

# Ein Einstieg in $\text{\LaTeX}$

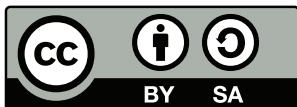
Prof. Dr. Christian Baun  
Frankfurt University of Applied Sciences  
(1971-2014: Fachhochschule Frankfurt am Main)  
Nibelungenplatz 1  
60318 Frankfurt am Main  
`christianbaun@fb2.fra-uas.de`

Version 0.1

16. Mai 2018



Dieses Werk und der zugehörige  $\text{\LaTeX}$ -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

<b>Vorwort</b>	<b>xi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Aufbau dieses Dokuments . . . . .	1
1.2 Zusammenhang von T <sub>E</sub> X/L <sup>A</sup> T <sub>E</sub> X und historischer Hintergrund . . . . .	2
1.3 Installation von L <sup>A</sup> T <sub>E</sub> X und Möglichkeiten damit zu arbeiten . . . . .	3
1.4 Das allererste Beispiel . . . . .	4
<b>2 Befehle, Gruppen, Umgebungen und Zeichen</b>	<b>7</b>
2.1 Befehle . . . . .	7
2.2 Gruppen . . . . .	7
2.3 Umgebungen . . . . .	8
2.4 Maßangaben mit absoluten und relativen Maßeinheiten . . . . .	9
2.4.1 Anwendungsbeispiele . . . . .	10
2.4.2 Elastische Maße . . . . .	10
2.5 Sonderzeichen . . . . .	11
2.5.1 Besondere Zeichen . . . . .	11
2.5.2 Einige häufig verwendete Zeichen bzw. Symbole . . . . .	12
2.5.3 Punkte und Striche . . . . .	14
2.5.4 Anführungszeichen . . . . .	15
2.5.5 Akzente . . . . .	15
2.5.6 Ligaturen . . . . .	15
<b>3 Dokumentklassen und Seitenstile</b>	<b>19</b>
3.1 Dokumentklassen . . . . .	19
3.1.1 Klassenoptionen . . . . .	20
3.2 Seitenstile . . . . .	23
3.3 Definition der Kopfzeile mit <code>markright</code> und <code>markboth</code> . . . . .	24
3.4 Definition der Kopfzeile mit <code>fancyhdr</code> . . . . .	26
3.5 Seitennummerierung . . . . .	28
3.6 Wichtige Abstände . . . . .	29
3.6.1 Zeilenabstand . . . . .	29

3.6.2	Absatzabstand . . . . .	30
3.6.3	Einrückung der ersten Zeile eines Absatzes . . . . .	31
3.7	Parameter zur Seitendeklaration . . . . .	31
3.8	Mehrspaltenlayout . . . . .	34
3.9	Querformat . . . . .	34
3.10	Befehle und Umgebungen zur Gliederung . . . . .	35
3.10.1	Titelei . . . . .	35
3.10.2	Die Zusammenfassung – das Abstract . . . . .	36
3.10.3	Gliederungsbefehle . . . . .	37
3.10.4	Tiefe der Nummerierung bei der Gliederung ändern . . . . .	39
3.10.5	Anfangswert der Nummerierung ändern . . . . .	40
3.10.6	Zusätzliche Untergliederung bei Büchern . . . . .	40
3.10.7	Anhang . . . . .	41
3.11	Inhaltsverzeichnis . . . . .	41
3.11.1	Manuell Einträge in das Inhaltsverzeichnis einfügen . . . . .	42
3.12	Abbildungs- und Tabellenverzeichnis . . . . .	43
3.13	Querverweise . . . . .	43
3.14	Index . . . . .	44
<b>4</b>	<b>Formatierungshilfen</b>	<b>45</b>
4.1	Abstände von Wörtern und Zeichen . . . . .	45
4.1.1	Abstände in Sätzen . . . . .	45
4.1.2	Horizontale Abstände . . . . .	46
4.1.3	Vertikale Abstände . . . . .	47
4.1.4	Leerzeichen in Verbindung mit Satzzeichen – Frenchspacing . . . . .	48
4.1.5	Leerzeichen darstellen . . . . .	49
4.1.6	Zeilen mit Zwischenräumen auffüllen . . . . .	49
4.1.7	Zwischenräume mit Zeichen auffüllen . . . . .	50
4.1.8	Einhaltung der Grenzen des Textfeldes . . . . .	52
4.1.9	Trennungsregeln . . . . .	52
4.2	Absätze zentrieren, nach einer Seite ausrichten oder einrücken . . . . .	53
4.2.1	Zentrierter Text . . . . .	54
4.2.2	Linksbündiger Text . . . . .	54
4.2.3	Rechtsbündiger Text . . . . .	55
4.2.4	Beidseitig eingerückter Text . . . . .	55
4.3	Zeilenumbrüche . . . . .	56
4.4	Seitenumbrüche . . . . .	57

<b>5</b>	<b>Texthervorhebungen</b>	<b>59</b>
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
<b>6</b>	<b>Tabellen</b>	<b>71</b>
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 . . . . .	82
6.5	Horizontale Linien . . . . .	84
<b>7</b>	<b>Bilder einfügen</b>	<b>85</b>
7.1	Pfade der Bilddateien definieren . . . . .	86
<b>8</b>	<b>Mathematischer Formelsatz</b>	<b>89</b>
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 . . . . .	103
8.4.7	Mathematische Funktionen . . . . .	103
8.4.8	Akzente in Formeln . . . . .	105
8.4.9	Über- und Unterstreichungen . . . . .	105

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
<b>9 Briefe setzen</b>	<b>115</b>
9.1 Briefe mit der Dokumentklasse <code>g-brief2</code> . . . . .	115
9.2 Briefe mit der Dokumentklasse <code>scrlttr2</code> . . . . .	119
<b>10 Sonderschriften</b>	<b>123</b>
10.1 Altdeutsche Schriften . . . . .	123
10.2 Barcodes . . . . .	123
10.3 QR-Codes . . . . .	125
10.4 Besondere Zeichen . . . . .	125
10.5 Kalligrafie . . . . .	128
<b>11 Literaturverzeichnis</b>	<b>131</b>
<b>12 Präsentationsfolien</b>	<b>133</b>
<b>Literaturverzeichnis</b>	<b>135</b>
<b>Verzeichnis der Befehle, Umgebungen, Dokumentklassen und Optionen</b>	<b>137</b>
<b>Stichwortverzeichnis</b>	<b>143</b>

# Abbildungsverzeichnis

1.1	Resultat von Listing 1.1 . . . . .	6
2.1	Resultat von Listing 2.1 . . . . .	8
3.1	Aufteilung von <code>fancyheadings</code> bei geraden Seiten . . . . .	26
3.2	Aufteilung von <code>fancyheadings</code> bei ungeraden Seiten . . . . .	26
3.3	Voreingestelltes Seitenlayout bei <code>fancyhdr</code> . . . . .	27
3.4	Layout ungerader Seiten im aktuellen Dokument . . . . .	33
4.1	Auswirkungen von <code>\nonfrenchspacing</code> und <code>\frenchspacing</code> . . . . .	49
4.2	Resultat von Listing 4.1 . . . . .	51
5.1	Minipages mit Rahmen realisiert die Umgebung <code>boxedminipage</code> . . . . .	69
6.1	Felder über die Grenzen von Zeilen zusammenfassen . . . . .	83
6.2	Horizontale Linien mit fast beliebiger Breite in Tabellen setzen . . . . .	84
7.1	Abbildungen skalieren die Optionen <code>height</code> oder <code>width</code> . . . . .	85
7.2	Abbildungen drehen 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





# Tabellenverzeichnis

2.1	Absolute und relative Maßeinheiten . . . . .	9
2.2	Besondere Zeichen . . . . .	12
2.3	Zeichen (Symbole) . . . . .	12
2.4	Befehle zum komfortablen Satz der $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Schriftzüge . . . . .	13
2.5	Zeichen aus dem Erweiterungspaket <code>textcomp</code> . . . . .	13
2.6	Horizontale Striche . . . . .	14
2.7	Fortsetzungspunkte mit $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . . . . .	15
2.8	Anführungszeichen . . . . .	16
2.9	Akzente . . . . .	16
2.10	Spezialzeichen verschiedener europäischer Sprachen . . . . .	16
2.11	Ligaturen . . . . .	17
3.1	Papierformate . . . . .	22
3.2	Seitenstile . . . . .	23
3.3	Standardmäßiger Inhalt der Kopfzeile . . . . .	25
3.4	Seitennummerierungsstile . . . . .	28
3.5	Längenbefehle zur Beeinflussung des Layouts . . . . .	32
3.6	Gliederungsebenen und der Zähler <code>tocdepth</code> . . . . .	37
3.7	Auswirkungen von <code>(n)german.sty</code> auf die Gliederungselemente . . . . .	39
4.1	Beispiele für Abstände in Sätzen . . . . .	46
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> . . . . .	73
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 bestimmten Spalten einfügen . . . . .	75
6.6	Jede Spalte erhält den gleichen Anteil an der verfügbaren Tabellenbreite	76
6.7	Spalten mit automatischer Breite bei <code>tabularx</code> . . . . .	77
6.8	Kombination der Spaltenspezifikationen von <code>tabular</code> und <code>tabularx</code> . . .	77
6.9	Übersicht über die unterstützten Farbmodelle . . . . .	79
6.10	Einfärben ganzer Zeilen einer Tabelle . . . . .	79
6.11	Einfärben ganzer Spalten einer Tabelle . . . . .	80
6.12	Einfärben einzelner Felder einer Tabelle . . . . .	81
6.13	Spaltenübergreifendes Zusammenfassen von Feldern einer Tabelle . . . .	82
6.14	Änderung der Ausrichtung des Inhalts in einem einzelnen Feld . . . . .	83
7.1	Beispiele für relative und absolute Pfade . . . . .	87
8.1	Größen 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 . . . . .	106
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

# Vorwort

Wer sich schon etwas mit dem Thema Textsatz mit  $\text{\LaTeX}$  beschäftigt, und zum ersten Mal auf dieses Werk stößt, der wird sich vermutlich folgende Frage stellen:

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

Die Frage ist berechtigt. Immerhin sind in den letzten Jahrzehnten zahlreiche Bücher in deutscher und englischer Sprache zu diesem Thema erschienen und wer in den gängigen Suchmaschinen nach einer freien Einführung zum Thema sucht, der wird an vielen Stellen [1][2][3][4][5] fündig.

Das Ziel bei der Erstellung war und ist es auch zu keiner Zeit, ein möglichst umfangreiches Werk zu schreiben, das andere in den Schatten stellt.

Die primäre Motivation hinter der Entstehung dieses Werks ist das es nur wenige einführende Bücher zu  $\text{\LaTeX}$  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 [6].

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  
Mai 2018



# 1 Einleitung

L<sup>A</sup>T<sub>E</sub>X 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“, was bedeutet, das das gedruckte Dokument der Darstellung auf dem Bildschirm entspricht.

Im Gegensatz zu einem solchen Textverarbeitungsprogramm schreibt ein Autor der L<sup>A</sup>T<sub>E</sub>X verwendet, entweder mit einem beliebigen Editor oder mit einer speziellen für das Erstellen von L<sup>A</sup>T<sub>E</sub>X-Dokumenten konzipierten Anwendung den Text zusammen mit Formatanweisungen in eine Textdatei. Das Textsatzsystem L<sup>A</sup>T<sub>E</sub>X setzt diese Dokumente dann automatisch. Der Autor definiert also den Inhalt (also den Text) und die Eckpunkte des Layouts (den Satz). Die Fähigkeiten von L<sup>A</sup>T<sub>E</sub>X gehen deutlich über die eines gewöhnlichen Textverarbeitungsprogramms hinaus. L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X zu bekommen.

## 1.1 Aufbau dieses Dokuments

Dieses Kapitel enthält Informationen zum technischen Hintergrund und zur Arbeitsweise von L<sup>A</sup>T<sub>E</sub>X. Zudem erfahren Sie hier anhand erster Beispiele, wie eine L<sup>A</sup>T<sub>E</sub>X-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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X und historischer Hintergrund

Als Urvater des Textsatzsystems T<sub>E</sub>X gilt der amerikanische Mathematiker und Informatiker Donald E. Knuth [7]. 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 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 Bleilettern arbeitet, simuliert. Dieses Projekt namens T<sub>E</sub>X schloss Knuth 1986 ab – also noch lange vor der ersten Version von Linux. Das X steht dabei für den im lateinischen Alphabet nicht vorhandenen griechischen Buchstaben Chi.

Da die Markup-Sprache von  $\text{T}_{\text{E}}\text{X}$  an manchen Stellen nicht sehr intuitiv ist, entwickelte der Mathematiker Leslie Lamport [8] ab 1982  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Dieses enthält neben  $\text{T}_{\text{E}}\text{X}$  selbst auch unzählige Makropakete, die dazu da sind, den Autoren die Arbeit an Ihren Dokumenten zu vereinfachen und im Gegensatz zu  $\text{T}_{\text{E}}\text{X}$ , dessen Entwicklung abgeschlossen ist, wird  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  stetig weiterentwickelt.

