

Einführung in L^AT_EX – Theorie –

Felix Widrig

2008

Inhaltsverzeichnis

1	Installation und erste Schritte	4
1.1	L ^A T _E X herunterladen und installieren	4
1.2	Einen Editor herunterladen und installieren	4
1.3	Umgang mit dem Editor TexnicCenter	4
1.4	Umgang mit Fehlermeldungen	4
2	Grundlegende Eigenschaften von L^AT_EX	6
2.1	Grundstruktur eines L ^A T _E X-Dokumentes	6
2.2	L ^A T _E X-Befehle und deren Argumente	6
2.2.1	Definition eines Befehls	6
2.2.2	Wichtige Regel für Befehle	6
2.3	Typische Präambel für deutschsprachige Texte	6
2.4	Symbole, Sonderzeichen und Umlaute	7
2.5	Abstände und Zeilenwechsel, Seitenwechsel	7
2.5.1	Regeln für Abstände	7
2.5.2	Längen und Einheiten	8
3	Dokumentklassen	9
3.1	Standardklassen	9
3.2	Erweiterte Dokumentklassen	9
3.3	Klassenoptionen	9
3.4	Der Seitenstil	10
3.5	Gliederungsbefehle	10
3.6	Das Inhaltsverzeichnis	11
3.7	Der Titel	11
3.8	Die Zusammenfassung	11
3.9	Der Anhang	12
3.10	Sprachanpassungen	12
3.11	Verweise	12
4	Formatierungen im Text	13
4.1	Schriftarten und -größen	13
4.1.1	Ändern der Schriftgröße	13
4.1.2	Schriftarten	13
4.2	Textausrichtung	14
4.3	Tabulatoren	14
4.4	Einfache Listen und Aufzählungen	15
4.5	Text unterstreichen	15
4.6	Fußnoten und Randnotizen	15
4.7	Boxen	15
4.7.1	Allgemeine Boxen	16
4.7.2	Balkenboxen	16
4.7.3	Anpassbare Boxenparameter	16
5	Einbinden von Graphiken	17
5.1	Graphik einbinden	17
5.2	Bilder im separaten Verzeichnis sortieren	17
5.3	Bildformate	17
6	Gleitobjekte (Abbildungen, Tabellen)	18
6.1	Definition von Gleitobjekten	18
6.2	Positionieren von Gleitobjekten	18

6.3	Beschriften von Gleitobjekten	18
6.4	Gleitobjektverzeichnisse	19
6.5	<code>subfig</code> – Gruppieren mehrerer Gleitobjekte	19
7	Einfache Tabellen	20
7.1	Definition	20
7.2	Linien in Tabellen	20
7.3	Weitere Gestaltungselemente	20
7.4	Abstände in Tabellen	21
7.5	<code>array</code> – Eine Erweiterung für Tabellen	21
7.6	Gesamtbreite von Tabellen festlegen	22
7.7	<code>tabularx</code> – Spaltenbreite automatisch berechnen	22
7.8	<code>multirow</code> – Vertikale Ausrichtung	22
7.9	<code>tbls</code> – Einstellen von Zeilenabständen	22
7.10	<code>dcolumn</code> – Ausrichtung am Dezimalpunkt	22
7.11	<code>supertabular</code> – Mehrseitige Tabellen	23
7.12	Fußnoten in Tabellen	23
8	Mathematische Formeln	24
9	Literaturverzeichnisse erstellen	26
9.1	Einfache Literaturverzeichnisse	26
9.2	Literaturverzeichnisse mit <code>BibTeX</code>	26
10	Farbdruck und Text drehen mit <code>LaTeX</code>	28
10.1	Farben Definieren	28
10.2	Farben benutzen	28
10.3	Farbige Boxen	28
10.4	Gedrehter Text	28
11	Umgang mit Längen	29
11.1	Die Längen des Dokumentes automtisch skalieren	30
11.2	<code>layout</code> – Das Seitenlayout im Dokument ausgeben	30
12	Umgang mit Zählern	31
13	<code>LaTeX</code> Besonderheiten	33
13.1	Große Dokumente schreiben mit <code>LaTeX</code>	33
13.2	Eigene Befehle definieren	33
13.3	<code>fancyhdr</code> – Den Kolumnentitel anpassen	34
13.4	<code>hyperref</code> – Hyperlinks in PDFs	34
13.5	<code>nextpage</code> – Leerseiten automatisch berechnen	35
13.6	Mit der Kommandozeile arbeiten	35
	Literaturverzeichnis und Links zu <code>LaTeX</code>	36

1 Installation und erste Schritte

Folgende Komponenten werden benötigt, um `LaTeX` benutzen zu können.

1.1 `LaTeX` herunterladen und installieren

Für Windows existiert die weit verbreitete `LaTeX`-Distribution `MiKTeX`. Die Distribution ist kostenlos und wird unter einer Open Source Lizenz vertrieben. Link zum herunterladen: www.miktex.org
Für Linux gibt es eine ähnliche Distribution namens `TeXX`.

1.2 Einen Editor herunterladen und installieren

Um `LaTeX`-Dokumente erstellen zu können, wird ein Editor benötigt. Folgende Editoren sind speziell für `LaTeX` entwickelt worden und erleichtern somit die Arbeit durch komfortable tools:

LEd ein mächtiger `LaTeX` -Editor mit zahlreichen integrierten Tools. Zurzeit hat dieser Editor immer noch zahlreiche Fehler und ist für Anfänger nicht geeignet. Link zum herunterladen: www.latexeditor.org

TeXnicCenter unter Windows ein sehr weit verbreiteter `LaTeX` -Editor, etwas einfacher gestaltet als **LEd**. Link zum herunterladen: www.toolscenter.org. Nach der Installation muss **TeXnicCenter** konfiguriert werden. Der Pfad zu `latex` muss angegeben werden, das ist normalerweise im folgenden Ordner:
`C:/Programme/MiKTeX 2.7/miktex/bin`

1.3 Umgang mit dem Editor **TeXnicCenter**

Nach der Installation und Konfiguration von **TeXnicCenter** gibt 4 mögliche Wege, eine `*.tex`-Datei in ein fertiges Dokument umzuwandeln:

Latex \Rightarrow **DVI**: erzeugt eine **DVI**-Datei zum beobachten des Dokuments. Diese Dateien können nicht auf allen Druckern ausgedruckt werden, sie dienen eher zur Betrachtung am Bildschirm. Der Vorteil bei **DVI** ist, dass man die Datei nicht schliessen braucht, um sie zu überschreiben.

Latex \Rightarrow **PDF**: erzeugt das Dokument als **PDF**-Datei. Ist das Dokument bereits als **PDF** geöffnet, und möchte man es überschreiben, so muss die Datei zuerst geschlossen werden.

Latex \Rightarrow **PS**: erzeugt das Dokument als **PS**-Datei.

Latex \Rightarrow **PS** \Rightarrow **PDF**: erzeugt das Dokument zuerst als **PS**-Datei, und wandelt es dann automatisch nach **PDF** um. Dies ist manchmal notwendig, zB wenn das **Graphik**-Paket **PSTricks** benutzt wird, da **PdfLaTeX** die **Postscript**-Sprache nicht erkennt.

1.4 Umgang mit Fehlermeldungen

Nach der Übersetzung eines `LaTeX`-Quelltextes können Fehler Auftauchen. Dies kann anhand der letzten Zeile des Übersetzungsprotokolls erkannt werden. Ist ein Fehler (oder eine Warnung) aufgetreten, so ist er im Protokoll durch ein spezielles Symbol und Farben gekennzeichnet.

Es ist wichtig, die Fehler zu beseitigen, bevor die eigentliche Arbeit am Dokument fortgesetzt wird, da sie sonst später sehr schwer zu finden sind. Suchen Sie in der Fehlermeldung nach der Zeilennummer im Eingabetext, bei der `LaTeX` den Fehler bemerkt hat. Der eigentliche Eingabefehler kann in dieser Zeile liegen, aber auch davor. Falls das nicht zur Aufklärung führt:

- Bearbeitung möglichst zu Ende führen und die Ausgabe anschauen: Wo weicht das Ergebnis stark von dem erwarteten ab?

- Verdächtige Zeilen im Quelltext mit % auskommentieren, damit L^AT_EX sie beim nächsten Lauf ignoriert. Diesen Vorgang solange wiederholen, bis keine Fehlermeldung mehr auftritt, und den Fehler so eingrenzen.
- Beachten Sie die Namen der verwendeten Dateien, benutzen Sie keine von L^AT_EX geschützten Namen, und achten Sie auf evtl. unerlaubte Umlaute im Dateinamen.

2 Grundlegende Eigenschaften von L^AT_EX

2.1 Grundstruktur eines L^AT_EX-Dokumentes

Jedes L^AT_EX -Dokument besteht aus 2 Teilen:

1. Präambel oder Vorspann / Header:

- Angaben zum Aussehen des Textes: Papiergröße, Dokumentklasse, Schriftgröße, Seitenstil, etc.
- Zusätzlich zu ladende Makropakete (zB. zur Sprachanpassung oder Graphikbearbeitung). Um ein Makropaket zu laden benutzt man den Befehl `\usepackage[Optionen]{Paketname}`
- Zusatzbefehle, die auf das ganze Dokument *global* wirken sollen, wie zum Beispiel die Verwendung einer anderen Schrift.

Alles was im Vorspann, steht, wird nicht ausgedruckt. Hier darf kein herkömmlicher Text stehen!

Der Vorspann beginnt mit `\documentclass{Dokumentklasse}` endet mit `\begin{document}`.

2. Textteil oder Body:

- Der eigentliche Text
- Befehle innerhalb des Textes, also Überschriften, Textformatierung, etc.

Der Textteil umfaßt all das, was zwischen `\begin{document}` und `\end{document}` steht. Danach darf nichts mehr stehen.

2.2 L^AT_EX-Befehle und deren Argumente

2.2.1 Definition eines Befehls

Befehle steuern bestimmte Formatierungen im Text oder im gesamten Dokument. Befehle bestehen immer aus einem Backslash `\` und dem eigentlichen Namen des Befehls. Manche Befehle erwarten optionale und/oder obligatorische Argumente. Die Syntax eines Befehls mit ein optionales und ein obligatorisches Argument ist

$$\backslash\text{Befehl}[\text{Optionales Argument}]{\text{Obligatorisches Argument}}$$

Zum Beispiel ist der erste Befehl eines L^AT_EX-Quelltextes immer `\documentclass`. Normalerweise werden obligatorische Argumente immer in geschweiften Klammern geschrieben `{ }`, optionale in Eckigen Klammern `[]`. Ein Beispiel: `\documentclass[a4paper]{article}`.

2.2.2 Wichtige Regel für Befehle

Immer wenn ein Befehl innerhalb eines mit geschweiften Klammern `{ }` geschlossenen Bereichs geschrieben wird, so hat er nur innerhalb dieser Klammern Wirkung und wird danach wieder aufgehoben. Entsprechendes gilt auch für Befehle, die innerhalb von Umgebungen¹ geschrieben werden.

2.3 Typische Präambel für deutschsprachige Texte

Die Präambel deutschsprachiger Dokumente beinhaltet in der Regel zumindest diese Zeilen:

- `\usepackage[ngerman]{babel}` : lädt die Sprache. Wichtig für die Silbentrennung.
- `\usepackage[T1]{fontenc}` : für eine erweiterte Ausgabekodierung der Zeichensätze. Für eine gute Silbentrennung deutschsprachiger Dokumente.
- `\usepackage[latin1]{inputenc}` : ermöglicht das eingeben von vielen Sonderzeichen über die Tastatur, u.a. auch deutsche ü, ä, ö und ß.
- `\parindent=0mm` : damit neue Absätze nicht mehr eingerückt werden.
- `\renewcommand\baselinestretch{1.5}` : setzt den Faktor für den Zeilenabstand auf 1.5

¹Eine Umgebung beginnt immer mit `\begin{umgebung}` und endet mit `\end{umgebung}`

2.4 Symbole, Sonderzeichen und Umlaute

Symbole: In einem L^AT_EX-Quelltext dürfen fast keine Symbole (\backslash , $\&$, usw.) als Textsymbol eingegeben werden, denn sie werden von L^AT_EX anders interpretiert und dienen speziellen Zwecken. Manche von diesen Symbolen können getippt werden, indem sie mit einem \backslash eingeleitet werden, also zum Beispiel $\&$ für ein $\&$, oder $\%$ für ein $\%$. Die meisten Symbole werden jedoch mit Befehlen gesetzt, wie zum Beispiel $\textbackslash\textbackslash$ für ein \backslash .

Die geschweiften Klammern $\{ \}$ haben zum Beispiel die Wirkung, einen logischen Bereich abzuschließen, siehe Abschnitt 2.2.2.

Das Kommentarzeichen: Der gesamte Text zwischen einem Prozentzeichen $\%$ und dem Ende der Zeile bleibt beim Übersetzen eines Quelltextes unbeachtet.

Umlaute und Sonderzeichen: Mit dem Paket `inputenc` können Umlaute direkt über die Tastatur eingegeben werden. Manche Umlaute oder Sonderzeichen gibt es nicht auf jede Tastatur, sie können anders eingegeben werden. Der Befehl \backslash " erzeugt zwei Punkte über jeden nachfolgenden Buchstaben. Mit \backslash ~ bekommt man einen accent circonflexe, usw.

Anführungszeichen Nach dem Laden der deutschen Sprache können Anführungszeichen eingegeben werden gemäß Tabelle 1. Manche in der Tabelle angegebene Kurzformen stehen nur unter der T1-Kodierung (Seite 6) zur Verfügung.

Kurzform	Befehl	Beispiel	Kurzform	Befehl	Beispiel
"‘	$\backslash\text{glqq}$	„normale		$\backslash\text{flq}$	<halbierte
"’	$\backslash\text{grqq}\backslash$	deutsche“		$\backslash\text{frq}\backslash$	französische>
	$\backslash\text{glq}\backslash$	„halbierte	“‘	$\backslash\text{textquotedblleft}$	“anglikanische
	$\backslash\text{grq}\backslash$	deutsche’	”’	$\backslash\text{textquotedblright}\backslash$	Form”
<<	$\backslash\text{flqq}$	<französische	‘	$\backslash\text{textquoteleft}$	‘halbe
>>	$\backslash\text{frqq}\backslash$	guillemets>	’	$\backslash\text{textquoteright}\backslash$	anglikanische’
				$\backslash\text{dq}$	"doublequotes"

Tabelle 1: Anführungszeichen

2.5 Abstände und Zeilenwechsel, Seitenwechsel

2.5.1 Regeln für Abstände

Im Gegensatz zu sogenannten WYSIWYG (What You See Is What You Get) - Textverarbeitungsprogrammen (z. Bsp. MS Word), haben Veränderungen im L^AT_EX-Quelltext nicht unbedingt Auswirkungen auf den gedruckten Text:

- **Mehrere Leerzeichen** bewirken das Gleiche wie ein Leerzeichen.
- **Horizontaler Abstand** wird durch $\backslash\text{hspace}\{Abstand\}$ erzeugt.
- **Einen Zeilenumbruch** im laufenden Absatz wird durch das "Spezialkommando" $\backslash\backslash$ erreicht. Der Zeilenumbruch im gedruckten Dokument ist in der Regel unabhängig vom Zeilenumbruch im Quelltext. Durch hinzufügen von eckigen Klammern kann auch zusätzlicher vertikaler Abstand gesetzt werden, bevor die neue Zeile beginnt, zB. $\backslash\backslash[5\text{mm}]$ für 5 Millimeter.
- **Einen neuen Absatz** wird durch **eine oder mehrere Leerzeilen** im Quelltext erzeugt. Eine Leerzeile nach einem Zeilenumbruch ($\backslash\backslash$) erzeugt einen großen Leerraum über dem neuen Absatz.
- **Vertikalen Abstand** wird mit $\backslash\text{vspace}\{Abstand\}$ erreicht, oder den Kurzkommandos $\backslash\text{smallskip}$, $\backslash\text{medskip}$ und $\backslash\text{bigskip}$. Bei den Kurzkommandos muss eine Leerzeile folgen, bevor die nächste Zeile anfängt.

- **Einen Seitenwechsel** wird mit dem Befehl $\backslash\text{newpage}$ erzwungen. Alternativ wird auch mit $\backslash\text{pagebreak}$ einen Seitenwechsel erzwungen. Dann wird der Inhalt der aktuellen Seite gleichmäßig über die gesamte Höhe verteilt.

2.5.2 Längen und Einheiten

Ein Abstand wird in seiner Länge und in seinen Einheiten angegeben, zum Beispiel 5cm für 5 Zentimeter. Es gibt auch Längeneinheiten, die variabel sind, je nach Schriftgröße:

em : die Breite eines M in der aktuellen Schriftgröße. Zum Beispiel $\backslash\text{hspace}\{5\text{em}\}$ für einen Abstand, in dem 5 große M reinpassen würden, egal wie groß die Schrift eingestellt ist.

ex : die Höhe eines x in der aktuellen Schriftgröße.

Eine nützliche Länge ist $\backslash\text{textwidth}$: die Textbreite des Dokuments. Wird diese als Abstand für einen Befehl angegeben, so muss keine Längeneinheit angegeben werden.

Abstand automatisch berechnen

Mit

Text vorher $\backslash\text{hfill}$ *Text nachher*

wird soviel horizontaler Abstand nach *Text vorher* hinzugefügt, bis das *Text nachher* am rechten Rand der Seite gesetzt wird. Entsprechend gilt für vertikalen Abstand $\backslash\text{vfill}$.

3 Dokumentklassen

Jeder L^AT_EX-Quelltext beginnt zwangsläufig mit dem Laden einer Dokumentklasse, die das allgemeine Layout des Textes festlegt:

```
\documentclass[a4paper, 12pt]{article}
```

ist eine dieser Klassen. Als Option wurde hier das Papierformat auf a4 gesetzt (standardmäßig ist es sonst das amerikanische letter-Format) und die Standardschriftgröße von 10pt auf 12pt heraufgesetzt. Beim Laden einer Dokumentklasse können verschiedene Optionen in eckigen Klammern mitgeladen werden, die dann für das gesamte Dokument wirken.

3.1 Standardklassen

Standardmäßig gehören zu L^AT_EX folgende Dokumentklassen:

- **article**: für einfache Dokumente. Gliederung möglich in Abschnitten und Unterabschnitten.
- **report**: für umfangreichere Dokumente mit Kapitelgliederung, wie zum Beispiel Studien- und Diplomarbeiten.
- **book**: für Bücher.
- **letter**: für anspruchsvolle Briefe.

3.2 Erweiterte Dokumentklassen

Speziell für die Anforderungen europäischer Autoren gibt es Dokumentklassen, die eine Reihe von Zusatzbefehlen und ein oft angenehmeres Layout erzeugen (Seitenränder, Papiergröße) als die anglikanischen Standardklassen. Sie werden als KOMA-Script-Familie zusammengefasst: **scrartcl**, **scrreprt**, und **scrbook**. Es ist empfehlenswert, die zugehörige Dokumentation im Netz nachzulesen (siehe Hinweise auf S. 36).

3.3 Klassenoptionen

Diese Optionen können beim laden der Dokumentklasse angegeben werden und ändern das Layout des Dokumentes (es gibt noch viel mehr Optionen):

- **10pt**, **11pt**, **12pt**: Standardschriftgröße für das gesamte Dokument. Befehle, die später die Schriftgröße ändern, beziehen sich auf diese Größe. Voreingestellt ist 10pt.
- **onecolumn**, **twocolumn** einspaltig oder zweisepaltig. Voreingestellt ist einspaltig.
- **oneside**, **twoside**: einseitig oder doppelseitiges Seitenlayout. Bei **twoside** wird der Abstand des linken Seitenrandes für gerade Seiten anders eingestellt, als bei ungeraden Seiten, so dass die Textränder auf Doppelseiten übereinstimmen.
- **notitlepage**, **titlepage**: bei **book** und **report** wird der Titel des Dokumentes auf eine eigene Seite gesetzt, bei **article** oberhalb des nachfolgenden Textes. Mit dieser Option kann dies gesteuert werden.
- **openright**, **openany**: bei **book** beginnt jedes Kapitel auf einer rechten Seite, bei **report** auf der nächsten Seite. Mit dieser Option kann dies gesteuert werden.
- **landscape**: mit dieser Option wird der Text im Querformat formatiert.

plain	Leerer Seitenkopf, Seitenzahl zentriert unten. Dies ist das Standardseitenformat, wenn für pagestyle nichts angegeben wird.
empty	Ohne Seitenzahl, leerer Seitenkopf.
headings	Überschrift und Seitenzahl in der Kopfzeile. Daneben existiert noch myheadings , womit das Standardverhalten geändert werden kann.

Tabelle 2: Der Seitenstil

3.4 Der Seitenstil

Der Seitenstil legt den grundsätzlichen Aufbau einer Seite fest. Er wird mit dem Befehl `\pagestyle{<Stil>}` im Vorspann des Dokuments festgelegt. Die verschiedenen Stile können Tabelle 2 entnommen werden².

Die ersten Seiten von neuen Kapiteln werden von L^AT_EX automatisch mit dem Stil **plain** gesetzt, auch wenn in der Präambel das Layout **empty** gewählt wurde. Dies kann unterdrückt werden: mit dem Befehl `\thispagestyle{empty}` läßt sich der Seitenstil für die aktuelle Seite verändern.

Die Seitennummerierung läßt sich mit dem Befehl `\pagenumbering{<Stil>}` verändern. Dieser Befehl darf irgendwo im Dokument auftauchen, auch mehrfach. Verschiedene Stile stehen zur Verfügung, siehe Tabelle 3.

arabic	1, 2, 3, ...
roman	i, ii, iii, ...
Roman	I, II, III, ...
alph	a, b, c, ...
Alph	A, B, C, ...

