tabelle 7.1: Beispiele für relative und absolute Pfade

Pfad	Betriebssystem	Beispiel
relativ	Linux/UNIX	<code>\includegraphics [...] {pfad/zur/bilddatei}</code>
absolut	Linux/UNIX	<code>\includegraphics [...] {/pfad/zur/bilddatei}</code>
relativ	Windows	<code>\includegraphics [...] {pfad\zur\bilddatei}</code>
absolut	Windows	<code>\includegraphics [...] {C:\pfad\zur\bilddatei}</code>

Dieses Vorgehen hat den Nachteil, dass wenn sich der Pfad zu den eingefügten Bilddateien einmal ändert, sind an mehreren Stellen im Quelltext Änderungen nötig. Aus diesem Grund ist es sinnvoll, den oder die Pfade zu den verwendeten Bilddateien zentral an einer Stelle in der Präambel der `.tex`-Quelldatei zu definieren. Dieses geschieht mit dem Befehl `\graphicspath`.

```
\graphicspath{{Verzeichnis1/}{Verzeichnis2/}...{Verzeichnisn/}}
```

Jeder als Argument übergebene Pfad (egal ob es sich dabei um einen relativen oder um einen absoluten Pfad handelt), sollte unter Linux mit einem Slash enden und unter Windows mit einem Backslash. Zudem muss jeder Pfad von geschweiften Klammern umschlossen sein.

8 Mathematischer Formelsatz

Die Fähigkeiten von \LaTeX , wenn es um den optisch ansprechenden Satz mathematischer Formeln geht, sind einer der am häufigsten genannten Vorzüge dieses Textsatzsystems. Möglicherweise ist diese Eigenschaft auch einer der Gründe für die Popularität von \LaTeX im universitären Umfeld und im Verlagswesen.

Zum Formelsatz bietet \LaTeX verschiedene Umgebungen für eingebettete Formeln, die auch Textformeln heißen und für abgesetzte Formeln.

Textformeln befinden sich im fortlaufenden Text. Abgesetzte Formeln hingegen werden mit einem Zwischenraum über und unter der Formel gesetzt und horizontal zentriert.

8.1 Eingebettete Formeln (Textformeln)

Die kompakteste Möglichkeit, eine eingebettete Formel zu realisieren, ist im Quelltext vor und hinter die Formel ein $\text{\$}$ -Zeichen zu stellen.

```
 $\text{\$Formel\$}$ 
```

Ein negativer Aspekt dieser Vorgehensweisen ist, dass beide Begrenzungssymbole identisch und daher zweideutig sind. Je nach Struktur des Dokuments wirkt sich das negativ auf die Lesbarkeit des Quelltextes aus. Eleganter und weniger fehleranfällig ist das Umschließen der Formel mit den Zeichenketten \textbackslash(und \textbackslash) .

```
 $\text{\textbackslash(Formel\)}$ 
```

Zusätzlich existiert die Umgebung `math`, deren Verwendung zum gleichen Ergebnis führt wie die Verwendung von $\text{\$}\dots\text{\$}$ und $\text{\textbackslash(}\dots\text{\textbackslash)}$.

```
\begin{math}  
Formel  
\end{math}
```

8.2 Abgesetzte Formeln

Auch zur Erzeugung abgesetzter Formeln existieren verschiedene Möglichkeiten. Eine Möglichkeit ist, den Anfang und das Ende der abgesetzten Formel mit `$$` zu kennzeichnen. Allerdings gibt es auch hier das Problem, dass beide Begrenzungssymbole identisch und daher zweideutig sind.

```
$$Formel$$
```

Eine weitere Möglichkeit ist, den Anfang der Formel mit `\[` und das Ende mit `\]` zu kennzeichnen.

```
\[Formel\]
```

Zusätzlich existiert die Umgebung `displaymath`, deren Verwendung zum gleichen Ergebnis führt wie die Verwendung von `$$...$$` und `\[...\]`.

```
\begin{displaymath}
Formel
\end{displaymath}
```

Alle bisher vorgestellten Möglichkeiten zum Satz abgesetzter Formeln bewirken, dass der \LaTeX -Compiler den aktuellen Absatz abbricht und einen vertikalen Freiraum einfügt. Die abgesetzte Formel wird horizontal zentriert gesetzt, und nach einem weiteren vertikalen Freiraum geht es mit dem Text weiter. Das folgende Beispiel zeigt diese Arbeitsweise:

$$(a + b)^2 = a^2 + 2ab + b^2$$

```
1 \begin{displaymath}
2 (a+b)^2 = a^2 + 2ab + b^2
3 \end{displaymath}
```

Sollen Formeln nummeriert sein, um im Fließtext einfach darauf Bezug nehmen zu können, ist es sinnvoll die abgesetzten Formeln mit der Umgebung `equation` zu erzeugen. In diesem Fall erhält die Formel automatisch eine eindeutige Nummer.

```
\begin{equation}
Formel
\end{equation}
```

Zum Satz von Gleichungssystemen bzw. für den Satz mehrerer untereinander angeordneter Formeln existieren die Umgebungen `eqnarray` und `eqnarray*`. Beide Umgebungen unterscheiden sich nur darin, dass bei der Umgebung `eqnarray` jede Zeile nummeriert wird und bei `eqnarray*` ist das nicht der Fall.

```
\begin{eqnarray}
Formel
\end{eqnarray}
```



```
\begin{eqnarray*}
Formel
\end{eqnarray*}
```

Die Arbeitsweise der beiden Umgebungen `eqnarray` und `eqnarray*` ist vergleichbar mit der Arbeitsweise einer Tabelle, die in drei Spalten unterteilt ist, und deren Spaltendefinition `{rcl}` entspricht. Das heißt, der Inhalt der ersten Spalte wird rechtsbündig, der Inhalt der zweiten Spalte zentriert, und der Inhalt der dritten Spalte wird linksbündig gesetzt.

Genau wie bei Tabellen sind Spalten mit dem Zeichen `&` voneinander abgegrenzt und das Ende einer Zeile kennzeichnet der Befehl `\\`.

Der Grund, warum die zweite Spalte zentriert gesetzt wird, ist, dass sie für die Aufnahme eines Relationssymbols gedacht ist, wie es bei Gleichungssystemen in jeder Zeile verwendet wird. In den meisten Fällen wird es sich um das Zeichen `=` handeln.

Die folgende Gleichung wurde mit der Umgebung `eqnarray*` gesetzt:

$$\begin{aligned}
 P(\neg C \vee D) &= 1 - P(\neg D \wedge C) \\
 &= 1 - P(\neg D|C) \cdot P(C) \\
 &= 1 - P(C) \cdot (1 - P(D|C)) \\
 &= 1 - P(C) + P(D|C) \cdot P(C) \\
 &= P(\neg C) + P(D|C) \cdot P(C)
 \end{aligned}$$

Der Quelltest zu dieser Gleichung aus dem Fachgebiet der Wissensverarbeitung wurde mit folgendem Quelltext erzeugt:

```
1 \begin{eqnarray*}
2 P(\neg C \vee D) &= & 1 - P(\neg D \wedge C) \\
3 &= & 1 - P(\neg D|C) \cdot P(C) \\
4 &= & 1 - P(C) \cdot (1 - P(D|C)) \\
5 &= & 1 - P(C) + P(D|C) \cdot P(C) \\
6 &= & P(\neg C) + P(D|C) \cdot P(C) \\
7 \end{eqnarray*}
```

Sollen bei einer Gleichung, die mit der Umgebung `eqnarray` gesetzt wird, einzelne Zeilen keine Nummer erhalten, können Autoren dieses in den jeweiligen Zeilen mit dem Befehl `\nonumber` anweisen.

```
\nonumber
```

Der Befehl wird einfach an das Ende jeder Zeile einer `eqnarray`-Umgebung geschrieben, die von der Nummerierung ausgeschlossen sein soll.

8.3 Grundlegende mathematische Konstrukte

Dieser Abschnitt stellt die nötigen Befehle vor, um einige der wichtigsten mathematischen Konstrukte realisieren können. Dazu gehören Brüche, Wurzeln, Indizes und Exponenten, Summen und Integrale.

8.3.1 Brüche

Der Satz von Brüchen ist innerhalb eingebetteter und abgesetzter Formeln mit dem Befehl `\frac` möglich.

`\frac{Zähler}{Nenner}`

Das erste Argument des Befehls enthält den *Zähler* über dem Bruchstrich und das zweite Argument den *Nenner* unter dem Bruchstrich (siehe Abbildung 8.1). *Zähler* und *Nenner* selbst können Formeln von fast beliebiger Komplexität enthalten.

$$\frac{a}{b} + \frac{c}{d} = \frac{a \cdot d + c \cdot b}{b \cdot d}$$

```

1 \[
2   \frac{a}{b} + \frac{c}{d} =
3   \frac{a\cdot d+c\cdot b}{b\cdot d}
4 \]
```

Abbildung 8.1: Der Satz von Brüchen geschieht mit dem Befehl `\frac`

Als eingebettete Formel, unter Verwendung von `$...$`, `\(...\)` oder der Umgebung `math`, sieht das Beispiel von Abbildung 8.1 deutlich kompakter aus: $\frac{a}{b} + \frac{c}{d} = \frac{a \cdot d + c \cdot b}{b \cdot d}$.

8.3.2 Indizes und Exponenten

Das Setzen eines Index geschieht mit Hilfe eines Unterstriches `_`. Alles, was nach dem Unterstrich kommt, setzt der \LaTeX -Compiler als Index der vorhergegangenen Teilformel.

`_ {Index}`

Das Setzen von Exponenten geschieht mit dem Zeichen `^`. Das darauf folgende Zeichen (oder die darauf folgende Teilformel) setzt der \LaTeX -Compiler als Exponent.

`^ {Exponent}`

Es verbessert die Lesbarkeit des Quelltextes, wenn Indizes und Exponenten immer in geschweiften Klammern stehen und wenn ein Index oder Exponent aus mehr als

nur einem einzigen Zeichen besteht, müssen diese auch auch zwingend in geschweiften Klammern stehen.

$a^2 + a^b - a_2 + a_i^n$	<code>\[a^{2} + a^{b} - a_{2} + a^{n}_{i} \]</code>
$(a+b)^2 = a^2 + 2ab + b^2$	<code>\[(a+b)^{2} = a^{2} + 2ab + b^{2} \]</code>

Sollen gleichzeitig ein Index und ein Exponenten an ein Zeichen angehängt werden, ist deren Reihenfolge im Quelltext gleichgültig. Es ist also egal ob a_y^x so: `\(a^{\{x\}}_{\{y\}}\)` oder so: `\(a_{\{y\}}^{\{x\}}\)` realisiert ist.

$a^{n+1} + a^{n_x}$	$\backslash [\text{a}^{\{n+1\}} + \text{a}^{\{n_x\}} \backslash]$
$a^{b^{c^2}}$	$\backslash [\text{a}^{\{b^{\{c^2\}}\}} \backslash]$
$a_{c_{n+m}^{2n-1}}^{b_i^2}$	$\backslash [\text{a}^{\{b^2_i\}}_c^{\{2n-1\}}_{\{n+m\}} \backslash]$
$a_1^{e^{b^2}}$	$\backslash [\text{a}^{\{e^{\{b^2\}}\}}_1 \backslash]$

Da L^AT_EX die beiden Zeichen `^` oder `_` nur zur Definition von Indizes und Exponenten in mathematische Umgebungen akzeptiert, können diese im Fließtext nicht direkt eingegeben werden. In diesem Fall müssen Autoren auf die Befehle `\textasciicircum` für das Zeichen `^` und `\textunderscore` oder alternativ `_` für das Zeichen `_` zurückgreifen.

8.3.3 Wurzeln

Das Setzen von Wurzeln ermöglicht der Befehl `sqrt`.

$$\sqrt[n]{\text{Radikant}}$$

Im ersten Argument ist die *Ordnung* der Wurzel definiert und im zweiten Argument der *Radikant*. Die Ordnung wird immer in eckigen Klammern [...] geschrieben und der Radikant muss in geschweiften Klammern stehen {...}.

Beide Argumente können fast beliebig komplexe Formeln enthalten. Die Größe des Wurzelzeichens wird legt der L^AT_EX-Compiler automatisch fest, wie das Beispiel in Abbildung 8.2 anschaulich zeigt.

[illegible]

Abbildung 8.2: Der Satz von Wurzeln geschieht mit dem Befehl `\sqrt`

Es folgen einige weitere Beispiele, die das Setzen von Wurzeln demonstrieren:

$\sqrt{x} + \sqrt{x+y+2}$	<code>\[\sqrt{x} + \sqrt{x+y+2} \]</code>
$\sqrt[3]{x} + \sqrt[y+1]{x}$	<code>\[\sqrt[3]{x} + \sqrt[y+1]{x} \]</code>
$1 + \sqrt[x]{x+2}$	<code>\[1+\sqrt[x]{x+2} \]</code>
$\sqrt{x^9 + \sqrt{\alpha}}$	<code>\[\sqrt{x^9+\sqrt{\alpha}} \]</code>
$a^{-r} = \frac{1}{\sqrt[p]{a^p}}$	<code>\[a^{-r} = \frac{1}{\sqrt[p]{a^p}} \]</code>
$\sqrt[13]{\frac{y+b\sin\theta}{e^{i\vartheta}}}$	<code>\[\sqrt[13]{\frac{y+b\sin\theta}{e^{i\vartheta}}} \]</code>

8.3.4 Summen und Integrale

Dieser Abschnitt präsentiert einige mathematische Zeichen, darunter unter anderem das Summen-, das Integral-, das Produktzeichen und weitere Zeichen. Genau wie beim Wurzelzeichen, das im vorherigen Unterabschnitt vorgestellt wurde, hängt auch bei den Zeichen in diesem Abschnitt die Größe, in der das betreffende Zeichen gesetzt wird, von der verwendeten Umgebung ab. Das heißt konkret, dass die Darstellung in einer eingebetteten Formel kleiner ist als in einer abgesetzten Formel.

Stellvertretend für alle Zeichen in diesem Abschnitt wird an dieser Stelle das Summenzeichen vorgestellt, dessen Satz mit dem Befehl `\sum` geschieht.

$$\boxed{\text{\code{\sum_{unter dem Zeichen}^{über dem Zeichen}}}}$$

Die eingebettete Formel $\sum_{n=0}^x$ wurde mit folgenden L^AT_EX-Quelltext gesetzt: `\(\sum_{n=0}^x\)`. Als abgesetzte Formel, zum Beispiel mit der Umgebung `displaymath` oder unter Verwendung von `\[...\]`, sieht der Satz deutlich anders aus:

$$\sum_{n=0}^x \quad \text{\code{\[\sum_{n=0}^x \]}}$$

Tabelle 8.1 enthält eine Übersicht über die Größen verschiedener mathematischer Zeichen. Konkret ist in der Tabelle jedes Zeichen in der Form gesetzt, wie es als eingebettete Formel der Fall ist, und wie es in einer abgesetzten Formel der Fall ist. Da abgesetzte Formeln nicht in Tabellen vorkommen dürfen, wurde die gewünschte Darstellung mit dem Befehl `\displaystyle` erzwungen.

$$\boxed{\text{\code{\displaystyle}}}$$

Tabelle 8.1: Größenübersicht verschiedener mathematischer Zeichen

abgesetzte Formel	eingebettete Formel	Befehl
$\bigcap\limits_{x,n}$	\bigcap_x^n	<code>\bigcap_{x}^n</code>
$\bigcup\limits_{x,n}$	\bigcup_x^n	<code>\bigcup_{x}^n</code>
$\bigotimes\limits_{x,n}$	\bigotimes_x^n	<code>\bigotimes_{x}^n</code>
$\bigoplus\limits_{x,n}$	\bigoplus_x^n	<code>\bigoplus_{x}^n</code>
$\bigodot\limits_{x,n}$	\bigodot_x^n	<code>\bigodot_{x}^n</code>
$\bigsqcup\limits_{x,n}$	\bigsqcup_x^n	<code>\bigsqcup_{x}^n</code>
$\biguplus\limits_{x,n}$	\biguplus_x^n	<code>\biguplus_{x}^n</code>
$\bigvee\limits_{x,n}$	\bigvee_x^n	<code>\bigvee_{x}^n</code>
$\bigwedge\limits_{x,n}$	\bigwedge_x^n	<code>\bigwedge_{x}^n</code>
$\int\limits_{x,n}^x$	\int_x^n	<code>\int_{x}^n</code>
$\coprod\limits_{x,n}$	\coprod_x^n	<code>\coprod_{x}^n</code>
$\oint\limits_{x,n}^x$	\oint_x^n	<code>\oint_{x}^n</code>
$\prod\limits_{x,n}$	\prod_x^n	<code>\prod_{x}^n</code>
$\sqrt[n]{x}$	$\sqrt[n]{x}$	<code>\sqrt[n]{x}</code>
$\sum\limits_x^n$	\sum_x^n	<code>\sum_{x}^n</code>

8.3.5 Informationen über oder unter Zeichen positionieren

Das Setzen von einzeiligen Informationen in kleinerer Schrift über oder unter einzelnen Zeichen ermöglichen die Befehle `\overset`, `\underset`.

$\backslash\overset{oben}{\underset{unten}{}{}}$

Der Befehl `\overset` platziert den Inhalt des Arguments *oben* über dem Inhalt des Arguments *unten* und der Befehl `\underset` platziert den Inhalt des Arguments *unten* unter den Inhalt von *oben*.