### 1.3 Installation von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ und Möglichkeiten damit zu arbeiten

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  besteht aus dem eigentlichen  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Compiler und einer Vielzahl an Kommandozeilenwerkzeugen, Erweiterungspaketen, Schriften und Dokumentation. Die manuelle Installation dieser Komponenten wäre ein sehr aufwändiges Unterfangen. Aus diesem Grund existieren sogenannte  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Distributionen, um die Installation von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zu erleichtern.

Die bekanntesten aktiv gepflegten  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Distributionen sind  $\text{T}_{\text{E}}\text{X}$  Live [9]  $\text{MiK}_{\text{T}}\text{T}_{\text{E}}\text{X}$  [10] und  $\text{MacT}_{\text{E}}\text{X}$  [11]. Praktisch alle Linux-Distributionen enthalten  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Pakete von Haus aus. Während  $\text{T}_{\text{E}}\text{X}$  Live für alle gängigen UNIX-Betriebssysteme und Windows verfügbar ist, liegt der Fokus von  $\text{MiK}_{\text{T}}\text{T}_{\text{E}}\text{X}$  ganz klar auf Windows und der Fokus von  $\text{MacT}_{\text{E}}\text{X}$  auf Mac OS X.

Wenn eine Linux-Distribution  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  bei der Installation des Betriebssystems nicht schon automatisch mitinstalliert hat, ist das über die Paketverwaltung möglich. Andere Betriebssystemen wie Windows und Mac OS erfordern einen manuellen Bezug der Installationspakete und deren Installation.

Die Arbeit mit  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Compilers und ein PDF-Anzeigeprogramm zur Ergebniskontrolle nötig. Wer diese Form der Arbeit mit  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  mit drei Fenstern bevorzugt, der greift idealerweise auf einen Editor Komfortmerkmalen wie Syntaxhervorhebung zurück.

Einige Texteditor wie zum Beispiel Kate [12] bieten die Möglichkeit, eine Shell in die Oberfläche einzubinden. Das kann die Organisation des Desktop erleichtern.

Eine weitere Möglichkeit der Arbeit mit  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ist die Nutzung einer grafischen integrierten Entwicklungsumgebung (IDE = *Integrated Development Environment*) wie zum Beispiel  $\text{T}_{\text{E}}\text{X}$ studio [13], Texmaker [14], TeXworks [15] oder Eclipse [16] zusammen

mit der Erweiterung TeXlipse [17]. Diese Werkzeuge bieten eingebettete Anzeigeprogramme und Schaltflächen, um die gängigen Interaktionen mit dem L<sup>A</sup>T<sub>E</sub>X-Compiler und den beteiligten Werkzeugen mit der Maus anzuweisen. Zudem bieten integrierte Entwicklungsumgebung Schaltflächen um L<sup>A</sup>T<sub>E</sub>X-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-L<sup>A</sup>T<sub>E</sub>X-Editoren, die eigentlich Online-IDEs sind. Beispiele für solche Lösungen sind Authorea [18], Overleaf [19] und ShareLaTeX [20]. Alle diese Dienste bieten die Nutzung innerhalb gewisser Ressourcenkontingente kostenfrei an. Weitere Speicherressourcen und einzelne Funktionen erfordern in der Regel allerdings einen kostenpflichtigen Zugang. Weitere Nachteile solcher Werkzeuge sind die Tatsachen, dass die Software dieser Dienst ein der Regel 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 L<sup>A</sup>T<sub>E</sub>X gesetztes Dokuments mit einigen typischen Befehlen. Wenn Sie das Beispiel nicht aus dem Github-Verzeichnis kopieren können oder möchten, dann schreiben Sie Quellcode von Listing 1.1 mit einem Texteditor Ihrer Wahl in eine neue Textdatei mit der Dateierweiterung `.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 auch 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 werden.

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 in 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 aber

nach Möglichkeit sehr sparsam umgehen, da sie dem Prinzip der Trennung von Layout und Inhalt widersprechen.

Um mit Hilfe von  $\text{\LaTeX}$  und der Quelldatei von Listing 1.1 eine PDF-Datei zu erzeugen, rufen Sie auf der Kommandozeile den  $\text{\LaTeX}$ -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  $\text{\LaTeX}$ -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  $\text{\LaTeX}$ -Aufrufs
- `\listing1.pdf` – Die Ausgabe des  $\text{\LaTeX}$ -Dokuments im Portable Document Format (`pdf`)

Das erzeugte Dokument (siehe Abbildung 1.1) in der `pdf`-Datei lässt sich mit entsprechenden Betrachtern, wie sie alle aktuellen Betriebssysteme enthalten, anzeigen und ausdrucken.

Ihr erstes  $\text{\LaTeX}$ -Dokument.

Hier ist noch *nicht* so viel zu sehen,  
außer wenig **Text**.

Abbildung 1.1: Resultat von Listing 1.1

## 2 Befehle, Gruppen, Umgebungen und Zeichen

In diesem Kapitel erfahren Sie, was unter  $\text{\LaTeX}$  Befehle und Umgebungen sind, und wie Sie mit diesen arbeiten können. Zudem behalten dieses Kapitel wichtigste Sonderzeichen und präsentiert eine Auswahl grundlegender Befehle und Umgebungen anhand von Beispielen.

### 2.1 Befehle

Wie bereits auf Seite 5 beschrieben, leitet der Backslash ( $\backslash$ ), der sogenannte *escape character*  $\text{\LaTeX}$ -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  $\text{\LaTeX}$  *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 ist aber 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 möglich. Ein

Beispiel dafür ist der `\verb`-Befehl (für das Setzen von unformatiertem Text mit fester Zeichenbreite). Dieser 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 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 von  $\text{\LaTeX}$  unterstützten Maßeinheiten zeigt Tabelle 2.1. Dezimalzahlen dürfen übrigens bei  $\text{\LaTeX}$  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` erlaubt anzugeben.

$\text{\LaTeX}$  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 [21]

Abkürzung	Bedeutung	Umrechnungen	
<code>pt</code>	„Point“ (Punkt)	$1\text{ pt} = 0,013837\text{ in}$	$1\text{ pt} = 0,0351\text{ cm}$
<code>pc</code>	Pica	$1\text{ pc} = 12\text{ pt}$	$1\text{ pc} = 0,422\text{ cm}$
<code>in</code>	„Inch“ (Zoll)	$1\text{ in} = 72,27\text{ pt}$	$1\text{ in} = 2,54\text{ cm}$
<code>mm</code>	Millimeter	$1\text{ mm} = 2,845\text{ pt}$	
<code>cm</code>	Zentimeter	$1\text{ cm} = 28,45\text{ pt}$	$1\text{ cm} = 10\text{ mm}$
<code>bp</code>	Big Point	$72\text{ bp} = 1\text{ in}$	$1\text{ bp} = 0,03528\text{ cm}$
<code>dd</code>	Didot Punkt	$1157\text{ dd} = 1238\text{ pt}$	$1\text{ dd} = 0,03761\text{ cm}$
<code>cc</code>	Cicero	$1\text{ cc} = 12\text{ dd}$	$1\text{ cc} = 0,45132\text{ cm}$
<code>sp</code>	Scaled Point	$65536\text{ sp} = 1\text{ pt}$	$1\text{ sp} < 0,6 * 10^{-6}\text{ cm}$
<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 die dabei die Schriftgröße (Kegelhöhe).

## 2.4. MASSANGABEN MIT ABSOLUTEN UND RELATIVEN MASSEINHEITEN

Bei einer Schriftgröße von 12 Punkten hat ein Geviert dementsprechend eine Länge von 12 Punkten [22].

Bei der Arbeit mit  $\text{\LaTeX}$  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 [23].

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

In der Literatur zu  $\text{\LaTeX}$  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. [21][24]

### 2.4.1 Anwendungsbeispiele

Mit den in diesem Abschnitt beschriebenen Maßeinheiten werden unter anderem Längenbefehlen, die das Layout der Dokumente 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 nur 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, dass elastische Maße um einen festzulegenden Betrag gestreckt und gestaucht werden können. Die Definition eines elastischen Maßes erfolgt mit drei Werten [24]:

- 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 hierbei 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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>\&amp;</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

$\text{\LaTeX}$  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  $\text{\TeX}$  und  $\text{\LaTeX}$  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  $\text{\LaTeX}$ -Quelltextes



Tabelle 2.4: Befehle zum komfortablen Satz der  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Schriftzüge

Symbol	Bezeichnung	Befehl
$\text{T}_{\text{E}}\text{X}$	Schriftzug von $\text{T}_{\text{E}}\text{X}$	<code>\TeX</code>
$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$	Schriftzug von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$	<code>\LaTeX</code>
$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$	Schriftzug von seit 1994 aktuellen Version von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$	<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>	$\text{¢}$
<code>\textdollar</code>	$\$$	<code>\textcent</code>	$\text{¢}$
<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>\textttimes</code>	$\times$	<code>\textonesuperior</code>	$^1$
<code>\textttwosuperior</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>	$\sqrt{\phantom{x}}$
<code>\textperthousand</code>	$\text{‰}$		
<code>\textrightarrow</code>	$\rightarrow$	<code>\textleftarrow</code>	$\leftarrow$
<code>\textuparrow</code>	$\uparrow$	<code>\textdownarrow</code>	$\downarrow$
<code>\textdagger</code>	$\dagger$	<code>\textdaggerdbl</code>	$\ddagger$
<code>\textohm</code>	$\Omega$	<code>\textmho</code>	$\text{℧}$
<code>\textlbrackdbl</code>	$\llbracket$	<code>\textrbrackdbl</code>	$\rrbracket$
<code>\textcopyright</code>	$\text{©}$	<code>\textcopyrightleft</code>	$\text{©}$
<code>\textregistered</code>	$\text{®}$	<code>\textcircledP</code>	$\text{®}$
<code>\texttrademark</code>	$\text{™}$	<code>\textservicemark</code>	$\text{SM}$
<code>\textnumero</code>	$\text{№}$	<code>\textestimated</code>	$\text{€}$
<code>\textdegree</code>	$^\circ$	<code>\textcelsius</code>	$^\circ\text{C}$
<code>\textbullet</code>	$\bullet$	<code>\textopenbullet</code>	$\circ$

<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

L<sup>A</sup>T<sub>E</sub>X enthält Befehle, um unterschiedliche Arten von horizontalen Strichen zu setzen (siehe Tabelle 2.6).

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 erzeugt (--) 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 den Befehl `\textemdash` erzeugt.

Tabelle 2.7: Fortsetzungspunkte mit  $\text{\LaTeX}$ 

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

Auf den ersten Blick kurios erscheint es, das  $\text{\LaTeX}$  mit `\dots` einen Befehl vorhält, um drei aufeinander folgende Punkte, so genannte Fortsetzungspunkte, zu erzeugen. 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

$\text{\LaTeX}$  ermöglicht den Satz verschiedener Anführungszeichen. Eine Übersicht enthält 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

$\text{\LaTeX}$  ermöglicht den Satz verschiedener Akzente (siehe Tabelle 2.9).

Zudem ermöglicht  $\text{\LaTeX}$  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 \guillemontleft
›Halbe französische‹		\frq oder \guilemontright
		\flq oder \guillemontleft
“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}	öo	\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<sup>A</sup>T<sub>E</sub>X wie folgt gesetzt: ff, fi, fl, ffi, ffl.

Streng genommen sind auch der Gedankenstrich und der verlängerte Gedankenstrich (siehe Seite 14) Ligaturen, da Sie aus zwei bzw. drei einzelnen Zeichen zu einem zusammengefasst werden.

Tabelle 2.11: Ligaturen

Buchstabenkombination	Ligatur	unterdrückte Ligatur
<b>ff</b>	ff	ff
<b>fi</b>	fi	fi
<b>fl</b>	fl	fl
<b>ffi</b>	ffi	ffi
<b>ffl</b>	ffl	ffl

Wenn  $\text{\LaTeX}$  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 denn Wörter falsch getrennt werden.

Eine Ligatur kann unterdrückt werden, indem zwischen jedes Buchstabenpaar, das zur möglichen Ligatur gehören, ein Paar geschweifte 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 gehören, einen Backslash und einen Slash zu setzen `\./`.



## 3 Dokumentklassen und Seitenstile

Mit den Dokumentklassen und Seitenstilen definieren Sie 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 `.tex`-Datei steht, ist der Befehl `\documentclass`. Dieser legt im Parameter `Dokumentklasse` die Dokumentklasse für das gesamte Dokument. Die Syntax des Befehls lautet wie folgt:

`\documentclass[Optionen]{Dokumentklasse}`

Mehr als eine Dokumentklasse pro Dokument ist unzulässig. Moderne LaTeX-Distributionen bringen eine große Anzahl an Dokumentklassen mit. Dieses Dokument kann aus Platzgründen nicht alle davon beschreiben. Darum 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, sowie auf den neueren KOMA-Klassen [25], 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`, `slides` und `letter`.

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 `slides` eignet sich für Folien. Eine modernere Alternative ist `beamer`.

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.

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. Wie mehrere Optionen angegeben werden, 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  $\text{\LaTeX}$  standardmäßig die Schriftgröße **10pt** wählen.