Tabelle 3: Der Nummerierungsstil

Die Voreinstellung ist **arabic**. Dort, wo der Befehl `\pagenumbering{}` auftaucht, wird die Seitennummerierung zurückgesetzt und fängt dann im neuen Stil bei 1 wieder an.

Kolumnentitel auf leere Seiten: Es passiert manchmal, dass ein Kapitel auf einer ungeraden Seite endet. Der Kolumnentitel wird aber standardmäßig auf allen Seiten gesetzt, nur nicht auf denen, wo ein Kapitel beginnt. Um also einen Kolumnentitel auf einer sonst leeren Seite zu vermeiden, kann am ende jedes Kapitels der in 13.5 beschriebene Befehl geschrieben werden.

3.5 Gliederungsbefehle

Welche Gliederungsbefehle (Überschriften) zur Verfügung stehen, wird in Tabelle 4 aufgelistet.

Gliederungsbefehl	book	report	article
<code>\part</code>	✓	✓	✓
<code>\chapter</code>	✓	✓	
<code>\section</code>	✓	✓	✓
<code>\subsection</code>	✓	✓	✓
<code>\subsubsection</code>	✓	✓	✓
<code>\paragraph</code>	✓	✓	✓
<code>\subparagraph</code>	✓	✓	✓

Tabelle 4: Gliederungsbefehle

²Der Seitenstil kann auch frei formatiert werden, siehe Abschnitt 13.3

Die Überschriften werden fortlaufend nummeriert und ins Inhaltsverzeichnis eingetragen. Mit der Sternform `\section*` unterbleibt die Nummerierung, und es erfolgt auch kein automatischer Eintrag ins Inhaltsverzeichnis. Die Gliederungen `\paragraph` und `\subparagraph` werden standardmäßig nicht nummeriert und nicht ins Inhaltsverzeichnis aufgenommen. Um auch diese zu nummerieren, wird der Befehl `\addtocounter{secnumdepth}{2}` benutzt.

Die KOMA-script Dokumentklassen ermöglichen eine kleine, nicht nummerierte Überschrift durch `\minisec`. Außerdem können nicht nummerierte Abschnitte in diesen Klassen durch `\addchap` und `\addsec` ins Inhaltsverzeichnis aufgenommen werden.

3.6 Das Inhaltsverzeichnis

Das Inhaltsverzeichnis wird automatisch an der Stelle im Dokument eingefügt, wo der Befehl `\tableofcontents` steht. Dazu sind zwei \LaTeX -Läufe notwendig.

Einen zusätzlichen Eintrag ins Inhaltsverzeichnis wird erzeugt, indem an der gewünschten Stelle im Quelltext folgendes steht: `\addcontentsline{toc}{Ebene}{Text}`. Für *Ebene* steht dann zB `section`.

3.7 Der Titel

Die Titelseite für das Dokument kann entweder frei erstellt werden:

```
\begin{titlepage}
...freies formatieren...
\end{titlepage}
```

oder mit Befehlen:

```
\title{Titelüberschrift} \author{Autorenname} \date{Datum}
```

Die Angaben `\title` und `\author` sind zwingend notwendig. Fehlt `\date`, wird das aktuelle Datum eingesetzt. Mehrere Autorennamen werden mit `\and` aneinander gereiht. Diese Befehle dienen allerdings nur der Angabe der nötigen Daten; gesetzt wird der Titel erst mit dem Kommando `\maketitle`

Die KOMA-script-Klassen bieten für den Titel noch weitere Befehle:

```
\titlehead{Titelkopf} \subject{typ der Arbeit} \publishers{herausgeber}
\thanks{Fußnote}
```

Wird die Arbeit doppelseitig gesetzt, lassen sich auf der Rückseite des Titelblattes Textblöcke setzen, oben und unten durch `\uppertitleback` und `\lowertitleback`.

Die Titelseite wird standardmäßig nicht nummeriert.

3.8 Die Zusammenfassung

Zu Beginn eines Artikels kann die Zusammenfassung seines Inhalts mit der `abstract`-Umgebung gesetzt werden:

```
\begin{abstract}
Text der Zusammenfassung...
\end{abstract}
```

Diesem Text geht dann (je nach Spracheinstellung) die zentrierte Überschrift „Abstract“ voraus. Der Text selbst erscheint kleingedruckt und beidseitig eingerückt.

Bei den KOMA-script-Klassen muss die Option `abstracton` geladen werden, damit die Zusammenfassung eine Überschrift erhält.

3.9 Der Anhang

Der Anhang wird mit dem Befehl `\appendix` eingeleitet, der bewirkt, dass die Zähler für `\section` bei `article` und `chapter` bei `report` und `book` zurückgesetzt werden und die Nummerierung in Großbuchstaben erfolgt. Auch der Anhang erscheint im Inhaltsverzeichnis.

3.10 Sprachanpassungen

Mit dem Befehl

```
\selectlanguage{ }
```

kann die aktuelle Spracheinstellung verändert werden. Voraussetzung ist, dass die gewählte Sprache im Vorspann mit `babel` geladen wurde. Zugelassen sind u.a. `english`, `USenglish`, `french`, `german`, `ngerman`. Beispiel für neue Deutsche Rechtschreibung:

```
\usepackage[ngerman]{babel}
```

Werden mehrere Sprachen geladen, so wird die zuletzt geladene Sprache eingestellt. Eine Umschaltung der Spracheinstellung hat folgende Auswirkungen:

1. Die Silbentrennungsregeln der entsprechenden Sprache werden automatisch angewandt.
2. Die voreingestellten Parameter wie zum Beispiel `\tableofcontents` oder `\abstractname` werden angepasst. Bei `\selectlanguage{french}` erhält man dann *table des matières* anstatt *Inhaltsverzeichnis* und *résumé* anstatt *Zusammenfassung*.

3.11 Verweise

\LaTeX erlaubt es, im laufenden Text auf Graphiken, Tabellen, Abbildungen, Formeln und Seiten zu verweisen. Dazu ist ein zweiter Lauf von \LaTeX zum Verarbeiten der Referenzen notwendig. Zunächst muss eine Textstelle, auf die verweist wird, *gekennzeichnet* werden. Dies geschieht mit dem Befehl `\label{Schlüssel}`

Nun kann auf ein gekennzeichnetes Kapitel, eine Abbildung, eine Tabelle oder eine Mathematische Formel verwiesen werden: `\ref{Schlüssel}`

Dabei wird an der entsprechenden Stelle die Nummer des Objekts, auf das verwiesen wird, eingesetzt. Der Befehl `\pageref{Schlüssel}` verweist auf die Nummer der Seite, auf der das gekennzeichnete Objekt steht.

Wichtig: Bei Verweisen auf Tabellen und Abbildungen muss der zugehörige `\label`-Befehl direkt auf `\caption` folgen, ansonsten wird später mit `\ref` eventuell auf eine falsche Nummer verwiesen.

4 Formatierungen im Text

4.1 Schriftarten und -größen

Standardmäßig ist eine Roman-Schrift mittlerer Stärke der Größe 10pt eingestellt.

4.1.1 Ändern der Schriftgröße

Mit den in Tabelle 5 aufgeführten Befehlen kann die Schriftgröße geändert werden. Jeder dieser Befehle ist so lange wirksam, bis ein anderer, zum Beispiel die Rückstellung auf normale Schrift mit `\normalsize`, auftritt³.

Schriftgrößen in L ^A T _E X			
<code>\tiny</code>	Schrifttest	<code>\Large</code>	Schrifttest
<code>\scriptsize</code>	Schrifttest	<code>\LARGE</code>	Schrifttest
<code>\footnotesize</code>	Schrifttest		
<code>\small</code>	Schrifttest	<code>\huge</code>	Schrifttest
<code>\normalsize</code>	Schrifttest		
<code>\large</code>	Schrifttest	<code>\Huge</code>	Schrifttest

Tabelle 5: Schriftgrößen

4.1.2 Schriftarten

Schriften werden in L^AT_EX durch sogenannte Attribute charakterisiert. Davon gibt es drei:

- Die Schrift**familie**:
 - `\rmfamily` oder `\textrm{}` (Standardschrift)
 - `\ttfamily` oder `\texttt{}` (true type/Schreibmaschinenschrift)
 - `\sffamily` oder `\textsf{}` (sans serif/serifenlose Schrift)
- Die Schrift**form** :
 - `\itshape` oder `\textit{}` (*Italics/kursiv*)
 - `\slshape` oder `\textsl{}` (*slanted/geneigt*)
 - `\scshape` oder `\textsc{}` (SMALL CAPS/KAPITÄLCHEN)
 - `\upshape` oder `\textup{}` (aufrechte Standardschrift).
- Die Schrift**serie** :
 - `\bfseries` oder `\textbf{}` (**Fett**)
 - `\mdseries` oder `\textmd{}` (mittlere Stärke)

Die Attribute sind voneinander unabhängig. Die Änderung eines Attributs bleibt solange wirksam, bis eine Attributsänderung *gleichen Typs* erfolgt⁴. Ändert man zum Beispiel das Familienattribut auf `\sffamily`, und wählt danach `\itshape` als Schriftform, bleibt das erste Attribut erhalten: heraus kommt eine *serifenlose und dann eine serifenlose und kursive Schrift*.

³Es sei denn, die Regel 2.2.2 trifft zu

⁴Es sei denn, die Regel 2.2.2 trifft zu

Kurzformen

Die Kurzformen `\rm` `\bf` `\sl` `\it` `\tt` `\sc` `\sf` können ebenfalls zur Wahl einer Schriftart verwendet werden. Diese Kurzformen legen aber alle drei Attribute einer Schrift komplett fest und sollten deshalb nur verwendet werden, falls nur das entsprechende Attribut geändert werden soll.

4.2 Textausrichtung

Standardmäßig werden L^AT_EX - Dokumente im Blocksatz gesetzt. Für zentrierten und einseitig bündig ausgerichteten Text gibt es eigene Umgebungen.

- Für zentrierten Text gibt es die `{center}`-Umgebung:

```
\begin{center}
  Zentrierter Text
\end{center}
```

In Manchen Fällen wird der Befehl `\centering` benutzt – beispielsweise zum Zentrieren einer nicht nummerierten Überschrift: `\section*{\centering Überschrift}`

- Für einseitig bündigen Text gibt es die Umgebungen `{flushleft}` und `{flushright}` :

```
\begin{flushleft}
  linksbündig ausgerichteteter Text
\end{flushleft}
```

Die gleiche Funktion erfüllen die Befehle `\rightline{rechts}` und `{\raggedright rechtsbündiger Text}`, entsprechend für links

- Für eingerückten Text (zB. für kürzere Zitate) gibt es die `{quote}`-Umgebung, bei längeren (mehreseitigen Zitaten) sollte die `{quotation}`-Umgebung benutzt werden, damit die Seiten gleichmäßig mit Text gefüllt werden.

4.3 Tabulatoren

Die Benutzung der Tab-Taste bewirkt keinen Tabstop im L^AT_EX - Dokument. In L^AT_EX können Tabulatoren mit Hilfe der `tabbing` Umgebung erzeugt werden:

```
\begin{tabbing}
  Text1...Text2...Text3
  Text1... \=Text2... \=Text3 \
           \>steht unter Text2! \
  das steht \>\unter Text3! \
\end{tabbing}
```

Zunächst werden in der ersten Zeile dieser Umgebung die Tabstops festgelegt, mit dem Befehl `\=`. Die Zeile muß mit `\>` abgeschlossen werden. In den folgenden Zeilen kann nun mit `\>` zum nächsten Tabstop gesprungen werden. Es kann auch eine Musterzeile zur Festlegung der Tabstops geschrieben werden, durch den Befehl `\kill` an deren Ende wird sie nicht mit ausgedruckt. Das könnte so aussehen:

```
\begin{tabbing}
  \hspace*{2em}\=mustertext\hspace*{1em}\= \kill
  ...
\end{tabbing}
```

4.4 Einfache Listen und Aufzählungen

Eine einfache Liste wird so gesetzt:

```
\begin{itemize}
  \item Text...
  \item Mehr Text...
  ...
\end{itemize}
```

In L^AT_EX sind drei vordefinierte Arten von Listen bekannt: einfache Listen mit *itemize*, nummerierte Listen mit *enumerate* und beschreibende Listen mit *description*. Alle Listen lassen sich verschachteln (maximal 4 Ebenen), auch unterschiedliche Typen von Listen sind ineinander schachtelbar (maximal 6 Ebenen).

Für jede Liste kann die Form der *items* frei gewählt werden. Beispiel: für einen Spiegelstrich anstatt eines fetten Punktes, genügt `\item[-]`, usw. Beliebige Symbole sind zulässig. Die Form der *items* kann auch global für das ganze Dokument geändert werden. Zum Beispiel ändert

```
\renewcommand\labelitemi{neues symbol}
```

das Symbol für die erste Verschachtelungsebene von einfachen Listen. Für jede Verschachtelungsebene gibt es einen Befehl, der verändert werden kann: `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`, `\labelenumi`, `\labelenumii`, usw.

4.5 Text unterstreichen

Das kann man mit dem Befehl `\underline{Text}`. So siehts aus: Text.

4.6 Fußnoten und Randnotizen

Fußnoten: Fußnoten lassen sich mit dem Befehl `\footnote{}` erzeugen:

```
Text, der eine Fußnote\footnote{Fußnotentext} erhalten soll.
```

Die Fußnoten werden standardmäßig laufend durchnummeriert. Der *Fußnotentext* wird in der Schriftgröße `footnotesize` unten auf der Seite ausgegeben, vom übrigen Text durch eine horizontale Linie getrennt.

Randnotizen Die Erzeugung von Randnotizen (Marginalien) erfolgt mit dem Befehl

```
\marginpar{randnotiz}
```

Randnotizen erscheinen rechts von der Textzeile in einem schmalen Bereich. Als Randnotiz können auch Sonderzeichen, wie zum Beispiel `← ♣ <` verwendet werden, um einen Text zu markieren. ♡

4.7 Boxen

Eine Box ist ein Stück Text, das von L^AT_EX als eine Einheit (wie ein einzelnes, evtl. sehr großes Zeichen) interpretiert wird. Eine Box kann daher nicht umgebrochen (auf mehrere Zeilen verteilt) werden, selbst wenn sie ihrerseits mehrzeiligen Text enthält (ausgenommen ist hier die *parbox*).

4.7.1 Allgemeine Boxen

- `\mbox{text}` ist eine einfache horizontale Box. Diese kann benutzt werden, um unerwünschte Silbentrennungen zu verhindern. Die Breite der Box wird von der Breite des darin eingefügten Textes bestimmt.
- `\fbox{Inhalt}` umrandet den Inhalt mit einem Rahmen.
- mit `\makebox[Breite][Position]{Text}` läßt sich die *Breite* der Box unabhängig vom *Text* bestimmen. Der *Text* in der Box kann die *Position* 1 (linksbündig), r (rechtsbündig) oder s (angegebene *Breite* wird links- und rechtsbündig gleichmäßig mit Text ausgefüllt). Bleibt *Position* leer, wird der Text in der Box zentriert.
- `\framebox[] [] {}` liefert das gleiche mit Rand
- `\raisebox{shift}{Inhalt}` ermöglicht es, Den *Inhalt* vertikal um den Betrag *shift* zur Grundlinie verschieben.
- `\parbox[Ausrichtung][höhe][innen-pos]{Breite}{Text}` ist eine vertikale Box und ermöglicht den Zeilenumbruch innerhalb der Box. Für *Breite* kann eine feste Größe wie *150mm* verwendet werden; oft empfehlen sich jedoch Längenangaben, die bereits vorhandene Größen wie etwa die Textbreite verwenden. So erhält man eine Box mit halber Textbreite durch: `\parbox{0.5\textwidth}`.

Die Box läßt sich mit *Ausrichtung* entweder an ihrer oberen (t), mittig (c) oder an ihrer unteren (b) Zeile auf die laufende Textzeile ausrichten. Auch die *höhe* und *Innenposition* mit t (oben) oder b (unten) kann optional eingestellt werden. Um eine solche Box mit einem Rand zu versehen, kann die Parbox in eine `\fbox{}` gepackt werden: `\fbox{\parbox[] [] {} }`
- `\begin{minipage}[] {Breite} Inhalt \end{minipage}` hat eine ähnliche Funktion wie eine Parbox. Der Vorteil einer Minipage ist, dass hier Elemente angewandt werden können, die in einer Parbox verboten sind (zum Beispiel Fußnoten). Hinsichtlich ihres Einsatzes ist sie mit einer verkleinerten Ausgabe einer ganzen Seite vergleichbar.

4.7.2 Balkenboxen

Eine Balkenbox ist ein mit Druckfarbe gefülltes Rechteck; dazu gehören auch vertikale und horizontale Linien. Erzeugt wird eine Balkenbox mit

```
\rule[lift]{breite}{höhe}
```

Dabei bezeichnet *lift* die optionale Verschiebung der Box im Verhältnis zur Grundlinie. Für *breite* und *höhe* muß nicht unbedingt eine Zahl eingesetzt werden. So läßt sich mit `\rule{0,75\textwidth}{0,25mm}` eine horizontale Linie, die drei Viertel der Textbreite einnimmt, erzeugen.

Eine Balkenbox mit der *breite* null bezeichnet man als „Stütze“. Sie kann benutzt werden, um vertikalen Abstand in Situationen zu erzeugen, wo die Verwendung der üblichen Befehle für vertikale Sprünge nicht möglich ist (zB in Tabellen).

4.7.3 Anpassbare Boxenparameter

Zwei Stilparameter von Rahmenboxen (`\fbox` und `\framebox`) können verändert werden:

1. Liniestärke mit `\fboxrule` (Voreinstellung: 0,4pt)
2. Leerraum zwischen Rahmen und Text mit `\fboxsep` (Voreinstellung: 4pt)

```
\setlength{\fboxrule}{0,25mm}
\setlength{\fboxsep}{1,5mm}
```

Treten diese Deklarationen im Vorspann auf, gelten sie global (also für das ganze Dokument), treten sie im laufenden Text auf, gelten sie ab der Stelle, wo der Befehl auftritt⁵.

⁵Es sei denn, die Regel 2.2.2 trifft zu

5 Einbinden von Graphiken

5.1 Graphik einbinden

Nach dem Laden des Makropaketes `{graphicx}` im Vorspann lassen sich Graphiken in ein \LaTeX -Dokument einbinden. Eine Skalierung ist u. a. mit den Optionen `width`, `height` und `scale` möglich, zum Beispiel:

```
\includegraphics[width = .3*\textwidth]{Datei}
```

für eine Graphik namens Datei, die horizontal soviel Platz einnehmen soll, wie $\frac{1}{3}$ der Textbreite.

5.2 Bilder im separaten Verzeichnis sortieren

Bei Dokumenten mit vielen Bildern kann es sinnvoll sein, alle Bilder in einem separaten Verzeichnis zu bewahren, um eine bessere Übersicht zu erhalten. In diesem Fall muss in der Dokumentenpräambel der Pfad dieses Verzeichnisses angegeben werden: `\graphicspath{{Pfad}}`. Beispiele: `\graphicspath{{./bilder/}}` falls die Bilder sich in einem Unterverzeichnis namens `bilder` befinden oder `\graphicspath{{../bilder/}}` falls das Verzeichnis `bilder` sich eine Ebene höher befindet.

5.3 Bildformate

Es lassen sich nicht alle beliebigen Bildformate einbinden. Mit PdfLaTeX (LaTeX => PDF) können (derzeitig) nur die Formate `*.png`, `*.jpg`, `*.tiff` oder `*.pdf` verwendet werden. Wählt man `dvi` oder `ps` als Ausgabeformat des Dokumentes, so können nur Bilder im `.ps`- oder `*.eps`-Format eingebunden werden (encapsulated Postscript).

Am einfachsten ist es daher, Bilder in anderen Formaten, (`*.bmp`, `*.gif`, usw.) zunächst in das gewünschte Format zu konvertieren. Dazu gibt es verschiedene Tools, wie etwa ImageMagick, das für verschiedene (kostenlose) Betriebssysteme (Linux, Windows, ...) zur Verfügung steht. Es enthält das Kommando `convert`:

```
convert bild.gif bild.eps
```

wandelt die Datei `bild.gif` in `bild.eps` um. Analog geht es mit anderen Formaten.

6 Gleitobjekte (Abbildungen, Tabellen)

6.1 Definition von Gleitobjekten

Gleitobjekte in \LaTeX sind Boxen, die nicht unbedingt da erscheinen müssen, wo sie im Quelltext stehen. In wissenschaftlicher Literatur bezeichnet dieser Begriff normalerweise Abbildungen und Tabellen in einem Dokument. Die Position von Gleitobjekten in der Ausgabedatei hängt von vielen Parametern ab (wie weit ist die Seite gefüllt, wie groß das Gleitobjekt, usw.). Voreingestellt kennt \LaTeX zwei Gleitobjektumgebungen: `{figure}` und `{table}`. Beispiel für `{figure}`:

```
\begin{figure}[!ht]
  \centering
  \includegraphics{Graphik}
  \caption{Beschreibung}
\end{figure}
```

6.2 Positionieren von Gleitobjekten

Ein optionaler Parameter (bis zu vier Buchstaben) bestimmt, die gewünschte Platzierung des Gleitobjektes:

h *here*: An der Stelle, wo die Umgebung auftritt.

t *top*: Am oberen Rand der aktuellen Seite.

b *bottom*: Am unteren Rand der aktuellen Seite. Der Platz über dem Gleitobjekt wird mit dem nachfolgenden Text aufgefüllt.

p *page of floats*: Gleitobjekte werden auf einer eigenen Seite gesammelt (zum Beispiel am Ende des Dokuments).

H *Here* (nur mit dem Paket `float` möglich): Auf jeden Fall an der Stelle, wo die Umgebung auftritt. Allerdings kann es dann im ungünstigen Fall vorkommen, dass eine Seite nicht vollständig gefüllt wird und trotzdem umgebrochen wird.