Beide Befehle sind Teil des Erweiterungspakets **amsmath** und stehen zur Verfügung, sobald das Paket in der Präambel der Quelldatei mit dem Befehl `\usepackage{amsmath}` eingebunden ist.

$$\overset{y}{X} \quad \underset{y}{X} \quad \overset{a}{\underset{b}{X}}$$

```

1 \[
2   \overset{y}{X}
3   \qquad
4   \underset{y}{X}
5   \qquad
6   \overset{a}{\underset{b}{X}}
7 \]
```

Bei der Bildung einer Summe oder zur Indizierung ist es manchmal notwendig, mehrzeilige Angaben zu machen. Das Erweiterungspaket **amsmath** enthält für diesen Zweck den Befehl `\substack`.

$\backslash\substack{\textit{mehrzeilige Formel}}$

Das Ende einer Zeile wird im Argument *mehrzeilige Formel* mit dem Befehl `\` angewiesen.

$$\sum_{\substack{0 \leq i \leq m \\ 0 \leq j \leq m}} P(i, j)$$

```

1 \[
2   \sum_{\substack{0 \leq i \leq m \\ 0 \leq j \leq m}} P(i, j)
3
4 \]
```

8.4 Mathematische Zeichen und Symbole

Dieser Abschnitt enthält eine Übersicht über verschiedene mathematische Zeichen und Symbole.

8.4.1 Griechische Buchstaben im Mathematikmodus

Griechische Buchstaben sind in der Mathematik häufig verwendete Zeichen. Die Erzeugung einiger dieser Buchstaben geschieht in einer mathematischen Umgebung automatisch, indem *normale* Buchstaben verwendet werden. Für die übrigen Buchstaben existiert eigene Befehle (siehe Tabelle 8.2 und Tabelle 8.3).

Tabelle 8.2: Griechische Großbuchstaben

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
<i>A</i>	<code>A</code>	<i>I</i>	<code>I</code>	<i>P</i>	<code>P</code>
<i>B</i>	<code>B</code>	<i>K</i>	<code>K</code>	Σ	<code>\Sigma</code>
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	<i>T</i>	<code>T</code>
Δ	<code>\Delta</code>	<i>M</i>	<code>M</code>	Υ	<code>\Upsilon</code>
<i>E</i>	<code>E</code>	<i>N</i>	<code>N</code>	Φ	<code>\Phi</code>
<i>Z</i>	<code>Z</code>	Ξ	<code>\Xi</code>	<i>X</i>	<code>X</code>
<i>H</i>	<code>H</code>	<i>O</i>	<code>O</code>	Ψ	<code>\Psi</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Ω	<code>\Omega</code>

Tabelle 8.3: Griechische Kleinbuchstaben

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
α	<code>\alpha</code>	ι	<code>\iota</code>	ρ	<code>\rho</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	σ	<code>\sigma</code>
γ	<code>\gamma</code>	λ	<code>\lambda</code>	τ	<code>\tau</code>
δ	<code>\delta</code>	μ	<code>\mu</code>	υ	<code>\upsilon</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	ϕ	<code>\phi</code>
ζ	<code>\zeta</code>	ξ	<code>\xi</code>	χ	<code>\chi</code>
η	<code>\eta</code>	\omicron	<code>\omicron</code>	ψ	<code>\psi</code>
θ	<code>\theta</code>	π	<code>\pi</code>	ω	<code>\omega</code>

Wie in Tabelle 8.2 und Tabelle 8.3 zu sehen ist, setzt der L^AT_EX-Compiler griechischen Großbuchstaben üblicherweise in der Schrift Roman und griechische Kleinbuchstaben in der geeigneten Schrift *Italic* (siehe Abschnitt 5.3). Sollen auch Großbuchstaben in geneigter Schrift ausgegeben werden, können Autoren dieses mit dem Befehl `\mathnormal` anweisen.

`\mathnormal{Zeichen}`

Das Argument `Zeichen` enthält die kursiv zu schreibenden griechischen Großbuchstaben.

 $\Gamma\Delta\Theta\Pi$

`\(\ \mathnormal{\Gamma\Delta\Theta\Pi} \)`

Tabelle 8.4 zeigt Varianten einiger griechischen Kleinbuchstaben, die u.a. als Variablen nützlich sind.

Tabelle 8.4: Variierte griechische Kleinbuchstaben

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
ε	<code>\varepsilon</code>	ϖ	<code>\varpi</code>	ς	<code>\varsigma</code>
ϑ	<code>\vartheta</code>	ϱ	<code>\varrho</code>	φ	<code>\varphi</code>

8.4.2 Kalligrafische Buchstaben

Den Satz von *KALLIGRAFISCHEN* Buchstaben innerhalb mathematischer Umgebungen ermöglicht der Befehl `\mathcal`.

`\mathcal{Zeichen}`

Im Gegensatz zu den griechischen Buchstaben sind die kalligrafischen Buchstaben nur als Großbuchstaben verfügbar.

$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}$

`\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}`

Eine Übersicht über die verfügbaren kalligrafischen Buchstaben enthält Tabelle 8.5.

Tabelle 8.5: Kalligrafische Zeichen

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
\mathcal{A}	<code>\mathcal{A}</code>	\mathcal{J}	<code>\mathcal{J}</code>	\mathcal{S}	<code>\mathcal{S}</code>
\mathcal{B}	<code>\mathcal{B}</code>	\mathcal{K}	<code>\mathcal{K}</code>	\mathcal{T}	<code>\mathcal{T}</code>
\mathcal{C}	<code>\mathcal{C}</code>	\mathcal{L}	<code>\mathcal{L}</code>	\mathcal{U}	<code>\mathcal{U}</code>
\mathcal{D}	<code>\mathcal{D}</code>	\mathcal{M}	<code>\mathcal{M}</code>	\mathcal{V}	<code>\mathcal{V}</code>
\mathcal{E}	<code>\mathcal{E}</code>	\mathcal{N}	<code>\mathcal{N}</code>	\mathcal{W}	<code>\mathcal{W}</code>
\mathcal{F}	<code>\mathcal{F}</code>	\mathcal{O}	<code>\mathcal{O}</code>	\mathcal{X}	<code>\mathcal{X}</code>
\mathcal{G}	<code>\mathcal{G}</code>	\mathcal{P}	<code>\mathcal{P}</code>	\mathcal{Y}	<code>\mathcal{Y}</code>
\mathcal{H}	<code>\mathcal{H}</code>	\mathcal{Q}	<code>\mathcal{Q}</code>	\mathcal{Z}	<code>\mathcal{Z}</code>
\mathcal{I}	<code>\mathcal{I}</code>	\mathcal{R}	<code>\mathcal{R}</code>		

8.4.3 Operationssymbole

Werden in der Mathematik zwei Objekte mit jeweils einer bestimmten Größe miteinander verknüpft und entsteht dabei ein neues Objekt mit einer bestimmten Größe, dann heißt diese Verknüpfung eine *binäre Operation*. Tabelle 8.6 zeigt eine Auswahl an binären Operationssymbolen.

Tabelle 8.6: Binäre Operationssymbole

Zeichen	Befehl	Zeichen	Befehl
\amalg	<code>\amalg</code>	\ominus	<code>\ominus</code>
\ast	<code>\ast</code>	\oplus	<code>\oplus</code>
\bigcirc	<code>\bigcirc</code>	\oslash	<code>\oslash</code>
\bigtriangledown	<code>\bigtriangledown</code>	\otimes	<code>\otimes</code>
\bigtriangleup	<code>\bigtriangleup</code>	\pm	<code>\pm</code>
\Box	<code>\Box</code>	\rhd	<code>\rhd</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>
\cap	<code>\cap</code>	\sqcap	<code>\sqcap</code>
\cdot	<code>\cdot</code>	\sqcup	<code>\sqcup</code>
\circ	<code>\circ</code>	\star	<code>\star</code>
\cup	<code>\cup</code>	\times	<code>\times</code>
\dagger	<code>\dagger</code>	\triangleleft	<code>\triangleleft</code>
\ddagger	<code>\ddagger</code>	\triangleright	<code>\triangleright</code>
\diamond	<code>\Diamond</code>	\trianglelefteq	<code>\unlhd</code>
\diamond	<code>\diamond</code>	\trianglerighteq	<code>\unrhd</code>
\div	<code>\div</code>	\oplus	<code>\uplus</code>
\triangleleft	<code>\lhd</code>	\vee	<code>\vee</code>
\mp	<code>\mp</code>	\wedge	<code>\wedge</code>
\odot	<code>\odot</code>	\wr	<code>\wr</code>

Diejenigen binären Operationssymbole in Tabelle 8.6, die mit einem `*` gekennzeichnet sind, erfordern das Einbinden des `latexsym`-Pakets mit dem Befehl `\usepackage{latexsym}` in der Präambel der Quelldatei.

8.4.4 Vergleichssymbole

Tabelle 8.7 zeigt eine Auswahl an Vergleichssymbolen.

Die umgekehrte, also verneinende Bedeutung eines solchen Vergleichssymbols wird in der Mathematik durch ein Durchstreichen des betreffenden Symbols mit einem Schrägstrich (*Slash*) `/` gekennzeichnet. So ist die Negation von $=$ (*gleich*) beispielsweise \neq (*ungleich*). Die meisten Negationen können durch Voranstellen eines `\not` erzeugt werden. Für einige wenige existiert ein eigener, spezieller Befehl (z.B. `\ne` für \neq oder `\notin` für \notin).

Tabelle 8.8 zeigt eine Auswahl an Negationen von Vergleichssymbolen.

Tabelle 8.7: Vergleichssymbole

Zeichen	Befehl	Zeichen	Befehl
$<$	<code><</code>	\ni	<code>\ni</code>
$>$	<code>></code>	\parallel	<code>\parallel</code> oder <code>\ </code>
$=$	<code>=</code>	\perp	<code>\perp</code>
\approx	<code>\approx</code>	\prec	<code>\prec</code>
\asymp	<code>\asymp</code>	\preceq	<code>\preceq</code>
\bowtie	<code>\bowtie</code>	\propto	<code>\propto</code>
\cong	<code>\cong</code>	\sim	<code>\sim</code>
\dashv	<code>\dashv</code>	\smile	<code>\smile</code>
\doteq	<code>\doteq</code>	\sqsubset	<code>\sqsubset</code>
\equiv	<code>\equiv</code>	\sqsupset	<code>\sqsupset</code>
\frown	<code>\frown</code>	\subset	<code>\subset</code>
\geq	<code>\ge</code> oder <code>\geq</code>	\subseteq	<code>\subseteq</code>
\gg	<code>\gg</code>	\succ	<code>\succ</code>
\in	<code>\in</code>	\succeq	<code>\succeq</code>
\leq	<code>\le</code> oder <code>\leq</code>	\supset	<code>\supset</code>
\ll	<code>\ll</code>	\supseteq	<code>\supseteq</code>
$ $	<code>\mid</code> oder <code> </code>	\simeq	<code>\simeq</code>
\models	<code>\models</code>	\vdash	<code>\vdash</code>

Diejenigen Symbole in Tabelle 8.9, die mit einem * markiert sind, sind nur nach einer Einbindung des Erweiterungspakets `latexsym` verfügbar.

Einige zusätzliche Vergleichssymbole (siehe Tabelle 8.9) bietet das Erweiterungspaket `amssymb`, das mit dem Befehl `\usepackage{amssymb}` in der Präambel der Quelldatei eingebunden wird.

8.4.5 Pfeile und Zeiger

Tabelle 8.10 zeigt eine Auswahl an Pfeilen. Diejenigen Symbole, die mit einem * markiert sind, sind nur nach einer Einbindung des Erweiterungspakets `latexsym` verfügbar.

Es existieren zwei Möglichkeiten, um den Doppelpfeil \iff zu erzeugen. Eine Möglichkeit, ist die Verwendung des aus Tabelle 8.10 bekannten Befehls `\Longleftarrow`. Eine alternative Möglichkeit ist die Verwendung des Befehls `\iff`. Allerdings unterscheiden sich beide Befehle im Zwischenraum, den \LaTeX vor und hinter dem Pfeil einfügt. Bei `\Longleftarrow` (\iff) ist der Zwischenraum etwas kleiner als bei `\iff`.

Tabelle 8.8: Negierte Vergleichssymbole

Zeichen	Befehl	Zeichen	Befehl
\nless	<code>\not<</code>	\nparallel	<code>\not\parallel</code>
\ngtr	<code>\not></code>	\nperp	<code>\not\perp</code>
\neq	<code>\not=</code> oder <code>\neq</code> oder <code>\ne</code>	\nprec	<code>\not\prec</code>
\napprox	<code>\not\approx</code>	\npreceq	<code>\not\preceq</code>
\nasymp	<code>\not\asymp</code>	\npropto	<code>\not\propto</code>
\nbowtie	<code>\not\bowtie</code>	\nsim	<code>\not\sim</code>
\ncong	<code>\not\cong</code>	\nsimeq	<code>\not\simeq</code>
\ndashv	<code>\not\dashv</code>	\nsmile	<code>\not\smile</code>
\ndoteq	<code>\not\doteq</code>	\nsubset^*	<code>\not\subset^*</code>
\nequiv	<code>\not\equiv</code>	\nsubseteq	<code>\not\subseteq</code>
\nfrown	<code>\not\frown</code>	\nsupset^*	<code>\not\supset^*</code>
\nge oder \ngeq	<code>\not\ge</code> oder <code>\not\geq</code>	\nsupseteq	<code>\not\supseteq</code>
\ngg	<code>\not\gg</code>	\nsubset	<code>\not\subset</code>
\notin	<code>\notin</code>	\nin	<code>\not\in</code>
\nsubseteq	<code>\not\subseteq</code>	\nJoin^*	<code>\not\Join^*</code>
\nsucc	<code>\not\succ</code>	\nle oder \nleq	<code>\not\le</code> oder <code>\not\leq</code>
\nsucceq	<code>\not\succeq</code>	\nll	<code>\not\ll</code>
\nsupset	<code>\not\supset</code>	\nmid	<code>\not\mid</code>
\nsupseteq	<code>\not\supseteq</code>	\nmodels	<code>\not\models</code>
\nvdash	<code>\not\vdash</code>	$\n ni$	<code>\not\ni</code>

Neben den in Tabelle 8.10 vorgestellten Pfeilen, bietet das Erweiterungspaket `amsmath` zahlreiche Pfeile und Zeiger. Eine Übersicht enthält Tabelle 8.11

Zusätzlich stellt das Erweiterungspaket `amsmath` noch einige negierte (durchgestrichene) Pfeile bereit. Eine Übersicht enthält Tabelle 8.12.

8.4.6 Sonstige mathematische Zeichen

Tabelle 8.13 präsentiert weitere Symbole aller Art, die dem Bereich der mathematischen Zeichen zugeordnet werden können. Einige dieser Zeichen sind Teil des Erweiterungspakets `amsmath`.

Tabelle 8.9: Zusätzliche Vergleichssymbole

Zeichen	Befehl	Zeichen	Befehl
\ni	<code>\backepsilon</code>	\backsimeq	<code>\precsim</code>
\backsimeq	<code>\backsimeq</code>	\risingdotseq	<code>\risingdotseq</code>
\bumpeq	<code>\Bumpeq</code>	\shortparallel	<code>\shortparallel</code>
\circeq	<code>\circeq</code>	\smallsmile	<code>\smallsmile</code>
\eqslantgtr	<code>\eqslantgtr</code>	\sqsubset	<code>\sqsubset</code>
\gtrdot	<code>\gtrdot</code>	\sqsupset	<code>\sqsupset</code>
\gtreqless	<code>\gtreqless</code>	\succsim	<code>\succsim</code>
\Join	<code>\Join</code>	\thickapprox	<code>\thickapprox</code>
\leqq	<code>\leqq</code>	\trianglelefteq	<code>\trianglelefteq</code>
\lessdot	<code>\lessdot</code>	\trianglerighteq	<code>\trianglerighteq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\varpropto	<code>\varpropto</code>
\lessssim	<code>\lessssim</code>		

Tabelle 8.10: Pfeile

Zeichen	Befehl	Zeichen	Befehl
\downarrow	<code>\downarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Downarrow	<code>\Downarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\hookleftarrow	<code>\hookleftarrow</code>	\mapsto	<code>\mapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\nearrow	<code>\nearrow</code>
\leadsto^*	<code>\leadsto^*</code>	\nwarrow	<code>\nwarrow</code>
\leftarrow oder <code>\gets</code>	<code>\leftarrow</code> oder <code>\gets</code>	\rightarrow oder <code>\to</code>	<code>\rightarrow</code> oder <code>\to</code>
\Leftarrow	<code>\Leftarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftrightarrow	<code>\leftrightarrow</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\searrow	<code>\searrow</code>
\longleftarrow	<code>\longleftarrow</code>	\swarrow	<code>\swarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\uparrow	<code>\uparrow</code>
\longleftrightarrow	<code>\longleftrightarrow</code>	\Uparrow	<code>\Uparrow</code>
\Longleftrightarrow	<code>\Longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\longmapsto	<code>\longmapsto</code>	\Updownarrow	<code>\Updownarrow</code>

Tabelle 8.11: Pfeile und Zeiger

Zeichen	Befehl	Zeichen	Befehl
\circlearrowleft	<code>\circlearrowleft</code>	\looparrowleft	<code>\looparrowleft</code>
\circlearrowright	<code>\circlearrowright</code>	\Lsh	<code>\Lsh</code>
\curvearrowleft	<code>\curvearrowleft</code>	\multimap	<code>\multimap</code>
\curvearrowright	<code>\curvearrowright</code>	\rightarrowtail	<code>\rightarrowtail</code>
\dashleftarrow	<code>\dashleftarrow</code>	\rightleftarrows	<code>\rightleftarrows</code>
\dashrightarrow	<code>\dashrightarrow</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\downdownarrows	<code>\downdownarrows</code>	\rightrightarrows	<code>\rightrightarrows</code>
\downharpoonleft	<code>\downharpoonleft</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\downharpoonright	<code>\downharpoonright</code>	\Rightarrow	<code>\Rightarrow</code>
\leftarrowtail	<code>\leftarrowtail</code>	\Rsh	<code>\Rsh</code>
\leftrightsquigarrow	<code>\leftrightsquigarrow</code>	\twoheadleftarrow	<code>\twoheadleftarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>
		\upuparrows	<code>\upuparrows</code>
		\upharpoonleft	<code>\upharpoonleft</code>
		\upharpoonright	<code>\upharpoonright</code>

Tabelle 8.12: Negierte Pfeile

Zeichen	Befehl	Zeichen	Befehl
\nleftarrow	<code>\nleftarrow</code>	\nLeftarrow	<code>\nLeftarrow</code>
\nrightarrow	<code>\nrightarrow</code>	\nRightarrow	<code>\nRightarrow</code>
\nleftrightarrow	<code>\nleftrightarrow</code>	\nLeftrightarrow	<code>\nLeftrightarrow</code>
\nrightarrowtail	<code>\nrightarrowtail</code>	\nRrightarrow	<code>\nRrightarrow</code>

8.4.7 Mathematische Funktionen

In der Literatur werden mathematische Funktionen üblicherweise nicht – wie die Namen von Variablen – kursiv geschrieben, sondern mit aufrechter Roman-Schrift gesetzt. Eine Übersicht über mathematische Funktionen und deren Befehl zeigt Tabelle 8.14.