**onecolumn oder twocolumn**

Diese beiden Optionen dienen der Festlegung der Seitenformatierung. Zur Auswahl stehen einspaltiges und zweispaltiges Seitenlayout. Standardmäßig verwendet  $\text{\LaTeX}$  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 der Titel horizontal zentriert auf der ersten Dokumentseite oben erscheint, 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<sup>A</sup>T<sub>E</sub>X 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  $\text{\LaTeX}$  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]		
<b>a4paper</b>	297	x	210	11,7	x	8,3
<b>a5paper</b>	210	x	148	8,3	x	5,8
<b>b5paper</b>	250	x	176	9,8	x	6,9
<b>letterpaper</b>	279,4	x	215,9	11	x	8,5
<b>legalpaper</b>	355,6	x	215,9	14	x	8,5
<b>executivepaper</b>	266,7	x	184,1	10,5	x	7,25

**landscape**

Mit der Option **landscape** wird das Dokument im Querformat gesetzt. Diese Option ist besonders dann sinnvoll, wenn Aushänge mit breiten Bildern oder Tabellen gesetzt werden sollen.

**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  $\text{\LaTeX}$  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 **openright** können Autoren festlegen, dass Kapitel 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 fest. Der gewählte Seitenstil gilt dann für das gesamte Dokument.

`\pagestyle{Seitenstil}`

L<sup>A</sup>T<sub>E</sub>X bietet einige vorgefertigte Seitenstile (siehe Tabelle 3.2).

Tabelle 3.2: Seitenstile

Seitenstil	Aussehen
<b>empty</b>	Die Kopf- und Fußzeilen bleiben leer und es wird keine Seitennummer gesetzt. Nur das Textfeld ist zu sehen.
<b>plain</b>	Die Kopfzeile bleibt leer. In der Fußzeile wird die Seitennummer zentriert ausgegeben. Dieser Seitenstil ist die Standardauswahl von L <sup>A</sup> T <sub>E</sub> X, wenn kein anderer Seitenstil mit <code>\pagestyle</code> ausgewählt ist.
<b>headings</b>	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.
<b>myheadings</b>	Der einzige Unterschied zum Seitenstil <b>headings</b> ist, dass die Kopfzeile nicht automatisch gefüllt wird, sondern das 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 Seitenstile `headings` oder `myheadings` gesetzt, können Autoren mit Hilfe der Befehle `\markright` und `markboth` die Inhalte der Kopfzeilen im Dokument definieren<sup>1</sup>. und damit die Standardeinstellung (siehe Tabelle 3.3) von L<sup>A</sup>T<sub>E</sub>X zu überschreiben.

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 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 der Bearbeitungsklasse `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 der Bearbeitungsklasse `book` oder `article` wird von `\markright` standardmäßig die Überschrift des aktuellen Kapitels (`\chapter`) in die Kopfzeile der rechten Seiten gesetzt:

<sup>1</sup>Die Wirkung der beiden Befehle `\markright` und `\markboth` setzt erst ab der zweiten Seite des Dokuments ein.

```
\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 [24]

Stil	Befehl	Dokumentklasse	
		<code>book, report</code>	<code>article</code>
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 beiden folgenden Befehlen realisiert werden:

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

Befinden sich auf einer Seite mehrere `\section`- oder `\subsection`-Befehle, fügt L<sup>A</sup>T<sub>E</sub>X 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 auch 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 [26]

<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 [26]

Um **fancyhdr** zu verwenden, muss das Erweiterungspaket in der Präambel der L<sup>A</sup>T<sub>E</sub>X-Quelldatei mit dem Befehl `\usepackage{fancyheadings}` eingebunden und der Seitenstil **fancy** mit dem Befehl `\pagestyle{fancy}` ausgewählt sein.

```
\pagestyle{fancy}
```

Die Befehle, zur Definition in den Abbildungen 3.1 und 3.2 dargestellten Felder der Kopf- bzw. Fußzeilen auf den gerade und ungeraden Seiten geschieht mit den folgenden Befehlen:

```

\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  $\text{\LaTeX}$ -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 Linien unter der Kopfzeile und über der Fußzeile wird mit den Befehlen `\headrulewidth` und `\footrulewidth` beeinflusst. Auch diese beiden Befehle müssen in der Präambel der  $\text{\LaTeX}$ -Quelldatei stehen.

```

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

```

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`.

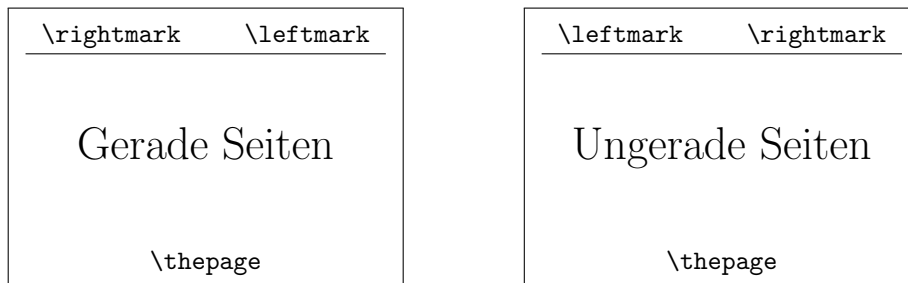


Abbildung 3.3: Voreingestelltes Seitenlayout bei `fancyhdr` [26]

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

L<sup>A</sup>T<sub>E</sub>X 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 das Festlegen des Seitennummerierungsstils.

```
\pagenumbering{Stil}
```

Weitere, von L<sup>A</sup>T<sub>E</sub>X 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 für die Seitenzahlen `page!`. Das bedeutet, auch wenn die Seitenzahlen in Buchstaben angezeigt werden, rechnet L<sup>A</sup>T<sub>E</sub>X intern ganz normal mit Zahlen weiter.

Eine Änderung des Seitennummerierungsstils mit `\pagenumbering` setzt den L<sup>A</sup>T<sub>E</sub>X-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 wird 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

Den Abstand zwischen zwei aufeinander folgenden Zeilen heißt Zeilenabstand. Diesen definiert der Wert in der Variablen `\baselineskip`. Bei  $\text{\LaTeX}$  wird der Zeilenabstand automatisch festgelegt und hängt von der aktuellen Schriftgröße ab.

Um den Zeilenabstand zu ändern, sollte aber 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  $\text{\LaTeX}$  nach sich zieht.

- Wird `\baselineskip` innerhalb eines Absatzes geändert, gilt das rückwirkend für den gesamten Absatz.
- Wird innerhalb eines Absatzes mehrmals mit `\baselineskip` den Zeilenabstand geändert, gilt nur die letzte Änderung im Absatz, da diese als letzte rückwirkend gilt und alle anderen 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 Längenbefehl `\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 L<sup>A</sup>T<sub>E</sub>X zwischen zwei Absätzen setzt, hängt vom Wert des Längenbefehls `\parskip` ab.

Es ist ratsam, für die Definition des Absatzabstands die Maßeinheit `ex` (siehe Seite 9) zu verwenden, da diese sich an der aktuellen Zeichengröße orientiert und den Absatzabstand als elastisches Maß (siehe Seite 10) zu definieren, damit L<sup>A</sup>T<sub>E</sub>X das Entstehen von *Hurenkindern* und *Schusterjunge* 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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X die erste Zeile eines jeden Absatzes nach rechts einrückt, ist im Längenbefehl `\parindent` gespeichert.

Das Einrückverhalten von L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X-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. Zudem existiert der Befehl `\indent`, der das Einrücken nur des nächsten Absatzes anweist.

`\noindent`

`\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*).

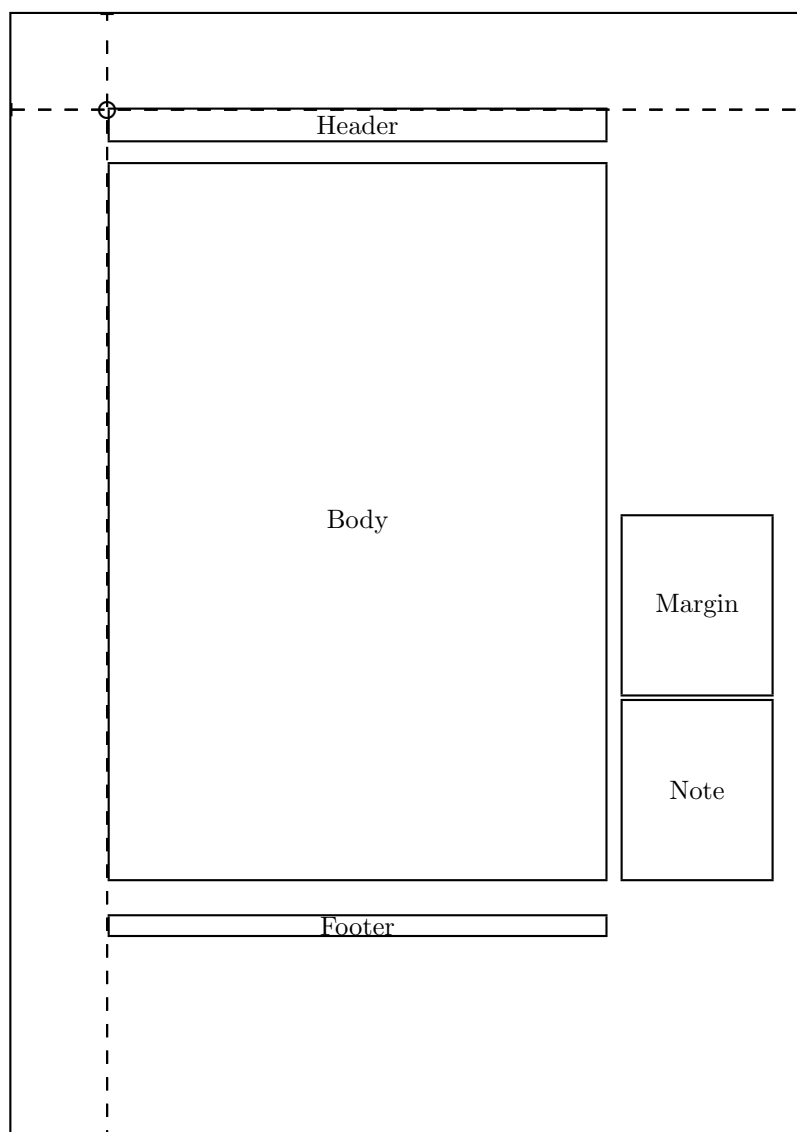
Das Zuweisen neuer Werte zu den Längenbefehlen in Tabelle 3.5 erfolgt mit dem Befehl `\setlength` (siehe Seite 30).

Tabelle 3.5: Längenbefehle zur Beeinflussung des Layouts [26]

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 Druck)
<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 Druck)
<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

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 `\currentpage` die aktuellen Werte der Längenbefehle auslesen, legen Sie mit `\oddpagelayouttrue` oder mit `\oddpagelayoutfalse` fest, ob Sie das Layout einer ungeraden oder einer geraden Seiten ausgeben wollen und setzen die Übersicht mit `\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

## 3.8 Mehrspaltenlayout

Soll ein Text 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  $\text{\LaTeX}$  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 den beiden 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.

```
\onecolumn
```

Soll wieder auf ein einspaltiges Layout umgeschaltete werden, geschieht dieses mit dem Befehl `\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 wird auch *Hochformat* oder *Portrait* genannt. 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 (*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, wo der Befehl `\portrait` oder der Befehl `\landscape` aufgerufen wird, beginnt  $\text{\LaTeX}$  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, dann ist es bequemer, die `\landscape`-Umgebung zu nutzen, als für jede Seite den `\landscape`-Befehl 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  $\text{\LaTeX}$  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 der die Namen und Adresse des Autors oder auch mehrerer Autoren definiert werden. Gibt es mehr als einen Autor, so ist es möglich, deren Namen mit dem Befehl `\and` zu verknüpfen.

```
\author{Text}
```