Wird keine Angabe gemacht, gilt `[tbp]`. Diese Angaben werden der Reihe nach ausgewertet. Die *tatsächliche* Platzierung erfolgt an der frühestmöglichen Stelle. Durch hinzufügen eines `!` ignoriert \LaTeX die meisten Einschränkungen im Platzierungsalgorithmus.

Gleitobjekte werden immer in der Reihe ihres Auftretens im Quelltext nummeriert und dargestellt.

6.3 Beschriften von Gleitobjekten

Allgemeines Prinzip: Gleitobjekte werden mit `\caption[Kurzform]{Beschriftung}` beschriftet. \LaTeX erzeugt eine durchlaufende Nummerierung aller Gleitobjekten. Die Kurzform ist optional und ermöglicht eine Kurzvariante der Beschriftung für das Gleitobjektverzeichnis (sinnvoll bei langen Beschriftungen). Standardmäßig werden Beschriftungen zentriert gesetzt. Deshalb werden Gleitobjekte meist zentriert positioniert: `\begin{table} \centering ... \end{table}`

Gleitobjektbeschriftung linksbündig mit caption: Mit dem Makropaket `\usepackage{caption}` lässt sich die Beschriftung weiter anpassen. Beispielsweise kann die Beschriftung von Abbildungen linksbündig gesetzt werden durch den Befehl `\captionsetup{singlelinecheck=off}`

Beschriftung über das Gleitobjekt mit topcapt: Manche Gleitobjekte (zB Tabellen) werden oft oben beschriftet. Dies ermöglicht das Paket `topcapt`, mit dem Befehl `\topcaption{}` statt `caption`. Damit wird der Abstand zwischen der Beschriftung und dem Gleitobjekt etwas größer. Für Beschriftungen unter dem Gleitobjekt kann weiterhin `\caption{}` verwendet werden.

Objektbezeichner anpassen: Tabellen werden bei der Verwendung des `\caption{}`-Befehls auf deutsch standardmäßig mit „Tabelle“ bezeichnet. Dieser Name kann geändert werden mit `\renewcommand{\tablename}{Bezeichnung}`. Analog bei Abbildungen mit `figurename`.

6.4 Gleitobjektverzeichnisse

Jede Abbildung, die in einer `{figure}`-Umgebung eingebettet ist, erhält einen Eintrag ins Abbildungsverzeichnis, welches mit `\listoffigures` gesetzt wird. Das Verzeichnis wird an der Stelle des Auftretens dieses Befehls gesetzt, dazu sind 2 L^AT_EX - Läufe notwendig. Dort stehen dann alle Abbildungen durchnummeriert und mit Seitenzahl versehen unter dem Titel der in `\caption{}` angegebenen Beschriftung.

Analog gibt es das Tabellenverzeichnis, welches mit `\listoftables` gesetzt wird und alle Tabellen aufnimmt, die in einer `{table}`-Umgebung eingebettet sind.

6.5 subfig – Gruppieren mehrerer Gleitobjekte

Mit dem Makropaket `\usepackage{subfig}` lassen sich mehrere kleine Objekte (Abbildungen, Tabellen) geschickt als ein einziges Gleitobjekt zusammenfassen. Die Syntax jedes so plazierten Objektes sieht so aus (innerhalb einer `figure`- oder `table`-Umgebung) :

```
\begin{figure}[!ht]
\centering
\subfloat[Beschriftung]{das erste Objekt \label{f1}} \hfill
\subfloat[Beschriftung]{zweites Objekt}\hfill
\subfloat[Beschriftung]{anderes Objekt}\hfill
\caption{Mehrere Objekte, siehe \subref{f1}}
\end{figure}
```

das erste Objekt	zweites Objekt	anderes Objekt
(a) Beschriftung	(b) Beschriftung	(c) Beschriftung

Abbildung 1: Mehrere Objekte, das wichtigste ist (a)

Die Unterobjekte werden dann alphabetisch nummeriert und können einzeln beschriftet werden. Auch hier können Labels gesetzt werden, um später in der Gesamtbeschriftung (`\caption{}`) mit `\subref{Schlüssel}` auf die Unterobjekte zu verweisen.

7 Einfache Tabellen

7.1 Definition

Eine Tabelle liefert uns die `{tabular}`-Umgebung. Sie hat einen notwendigen Parameter, der die Spalten festlegt und einen optionalen Parameter, der die Position der Tabellenzeile zum laufenden Text angibt. Das kann dann so aussehen:

```
\begin{tabular}[t]{cc}
Text & Text \\
\end{tabular}
```

In den geschweiften Klammern steht, wie die Textzeile einer jeden Spalte ausgerichtet sein soll (und damit auch, wie viele Spalten es geben soll). Im obigen Beispiel sind es zwei Spalten mit zentriertem Text (`{cc}`). Mögliche Formate für die Ausrichtung der Tabellenzeile sind:

1. `c`: zentriert
2. `l`: linksbündig
3. `r`: rechtsbündig
4. `p{Breite}`: Der Text dieser Spalte wird als top ausgerichtete `\parbox` der angegebenen Breite gesetzt. Analog gibt es noch `b{Breite}` und `c{Breite}` für entsprechend ausgerichtete Boxen.

Über den optionalen Parameter läßt sich die Ausrichtung der Tabelle angeben (in obigem Beispiel `t`). Ist nichts angegeben, so wird die Tabelle zentriert ausgerichtet. Wie bei Boxen gibts hier `t`, `c` und `b`.

Die Einträge der einzelnen Spalten werden mit einem `&` voneinander getrennt. Mit einem Doppelslash `\\` markieren wir das Ende der Tabellenzeile. Hier kann durch eckigen Klammern ein beliebiger vertikaler Abstand hinzugefügt werden, zum Beispiel: `\\[2ex]`.

7.2 Linien in Tabellen

- Mit `|` läßt sich in der Spaltendeklaration eine vertikale Linie zwischen 2 Textspalten setzen, also zum Beispiel so: `\begin{tabular}[t]{l|l}`. Auch Doppellinien sind möglich (mit `||`).
- Das Kommando `\hline` am Tabellenzeilenende (nach `\\`) erzeugt eine horizontale Linie unter dieser Zeile, mit `\hline\hline` bekommt man eine Doppellinie und mit `\hline` vor der ersten Tabellenzeile eine Linie über dieser Zeile.
- `\cline{n-m}` erzeugt eine horiz. Linie vom linken Rand der Spalte `n` bis zum rechten Rand der Spalte `m`. Dieser Befehl darf nur am Ende des Zeilentrennzeichens `\\` auftreten!
- `\vline` erzeugt einen vertikalen Strich über die Zeilenhöhe an der Stelle seines Auftretens.

7.3 Weitere Gestaltungselemente

Benutzerdefinierte Spaltentrenner Der Befehl `@{Inhalt}` definiert einen beliebigen (zusätzlichen) Spaltentrenner innerhalb einer existierenden Spalte. Beispiel: `\begin{tabular}[t]{r@{:}l}` für einen Doppelpunkt zwischen den zwei Spalten.

Zellen zusammenfassen Der Befehl `\multicolumn{num}{Ausr.}{Inhalt}` benutzt die nächsten `num` Spalten für `Inhalt`. Dabei werden vorige Formatierungen und Spaltentrenner der benutzten Spalten aufgehoben. `Ausr.` definiert die Ausrichtung der neuen Zelle, also `l`, `r` oder `c`, ggf. ergänzt durch eine vertikale Linie `|`. Dieser Befehl kann auch benutzt werden, um die Ausrichtung und Positionierung einzelner Zellen in der Tabelle zu verändern. Beispiel: `\multicolumn{2}{c}{Inhalt}` ersetzt an der entsprechenden Stelle die nächsten 2 Spalten durch eine einzige und zentriert dort den Inhalt.

Zeilenabstand Der Zeilenabstand für Tabellen kann global verändert werden durch `\renewcommand{\arraystretch}{Faktor}`. Default-Faktor ist 1.

Eine weitere einstellbare Länge ist `\extrarowheight`: damit kann zusätzlicher vertikaler Leerraum über Jede Zeile eingefügt werden. Beispiel: `\addtolength{\extrarowheight}{2mm}`.

7.4 Abstände in Tabellen

In manchen Fällen, zum Beispiel wenn die Schriftgröße in einer Tabelle nur für eine einzige Zelle geändert wird, ist der Text in einer Zelle zu nah an den vertikalen Rändern der Zelle. Um dies zu umgehen, gibt es zwei Möglichkeiten. Man kann innerhalb der Zelle unsichtbarer Abstand mit einer Balkenbox (`\rule{}{}`) erzeugt werden (siehe 4.7.2). Beispiel:

Titel 1	Titel 2
Eintrag	Eintrag
Eintrag	Eintrag

Titel 1	Titel 2
Eintrag	Eintrag
Eintrag	Eintrag

Titel 1	Titel 2
Eintrag	Eintrag
Eintrag	Eintrag

Die erste Tabelle oben wurde ohne Stütze gesetzt. Die mittlere Tabelle wurde mit Stütze gesetzt:

```
\begin{tabular}[t]{|c|c|}
\hline
\Large \bf \rule{0mm}{2ex} Titel 1 & \Large \bf Titel 2 \\ \hline
Eintrag & Eintrag \\ \hline
Eintrag & Eintrag \\ \hline
\end{tabular}
```

Die dritte Tabelle zeigt, was die Stütze wirklich macht (mit 1mm Breite statt 0mm).

Die zweite Möglichkeit ist es, in der Zelle eine parbox zu definieren mit festgelegter Höhe⁶:

```
\begin{tabular}[t]{|c|c|}
\hline
\parbox[c][5.5ex][c]{1.6cm}{\Large \bf Titel 1} &
\parbox[c][5.5ex][c]{1.6cm}{\Large \bf Titel 2} \\ \hline
Eintrag & Eintrag \\ \hline
Eintrag & Eintrag \\ \hline
\end{tabular}
```

Titel 1	Titel 2
Eintrag	Eintrag
Eintrag	Eintrag

7.5 array – Eine Erweiterung für Tabellen

Das Paket *array* ermöglicht in Tabellen das Einfügen eines beliebigen Inhaltes vor und/oder nach einer Spaltendeklaration (also l, c, r, usw). Die Syntax vor einer Spaltendeklaration sieht so aus: `>{Inhalt}`, nach einer Spaltendeklaration: `<{Inhalt}`. Kleines Beispiel:

```
\begin{tabular}{|>{\bf}c|c|} ...
```

für eine Tabelle, in der die gesamte erste Spalte in Fettschrift gesetzt wird.

⁶Um sich hier Tipparbeit zu ersparen, kann ein eigener Befehl dazu definiert werden, mehr dazu in Abschnitt 13.2

7.6 Gesamtbreite von Tabellen festlegen

Dies ist mit der Umgebung *tabular** möglich:

```
\begin{tabular*}{breite}{Spaltendekl.}
...
\end{tabular*}
```

7.7 tabularx – Spaltenbreite automatisch berechnen

Das Paket *tabularx* ermöglicht es, nicht nur die Gesamtbreite von Tabellen festzulegen, sondern auch mit einer automatischen Berechnung der Breite bestimmter Spalten. Eine solche Spalte wird mit **X** bezeichnet und wird von L^AT_EX als p{}-Spalte gesetzt. Beispiel:

```
\begin{tabularx}{\textwidth}{p{2cm}Xp{2cm}}
...
\end{tabularx}
```

für eine Tabelle, die so Breit wie der Text ist, mit 3 Spalten, von denen die erste und die letzte 2 cm Breit ist. Die Breite der mittleren Spalte (X) wird automatisch berechnet, so dass die Tabelle die Textbreite hat.

7.8 multirow – Vertikale Ausrichtung

Das Paket *multirow* ermöglicht es, ähnlich wie *multicolumn*, Tabellen mit mehrzeiligen Spalten Konstruieren:

```
\multirow{Anz. Zeilen}{Breite}{Inhalt}
```

Es ist auch möglich, *multirow* mit *multicolumn* zu kombinieren: `\multicolumn{}{}{\multirow{}{}}{}`.

7.9 tabs – Einstellen von Zeilenabständen

Wird das *tabs*-Paket im Vorspann geladen, so vergrößert sich der Zeilenabstand in Tabellen. Dabei gibt es 2 veränderbare Längenbefehle:

- `\tablinesep` : Der Mindestdurchschuss zwischen aufeinander folgenden Textzeilen in einer Tabelle (für die *array*-Umgebung ist die Entsprechung `\arraylinesep`). Standardwert ist 1 pt.
- `\extrarulesep` : Zusatzabstand, der über und unter allen `\hline` - und `\cline` - Befehlen eingefügt wird. Standardwert ist 3pt.

Außerdem ermöglicht *tabs* noch, den Abstand unterhalb von Linien manuell zu vergrößern, zB. fügt `\hline[2mm]` zwei millimeter Zusatzabstand unter der entsprechenden Linie ein.

Vorsicht: *tabs* nicht mit dem Paket *array* kombiniert werden.

7.10 dcolumn – Ausrichtung am Dezimalpunkt

Das Paket *dcolumn* ermöglicht in Tabellen die Ausrichtung von Kommazahlen am Dezimalpunkt. Die kann sinnvoll sein, bei Tabellen mit Text und Zahlen. Der entsprechende Tabelleneintrag sieht dann so aus:

```
D{Eingabezeichen}{Ausgabezeichen}{Kommastellen links . Kommastellen rechts}
```

Kleines Beispiel: `\begin{tabular}{|c|c|D.,{2.2}|} Text & Text & 32.41 \\ \dots` für eine Tabelle mit 3 Spalten, wo in der dritten Spalte Zahlen (oder sonstiges) eingetragen mit maximal zwei Kommastellen links und maximal 2 rechts. Die Zahlen werden dann untereinander zentriert am Komma ausgerichtet.

7.11 supertabular – Mehrseitige Tabellen

Hier ist die Syntax dazu:

```
\tablecaption{beschriftung}

\tablefirsthead{erste Zeile auf der ersten Seite}
\tablehead{erste Zeile auf folgenden Seiten}
\tabletail{letzte Zeile}
\tablelasttail{letzte Zeile auf der letzten Seite}

\begin{supertabular}{spaltendekl.}
...
\end{supertabular}
```

Analog zu normalen Tabellen gibt es auch *supertabular**, wenn die Breite explizit festgelegt werden muss. Oft nehmen solche Tabellen die gesamte Textbreite:

```
\begin{supertabular*}{\textwidth}{1!\extracolsep{\fill}}{cr} ...
```

7.12 Fußnoten in Tabellen

Der `footnote`- Befehl ist innerhalb von LR-Boxen, Tabellen und im mathematischen Modus *nicht* erlaubt, stattdessen wird der Befehl

```
\footnotemark
```

verwendet. Er setzt lediglich eine *Fußnotenmarkierung* in den o. g. unerlaubten Modi (Tabelle, etc.). Außerhalb des unerlaubten Modi (nach der Tabelle, o. ä.) wird nun der Fußnotentext zu unserer Markierung gesetzt:

```
\footnotetext{fußnotentext}
```

Hinweis

Treten in einer Umgebung zwei `\footnotemark`-Befehle hintereinander auf, steht der Fußnotenzähler am Ende der Umgebung auf 2. Dieser Wert wird dann an `\footnotetext` übergeben, so daß der Fußnotentext von Fußnote 1 nun die Nummer 2 trägt und der von Fußnote 2 ebenfalls – Der Zähler steht ja immer noch auf 2, da kein neuer `\footnotemark`-Befehl erfolgte. Man muß in diesem Fall also vor dem ersten `\footnotetext`-Feld den Zähler auf 1 setzen und ihn dann vor dem zweiten `\footnotetext`-Feld um 1 erhöhen:

```
\fbox{Erste Fußnotenmarkierung\footnotemark Zweite Fußnotenmarkierung\footnotemark }
```

```
\setcounter{footnote}{1}
\footnotetext{Text zur ersten Fußnote}
\stepcounter{footnote}\footnotetext{Text zur zweiten Fußnote}
```

8 Mathematische Formeln

Bei Dokumenten mit anspruchsvollen Mathematikformeln empfiehlt es sich, in der Präambel die Pakete *amsmath* und *amssymb* zu laden, damit mehr Symbole und Befehle zur Verfügung stehen.

Formeln werden \LaTeX durch einen beschreibenden Text erzeugt. Damit \LaTeX weiß, dass der folgende Text als Formel zu interpretieren ist, muss man eine der mathematischen Umgebungen benutzen. Formeln können im laufenden Text gesetzt werden, dazu benutzt man die Umgebung `math`:

```
\begin{math} Formeltext \end{math}
```

Als Abkürzung kann man `$ Formeltext $` verwenden.

Innerhalb einer mathematischen Umgebung arbeitet \LaTeX im mathematischen Modus. Hier haben Leerzeichen keinerlei direkte Wirkung mehr, vielmehr setzt \LaTeX eigene Leerzeichen und Abstände nach vorgegebenen Kriterien. Leerzeilen dürfen gar nicht benutzt werden. Will man Leerzeilen benutzen, um den Eingabetext übersichtlicher zu machen, muss man sie mit % auskommentieren.

Mathematische Umgebungen: Die Umgebung `math` erlaubt es, Formeln im laufenden Text zu integrieren.

Lange Formeln werden gewöhnlich nicht im laufenden Text, sondern abgesetzt dargestellt. Abgesetzte Formeln werden häufig nummeriert. Folgende mathematische Umgebungen stehen zur Verfügung:

<code>math</code>	für Formeln im laufenden Text
<code>displaymath</code>	für einfache abgesetzte Formeln. Kurzform: <code>\$\$ Formeltext \$\$</code>
<code>equation</code>	für nummerierte abgesetzte Formeln
<code>eqnarray</code>	für mehrzeilig ausgerichtete nummerierte Formeln
<code>eqnarray*</code>	für mehrzeilig ausgerichtete, nicht nummerierte Formeln

Zeichen im mathematischen Modus: Im mathematischen Modus können folgende Zeichen direkt eingegeben werden:

```
+ - = < > / : ! ' | [ ( ) ]
```

Text in Formeln: Möchte man normalen Text in Formeln einfügen, so muss man ihn mit `\text{Text}` schreiben, da er sonst als Symbole erfasst wird und entsprechend kursiv gesetzt wird. Für einzelne Wörter kann auch eine Abkürzung genutzt werden: `\rm Wort` innerhalb einer Formel.

Klammern: Klammern, die ihre Größe an der folgenden Formel anpassen, werden erzeugt mit

```
\left Klammer Formel \right Klammer
```

Wünscht man nur auf einer Seite einer Gleichung eine große Klammer, so muss an der Stelle, wo keine Klammer erscheinen soll, ein Punkt stehen, zB: `$$\left. Formel \right \}$` für eine große, geschweifte Klammer auf der Linken Seite.

Abstand: Im Mathemodus wird horizontaler Abstand automatisch formatiert. Mehr Abstand wird zB. durch `\;` erzeugt. Für größere Abstände stehen die Befehle `\quad`, `\qquad` und `\hspace{}` zur Verfügung.

Hoch- / Tiefstellung: Hochstellung und Tiefstellung erfolgt mit den Zeichen `^` und `_`. Also x^2 mit `x^2` und x_{i+1}^n mit `x_{i+1}^n`.

Brüche: Für Brüche steht der Befehl `\frac{Zähler}{Nenner}` zur Verfügung. `$ \frac{1}{x+y} $` erzeugt $\frac{1}{x+y}$

Wurzelausdrücke: Wurzelausdrücke werden mit dem Befehl `\sqrt{Wurzel}{Radikant}` erzeugt. Also `$$\sqrt[3]{27}$$` für $\sqrt[3]{27}$.

Summen, Integrale und Limes: Summen und Integrale werden mit den Befehlen `\sum` für \sum und `\int` für \int erzeugt. Obere und untere Grenzen können entweder mit den Hoch- bzw. Tiefstellungszeichen angegeben werden, oder mit

`\limits_{\textit{untere Grenze}}^{\textit{obere Grenze}}`. Beispiel: `\sum\limits_{i=1}^n n^2` ergibt $\sum_{i=1}^n n^2$
und `\lim\limits_{x\to\infty}\ln(x)` erzeugt $\lim_{x\rightarrow\infty} \ln(x)$

Arrays: Arrays funktionieren im Mathemodus genauso wie die *tabular*-Umgebung. Ein Beispiel:

```


$$\begin{array}{r} 1 + 2(3z-2) \\ 3x+4 + 3(2z-5) \end{array}$$


```

ergibt:
$$\begin{cases} 1 + 2(3z-2) \\ 3x+4 + 3(2z-5) \end{cases}$$

Auch die gleichen Spaltenrenner wie in der *tabular*-Umgebung (`@{}`) sind möglich.

Matrizen: Das *amsmath*-Paket liefert u.A. die *matrix*-Umgebung, mit maximal 10 Spalten:

```

\begin{matrix}
Eintrag & \& Eintrag & \& \dots \\
\end{matrix}

```

Drunter und drüber Stapeln: Mit `\stackrel{\textit{oben}}{\textit{unten}}` können beliebige Symbole (oder Text) über andere gestapelt werden:

`(3x)' \stackrel{\text{Formel 61}}{=} 3` ergibt $(3x)' \stackrel{\text{Formel 61}}{=} 3$.

Mit `\overbrace{\textit{unten}}^{\textit{oben}}` und `\underbrace{\textit{oben}}_{\textit{unten}}` werden entsprechend geschweifte Klammern gesetzt.