Der Eintrag `gcd` in Tabelle 8.14 entspricht dem deutschen `ggT` (größter gemeinsamer Teiler) und das `mod` nach rechts eingezogen ist, ist kein Druckfehler, sondern eine Eigenheit des L^AT_EX-Compilers.

Tabelle 8.13: Sonstige mathematische Zeichen

Zeichen	Befehl	Zeichen	Befehl
\aleph	<code>\aleph</code>	\flat	<code>\flat</code>
\neg	<code>\neg</code>	\angle	<code>\angle</code>
\forall	<code>\forall</code>	∂	<code>\partial</code>
\backslash	<code>\backslash</code>	\hbar	<code>\hbar</code>
\prime	<code>\prime</code>	\bot	<code>\bot</code>
\heartsuit	<code>\heartsuit</code>	\Re	<code>\Re</code>
\clubsuit	<code>\clubsuit</code>	\imath	<code>\imath</code>
\spadesuit	<code>\spadesuit</code>	\diamondsuit	<code>\diamondsuit</code>
∞	<code>\infty</code>	\surd	<code>\surd</code>
ℓ	<code>\ell</code>	\diamond	<code>\diamond</code>
\top	<code>\top</code>	\emptyset	<code>\emptyset</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>
\exists	<code>\exists</code>	\natural	<code>\natural</code>
\wp	<code>\wp</code>	\Im	<code>\Im</code>
\sharp	<code>\sharp</code>	\parallel	<code>\parallel</code>
\backprime	<code>\backprime</code>	\diamond	<code>\Diamond</code> oder <code>\lozenge</code>
\Bbbk	<code>\Bbbk</code>	\eth	<code>\eth</code>
\bigstar	<code>\bigstar</code>	\Finv	<code>\Finv</code>
\blacksquare	<code>\blacksquare</code>	\Game	<code>\Game</code>
\blacklozenge	<code>\blacklozenge</code>	\hslash	<code>\hslash</code>
\blacktriangle	<code>\blacktriangle</code>	\measuredangle	<code>\measuredangle</code>
\blacktriangledown	<code>\blacktriangledown</code>	\mho	<code>\mho</code>
\square	<code>\Box</code> oder <code>\square</code>	\nexists	<code>\nexists</code>
\textcircled{S}	<code>\circledS</code>	\sphericalangle	<code>\sphericalangle</code>
\complement	<code>\complement</code>	\triangledown	<code>\triangledown</code>
\diagdown	<code>\diagdown</code>	\varnothing	<code>\varnothing</code>
\diagup	<code>\diagup</code>	\triangle	<code>\vartriangle</code>

8.4.8 Akzente in Formeln

Auch in einer mathematischen Umgebung ist es möglich, Akzente zu verwenden. Die in Tabelle 8.15 gezeigten Akzente unterscheiden sich von denen im Fließtext.

Doppelakzente sind mit den in Tabelle 8.15 gezeigten Befehlen auch realisierbar, wie Tabelle 8.16 zeigt.

Zusätzlich zu den in diesem Abschnitt vorgestellten Akzenten, stellt das Erweiterungs-

Tabelle 8.14: Mathematische Funktionen

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
\arccos	<code>\arccos</code>	\exp	<code>\exp</code>	\max	<code>\max</code>
\arcsin	<code>\arcsin</code>	\gcd	<code>\gcd</code>	\min	<code>\min</code>
\arctan	<code>\arctan</code>	\hom	<code>\hom</code>	\mod	<code>\bmod</code>
\arg	<code>\arg</code>	\inf	<code>\inf</code>	\mod	<code>\mod</code>
\cos	<code>\cos</code>	\ker	<code>\ker</code>	\Pr	<code>\Pr</code>
\cosh	<code>\cosh</code>	\lg	<code>\lg</code>	\sec	<code>\sec</code>
\cot	<code>\cot</code>	\lim	<code>\lim</code>	\sin	<code>\sin</code>
\coth	<code>\coth</code>	\liminf	<code>\liminf</code>	\sinh	<code>\sinh</code>
\csc	<code>\csc</code>	\limsup	<code>\limsup</code>	\sup	<code>\sup</code>
\deg	<code>\deg</code>	\ln	<code>\ln</code>	\tan	<code>\tan</code>
\det	<code>\det</code>	\log	<code>\log</code>	\tanh	<code>\tanh</code>
\dim	<code>\dim</code>				

Tabelle 8.15: Akzente in Formeln

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
\acute{a}	<code>\acute{a}</code>	\dot{a}	<code>\dot{a}</code>	\hat{a}	<code>\hat{a}</code>
\bar{a}	<code>\bar{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\tilde{a}	<code>\tilde{a}</code>
\breve{a}	<code>\breve{a}</code>	\grave{a}	<code>\grave{a}</code>	\vec{a}	<code>\vec{a}</code>
\check{a}	<code>\check{a}</code>				

Tabelle 8.16: Doppelakzente in Formeln

Zeichen	Befehl	Zeichen	Befehl
$\acute{\acute{a}}$	<code>\acute{\acute{a}}</code>	$\ddot{\ddot{a}}$	<code>\ddot{\ddot{a}}</code>
$\bar{\bar{a}}$	<code>\bar{\bar{a}}</code>	$\grave{\grave{a}}$	<code>\grave{\grave{a}}</code>
$\breve{\breve{a}}$	<code>\breve{\breve{a}}</code>	$\hat{\hat{a}}$	<code>\hat{\hat{a}}</code>
$\check{\check{a}}$	<code>\check{\check{a}}</code>	$\tilde{\tilde{a}}$	<code>\tilde{\tilde{a}}</code>
$\dot{\dot{a}}$	<code>\dot{\dot{a}}</code>	$\vec{\vec{a}}$	<code>\vec{\vec{a}}</code>

paket `amsmath` die Befehle `\dddot` und `\ddddot` zur Verfügung, die dreifache bzw. vierfache Punktakzente erzeugen (siehe Tabelle 8.17).

Tabelle 8.17: Dreifache bzw. vierfache Punktakzente

Zeichen	Befehl	Zeichen	Befehl
\ddot{a}	<code>\ddot{a}</code>	ä	<code>\text{ä}</code>

8.4.9 Über- und Unterstreichungen

Zum Satz verschiedener Über- und Unterstreichungen existiert eine Reihe von Befehlen, die Tabelle 8.18 zeigt. Überstreichungen realisieren die Befehle `\overline`, `\overbrace`, `\widehat` und `\widetilde`. Für Unterstreichungen stehen `\underline` und `\underbrace` zur Verfügung.

Tabelle 8.18: Über- und Unterstreichungen in Formeln

Zeichen	Befehl	Zeichen	Befehl
\overline{abc}	<code>\overline{abc}</code>	\underline{abc}	<code>\underline{abc}</code>
\overbrace{abc}	<code>\overbrace{abc}</code>	\underbrace{abc}	<code>\underbrace{abc}</code>
\widehat{abc}	<code>\widehat{abc}</code>	\widetilde{abc}	<code>\widetilde{abc}</code>

Bei diesen Befehlen muss der Inhalt, der über- oder unterstrichen werden soll, in geschweiften Klammern als Argument angegeben sein.

Die Breite der Über- und Unterstreichungssymbole legt der \LaTeX -Compiler, abhängig von der Breite der zu über- oder unterstreichenden Inhalte, automatisch fest. Es ist auch möglich, die Befehle aus Tabelle 8.18 zu verschachteln, und so Symbole oder mathematische Inhalte mit mehreren dieser Über- oder Unterstreichungssymbole zu versehen, wie das folgende Beispiel zeigt:

$$\overbrace{\underline{\overbrace{abc\underline{\overbrace{def\underline{\overbrace{gih}}\{jkl\}}}\{}}\{}}\{}$$

Mit Hilfe der Befehle aus Tabelle 8.18 ist es auch möglich, Texte oder mathematische Inhalte über bzw. unter Inhalten zu setzen. Im folgenden Beispiel werden Zeichen *unter* einen Inhalt gesetzt:

$$x_1 * \underbrace{x_2 * x_3}_{2x_2} * \underbrace{x_4 * x_5 * x_6 * x_7}_{2^2 x_4} * \underbrace{x_8 * \dots * x_{15}}_{2^3 x_8} * \dots$$

```

1 \[ x_1 * \underbrace{x_2 * x_3}_{2x_2} *
2   \underbrace{x_4 * x_5 * x_6 * x_7}_{2^2 x_4} *
3   \underbrace{x_8 * \ldots * x_{15}}_{2^3 x_8} * \ldots \]
```


Im folgenden Beispiel wird ein Zeichen *über* einen Inhalt gesetzt:

$$\overbrace{1, x-1, x^2-1, x^3-1}^4$$

`1 \(\overbrace{1, x-1, x^2-1, x^3-1}^4\) \)`

Mit Ausnahme des Befehls `\underline` funktionieren die in diesem Abschnitt vorgestellten Befehle nur innerhalb mathematischer Umgebungen.

8.4.10 Pfeile über und unter Formeln

Das setzen von Pfeilen über oder unter mathematische Inhalte geschieht mit den Befehlen `\overleftarrow` bzw. `\overrightarrow` für Pfeile über Inhalten und `\underleftarrow` bzw. `\underrightarrow` für Pfeile unter Inhalten

`\overleftarrow{Inhalt}` bzw. `\overrightarrow{Inhalt}`
`\underleftarrow{Inhalt}` bzw. `\underrightarrow{Inhalt}`

Die Verschaltung der in diesem Abschnitt vorgestellten Befehle ist problemlos möglich. Die Größe der Pfeile legt der L^AT_EX-Compiler selbstständig fest.

$$\{\varphi(\overrightarrow{X})\}\beta\{\psi(\overrightarrow{XY})\}$$

`1 \(\{\varphi(\overrightarrow{X})\}\beta\{\psi(\overrightarrow{XY})\}\)`
`2 \beta\{\psi(\overrightarrow{XY})\}\)`

$$\overrightarrow{X_\delta a Y_\beta h} = \overrightarrow{X_\delta a Y_\beta h}$$

`1 \(\overrightarrow{\overrightarrow{X_\delta a Y_\beta h}} =`
`2 \underrightarrow{X_\delta a Y_\beta h}\)`

$$\overrightarrow{qX_{\delta\gamma} a Y_{\beta\gamma} h}$$

`1 \(\overrightarrow{\underrightarrow{`
`2 qX_{\overleftarrow{\delta\gamma}} a Y_{\overleftarrow{\beta\gamma}} h}`
`3 \underrightarrow{qX_{\overleftarrow{\delta\gamma}} a Y_{\overleftarrow{\beta\gamma}} h}\)`

8.4.11 Die Größe von Klammern anpassen

Ist eine Klammer, die zum Beispiel einen Bruch oder eine Matrix umschließt, zu klein, kann deren Größe automatisch mit den Befehlen `\left` und `\right` angepasst werden. Diese Befehle werden vor eine öffnende oder schließende Klammer oder vor einen Befehl zum Satz eines anderen Begrenzungssymbols gesetzt.

Die Anpassung kann auch manuell durch ein Voranstellen der Befehle `\big`, `\Big`, `\bigg` und `\Bigg` vorgenommen werden.

Eine Übersicht, wie sich diese Befehle auf die Größe der Klammern auswirken, enthält Tabelle 8.19 und Tabelle 8.20.

Tabelle 8.21 enthält eine Übersicht über die Klammern und Begrenzungssymbole aus den Tabellen 8.19 und 8.20 und die Befehle, mit denen diese erzeugt werden.

Tabelle 8.19: Manuelle Größenanpassung der Klammern (Teil 1)

<code>\big</code>	$()$	$[]$	$\{ \}$	$ $	\parallel	$\lfloor \rfloor$	$\lceil \rceil$	$/$	\backslash	$\langle \rangle$
<code>\Big</code>	$()$	$[]$	$\{ \}$	$ $	\parallel	$\lfloor \rfloor$	$\lceil \rceil$	$/$	\backslash	$\langle \rangle$
<code>\bigg</code>	$()$	$[]$	$\{ \}$	$ $	\parallel	$\lfloor \rfloor$	$\lceil \rceil$	$/$	\backslash	$\langle \rangle$
<code>\Bigg</code>	$()$	$[]$	$\{ \}$	$ $	\parallel	$\lfloor \rfloor$	$\lceil \rceil$	$/$	\backslash	$\langle \rangle$

Tabelle 8.20: Manuelle Größenanpassung der Klammern (Teil 2)

<code>\big</code>	\uparrow	\Uparrow	\downarrow	\Downarrow	\updownarrow	\Updownarrow
<code>\Big</code>	\uparrow	\Uparrow	\downarrow	\Downarrow	\updownarrow	\Updownarrow
<code>\bigg</code>	\uparrow	\Uparrow	\downarrow	\Downarrow	\updownarrow	\Updownarrow
<code>\Bigg</code>	\uparrow	\Uparrow	\downarrow	\Downarrow	\updownarrow	\Updownarrow

8.4.12 Matrizen und Felder

Das Setzen von *Matrizen* geschieht mit der Umgebung `array`. Die Definition der Ausrichtung der einzelnen Spalten geschieht mit Spaltenformatierungseinträgen, analog zur Umgebung `tabular` (siehe Kapitel 6).

```
\begin{array}[Ausrichtung]{Präambel}
\end{array}
```

Das folgende Beispiel zeigt eine Matrix, die mit der Umgebung `array` gesetzt wurde:

Tabelle 8.21: Befehle um unterschiedliche Klammersymbole zu erzeugen

Zeichen	Befehl	Zeichen	Befehl
())	(
[]]	[
{	\{	}	\}
			\
⌊	\lfloor	⌋	\rfloor
⌈	\lceil	⌉	\rceil
/	/	\	\backslash
⟨	\langle	⟩	\rangle
↑	\uparrow	↗	\Uparrow
↓	\downarrow	↘	\Downarrow
↕	\updownarrow	↕	\Updownarrow

$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1x} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2x} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3x} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{y1} & a_{y2} & a_{y3} & \cdots & a_{yx} \end{pmatrix}$	<pre> 1 \left(\begin{array}{ccccc} 2 a_{11} & a_{12} & a_{13} & \cdots & a_{1x} \\ 3 a_{21} & a_{22} & a_{23} & \cdots & a_{2x} \\ 4 a_{31} & a_{32} & a_{33} & \cdots & a_{3x} \\ 5 \vdots & \vdots & \vdots & \ddots & \vdots \\ 6 a_{y1} & a_{y2} & a_{y3} & \cdots & a_{yx} \end{array}\right) 7 \end{array}\right) 8 \end{array}\right) 9 \end{array}\right) </pre>
--	--

Auslassungspunkte werden, wie im Beispiel gezeigt, nicht einzeln und per Hand gesetzt, sondern mit Hilfe geeigneter Befehle realisiert. Eine Übersicht dieser Befehle enthält Tabelle 8.22.

Tabelle 8.22: Befehle um Auslassungspunkte zu setzen

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
...	\ldots	...	\cdots	⋯	\ddots	⋮	\vdots
.	.	:	:	.	\cdot	.	\dot{}

Innerhalb einer **array**-Umgebung können sich weitere **array**-Umgebungen befinden. Die Funktionsweise ist vergleichbar mit geschachtelten Tabellen.