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 \\Adresse des \\zweiten Autors}
```

Auf der Titelseite würde dieses Beispiel zu folgendem Ergebnis führen:

erster Autor  
Adresse des  
ersten Autors

zweiter Autor  
Adresse des  
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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X.

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 das Dokument 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<sup>A</sup>T<sub>E</sub>X zu setzen, existiert die Umgebung `abstract`:

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



In der Dokumentklasse `book` ist die `abstract`-Umgebung 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 [24].

Bei der Dokumentklasse `article` wird die Zusammenfassung auf der ersten Seite unter der Überschrift gesetzt. Wurde die Option `\titlepage` bei der Deklaration der Dokumentklasse verwendet, erscheint die Zusammenfassung 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

L<sup>A</sup>T<sub>E</sub>X enthält für die fortlaufende Untergliederung von Dokumenten in Teile, Kapitel, Abschnitte, Paragraphen usw. entsprechende Befehle

Tabelle 3.6: Gliederungsebenen und der Zähler `tocdepth` [27]

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

Tabelle 3.6 enthält eine Zusammenstellung der unter L<sup>A</sup>T<sub>E</sub>X verfügbaren Gliederungsbefehle und der zugehörigen Werte des Zählers `tocdepth`.

Alle diese Befehle zur Untergliederung von Dokumenten, mit Ausnahme von `\part`, 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 *Unterabschnitt* (`\subsection`) und so weiter [24].

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 dann anstatt der langen Überschrift im Inhaltsverzeichnis und in der Kopfzeile verwendet.

`\Gliederungsbefehl[Kurzform]{Überschrift}`

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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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. Wird in einem Dokument das Erweiterungspaket `german.sty` oder `ngerman.sty` eingebunden, setzt L<sup>A</sup>T<sub>E</sub>X anstatt **Chapter x** das deutsche **Kapitel x**.

Tabelle 3.7 enthält eine Übersicht, wie das Erweiterungspaket (n)german.sty die Bezeichnungen und Überschriften der Gliederungselemente eines L<sup>A</sup>T<sub>E</sub>X-Dokuments 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. Die Syntax ist in diesem Fall wie folgt:

`\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.

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

### 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  $\text{\LaTeX}$  der Zähler `secnumdepth` und eine Änderung seines Werts kann mit folgendem Wert erfolgen.

`\setcounter{secnumdepth}{Zahl}`

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

### 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}` zu 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  $\text{\LaTeX}$ -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  $\text{\LaTeX}$  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 unter L<sup>A</sup>T<sub>E</sub>X der Befehl `\appendix`.

`\appendix`

An der Stelle im Quelltext, wo dieser Befehl aufgerufen wird, setzt L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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. `\textbf{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 L<sup>A</sup>T<sub>E</sub>X-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 L<sup>A</sup>T<sub>E</sub>X werden Änderungen in der Struktur (Gliederung) des Dokuments sich erst nach zwei Durchläufen des L<sup>A</sup>T<sub>E</sub>X-Compiler im Inhaltsverzeichnis niederschlagen.

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  $\text{\LaTeX}$ -Compiler zwei mal das Dokument durchlaufen. Beim ersten Durchlauf durch das Dokument wird  $\text{\LaTeX}$  durch den Befehl `\tableofcontents` angewiesen, 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.

Die Syntax des Befehls `\addcontentsline`, der das Einfügen eigener Einträge ins Inhaltsverzeichnis ermöglicht, ist folgendermaßen: `\listoftables`

`\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, den  $\text{\LaTeX}$  vor die Seitennummer schreibt.

## 3.12 Abbildungs- und Tabellenverzeichnis

Bei umfangreichen Dokumenten mit vielen sind ein Abbildungsverzeichnis und ein Tabellenverzeichnis 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 Sonderzeichen aus Tabelle 2.2 sollte 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 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 unter L<sup>A</sup>T<sub>E</sub>X weitgehend automatisiert. Hierfür muss das Erweiterungspaket `makeidx` mit dem Befehl `\usepackage{makeidx}` in der Präambel der `.tex`-Quelldatei eingebunden sein. Zudem muss im Quelltext des Dokuments an den entsprechenden Stellen die Einträge für das Indexregister mit dem Befehl `\item` definiert sein.

```
\item{Indexeintrag}
```

Zudem 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 L<sup>A</sup>T<sub>E</sub>X-Compiler abgearbeitet wurde...

**1) `pdflatex dateiname.tex`**

erzeugt L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X-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 Möglichkeiten aller Art, 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

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

L<sup>A</sup>T<sub>E</sub>X interpretiert einen Punkt, Ausrufezeichen, Fragezeichen oder Doppelpunkt, der nach einem Kleinbuchstaben oder einer Ziffer kommt, als das Ende eines Satzes. Nach dem Punkt fügt L<sup>A</sup>T<sub>E</sub>X 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 (~) oder mit dem Befehl \\_ ein Wortabstand eingefügt werden, der kleiner ist als der übliche Abstand zwischen Sätzen. \\_ steht an dieser Stelle für einen Backslash, direkt gefolgt von einem Leerzeichen.

~

\\_

Der Abstand, den ~ und \\_ realisieren, ist identisch. Der einzige Unterschied zwischen beiden Vorgehensweisen ist, dass bei der Verwendung von ~ 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 auch 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 sollte es nur in Ausnahmefällen verfolgen, weil es dem Prinzip der Trennung von Layout und Inhalt widerspricht. Zudem besteht die Gefahr, dass beim weiteren Einfügen von Texten, Bildern oder Tabellen im Dokument diese festen Abstände sich verschieben und dann den optischen Eindruck negativ beeinflussen.

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

`\hspace{Abstand}`

Bei einem Zeilenumbruch oder ganz am Anfang einer Zeile wird `\hspace` von  $\text{\LaTeX}$  ignoriert.

Eine Variante des Befehls `\hspace` ist der Befehl `\hspace*`. Dieser fügt im Gegensatz zu `\hspace` den Zwischenraum auch 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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 <del>Text</del> 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.

<code>\vspace{Abstand}</code>
-------------------------------

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 L<sup>A</sup>T<sub>E</sub>X 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 hier auch 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<sup>A</sup>T<sub>E</sub>X ein, wenn es dazu mit dem Befehl `\nonfrenchspacing` angewiesen wird.

`\nonfrenchspacing`

Die Wirksamkeit des Befehls geht 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 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 übrigens 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  $\text{\LaTeX}$  geschieht einfach durch die direkte Eingabe mit der Leertaste.

Stehen im Quelltext mehrere Leerzeichen hintereinander, wird  $\text{\LaTeX}$  dieses ignorieren und nur ein einziges Leerzeichen ausgeben.

Ein Leerzeichen kann auch mit Hilfe des Befehls `\textvisiblespace` erzeugt 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, können die Befehle `\hrulefill` und `\dotfill` Zwischenräume mit einer durchgezogenen bzw. mit einer gepunkteten Linie zu füllen.

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.

```
\dotfill
```

Mit diesen einfachen Möglichkeiten, durchgezogene und gepunktete Linien zu erzeugen und Abstände zu beeinflussen, ist es bereits 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 dieses 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 Einhaltung der Grenzen des Textfeldes

Je nach dem Layout des Dokuments und dem zu setzenden Text kann es passieren, dass der  $\text{\LaTeX}$ -Compiler einzelne Wörter über das Textfeld herausragen lässt oder Wortabstände sehr groß erscheinen. In diesem Zusammenhang sind auch Warnungen des  $\text{\LaTeX}$ -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, dass  $\text{\LaTeX}$  bereit ist, Wortabstände wenn nötig weiter zu dehnen als das üblich der Fall ist.

Dem Autor dieser Zeilen ist bewusst, dass der Befehl `\sloppy` nicht nur Fans hat, sondern auch an einigen Stellen von seiner Verwendung abgeraten wird [28]. Diese Ablehnung teilt der Autor dieses Werkes aber explizit nicht.

### 4.1.9 Trennungsregeln

Bei Verwendung der Erweiterungspakete `german` oder `ngerman` hält sich der  $\text{\LaTeX}$ -Compiler an die deutschen Trennungsregeln. Gerade bei Fremdwörtern kann es jedoch vorkommen, dass 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  $\text{\LaTeX}$ -Compiler berücksichtigt, wenn das Wort am Ende einer Zeile steht und getrennt werden muss. Ansonsten sind im fertigen Dokument die im Quelltext eingefügten Trennungshilfen nicht erkennbar.

Kommen ein oder mehr Wörter mit aus Sicht von  $\text{\LaTeX}$  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  $\text{\LaTeX}$ -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  $\text{\LaTeX}$  Texte in 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, ist 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. in Form einer 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 Text beidseitig gleich weit eingerückt ist, um ihn dadurch hervorzuheben. Zu diesem Zweck existierenden die beiden Umgebungen `quote` und alternativ `quotation`.

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

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

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

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac.

Vestibulum quis pellentesque diam. Aliquam erat volutpat.

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

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

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

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

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

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Curabitur ullamcorper lacus ante, vitae bibendum odio sodales ac.

Vestibulum quis pellentesque diam. Aliquam erat volutpat.

## 4.3 Zeilenumbrüche

Ein Zeilenumbruch kann an beliebiger Stelle im Text 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* von  $\text{\LaTeX}$  ignoriert und die nächste Seite beginnt mit der nächsten Textzeile.

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

```
\[Abstand]
\*[Abstand]
```

Enthält ein Dokument beispielsweise den Befehl `\[1cm]`, dann wird  $\text{\LaTeX}$  einen Zeilenumbruch mit einem vertikalen Zeilenabstand von 1 cm einfügen. Kommt durch diesen Befehl ein Seitenumbruch zustande, dann wird  $\text{\LaTeX}$  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 von  $\text{\LaTeX}$  selbst vorgenommen.  $\text{\LaTeX}$  kümmert sich dabei darum, dass eventuell vorhandene Gleitobjekte wie Bilder und Tabellen an die geeignetste Position kommen, und 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 aktuellen Seite bei einem zweispaltigen Seitenlayout (Klassenoption `twocolumn` oder Befehl `\twocolumn`) geschieht mit dem Befehl `\clearpage`.

`\clearpage`

Bei einem doppelseitigem Layout (Klassenoption `twoside`) geschieht das manuelle Umbrechen der aktuellen 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, wie es möglich ist, 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 Wechseln des Schriftschnitts her.

```
\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 dann natürlich 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 horizontale 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, das L<sup>A</sup>T<sub>E</sub>X 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

Die in Tabelle 5.1 dargestellten Befehle zur Veränderung der Schriftgröße 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 festgelegt 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<sup>A</sup>T<sub>E</sub>X-Quelltext um. Die Wirkung dieser Befehle durch die aktuelle Gruppe begrenzt. Ein Aufruf eines dieser Befehle in der Präambel eines Dokuments, beeinflusst das gesamte Dokument.



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>	<b>schaltet auf fette Schrift</b>

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>		<b>Fette Schrift</b>

## 5.4 Aufzählungen

Um Aufzählungen zu erzeugen, enthält L<sup>A</sup>T<sub>E</sub>X die 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 von Ihnen 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 verschaltet werden. Mehr als vier Ebenen sind aber nicht zulässig.

Bei der Umgebung `itemize` ist für jede Ebene ein Markierungszeichen voreingestellt. In der folgenden kleinen Übersicht sehen Sie, 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 Numerierung 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}
\item Beim Verschachteln...
\end{enumerate}
\end{enumerate}
\end{enumerate}

```

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

Die Umgebung `description` ist besonders gut geeignet 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}
```

Den Befehl `\footnote` schreibt man immer direkt – also ohne Leerzeichen – nach dem Wort, an das die Markierung der Fußnote angehängt sein soll. Der Text der Fußnote wird von  $\text{\LaTeX}$  in einer kleineren Schrift von der Größe `footnotesize` geschrieben. Standardmäßig rückt  $\text{\LaTeX}$  die erste Zeile einer Fußnote einen halben Zentimeter ein.

Zwischen den Fußnoten einer Seite und dem gewöhnlichen Text des Dokuments fügt  $\text{\LaTeX}$  automatisch eine horizontale Linie ein, um die Fußnoten deutlich vom Text abzuheben.

Der Befehl `\footnote` darf nicht innerhalb von mathematischen Umgebungen oder Tabellen aufgerufen werden. Es gibt aber verschiedene Tricks, um sich in einem solchen Fall zu behelfen. Eine Möglichkeit, um Fußnoten innerhalb 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

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

Randbemerkungen, die auch *Marginalien* genannt werden, 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}`

## 5.7 Unformatierter Text

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

Unformatierter Text kann u.a. mit der Umgebung `verbatim` gesetzt werden.

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

Text, der sich innerhalb dieser Umgebung befindet, wird von  $\text{\LaTeX}$  in `TypeWriter`, also `Schreibmaschinenschrift` gesetzt und nicht verändert oder interpretiert. Das heißt, dass dieser Text auch alle Sonderzeichen von  $\text{\LaTeX}$  und beliebige Kombinationen von Leerzeichen und Zeilenumbrüche enthalten kann.

Zu Beginn und Ende der Umgebung `verbatim` fügt  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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 umrahmen den 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 kann.