Mit `\overset{\textit{oben}}{\textit{unten}}` und `\underset{\textit{unten}}{\textit{oben}}` kann ebenfalls ein Beliebiger Text über oder unter einem Symbol gesetzt werden, und `\substack{\textit{Text} \\ \textit{Text}}` ermöglicht sogar Zeilenumbrüche innerhalb des Textes:

`(3x)' \underset{\substack{\text{Ableitungs-} \\ \text{regeln}}}{=} 3` ergibt $(3x)' \underset{\substack{\text{Ableitungs-} \\ \text{regeln}}}{=} 3$

Einfache Fallunterscheidungen: Die *cases*-Umgebung ermöglicht einfache Fallunterscheidungen:

```

f(x) = \begin{cases}
3 & \text{falls } x > 3 \\
0 & \text{sonst}
\end{cases}

```

ergibt: $f(x) = \begin{cases} 3 & \text{falls } x > 3 \\ 0 & \text{sonst} \end{cases}$

9 Literaturverzeichnisse erstellen

9.1 Einfache Literaturverzeichnisse

Ein Literaturverzeichnis kann mit der folgenden Umgebung gesetzt werden:

```

\begin{thebibliography}{Einrücktiefe}
\bibitem[Marke]{Schlüssel} Eintrag
\end{thebibliography}

```

Sie setzt das Literaturverzeichnis an der Stelle ihres Auftretens und versieht es je nach der jeweiligen Spracheinstellung mit einer Überschrift wie **Literaturverzeichnis**, **Bibliography**, etc. Folgende Parameter können angegeben werden:

<i>Einrücktiefe</i>	bestimmt, wie weit der Text hinter der Markierung eingerückt wird. Die Breitenangabe kann mit <code>\hspace{}</code> gemacht werden. Besser ist es, ein Stück Mustertext einzufügen, etwa die höchste erwartete laufende Nummer (zB. 999 bei maximal dreistelligen Nummern).
<code>\bibitem</code>	fügt einen Eintrag zum Literaturverzeichnis hinzu.
<i>Marke</i>	ist eine (optionale) Textmarke, mit der der Eintrag anstelle einer laufenden Nummer im Literaturverzeichnis markiert wird.
<i>Schlüssel</i>	ist ein notwendiger Parameter, der ein eindeutiges Schlüsselwort angibt, unter dem im laufenden Text auf den Eintrag Bezug genommen wird.
<i>Eintrag</i>	ist ein (beliebiger) Text, der die Literaturangabe darstellt.

Auf den Eintrag mit dem Schlüsselwort *Schlüssel* kann nun mit

```
\cite{Schlüssel}
```

im Text Bezug genommen werden. Es erscheint an dieser Stelle die laufende Nummer des Eintrags oder die optionale *Marke*.

9.2 Literaturverzeichnisse mit BibTeX

Man kann zwar alle Literaturangaben von Hand eingeben, doch ist es nützlicher und praktischer, die Literatur in einer Datenbank, i.e. einer oder mehrerer Dateien zu vereinigen und mit einem Schlüsselwort im Dokument aufrufen zu können. Diese Möglichkeit bietet das Zusatzprogramm BibTeX.

Anstatt in den Text schreiben wir unsere Literaturangaben in eine „Datenbank“, für BibTeX sind das Dateien mit der Endung `*.bib`. Ein Eintrag in einer solchen Datenbank könnte so aussehen:

```

@Book{kop,
author = {Kopka, Helmut},
ALTEditor = {},
title = {LaTeX Einführung},
publisher = {Addison-Wesley},
year = {1994},
}

```

Der *Eintrag* wurde mit dem Schlüsselwort *kop* versehen, damit auf ihn Bezug genommen werden kann. Es gibt unterschiedliche Eintragstypen, in obigem Beispiel `book`. Sie unterscheiden sich darin, dass sie je unterschiedliche Angaben erfordern. Eine Übersicht über die verschiedenen Eintragstypen befindet sich

im TexnicCenter in der Online-Hilfe, Stichwort "bibtex entry types", oder auch im Internet unter <http://de.wikipedia.org/wiki/Bibtex>.

Bei `book` sind die Felder `author` oder `editor`, sowie `title`, `publisher` und `year` zwingend, alle anderen optional. In anderen Eintragstypen kommen zusätzliche Felder hinzu und andere fallen weg. Zur Kennzeichnung der Einträge sind sowohl geschweifte Klammern als auch Anführungszeichen erlaubt.

Die Datenbank muß noch in das \LaTeX -Dokument eingebunden werden. Dies geschieht (an der Stelle, wo das Literaturverzeichnis stehen soll) durch den Befehl

```
\bibliography{Datenbank}
```

Als Datenbank läßt sich auch eine Liste von Dateien – durch Kommata getrennt – angeben. Der Name einer `.bib`-Datei darf keine Umlaute enthalten, und die Datei(en) müssen ohne `.bib` angegeben werden. Die Einträge für die Datenbank können entweder manuell eingegeben werden, oder im Internet gesucht werden, zum Beispiel unter google scholar: <http://scholar.google.de/>. Hier muss bei den Einstellungen der Bibliographiemanager konfiguriert werden, so dass nach BibTeX importiert werden kann.

Stile in BibTeX

Für das Literaturverzeichnis stehen eine Reihe von *Stilen* zur Verfügung, die über den Befehl

```
\bibliographystyle{Stil}
```

aufgerufen werden:

plain Die Einträge im Literaturverzeichnis erfolgen nach alphabetisch geordneten Autorennamen. Die einzelnen Eintragungen erhalten als Kennzeichnung laufende Nummern in eckigen Klammern, beginnend mit 1. Diese Kennzeichnung erscheint im laufenden Text bei den einzelnen `\cite`-Befehlen.

unsrt Die Eintragungen im Literaturverzeichnis erfolgen in der Reihenfolge der einzelnen `cite` und `nocite`-Befehle. Ansonsten wie `plain`.

alpha Anordnung wie `plain`, jedoch erhalten die Eintragungen keine Zahl, sondern eine Abkürzung des Autorennamens, z. Bsp. `kop94`.

abbrv Wie `plain`, jedoch werden Vornamen, Monatsnamen und Titel von Zeitschriften abgekürzt.

Für deutschsprachige Dokumente wurden diese Stile ins deutsche übersetzt. Lädt man das Paket *bibgerm* im Vorspann, so kann man die Stile `gerplain`, ..., `gerabbrv` benutzen. Es gibt mittlerweile auch DIN-Normen⁷ für Literaturverzeichnisse und -zitate in deutschsprachigen Dokumenten: `plaindin`, `unsrtin`, `alphadin`, `abbrvdin` und `natdin` (für naturwissenschaftliche Arbeiten).

Den BibTeX-eigenen Stilen lassen sich durch Zusatzpakete noch andere hinzufügen.

Innerhalb des Textes wird auf Einträge im Literaturverzeichnis wie gewohnt mit dem `\cite`-Befehl verwiesen. Auch hier gibt es mehr Möglichkeiten mit Zusatzpakete, zB kann mit dem Paket *natbib* der deutsche stil `dinat` ausgewählt werden.

Hinweis: Im Literaturverzeichnis tauchen nur die Einträge auf, die im Dokument auch zitiert wurden. Möchte man, dass ein nicht im Text durch `\cite` zitierter Eintrag dennoch im Literaturverzeichnis auftaucht, muß dieser innerhalb des Textes oder im Vorspann mit

```
\nocite{Schlüsse} oder
\nocite{*}
```

gekennzeichnet werden. Die Sternvariante stellt sicher, dass alle Einträge aufgenommen werden.

⁷siehe www.haw-hamburg.de/pers/Lorenzen/bibtex/

10 Farbdruck und Text drehen mit \LaTeX

Um unter \LaTeX farbig zu drucken, steht dem Anwender das Paket `color` zur Verfügung.

10.1 Farben Definieren

Um eine bestimmte Farbe verwenden zu können, muß diese erst einmal definiert werden (Die Basisfarben sind bereits vordefiniert):

```
\definecolor{Farbname}{Farbmodell}{Farbdefinition}
```

Farbname ist ein beliebiger, vom Anwender frei wählbarer Name. Die Seitenbeschreibungssprache Postscript stellt eine Reihe von *Farbmodellen* zur Mischung und zum Aufruf von Farben zur Verfügung. Dies sind `rgb`, `cmymk`, `gray`, `named`.

Von diesen *Farbmodellen* ist die *Farbdefinition* abhängig. So basiert das Modell `cmymk` auf den Grundfarben Cyan, Magenta, Gelb und einem zusätzlichen Schwarzanteil. Die Anteile dieser Grundfarben an der durch *Farbname* definierten Farbe werden mit einer Kombination aus vier Dezimalzahlen dargestellt. Beispiele:

```
\definecolor{OliveGreen}{cmymk}{.64,0,.95,.40}
\definecolor{col1}{rgb}{.5,.2,.65}
\definecolor{col2}{gray}{.3}
```

10.2 Farben benutzen

Ein aus bereits definierten Farben bestehender Text läßt sich mit folgenden Befehlen erzeugen:

```
\color{Farbname} farbiger Text      oder      \textcolor{Farbname}{farbiger Text}
```

10.3 Farbige Boxen

Farbige Boxen lassen sich mit den Befehlen `\colorbox` (entsprechend `\mbox`) und `\fcolorbox` (entsprechend `\fbox`) erzeugen:

```
\colorbox{Farbname}{Text}      bzw.
\colorbox[Farbmodell]{Farbdefinition}{Text}
```

```
\fcolorbox{Farbe-der-Umrandung}{Farbe-des-Hintergrunds}{Text}      bzw.
\fcolorbox[Farbmodell]{Farbdefinition-Umrandung}{Farbdefinition-Hintergrund}{Text}
```

10.4 Gedrehter Text

Eine Möglichkeit, Text gedreht darzustellen bietet das Paket `rotating`. Drei Umgebungen stehen zur Verfügung:

```
sideways  gibt Inhalt der Umgebung um 90 Grad gedreht aus (gegen die Uhr)
turn      gibt den Inhalt der Umgebung gedreht aus (beliebiger Winkel)
rotate    wie turn, läßt jedoch keinen zusätzlichen Platz frei
```

Während `sideways` ohne Parameter auskommt, muß bei `turn` und `rotate` der Drehwinkel angegeben werden:

```
\begin{rotate}{Winkel}
Inhalt
\end{rotate}
```

bzw. `\begin{turn}{Winkel} ...`

Innerhalb dieser drei Umgebungen kann ein Text, aber auch Boxen oder Tabellen stehen. Für gedrehte `table` und `figure`-Umgebungen gibt es noch die Umgebungen `sidewaystable` und `sidewaysfigure`.

11 Umgang mit Längen

L^AT_EX kennt zahlreiche vordefinierte Längen, die mit Standardwerten vorbesetzt sind. Eine Auswahl der gängigsten vorhandenen Längenbefehle können Sie Tabelle 6 entnehmen.

Diesen voreingestellten Längen können Sie beliebige Größen zuweisen:

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

Maßangaben müssen dabei immer aus einer Zahl und einer Dimension bestehen. Dabei gibt es zwei Möglichkeiten für die Dimension:

1. Verwendung einer *festen Dimension* wie **cm**, **mm**, **in**.
2. Verwendung einer *variablen Dimension*: **ex** ist die Höhe eines kleinen x, **em** die Länge eines großen M in der gerade aktuellen Schriftart.

Die Verwendung einer variablen Dimension bietet sich da an, wo die Länge stark mit der Schriftgröße zusammenhängt, also etwa bei Zeilenabständen, der Größe von Boxen, usw.

Als Maßangabe kann auch der Wert eines anderen Längenbefehls verwendet werden. Einige L^AT_EX - Längenbefehle sind in Tabelle 6 aufgelistet.

<code>\parindent</code>	Einrückung der ersten Zeile eines Absatzes	<code>\textwidth</code>	Textbreite
		<code>\textheight</code>	Texthöhe
		<code>\paperheight</code>	phys. Seitenhöhe
<code>\parskip</code>	Abstand zwischen zwei Absätzen	<code>\paperwidth</code>	phys. Seitenbreite
		<code>\oddsidemargin</code>	linker Rand ¹
<code>\baselineskip</code>	Abstand zwischen zwei Zeilen ³	<code>\evensidemargin</code>	rechter Rand ²
		<code>\topmargin</code>	oberer Rand bis Oberkante der Kopfzeile
		<code>\headheight</code>	Höhe der Kopfzeile
		<code>\headsep</code>	Abstand zwischen Unterkante Kopf und Oberkante des Textkörpers
<code>\footskip</code>	Abstand zwischen Unterkante des Textkörpers und Unterkante der Fußzeile	<code>\topskip</code>	Abstand von der Oberkante des Textkörpers bis zur Grundlinie der 1. Zeile

Tabelle 6: Einige L^AT_EX - Längenbefehle

¹Bzw. Rand der ungeraden Seite im doppelseitigen Layout

²Bzw. Rand der geraden Seite im doppelseitigen Layout

³Der tatsächliche Zeilenabstand ergibt sich aus `baselineskip` multipliziert mit `baselinestretch`. Letzterer liegt standardmäßig bei 1. Zur sinnvollen Änderung des Zeilenabstands sollte nur `baselinestretch` verändert werden.

Mit dem Befehl

```
\addtolength{Längenbefehl}{Maßangabe}
```

kann der Wert von *Längenbefehl* um den Wert *Maßangabe* vergrößert oder (bei negativer *Maßangabe*) verkleinert werden.

Die Maßangabe lässt sich auch von der Größe eines Mustertextes ableiten. Dies funktioniert für die Größen Breite, Höhe und Tiefe (Tiefe: Ausdehnung unterhalb der Grundlinie) eines Textes:

```
\settowidth{Längenbefehl}{Text}
```

```
\settoheight{Längenbefehl}{Text}
```

```
\settodepth{Längenbefehl}{Text}
```

Auch eigene Längenbefehle können definiert werden:

```
\newlength{meinelängenbefehl}
```

Dies kann vor allem in längeren Arbeiten praktisch sein, weil dann bestimmte Längen an einer einzigen Stelle für das ganze Dokument eingestellt werden können. Eigene Längenbefehle können auch nötig sein, wenn eine Länge weiter verarbeitet werden muss.

Den Wert einer Länge kann auch ausgegeben werden. So liefert

```
\the\parskip
```

den aktuellen Wert von `\parskip`, in diesem Script wäre das: 0.0pt plus 1.0pt, ein variabler Wert: plus und minus können drangehängt werden, damit L^AT_EX beim formatieren der Absätze ggf „schummeln“ kann, um ungünstige Seitenumbrüche zu verhindern.

11.1 Die Längen des Dokumentes automatisch skalieren

Die scr-Dokumentklassen ermöglichen die Einstellung der Längen als DIV-Werte. Diese werden als Optionen für die Dokumentklassen angegeben:

```
\documentclass[DIV16]{scrreprt}
```

Hier wurde der DIV-Wert auf 16 gesetzt, und alle Größen wurden automatisch skaliert, damit das Layout für die Längen des Dokumentes größer wird.

Alternativen dazu bietet das Paket `geometry`.

11.2 layout – Das Seitenlayout im Dokument ausgeben

Das Seitenlayout kann durch die im Kapitel **Längen** angegebenen Parameter beeinflusst werden. Mit dem Befehl `\layout` wird eine Übersicht aller Längen des aktuellen Dokuments ausgegeben:

```
...
\usepackage{layout}
\begin{document}
... \layout
```

12 Umgang mit Zählern

Definition: \LaTeX verfügt über interne Zähler, die zum Beispiel für die durchgehende Nummerierung von Kapiteln, Abbildungen, Gleichungen, Fußnoten usw. zuständig sind. Bei jedem Auftreten der entsprechenden Umgebung wird der Wert des zugeordneten Zählers um 1 erhöht. \LaTeX -interne Zähler sind in Tabelle 7 aufgelistet.

part	page	enumi
chapter	equation	enumii
section	figure	enumiii
subsection	table	enumiv
subsubsection	footnote	
paragraph	mpfootnote	
subparagraph		

Tabelle 7: \LaTeX -eigene Zähler

Die meisten Zähler sind selbsterklärend. `enumi` ... `enumiv` beziehen sich auf die vier Schachtelungsebenen der `enumerate`-Umgebung, `mpfootnote` auf die Fußnoten in Minipages.

Zählerwert ändern: Mit dem Befehl

```
\setcounter{Zaehler}{num}
```

kann jedem dieser *Zähler* ein ganzzahliger Wert *num* zugewiesen werden. Mit

```
\stepcounter{Zaehler}
```

wird der Zähler um 1 erhöht; so wird in der durchgehenden Nummerierung eine Einheit „übersprungen“. Mit

```
\addtocounter{Zaehler}{num}
```

kann ein Zähler um einen beliebigen Betrag erhöht oder durch Eingabe einer negativen Zahl für *num* vermindert werden.

Wert des Zählers ausgeben: Die Ausgabe eines Zählerstandes ist zum Beispiel mit dem Befehl

```
\arabic{Zaehler}
```

möglich. Hier wird der Zählerstand in arabischen Ziffern ausgegeben. Weitere Möglichkeiten zum Format der Zählerstandsausgabe sind `roman`, `Roman`, `alph` und `Alph`.

Neue Zähler definieren: Auch eigene Zähler können definiert werden, der verwendete Name darf aber nicht mit dem Namen eines vordefinierten Zählers übereinstimmen:

```
\newcounter{meinzaehler}
```

Der neu definierte Zähler kann nun in gleicher Weise wie die vordefinierten \LaTeX -Zähler verwendet werden.

Undefinieren von \LaTeX -interne Zähler Auch \LaTeX -interne Makros können undefiniert werden. Ein solches Makro ist zum Beispiel

```
\thesection
```

Es wertet den zu `\section{}` gehörenden Zähler aus und führt bei Aufruf in der Bearbeitungsklasse `article` diese Befehlsfolge aus:

```
\arabic{section}
```

Möchte man stattdessen eine Kapitelnummerierung in großen römischen Zahlen, läßt sich dies folgendermaßen umdefinieren:

```
\renewcommand{\thesection}{\Roman{section}}
```

Verschiedene Zähler können so verändert werden: `\thepage` für die Seitennummerierung, `\theequation` für die Nummerierung mathematischer Gleichungen, usw.

13 L^AT_EX Besonderheiten

13.1 Große Dokumente schreiben mit L^AT_EX

Bei längeren L^AT_EX - Dokumenten, ist es sinnvoll, den Quelltext in mehreren Teildokumenten zu untergliedern. Dies gestatten die Befehle `\input` und `\include`. Mit dem Befehl

```
\input{Datei}
```

wird der Inhalt von *Datei* (Dateiname einer *.tex-Datei ohne Endung) an dieser Stelle in das Dokument eingefügt. Dieser Befehl darf sowohl im Vorspann als auch im Textteil stehen. Schachtelungen mehrerer `\input`-Befehle sind möglich. Während `\input` stets alle angegebenen Dateien bearbeitet, kann der Befehl

```
\include{Datei}
```

im Textteil, kombiniert mit dem Befehl

```
\includeonly{Datei1,Datei2,Datei3,...}
```

im Vorspann erreichen, daß jeweils nur die in `\includeonly` festgelegte Liste von Dateien eingefügt wird. Dies kann sinnvoll sein, bei großen Projekten mit vielen Teildokumenten, wenn gerade nur ein Teildokument bearbeitet und betrachtet werden soll. Folgende Dinge sind noch zu beachten:

1. Es werden alle Verweise aufgelöst, auch auf diejenigen Teile des Dokuments, die nicht in `\includeonly` aufgeführt sind und die daher gerade nicht bearbeitet werden.
2. Jeder Teil eines Dokuments, der mit `\include` eingelesen wurde, fängt auf einer neuen Seite an.
3. `\include`-Befehle dürfen nicht geschachtelt werden! Eine Eingabedatei, die mit `\include` eingelesen wird, darf aber `\input`-Befehle enthalten.
4. Verweise auf Teile des Dokuments, die gerade nicht bearbeitet werden, werden nicht aktualisiert, wenn sich beispielsweise Seitenzahlen durch hinzugefügten Text verschieben.

13.2 Eigene Befehle definieren

Oft benutzte Befehlsfolgen können in eigenen Kurzkommandos zusammenfaßt werden:

```
\newcommand{\Befehl}{Definition}
```

Der neu definierte `\Befehl` darf einen beliebigen Namen tragen, welcher jedoch nicht ein schon existierendes L^AT_EX -Kommando bezeichnen darf. Dem `Befehl` ist ein `\` vorangestellt, und er kann dann wie ein normaler L^AT_EX -Befehl verwendet werden. In *Definition* erscheint der Befehl oder eine Folge von Befehlen, die an der Stelle des Aufrufs von `\Befehl` ausgeführt werden soll.

Befehle mit Argumente

Möglich ist auch, mehrere existierende Befehle, die eine variable Struktur (Formel, Text) einschließen, zu einem einzigen Befehl zusammenzufassen, dem eben jene Struktur als Parameter übergeben wird:

```
\newcommand{\Befehl}[Anzahl-der-Parameter]{Definition}
```

Anzahl-der-Parameter ist eine Zahl und legt fest, wie viele Argumente der neue `Befehl` haben soll. In die *Definition* werden die auszuführenden L^AT_EX -Befehle eingetragen, und die Stellen, an denen die Argumente eingefügt werden sollen, mit *#Parameter* gekennzeichnet. Steht also in *Anzahl-der-Parameter* 2, muss in der Definition an den entsprechenden Stellen #1 und #2 stehen. Kleines Beispiel: ein neuer Befehl mit 3 Parametern.

```
\newcommand{\beschr}[3]{#1 ist #2 und hat #3 Augen}
```

```
Wir beschreiben nun ein Paar Menschen:\
\beschr{David}{groß}{blaue}. \beschr{Tina}{klein}{grüne}. \beschr{Berta}{alt}{braune}.
```

Dies ergibt:

Wir beschreiben nun ein Paar Menschen:

David ist groß und hat blaue Augen. Tina ist klein und hat grüne Augen. Berta ist alt und hat braune Augen.

Vorsicht!