Das folgende Beispiel zeigt zwei geschachtelte **array**-Umgebungen:

$$\left(\begin{array}{c|c} a & 1 \\ b & 2 \\ c & 3 \end{array} \right) \\ x^y \\ \sqrt[y]{x-1}$$

```

1 \ (
2   \left( \begin{array}{c}
3     \left| \begin{array}{cc}
4       a & 1 \\ b & 2 \\ c & 3
5     \end{array} \right| \\
6     x^y \\
7     \sqrt[y]{x-1}
8   \end{array} \right)
9 \ )

```

8.4.13 Mengensymbole

Für Mengensymbole existiert der Befehl `\mathbb` (siehe Tabelle 8.23).

`\mathbb{Zeichen}`

Tabelle 8.23: Befehle um Mengensymbole zu setzen

Zeichen	Bedeutung	Befehl
\mathbb{N}	<i>natürliche Zahlen</i>	<code>\mathbb{N}</code>
\mathbb{Z}	<i>ganze Zahlen</i>	<code>\mathbb{Z}</code>
\mathbb{Q}	<i>rationale Zahlen</i>	<code>\mathbb{Q}</code>
\mathbb{I}	<i>irrationale Zahlen</i>	<code>\mathbb{I}</code>
\mathbb{A}	<i>algebraische Zahlen</i>	<code>\mathbb{A}</code>
\mathbb{R}	<i>reelle Zahlen</i>	<code>\mathbb{R}</code>
\mathbb{C}	<i>komplexe Zahlen</i>	<code>\mathbb{C}</code>

8.4.14 Normaler Text in Formeln

Soll ein „normaler“ Text innerhalb einer mathematischen Umgebung gesetzt werden, muss diese nicht zwingend verlassen werden. Eine einfache Möglichkeit ist die Verwendung des Befehls `\mbox`.

`\mbox{Text}`

Der zu setzende Text wird dem Befehl in geschweiften Klammern übergeben.

Das folgende Beispiel zeigt die Arbeitsweise:

$$M_i \cap M_j = \emptyset \quad \text{falls} \quad i \neq j$$

```

1 \ ( M_i \cap M_j = \emptyset
2   \quad \mbox{falls} \quad i \neq j \ )

```

Wie das Beispiel zeigt, kann es nützlich sein, um die `\mbox` horizontale Abstände einzufügen. Eine Möglichkeit, solche Abstände einzufügen, ist die Verwendung der Befehle `\quad` und `\qquad` (siehe Abschnitt 4.1.2).

`\quad`

Der Befehl `\quad` fügt einen horizontalen Abstand ein, der etwas breiter ist als ein gewöhnlicher Wortabstand, und der Befehl `\qquad` fügt einen Abstand ein, der doppelt so breit ist wie ein von `\quad` erzeugter.

`\qquad`

8.4.15 Mathematische Inhalte einrahmen

Das Einrahmen mathematischer Inhalte kann mit dem Befehl `\boxed` aus dem Erweiterungspaket `amsmath` realisiert werden. Der Befehl zieht einen Rahmen um die Inhalte, die er in geschweiften Klammern als Argument übergeben bekommt.

`\boxed{Inhalte}`

Der Befehl `\boxed` verhält sich im Grunde genau so wie der Befehl `\fbox` (siehe Abschnitt 5.9). Der einzige Unterschied ist, das `\boxed` nur in mathematischen Umgebungen funktioniert.

$$\sqrt[n]{(a * b * c)^2}$$

1 `\(\boxed{\sqrt[\alpha]{(a*b*c)^2}} \)`

Eine weitere Möglichkeit, um Formeln einzurahmen, ist die Verwendung des bereits bekannten Befehls `\fbox`. Dieser Befehl funktioniert zwar nur außerhalb mathematischer Umgebungen, es ist aber möglich, auch eingebettete Formeln damit einzurahmen.

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

1 `\fbox{\(a^{\{2\}} + b^{\{2\}} = c^{\{2\}} \)}`

8.4.16 Ein paar Formeln zum üben...

Dieser Abschnitt enthält einige Formeln zum üben.

$$\frac{3(x+2)^2 - (-2) * 2^{2^2-2}}{3x+6}$$

1 `\(`
2 `\frac{\{3(x+2)^{\{2\}} - (-2) * \{2^{\{2^{\{2\}}-2}\}}\}{3x+6}}`
3 `\)`

$$\frac{-2^{3^2} + (2^3)^2 x}{(2^2)^3}$$

1 `\(`
2 `\frac{-2^{\{3^{\{2\}}\}} + (2^{\{3\}})^{\{2\}} x}{(\{2^{\{2\}}\})^{\{3\}}}`
3 `\)`

$$\sqrt[3]{\left(\frac{x^4-3x^2}{(x-7)}+e^{x^2}\right)^2\sqrt{\sin(x^7-x^3)}}$$

```

1 \ (
2 \sqrt[3]{\Biggl(\frac{
3 \frac{x^4-3x^2}{(x-7)}+e^{x^2}}
4 {\sqrt{\sin(x^7-x^3)}}\Biggr)^2}
5 \ )

```

$$A(x) = \sum_{j=0}^{n-1} b_j y^j$$

```

1 \ (
2 A(x) = \sum_{j=0}^{n-1} b_{j} y^{j}
3 \ )

```

$$\sin z = \sum_{n=0}^{\infty} (-1)^n \frac{z^{2n+1}}{(2n+1)!}$$

```

1 \ (
2 \sin z = \sum_{n=0}^{\infty} (-1)^n
3 \frac{z^{2n+1}}{(2n+1)!}
4 \ )

```

$$\cos z = \sum_{n=0}^{\infty} (-1)^n \frac{z^{2n}}{(2n)!}$$

```

1 \ (
2 \cos z = \sum_{n=0}^{\infty} (-1)^n \frac{z
3 ^{2n}}{(2n)!}
4 \ )

```

$$\sum_{k=1}^n e^{ikx} = e^{ix} * \frac{1 - e^{inx}}{1 - e^{ix}} = \frac{\sin \frac{nx}{2}}{\sin \frac{x}{2}} * e^{i(n+1)\frac{x}{2}} \quad (x \neq 2n\pi)$$

```

1 \ [
2 \sum_{k=1}^n e^{ikx} = e^{ix} * \frac{1-e^{inx}}{1-e^{ix}} =
3 \frac{\sin \frac{nx}{2}}{\sin \frac{x}{2}} * e^{i(n+1)
4 \frac{x}{2}} \enskip (x \neq 2n\pi)
5 \ ]

```

$$\int \frac{ct}{(t^2 + a^2)} dt = \begin{cases} \frac{c}{2} \ln(t^2 + a^2) & n = 1 \\ -\frac{c}{2(n-1)(t^2 + a^2)^{n-1}} & n \geq 0 \end{cases}$$

```

1 \ [
2 \int \frac{ct}{(t^2 + a^2)} dt = \left\{
3 \begin{array}{lr}
4 \frac{c}{2} \ln(t^2 + a^2) & \& n=1 \\
5 - \frac{c}{2(n-1)(t^2 + a^2)^{n-1}} & \& n \geq 0
6 \end{array}
7 \right.
8 \ ]

```

$$\frac{u_{i,j+1}^h - u_{i,j}^h}{\Delta t} = \frac{u_{i+1,j}^h - 2u_{i,j}^h + u_{i-1,j}^h}{(\Delta x)^2}$$

```

1 \ (
2 \frac{u^{\{h\}}_{\{i,j+1\}}-u^{\{h\}}_{\{i,j\}}{\triangle t} =
3 \frac{u^{\{h\}}_{\{i+1,j\}}-2u^{\{h\}}_{\{i,j\}}+u^{\{h\}}_{\{i-1,j\}}{(\triangle x)^2}
4 \ )

```

$$s \in G \implies \langle s \rangle \subseteq G \quad \text{ist Untergruppe} \quad \stackrel{\text{Satz}}{\implies} \quad |\langle s \rangle| \mid |G|$$

```

1 \ [
2 s \in G \Longrightarrow \langle s \rangle \subseteq G
3 G \quad \text{ist Untergruppe} \quad \stackrel{\text{Satz}}{\implies}
4 \{ \Longrightarrow \quad |\langle s \rangle| \mid |G|
5 \ ]

```


9 Briefe setzen

Es existieren mehrere Erweiterungspaket, die alle ein Layout und eine Reihe von Befehlen definieren, um das Setzen von Briefen zu erleichtern.

Die Dokumentklasse, `letter` ist auf den englischen Sprachraum zugeschnitten und wird in diesem Dokument nicht weiter vertieft. Für Briefe nach deutschem Standard sind die Dokumentklasse `dinbrief`, `g-brief` oder `g-brief2` besser geeignet. Eine Alternative KOMA-Klasse ist `scrlettr2`.

In diesem Dokument liegt der Fokus auf je einem funktionierenden Beispiel mit `g-brief2` und `scrlettr2`. Beide Dokumentklassen definieren eine Reihe selbsterklärender Befehle.

9.1 Briefe mit der Dokumentklasse `g-brief2`

Zur Dokumentklasse `g-brief2` existiert vom Autor Michael Lenzen eine gut verständliche Dokumentation [34] in deutscher Sprache.

```
1 \documentclass[11pt,ngerman,a4paper]{g-brief2}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{lmodern}
5
6 \renewcommand\familydefault{\sfdefault}           % Serifenlose Schrift verwenden
7
8 % \fenstermarken
9 \faltmarken
10 \lochermarke
11 % \trennlinien
12
13 \RetourAdresse{Max Mustermann, Musterstraße 1, 12345 Musterstadt}
14
15 \Name{Max Mustermann} % Absender
16 \AdressZeileA{Musterstraße 1}
17 \AdressZeileB{12345 Musterstadt}
18 % \AdressZeileC{}
19 % \AdressZeileD{}
20 % \AdressZeileE{}
21 % \AdressZeileF{}
22
```

```

23 \TelefonZeileA{+49-123-456789}
24 \TelefonZeileB{+49-123-345678}
25 % \TelefonZeileC{}
26 % \TelefonZeileD{}
27 % \TelefonZeileE{}
28 % \TelefonZeileF{}
29
30 \InternetZeileA{max@mustermann.de}
31 \InternetZeileB{http://www.mustermann.com}
32 % \InternetZeileC{}
33 % \InternetZeileD{}
34 % \InternetZeileE{}
35 % \InternetZeileF{}
36
37 \BankZeileA{Musterbank Musterstadt}
38 \BankZeileB{IBAN DE12345678901234567890}
39 \BankZeileC{BIC 12345678901}
40 % \BankZeileD{}
41 % \BankZeileE{}
42 % \BankZeileF{}
43
44 \Unterschrift {Max Mustermann}
45
46 \Postvermerk {} % Einschreiben, Mahnung, etc.
47
48 % Empfänger
49 \Adresse {Erika Musterfrau\\
50 Musterstraße 2\\
51 12345 Musterstadt
52 }
53
54 \Datum{10. Mai 2018}
55 \IhrZeichen{Mu/456}
56 \IhrSchreiben{9. Mai 2018}
57 \MeinZeichen{Mn/123}
58
59 \Betreff{Beispiel für einen Brief mit der Klasse \texttt{g-brief2}}
60
61 \Anrede{Sehr geehrte Damen und Herren,}
62
63 % Der Abstand entspricht dem Absatz zwischen Brieftext und Grussformel
64 \Gruss{Mit freundlichen Grüßen}{0.0cm}
65
66 \Verteiler{}
67
68 \Anlagen{\textbf{Anlagen:}\\
69 Anhang\\
70 Weiterer Anhang}
71
72 \begin{document}
73 \begin{g-brief}
74
75 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
76 tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
77
78 At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd
79 gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
80

```

```
81 Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod  
82 tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At  
83 vero eos et accusam et justo duo dolores et ea rebum.  
84  
85 \end{g-brief}  
86 \end{document}
```

Listing 9.1: Ein Beispiel zur Dokumentklasse `g-brief2`

MAX MUSTERMANN			
Max Mustermann, Musterstraße 1, 12345 Musterstadt			
Erika Musterfrau Musterstraße 2 12345 Musterstadt			
IHR ZEICHEN	IHR SCHREIBEN VOM	MEIN ZEICHEN	DATUM
Mu/456	9. Mai 2018	Mn/123	10. Mai 2018
Beispiel für einen Brief mit der Klasse g-brief2			
Sehr geehrte Damen und Herren,			
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.			
At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.			
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.			
Mit freundlichen Grüßen			
Max Mustermann			
Anlagen: Anhang Weiterer Anhang			
Adresse: Musterstraße 1 12345 Musterstadt	Telefon: +49-123-456789 +49-123-345678	Internet: max@mustermann.de http://www.mustermann.com	Bankverbindung: Musterbank Musterstadt IBAN DE12345678901234567890 BIC 12345678901

Abbildung 9.1: Resultat von Listing 9.1

9.2 Briefe mit der Dokumentklasse `sclttr2`

Eine moderne Alternative zu zur Dokumentklasse `g-brief2` ist die Klasse `sclttr2`. Diese ist Teil des \LaTeX 2_ε-Erweiterungspakets KOMA-Script. Die Klasse bietet sehr vielfältige Optionen zur Briefgestaltung und ist sehr umfangreich auch in deutscher Sprache dokumentiert [30].

```

1 \documentclass[
2   a4paper,
3   fromalign=left, % Absender im Briefkopf links setzen
4   fromlogo=true,  % Logo im Briefkopf setzen
5   fromphone=true,
6   fromfax=true,
7   fromemail=true,
8   fromurl=true,
9   fromrule=false, % Unter den Absender eine Linie setzen
10  locfield=wide,
11  foldmarks=on,    % Faltmarken setzen oder ausblenden
12  parskip=full,    % Zwischen Absätzen eine freie Zeile einfügen
13  version=2.9t     % So ist der Abstand zwischen Fuß und Seitenrand besser
14 ]{sclttr2}
15
16 \usepackage{ngerman}
17 \usepackage[utf8]{inputenc}
18 \usepackage[T1]{fontenc}
19 \usepackage{graphicx}
20 \usepackage{url}
21 \usepackage{lmodern}
22
23 \renewcommand\familydefault{\sfdefault} % Serifenlose Schrift verwenden
24 \renewcommand*{\raggedsignature}{\raggedright} % Linksbündige Unterschrift
25 \renewcaptionname{ngerman}{\enclname}{Anlagen} % Anstatt "Anlage(n)"...
26
27 \begin{document}
28
29 \setkomavar{fromname}{Max Mustermann}
30 \setkomavar{fromaddress}{Musterstraße 1\\ 12345 Musterstadt}
31 \setkomavar{fromphone}{+49-123-456789}
32 \setkomavar{fromfax}{+49-123-987654}
33 \setkomavar{fromemail}{\url{max@mustermann.de}}
34 \setkomavar{fromurl}{\url{http://www.mustermann.com}}
35
36 \setkomavar{fromlogo}{\includegraphics[width=.2\linewidth]{Tux.pdf}}
37
38 \setkomavar{signature}{Max Mustermann} % Standardmäßig wird "fromname" verwendet
39
40 \setkomavar{customer}{Kundennummer}
41 \setkomavar{invoice}{Rechnungsnummer}
42 \setkomavar{myref}{Mn/123}
43 \setkomavar{place}{Musterstadt}
44 \setkomavar{yourmail}{9. Mai 2018}
45 \setkomavar{yourref}{Mu/456}
46
47 \setkomavar{date}{11. Mai 2018}
48
49 \setkomavar{frombank}{Musterbank Musterstadt\\

```

```

50 |                                     IBAN DE12345678901234567890\\
51 |                                     BIC 12345678901}
52 |
53 | \firstfoot{
54 | \footnotesize
55 | \begin{tabular}[t]{l}
56 | \textbf{Geschäftsführer:} \\
57 | Tom Mustermann \\
58 | Anika Mustermann
59 | \end{tabular}
60 | \hfill
61 | \begin{tabular}[t]{l}
62 | \textbf{Gerichtsstand:} \\
63 | Musterstadt \\
64 | \textbf{USt-ID:} \\
65 | DE-1234567890
66 | \end{tabular}
67 | \hfill
68 | \begin{tabular}[t]{l}
69 | \textbf{\usekomavar*{frombank}:} \\
70 | \usekomavar{frombank}
71 | \end{tabular}
72 | }
73 |
74 | \setkomavar{specialmail}{} % Einschreiben, Mahnung, etc.
75 |
76 | \setkomavar{title}{} % Einschreiben, Mahnung, etc.
77 |
78 | \setkomavar{subject}{Beispiel für einen Brief mit der Klasse \texttt{sclttr2}}
79 |
80 |
81 | \begin{letter}{Erika Musterfrau \\
82 |               Musterstraße 2 \\
83 |               12345 Musterstadt}
84 |
85 | \opening{Sehr geehrte Damen und Herren,}
86 |
87 | Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
88 | tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
89 |
90 | At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd
91 | gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
92 |
93 | Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
94 | tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At
95 | vero eos et accusam et justo duo dolores et ea rebum.
96 |
97 | \closing{Mit freundlichen Grüßen}
98 |
99 | \encl{Anhang\Weiterer Anhang}
100 |
101 | % \cc{Verteiler} % Verteiler
102 |
103 | \end{letter}
104 | \end{document}

```

Listing 9.2: Ein Beispiel zur Dokumentklasse `sclttr2`

Max Mustermann
 Musterstraße 1
 12345 Musterstadt
 Telefon: +49-123-456789
 Fax: +49-123-987654
 E-Mail: max@mustermann.de
 URL: <http://www.mustermann.com>
 Max Mustermann, Musterstraße 1, 12345 Musterstadt



Erika Musterfrau
 Musterstraße 2
 12345 Musterstadt

Ihr Zeichen	Ihr Schreiben vom	Unser Zeichen	Kundennummer	Rechnungsnummer	Datum
Mu/456	9. Mai 2018	Mn/123	Kundennummer	Rechnungsnummer	11. Mai 2018

Beispiel für einen Brief mit der Klasse `scrlttr2`

Sehr geehrte Damen und Herren,

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Mit freundlichen Grüßen

Max Mustermann

Anlagen: Anhang
 Weiterer Anhang

Geschäftsführer:
 Tom Mustermann
 Anika Mustermann

Gerichtsstand:
 Musterstadt
USt-ID:
 DE-1234567890

Bankverbindung:
 Musterbank Musterstadt
 IBAN DE12345678901234567890
 BIC 12345678901

Abbildung 9.2: Resultat von Listing 9.2

10 Sonderschriften

Standardmäßig verwendet L^AT_EX die Schrift Computer Modern. Diese verleiht Dokumenten einen gewissen „L^AT_EX-Look“. Die Verwendung anderer Schriften, auch ungewöhnlicher Schriften, ist auf verschiedene Art und Weise möglich.