Ausgefallener Boxentypen bietet das Erweiterungspaket `fancybox`. Wird es mit dem Befehl `\usepackage{fancybox}` in der Präambel der `.tex`-Quelldatei eingebunden, stehen folgende Boxentypen zur Verfügung:

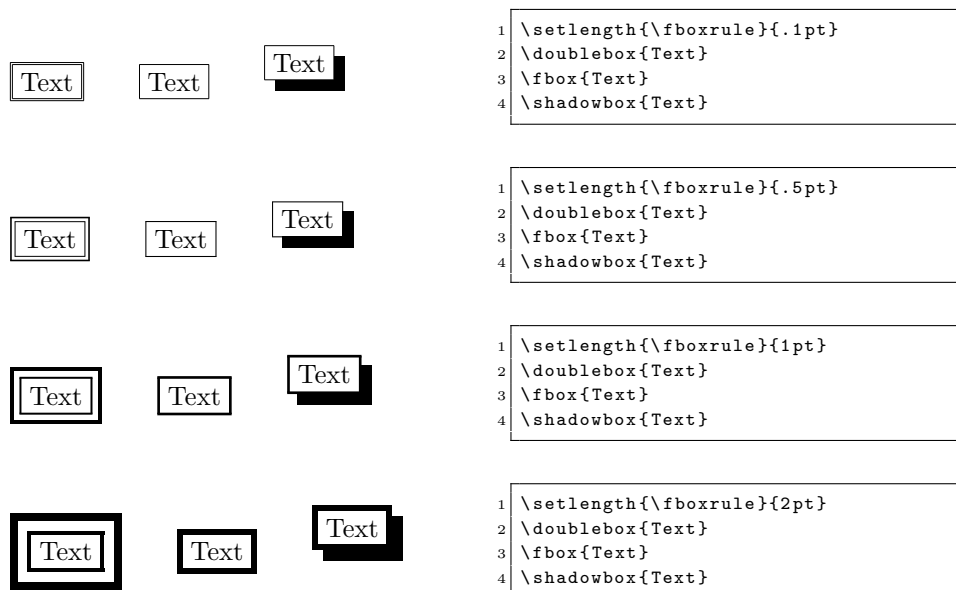


```
\doublebox{Text}
\ovalbox{Text}
\Ovalbox{Text}
\shadowbox{Text}
```

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

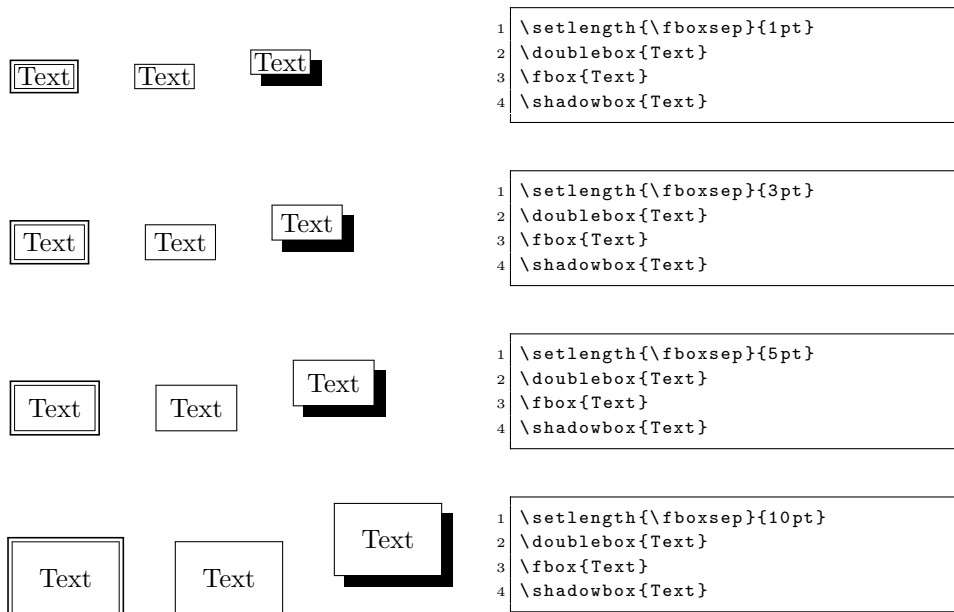
Bei einer `doublebox` ist die Linienstärke des äußeren Rahmens `1.5\fboxrule`, also das Anderthalbfache von `\fboxrule`, und die Linienstärke des inneren Rahmens ist `.75\fboxrule`.

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 Standardwert ist in der Datei `latex.ltx` definiert und hat den Wert `3pt`.

Die folgenden Beispiele zeigen die Auswirkungen der Änderung des Längenbefehls `\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>	<code>.4pt</code>
<code>\fboxsep</code>	Definiert den Abstand zwischen Inhalt und Rahmen	<code>3tp</code>
<code>\shadowsize</code>	Definiert die Breite des Schattens bei <code>\shadowbox</code>	<code>4tp</code>
<code>\thinlines</code>	Definiert die Linienstärke bei <code>\ovalbox</code>	
<code>\thicklines</code>	Definiert die Linienstärke bei <code>\Ovalbox</code>	

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`.

Dem erste optionalen Argument **Ausrichtung**, definiert die Ausrichtung der Umgebung anhand der aktuellen Zeil. 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. [29] Der Standardwert ist **c**.

Das optionale Argument **Höhe** definiert die Höhe als Absolutwert inklusive einer Maßeinheit (z.B. 5cm) als relativer 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. L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X bietet mehrere Umgebungen zum Satz von Tabellen. Aus Platzgründen kann dieses Dokument 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<sup>A</sup>T<sub>E</sub>X 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. Enthält *Position* den Wert `t`, dann wird die oberste Zeile der Tabelle an der laufenden Umgebung ausgerichtet. Der Wert `b` hingegen sorgt dafür, dass L<sup>A</sup>T<sub>E</sub>X die Tabelle mit der untersten Zeile an der laufenden Umgebung ausrichtet. 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. Wenn der Inhalt eines Feldes die definierte Spaltenbreite übersteigt, wird dieser in mehrere Zeilen gebrochen. 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.

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 ist ein einfaches Beispiel für eine Tabelle mit drei Spalten. Der Quelltext von Tabelle 6.2 zeigt Listing 6.1. In der Spaltendeklaration in 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: Das allererste Beispiel

Tabelle 6.1: Mögliche Einträge im Parameter *Spalten*

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

Tabelle 6.2: Eine einfache Tabelle

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

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 sind durch vertikale und horizontale Linien voneinander abgegrenzt. Den Quelltext von Tabelle 6.3 zeigt 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 \\
\end{pre>
```

```

6 | x1y2 & x2y2 & x3y2 \\
7 | x1y3 & x2y3 & x3y3 \\
8 | \hline
9 | \end{tabular}

```

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

In der Spaltendeklaration in der Tabellenpräambel sind nun 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 und mit weniger horizontalen Linien wirkt die Tabelle eleganter

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|`, dann 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.  $\Rightarrow$ ), kann dieses einfach mit dem Spaltenformatierungseintrag `!Text` angewiesen werden.

Ein Beispiel für den sinnvollen Einsatz des Eintrags `!\{Text\}` in der Spaltendeklaration 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 bestimmten Spalten einfügen

$$\begin{array}{rcl} (a+b)^2 & = & a^2 + 2ab + b^2 \\ (a-b)^2 & = & a^2 - 2ab + b^2 \\ (a+b) * (a-b) & = & a^2 - b^2 \end{array}$$

```

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

```

Listing 6.4: Anstatt einer zusätzlichen Spalte nur für das Gleichheitszeichen führt `!\{Text\}` in der Spaltendeklaration zum gleichen Ergebnis

## 6.2 Die Umgebung `tabularx`

Außer der Umgebung `tabular` bietet L<sup>A</sup>T<sub>E</sub>X zahlreiche weitere Umgebungen, um Tabellen zu setzen, die für spezielle Anwendungsfälle 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 L<sup>A</sup>T<sub>E</sub>X 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.

```

1 \begin{tabularx}{\textwidth}{|X|X|X|}
2 \hline
3 Spalte 1 & Spalte 2 & Spalte 3 \\
4 \hline\hline
5 x1y1 & x2y1 & x3y1

```

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

```

6 | x1y2 & x2y2 & x3y2 \\
7 | x1y3 & x2y3 & x3y3 \\
8 | \hline
9 | \end{tabularx}

```

Listing 6.5: Die Umgebung `tabularx` ermöglicht Tabellen mit definierbarer Breite und automatischer Spaltenbreite

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 Textfeldes im Dokument. Alternativ könnte beispielsweise eine 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, wird das durch 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.

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



Tabelle 6.7: Spalten mit automatischer Breite bei `tabularx`

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

```
1 \newcolumntype{Y}{>\centering\arraybackslash}X}
2 \newcolumntype{Z}{>\hfill\arraybackslash}X}
3
4 ...
5
6 \begin{tabularx}{.8\textwidth}{|X|Y|Z|}
7 \hline
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: Die Umgebung `tabularx` ermöglicht Tabellen mit definierbarer Breite und automatischer Spaltenbreite

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}
```

```

7 x1y3 & x2y3 & x3y3 \\
8 \hline
9 \end{tabularx}

```

Listing 6.7: Auch in der Umgebung `tabularx` sind die Spaltenspezifikationen von `tabular` erlaubt

## 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 Erweiterungspaket `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 `cmyk`, `HTML`, `gray`, `rgb` und `RGB`. Das zwingend erforderliche Argument `Farbe` definiert eine Farbe innerhalb des gewählten Farbmodells. Eine Übersicht über die von `LATEX` unterstützten Farbmodelle enthält Tabelle 6.9.

Ein Beispiel zum Befehl `\rowcolor` enthält Tabelle 6.10 und der zugehörige Quelltext in Listing 6.8. In diesem Beispiel wurden einzelne Zeilen mit dem Farbmodell `gray` eingefärbt.

```

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 Zeilen einer Tabelle geschieht mit dem Befehl `rowcolor`

Tabelle 6.9: Übersicht über die unterstützten Farbmodelle

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

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

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 diesen können Autoren festlegen, wie weit die Färbung links und rechts über den Inhalt der Spalte hinausragt. Da diese beiden Argumente in der Praxis eher selten eingesetzt werden, werden Sie in diesem Werk nicht weiter betrachtet.

Ein Beispiel zum Befehl `\columncolor` enthält Tabelle 6.11 und der zugehörige Quelltext in Listing 6.9. Auch in diesem Beispiel wurden einzelne Zeilen mit dem Farbmodell `gray` eingefärbt. Der Befehl `\columncolor` wird in der Spaltendeklaration für 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 Spalten einer Tabelle geschieht mit dem 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 und der zugehörige Quelltext in Listing 6.10.

```

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 &

```

Tabelle 6.12: Einfärben einzelner Felder einer Tabelle

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

```

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 ganzer Felder (Zellen) einer Tabelle geschieht mit dem Befehl `cellcolor`

Wird eine Farbe häufiger im Dokument verwendet, ist es sinnvoll, diese Farbe im Präambel des Quelltextes mit dem Befehl `\definecolor` 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 wird mit dem Befehl `\definecolor` die Farbe Deep Sky Blue definiert.

```

\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 wird an dieser Stelle exemplarisch ein leichtes Grau definiert.

```

\definecolor{lightgray}{gray}{0.6}

```

Anschließend kann einfach via `rowcolor{deepskyblue}`, `\columncolor{deepskyblue}` oder `\cellcolor{deepskyblue}` auf die definierte Farbe zugegriffen werden.

## 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 Positionierungszeichen `l`, `r` oder `c` definiert, ob der Inhalt des neuen Felds linksbündig, rechtsbündig oder zentriert gesetzt wird. Auch vertikale Linien können hier mit `|` oder `||` angegeben werden.

Ein Beispiel zum Befehl `\multicolumn` enthält Tabelle 6.13 und der zugehörige Quelltext 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 Positionierungszeichen. 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<sup>A</sup>T<sub>E</sub>X die Breite automatisch anhand des Inhalts fest.

Ein Beispiel zum Befehl `\multirow` enthält Abbildung 6.1.

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}

```

Abbildung 6.1: Felder über die Grenzen von Zeilen zusammenfassen

## 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 der 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 enthält Abbildung 6.2. Wäre in diesem Beispiel die Linie von der zweiten bis zur dritten Spalte anstatt mit dem Befehl mit realisiert worden, hätte die horizontale Linie das zusammengefasste Feld in der ersten Spalte durchschnitten.

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}

```

Abbildung 6.2: 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.

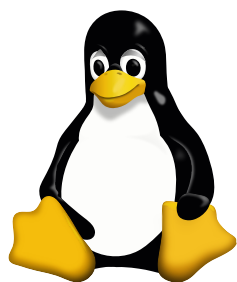
Das beste optische Ergebnis liefern Abbildungen, die als Vektorgrafiken vorliegen. Besonders bei selbst erstellten Diagrammen ist das problemlos möglich. Im Gegensatz zu Rastergrafiken ist das skalieren von Vektorgrafiken ohne Qualitätsverlust stufenlos und verlustfrei möglich. Dateiformate für Vektorgrafiken 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 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 `width`

Soll eine Abbildung genauso breit sein wie das Textfeld, geht dieses einfach mit

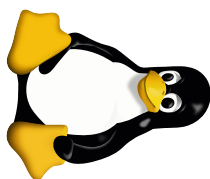
`width=\textwidth`. Das folgende Beispiel würde die Abbildung `bilddatei.pdf` so skalieren, dass die Bildbreite der halben Breite des Textes entspricht.

```
\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 die Option `angle`

Meist ist es sinnvoll, Abbildungen mit der Umgebung `figure` (siehe Abschnitt ??) 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{abb_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}` verwiesen werden.

Die Definition einer Bildunterschrift geschieht mit dem Befehl `\caption{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 werden. 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  $\text{\LaTeX}$ , 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  $\text{\LaTeX}$  im universitären Umfeld und im Verlagswesen.

Zum Formelsatz enthält  $\text{\LaTeX}$  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 vor und hinter die Formel ein  $\$$ -Zeichen zu stellen.

$\$Formel\$$

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 mit verschiedenen Erweiterungspaketen, gilt das Umschließen der Formel mit den Zeichenketten  $\backslash($  und  $\backslash)$ .

$\backslash(Formel\backslash)$

Zusätzlich existiert die Umgebung `math.`, deren Verwendung zum gleichen Ergebnis führt wie die Verwendung von  $\$...\$$  und  $\backslash(...\backslash)$ .

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

## 8.2 Abgesetzte Formeln

Auch zur Erzeugung abgesetzter Formeln existieren unter L<sup>A</sup>T<sub>E</sub>X verschiedene Möglichkeiten. Eine Möglichkeit ist, den Anfang und das Ende der abgesetzten Formel mit `$$` zu kennzeichnen. Aber auch hier gibt es wieder das Problem, dass beide Begrenzungs-symbole 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 das L<sup>A</sup>T<sub>E</sub>X 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.

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

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

Sollen Formeln nummeriert sein, um im Fließtext darauf Bezug nehmen zu können, ist es sinnvoll die abgesetzten Formeln mit der Umgebung `equation` zu erzeugen. In diesem Fall wird L<sup>A</sup>T<sub>E</sub>X die Formel automatisch mit einer eindeutigen Nummer versehen.

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

Zum Satz von Gleichungssystemen existieren die Umgebungen `eqnarray` und `eqnarray*`. Beide eignen sich für den Satz mehrerer untereinander angeordneter Formeln.

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 wie die Arbeitsweise mit 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 eine 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.

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

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 `eqnarray`-Umgebung einzelne bestimmte Zeilen nur keine Nummer erhalten, dann kann dieses mit dem Befehl `\nonumber` angewiesen werden.