Wenn ein schon existierender Befehl modifiziert werden soll, kann nicht mit `\newcommand` gearbeitet werden. Stattdessen benutzt man `\renewcommand`. Dieser Befehl hat dieselbe Syntax wie `\newcommand`.

13.3 fancyhdr – Den Kolumnentitel anpassen

Das Paket *fancyhdr* ermöglicht eine freie Gestaltung des Kolumnentitels, am oberen und unteren Rand der Seite. Durch

```
\pagestyle{fancy}
```

wird eine horizontale Linie unter dem oberen Kolumnentitel gezogen. Sechs Befehle stehen u.a. zur Verfügung: mit `\lhead{}`, `\chead{}`, `\rhead{}`, `\lfoot{}`, `\cfoot{}`, `\rfoot{}` können an den entsprechenden Stellen beliebige Inhalte gesetzt werden, `\lhead{}` zum Beispiel für links oben.

An beliebigen Stellen im Quelltext können diese Inhalte neu definiert werden, so kann zB an einer gewünschten Stelle im Dokument der Kopf verändert werden.

Vorsicht bei doppelseitigem Layout

Obige Befehle setzen den Kolumnentitel auf allen Seiten gleich. Um den Kolumnentitel in Abhängigkeit der aktuellen Seite zu ändern gibt es die Befehle

```
\fancyhead[Optionen]{Inhalt} und
\fancyfoot[Optionen]{Inhalt}
```

Als *Optionen* kann eine beliebige (sinnvolle) Kombination gewählt werden aus E (Even=gerade Seiten), O (Odd=ungerade Seiten), L (Left=Links), C (Center=Mitte) und R (Right=Rechts).

Als *Inhalt* können beliebige Formatierungsbefehle geschrieben werden, sowie die Textmarken `\leftmark` (die aktuell übergeordnete Überschrift, also `chapter` bei `book`) und `\rightmark` (die aktuell untergeordnete Überschrift, also `section` bei `book`).

Als Schriftart ist für diese Marken spezielle Kapitälchen (`\MakeUppercase`) eingestellt. Möchte man dies nicht, so muss die gewünschte Marke (in der Dokumentenpräambel) umdefiniert werden. Hier ein Beispiel, wie die Schriftart der `section`-Marke kursiv gesetzt wird.

```
\renewcommand{\chaptermark}[1]{\markleft{\itshape \thechapter\ #1}}
```

Analog kann auch `\sectionmark` verändert werden.

13.4 hyperref – Hyperlinks in PDFs

Mit dem Paket *hyperref* werden Hyperlinks in pdf-Dateien aktiviert (Inhaltsverzeichnis, Verweise, Websites, usw). Einen Hyperlink auf eine Internetseite kann dann mit `\url{www...}` erzeugt werden.

13.5 nextpage – Leerseiten automatisch berechnen

Manchmal ist es notwendig, das ein Text auf einer ungeraden Seite beginnt. Das Paket *nextpage* ermöglicht mit dem Befehl `\cleartooddpage[\thispagestyle{empty}]` das Einfügen einer eventuell nötigen Leerseite, so das alles, was nach diesem Befehl folgt auf der nächstmöglichen ungeraden Seite beginnt. Analog gibt es `\cleartoevenpage` für gerade Seiten.

13.6 Mit der Kommandozeile arbeiten

Um mit der Kommandozeile zu arbeiten, muss zunächst in das Verzeichnis navigiert werden, wo sich die Quelldatei befindet (Befehl: `cd Pfad des Arbeitsverzeichnisses`).

Eine dvi-datei erzeugt man mit dem Befehl `latex dateiname`.

Eine pdf-datei erzeugt man mit dem Befehl `pdflatex dateiname`.

Eine ps-datei erzeugt man nach der DVI-Datei mit dem Befehl `dvi2ps dateiname`.

Um von PS nach PDF umzuwandeln gibt es von Ghostscript den Befehl `ps2pdf <dateiname.ps>`

Standardmäßig wandelt `ps2pdf` im amerikanischen letter-Format um. Gegebenfalls muss das Papierformat angepasst werden: `ps2pdf -sPAPERSIZE#a4 dateiname.ps wunschname.pdf`

Unter Linux muss dabei das `#` durch `=` ersetzt werden.

Werden Literaturverzeichnisse mit BibTeX gesetzt, so geschieht die Bearbeitung des Dokuments einschließlich Erzeugen des Literaturverzeichnisses mit den folgenden Schritten (Befehle auf der Kommandozeile):

```
latex datei
bibtex datei
latex datei
(latex datei)
```

Die ersten beiden Schritte sind nur notwendig, wenn neue Literaturzitate aufgenommen wurden oder die Literaturdatenbank geändert wurde. Am Ende sollte L^AT_EX wie üblich ein zweites Mal aufgerufen werden, um alle Fußnoten und Querverweise richtig aufzulösen.

Literatur

- [1995 1994] MICHEL GOOSENS, Frank Mittelbach und Alexander S.: *Der L^AT_EX-Begleiter*. Bonn : Addison-Wesley Longman (Deutschland), 1994. – (Deutsche Übersetzung von Michel Goosens (1994))
- [Cameron 1998] CAMERON, Debra: *GNU Emacs kurz & gut*. O'Reilly, 1998. – Eine hervorragende Kurzreferenz, auch für Anfänger geeignet
- [Dalheimer 1998] DALHEIMER, Matthias K.: *L^AT_EX kurz & gut*. O'Reilly, 1998. – (Brauchbare Kurzreferenz für Anfänger)
- [Kopka 1996] KOPKA, Helmut: *L^AT_EX-Erweiterungen*. Addison Wesley, 1996
- [Kopka 2000] KOPKA, Helmut: *L^AT_EX-Einführung*. 3. erweiterte. Addison-Wesley, 2000
- [Kopka 2002] KOPKA, Helmut: *L^AT_EX-Ergänzungen – mit einer Einführung in Metafont*. 3. überarbeitete. Addison Wesley, 2002
- [Lamport 1994] LAMPORT, Leslie: *L^AT_EX – A Document Preparation System*. 2. Reading MA : Addison Wesley Longman Inc., 1994
- [Lamport 1995] LAMPORT, Leslie: *Das L^AT_EX-Handbuch*. Bonn : Addison Wesley Longman (Deutschland), 1995. – (Deutsche Übersetzung von Lamport (1994) mit einer Ergänzung über `german.sty`)
- [Michel Goosens 1994] MICHEL GOOSENS, Frank Mittelbach und Alexander S.: *The L^AT_EX Companion*. Reading MA : Addison-Wesley Longman Inc., 1994
- [Zilm 2003] ZILM, Thorsten: *Das Einsteigeseminar L^AT_EX*. moderne industrie Buch AG & Co., 2003. – (eine hervorragende Kurzreferenz für Anfänger)

Informationen im Netz

CTAN: the Comprehensive TeX Archive Network

<http://www.ctan.org/>
<http://dante.ctan.org/>

CTAN: englische Beschreibung aller Makropakete, alphabetisch sortiert:
<http://www.ctan.org/tex-archive/help/Catalogue/alpha.html>

BibTeX: deutsche DIN-Normen: www.haw-hamburg.de/pers/Lorenzen/bibtex/

KOMA Skript Guide <ftp://ftp.dante.de/tex-archive/macros/latex/contrib/koma-script/scrguide.pdf>

Dieses Skript Online: <http://www.zdv.uni-tuebingen.de/service/anleitungen>

DANTE (Deutschsprachige Anwendervereinigung T_EX e. V.): <http://www.dante.de>

Newsgroups: `comp.text.tex` (international)
`de.comp.text.tex` (deutsch)

The TeX Catalogue Online (Sehr! umfangreich): L^AT_EX-Pakete, Informationen, Software, etc.:
<ftp://ftp.dante.de/tex-archive/help/Catalogue/catalogue.html>

L^AT_EX symbols Comprehensive List (75 Seiten!)
<ftp://ftp.dante.de/tex-archive/help/Catalogue/entries/symbols.html>

MiKTeX <http://www.miktex.org/>

PSTricks (zeichnen und plotten mit L^AT_EX) <http://tug.org/pstricks/main.cgi?file=docs>

ImageMagick (Graphikformate konvertieren) <http://www.imagemagick.org/>

Meine Kontaktadresse felix@widrig.eu

Einführung in \LaTeX
– Beispiele –
(Quelltexte und Ausgabe)

Felix Widrig

2008


```
\documentclass{article}
```

```
\begin{document}
```

```
Hallo Welt!
```

```
\end{document}
```

Hallo Welt!

```

\documentclass{article}
\usepackage[german]{babel}
\usepackage[latin1]{inputenc}

\begin{document}

Zum Glück beherrscht \LaTeX\ auch nationale Sonderzeichen, wie die deutschen
Umlaute \"a \"o \"u \"A \"O \"U oder das scharfe \ss.

\smallskip
Es geht aber auch einfacher: "a "o "u "A "O "U "s.

Andere Variante: \"a "a \"A "A. \\

Probieren wir doch mal \"p und "p. \\

Nett! Weitere interessante Konstruktionen sind die deutschen
Anführungszeichen "‘oben und unten’".

\smallskip
Mit einer deutschen Tastatur kann man auch ä ö ü Ä Ö Ü ß benutzen.

\smallskip
In ähnlicher Weise konstruiert man andere nationale Sonderzeichen:\\

\'a \'a \^a \~a \=a \.a \u{a} \v{a} \H{a} \t{aa} \d{a} \b{a}
\c{c} \aa \AA \ae \AE \o \O \oe \OE \\

\end{document}

```

Zum Glück beherrscht \LaTeX auch nationale Sonderzeichen, wie die deutschen Umlaute ä ö ü Ä Ö Ü oder das scharfe ß.

Es geht aber auch einfacher: ä ö ü Ä Ö Ü ß.
Andere Variante: ä ä Ä Ä.

Probieren wir doch mal p̈ und p.

Nett! Weitere interessante Konstruktionen sind die deutschen Anführungszeichen „oben und unten“.

Mit einer deutschen Tastatur kann man auch ä ö ü Ä Ö Ü ß benutzen.

In ähnlicher Weise konstruiert man andere nationale Sonderzeichen:

à á â ã ä å ä å ä å ä å ç ä Å æ Æ ö œ

```
\documentclass{article}
```

```
\begin{document}
```

Dieses Beispiel soll uns die Auswirkungen von Leerzeichen im Quelltext demonstrieren. Dies steht im Quelltext zwar

weiter hinten, in der Ausgabe ist das aber nicht zu erkennen.

Wollen wir einen frei einstellbaren horizontalen Abstand zwischen Zeichen oder Worten innerhalb der laufenden Zeile, so erreichen wir dies mit einem `\hspace{2cm}` geeigneten Kommando.

Eine Leerzeile erzeugt einen neuen Absatz, mehrere Leerzeilen ebenso.

Ein Zeilenumbruch kann auch im laufenden Absatz `\` erzwungen werden.

Dabei kann man, wenn man will, auch gleich einen beliebigen vertikalen Abstand `\\[3.mm]` einbauen. Eine (oder mehrere) Leerzeile(n), die unmittelbar nach einem Zeilenumbruch folgt`\`

erzeugt einen neuen Absatz, über dem automatisch etwas Abstand gesetzt wird. Es gibt aber auch einen passenden Befehl, mit dem beliebiger vertikaler Abstand zwischen Textteilen erzeugt werden kann:

```
\vspace{2cm}
```

Und hier geht's weiter! Vertikaler Abstand in drei festen Stufen kann auch mit eigens bereit gestellten

```
\smallskip
```

Kurzkommandos

```
\medskip erzeugt
```

```
\bigskip werden.
```

```
% Auch Kommentare kann man einbauen: Der gesamte Text zwischen  
% einem Prozentzeichen % und dem Ende der Zeile bleibt beim  
% Latex-Lauf unbeachtet.
```

```
Hier folgt ein Kommentar: % Dies ist ein Kommentar, beim Latex-Lauf unbeachtet  
%
```

Auch Leerzeilen kann man auskommentieren, wenn ihre Wirkung, das Beenden des laufenden Absatzes, nicht gewollt oder an der fraglichen Stelle nicht erlaubt ist.

```
\end{document}
```

Dieses Beispiel soll uns die Auswirkungen von Leerzeichen im Quelltext demonstrieren. Dies steht im Quelltext zwar weiter hinten, in der Ausgabe ist das aber nicht zu erkennen. Wollen wir einen frei einstellbaren horizontalen Abstand zwischen Zeichen oder Worten innerhalb der laufenden Zeile, so erreichen wir dies mit einem geeigneten Kommando.

Eine Leerzeile erzeugt einen neuen Absatz, mehrere Leerzeilen ebenso.

Ein Zeilenumbruch kann auch im laufenden Absatz erzwungen werden. Dabei kann man, wenn man will, auch gleich einen beliebigen vertikalen Abstand

einbauen. Eine (oder mehrere) Leerzeile(n), die unmittelbar nach einem Zeilenumbruch folgt

erzeugt einen neuen Absatz, bei dem automatisch etwas Abstand gesetzt wird. Es gibt aber auch einen passenden Befehl, mit dem beliebiger vertikaler Abstand zwischen Textteilen erzeugt werden kann:

Und hier geht's weiter! Vertikaler Abstand in drei festen Stufen kann auch mit eigens bereit gestellten

Kurzkommandos

erzeugt

werden.

Hier folgt ein Kommentar: Auch Leerzeilen kann man auskommentieren, wenn ihre Wirkung, das Beenden des laufenden Absatzes, nicht gewollt oder an der fraglichen Stelle nicht erlaubt ist.

```
\documentclass[abstracton]{scrartcl}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm

\pagestyle{headings}
\pagenumbering{Roman}

\begin{document}

\titlehead{\sc Fachhochschule für die ästhetik des Textsatzes}
\subject{Typ der Arbeit}
\title{Der Titel des Dokumentes}
\author{Maxine Musterfrau \and Max Musterman}
\date{22.2.2001}

\maketitle

\begin{abstract} \noindent
  Dieses Buch wendet sich an \LaTeX-Anwender (Einsteiger und
  Fortgeschrittene), die \LaTeX~effizient nutzen und erweitern wollen.
  Viele grundlegende Befehle werden erklärt.
\end{abstract}

\section{Ein Kapitel jagt das andere}

Es war einmal ein kleines Kapitel, das ich so sehr nach Unterkapiteln
sehnte\dots

\subsection{Eine Stufe tiefer}

\subsection{Auf gleicher Höhe}

\newpage
```

Typ der Arbeit

Der Titel des Dokumentes

Maxine Musterfrau Max Musterman

22.2.2001

Zusammenfassung

Dieses Buch wendet sich an \LaTeX -Anwender (Einsteiger und Fortgeschrittene), die \LaTeX effizient nutzen und erweitern wollen. Viele grundlegende Befehle werden erklärt.

1. Ein Kapitel jagt das andere

Es war einmal ein kleines Kapitel, das ich so sehr nach Unterkapiteln sehnte...

1.1. Eine Stufe tiefer

1.2. Auf gleicher Höhe

```
\section{Dies ist die Überschrift des zweiten Kapitels}
```

Jetzt fällt mir langsam nichts mehr ein, was ich schreiben kann. Daher ist dieser interessante Artikel hier zu Ende.

```
\appendix
```

```
\section{Der erste Anhang}
```

Aber einen Anhang wollen wir doch gerne noch haben!

```
\section{Noch ein Anhang}
```

Allmählich reicht's aber...

```
\tableofcontents
```

```
\end{document}
```

2. Dies ist die Überschrift des zweiten Kapitels

2. Dies ist die Überschrift des zweiten Kapitels

Jetzt fällt mir langsam nichts mehr ein, was ich schreiben kann. Daher ist dieser interessante Artikel hier zu Ende.

A. Der erste Anhang

Aber einen Anhang wollen wir doch gerne noch haben!

B. Noch ein Anhang

Allmählich reicht's aber...

Inhaltsverzeichnis

1. Ein Kapitel jagt das andere	I
1.1. Eine Stufe tiefer	I
1.2. Auf gleicher Höhe	I
2. Dies ist die Überschrift des zweiten Kapitels	II
A. Der erste Anhang	II
B. Noch ein Anhang	II

```

\documentclass{article}
\usepackage[ngerman,french,american,english]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}

  \title{Der Titel des Dokumentes}
  \author{Maxine Musterfrau \and Max Musterman}
  \date{22.2.2001}
  \maketitle

\begin{abstract}
  Hier also nun eine Zusammenfassung, betitelt in Englisch.
  Gleichzeitig werden amerikanische Trennungsregeln verwendet.\\

  J. Conrad (Heart of Darkness)\hspace{7mm}\emph{As an outsider, Marlow
  finds himself drawn to the charismatic M Kurtz, but after witnessing
  the malevolence and tyranny that emanate from this man, Marlow is
  forced to reassess his values.}

\end{abstract}

\selectlanguage{ngerman}

\begin{abstract}
  Hier also nun eine Zusammenfassung, betitelt in Deutsch.
  Gleichzeitig werden deutsche Trennungsregeln verwendet.\smallskip

  \emph{Dieses Buch wendet sich an \LaTeX-Anwender (Einsteiger und
  Fortgeschrittene), die \LaTeX~effizient nutzen und erweitern wollen.
  Viele grundlegende Befehle werden erklärt.}
\end{abstract}

\selectlanguage{french}

\begin{abstract}
  Hier also nun eine Zusammenfassung, betitelt in Französisch.
  Gleichzeitig werden französische Trennungsregeln
  verwendet.\smallskip

  \emph{A travers son personnage de des Esseintes, Huysmans n'a pas
  seulement résumé, immortalisé les torpeurs, les langueurs,
  les névroses extraordinaires et perverses du siècle finissant.}

\end{abstract}

\end{document}

```

Der Titel des Dokumentes

Maxine Musterfrau Max Musterman

22.2.2001

Abstract

Hier also nun eine Zusammenfassung, betitelt in Englisch. Gleichzeitig werden amerikanische Trennungsregeln verwendet.

J. Conrad (Heart of Darkness) *As an outsider, Marlow finds himself drawn to the charismatic M Kurtz, but after witnessing the malevolence and tyranny that emanate from this man, Marlow is forced to reassess his values.*

Zusammenfassung

Hier also nun eine Zusammenfassung, betitelt in Deutsch. Gleichzeitig werden deutsche Trennungsregeln verwendet.

Dieses Buch wendet sich an \LaTeX -Anwender (Einsteiger und Fortgeschrittene), die \LaTeX effizient nutzen und erweitern wollen. Viele grundlegende Befehle werden erklärt.

Résumé

Hier also nun eine Zusammenfassung, betitelt in Französisch. Gleichzeitig werden französische Trennungsregeln verwendet.

A travers son personnage de des Esseintes, Huysmans n'a pas seulement résumé, immortalisé les torpeurs, les langueurs, les névroses extraordinaires et perverses du siècle finissant.

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc} \parindent=0mm

\begin{document}

\bfseries Die verschiedenen Schriftarten\ \mdseries

In \LaTeX{} steht uns eine Fülle von verschiedenen Schrifttypen und -größen
zur Verfügung, zunächst die Standardschrift:\

\rmfamily Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\itshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\slshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\scshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\ \upshape

Das ganze nochmal in Fett: \bfseries \
\rmfamily Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\itshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\slshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\scshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\

\upshape \mdseries Dann die Schreibmaschinenschrift: \

\ttfamily Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\itshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\slshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\scshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\ \upshape

Das ganze nochmal in Fett: \bfseries \
\rmfamily Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\itshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\slshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\scshape Fette Kapitälchen sehen immer gleich aus!!!!\

\upshape \mdseries Dann die serifenlose Schrift:\

\sfamily Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\itshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\slshape serifenlos sind kursiv und geneigt unter der T1-Kodierung das gleiche!!!!\
\scshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\ \upshape

Das ganze nochmal in Fett: \bfseries \
\rmfamily Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\itshape Ein Text, der lang genug ist, damit man die Schriftart erkennt\
\slshape Aber Fett gibts einen Unterschied zwischen kursiv und geneigt!!!!\
\scshape Fette Kapitälchen sehen immer gleich aus!!!!\

\end{document}

```

Die verschiedenen Schriftarten

In \LaTeX steht uns eine Fülle von verschiedenen Schrifttypen und -größen zur Verfügung, zunächst die Standardschrift:

Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
EIN TEXT, DER LANG GENUG IST, DAMIT MAN DIE SCHRIFTART ERKENNT

Das ganze nochmal in Fett:

Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
EIN TEXT, DER LANG GENUG IST, DAMIT MAN DIE SCHRIFTART ERKENNT

Dann die Schreibmaschinenschrift:

Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
EIN TEXT, DER LANG GENUG IST, DAMIT MAN DIE SCHRIFTART ERKENNT

Das ganze nochmal in Fett:

Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
FETTE KAPITÄLCHEN SEHEN IMMER GLEICH AUS!!!!

Dann die serifenlose Schrift:

Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
serifenlos sind kursiv und geneigt unter der T1-Kodierung das gleiche!!!!
EIN TEXT, DER LANG GENUG IST, DAMIT MAN DIE SCHRIFTART ERKENNT

Das ganze nochmal in Fett:

Ein Text, der lang genug ist, damit man die Schriftart erkennt
Ein Text, der lang genug ist, damit man die Schriftart erkennt
Aber Fett gibts einen Unterschied zwischen kursiv und geneigt!!!!
FETTE KAPITÄLCHEN SEHEN IMMER GLEICH AUS!!!!

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\parindent=0mm

\begin{document}

\textbf{Die verschiedenen Schriftarten, verschachtelt und in Gruppen} \\

Wenn
\textsf{wir
  \textit{uns}      \textbf{diesen}}
  \textsl{Text}    \textmd{anschauen,}}
  \textsc{so verstehen} \textbf{wir die}}
  \textup{Empfehlung,} \textmd{in einem}}
}
\texttt{Dokument
  \textit{nicht}      \textbf{mehr als}}
  \textsl{zwei}      \textmd{oder}}
  \textsc{maximal}   \textbf{drei}}
  \textup{verschiedene} \textmd{Schriftarten}}
}
\textrm{zu
  \textit{verwenden,} \textbf{auch wenn}}
  \textsl{die Versuchung} \textmd{sehr groß ist,}}
  \textsc{so richtig} \textbf{aus dem}}
  \textup{Vollen} \textmd{zu schöpfen.}}
} \\

Als {\bf Abkürzung} {\it kann man} {\sl auch alte}
{\sc Definitionen} {\tt für einige} {\sf feste Schrifttypen}
verwenden. \\

In einer Gruppe {\itshape \bfseries kann man Schriftattribute ändern,}
die außerhalb dieser Gruppe nicht mehr gelten.\\[2cm]

\textbf{\Large Verschiedene Schriftgrößen}

Entsprechendes \large gilt \Large für die \LARGE Schriftgrößen.
\huge Überschriften \Huge werden \huge meist \LARGE größer \Large gesetzt,
\large Indizes \normalsize und andere \small Bestandteile \footnotesize
mathematischer \scriptsize Formeln \tiny kleiner. \scriptsize Davon
\footnotesize abgesehen, \small sollte \normalsize man beim \large Einsatz
\Large unterschiedlicher \LARGE Schriftgrößen \huge sehr \Huge behutsam
\huge vorgehen.

\end{document}

```

Die verschiedenen Schriftarten, verschachtelt und in Gruppen

Wenn wir *uns diesen Text anschauen*, SO VERSTEHEN **wir die** Empfehlung, in einem Dokument *nicht mehr als zwei oder MAXIMAL drei verschiedene Schriftarten* zu verwenden, **auch wenn** die Versuchung sehr groß ist, SO RICHTIG AUS DEM Vollen zu schöpfen.