Eine hilfreiche Übersicht über freie Schriften und Beschreibungen, wie diese mit L^AT_EX verwendet werden können, bietet der L^AT_EX Font Catalogue [35].

10.1 Altdeutsche Schriften

Dank der Arbeit von Yannis Haralambous [36] ist es auf recht einfache Art und Weise möglich, altdeutsche Schriften in L^AT_EX-Dokumenten zu verwenden.

Eine einfache Möglichkeit, um altdeutsche Schriften in L^AT_EX-Dokumenten zu verwenden, Erweiterungspaket `yfonts`. Dieses muss mit dem Befehl `\usepackage{yfonts}` in der Präambel der L^AT_EX-Quelldatei eingebunden sein. Es definiert drei Befehle `\textgoth` für **Gothische Schrift**, `\textswab` für **Schwabacher Schrift** und `\textfrak` für **Fraktur**.

```
\textgoth{Text}  
\textswab{Text}  
\textfrak{Text}
```

10.2 Barcodes

Es existieren einige Erweiterungspakete, um Barcodes mit L^AT_EX komfortabel zu setzen.

Ein Beispiel ist das Erweiterungspaket `makebarcode` [37], das den Befehl `\barcode` definiert.

```
\barcode{Text}
```

Der folgende Befehl in der Präambel der L^AT_EX-Quelldatei bindet das Erweiterungspaket ein und definiert die Kodierung Code 39, eine Zeichenhöhe (Parameter `H`) von

1 cm, das dünne Linien 0,5 mm breit (Parameter `X`) sind und das dicke Linien 2,5 mal so breit sind wie dünne Linien (Parameter `ratio`).

```
\usepackage[code=Code39,X=.5mm,ratio=2.5,H=1cm]{makebarcode}
```

In der Praxis sehen einzelne Zeichen dann wie in Tabelle 10.1 aus.

Tabelle 10.1: Einzelne Zeichen, als Barcodes gesetzt

0	1	2	3	4
5	6	7	8	9
+	-	/	.	A
B	C	D	E	F
G	H	I	J	K
L	M	N	O	P
Q	R	S	T	U
V	W	X	Y	Z

10.3 QR-Codes

Bei QR-Codes¹ handelt es sich um zweidimensionale Codes. Ein QR-Code besteht aus einer quadratischen Matrix aus schwarzen und weißen Quadraten, die die kodierten Daten binär darstellen. Jeder QR-Code codiert eine Zeichenkette. Die Daten im QR-Code sind durch einen fehlerkorrigierenden Code geschützt. Dadurch wird der Verlust von bis zu 30% des Codes toleriert. Der maximale Informationsgehalt eines QR-Codes (177x177 Elemente) beträgt ca. 3 kB). [38] Mobiltelefone verfügen in der Regel über eine Software, die in der Lage ist, QR-Codes zu erkennen und zu dekodieren. Eine typische Anwendung ist es, Internetadressen als QR-Codes bereitzustellen, um diese einfach und ohne manuelle Eingabe an ein mobiles Gerät zu übertragen.

Eine Möglichkeit, QR-Codes mit \LaTeX zu erstellen bietet das Erweiterungspaket `qrcode`. Dieses Paket bietet zahlreiche Möglichkeiten, Informationen als QR-Codes zu kodieren, und es definiert hierzu den Befehl `\qrcode`.

```
\qrcode{Optionen}{Text}
```

Von den möglichen Optionen des Befehls ist an dieser Stelle nur `height` vorgestellt. Hier ist die Höhe (und Breite) des zu erzeugenden QR-Codes inklusive einer Maßeinheit (siehe Abschnitt ??) definiert. Um beispielsweise einen QR-Code mit der URL zu diesem Werk zu erzeugen, der 3 cm hoch und breit ist, genügt der folgende Befehl:

```
1 \qrcode[height=3cm]{https://github.com/christianbaun/einstieginlatex}
```

Das Ergebnis sieht wie folgt aus:






























































































































































































10.4 Besondere Zeichen

Einen Zeichensatz, der statt Buchstaben, Ziffern und Satzzeichen mit Sonderzeichen aller Art gefüllt ist, ist *ZapfDingbats* (siehe Tabelle 10.2). Dessen Verwendung ist mit dem Erweiterungspaket `pifont.sty` besonders einfach.

¹Das QR steht für Quick Response


Tabelle 10.2: Zeichen aus dem Font ZapfDingbats

	'0	'1	'2	'3	'4	'5	'6	'7
'04x		 33	 34	 35	 36	 37	 38	 39
'05x	 40	 41	 42	 43	 44	 45	 46	 47
'06x	 48	 49	 50	 51	 52	 53	 54	 55
'07x	 56	 57	 58	 59	 60	 61	 62	 63
'10x	 64	 65	 66	 67	 68	 69	 70	 71
'11x	 72	 73	 74	 75	 76	 77	 78	 79
'12x	 80	 81	 82	 83	 84	 85	 86	 87
'13x	 88	 89	 90	 91	 92	 93	 94	 95
'14x	 96	 97	 98	 99	 100	 101	 102	 103
'15x	 104	 105	 106	 107	 108	 109	 110	 111
'16x	 112	 113	 114	 115	 116	 117	 118	 119
'17x	 120	 121	 122	 123	 124	 125	 126	
'26x		 161	 162	 163	 164	 165	 166	 167
'27x	 168	 169	 170	 171	 172	 173	 174	 175
'28x	 176	 177	 178	 179	 180	 181	 182	 183
'29x	 184	 185	 186	 187	 188	 189	 190	 191
'30x	 192	 193	 194	 195	 196	 197	 198	 199
'31x	 200	 201	 202	 203	 204	 205	 206	 207
'32x	 208	 209	 210	 211	 212	 213	 214	 215
'33x	 216	 217	 218	 219	 220	 221	 222	 223
'34x	 224	 225	 226	 227	 228	 229	 230	 231
'35x	 232	 233	 234	 235	 236	 237	 238	 239
'36x		 241	 242	 243	 244	 245	 246	 247
'37x	 248	 249	 250	 251	 252	 253	 254	

Das Einbinden des Erweiterungspakets in eigene Dokumente geschieht mit dem Befehl `\usepackage{pifont}` in die Präambel des Dokuments.

Ein Befehl, den das Erweiterungspaket definiert, ist `\ding`. Mit diesem Befehl werden einzelne Zeichen aus dem Font *ZapfDingbats* ausgegeben.

`\ding{Zeichen}`

Dem Befehl wird die Nummer (siehe Tabelle 10.2) des Zeichens, das gesetzt werden soll, in geschweiften Klammern als Argument übergeben. Die Schneeflocke  mit der Zeichennummer 100 beispielsweise setzt der Befehl `\ding{100}`.

Alternativ kann auf die einzelnen Zeichen auch durch Angabe der Spalten Kombination von Zeilen- und Spaltennummer angegeben werden. Die In diesem Fall wird ein einzelnes Anführungszeichen der Zeichennummer vorgestellt. Die bereits beschriebene Schneeflocke befindet sich in Zeile 14 und Spalte 4. Das heißt, sie kann auch mit Hilfe dieses Befehls gesetzt werden: `\ding{'144}`

Zwei weitere Befehle, die das Erweiterungspaket `pifont` definiert, sind `\dingline` und `\dingfill`.

```
\dingline{Zeichen}
\dingfill{Zeichen}
```

Wird der Befehl `\dingline` ausgeführt, beginnt L^AT_EX eine neue Zeile, die mit dem als Argument übergeben Zeichen gefüllt wird. Die beidseitige Einrückung entspricht 0.5 Zoll (also ca. 1,3 cm). Ein sinnvolles Anwendungsbeispiel für diesen Befehl sind Formulare, wo ein Teil des Blattes abgetrennt werden soll, beispielsweise so:

✂ ✂

Die Zeile mit den Scheren wurde mit dem Befehl `\dingline{34}` erzeugt.

Im Gegensatz zu `\dingline` wird bei `\dingfill` nur der in einer Zeile verfügbare Leerraum mit dem als Argument übergeben Zeichen gefüllt. So erzeugt der Befehl `\dingfill{51}` hier: ✓

Der Befehl `\dingfill` füllt immer bis zum Zeilenende. Es sei denn, es kommt noch ein Text nach dem Befehl. Der Befehl `\dingfill{51}` gefolgt von der Zeichenkette `Ende.` erzeugt das folgende Ergebnis: ✓ Ende.

Außer den bereits vorgestellten Befehlen definiert das Erweiterungspaket `pifont` auch die Umgebung `dinglist`. Diese ermöglicht es Aufzählungen (siehe Abschnitt 5.4) ähnlich wie `itemize`.

```
\begin{dinglist}{Zeichen} ... \end{dinglist}
```

Die einzelnen Listenelemente beginnen wie gehabt mit dem Befehl `\item`.

Bei der Umgebung `dinglist` wird die Nummer (siehe Tabelle 10.2) des Zeichens, das als Markierungszeichen für jeden Auflistungspunkt verwendet werden soll, in geschweiften Klammern als Argument übergeben.

```
\begin{dinglist}{52}
  \item Erster Listenzpunkt
  \item Zweiter Listenzpunkt
  \begin{dinglist}{234}
    \item Erster Unterpunkt
    \item Zweiter Unterpunkt
  \end{dinglist}
  \item Dritter Listenzpunkt
\end{dinglist}
```

- ✓ Erster Listenzpunkt
- ✓ Zweiter Listenzpunkt
- ⇔ Erster Unterpunkt
- ⇔ Zweiter Unterpunkt
- ✓ Dritter Listenzpunkt

Neben der Umgebung `dinglist`, die ein Äquivalent zur Umgebung `itemize` ist, definiert das Erweiterungspaket `pifont` mit der Umgebung `dingautolist` auch ein Äquivalent zur Umgebung `enumerate`.

```
\begin{dingautolist}{Zeichen} ... \end{dingautolist}
```

Im Gegensatz zur `dinglist` wird bei der Umgebung `dingautolist` mit jedem weiteren Aufzählungspunkt (`\item`) die Nummer des verwendeten Zeichens um eins erhöht. Konkret wird in der Zeichentabelle (siehe Tabelle 10.2) immer ein Zeichen weiter gegangen. So hat jeder Aufzählungspunkt einer Ebene ein anderes Markierungszeichen.

```
\begin{dingautolist}{172}
  \item Erster Aufzählungspunkt
  \item Zweiter Aufzählungspunkt
  \begin{dingautolist}{182}
    \item Erster Unterpunkt
    \item Zweiter Unterpunkt
  \end{dingautolist}
  \item Dritter Aufzählungspunkt
\end{dingautolist}
```

- ① Erster Aufzählungspunkt
- ② Zweiter Aufzählungspunkt
- ① Erster Unterpunkt
- ② Zweiter Unterpunkt
- ③ Dritter Aufzählungspunkt

10.5 Kalligrafie

Das Erweiterungspaket `calligra` [39] bietet definiert den Befehl `\textcalligra` und bietet dadurch eine einfache Möglichkeit, Texte in einem Kalligrafie-Font zu setzen.

```
\textcalligra{Text}
```

```
\textcalligra{Lorem ipsum dolor sit  
amet, consetetur sadipscing elitr,  
sed diam nonumy eirmod tempor  
invidunt ut labore et dolore magna  
aliquyam erat, sed diam voluptua.  
At vero eos et accusam et justo duo  
dolores et ea rebum.}
```

```
\textcalligra{0123456789}
```

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

0123456789

11 Literaturverzeichnis

Das Einbinden eines Literaturverzeichnisses in L^AT_EX-Dokumente kann auf verschiedene Arten geschehen. Diese unterscheiden sich in erster Linie hinsichtlich des Automatisierungsgrads. In diesem Dokument ist das Erstellen von Literaturverzeichnissen mit Hilfe des Abschnitts `thebibliography` und mit Hilfe von BibT_EX beschrieben.

Ganz unabhängig von der Art und Weise, wie das Literaturverzeichnis realisiert wird, geschieht der Verweis auf Einträge im Literaturverzeichnis mit dem Befehl `\cite`.

```
\cite[Text]{Schlüssel}
```

Der zwingend nötige Parameter *Schlüssel* verweist auf einen oder mehrere Einträge im Literaturverzeichnis. Der Inhalt des optionalen Parameters *Text* wird zusätzlich Ausgegeben und kann eine weiterführende Information für den Leser enthalten wie zum Beispiel „S. 42-46“ oder „Kapitel 5“.

Soll auf mehrere Einträge im Literaturverzeichnis gleichzeitig referenziert werden, müssen die Angaben im Parameter *Schlüssel* durch Kommata voneinander getrennt sein.

11.1 Manueller Satz des Literaturverzeichnisses

Das Einfügen des Inhalts des Literaturverzeichnisses geschieht in einer Umgebung `thebibliography` am Ende der Datei mit dem L^AT_EX-Quelltext.

```
\begin{thebibliography}{Markierungsbreite} ... \end{thebibliography}
```

Die Breite (Anzahl der Zeichen) der einzelnen Labels (Markierungen) im Literaturverzeichniss definiert der Inhalt des Parameters *Markierungsbreite*. Dieser kann eine beliebige Zeichenkette enthalten (z.B. 999) [40].

Die einzelnen Einträge im Literaturverzeichnis realisiert der Befehl `\bibitem` innerhalb der Umgebung `thebibliography`.

```
\bibitem[Markierung]{Schlüssel} Text
```

Ein Beispiel für einen `thebibliography`-Umgebung mit drei Einträgen enthält Listing 11.1 und Abbildung 11.1 das resultierende Literaturverzeichnis.

```

1 \begin{thebibliography}{9}
2 \bibitem[Go93]{GoMiSa93}
3 Michel Goossens, Frank Mittelbach and Alexander Samarin.
4 \emph{The \LaTeX\ Companion}.
5 Addison-Wesley, Reading, Massachusetts, 1993.
6 \bibitem[La94]{Lamport94}
7 Leslie Lamport, \emph{\LaTeX: A Document Preparation System},
8 Addison Wesley, Massachusetts, 2nd edition, 1994.
9 \bibitem[Kn86]{Knuth96}
10 Donald E. Knuth, \emph{Computers and Typesetting -- Volume A:
11 The \TeX book}, Addison Wesley, 1986.
12 \end{thebibliography}

```

Listing 11.1: Das allererste Beispiel

Literatur

- [Go93] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The \LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [La94] Leslie Lamport, *\LaTeX : A Document Preparation System*, Addison Wesley, Massachusetts, 2nd edition, 1994.
- [Kn86] Donald E. Knuth, *Computers and Typesetting – Volume A: The \TeX book*, Addison Wesley, 1986.

Abbildung 11.1: Resultat von Listing 11.1

Ein Unterschied zum Satz des Literaturverzeichnisses mit Bib \TeX ist das Literaturverzeichnis im fertig gesetzten Dokument alle im Quelltext aufgeführten Einträge enthält. Es spielt keine Rolle, ob auf einzelne Einträge referenziert ist.

11.2 Automatisierter Satz des Literaturverzeichnisses

Vielfältige Möglichkeiten ein Literaturverzeichniss einzufügen und dessen Formatierung zu beeinflussen bietet \LaTeX in Zusammenarbeit mit dem Programm Bib \TeX und den Befehlen `\bibliography` und `\bibliographystyle`. Die Kombination aus diesen Programmen und Befehlen ermöglicht es, Quellen aus einer oder mehreren Literaturdatenbanken herauszusuchen und beispielsweise nach dem Nachnamen des ersten Autors oder gemäß der Reihenfolge der Referenzierung im Dokument zu sortieren.

Die Einbindung einer Literaturdatenbank (`.bib`-Datei) geschieht mit dem Befehl `\bibliography`. Der Dateiname wird ohne Dateiendung angegeben.

```
\bibliography{Dateiname}
```

Sollen mehrere Literaturdatenbanken eingebunden werden, müssen die entsprechenden `.bib`-Dateien durch Kommata voneinander abgetrennt sein.

```
\bibliography{Dateiname1,Dateiname2,Dateiname3,...}
```

Die Definition des Stils des Literaturverzeichnisses geschieht mit dem Befehl `\bibliographystyle`.

```
\bibliographystyle{Stil}
```

Der Befehl darf an (fast) beliebiger Stelle in der `.tex`-Quelldatei stehen.