```
\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 mit L<sup>A</sup>T<sub>E</sub>X 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 Befehl 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.

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

$$\begin{array}{l} 1 \backslash[ \\ 2 \quad \backslashfrac{a}{b} + \backslashfrac{c}{d} = \\ 3 \quad \backslashfrac{a\cdot d+c\cdot b}{b\cdot d} \\ 4 \backslash] \end{array}$$

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 L<sup>A</sup>T<sub>E</sub>X als Index der vorhergegangenen Teilformel gesetzt.

`_ {Index}`

Das Setzen von Exponenten geschieht mit dem Zeichen `^`. Das darauf folgende Zeichen (oder die darauf folgende Teilformel) setzt L<sup>A</sup>T<sub>E</sub>X 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.

Da L<sup>A</sup>T<sub>E</sub>X 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.

$a^2 + a^b - a_2 + a_i^n$	<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  $a_y^x$  auf diese Art und Weise so: `\(a^{\{x\}}_{\{y\}}\)` oder so: `\(a_{\{y\}}^{\{x\}}\)` realisiert ist.

$a^{n+1} + a^{n_x}$	<code>\[ a^{n+1} + a^{n_x} \]</code>
$a^{b^{c^2}}$	<code>\[ a^{b^{c^2}} \]</code>
$a^{b_i^2}$	<code>\[ a^{b_i^2} \]</code>
$c_{n+m}^{2n-1}$	<code>\[ c_{n+m}^{2n-1} \]</code>
$a_1^{e^{b^2}}$	<code>\[ a_1^{e^{b^2}} \]</code>

### 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` L<sup>A</sup>T<sub>E</sub>X automatisch fest, wie das Beispiel in Abbildung 8.2 anschaulich zeigt:

$\sqrt[1018]{-1} =$

```
1 \[\n2   \sqrt{1018}{-1}=n3   \sqrt{\sqrt{\sqrt{\sqrt{n4     \sqrt{\sqrt{\sqrt{\sqrt{n5       \sqrt{\sqrt{-1}}}}}}}}}n6 ]\]
```

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

Es folgen einige weitere Beispiele, die den Satz von Wurzel 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>
$\sqrt[13]{\frac{y+b\sin\theta}{e^{i\theta}}}$	<code>\[ \sqrt[13]{\frac{y+b\sin\theta}{e^{i\theta}}} \]</code>

### 8.3.4 Summen und Integrale

In diesem Abschnitt werden einige mathematische Zeichen, darunter unter anderem das Summen-, das Integral-, das Produktzeichen und weitere weitere Zeichen präsentiert. Genau wie bei dem 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 deutlich 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.

$$\boxed{\text{\texttt{\textbackslash sum\_}\{unter dem Zeichen\}^{\{über dem Zeichen\}}}}$$

Die eingebettete Formel  $\sum_{n=0}^x$  wurde mit folgenden L<sup>A</sup>T<sub>E</sub>X-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. Das dann schon ein wenig anders aus:

$$\sum_{n=0} \quad \quad \quad \text{\texttt{\textbackslash \sum_{n=0}^x}}$$

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 Art des Satzes mit dem Befehl `\displaystyle` erzwungen.

$$\boxed{\text{\texttt{\textbackslash displaystyle}}}$$

Tabelle 8.1: Größen 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 von *unten* unter den Inhalt von *oben*.

Beide Befehle sind Teil des Erweiterungspakets `amsmath` und stehen zur Verfügung, sobald das Paket in der Präambel der L<sup>A</sup>T<sub>E</sub>X-Quelldatei mit dem Befehl `\usepackage{amsmath}` eingebunden ist.

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

```

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

Bei der Summation 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 zahlreiche mathematische Zeichen und Symbole, deren Satz L<sup>A</sup>T<sub>E</sub>X ermöglicht.

### 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 ein eigener Befehl (siehe Tabelle 8.2 und Tabelle 8.3).

Tabelle 8.2: Griechische Großbuchstaben

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
$A$	<code>A</code>	$I$	<code>I</code>	$P$	<code>P</code>
$B$	<code>B</code>	$K$	<code>K</code>	$\Sigma$	<code>\Sigma</code>
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$T$	<code>T</code>
$\Delta$	<code>\Delta</code>	$M$	<code>M</code>	$\Upsilon$	<code>\Upsilon</code>
$E$	<code>E</code>	$N$	<code>N</code>	$\Phi$	<code>\Phi</code>
$Z$	<code>Z</code>	$\Xi$	<code>\Xi</code>	$X$	<code>X</code>
$H$	<code>H</code>	$O$	<code>O</code>	$\Psi$	<code>\Psi</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Omega$	<code>\Omega</code>

Tabelle 8.3: Griechische Kleinbuchstaben

Zeichen	Befehl	Zeichen	Befehl	Zeichen	Befehl
$\alpha$	<code>\alpha</code>	$\iota$	<code>\iota</code>	$\rho$	<code>\rho</code>
$\beta$	<code>\beta</code>	$\kappa$	<code>\kappa</code>	$\sigma$	<code>\sigma</code>
$\gamma$	<code>\gamma</code>	$\lambda$	<code>\lambda</code>	$\tau$	<code>\tau</code>
$\delta$	<code>\delta</code>	$\mu$	<code>\mu</code>	$\upsilon$	<code>\upsilon</code>
$\epsilon$	<code>\epsilon</code>	$\nu$	<code>\nu</code>	$\phi$	<code>\phi</code>
$\zeta$	<code>\zeta</code>	$\xi$	<code>\xi</code>	$\chi$	<code>\chi</code>
$\eta$	<code>\eta</code>	$o$	<code>o</code>	$\psi$	<code>\psi</code>
$\theta$	<code>\theta</code>	$\pi$	<code>\pi</code>	$\omega$	<code>\omega</code>

Wie in Tabelle 8.2 und Tabelle 8.3) zu sehen ist, setzt  $\text{\LaTeX}$  griechischen Großbuchstaben üblicherweise in der Schrift Roman und griechische Kleinbuchstaben in der geneigten Schrift *Italic* (siehe Abschnitt 5.3). Sollen auch Großbuchstaben in geneigter Schrift ausgegeben werden, kann dieses mit dem Befehl `\mathnormal` angewiesen werden.

`\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
$\varepsilon$	<code>\varepsilon</code>	$\varpi$	<code>\varpi</code>	$\varsigma$	<code>\varsigma</code>
$\vartheta$	<code>\vartheta</code>	$\varrho$	<code>\varrho</code>	$\varphi$	<code>\varphi</code>

### 8.4.2 Kalligrafische Buchstaben

$\text{\LaTeX}$  ermöglicht innerhalb mathematischer Umgebungen mit dem Befehl `\mathcal` den Satz von 26 *KALLIGRAFISCHE* Buchstaben.

`\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 der 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 einige binäre Operationssymbole, deren Satz mit  $\text{\LaTeX}$  möglich ist.

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  $\text{\LaTeX}$ -Quelldatei.

#### 8.4.4 Vergleichssymbole

Tabelle 8.7 zeigt einige Vergleichssymbole, deren Satz mit  $\text{\LaTeX}$  möglich ist.

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 8.7: Vergleichssymbole

Zeichen	Befehl	Zeichen	Befehl
$<$	<code>&lt;</code>	$\ni$	<code>\ni</code>
$>$	<code>&gt;</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>	$\sqsubseteq$	<code>\sqsubseteq</code>
$\equiv$	<code>\equiv</code>	$\sqsupseteq$	<code>\sqsupseteq</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>

Eine Übersicht der unter  $\text{\LaTeX}$  verfügbaren Negationen von Vergleichssymbolen enthält Tabelle 8.8.

Diejenigen Symbole, die mit einem \* markiert sind, sind nur nach einer Einbindung des Erweiterungspaketes `latexsym` verfügbar

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

### 8.4.5 Pfeile und Zeiger

Eine Übersicht über mögliche Pfeile, deren Satz  $\text{\LaTeX}$  ermöglicht, präsentiert Tabelle 8.10.

Diejenigen Symbole, die mit einem \* markiert sind, sind nur nach einer Einbindung des Erweiterungspaketes `latexsym` verfügbar



Tabelle 8.8: Negierte Vergleichssymbole

Zeichen	Befehl	Zeichen	Befehl
$\nless$	<code>\not&lt;</code>	$\nparallel$	<code>\not\parallel</code>
$\ngtr$	<code>\not&gt;</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>

Es existieren zwei Möglichkeiten, um den Doppelpfeil  $\iff$  zu erzeugen. Eine Möglichkeit, ist die Verwendung des aus Tabelle 8.10 bekannten Befehls `\Longlefttrightarrow`. Eine alternative Möglichkeit ist die Verwendung des Befehls `\iff`. Allerdings unterscheiden sich beide Befehle im Zwischenraum, den L<sup>A</sup>T<sub>E</sub>X vor und hinter dem Pfeil einfügt. Bei `\Longlefttrightarrow` ( $\iff$ ) ist der Zwischenraum etwas kleiner als bei `\iff`

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

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

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>	$\approx$	<code>\shortparallel</code>
$\circeq$	<code>\circeq</code>	$\smile$	<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>
$\leftarrow$	<code>\leftarrow</code> oder <code>\gets</code>	$\rightarrow$	<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>	$\nRrightarrow$	<code>\nRrightarrow</code>

### 8.4.6 Sonstige mathematische Zeichen

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

### 8.4.7 Mathematische Funktionen

In der Literatur werden mathematische Funktionen üblicherweise nicht kursiv geschrieben, wie die Namen von Variablen, sondern mit aufrechter Roman-Schrift gesetzt. Eine Übersicht über die mathematischen Funktionen, für die  $\text{\LaTeX}$  Befehle bereitstellt, zeigt Tabelle 8.14.

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>	$\partial$	<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>
$\infty$	<code>\infty</code>	$\surd$	<code>\surd</code>
$\ell$	<code>\ell</code>	$\jmath$	<code>\jmath</code>
$\top$	<code>\top</code>	$\emptyset$	<code>\emptyset</code>
$\nabla$	<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>
$\circledS$	<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>

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 von  $\text{\LaTeX}$ .

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>				

### 8.4.8 Akzente in Formeln

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

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>				

Doppelakzente sind mit den in Tabelle 8.15 gezeigten Befehlen auch machbar, wie Tabelle 8.16 demonstriert.

Zusätzlich zu den in diesem Abschnitt vorgestellten Akzenten, die L<sup>A</sup>T<sub>E</sub>X von Haus aus bereitstellt, stellt das Erweiterungspaket `amsmath` die Befehle `\ddddot` und `\ddddot` zur Verfügung, die dreifache bzw. vierfache Punktakzente erzeugen (siehe Tabelle 8.17).

### 8.4.9 Über- und Unterstreichungen

Zum Satz verschiedener Über- und Unterstreichungen existiert eine Reihe von Befehlen, die Tabelle 8.18 präsentiert. Überstreichungen realisieren die Befehle `\overline`,

Tabelle 8.16: Doppelakzente in Formeln

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

Tabelle 8.17: Dreifache bzw. vierfache Punktakzente

Zeichen	Befehl	Zeichen	Befehl
$\dddot{a}$	<code>\dddot{a}</code>	$\ddddot{a}$	<code>\ddddot{a}</code>

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

Die Breite der Über- und Unterstreichungssymbole wird von L<sup>A</sup>T<sub>E</sub>X automatisch festgelegt und ist abhängig von der Breite der zu über- oder unterstreichenden Inhalte. Es ist es auch kein Problem, die Befehle aus Tabelle 8.18 zu verschachteln, und so Symbole oder mathematische Inhalte mit mehreren dieser Über- oder Unterstreichungssymbole zu versehen.

$$\overbrace{\underbrace{\overbrace{abc\,\underline{\overbrace{def\,\overline{\overbrace{gih}}\,jkl}}\,}\,}\,}\,}^1$$

Mit Hilfe der Befehle aus Tabelle 8.18 ist es auch möglich, Texte oder mathematische Inhalte über bzw. *unter* Inhalten zu setzen.

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