Als **Abkürzung** kann man auch alte DEFINITIONEN für einige feste Schrifttypen verwenden.

In einer Gruppe **kann man Schriftattribute ändern**, die außerhalb dieser Gruppe nicht mehr gelten.

Verschiedene Schriftgrößen

Entsprechendes gilt für die Schriftgrößen. Überschriften werden meist größer gesetzt, Indizes und andere Bestandteile mathematischer Formeln kleiner. Davon abgesehen, sollte man beim Einsatz unterschiedlicher Schriftgrößen sehr behutsam vorgehen.

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}

\begin{document}
\section{Die \emph{center}-Umgebung}

\begin{center}

In einer Umgebung gelten andere Regeln: Hier wird beispielsweise jede Zeile
zentriert ausgegeben, getrennt wird nicht, auch nicht bei solchen
Spezialriesenwortungetümskonstruktionen.\\ \bigskip

Meist\\will man aber sowieso\\kurze Zeilen zentriert ausgeben.

\end{center}

\section{Noch 'ne Umgebung: \emph{quote}}
Auch zum Zitieren von Textpassagen steht eine Umgebung bereit. So erklärt
etwa Helmut Kopka:

\begin{quote}
Mit \LaTeX{} wird der Autor und/oder seine Sekretärin in die Lage
versetzt, dem Verlag eine druckfertige Vorlage zu liefern. Damit werden die
Kosten für wissenschaftliche Veröffentlichungen der bekanntesten Journale
teilweise drastisch gesenkt, da dem Verlag die gesamten Satz- und
Korrekturaufwendungen erspart bleiben. Diese Kostenminderung führt bei
etlichen Verlagen zu einem erhöhten Autorenhonorar, was für Berufsautoren
ein nicht unerheblicher Anreiz zur Verwendung von \LaTeX{} sein mag.
\end{quote}

\section{Bündig, bündig}

\begin{flushleft}
Linksbündigen Flattersatz kennen wir von der Schreibmaschine: Alle
Zeilen fangen links an der gleichen Stelle an, haben aber, je nach
Möglichkeiten für einen Zeilenumbruch, geringfügig verschiedene
Länge. Der rechte Rand wirkt dadurch gewissermaßen ausgefranst.
\end{flushleft}

\begin{flushright}
Dagegen ist der rechtsbündige Flattersatz bei der Schreibmaschine
unbekannt. \LaTeX{} beherrscht aber auch diese Variante! Das sieht zwar
ein wenig ungewohnt aus, aber wer weiß, wozu es gut sein kann...
\end{flushright}

\end{document}

```

1 Die *center*-Umgebung

In einer Umgebung gelten andere Regeln: Hier wird beispielsweise jede Zeile zentriert ausgegeben, getrennt wird nicht, auch nicht bei solchen Spezialriesenwortungetümskonstruktionen.

Meist
will man aber sowieso
kurze Zeilen zentriert ausgeben.

2 Noch 'ne Umgebung: *quote*

Auch zum Zitieren von Textpassagen steht eine Umgebung bereit. So erklärt etwa Helmut Kopka:

Mit \LaTeX wird der Autor und/oder seine Sekretärin in die Lage versetzt, dem Verlag eine druckfertige Vorlage zu liefern. Damit werden die Kosten für wissenschaftliche Veröffentlichungen der bekanntesten Journale teilweise drastisch gesenkt, da dem Verlag die gesamten Satz- und Korrekturaufwendungen erspart bleiben. Diese Kostenminderung führt bei etlichen Verlagen zu einem erhöhten Autorenhonorar, was für Berufsautoren ein nicht unerheblicher Anreiz zur Verwendung von \LaTeX sein mag.

3 Bündig, bündig

Linksbündigen Flattersatz kennen wir von der Schreibmaschine: Alle Zeilen fangen links an der gleichen Stelle an, haben aber, je nach Möglichkeiten für einen Zeilenumbruch, geringfügig verschiedene Länge. Der rechte Rand wirkt dadurch gewissermaßen ausgefranst.

Dagegen ist der rechtsbündige Flattersatz bei der Schreibmaschine unbekannt. \LaTeX beherrscht aber auch diese Variante! Das sieht zwar ein wenig ungewohnt aus, aber wer weiß, wozu es gut sein kann...

```
\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc} \parindent=0mm
```

```
\begin{document}
\section*{Tabulatoren}
```

Auch Tabulatoren werden in \LaTeX durch eine Umgebung realisiert:

```
\begin{tabbing}
Auf \=der \=Mauer, \=auf \=der \=Lauer\\
\>der Mauer, auf der Lauer\\
\>\>Mauer, auf der Lauer\\
\>\>\>auf der Lauer\\
\>\>\>der Lauer\\
\>\>\>\>Lauer\\
\end{tabbing}
```

Nun das gleiche nochmal, nur definieren wir die Tabstops nicht alle in der ersten Zeile, sondern schrittweise in jeder Zeile:

```
\begin{tabbing}
Auf \=der Mauer, auf der Lauer\\
\>der \=Mauer, auf der Lauer\\
\>\>Mauer, \=auf der Lauer\\
\>\>\>auf \=der Lauer\\
\>\>\>der \=Lauer\\
\>\>\>\>Lauer\\
\end{tabbing}
```

```
\section*{Aufgepasst!!}
```

\LaTeX 's Tabulatoren verhalten sich ein wenig anders als die auf der guten alten Schreibmaschine:

```
\begin{tabbing}
Auf der Mauer, \=auf der Lauer\\
Auf der hohen Mauer, \>auf der Lauer\\
\end{tabbing}
```

```
\section*{Tabstops in einer Musterzeile}
```

Die Tabulatorstops können auch in einer Musterzeile angegeben werden, die nicht ausgegeben wird:

```
\begin{tabbing}
Auf \=der \=Mauer, \=auf \=der \=Lauer\kill
Auf \>\>\> auf \\
\>der \>\>\> der \\
\>\>Mauer \>\>\> Lauer \\
\end{tabbing}
```

```
\end{document}
```

Tabulatoren

Auch Tabulatoren werden in \LaTeX durch eine Umgebung realisiert:

```
Auf der Mauer, auf der Lauer
      der Mauer, auf der Lauer
            Mauer, auf der Lauer
                  auf der Lauer
                        der Lauer
                              Lauer
```

Nun das gleiche nochmal, nur definieren wir die Tabstops nicht alle in der ersten Zeile, sondern schrittweise in jeder Zeile:

```
Auf der Mauer, auf der Lauer
      der Mauer, auf der Lauer
            Mauer, auf der Lauer
                  auf der Lauer
                        der Lauer
                              Lauer
```

Aufgepasst!!

\LaTeX 's Tabulatoren verhalten sich ein wenig anders als die auf der guten alten Schreibmaschine:

```
Auf der Mauer, auf der Lauer
Auf der hohen Mauer, auf der Lauer
```

Tabstops in einer Musterzeile

Die Tabulatorstops können auch in einer Musterzeile angegeben werden, die nicht ausgegeben wird:

```
Auf          auf
  der        der
            Mauer      Lauer
```

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}

\begin{document}
\section{Listen, Listen, nichts als Listen}

Listen sind eine grundlegende Gliederungsmöglichkeit eines Textes.
\LaTeX\ stellt dazu einige vordefinierte Umgebungen zur Verfügung.

\subsection{Einfache Listen}
Lecker, lecker:

\begin{itemize}
\item Spätzle
\item Maultaschen
\item Kartoffelsalat
\end{itemize}

\subsection{Nummerierte Listen}
Diesmal verschachteln wir Sie sogar:

\begin{enumerate}
\item Zwiebel in kleine Würfel schneiden
\begin{enumerate}
\item Haut abschälen
\item Scharfes Messer verwenden und schneiden
\end{enumerate}
\item Andünsten
\begin{itemize}
\item Mit Butter oder Öl
\item Nicht zu heiss!!
\end{itemize}
\item Mit Olivenstückchen abrunden.
\end{enumerate}

\subsection{Beschreibende Listen}

\begin{description}
\item[Schachtelungen:] Listen können in andere Listen eingebaut werden.
\item[Verschachtelungstiefe:] Es können vier Ebenen einer Listenart
geschachtelt werden; bei verschiedenen Listenarten insgesamt
maximal sechs Ebenen.
\item[Zähler:] Die nummerierten Listen haben eigene Zähler:
\emph{enumi}, \emph{enumii}, \emph{enumiii} und \emph{enumiv}.
\item[Freie Gestaltung:] Es existiert eine \emph{list}-Umgebung, die eine
freie Formatierung von Listen zulässt.
\end{description}

\end{document}

```

1 Listen, Listen, nichts als Listen

Listen sind eine grundlegende Gliederungsmöglichkeit eines Textes. \LaTeX stellt dazu einige vordefinierte Umgebungen zur Verfügung.

1.1 Einfache Listen

Lecker, lecker:

- Spätzle
- Maultaschen
- Kartoffelsalat

1.2 Nummerierte Listen

Diesmal verschachteln wir Sie sogar:

1. Zwiebel in kleine Würfel schneiden
 - (a) Haut abschälen
 - (b) Scharfes Messer verwenden und schneiden
2. Andünsten
 - Mit Butter oder Öl
 - Nicht zu heiss!!
3. Mit Olivenstückchen abrunden.

1.3 Beschreibende Listen

Schachtelungen: Listen können in andere Listen eingebaut werden.

Verschachtelungstiefe: Es können vier Ebenen einer Listenart geschachtelt werden; bei verschiedenen Listenarten insgesamt maximal sechs Ebenen.

Zähler: Die nummerierten Listen haben eigene Zähler: *enumi*, *enumii*, *enumiii* und *enumiv*.

Freie Gestaltung: Es existiert eine *list*-Umgebung, die eine freie Formatierung von Listen zulässt.

```
\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm
```

```
\begin{document}
\section*{Fußnoten und Randbemerkungen}
```

Dies also nun ein kleiner Text, um die Fußnotengenerierung`\footnote{Was für ein Wort!!!}` zu veranschaulichen. Geht doch ganz einfach!

Eine `\emph{minipage}`-Umgebung bekommt ihre eigenen Fußnoten:

```
\begin{center}
\begin{minipage}[t]{5cm}
  Dies also eine kleine Minipage, die nach außen hin auch wieder eine
  Box\footnote{alles in \LaTeX\ ist eine Box} darstellt.
\end{minipage}
\end{center}
```

Und nun setzen wir eine Randbemerkung, die uns ganz besonders auf diese Textstelle hinweist.`\marginpar{\Huge !}` Damit können wir sicher sein, dass sie nicht übersehen wird.

```
\end{document}
```

Fußnoten und Randbemerkungen

Dies also nun ein kleiner Text, um die Fußnotengenerierung¹ zu veranschaulichen. Geht doch ganz einfach!

Eine `minipage`-Umgebung bekommt ihre eigenen Fußnoten:

Dies also eine kleine Minipage,
die nach außen hin auch wieder
eine Box^a darstellt.

^aalles in `LATEX` ist eine Box

Und nun setzen wir eine Randbemerkung, die uns ganz besonders auf diese Textstelle hinweist. Damit können wir sicher sein, dass sie nicht übersehen wird. !

¹Was für ein Wort!!!

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc} \parindent=0mm

\begin{document}
\section{Boxen mit Rand}
\subsection{Eine einfache Box}
Packen wir doch mal ein Stück Text \fbox{in eine Box}.
Eine solche Box \fbox{steht immer auf
einer Zeile} und wird nicht umgebrochen.

\subsection{Box mit beliebiger Breite}
Packen wir diesmal \framebox[5cm][c]{ein wenig Text} in eine Box, die 5cm breit
ist.

\subsection{Dick und dünn}
\fboxrule=2.pt
Die \fbox{Liniendicke} für die Umrandung lässt sich einstellen, aber damit
sollte man behutsam umgehen --- sonst erinnert das Ganze schnell
\framebox[5cm][l]{an eine Traueranzeige\dots}

\fboxrule=0.4pt

\fboxsep=5.mm
Oder wie wär's mit \fbox{etwas mehr Platz} zwischen Text und Umrandung?
\fboxsep=0.mm
Oder zur Abwechslung \fbox{mal ganz eng}!
\fboxsep=-1.pt
Das ist jetzt aber \fbox{entschieden zuviel des Guten}!
\fboxsep=4.pt

\section{Balkenbox}
Eine Box, die komplett mit Druckerschwärze gefüllt ist, ergibt eine
horizontale \rule{1cm}{1mm} oder vertikale \rule{1mm}{1cm} Linie ---
oder auch ein Quadrat \rule{1cm}{1cm} \medskip

Verschieben wir die Box doch mal ein wenig nach oben oder unten:
\rule[-.5ex]{1ex}{1ex}\rule[.5ex]{1ex}{1ex}
\rule[-.5ex]{1ex}{1ex}\rule[.5ex]{1ex}{1ex} \medskip

Eine solche Linie darf auch die Länge oder Höhe Null haben. Sie bleibt
dann zwar unsichtbar, \rule{2cm}{0mm} hat aber trotzdem eine Wirkung,
wie man hier sehr deutlich \rule{0mm}{1cm} sehen kann.\medskip

Eine unsichtbare vertikale Linie kann als Stütze verwendet werden, um
Textteile vertikal auf genügend Abstand zu halten. Interessant ist dies
beispielsweise beim Einsatz {\Large großer Schrifttypen}, wo der
normalerweise vorhandene zusätzliche Zwischenraum zwischen den Zeilen
nicht für die höheren Buchstaben ausreicht. Die Verwendung einer Stütze
sorgt hier für {\Large\rule[-.5ex]{0mm}{2.9ex}Abhilfe.}
\end{document}

```

1 Boxen mit Rand

1.1 Eine einfache Box

Packen wir doch mal ein Stück Text in eine Box. Eine solche Box steht immer auf einer Zeile und wird nicht umgebrochen.

1.2 Box mit beliebiger Breite

Packen wir diesmal ein wenig Text in eine Box, die 5cm breit ist.

1.3 Dick und dünn

Die Liniendicke für die Umrandung lässt sich einstellen, aber damit sollte man behutsam umgehen — sonst erinnert das Ganze schnell an eine Traueranzeige...

Oder wie wär's mit etwas mehr Platz zwischen Text und Umrandung?

Oder zur Abwechslung mal ganz eng! Das ist jetzt aber entschieden zuviel des Guten!

2 Balkenbox

Eine Box, die komplett mit Druckerschwärze gefüllt ist, ergibt eine horizontale  oder vertikale  Linie — oder auch ein Quadrat 

Verschieben wir die Box doch mal ein wenig nach oben oder unten: 

Eine solche Linie darf auch die Länge oder Höhe Null haben. Sie bleibt dann zwar unsichtbar,  hat aber trotzdem eine Wirkung, wie man hier

sehr deutlich  sehen kann.

Eine unsichtbare vertikale Linie kann als Stütze verwendet werden, um Textteile vertikal auf genügend Abstand zu halten. Interessant ist dies beispielsweise beim Einsatz **großer Schrifttypen**, wo der normalerweise vorhandene zusätzliche Zwischenraum zwischen den Zeilen nicht für die höheren Buchstaben ausreicht. Die Verwendung einer Stütze sorgt hier für **Abhilfe**.

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc} \parindent=0mm

\begin{document}

\section{Mehrzeilige Boxen}

\subsection{Die Parbox}
Wenn wir mehrzeiligen Text in eine Box packen wollen, brauchen wir einen
weiteren Befehl:
\fbbox{\parbox{0.5\textwidth}{Hier der mehrzeilige Text, der die Hälfte der
normalen Breite einnehmen sollte. Mal schau'n ob er dies auch tut.}}
%
Jetzt schreiben wir ganz normal weiter, ohne irgendwelche Boxen --- immer noch
im selben Absatz.\\

Alternativ \fbbox{\parbox[t]{3cm}{Oben ausgerichtet, damit man sieht, was passiert...}}
und wir schreiben hier weiter...

\subsection{Die \emph{minipage}}

\fbbox{\begin{minipage}[t]{0.4\textwidth}
Hier wieder ein wenig Text, der zur Abwechslung mal am unteren Rand
ausgerichtet ist.
\end{minipage}}
Diese Umgebung ist flexibler, sie darf einige Befehle enthalten, die die
einfache Variante nicht zulässt.

\section{Boxen hoch und tief stellen mit \emph{raisebox}}

Man kann auch eine Box konstruieren, die \raisebox{0.8ex}{\fbbox{hoch}} und
\raisebox{-0.8ex}{\fbbox{runter}} geschoben werden kann. Innerhalb einer Box
\raisebox{1.ex}{\fbbox{\parbox[t]{0.1\textwidth}{dürfen weitere Boxen}}}}
enthalten sein.

\end{document}

```

1 Mehrzeilige Boxen

1.1 Die Parbox

Wenn wir mehrzeiligen Text in eine Box packen wollen, brauchen wir einen wei-
 teren Befehl: Hier der mehrzeilige Text, der die Hälfte
 der normalen Breite einnehmen sollte.
 Mal schau'n ob er dies auch tut. Jetzt schreiben wir ganz
 normal weiter, ohne irgendwelche Boxen — immer noch im selben Absatz.

Alternativ Oben ausgerichtet,
 damit man sieht,
 was passiert... und wir schreiben hier weiter...

1.2 Die *minipage*

Hier wieder ein wenig Text, der
 zur Abwechslung mal am unteren
 Rand ausgerichtet ist. Diese Umgebung ist flexibler, sie darf einige
 Befehle enthalten, die die einfache Variante nicht zulässt.

2 Boxen hoch und tief stellen mit *raisebox*

Man kann auch eine Box konstruieren, die hoch und runter geschoben werden
 kann. Innerhalb einer Box dürfen
 weitere
 Boxen enthalten sein.

```
\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\usepackage{graphicx}
\graphicspath{{./graphik/}}
\parindent=0mm

\begin{document}

\section*{Graphiken einbinden}

Bilder, Grafiken, usw, können (man beachte das Dateiformat) einfach in den fließenden Text
\includegraphics[width=2cm]{kampf.jpg} eingebaut werden.\\

Genau wie für Tabellen gibt es aber auch hier eine Umgebung, die eine
Abbildung zum Gleitobjekt macht:

\begin{figure}[ht]
\begin{center}
\includegraphics[width=8cm]{kampf.jpg}
\caption{Ein Bild}
\end{center}
\end{figure}

\listoffigures

\end{document}
```

Graphiken einbinden

Bilder, Grafiken, usw, können (man beachte das Dateiformat) einfach in den



fließenden Text eingebaut werden.

Genau wie für Tabellen gibt es aber auch hier eine Umgebung, die eine Abbildung zum Gleitobjekt macht:



Abbildung 1: Ein Bild

Abbildungsverzeichnis

1	Ein Bild	1
---	----------------	---

```

\documentclass{scrartcl}
\usepackage{ngerman,subfig,caption}
\usepackage[latin1]{inputenc} \parindent=0mm
\usepackage{graphicx}
\graphicspath{{./graphik/}}
\begin{document}

\section{Gleitobjekte}
\subsection{Einfaches Gleitobjekt einbinden}

\begin{figure}[!ht] \centering
  \includegraphics[width=2cm]{kampf.jpg}
  \caption{Das erste Gleitobjekt.}
\end{figure}

\subsection{Abbildung mit mehreren Figuren}

\begin{figure}[!ht] \centering
  \subfloat[so]{
    \fbox{\Large \bf A}
    \label{rechts}
  }
  %
  \qqquad
  %
  \subfloat[oder so]{
    \fbox{\Large \bf BCD}
    \label{links}
  }
  \caption[Buchstaben]{Buchstaben: unter \subref{rechts} das erste
    und unter \subref{links} weitere\dots}
\end{figure}

\subsection{Gleitobjektbezeichner ändern}
Geht so: \renewcommand{\figurename}{Abb.}

\begin{figure}[!ht] \centering
  \fbox{\Large \bf A}
  \caption{Hier steht Abb. statt Abbildung!}
\end{figure}

\section{Abbildungen Linksbündig beschriften}
Geht so: \captionsetup{singlelinecheck=off}

\begin{figure}[!ht]
  \fbox{\Large \bf A}
  \caption{Linksbündig beschriftet!}
\end{figure}

\listoffigures
\end{document}

```

1 Gleitobjekte

1.1 Einfaches Gleitobjekt einbinden



Abbildung 1: Das erste Gleitobjekt.