Es existiert eine sehr große Anzahl an Stilen. Umfangreiche Übersichten enthalten [40, 41] Eine Auswahl an häufig verwendeten Stilen enthält Tabelle 11.1.

Tabelle 11.1: Auswahl an Stilen für Literaturverzeichnisse [42]

Stilname	Sortierung	Referenzierung
abbrv	Erstautor	[IdNr] (Vornamen nur als Initialen)
alpha	Erstautor	bis zu drei Anfangsbuchstaben der Autorennamen, evtl. ein hochgestelltes Pluszeichen, bei mehr als drei Autoren, Jahresangabe zweistellig
apalike	Erstautor	[Autorenname(n), Jahr]
plain	Erstautor	[IdNr] (Vornamen werden ausgeschrieben)
plainyr	Referenzart, Jahr	[IdNr] (Vornamen werden ausgeschrieben)
unsrt	Reihenfolge der Referenzierung	[IdNr]

Für jeden Stil muss eine so Datei gleichen Namens mit der Dateiondung `.bst` existieren. Dort ist das Layout des Stils definiert. Bei `TEX Live` befinden sich die `.bst`-Dateien im Unterverzeichnis `texlive/texmf-dist/bibtex/bst/`.

Bei der Arbeit mit `BibTEX` müssen die einzelnen Literatureinträge in einer oder mehreren Dateien mit der Dateiondung `.bib` vorliegen. Je nach Art des jeweiligen Eintrags (zum Beispiel Bücher, Artikel, technische Dokumentationen, etc.) sind bestimmte Angaben („Felder“) zum Eintrag zwingend nötig oder optional möglich. Häufig verwendete Typen von Einträgen in Literaturdatenbanken sind `article`, `book`, `booklet`, `conference`, `inbook`, `incollection`, `inproceedings`, `manual`, `mastersthesis`, `misc`, `phdthesis`, `proceedings`, `techreport` und `unpublished`.

Jedem Eintrag ist ein Klammeraffe (`@`) vorangestellt. Zudem ist es egal, ob das Schlüsselwort eines Eintrags mit Klein- oder Großbuchstaben geschrieben ist. Jede der folgenden Schreibweisen ist somit korrekt: `@book`, `@BOOK`, `@book`, `@BoOk`, etc.

Auf den Typ eines Eintrags folgt immer ein Klammersn paar aus geschweiften Klammern, und zwischen diesen Klammern befinden sich ein Schlüsselwort und die einzelnen Felder. Das Schlüsselwort kann aus einer fast beliebigen Folge und Anordnung von Buchstaben, Zahlen und Zeichen jeder Art bestehen. Kommata sind im Schlüsselwort tabu. Idealerweise ist das Schlüsselwort eindeutig und nachvollziehbar.

Die einzelnen Felder bestehen aus jeweils einem Feldnamen und einer Information (Feldtext). Mögliche Felder sind u.a. `address`, `author`, `booktitle`, `chapter`, `edition`, `editor`, `isbn`, `journal`, `volume`, `month`, `note`, `number`, `organization`, `pages`, `series`, `school`, `title` und `year`. Es werden allerdings nicht alle Felder bei allen Arten von Einträgen berücksichtigt und bei allen Arten von Einträgen sind bestimmten Felder zwingend erforderlich. Eine Übersicht über diese Typen von Einträgen in Literaturdatenbanken und deren zwingende sowie optionale Felder enthält 11.4

Zwischen Feldname und Feldtext befindet sich ein `=`. Der Feldtext ist von geschweiften Klammern oder alternativ mit Anführungsstrichen (`" ... "`) umschlossen. Besteht der Feldtext nur aus Ziffern, sind die geschweiften Klammern bzw. Anführungsstriche nicht nötig. Die einzelnen Felder sind durch Kommata voneinander getrennt.

Syntaktisch korrekt sehen Einträge in Literaturdatenbanken folgendermaßen aus:

```
@Typ{Schlüsselwort,
    Feldname = {Feldtext},
    Feldname = {Feldtext},
    Feldname = {Feldtext},
    ...
    Feldname = {Feldtext}
}
```

Ein paar Beispiele für Bib_{TEX}-Einträge enthält Listing 11.2.

```
1 @article{InformatikSpektrum2018,
2   author    = {Christian Baun and
3               Henry-Norbert Cocos and
4               Rosa-Maria Spanou},
5   title     = {Erfahrungen beim Aufbau von großen Clustern aus
6               Einplatinencomputern für Forschung und Lehre},
7   journal   = {Informatik Spektrum},
8   volume    = {41},
9   number    = {3},
10  pages     = {189--199},
11  year      = {2018},
12  doi       = {10.1007/s00287-017-1083-9}
13 }
14
15 @book{ComputernetzeKompaktBuch2018,
16  author    = {Christian Baun},
17  title     = {Computernetze kompakt},
```

```

18 publisher = {Springer Vieweg},
19 edition   = {4},
20 year       = {2018},
21 isbn       = {978-3-662-57468-3}
22 }

```

Listing 11.2: Einige Beispiele für Bib_TE_X-Einträge

In den meisten Fällen ist die effizienteste Vorgehensweise fertige Bib_TE_X-Einträge zu den gewünschten Quellen von Online-Angeboten wie Google Scholar [43] oder dblp [44] zu beziehen.

Weitere Möglichkeiten, um das Aussehen des Literaturverzeichnisses und der Referenzen zu definieren, bietet das Erweiterungspaket `natbib` [45]. Diese bietet auch weitere Felder wie zum Beispiel `doi`, `issn` und `url`.

11.3 Funktionsweise von Bib_TE_X

Beim Durchlauf des L^AT_EX-Compilers erzeugt dieser eine Textdatei mit der Endung `.aux`. Ein nachfolgender Durchlauf von Bib_TE_X erzeugt aus der `.aux`-Datei mit Hilfe der `.bib`-Datei(en) und der `.bst`-Datei eine weitere Textdatei mit der Endung `.bbl`. Diese letzte Datei enthält das mit `thebibliography` realisierte Literaturverzeichnis mit den im Dokument angeforderten Einträgen. Beim nächsten Durchlauf des L^AT_EX-Compilers fügt dieser das Literaturverzeichnis in das fertige Dokument ein.

11.4 Einträgen in Literaturdatenbanken und deren Felder

Dieser Abschnitt enthält eine Übersicht über einige gängige Typen von Einträgen in Literaturdatenbanken und deren zwingende sowie optionale Felder [46, 47].

@article Literaturangabe für einen Artikel aus einer Zeitschrift, oder Zeitung.

<i>zwingende Felder</i>	<code>author, title, journal, year</code>
<i>optionale Felder</i>	<code>volume, number, month, note, pages</code>

@book Literaturangabe für ein Buch aus einem Verlag.

<i>zwingende Felder</i>	<code>author oder editor, title, publisher, year</code>
<i>optionale Felder</i>	<code>volume oder number, series, address, edition, month, note, isbn</code>

13611.4. EINTRÄGEN IN LITERATURDATENBANKEN UND DEREN FELDER

@booklet Literaturangabe für ein Buch (gebundenes Druckwerk) ohne Verlag.

zwingende Felder **title**

optionale Felder **author, howpublished, address, month, year, note**

@conference identisch mit **@inproceedings**.

@inbook Literaturangabe für einen Buchauszug, z. B. ein Kapitel oder ein paar Seiten.

zwingende Felder **author oder editor, title, chapter und/oder pages, publisher, year**

optionale Felder **volume oder number, series, type, address, edition, month, note**

@incollection Literaturangabe für einen Buchauszug mit einem eigenen Titel, z. B. einen Aufsatz in einem Sammelband.

zwingende Felder **author, title, booktitle, publisher, year**

optionale Felder **editor, volume oder number, type, series, edition, chapter, pages, address, month, note**

@inproceedings Literaturangabe für einen Artikel aus einem Konferenzbericht.

zwingende Felder **author, title, booktitle, year**

optionale Felder **editor, volume oder number, organisation, series, pages, publisher, address, month, note**

@manual Literaturangabe für eine technische Dokumentation.

zwingende Felder **title, address, year**

optionale Felder **author, organisation, edition, month, note**

@mastersthesis Literaturangabe für eine Diplomarbeit oder Masterthesis.

zwingende Felder **author, title, school, year**

optionale Felder **address, month, note, type**

@misc Literaturangabe die zu keiner dieser Eingabetypen passt (z. B. eine Web-Seite).

zwingende Felder **mindestes eines der *optionalen Felder***

optionale Felder **author, title, howpublished, month, year, note**

@phdthesis Literaturangabe für eine Doktorarbeit (Promotion).

zwingende Felder **author, title, school, year**

optionale Felder **address, month, note, type**

@proceedings Literaturangabe für einen Konferenzbericht.

zwingende Felder **title, year**

optionale Felder **editor, publisher, volume oder number, organisation, series, address, month, note**

@techreport Literaturangabe für einen Bericht einer Hochschule oder eines Instituts.

zwingende Felder **author, title, institution, year**
optionale Felder **type, number, address, month, note**

@unpublished Literaturangabe für eine unveröffentlichte Arbeit mit einem Autor und einem Titel.

zwingende Felder **author, title, note**
optionale Felder **month, year**

11.5 Das Aussehen des Literaturverzeichnisses anpassen

Die Änderung der Überschrift eines Literaturverzeichnisses geschieht mit folgendem Befehl:

```
\renewcommand{\refname}{Literaturverzeichnis}
```

Einige Autoren bevorzugen anstatt „Literaturverzeichnis“ beispielsweise die Überschrift „Quelle“ oder einfach „Literatur“.

Das Hinzufügen eines Eintrags in das Inhaltsverzeichnis realisiert der folgende Befehl:

```
\addcontentsline{toc}{chapter}{Literaturverzeichnis}
```

11.6 Die Literaturdatenbank mit der L^AT_EX-Quelldatei erzeugen

Mit dem Erweiterungspaket **filecontents** ist es möglich aus einer L^AT_EX-Quelldatei eine andere Datei zu erstellen und mit Inhalt zu befüllen bzw. deren Inhalt zu überschreiben. Sobald das Paket mit dem Befehl **\usepackage{textcomp}** in der Präambel eines L^AT_EX-Quelltextes eingebunden wurde, ist die gleichnamige Umgebung **filecontents** verfügbar.

Damit ist es z. B. möglich eine BibT_EX-Literaturdatenbank aus der L^AT_EX-Quelldatei heraus zu erzeugen. Listing 11.3 zeigt exemplarisch die Vorgehensweise. Das Ergebnis des Beispiels zeigt Abbildung 11.2.

Beim Start der Umgebung **filecontents** wird in geschweiften Klammern als Parameter der Dateiname der zu erzeugenden bzw. zu überschreibenden Datei definiert. Der in Listing 11.3 verwendete Befehl **\jobname** fügt an der Stelle seines Aufrufs den Namen der L^AT_EX-Quelldatei ohne die Dateiendung ein. Auf diese Art und Weise wird eine **.bib**-Datei mit dem gleichen Dateinamen wie die L^AT_EX-Quelldatei erzeugt.

```

1 \documentclass[12pt,a4paper,oneside,ngerman]{scrartcl}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{url}
5
6 \usepackage{filecontents}
7
8 \begin{filecontents}{\jobname.bib}
9 @book{BetriebssystemeKompaktBuch2017,
10     author    = {Christian Baun},
11     title     = {{Betriebssysteme kompakt}},
12     publisher = {Springer Vieweg},
13     note      = {2.~Auflage},
14     year      = 2017,
15     month     = {April},
16     isbn      = {978-3-662-53142-6}
17 }
18
19 @unpublished{baun2018,
20     author    = "Christian Baun",
21     title     = {{Ein Einstieg in LaTeX}},
22     note      = "\url{https://github.com/christianbaun/einstieginlatex}",
23     year      = 2018
24 }
25 \end{filecontents}
26
27 \begin{document}
28 Hier ist eine Quelle~\cite{baun2018}.
29 Hier ist noch eine weitere Quelle~\cite{BetriebssystemeKompaktBuch2017}.
30
31 \bibliographystyle{unsrt}
32 \bibliography{\jobname}
33 \end{document}

```

Listing 11.3: Eine BibT_EX-Literaturdatenbank aus der L^AT_EX-Quelldatei heraus erzeugen

Hier ist eine Quelle [1]. Hier ist noch eine weitere Quelle [2].

References

- [1] Christian Baun. Ein Einstieg in LaTeX. <https://github.com/christianbaun/einstieginlatex>, 2018.
- [2] Christian Baun. *Betriebssysteme kompakt*. Springer Vieweg, April 2017. 2. Auflage.

Abbildung 11.2: Resultat von Listing 11.3

12 Präsentationsfolien

Die Verwendung der Dokumentklasse **beamer** ermöglicht auf komfortable Art und Weise die Erstellung optisch ansprechender Präsentationsfolien mit \LaTeX .

Ein einfaches Beispiel für ein Grundgerüst einer \LaTeX -Quelldatei mit **beamer** zeigt Listing 12.1. Das Ergebnis dieses Beispiels zeigt Abbildung 12.1.

```
1 \documentclass{beamer}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{ngerman}
5 \usepackage{lmodern}           % Die Latin Modern Schriften sind besser für PDFs
6
7 \usetheme{Dresden}
8 \usecolortheme{default}
9
10 \begin{document}
11
12 \section{Ein Abschnitt}
13
14 \begin{frame}
15 \frametitle{Eine Überschrift}
16
17 Hier steht der Inhalt dieser Folie
18
19 \end{frame}
20 \end{document}
```

Listing 12.1: Einfaches Grundgerüst einer \LaTeX -Quelldatei mit **beamer**

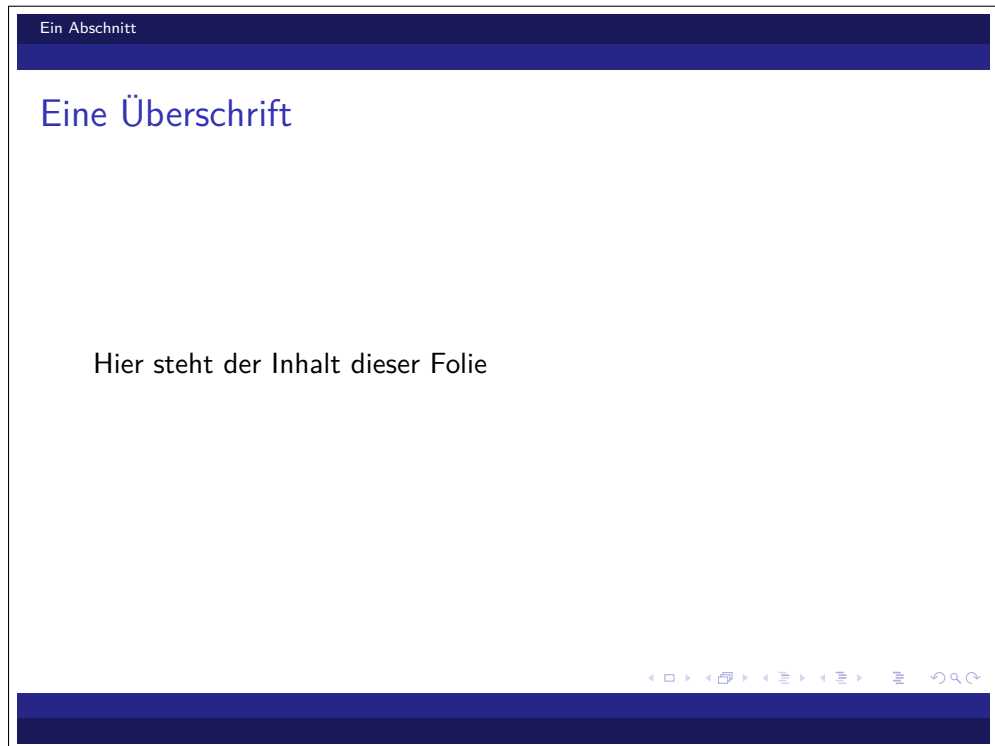


Abbildung 12.1: Resultat von Listing 12.1

Das grundlegende Layout der Präsentationsfolien definieren sogenannte Themen¹. In jedem Foliensatz muss exakt ein Thema definiert sein. Die farbliche Gestaltung kann mit Hilfe von Farbthemen² beeinflusst werden. Eine hilfreiche Übersicht über die verschiedenen Kombinationen aus standardmäßig enthaltenen (Layout-)Themen und Farbthemen bietet [48].

Jede Folie, die mit **beamer** erstellt wird, besteht aus einer Umgebung **frame**. Innerhalb der Umgebung kann Folientitel mit dem Befehl **frametitle** definiert sein, indem er dem Befehl in geschweiften Klammern als Argument übergeben wird.

```
\begin{frame}
  \frametitle{Folientitel}
  ... Inhalt der Folie ...
\end{frame}
```

¹Die (Layout-)Themen von **beamer** sind immer nach Städten benannt.

²Die Farbthemen von **beamer** sind immer nach Tieren benannt.

12.1 Titelfolie

Der automatische Satz einer Titelfolie geschieht mit dem Befehl `\titlepage`

```
\begin{frame}
  \titlepage
\end{frame}
```

Der Befehl `\titlepage` greift zum Satz der Titelfolie auf die Argumente zurück, die den Befehlen `\title`, `\subtitle`, `\author`, `\date` und `\institute` mit geschweiften Klammern übergeben werden.