```

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

```

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 L<sup>A</sup>T<sub>E</sub>X selbstständig fest.

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

```

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

```

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

```

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

```

$$\overrightarrow{qX_{\delta\gamma} \overleftarrow{aY_{\beta\gamma} h}}$$

```

1 \(\overrightarrow{qX_{\delta\gamma} \overleftarrow{aY_{\beta\gamma} h}}\)
2
3

```

### 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 8.19: 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>	$\uparrow$	$\Uparrow$	$\downarrow$	$\Downarrow$	$\updownarrow$	$\Updownarrow$

---

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.

### 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).



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

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

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

```

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

```

### 8.4.13 Mengensymbole

Der Satz der Mengensymbole geschieht mit dem 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 ganz „normaler“ Text innerhalb einer mathematischen Umgebung gesetzt werden, muss diese nicht zwingend verlassen werden. Die Einfachste Möglichkeit ist die Verwendung des Befehls `\mbox`.

`\mbox{Text}`

Der zu setzende Text wird dem Befehl in geschweiften Klammern übergeben.

$$M_i \cap M_j = \emptyset \quad \text{falls} \quad i \neq j$$

```

1 \left( M_i \cap M_j = \emptyset
2 \quad \quad \quad \mbox{falls} \quad i \neq j \right)

```

Wie das obige Beispiel zeigt, kann es nützlich sein, um die `\mbox` einen horizontale Abstände einzufügen. Eine Möglichkeit, solche Abstände einzufügen, ist die Verwendung der Befehle `\quad` und `\qquad`.

`\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 ihm in geschweiften Klammern als Argument übergeben wird.

`\boxed{Inhalte}`

Der Befehl `\boxed` verhält sich im Grunde genau so wie der Befehl `\fbox` (siehe Abschnitt 5.9). Der einzige Unterschied ist, dass `\boxed` nur in mathematischen Umgebungen funktioniert.

$$\sqrt[\alpha]{(a * b * c)^2}$$

```
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 \(\
2 \frac{-2^{\{3^{\{2\}}\}} + (\{2^{\{3\}}\})^{\{2\}} x}{(\{2^{\{2\}}\})^{\{3\}}}
3 \)
```

$$\sqrt[3]{\left(\frac{x^4 - 3x^2}{(x-7)} + e^{x^2}\right)^2 \sqrt{\sin(x^7 - x^3)}}^2$$

```
1 \(\
2 \sqrt[3]{\Bigl(\frac{\sqrt[3]{\{x^4 - 3x^2\}}{\{x-7\}} + e^{\{x^2\}}}{\sqrt{\sin(x^7 - x^3)}}\Bigr)^{\{2\}}}
3
4
5 \)
```

$$A(x) = \sum_{j=0}^{n-1} b_j y^j$$

```

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)!}
3 \ )

```

$$\sum_{k=1}^n e^{ikx} = e^{ix} * \frac{1 - e^{inx}}{1 - e^{ix}} = \frac{\sin \frac{nx}{2}}{\sin \frac{x}{2}} * e^{i(n+1)\frac{x}{2}} \quad (x \neq 2n\pi)$$

```

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 \ )

```

$$s \in G \implies \langle s \rangle \subseteq G \quad \text{ist Untergruppe} \quad \xRightarrow{\text{Satz}} \quad |\langle s \rangle| \mid |G|$$

```
1 \[
2 s\in G \Longrightarrow \langle s\rangle\subseteq
3 G\quad\text{ist Untergruppe}\quad\stackrel{\text{Satz}}{\Longrightarrow}
4 {\Longrightarrow}\quad |\langle s\rangle|\leq |G|
5 \]
```



## 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 [30] 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
<b>Beispiel für einen Brief mit der Klasse g-brief2</b>			
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			
<b>Anlagen:</b> Anhang Weiterer Anhang			
<b>Adresse:</b> Musterstraße 1 12345 Musterstadt	<b>Telefon:</b> +49-123-456789 +49-123-345678	<b>Internet:</b> max@mustermann.de <a href="http://www.mustermann.com">http://www.mustermann.com</a>	<b>Bankverbindung:</b> Musterbank Musterstadt IBAN DE12345678901234567890 BIC 12345678901

Abbildung 9.1: Resultat von Listing 9.1

## 9.2 Briefe mit der Dokumentklasse `scrlettr2`

Eine moderne Alternative zu zur Dokumentklasse `g-brief2` ist die Klasse `scrlettr2`. Diese ist Teil des  $\text{\LaTeX}$  2<sub>ε</sub>-Erweiterungspakets KOMA-Script. Die Klasse bietet sehr vielfältige Optionen zur Briefgestaltung und ist sehr umfangreich auch in deutscher Sprache dokumentiert [25].

```

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 ]{scrlettr2}
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](mailto: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<sup>A</sup>T<sub>E</sub>X die Schrift Computer Modern. Diese verleiht Dokumenten einen gewissen „L<sup>A</sup>T<sub>E</sub>X-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<sup>A</sup>T<sub>E</sub>X verwendet werden können, bietet der L<sup>A</sup>T<sub>E</sub>X Font Catalogue [31].

<http://www.tug.dk/FontCatalogue/>

### 10.1 Altdeutsche Schriften

Dank der Arbeit von Yannis Haralambous [32] ist es auf recht einfache Art und Weise möglich, altdeutsche Schriften in L<sup>A</sup>T<sub>E</sub>X-Dokumenten zu verwenden.

Eine einfache Möglichkeit, um altdeutsche Schriften in L<sup>A</sup>T<sub>E</sub>X-Dokumenten zu verwenden, Erweiterungspaket **yfonts**. Dieses muss mit dem Befehl `\usepackage{yfonts}` in der Präambel der L<sup>A</sup>T<sub>E</sub>X-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<sup>A</sup>T<sub>E</sub>X komfortabel zu setzen.

Ein Beispiel ist das Erweiterungspaket **makebarcode** [33], das den Befehl `\barcode` definiert.

```
\barcode{Text}
```

Der folgende Befehl in der Präambel der  $\text{\LaTeX}$ -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-Codes<sup>1</sup> 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). [34] 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  $\text{\LaTeX}$  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 2.4) 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.

---

<sup>1</sup>Das 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. In diesem Fall wird ein einzelnes Führungszeichen 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  $\text{\LaTeX}$  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` [35] 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

!!! Fehlt noch !!!





## 12 Präsentationsfolien

!!! Fehlt noch !!!



# Literaturverzeichnis

- [1] S. Krause, “Eine nicht allzu lange Einführung in  $\text{\LaTeX}2\epsilon$ ,” 2012. <http://www.jokr.de/stefan/latex/LatexEinfuehrung.pdf>.
- [2] M. Jürgens and T. Feuerstack, “ $\text{\LaTeX}$  – eine Einführung und ein bisschen mehr...,” 2017. [https://www.fernuni-hagen.de/imperia/md/content/zmi\\_2010/a026\\_latex\\_einf.pdf](https://www.fernuni-hagen.de/imperia/md/content/zmi_2010/a026_latex_einf.pdf).
- [3] T. Richter, “ $\text{\LaTeX}$  – Tipps und Tricks,” 2013. [https://www.tortools.de/lib/exe/fetch.php/latex:richter\\_latex\\_tipps\\_1.2.pdf](https://www.tortools.de/lib/exe/fetch.php/latex:richter_latex_tipps_1.2.pdf).
- [4] M. Marx and O. Bücker, “Erstellung wissenschaftlicher Texte mit  $\text{\LaTeX}$ ,” 2017. [https://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/latex.pdf?\\_\\_blob=publicationFile](https://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/latex.pdf?__blob=publicationFile).
- [5] T. Nagel, “Einführung in  $\text{\LaTeX}$ ,” 2018. [http://www.nagel-net.de/Latex/DOKU/Latexkurs\\_Skript.pdf](http://www.nagel-net.de/Latex/DOKU/Latexkurs_Skript.pdf).
- [6] 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/>.
- [7] Don Knuth’s Home Page. <https://www-cs-faculty.stanford.edu/~knuth/>.
- [8] Leslie Lamport’s Home Page. <http://www.lamport.org>.
- [9]  $\text{\TeX}$  Live. <http://www.tug.org/texlive/>.
- [10] MiK $\text{\TeX}$ . <https://miktex.org>.
- [11] Mac $\text{\TeX}$ . <http://www.tug.org/mactex/>.
- [12] Kate. <https://kate-editor.org>.
- [13]  $\text{\TeX}$ studio. <http://texstudio.sourceforge.net>.
- [14] Texmaker. <http://www.xm1math.net/texmaker/>.

- [15] TeXworks. <https://www.tug.org/texworks/>.
- [16] Eclipse. <https://www.eclipse.org>.
- [17] Eclipse TeXlipse. <https://projects.eclipse.org/projects/science.texlipse>.
- [18] Authorea. <https://www.authorea.com>.
- [19] Overleaf. <https://www.overleaf.com>.
- [20] ShareLaTeX. <https://www.sharelatex.com>.
- [21] N. Schwarz, *Einführung in T<sub>E</sub>X*. Addison-Wesley, 1988.
- [22] U. Steinacker, “Leerraum in Typografie und Layout: So setzen Sie Gevierte, Halbgevierte und Co. richtig ein.” <http://typeschool.de>.
- [23] minipage – goLaTeX. <https://golatex.de/wiki/minipage>.
- [24] H. Kopka, *L<sup>A</sup>T<sub>E</sub>X Einführung Band 1*. Addison-Wesley, 3 ed., 2000.
- [25] KOMA-Script. <http://mirrors.ctan.org/macros/latex/contrib/koma-script/doc/scrguide.pdf>.
- [26] M. Goossens, F. Mittelbach, and A. Samarin, *Der L<sup>A</sup>T<sub>E</sub>X-Begleiter*. Addison-Wesley, 2000.
- [27] J. Schlosser, *Wissenschaftliche Arbeiten schreiben mit L<sup>A</sup>T<sub>E</sub>X*. mitp, 3 ed., 2009.
- [28] M. Ensenbach and M. Trettin, “Das L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>-Sündenregister oder Veraltete Befehle, Pakete und andere Fehler Version 2.7,” 2016. <ftp://ftp.dante.de/tex-archive/info/l2tabu/german/l2tabu.pdf>.
- [29] Geviert – Typografie-Fachbegriffe-Wiki. <http://www.typografie.info/3/wiki.html/g/geviert-r165/>.
- [30] Geschäftsbriefe mit L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> – der g-brief und g-brief2. <http://tug.ctan.org/macros/latex/contrib/g-brief/g-brief.pdf>.
- [31] L<sup>A</sup>T<sub>E</sub>X Font Catalogue. <http://www.tug.dk/FontCatalogue/>.
- [32] 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>.

- [33] makebarcode – Print various kinds 2/5 and Code 39 bar codes.  
<http://mirrors.ctan.org/macros/latex/contrib/makebarcode/doc/latex/makebarcode/makebarcode.pdf>.
- [34] QR-Code. <https://de.wikipedia.org/wiki/QR-Code/>.
- [35] The calligra package for use with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. <http://mirrors.ctan.org/macros/latex/contrib/fundus/calligra/calligra.pdf>.