1.2 Abbildung mit mehreren Figuren



Abbildung 2: Buchstaben: unter (a) das erste und unter (b) weitere...

1.3 Gleitobjektbezeichner ändern

Geht so:



Abb. 3: Hier steht Abb. statt Abbildung!

2 Abbildungen Linksbündig beschriften

Geht so:



Abb. 4: Linksbündig beschriftet!

Abbildungsverzeichnis

1	Das erste Gleitobjekt.	1
2	Buchstaben	1
3	Hier steht Abb. statt Abbildung!	1
4	Linksbündig beschriftet!	1

```

\documentclass{article}
\usepackage{ngerman,array}
\usepackage[latin1]{inputenc}
\parindent=0mm

\begin{document}

\section{Eine Einführung zum Thema Tabellen}

Tabellen sind in \LaTeX\ ein Kinderspiel:
%
\begin{tabular}[t]{ccc}
\bf Eins & 2 & & 3 & \\
1 & & \bf Zwei & & III & \\
I & & II & & \bf Drei & \\
\end{tabular} \\

Ein zweites Beispiel, diesmal mit anderen inneren und äußeren
Positionsparametern und ein paar Linien zur Verschönerung:
%
\begin{tabular}[b]{|l|c|r|}
\hline
\bf Eins & 2 & & 3 & \\
1 & & \bf Zwei & & III & \\
I & & II & & \bf Drei & \\
\hline
\end{tabular}
Wir erkennen hier, dass eine Tabelle ganz normal in den laufenden Text
eingebaut wird.

\end{document}

```

1 Eine Einführung zum Thema Tabellen

Tabellen sind in \LaTeX ein Kinderspiel: **Eins** 2 3
1 **Zwei** III
I II **Drei**

Ein zweites Beispiel, diesmal mit anderen inneren und äußeren Positionspara-

Eins	2	3
1	Zwei	III
I	II	Drei

metern und ein paar Linien zur Verschönerung:

Eins	2	3
1	Zwei	III
I	II	Drei

 Wir erkennen hier, dass eine Tabelle ganz normal in den laufenden Text eingebaut wird.

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}

\begin{document}

\section{Gleitobjekte: Tabellen im Tabellenverzeichnis}
\subsection{Die einfache Variante}
Hier ist eine Tabelle komplett mit Erläuterung, die in ein
Tabellenverzeichnis aufgenommen werden kann:

\begin{table}[!ht]
\centering
\begin{tabular}{|r|c|l|}
\hline
\bf Rang&\bf Verein&\bf Punkte\\
\hline
\hline
1 & 1.FC Haferstroh & 10: 0 \\ \hline
2 & SV Weizenkleie & 8: 5 \\ \hline
3 & Sojasprossen 1860 & 0:10 \\ \hline
\end{tabular}
\caption{Die Bezeichnung der Tabelle}
\end{table}

\subsection{Schöner mit \emph{multicolumn}}

\begin{table}[!ht]

\centering
\begin{tabular}{|r|c|r@{:}l|}
\hline
\bf Rang&\bf Verein&
\multicolumn{2}{c|}{\bf Punkte} \\
\hline \hline
1 & 1.FC Haferstroh & 10 & 0 \\ \hline
2 & SV Weizenkleie & 8 & 5 \\ \hline
3 & Sojasprossen 1860 & 0 & 8 \\ \hline
\end{tabular}
\caption{Der zweite Versuch, eine schöne Tabelle zu setzen}

\end{table}

\listoftables

\end{document}

```

1 Gleitobjekte: Tabellen im Tabellenverzeichnis

1.1 Die einfache Variante

Hier ist eine Tabelle komplett mit Erläuterung, die in ein Tabellenverzeichnis aufgenommen werden kann:

Rang	Verein	Punkte
1	1.FC Haferstroh	10: 0
2	SV Weizenkleie	8: 5
3	Sojasprossen 1860	0:10

Tabelle 1: Die Bezeichnung der Tabelle

1.2 Schöner mit *multicolumn*

Rang	Verein	Punkte
1	1.FC Haferstroh	10:0
2	SV Weizenkleie	8:5
3	Sojasprossen 1860	0:8

Tabelle 2: Der zweite Versuch, eine schöne Tabelle zu setzen

Tabellenverzeichnis

```

\documentclass{scrartcl}
\usepackage{ngerman,array,tabularx,multirow,dcolumn}
\usepackage[latin1]{inputenc} \parindent=0mm

\begin{document}

\section{Tabelle mit array}
Wir möchten die erste Spalte dieser Tabelle in Fettschrift setzen:

\begin{tabular}{|>{\bf}c|c|c|} \hline
1&2&3 \\ \hline
4&5&6 \\ \hline
\end{tabular}

\section{Tabelle mit tabularx}
Wir möchten eine Tabelle, dessen Breite genau die Textbreite ist, wobei die Breite der
2. und 4. Spalte von \LaTeX automatisch berechnet wird: \bigskip

\begin{tabularx}{\textwidth}{|p{3cm}|X|p{3cm}|X|} \hline
Rechts ist Text & 1,2,3,4,5,6,7,8 & Und links ist Text & Hinten dran noch et- \\
\hline
was
\end{tabularx}

\section{Tabelle mit multirow}

\newlength{\mylen} % zuerst definieren wir uns eine neue Länge, die wir auf die
\settowidth{\mylen}{Z} % Breite des Eintrags setzen, und später benutzen.

\begin{tabular}{|c|c|c|} \hline
A & \multirow{2}{\mylen}{Z} & C \\ \hline
D & & F \\ \hline
G & H & I \\ \hline
\end{tabular}

\section{Tabelle mit dcolumn}

Anstatt (wie in Beisp. 10-1) den Abstand für die zweite Zeile der dritten Spalte
manuell zu berechnen, kann man das mit Hilfe von
\emph{dcolumn} machen:\\

\begin{tabular}{|c|c|D.:{2.2}|}
\hline
\bf Rang&\bf Verein & \multicolumn{1}{c|}{\bf Torverhältnis}\\ \hline
1& Fc Bayern München & 10.0\\ \hline
\end{tabular}

\end{document}

```

1 Tabelle mit array

Wir möchten die erste Spalte dieser Tabelle in Fettschrift setzen:

1	2	3
4	5	6

2 Tabelle mit tabularx

Wir möchten eine Tabelle, dessen Breite genau die Textbreite ist, wobei die Breite der
2. und 4. Spalte von \LaTeX automatisch berechnet wird:

Rechts ist Text	1,2,3,4,5,6,7,8	Und links ist Text	Hinten dran noch et- was
-----------------	-----------------	--------------------	-----------------------------

3 Tabelle mit multirow

A	Z	C
D		F
G	H	I

4 Tabelle mit dcolumn

Anstatt (wie in Beisp. 10-1) den Abstand für die zweite Zeile der dritten Spalte manuell
zu berechnen, kann man das mit Hilfe von *dcolumn* machen:

Rang	Verein	Torverhältnis
1	Fc Bayern München	10:0

```

\documentclass{scrartcl}
\usepackage{ngerman,tbls}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc} \parindent=0mm

\begin{document}
\section{Treppentabellen}

\begin{center}
\begin{tabular}{|c|c|c|} \hline
a&b&c \\ \hline
a&b \\ \hline
a \\ \hline
\end{tabular}
%
\qqquad
%
\begin{tabular}{|c|c|c|} \hline
a&b&c \\ \hline
\multicolumn{1}{c|}{&b&c} \\ \hline
\multicolumn{2}{c|}{&c} \\ \hline
\end{tabular}
\end{center}

\section{Tabellen mit \emph{tbls}}
Achtung, das Paket \emph{tbls} kann (derzeit) nicht mit dem Paket \emph{array}
kominiert werden! Hier ein kleines Beispiel, der Zeilenabstand ist größer:

\begin{table}[ht]
\centering
\begin{tabular}{|c|c|c|} \hline
1&2&3 \\ \hline
4&5&6 \\ \hline
\end{tabular}
\end{table}

Versuchen wir noch mehr, wir verkleinern wieder den Abstand über und unter den Linien:
\extrarulesep=1pt \bigskip

\begin{tabular}{|c|c|c|} \hline
1&2&3 \\ \hline
4&5&6 \\ \hline
\end{tabular} \bigskip

Oder auch so: \bigskip

\begin{tabular}{|c|c|c|} \hline[3mm]
1&2&3 \\ \hline[2mm]
4&5&6 \\ \hline
\end{tabular}

\end{document}

```

1 Treppentabellen

a	b	c
a	b	
a		

a	b	c
	b	c
		c

2 Tabellen mit *tbls*

Achtung, das Paket *tbls* kann (derzeit) nicht mit dem Paket *array* kominiert werden!
Hier ein kleines Beispiel, der Zeilenabstand ist größer:

1	2	3
4	5	6

Versuchen wir noch mehr, wir verkleinern wieder den Abstand über und unter den Linien:

1	2	3
4	5	6

Oder auch so:

1	2	3
4	5	6

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\usepackage{array}

\begin{document}

\section{Fußnoten in Tabellen}

\begin{table}[htbp] \centering
\begin{tabular}{|r|c|>{\hspace{2.3em}}r@{:}r<{\hspace{2.3em}}|}
\hline
{\bf Rang}&{\bf Verein}&
\multicolumn{2}{c|}{\bf Torverhältnis}\
\hline \hline
1 & 1.FC Haferstroh & 10 & 0 \\\hline
2 & SV Weizenkleie & 8 & 5 \\\hline
3 & Sojasprossen 1860\footnotemark & 0 & 10 \\\hline
\end{tabular}
\caption{Krönung mit Zugabe}
\end{table}

\footnotetext{Abstiegskandidat, zweiter Anlauf}

\begin{table}[htbp] \centering
\begin{tabular}{|r|c|>{\hspace{2.3em}}r@{:}r<{\hspace{2.3em}}|}
\hline
{\bf Rang}&{\bf Verein}&
\multicolumn{2}{c|}{\bf Torverhältnis}\
\hline \hline
1 & 1.FC Haferstroh\footnotemark & 10 & 0 \\\hline
2 & SV Weizenkleie & 8 & 5 \\\hline
3 & Sojasprossen 1860\footnotemark & 0 & 10 \\\hline
\end{tabular}
\caption{Krönung mit noch mehr Anmerkungen}
\end{table}

\addtocounter{footnote}{-1}
\footnotetext{Tabellenspitze}
\stepcounter{footnote}
\footnotetext{Abstiegskandidat, dritter Versuch}

\end{document}

```

1 Fußnoten in Tabellen

Rang	Verein	Torverhältnis
1	1.FC Haferstroh	10: 0
2	SV Weizenkleie	8: 5
3	Sojasprossen 1860 ¹	0:10

Tabelle 1: Krönung mit Zugabe

Rang	Verein	Torverhältnis
1	1.FC Haferstroh ²	10: 0
2	SV Weizenkleie	8: 5
3	Sojasprossen 1860 ³	0:10

Tabelle 2: Krönung mit noch mehr Anmerkungen

¹Abstiegskandidat, zweiter Anlauf
²Tabellenspitze
³Abstiegskandidat, dritter Versuch

```
\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc} \parindent=0mm
```

```
\begin{document}
\section{Formeln im Text}
```

Formeln werden immer im mathematischen Modus gesetzt. Formeln können im laufenden Text stehen, denn in rechtwinkligen Dreiecken gilt:

```
$ a^2+b^2=c^2 $
```

Dabei werden Formeln auch umgebrochen --- nicht immer zur Freude des Authors. Eine grundlegende Eigenschaft des Skalarproduktes ist die Kommutativität: $\langle \vec{x}, \vec{y} \rangle = \langle \vec{y}, \vec{x} \rangle$.

```
\section{Abgesetzte Formeln}
```

Meist sind Formeln aber zu lang und können nicht gut im laufenden Text untergebracht werden. Dann bekommen sie halt mehr Platz:

```
$$ \langle \vec{x}, \vec{y} \rangle = \langle \vec{y}, \vec{x} \rangle $$
```

für alle Vektoren \vec{x} und \vec{y} .

```
\section{Nummerierte Formeln}
```

```
\begin{equation}
\vec{y} := (y_1 \dots y_n)
\end{equation}
```

```
\section{Mehrzeilige ausgerichtete Formeln}
```

```
\begin{eqnarray}
\langle \vec{x}, \vec{y} \rangle &=& \sum_k x_k y_k \\
&=& \langle \vec{y}, \vec{x} \rangle
\end{eqnarray}
```

Vielleicht wollen wir nicht jede Zeile nummerieren:

```
\begin{eqnarray}
\langle \vec{x}, \vec{y} \rangle &=& \sum_k x_k y_k \nonumber \\
&=& \langle \vec{y}, \vec{x} \rangle
\end{eqnarray}
```

Oder wir lassen die Nummern gleich ganz weg:

```
\begin{eqnarray*}
\langle \vec{x}, \vec{y} \rangle &=& \sum_k x_k y_k \\
&=& \langle \vec{y}, \vec{x} \rangle
\end{eqnarray*}
```

```
\end{document}
```

1 Formeln im Text

Formeln werden immer im mathematischen Modus gesetzt. Formeln können im laufenden Text stehen, denn in rechtwinkligen Dreiecken gilt: $a^2 + b^2 = c^2$ Dabei werden Formeln auch umgebrochen — nicht immer zur Freude des Authors. Eine grundlegende Eigenschaft des Skalarproduktes ist die Kommutativität: $\langle \vec{x}, \vec{y} \rangle = \langle \vec{y}, \vec{x} \rangle$.

2 Abgesetzte Formeln

Meist sind Formeln aber zu lang und können nicht gut im laufenden Text untergebracht werden. Dann bekommen sie halt mehr Platz:

$$\langle \vec{x}, \vec{y} \rangle = \langle \vec{y}, \vec{x} \rangle$$

für alle Vektoren \vec{x} und \vec{y} .

3 Nummerierte Formeln

$$\vec{y} := (y_1 \dots y_n) \quad (1)$$

4 Mehrzeilige ausgerichtete Formeln

$$\begin{aligned} \langle \vec{x}, \vec{y} \rangle &= \sum_k x_k y_k \\ &= \langle \vec{y}, \vec{x} \rangle \end{aligned} \quad (2)$$

$$= \langle \vec{y}, \vec{x} \rangle \quad (3)$$

Vielleicht wollen wir nicht jede Zeile nummerieren:

$$\begin{aligned} \langle \vec{x}, \vec{y} \rangle &= \sum_k x_k y_k \\ &= \langle \vec{y}, \vec{x} \rangle \end{aligned} \quad (4)$$

Oder wir lassen die Nummern gleich ganz weg:

$$\begin{aligned} \langle \vec{x}, \vec{y} \rangle &= \sum_k x_k y_k \\ &= \langle \vec{y}, \vec{x} \rangle \end{aligned}$$

```

\documentclass{article}
\usepackage{ngerman,amsmath}
\usepackage[latin1]{inputenc} \parindent=0mm

\begin{document}

\section{Mathematische Spielereien}

\begin{eqnarray}
&& f(x) = \frac{1}{x^2} \\
&& \int_1^\infty \frac{1}{x^2} = \int_1^\infty \frac{1}{x^2} = 1
\end{eqnarray}

Gelegentlich muss man bei den Abständen ein wenig nachhelfen:

\begin{eqnarray}
\Omega &= & \int \int_{\Gamma} f g dx dy \\
&= & \int \int_{\Gamma} f g dx dy \\
&\approx & \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} f_{mn} g_{mn} \\
A_{ij} &= & e^{-i\omega t} \sqrt{x_i x_j}
\end{eqnarray}

\begin{equation}
A_{\{ij\}} = e^{-i\omega t} \sqrt{x_i x_j}
\end{equation}

Logische Zusammenhänge können mit diversen Pfeilen deutlich gemacht werden:

\begin{equation}
\dot{v} = 0 \Leftrightarrow v = \text{konstant}
\end{equation}

Symbole und ganze Formelteile können mit diversen Aufsätzen und Unterbauten
versehen werden:

\begin{equation}
x = 1 \hookrightarrow \hat{x} = \bar{x} = 0
\end{equation}

\begin{equation}
\widehat{F \circ G} = \widetilde{f \circ g}
\end{equation}

\begin{equation}
S = \overbrace{(x_\alpha + \dots + x_\beta)}^{\text{mindestens } N \text{ Terme}} \times \underbrace{(y^\mu + \dots + y^\nu)}_{\text{höchstens } M \text{ Terme}}
\end{equation}

\end{document}

```

1 Mathematische Spielereien

$$f(x) = \frac{1}{x^2} \quad (1)$$

$$\int_1^\infty \frac{1}{x^2} = \int_1^\infty \frac{1}{x^2} = 1 \quad (2)$$

Gelegentlich muss man bei den Abständen ein wenig nachhelfen:

$$\begin{aligned} \Omega &= \int \int_{\Gamma} f g dx dy \\ &= \int \int_{\Gamma} f g dx dy \\ &\approx \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} f_{mn} g_{mn} \end{aligned} \quad (3)$$

$$A_{ij} = e^{-i\omega t} \sqrt{x_i x_j} \quad (4)$$

Logische Zusammenhänge können mit diversen Pfeilen deutlich gemacht werden:

$$\dot{v} = 0 \Leftrightarrow v = \text{konstant} \quad (5)$$

Symbole und ganze Formelteile können mit diversen Aufsätzen und Unterbauten versehen werden:

$$x = 1 \hookrightarrow \hat{x} = \bar{x} = 0 \quad (6)$$

$$\widehat{F \circ G} = \widetilde{f \circ g} \quad (7)$$

$$S = \overbrace{(x_\alpha + \dots + x_\beta)}^{\text{mindestens } N \text{ Terme}} \times \underbrace{(y^\mu + \dots + y^\nu)}_{\text{höchstens } M \text{ Terme}} \quad (8)$$

```

\documentclass{article}
\usepackage{ngerman,amsmath,amssymb}
\usepackage[latin1]{inputenc} \parindent=0mm

\begin{document}

\section{Drunter und drüber im Mathe-Modus}
Für kompliziertere Gebilde, wie bspw. Vektoren oder Matrizen, gibt es
die \emph{array}-Umgebung, die die gleiche Syntax aufweist wie die
\emph{tabular}-Umgebung, und die ebenfalls eine Tabelle erzeugt.

$$ A = \left( \begin{array}{ccc}
x^1_1 & \dots & x^1_n \\
\vdots & \ddots & \vdots \\
x^n_1 & \dots & x^n_n \end{array} \right) $$

Das kann man aber auch ohne Spaltenangabe mit der \emph{matrix}-Umgebung von amsmath:

$$ \det A = \left| \begin{matrix}
x^1_1 & \dots & x^1_n \\
\vdots & \ddots & \vdots \\
x^n_1 & \dots & x^n_n \end{matrix} \right| \stackrel{!}{=} 0 $$

\section{Mehr Spielereien}

$$ f(x) \underset{x \to \infty}{\longrightarrow} 0 $$

$$ g(x) = e^{-\frac{(x-x_0)^2}{\sigma^2}} \quad \lim_{x \to \infty} g(x) = \varepsilon $$

$$ f(x) = \begin{cases}
-1 & \text{falls } x < 0 \\
0 & \text{falls } x = 0 \\
1 & \text{sonst} \end{cases} $$

Die -1 wird in der \emph{cases}-Umgebung leider linksbündig ausgerichtet, deshalb:

$$ f(x) = \left[ \begin{array}{l}
-1 & \text{falls } x < 0 \\
0 & \text{falls } x = 0 \\
1 & \text{sonst} \end{array} \right. $$

$$ \sum_{\substack{n=0 \\ 5 \uparrow n}}^{\infty} n^2 = \varphi $$

$$ \vec{x}(t) = \binom{x_1(t)}{x_2(t)} \in \mathbb{R}^2 $$

\end{document}

```

1 Drunter und drüber im Mathe-Modus

Für kompliziertere Gebilde, wie bspw. Vektoren oder Matrizen, gibt es die *array*-Umgebung, die die gleiche Syntax aufweist wie die *tabular*-Umgebung, und die ebenfalls eine Tabelle erzeugt.

$$A = \left(\begin{array}{ccc} x_1^1 & \dots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^n & \dots & x_n^n \end{array} \right)$$

Das kann man aber auch ohne Spaltenangabe mit der *matrix*-Umgebung von amsmath:

$$\det A = \left| \begin{matrix} x_1^1 & \dots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^n & \dots & x_n^n \end{matrix} \right| \stackrel{!}{=} 0$$

2 Mehr Spielereien

$$f(x) \xrightarrow{x \rightarrow \infty} 0$$

$$g(x) = e^{-\frac{(x-x_0)^2}{\sigma^2}} \quad \lim_{x \rightarrow \infty} g(x) = \varepsilon$$

$$f(x) = \begin{cases} -1 & \text{falls } x < 0 \\ 0 & \text{falls } x = 0 \\ 1 & \text{sonst} \end{cases}$$

Die -1 wird in der *cases*-Umgebung leider linksbündig ausgerichtet, deshalb:

$$f(x) = \left[\begin{array}{l} -1 & \text{falls } x < 0 \\ 0 & \text{falls } x = 0 \\ 1 & \text{sonst} \end{array} \right.$$

$$\sum_{\substack{n=0 \\ 5 \uparrow n}}^{\infty} n^2 = \varphi$$

$$\vec{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \in \mathbb{R}^2$$

```

\documentclass[a4paper]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm

\begin{document}

\section{Einige Bücher}
\label{sec:buecher}

Der \LaTeX{}-Begleiter \cite{gms} ist eine immer gern zitierte Quelle. Seine
Abhandlung zum Thema mathematischem Formelsatz \cite{gms8}
ist lesenswert für alle, die mathematische Formeln setzen müssen.\\

Von Helmut Kopka stammt die dreibändig Reihe
\LaTeX-Einführung \cite{kop}.\\

Das Zitat \cite{GMS} führt zu einer Fehlermeldung, da es keinen Eintrag mit
dem Schlüssel ‘‘GMS’’ gibt.

\begin{thebibliography}{9}

\bibitem{gms} \textbf{The \LaTeX\ Companion:} Goosens,
Mittelbach, Samarin \\
Addison-Wesley \textit{1994}

\bibitem{gms8} Kapitel 8 in \cite{gms}

\bibitem{kop} \textbf{\LaTeX\ Einführung:} Helmut Kopka\\
Addison-Wesley \textit{1994}

\end{thebibliography}

\end{document}

```

1 Einige Bücher

Der \LaTeX -Begleiter [1] ist eine immer gern zitierte Quelle. Seine Abhandlung zum Thema mathematischem Formelsatz [2] ist lesenswert für alle, die mathematische Formeln setzen müssen.