Ein Beispiel für das automatische Erzeugen der Titelfolie mit `beamer` zeigt Listing 12.2. Das Ergebnis dieses Beispiels zeigt Abbildung 12.2.

```
1 \documentclass{beamer}
2 \usepackage[utf8]{inputenc}
3 \usepackage{ngerman}
4 \usepackage[T1]{fontenc}
5 \usepackage{lmodern}      % Die Latin Modern Schriften sind besser für PDFs
6 \usepackage{url}
7
8 \usetheme{Frankfurt}
9 \usecolortheme{default}
10
11 \title{Hier steht der Titel}
12 \subtitle{Hier steht ein Untertitel (wenn man so etwas braucht)}
13 \author{Vorname Nachname}
14 \institute{Name des Arbeitgebers \\
15             oder der Hochschule. \\
16             Eventuell auch eine Email-Adresse \\
17             \url{benutzername@server.de}}
18 \date{Datum}
19
20 % Kleine Navigationsleiste am unteren Rand abschalten
21 \beamertemplatenavigationsymbolempty
22
23 %Inhalt der Fußzeile
24 \setbeamertemplate{footline}
25 {
26   \begin{beamercolorbox}{title}
27     \hspace*{0.2cm}
28     In der Fußzeile können z.B. Informationen zum Autor und zum Thema der Prä-
29       sentation stehen
30     \hfill
31     % Foliennummer einfügen
32     \insertframenumber/\inserttotalframenumber
33     \hspace*{0.2cm}
34   \end{beamercolorbox}
35 }
36
37 %Inhalt der Kopfzeile
38 \setbeamertemplate{headline}
39 {
```

```

39 \begin{beamercolorbox}{section in head/foot}
40   \vskip5pt\insertnavigation{12.75cm}\vskip5pt
41 \end{beamercolorbox}
42 }
43
44 \begin{document}
45
46 \section{Hier fängt ein Abschnitt an}
47 \subsection{Hier fängt ein Unterabschnitt an}
48
49 \begin{frame}
50 % Titelfolie generieren
51 \titlepage
52 \end{frame}
53
54 \end{document}

```

Listing 12.2: Automatisches Erzeugen der Titelfolie



Abbildung 12.2: Resultat von Listing 12.2

Das Beispiel in Listing 12.2 zeigt auch eine mögliche Verwendung der Kopf- und Fußzeilen. Im Beispiel wird mit Hilfe des Befehls `\insertnavigation` die Gliederungs-

struktur des Foliensatzes in der Kopfzeile ausgegeben. Abschnitte (`section`) sind mit ihren Abschnittsnamen aufgeführt. Jeder Unterabschnitt (`subsection`) und jede Folie sind durch Punkte unter den Abschnittsnamen repräsentiert.

Im Beispiel ist auch die Ausgabe der kleinen Navigationsleiste am unteren Rand durch den Befehl `\beamertemplatenavigationsymbolsempy` abgeschaltet

12.2 Automatischer Satz eines Inhaltsverzeichnisses

Wie bei anderen Dokumentklassen auch, ist es bei `beamer` üblich, Dokumente mit den Gliederungsbefehlen `\section`, `\subsection` und `\subsubsection` zu gliedern.

Der automatische Satz einer Folie mit der Agenda der Präsentation, also mit dem Inhalt des Foliensatzes, kann mit dem Befehl `\tableofcontents` geschehen.

```
\begin{frame}
  \frametitle{Agenda}
  \tableofcontents[Argument(e)]
\end{frame}
```

Das Verhalten des Befehls kann durch Argumente beeinflusst werden, die in eckigen Klammern übergeben werden. Sollen beispielsweise nur die Namen der Abschnitte (`sections`) im Inhaltsverzeichnis erscheinen, so realisiert dieses der Befehl

```
\tableofcontents[current,hideallsubsections]
```

Soll nur die Gliederung des aktuellen Abschnitts (`section`) inklusive seiner Unterabschnitte berücksichtigt werden, legt dieses der folgende Befehl fest.

```
\tableofcontents[currentsection]
```

12.3 Inhalte auf Präsentationsfolien setzen

Klassischerweise werden Inhalte auf Präsentationsfolien mit Aufzählungen (siehe Abschnitt 5.4), also mit den Umgebungen `itemize` und `enumerate` gesetzt.

Sinnvoll kann auch der Gebrauch von farbigen Blöcken sein. `beamer` stellt standardmäßig Umgebungen für drei verschiedene Blöcke zur Verfügung. Diese sind `block`, `exampleblock` und `alertblock`. Der Titel eines Blocks wird der Umgebung als Argument in geschweiften Klammern übergeben.

```

\begin{block}{Blocktitel}
  Blocktext
\end{block}

\begin{exampleblock}{Blocktitel}
  Blocktext
\end{exampleblock}

\begin{alertblock}{Blocktitel}
  Blocktext
\end{alertblock}

```

Ein Beispiel die Verwendung der genannten Umgebungen für Aufzählungen und Blöcke zeigt Listing 12.3. Der Quellcode zeigt nur den relevanten Ausschnitt aus der L^AT_EX-Quelldatei. Das Ergebnis dieses Beispiels zeigt Abbildung 12.3.

```

56 \begin{frame}
57 \frametitle{Aufzählungen und Blöcke}
58 \begin{itemize}
59 \item Eine Aufzählung\dots
60 \begin{enumerate}
61 \item mit mehreren
62 \item Unterpunkten.
63 \end{enumerate}
64 \item Es gibt verschiedene Arten von Blöcken:
65 \end{itemize}
66
67 \begin{block}{Blocktitel}
68 Blocktext
69 \end{block}
70
71 \begin{exampleblock}{Blocktitel}
72 Blocktext
73 \end{exampleblock}
74
75 \begin{alertblock}{Blocktitel}
76 Blocktext
77 \end{alertblock}
78 \end{frame}

```

Listing 12.3: Aufzählungen und farbige Blöcke auf Präsentationsfolien

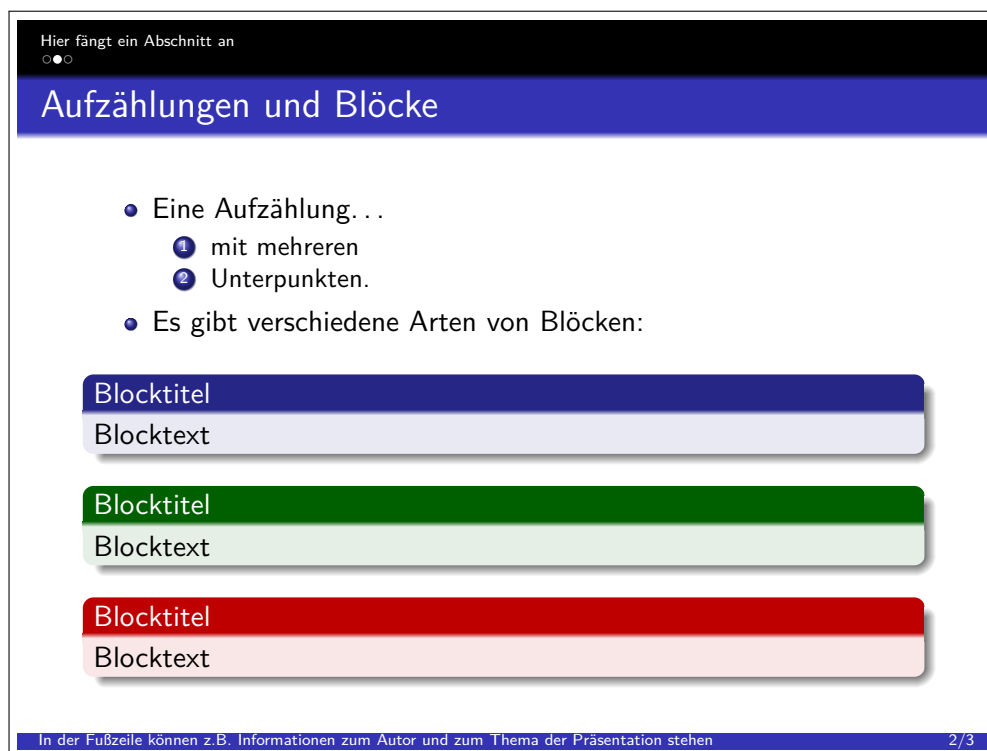


Abbildung 12.3: Resultat von Listing 12.3

12.4 Mehrspaltige Präsentationsfolien setzen

Ein hilfreiches Werkzeug, um das Layout von Präsentationsfolien zu definieren, ist die Umgebung `columns` zum Satz mehrspaltiger Folien.

Innerhalb einer Umgebung `columns` definiert jede Umgebung `column` eine neue Spalte und umschließt deren Inhalt. Die breite einer Spalte wird als Argument nach dem `\begin{column}` in leeren geschweiften Klammern als absolute oder relative Maßeinheit oder alternativ mit Hilfe eines Längenbefehls wie zum Beispiel `\textwidth` definiert. Zusätzlich ist es auch möglich, die vertikale Ausrichtung innerhalb jeder Spalte als Option in eckigen Klammern zu definieren. Der Buchstabe `T` steht dabei für *top*, also für eine Ausrichtung am oberen Rand der Spalte. Der Buchstabe `c` steht für *center*, also für eine vertikal zentrierte Ausrichtung des Inhalts.

Ein Beispiel für eine zweispaltige Folie, bei der beide Spalten gleich groß sind (jeweils die halbe Breite des Textfeldes) und sich in der rechten Spalte eine Abbildung befindet,

zeigt Listing 12.4. Der Quelltext zeigt nur den relevanten Ausschnitt aus der \LaTeX -Quelldatei. Das Ergebnis dieses Beispiels zeigt Abbildung 12.4.

```
83 \begin{frame}
84 \frametitle{Mehrspaltige Präsentationsfolien}
85 %
86 \begin{columns}
87 \begin{column}[T]{.5\textwidth}
88
89 \begin{block}{Blocktitel}
90 Blocktext
91 \end{block}
92
93 \begin{itemize}
94 \item Aufzählungspunkt 1
95 \item Aufzählungspunkt 2
96 \item Aufzählungspunkt 3
97 \item Aufzählungspunkt 4
98 \item Aufzählungspunkt 5
99 \end{itemize}
100 \end{column}
101 %
102 \begin{column}[T]{.5\textwidth}
103 \includegraphics[width=\textwidth]{../Bilder/Tux.pdf}
104 \end{column}
105 \end{columns}
106 \end{frame}
```

Listing 12.4: Aufzählungen und farbige Blöcke auf Präsentationsfolien

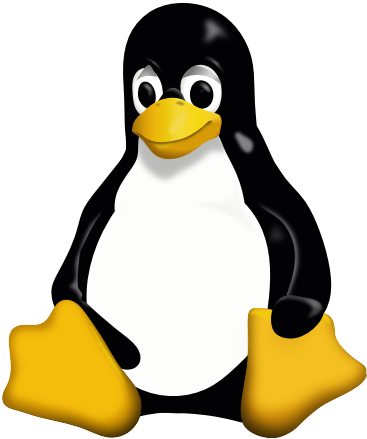
Hier fängt ein Abschnitt an
○○●

Mehrspaltige Präsentationsfolien

Blocktitel

Blocktext

- Aufzählungspunkt 1
- Aufzählungspunkt 2
- Aufzählungspunkt 3
- Aufzählungspunkt 4
- Aufzählungspunkt 5



In der Fußzeile können z.B. Informationen zum Autor und zum Thema der Präsentation stehen

3/3

Abbildung 12.4: Resultat von Listing 12.4

Literaturverzeichnis

- [1] H. Kopka, *L^AT_EX Einführung Band 1*. Addison-Wesley, 3 ed., 2000.
- [2] M. Goossens, F. Mittelbach, and A. Samarin, *Der L^AT_EX-Begleiter*. Addison-Wesley, 2000.
- [3] H. Voß, *Einführung in LaTeX: unter Berücksichtigung von pdfLaTeX, XLaTeX und LuaLaTeX*. Lehmanns Media, 2016.
- [4] F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley, *The L^AT_EX companion*. Addison-Wesley Professional, 2004.
- [5] H. Kopka and P. W. Daly, *Guide to L^AT_EX*. Pearson Education, 2003.
- [6] S. Krause, “Eine nicht allzu lange Einführung in L^AT_EX2 ϵ ,” 2012. <http://www.jokr.de/stefan/latex/LatexEinfuehrung.pdf>.
- [7] M. Jürgens and T. Feuerstack, “L^AT_EX – eine Einführung und ein bisschen mehr...,” 2017. https://www.fernuni-hagen.de/imperia/md/content/zmi_2010/a026_latex_einf.pdf.
- [8] T. Richter, “L^AT_EX – Tipps und Tricks,” 2013. https://www.tortools.de/lib/exe/fetch.php/latex:richter_latex_tipps_1.2.pdf.
- [9] M. Marx and O. Bückner, “Erstellung wissenschaftlicher Texte mit L^AT_EX,” 2017. https://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/latex.pdf?__blob=publicationFile.
- [10] T. Nagel, “Einführung in L^AT_EX,” 2018. http://www.nagel-net.de/Latex/DOKU/Latexkurs_Skript.pdf.
- [11] A. Gitter, “L^AT_EX,” 2018. http://web.eah-jena.de/~gitter/ahg_latex.pdf.
- [12] Creative Commons: Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland (CC BY-SA 3.0 DE). <https://creativecommons.org/licenses/by-sa/3.0/de/>.

- [13] Don Knuth's Home Page. <https://www-cs-faculty.stanford.edu/~knuth/>.
- [14] Leslie Lamport's Home Page. <http://www.lamport.org>.
- [15] T_EX Live. <http://www.tug.org/texlive/>.
- [16] MiK_TE_X. <https://miktex.org>.
- [17] MacT_EX. <http://www.tug.org/mactex/>.
- [18] Kate. <https://kate-editor.org>.
- [19] T_EXstudio. <http://texstudio.sourceforge.net>.
- [20] Texmaker. <http://www.xmlmath.net/texmaker/>.
- [21] TeXworks. <https://www.tug.org/texworks/>.
- [22] Eclipse. <https://www.eclipse.org>.
- [23] Eclipse TeXlipse. <https://projects.eclipse.org/projects/science.texlipse>.
- [24] Authorea. <https://www.authorea.com>.
- [25] Overleaf. <https://www.overleaf.com>.
- [26] ShareLaTeX. <https://www.sharelatex.com>.
- [27] N. Schwarz, *Einführung in T_EX*. Addison-Wesley, 1988.
- [28] U. Steinacker, "Leerraum in Typografie und Layout: So setzen Sie Gevierte, Halbgevierte und Co. richtig ein." <http://typeschool.de>.
- [29] minipage – goLaTeX. <https://golatex.de/wiki/minipage>.
- [30] KOMA-Script. <http://mirrors.ctan.org/macros/latex/contrib/koma-script/doc/scrguide.pdf>.
- [31] J. Schlosser, *Wissenschaftliche Arbeiten schreiben mit L^AT_EX*. mitp, 3 ed., 2009.
- [32] M. Ensenbach and M. Trettin, "Das L^AT_EX2_ε-Sündenregister oder Veraltete Befehle, Pakete und andere Fehler Version 2.7," 2016. <ftp://ftp.dante.de/tex-archive/info/l2tabu/german/l2tabu.pdf>.
- [33] Geviert – Typografie-Fachbegriffe-Wiki. <http://www.typografie.info/3/wiki.html/g/geviert-r165/>.

- [34] Geschäftsbriefe mit L^AT_EX 2_ε – der g-brief und g-brief2. <http://tug.ctan.org/macros/latex/contrib/g-brief/g-brief.pdf>.
- [35] L^AT_EX Font Catalogue. <http://www.tug.dk/FontCatalogue/>.
- [36] Y. Haralambous, “Typesetting old german: Fraktur, schwabacher, gotisch and initials,” *TUGboat*, vol. 12, no. 1, pp. 129–138, 1991. <https://www.tug.org/TUGboat/tb12-1/tb31hara.pdf>.
- [37] makebarcode – Print various kinds 2/5 and Code 39 bar codes. <http://mirrors.ctan.org/macros/latex/contrib/makebarcode/doc/latex/makebarcode/makebarcode.pdf>.
- [38] QR-Code. <https://de.wikipedia.org/wiki/QR-Code/>.
- [39] The calligra package for use with L^AT_EX 2_ε. <http://mirrors.ctan.org/macros/latex/contrib/fundus/calligra/calligra.pdf>.
- [40] H. Voß, *LaTeX Referenz*. Lehmanns Media, 2007.
- [41] Bibtex bibliography styles. https://www.sharelatex.com/learn/Bibtex_bibliography_styles.
- [42] LaTeX-Wörterbuch: bibliographystyle. https://de.wikibooks.org/wiki/LaTeX-W%C3%B6rterbuch:_bibliographystyle.
- [43] Google Scholar. <https://scholar.google.de>.
- [44] dblp computer science bibliography. <https://dblp.uni-trier.de>.
- [45] Natural Sciences Citations and References. <http://ftp.uni-erlangen.de/ctan/macros/latex/contrib/natbib/natbib.pdf>.
- [46] “Wikipedia – BibTeX.” <https://de.wikipedia.org/wiki/BibTeX>.
- [47] LaTeX-Kompendium: Zitieren mit BibTeX. https://de.wikibooks.org/wiki/LaTeX-Kompendium:_Zitieren_mit_BibTeX.
- [48] Beamer Theme Matrix. <https://hartwork.org/beamer-theme-matrix>.