# Befehle, Umgebungen, Dokumentklassen und Optionen

10pt, 20	\boxed, 111
11pt, 20	\breve, 105, 106
12pt, 20	\caption, 86
\Huge, 61	\cellcolor, 80
\LARGE, 61	\centering, 54
\Large, 61	\chaptername, 39
\Pr, 105	\chapter, 37
\abstractname, 39	\check, 105, 106
\acute, 105, 106	\cleardoublepage, 57
\addcontentsline, 42	\clearpage, 57
\and, 35	\cline, 84
\appendixname, 39	\columncolor, 79
\appendix, 41	\columnseprule, 32
\arccos, 105	\columnsep, 32
\arcsin, 105	\columnwidth, 32
\arctan, 105	\contentsname, 39
\arg, 105	\cosh, 105
\author, 35	\cos, 105
\backmatter, 40, 41	\coth, 105
\barcode, 123	\cot, 105
\bar, 105, 106	\csc, 105
\baselineskip, 29	\currentpage, 32
\baselinestretch, 30	\dashuline, 60
\bfseries, 62	\date, 36
\bibname, 39	\ddddot, 105
\bigg, 107	\dddot, 105
\big, 107	\ddot, 105, 106
\bmod, 105	\definecolor, 81

<code>\deg</code> , 105	<code>\includegraphics</code> , 85
<code>\det</code> , 105	<code>\indent</code> , 31
<code>\dim</code> , 105	<code>\indexname</code> , 39
<code>\displaystyle</code> , 94	<code>\inf</code> , 105
<code>\documentclass</code> , 19	<code>\item</code> , 62
<code>\dotfill</code> , 50	<code>\itshape</code> , 62
<code>\dotuline</code> , 60	<code>\ker</code> , 105
<code>\dot</code> , 105, 106	<code>\landscape</code> , 34
<code>\doublebox</code> , 48, 67	<code>\large</code> , 61
<code>\emph</code> , 59	<code>\left</code> , 107
<code>\evensidemargin</code> , 32	<code>\lg</code> , 105
<code>\exp</code> , 105	<code>\liminf</code> , 105
<code>\fboxsep</code> , 67	<code>\limsup</code> , 105
<code>\fbox</code> , 111	<code>\lim</code> , 105
<code>\figurename</code> , 39	<code>\linewidth</code> , 32
<code>\flushleft</code> , 54	<code>\listfigurename</code> , 39
<code>\flushright</code> , 55	<code>\listoffigures</code> , 43
<code>\footnotemark</code> , 64	<code>\listoftables</code> , 43
<code>\footnotesize</code> , 61	<code>\listtablename</code> , 39
<code>\footnotetext</code> , 64	<code>\ln</code> , 105
<code>\footnote</code> , 64	<code>\log</code> , 105
<code>\footrulewidth</code> , 27	<code>\makeindex</code> , 44
<code>\footskip</code> , 32	<code>\maketile</code> , 35, 36
<code>\frac</code> , 92	<code>\marginparpush</code> , 32, 65
<code>\frontmatter</code> , 40	<code>\marginparsep</code> , 32, 65
<code>\gcd</code> , 105	<code>\marginparwidth</code> , 32, 65
<code>\graphicspath</code> , 87	<code>\markboth</code> , 24
<code>\grave</code> , 105, 106	<code>\markright</code> , 24
<code>\har</code> , 105	<code>\mathbb</code> , 110
<code>\hat</code> , 106	<code>\mathcal</code> , 98
<code>\headheight</code> , 27, 32	<code>\mathnormal</code> , 97
<code>\headrulewidth</code> , 27	<code>\max</code> , 105
<code>\headsep</code> , 32	<code>\mbox</code> , 110
<code>\hfill</code> , 49	<code>\mdseries</code> , 62
<code>\hline</code> , 84	<code>\min</code> , 105
<code>\hom</code> , 105	<code>\mod</code> , 105
<code>\hrulefill</code> , 50	<code>\multicolumn</code> , 82
<code>\hspace</code> , 46	<code>\multirow</code> , 83, 84
<code>\huge</code> , 61	<code>\newline</code> , 56



<code>\newpage</code> , 57	<code>\rowcolor</code> , 78
<code>\nonfrenchspacing</code> , 48	<code>\scriptsize</code> , 61
<code>\nonumber</code> , 91	<code>\scshape</code> , 62
<code>\nopagebreak</code> , 57	<code>\secnumdepth</code> , 39
<code>\normalsize</code> , 61	<code>\section</code> , 37
<code>\notin</code> , 99	<code>\sec</code> , 105
<code>\not</code> , 99	<code>\setcounter</code> , 28, 40
<code>\oddpagelayoutfalse</code> , 32	<code>\sffamily</code> , 62
<code>\oddpagelayouttrue</code> , 32	<code>\shadowbox</code> , 67
<code>\oddsidemargin</code> , 32	<code>\sinh</code> , 105
<code>\onecolumn</code> , 34	<code>\sin</code> , 105
<code>\ovalbox</code> , 67	<code>\sloppy</code> , 52
<code>\overbrace</code> , 106	<code>\slshape</code> , 62
<code>\overleftarrow</code> , 107	<code>\small</code> , 61
<code>\overline</code> , 105	<code>\sout</code> , 59
<code>\overrightarrow</code> , 107	<code>\space</code> , 49
<code>\overset</code> , 95	<code>\sqrt</code> , 93
<code>\pagename</code> , 39	<code>\subparagraph</code> , 37
<code>\pagenumbering</code> , 28	<code>\subsection</code> , 37
<code>\pagestyle</code> , 23	<code>\substack</code> , 96
<code>\page</code> , 28	<code>\subsubsection</code> , 37
<code>\paperheight</code> , 32	<code>\sum</code> , 94
<code>\paperwidth</code> , 32	<code>\sup</code> , 105
<code>\paragraph</code> , 37	<code>\tablename</code> , 39
<code>\parindent</code> , 31	<code>\tableofcontents</code> , 41
<code>\parskip</code> , 30	<code>\tanh</code> , 105
<code>\partname</code> , 39	<code>\tan</code> , 105
<code>\part</code> , 37	<code>\textasciicircum</code> , 93
<code>\portrait</code> , 34	<code>\textasciitilde</code> , 45
<code>\prefacename</code> , 39	<code>\textbf</code> , 61
<code>\printindex</code> , 44	<code>\textcalligra</code> , 128
<code>\qquad</code> , 111	<code>\textfrac</code> , 123
<code>\qrcode</code> , 125	<code>\textgoth</code> , 123
<code>\quad</code> , 111	<code>\textheight</code> , 32
<code>\raggedleft</code> , 54	<code>\textit</code> , 61
<code>\raggedright</code> , 55	<code>\textmd</code> , 61
<code>\refname</code> , 39	<code>\textnormal</code> , 61
<code>\right</code> , 107	<code>\textrm</code> , 61
<code>\rmfamily</code> , 62	<code>\textsc</code> , 61

<code>\textsf</code> , 61	<code>\\</code> , 56
<code>\textsl</code> , 61	<code>a4paper</code> , 22
<code>\textswab</code> , 123	<code>a5paper</code> , 22
<code>\texttt</code> , 61	<code>abstract</code> , 36
<code>\textunderscore</code> , 93	<code>arabic</code> , 28
<code>\textup</code> , 61	<code>array</code> , 108
<code>\textvisiblespace</code> , 45, 49	<code>article</code> , 19
<code>\textwidth</code> , 32	<code>b5paper</code> , 22
<code>\thanks</code> , 36	<code>beamer</code> , 20
<code>\tilde</code> , 105, 106	<code>book</code> , 19
<code>\tiny</code> , 61	<code>boxedimage</code> , 68
<code>\title</code> , 35	<code>center</code> , 54
<code>\tocdepth</code> , 37	<code>description</code> , 62
<code>\today</code> , 36	<code>dinbrief</code> , 20
<code>\topmargin</code> , 32	<code>dingautolist</code> , 128
<code>\ttfamily</code> , 62	<code>dingfill</code> , 127
<code>\twocolumn</code> , 34	<code>dingline</code> , 127
<code>\uline</code> , 59	<code>dinglist</code> , 127
<code>\underbrace</code> , 106	<code>ding</code> , 126
<code>\underleftarrow</code> , 107	<code>displaymath</code> , 90
<code>\underline</code> , 59, 106	<code>draft</code> , 21
<code>\underrightarrow</code> , 107	<code>em</code> , 59
<code>\underset</code> , 95	<code>enumerate</code> , 62
<code>\upshape</code> , 62	<code>eqnarray</code> -Umgebung, 91
<code>\url</code> , 66	<code>eqnarray</code> , 90
<code>\uuline</code> , 59	<code>eqnarray</code> -Umgebung, 91
<code>\uwave</code> , 59	<code>equation</code> -Umgebung, 90
<code>\varepsilon</code> , 98	<code>executivepaper</code> , 22
<code>\varphi</code> , 98	<code>fancybox</code> , 66
<code>\varpi</code> , 98	<code>final</code> , 21
<code>\varrho</code> , 98	<code>fleqn</code> , 21
<code>\varsigma</code> , 98	<code>g-brief2</code> , 20, 115
<code>\vartheta</code> , 98	<code>itemize</code> , 62
<code>\vec</code> , 105, 106	<code>landscape</code> , 22, 34
<code>\verb</code> , 65	<code>legalpaper</code> , 22
<code>\vspace</code> , 47	<code>leqno</code> , 21
<code>\widehat</code> , 106	<code>letterpaper</code> , 22
<code>\widetilde</code> , 106	<code>letter</code> , 19
<code>\xout</code> , 60	<code>math</code> , 89

notitlepage, 21	scrbook, 20
onecolumn, 20	scrllttr2, 20, 119
oneside, 21	scrreprt, 20
openany, 22	slides, 19
openright, 22	tabularx, 75
portrait, 34	tabular, 71
proc, 19	titlepage, 21
quotation, 55	twocolumn, 20
quote, 55	twoside, 21
report, 19	verbatim, 65
scrartcl, 20	



# Stichwortverzeichnis

Überstreichungssymbole, 105

Abbildungen, 85

Abbildungsverzeichnis, 43

Abkürzungen, 46

Absatzabstand, 30

Absatzeinzüge, 10

Abstand

horizontal, 46

vertikal, 47

Abstract, 36

Altdeutsche Schrift, 123

Anführungszeichen, 11

Anhang, 41

Aufzählung, 62

Aufzählungspunkt, 12

Auslassungspunkte, 109

Backslash, 11

Barcode, 123

Basisschrift, 20

Befehle, 7

Bilder, 85

binäre Operationen, 98

Boxen, 66

Brief, 115

Bruch, 92

Bruchstrich, 92

Buchhauptteil, 40

Buchnachspann, 40

Buchstaben

Griechische, 96

Kalligrafische, 98

Buchvorspann, 40

CMYK, 79

Copyright, 12

Dagger-Zeichen, 12

DIN-A4, 22

DIN-A5, 22

Dokumentklasse, 19

Doppel-Dagger-Zeichen, 12

eingetrückter Text, 55

Einrückung

Absatz, 31

Euro, 12

Exponent, 92

fancyheadings, 26

Farbmodell, 78

Flattersatz, 53

Formatierungshilfen, 45

Formel

abgesetzt, 90

eingebettete, 89

nummeriert, 90

Formelsatz, 89

Frenchspacing, 48

Fußnote, 64

Geviert, 9

- Gleichungssysteme, 90
- Gliederungsbefehl, 37
- Grafiken, 85
- Griechische Buchstaben, 96
- Gruppe, 7
  
- Hochformat, 34
- HTML, 79
- Hurenkind, 30
  
- Index, 92
- Indexregister, 44
- Inhaltsverzeichnis, 41
  - Eintrag, 42
- Integral, 94
- Integralzeichen, 94
- Internetadressen, 66
  
- Kalligrafie, 98
- Klammer
  - Größe, 107
- Klassenoptionen, 20
- Kopfzeile, 24
  
- Längenbefehle, 10
- Layout, 45
- Leerzeichen, 49
- Ligatur, 15
- Linksbündige Spalte, 73
- Linksbündiger Text, 54
  
- Marginalie, 64
- Markierung, 43
- Mathematik, 89
  - Bruch, 92
  - Konstrukte, 92
- Matrizen, 108
- Mengensymbol, 110
  
- Nenner, 92
  
- Operationssymbole, 98
- Ordnung, 93
  
- Papiergröße, 22
- Paragraph-Zeichen, 12
- Pfeil, 100
- Pfund Sterling, 12
- Portrait, 34
- Produktzeichen, 94
  
- QR-Code, 125
- Querformat, 34
- Querverweis, 43
  
- Radikant, 93
- Randbemerkung, 64
- Randnotiz, 64
- Rastergrafik, 85
- Rechtsbündige Spalte, 73
- Rechtsbündiger Text, 55
- Registered-Zeichen, 12
- RGB, 79
  
- Schriftgröße, 20
- Schriftschnitt, 59
- Schusterjunge, 30
- Seitendeklaration, 31
- Seitenstil, 23
- Seitenumbruch, 57
- Spalten, 71
- Spaltenformatierungseintrag, 72
- Spezialzeichen, 15
- Stichwortverzeichnis, 44
- Summe, 94
- Summenzeichen, 94
  
- Tabellen, 71
  - Doppelstrich, 73
  - Farben, 78
  - Felder zusammenfassen, 82

- Spalten, 72
- Strich, 73
- Tabellendefinition, 71
- Tabellenpräambel, 71
- Tabellenverzeichnis, 43
- Text
  - eingerrückt, 55
  - linksbündig, 54
  - rechtsbündig, 55
  - unformatierter, 65
  - zentriert, 54
- Textbreite, 10
- Textfeld, 52
- Textformel, 89
- Texthöhe, 10
- Texthervorhebungen, 59
- Titelei, 35
- Titelseite, 21
- Trademark-Zeichen, 12
- Trennungsregel, 52
- Umgebung, 8
- Unformatierter Text, 65
- Unterabschnitt, 38
- Unterstreichungsymbole, 105
- URL, 66
- Vektorgrafik, 85
- Vergleichssymbole, 99
- Wortabstand, 45
- Wurzel, 93
- Wurzelzeichen, 93
- Yen, 12
- Zähler, 92
- ZapfDingbats, 125
- Zeichen
  - Abstand, 45
  - positionieren, 95
- Zeichenabstand, 45
- Zeiger, 100
- Zeilenabstand, 29
- Zeilenumbruch, 21, 46, 56
- Zentrierte Spalte, 73
- Zentrierter Text, 54
- Ziffern
  - arabisch, 28
  - römisch, 28
- Zusammenfassung, 36
- Zwischenraum, 49, 50