Befehle, Umgebungen, Dokumentklassen und Optionen

10pt, 22	<code>\bibliographystyle</code> , 133
11pt, 22	<code>\bibliography</code> , 132
12pt, 22	<code>\bibname</code> , 40
<code>\Huge</code> , 61	<code>\bigg</code> , 107
<code>\LARGE</code> , 61	<code>\big</code> , 107
<code>\Large</code> , 61	<code>\block</code> , 143
<code>\Pr</code> , 105	<code>\bmod</code> , 105
<code>\abstractname</code> , 40	<code>\boxed</code> , 111
<code>\acute</code> , 105	<code>\breve</code> , 105
<code>\addcontentsline</code> , 44	<code>\caption</code> , 86
<code>\alertblock</code> , 143	<code>\cellcolor</code> , 80
<code>\and</code> , 37	<code>\centering</code> , 55
<code>\appendixname</code> , 40	<code>\chaptername</code> , 40
<code>\appendix</code> , 42	<code>\chapter</code> , 39
<code>\arccos</code> , 105	<code>\check</code> , 105
<code>\arcsin</code> , 105	<code>\cite</code> , 131
<code>\arctan</code> , 105	<code>\cleardoublepage</code> , 58
<code>\arg</code> , 105	<code>\clearpage</code> , 58
<code>\author</code> , 37, 141	<code>\cline</code> , 84
<code>\backmatter</code> , 42	<code>\columncolor</code> , 79
<code>\barcode</code> , 123	<code>\columnseprule</code> , 35
<code>\bar</code> , 105	<code>\columnsep</code> , 35
<code>\baselineskip</code> , 31	<code>\columnwidth</code> , 35
<code>\baselinestretch</code> , 32	<code>\contentsname</code> , 40
<code>\beamertemplatenavigationsymbolsempty</code> , 143	<code>\cosh</code> , 105
<code>\bfseries</code> , 62	<code>\cos</code> , 105
<code>\bibitem</code> , 131	<code>\coth</code> , 105
	<code>\cot</code> , 105

<code>\csc</code> , 105	<code>\headheight</code> , 29, 35
<code>\currentpage</code> , 33	<code>\headrulewidth</code> , 29
<code>\dashuline</code> , 60	<code>\headsep</code> , 35
<code>\date</code> , 38, 141	<code>\hfill</code> , 51
<code>\ddddot</code> , 105	<code>\hline</code> , 84
<code>\dddot</code> , 105	<code>\hom</code> , 105
<code>\ddot</code> , 105	<code>\hrulefill</code> , 52
<code>\definecolor</code> , 81	<code>\hspace</code> , 48
<code>\deg</code> , 105	<code>\huge</code> , 61
<code>\det</code> , 105	<code>\includegraphics</code> , 85
<code>\dim</code> , 105	<code>\indent</code> , 33
<code>\displaystyle</code> , 94	<code>\indexname</code> , 40
<code>\documentclass</code> , 21	<code>\inf</code> , 105
<code>\dotfill</code> , 52	<code>\insertnavigation</code> , 142
<code>\dotuline</code> , 60	<code>\institute</code> , 141
<code>\dot</code> , 105	<code>\item</code> , 62
<code>\doublebox</code> , 50, 67	<code>\itshape</code> , 62
<code>\emph</code> , 59	<code>\jobname</code> , 137
<code>\evensidemargin</code> , 35	<code>\ker</code> , 105
<code>\exampleblock</code> , 143	<code>\landscape</code> , 36
<code>\exp</code> , 105	<code>\large</code> , 61
<code>\fboxsep</code> , 67	<code>\left</code> , 107
<code>\fbox</code> , 111	<code>\lg</code> , 105
<code>\figurename</code> , 40	<code>\liminf</code> , 105
<code>\flushleft</code> , 56	<code>\limsup</code> , 105
<code>\flushright</code> , 56	<code>\lim</code> , 105
<code>\footnotemark</code> , 64	<code>\linewidth</code> , 35
<code>\footnotesize</code> , 61	<code>\listfigurename</code> , 40
<code>\footnotetext</code> , 64	<code>\listoffigures</code> , 44
<code>\footnote</code> , 64	<code>\listoftables</code> , 44
<code>\footrulewidth</code> , 29	<code>\listtablename</code> , 40
<code>\footskip</code> , 35	<code>\ln</code> , 105
<code>\frac</code> , 92	<code>\log</code> , 105
<code>\frontmatter</code> , 42	<code>\makeindex</code> , 46
<code>\gcd</code> , 105	<code>\maketile</code> , 37, 38
<code>\graphicspath</code> , 87	<code>\marginparpush</code> , 35, 65
<code>\grave</code> , 105	<code>\marginparsep</code> , 35, 65
<code>\har</code> , 105	<code>\marginparwidth</code> , 35, 65
<code>\hat</code> , 105	<code>\markboth</code> , 25

<code>\markright</code> , 25	<code>\part</code> , 39
<code>\mathbb</code> , 110	<code>\portrait</code> , 36
<code>\mathcal</code> , 98	<code>\prefacename</code> , 40
<code>\mathnormal</code> , 97	<code>\printindex</code> , 46
<code>\max</code> , 105	<code>\qqquad</code> , 111
<code>\mbox</code> , 110	<code>\qrcode</code> , 125
<code>\mdseries</code> , 62	<code>\quad</code> , 111
<code>\min</code> , 105	<code>\raggedleft</code> , 56
<code>\mod</code> , 105	<code>\raggedright</code> , 56
<code>\multicolumn</code> , 81	<code>\refname</code> , 40
<code>\multirow</code> , 83, 84	<code>\right</code> , 107
<code>\newline</code> , 58	<code>\rmfamily</code> , 62
<code>\newpage</code> , 58	<code>\rowcolor</code> , 78
<code>\nonfrenchspacing</code> , 50	<code>\scriptsize</code> , 61
<code>\nonumber</code> , 91	<code>\scshape</code> , 62
<code>\nopagebreak</code> , 58	<code>\secnumdepth</code> , 41
<code>\normalsize</code> , 61	<code>\section</code> , 39, 143
<code>\notin</code> , 99	<code>\sec</code> , 105
<code>\not</code> , 99	<code>\setcounter</code> , 30, 41
<code>\oddpagelayoutfalse</code> , 33	<code>\sffamily</code> , 62
<code>\oddpagelayouttrue</code> , 33	<code>\shadowbox</code> , 67
<code>\oddsidemargin</code> , 35	<code>\sinh</code> , 105
<code>\onecolumn</code> , 36	<code>\sin</code> , 105
<code>\ovalbox</code> , 67	<code>\sloppy</code> , 54
<code>\overbrace</code> , 106	<code>\slshape</code> , 62
<code>\overleftarrow</code> , 107	<code>\small</code> , 61
<code>\overline</code> , 106	<code>\sout</code> , 59
<code>\overrightarrow</code> , 107	<code>\space</code> , 51
<code>\overset</code> , 95	<code>\sqrt</code> , 93
<code>\pagename</code> , 40	<code>\subparagraph</code> , 39
<code>\pagenumbering</code> , 30	<code>\subsection</code> , 39, 143
<code>\pagestyle</code> , 24	<code>\substack</code> , 96
<code>\page</code> , 30	<code>\subsubsection</code> , 39, 143
<code>\paperheight</code> , 35	<code>\subtitle</code> , 141
<code>\paperwidth</code> , 35	<code>\sum</code> , 94
<code>\paragraph</code> , 39	<code>\sup</code> , 105
<code>\parindent</code> , 33	<code>\tablename</code> , 40
<code>\parskip</code> , 32	<code>\tableofcontents</code> , 43, 143
<code>\partname</code> , 40	<code>\tanh</code> , 105

<code>\tan</code> , 105	<code>\uuline</code> , 59
<code>\textasciicircum</code> , 93	<code>\uwave</code> , 59
<code>\textasciitilde</code> , 47	<code>\varepsilon</code> , 98
<code>\textbf</code> , 61	<code>\varphi</code> , 98
<code>\textcalligra</code> , 128	<code>\varpi</code> , 98
<code>\textfrac</code> , 123	<code>\varrho</code> , 98
<code>\textgoth</code> , 123	<code>\varsigma</code> , 98
<code>\textheight</code> , 35	<code>\vartheta</code> , 98
<code>\textit</code> , 61	<code>\vec</code> , 105
<code>\textmd</code> , 61	<code>\verb</code> , 65
<code>\textnormal</code> , 61	<code>\vspace</code> , 49
<code>\textrm</code> , 61	<code>\widehat</code> , 106
<code>\textsc</code> , 61	<code>\widetilde</code> , 106
<code>\textsf</code> , 61	<code>\xout</code> , 60
<code>\textsl</code> , 61	<code>\\</code> , 57
<code>\textswab</code> , 123	<code>a4paper</code> , 24
<code>\texttt</code> , 61	<code>a5paper</code> , 24
<code>\textunderscore</code> , 93	<code>abstract</code> , 38
<code>\textup</code> , 61	<code>arabic</code> , 30
<code>\textvisiblespace</code> , 47, 51	<code>array</code> , 108
<code>\textwidth</code> , 35	<code>article</code> , 21
<code>\thanks</code> , 38	<code>b5paper</code> , 24
<code>\tilde</code> , 105	<code>beamer</code> , 22
<code>\tiny</code> , 61	<code>book</code> , 21
<code>\titlepage</code> , 141	<code>boxedimage</code> , 68
<code>\title</code> , 37, 141	<code>center</code> , 55
<code>\tocdepth</code> , 39	<code>columns</code> , 145
<code>\today</code> , 38	<code>description</code> , 62
<code>\topmargin</code> , 35	<code>dinbrief</code> , 22
<code>\ttfamily</code> , 62	<code>dingautolist</code> , 128
<code>\twocolumn</code> , 35	<code>dingfill</code> , 127
<code>\uline</code> , 59	<code>dingline</code> , 127
<code>\underbrace</code> , 106	<code>dinglist</code> , 127
<code>\underleftarrow</code> , 107	<code>ding</code> , 126
<code>\underline</code> , 59, 106	<code>displaymath</code> , 90
<code>\underrightarrow</code> , 107	<code>draft</code> , 23
<code>\underset</code> , 95	<code>em</code> , 59
<code>\upshape</code> , 62	<code>enumerate</code> , 62
<code>\url</code> , 66	<code>eqnarray*-Umgebung</code> , 91

eqnarray, 90
eqnarray-Umgebung, 91
equation-Umgebung, 90
executivepaper, 24
fancybox, 66
filecontents, 137
final, 23
fleqn, 23
g-brief2, 22, 115
itemize, 62
landscape, 24, 36
legalpaper, 24
leqno, 23
letterpaper, 24
letter, 21
math, 89
natbib, 135
notitlepage, 23
onecolumn, 22
oneside, 23
openany, 24
openright, 24
portrait, 36
proc, 21
quotation, 56
quote, 56
report, 21
scrartcl, 22
scrbook, 22
scrlttr2, 22, 119
scrreprt, 22
slides, 21
tabularx, 75
tabular, 71
thebibliography, 131, 135
titlepage, 23
twocolumn, 22
twoside, 23
verbatim, 65

Stichwortverzeichnis

- Überstreichungssymbole, 106
- aux-Datei, 6
- log-Datei, 6

- Abbildungen, 85
- Abbildungsverzeichnis, 44
- Abkürzungen, 48
- Absatzabstand, 32
- Absatzeinzüge, 12
- Abschnitt, 39
- Abstände, 31
- Abstand
 - horizontal, 48
 - vertikal, 49
- Abstract, 38
- Altdeutsche Schrift, 123
- Anführungszeichen, 13
- Anhang, 42
- Argument
 - obligatorisch, 5
 - optinal, 5
 - zwingend, 9
- Aufzählung, 62
- Aufzählungspunkt, 14, 62
- Auslassungspunkte, 109

- Backslash, 13
- Barcode, 123
- Basisschrift, 22
- Befehle, 9
- BibT_EX, 132

- Bilder, 85
- binäre Operationen, 98
- Bindestrich, 12, 16
- Blocksatz, 55
- Boxen, 66
- Brief, 115
- Bruch, 92
- Bruchstrich, 92
- Buchhauptteil, 42
- Buchnachspann, 42
- Buchstaben
 - Griechische, 96
 - Kalligrafische, 98
- Buchvorspann, 42

- CMYK, 78
- Copyright, 14

- Dagger-Zeichen, 14
- Dezimalpunkt, 11
- Dezimalzahl, 11
- DIN-A4, 24
- DIN-A5, 24
- Dokumentklasse, 21
- Doppel-Dagger-Zeichen, 14

- eingерückter Text, 56
- Einrückung
 - Absatz, 32
- Einspaltig, 22, 36
- Euro, 14

- Exponent, 92
- fancyheadings, 27
- Farbmodell, 78
- Flattersatz, 55
- Formatierungshilfen, 47
- Formel
 - abgesetzt, 90
 - eingebettete, 89
 - nummeriert, 90
- Formelsatz, 89
- Fortsetzungspunkte, 17
- Frenchspacing, 50
- Fußnote, 64
- Fußzeile, 27, 33
- Gedankenstrich, 12, 17
- Geviert, 11
- Geviertstrich, 12, 17
- Gleichungssysteme, 90
- Gliederung, 37
- Gliederungsbefehl, 39
- Grafiken, 85
- Griechische Buchstaben, 96
- Gruppe, 9
- Halbgeviertstrich, 12, 17
- Hochformat, 36
- HTML, 78
- Hurenkind, 32
- Index, 92
- Indexregister, 45
- Inhaltsverzeichnis, 43
 - Eintrag, 44
- Integral, 94
- Integralzeichen, 94
- Internetadressen, 66
- Kalligrafie, 98
- Kapitel, 39
- Kegelhöhe, 11
- Klammer
 - Größe, 107
- Klammeraffe, 133
- Klammern
 - eckig, 5, 9
 - geschweift, 5, 9
 - rund, 5, 9
- Klassensoptionen, 22
- Kopfzeile, 25, 27, 33
- Längenbefehle, 12
- Layout, 47
- Leerzeichen, 51
- Ligatur, 17
- Linksbündige Spalte, 72
- Linksbündiger Text, 56
- Literaturverzeichnis, 131
- Maß
 - horizontal, 12
 - vertikal, 12
- Maßeinheit, 11
 - fest, 11
- Marginalie, 64
- Markierung, 45
- Massangaben, 11
- Mathematik, 89
 - Bruch, 92
 - Konstrukte, 92
- Matrizen, 108
- Mengensymbol, 110
- Nenner, 92
- Operationssymbole, 98
- Ordnung, 93
- Papiergröße, 24

- Paragraph-Zeichen, 14
- Pfeil, 100
- Pfund Sterling, 14
- Portrait, 36
- Präambel, 5
- Produktzeichen, 94

- QR-Code, 125
- Querformat, 36
- Querverweis, 45

- Radikant, 93
- Randbemerkung, 64
- Randnotiz, 64
- Rastergrafik, 85
- Rechtsbündige Spalte, 72
- Rechtsbündiger Text, 56
- Registered-Zeichen, 14
- RGB, 78

- Schlüsselwort, 134
- Schriftgröße, 22
- Schriftschnitt, 59
- Schusterjunge, 32
- Seitendeklaration, 33
- Seitenlayout
 - einspaltig, 22, 36
 - Hochformat, 36
 - Portrait, 36
 - Querformat, 36
 - zweispaltig, 22, 35
- Seitennummerierung, 30
 - Stil, 30
- Seitenstil, 24
- Seitenumbruch, 58
- Spalten, 71
- Spaltenformatierungseintrag, 72
- Spezialzeichen, 17
- Stichwortverzeichnis, 45
- Summe, 94

- Summenzeichen, 94

- Tabellen, 71
 - Doppelstrich, 72
 - Farben, 78
 - Felder zusammenfassen, 81
 - Spalten, 72
 - Strich, 72
- Tabellendefinition, 71
- Tabellenpräambel, 71
- Tabellenverzeichnis, 44
- Teil, 39
- Text
 - eingerückt, 56
 - linksbündig, 56
 - rechtsbündig, 56
 - unformatierter, 65
 - zentriert, 55
- Textbreite, 12
- Textfeld, 53
- Textformel, 89
- Texthöhe, 12
- Texthervorhebungen, 59
- Textkörper, 33
- Titelei, 37
- Titelseite, 23, 38
- Trademark-Zeichen, 14
- Trennstich, 16
- Trennungsregel, 54

- Umgebung, 10
- Unformatierter Text, 65
- Unterabschnitt, 39, 40
- Unterstreichungsymbole, 106
- URL, 66

- Vektorgrafik, 85
- Vergleichssymbole, 99
- Viertelgeviertstrich, 12, 16

Wortabstand, 47
Worttrennung, 54
Wurzel, 93
Wurzelzeichen, 93
WYSIWYG, 1
WYSIWYM, 1

Yen, 14

Zähler, 92
ZapfDingbats, 125
Zeichen
 Abstand, 47

 positionieren, 95
Zeichenabstand, 47
Zeiger, 100
Zeilenabstand, 31
Zeilenumbruch, 23, 48, 57
Zentrierte Spalte, 72
Zentrierter Text, 55
Ziffern
 arabisch, 30
 römisch, 30
Zusammenfassung, 38
Zweispaltig, 22, 35
Zwischenraum, 51, 52