Von Helmut Kopka stammt die dreibändig Reihe \LaTeX -Einführung [3].

Das Zitat [?] führt zu einer Fehlermeldung, da es keinen Eintrag mit dem Schlüssel ‘‘GMS’’ gibt.

Literatur

- [1] **The \LaTeX Companion:** Goosens, Mittelbach, Samarin
Addison-Wesley *1994*
- [2] Kapitel 8 in [1]
- [3] **\LaTeX Einführung:** Helmut Kopka
Addison-Wesley *1994*

```
\documentclass[a4paper]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm

\begin{document}

\section{Einige Bücher}
\label{sec:buecher}

Und nun: Das Ganze noch einmal, unter Verwendung einer
Literaturdatenbank und mit Hilfe von BIB\TeX:

Von Helmut Kopka stammt die dreibändig Reihe \LaTeX\ -Einführung
\cite{kop}.

Der \LaTeX\ Begleiter ist eine immer gern zitierte Quelle. Seine
Abhandlung zum mathematischen Formelsatz \cite[Kapitel 8]{gms}
ist lesenswert für alle, die mathematische Formeln setzen müssen.

\bibliography{./bibtex/LaTeXBuecher,./bibtex/mehrLaTeXBuecher}
\bibliographystyle{plain}

\end{document}
```

1 Einige Bücher

Und nun: Das Ganze noch einmal, unter Verwendung einer Literaturdatenbank und mit Hilfe von BIB\TeX:

Von Helmut Kopka stammt die dreibändig Reihe \LaTeX -Einführung [2].

Der \LaTeX Begleiter ist eine immer gern zitierte Quelle. Seine Abhandlung zum mathematischen Formelsatz [1, Kapitel 8] ist lesenswert für alle, die mathematische Formeln setzen müssen.

Literatur

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The \LaTeX Companion*. Addison-Wesley, 1994.
- [2] Helmut Kopka. *\LaTeX Einführung*. Addison-Wesley, 1994.

```
@Book{kop,
  author =      {Kopka, Helmut},
  editor =      {},
  title =       {\LaTeX} {E}inf"uhrung},
  publisher =   {Addison-Wesley},
  year =        {1994},
  key =         {},
  volume =     {},
  number =     {},
  series =     {},
  address =    {},
  edition =    {},
  month =      {},
  note =       {},
  annote =     {}
}
```

```
@Book{gms,
  author =      {Goossens, Michel and Mittelbach, Frank and
  Samarin, Alexander},
  editor =      {},
  title =       {The {\LaTeX} Companion},
  publisher =   {Addison-Wesley},
  year =        {1994},
  key =         {},
  volume =     {},
  number =     {},
  series =     {},
  address =    {},
  edition =    {},
  month =      {},
  note =       {},
  annote =     {}
}
```

```

\documentclass[a4paper]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}\parindent=0mm

\begin{document}

\tableofcontents

\section{Verweise}\label{sec:verweis}

Das Unterkapitel \ref{sec:formel} auf
Seite \pageref{sec:formel} zeigt, dass sich ein Verweis
immer auf die aktuelle, tiefste Gliederungsebene bezieht,
ein Seitenverweis immer auf die aktuelle Seitennummer.\\

Wir bemerken, dass \LaTeX\ keinen Unterschied zwischen Rückwärts-
und Vorwärtsverweisen macht.\\

Das bestätigt sich auch, wenn wir auf Seite \pageref{sec:verweis}, am
Anfang von Kapitel \ref{sec:verweis}, nachschauen.\\

Diese Zeilen sollen den Leser darauf vorbereiten, dass er in Kürze auf
eine überaus bedeutsame Formel stoßen wird!

\subsection{Eine wichtige Formel\label{sec:formel}}

Wir kennen die binomische Formel:

\begin{equation} \label{eq:binom}
(a+b)^2 = a^2+2ab+b^2 \end{equation}

Das Ganze funktioniert auch mit Zahlen, es ist in der Formel \ref{eq:binom}
auf Seite \pageref{eq:binom} eindrucksvoll
beschrieben. Vielleicht steht auch in Anh. \ref{sec:anhang} noch
etwas über diese interessante Gleichung.

\appendix

\section{Auf Tabellen verweisen} \label{sec:anhang}
Na klar: Abb. \ref{fig:leer} sagt mehr als tausend Worte!

\begin{figure}[ht]
\centering
\begin{tabular}{|c|}
\hline
\hspace*{2cm}\rule{0mm}{2cm}\ \hline
\end{tabular}
\caption{Die Leere \label{fig:leer}}
\end{figure}
\end{document}

```

Inhaltsverzeichnis

1 Verweise	1
1.1 Eine wichtige Formel	1
A Auf Tabellen verweisen	1

1 Verweise

Das Unterkapitel 1.1 auf Seite 1 zeigt, dass sich ein Verweis immer auf die aktuelle, tiefste Gliederungsebene bezieht, ein Seitenverweis immer auf die aktuelle Seitennummer.

Wir bemerken, dass \LaTeX keinen Unterschied zwischen Rückwärts- und Vorwärtsverweisen macht.

Das bestätigt sich auch, wenn wir auf Seite 1, am Anfang von Kapitel 1, nachschauen.

Diese Zeilen sollen den Leser darauf vorbereiten, dass er in Kürze auf eine überaus bedeutsame Formel stoßen wird!

1.1 Eine wichtige Formel

Wir kennen die binomische Formel:

$$(a + b)^2 = a^2 + 2ab + b^2 \tag{1}$$

Das Ganze funktioniert auch mit Zahlen, es ist in der Formel 1 auf Seite 1 eindrucksvoll beschrieben. Vielleicht steht auch in Anh. A noch etwas über diese interessante Gleichung.

A Auf Tabellen verweisen

Na klar: Abb. 1 sagt mehr als tausend Worte!

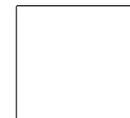


Abbildung 1: Die Leere

```

\documentclass[a4paper]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc} \parindent=0mm
\begin{document}

\section{Definition eigener Befehle}
Ich definiere mir jetzt einen eigenen Befehl, damit ich weniger tippen muss:

\newcommand{\xvec}{x_1 \ldots x_n} Jetzt geht das doch gleich viel einfacher:

\begin{equation}
\mathbf{A}=\left\{ \begin{array}{ccc}
\xvec & \cdots & \xvec \\
\vdots & \ddots & \vdots \\
\xvec & \cdots & \xvec
\end{array} \right\}
\end{equation}

\subsection{Befehle mit Parametern}

\newcommand{\pvec}[1]{#1_1 \ldots #1_n}

\begin{equation}
\mathbf{A}=\left\{ \begin{array}{ccc}
\pvec{x} & \cdots & \pvec{y} \\
\vdots & \ddots & \vdots \\
\pvec{y} & \cdots & \pvec{z}
\end{array} \right\}
\end{equation}

Das ist bestimmt ausbaufähig: \newcommand{\subvec}[3]{#1_#2 \ldots #1_#3}

\begin{equation}
\mathbf{A}=\left\{ \begin{array}{ccc}
\subvec{x}{1}{m} & \cdots & \subvec{y}{m}{n} \\
\vdots & \ddots & \vdots \\
\subvec{y}{m}{n} & \cdots & \subvec{z}{1}{n}
\end{array} \right\}
\end{equation}

Damit kann man doch sicher noch mehr machen:

\renewcommand{\subvec}[3]{#1_{#2} \ldots #1_{#3}}

\begin{equation}
\mathbf{A}=\left[ \subvec{A}{ij}{lk} \right]
\end{equation}

\end{document}

```

1 Definition eigener Befehle

Ich definiere mir jetzt einen eigenen Befehl, damit ich weniger tippen muss:
Jetzt geht das doch gleich viel einfacher:

$$\mathbf{A} = \begin{Bmatrix} x_1 \dots x_n & \cdots & x_1 \dots x_n \\ \vdots & \ddots & \vdots \\ x_1 \dots x_n & \cdots & x_1 \dots x_n \end{Bmatrix} \quad (1)$$

1.1 Befehle mit Parametern

$$\mathbf{A} = \begin{Bmatrix} x_1 \dots x_n & \cdots & y_1 \dots y_n \\ \vdots & \ddots & \vdots \\ y_1 \dots y_n & \cdots & z_1 \dots z_n \end{Bmatrix} \quad (2)$$

Das ist bestimmt ausbaufähig:

$$\mathbf{A} = \begin{Bmatrix} x_1 \dots x_m & \cdots & y_m \dots y_n \\ \vdots & \ddots & \vdots \\ y_m \dots y_n & \cdots & z_1 \dots z_n \end{Bmatrix} \quad (3)$$

Damit kann man doch sicher noch mehr machen:

$$\mathbf{A} = [A_{ij} \dots A_{lk}] \quad (4)$$

```

\documentclass[a4paper]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm

\begin{document}

\section{Umdefinieren von \LaTeX-Makros}

Ändern wir mal die Seiten- und Formelnummerierung.

\renewcommand{\thepage}{\Roman{section}.\arabic{page}}

\renewcommand{\theequation}{\Roman{section}.\arabic{equation}}

\begin{equation}\label{eq:A3}
\mathbf{A}=\left( \begin{array}{ccc}
a_{11} & \cdots & a_{1n} \\
\vdots & \ddots & \vdots \\
a_{m1} & \cdots & a_{mn}
\end{array} \right)
\end{equation}

\begin{equation} 1+1=2 \end{equation}

Wir wollen diese grundlegende
Gl.(\ref{eq:A3}) auf Seite \pageref{eq:A3} nicht vergessen!

\tableofcontents

\end{document}

```

1 Umdefinieren von \LaTeX -Makros

Ändern wir mal die Seiten- und Formelnummerierung.

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad (\text{I.1})$$

$$1 + 1 = 2 \quad (\text{I.2})$$

Wir wollen diese grundlegende Gl.(I.1) auf Seite I.1 nicht vergessen!

Inhaltsverzeichnis

1 Umdefinieren von \LaTeX -Makros	I.1
-------------------------------------	-----

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\usepackage{rotating} \parindent=0mm

\begin{document}

\section{\LaTeX schräg}

Jetzt wird's schräg: Beliebige Textteile lassen sich um einen
beliebigen Winkel drehen (falls der Druckertreiber mitmacht).
Probieren wir's doch mal: \\
Vorher
\begin{rotate}{60}
Dieser Text ist gedreht
\end{rotate}
Nachher

Es gibt aber noch eine andere Möglichkeit, bei der automatisch Platz
gelassen wird: \\
Vorher
\begin{turn}{60}
Dieser Text ist gedreht
\end{turn}
Nachher

\vspace{3cm}
\fbxsep=0.pt
Zur Verdeutlichung malen wir jetzt mal Rahmen um die gedrehten
Textteile: \\
Vorher
\begin{rotate}{60}
\fbx{Dieser Text ist gedreht}
\end{rotate}
Nachher

Es gibt aber noch eine andere Möglichkeit, bei der automatisch Platz
gelassen wird: \\
Vorher
\fbx{\begin{turn}{60}
\fbx{Dieser Text ist gedreht}
\end{turn}
}
Nachher

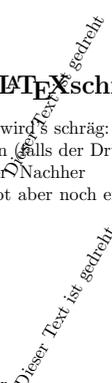
\bigskip
Ein weiterer Befehl lässt sich als Abkürzung einsetzen, wenn man
einfach mal \begin{sideways}senkrecht\end{sideways} Text erzeugen will.

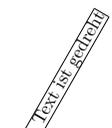
\newpage

```

1 ~~LaTeX~~ schräg

Jetzt wird's schräg: Beliebige Textteile lassen sich um einen beliebigen Winkel drehen (falls der Druckertreiber mitmacht). Probieren wir's doch mal:
 Vorher Nachher
 Es gibt aber noch eine andere Möglichkeit, bei der automatisch Platz gelassen wird:

Vorher  Nachher



Zur Verdeutlichung malen wir jetzt mal Rahmen um die gedrehten Textteile:
 Vorher Nachher
 Es gibt aber noch eine andere Möglichkeit, bei der automatisch Platz gelassen wird:

Vorher  Nachher

Ein weiterer Befehl lässt sich als Abkürzung einsetzen, wenn man einfach mal `\begin{sideways}senkrecht\end{sideways}` Text erzeugen will.

```
Der Drehpunkt ist immer der Bezugspunkt einer Box. Verwenden wir eine
\verb+\parbox+,
\begin{turn}{-45}
\parbox[t]{4.cm}{Erfreut stellen wir fest, dass
\begin{sideways}\LaTeX\end{sideways} sogar verschachtelte Drehungen
zulässt!}
\end{turn}
so haben wir also mehrere Möglichkeiten für die Position des
Drehpunkts.
```

```
\bigskip
Der Drehpunkt ist immer der Bezugspunkt einer Box. Verwenden wir eine
\verb+\parbox+,
\begin{turn}{-45}
\parbox{4.cm}{Erfreut stellen wir fest, dass
\begin{sideways}\LaTeX\end{sideways} sogar verschachtelte Drehungen
zulässt!}
\end{turn}
so haben wir also mehrere Möglichkeiten für die Position des
Drehpunkts.
```

```
\bigskip
Der Drehpunkt ist immer der Bezugspunkt einer Box. Verwenden wir eine
\verb+\parbox+,
\begin{turn}{-45}
\parbox[b]{4.cm}{Erfreut stellen wir fest, dass
\begin{sideways}\LaTeX\end{sideways} sogar verschachtelte Drehungen
zulässt!}
\end{turn}
so haben wir also mehrere Möglichkeiten für die Position des
Drehpunkts.
```

```
\end{document}
```

Der Drehpunkt ist immer der Bezugspunkt einer Box. Verwenden wir eine
`\parbox,` so haben wir also mehrere Möglichkeiten

*Erfreut stellen wir fest,
dass \LaTeX sogar verschachtelte
Drehungen zulässt!*

für die Position des Drehpunkts.

Der Drehpunkt ist immer der Bezugspunkt einer Box. Verwenden wir eine

`\parbox,` so haben wir also mehrere Möglichkeiten

*Erfreut stellen wir fest,
dass \LaTeX sogar verschachtelte
Drehungen zulässt!*

für die Position des Drehpunkts.

Der Drehpunkt ist immer der Bezugspunkt einer Box. Verwenden wir eine

`\parbox,` so haben wir also mehrere Möglichkeiten

*Erfreut stellen wir fest,
dass \LaTeX sogar verschachtelte
Drehungen zulässt!*

für die Position des Drehpunkts.

```

\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage{ngerman,rotating} \parindent=0mm

\begin{document}

Tabellensatz mal ein wenig anders: \bigskip\bigskip

\begin{center}
\begin{tabular}{cccc}

\begin{rotate}{45}{Messgröße}\end{rotate} &
\begin{rotate}{45}{Minimalwert}\end{rotate} &
\begin{rotate}{45}{Maximalwert}\end{rotate} &
\begin{rotate}{45}{Mittelwert}\end{rotate} \\
\hline
$L_1$ & 5.1 & 5.5 & 5.2 \\
$L_2$ & 8.9 & 9.2 & 9.1 \\
$B_1$ & 2.2 & 2.5 & 2.3 \\
$B_2$ & 1.8 & 2.1 & 1.9 \\
\hline

\end{tabular}

\end{center}

\bigskip\bigskip

\begin{center}
\begin{sideways}
\begin{tabular}{cccc}

\begin{rotate}{45}{Messgröße}\end{rotate} &
\begin{rotate}{45}{Minimalwert}\end{rotate} &
\begin{rotate}{45}{Maximalwert}\end{rotate} &
\begin{rotate}{45}{Mittelwert}\end{rotate} \\
\hline
$L_1$ & 5.1 & 5.5 & 5.2 \\
$L_2$ & 8.9 & 9.2 & 9.1 \\
$B_1$ & 2.2 & 2.5 & 2.3 \\
$B_2$ & 1.8 & 2.1 & 1.9 \\
\hline

\end{tabular}
\end{sideways}
\end{center}

\end{document}

```

Tabellensatz mal ein wenig anders:

	Messgröße	Minimalwert	Maximalwert	Mittelwert
L_1	5.1	5.5	5.2	
L_2	8.9	9.2	9.1	
B_1	2.2	2.5	2.3	
B_2	1.8	2.1	1.9	

Messgröße	Minimalwert	Maximalwert	Mittelwert
L_1	5.1	5.5	5.2
L_2	8.9	9.2	9.1
B_1	2.2	2.5	2.3
B_2	1.8	2.1	1.9

```
\documentclass[a4paper,12pt]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm
\renewcommand\baselinestretch{1.2}
\begin{document}
```

Wie wir in der Präambel sehen, wurde der faktor für den Zeilenabstand auf 1.2 verändert. Probier

```
\addtolength{\parskip}{.5\baselineskip}
```

Jetzt haben wir einen zusätzlichen Abstand zwischen der letzten Zeile des alten und der ersten Zeile eines neuen Absatzes, die gerade die Hälfte des normalen Zeilenabstands ausmacht. Ist doch auch nicht schlecht!

Damit man das auch gebührend bewundern kann, fangen wir hier jetzt einen neuen Absatz an. Dieser Absatz sollte so lang sein, dass er über mehrere Zeilen geht, damit man die Abstände gut vergleichen kann.

```
\addtolength{\baselineskip}{2mm}
```

Und jetzt haben wir den Zeilenabstand selbst verändert. Wie man hier gut sehen kann, ist \LaTeX sehr flexibel: Man darf fast alles verändern --- aber man sollte schon wissen, was man tut!

```
\the\baselineskip
```

```
\newlength{\boxbreit}
\settowidth{\boxbreit}{Schaut mal Alle her!}
```

```
\parbox{\boxbreit}{Schaut mal Alle her! Jetzt haben wir eine Box gemacht, die
genau so breit ist, dass ein bestimmter Text gerade hinein passt.}
```

Wir erkennen, daß der Zeilenabstand nur im "normalen" Text größer geworden ist. Um ihn global zu setzen, muss man in der Dokumentenpräambel `\verb+\renewcommand\baselinestretch{faktor}+` schreiben.

```
\begin{minipage}{\boxbreit}
Schaut mal Alle her! Jetzt haben wir eine Box gemacht, die
genau so breit ist, dass ein bestimmter Text gerade hinein passt.
\end{minipage}
```

Und das ist die Breite der Box: $\the\boxbreit$. Der Zeilenabstand beträgt gerade $\the\baselineskip$, und der zusätzliche Abstand zwischen Paragraphen ist $\the\parskip$.

```
\end{document}
```

Wie wir in der Präambel sehen, wurde der faktor für den Zeilenabstand auf 1.2 verändert. Probieren wir gleich noch ein paar Längenbefehle:

Jetzt haben wir einen zusätzlichen Abstand zwischen der letzten Zeile des alten und der ersten Zeile eines neuen Absatzes, die gerade die Hälfte des normalen Zeilenabstands ausmacht. Ist doch auch nicht schlecht!

Damit man das auch gebührend bewundern kann, fangen wir hier jetzt einen neuen Absatz an. Dieser Absatz sollte so lang sein, dass er über mehrere Zeilen geht, damit man die Abstände gut vergleichen kann.

Und jetzt haben wir den Zeilenabstand selbst verändert. Wie man hier gut sehen kann, ist \LaTeX sehr flexibel: Man darf fast alles verändern — aber man sollte schon wissen, was man tut! 23.09048pt

Schaut mal Alle her!

Jetzt haben wir eine Box gemacht, die genau so breit ist, dass ein bestimmter Text gerade hinein passt.

Wir erkennen, daß der Zeilenabstand nur im „normalen“ Text größer geworden ist. Um ihn global zu setzen, muss man in der Dokumentenpräambel `\renewcommand\baselinestretch{faktor}` schreiben.

Schaut mal Alle her!

Jetzt haben wir eine Box gemacht, die genau so breit ist, dass ein bestimmter Text gerade hinein passt.

Und das ist die Breite der Box: 105.74474pt. Der Zeilenabstand beträgt gerade 23.09048pt, und der zusätzliche Abstand zwischen Paragraphen ist 8.69997pt plus 1.0pt.

```
\documentclass{article}
\usepackage{ngerman,fancyhdr}
\usepackage[latin1]{inputenc} \parindent=0mm
```

```
\pagestyle{fancy}
```

```
\lhead{Links oben}
\chead{Mitte oben}
\rhead{Rechts oben}
\lfoot{Links unten}
\cfoot{Mitte unten}
\rfoot{Rechts unten}
\renewcommand{\footrulewidth}{.1pt}
```

```
\begin{document}
```

```
\section{Kopf und Fuß mit Fancyhdr}
Wie man oben und unten sieht\dots
```

```
\end{document}
```

1 Kopf und Fuß mit Fancyhdr

Wie man oben und unten sieht...

```

\documentclass[a4paper]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}

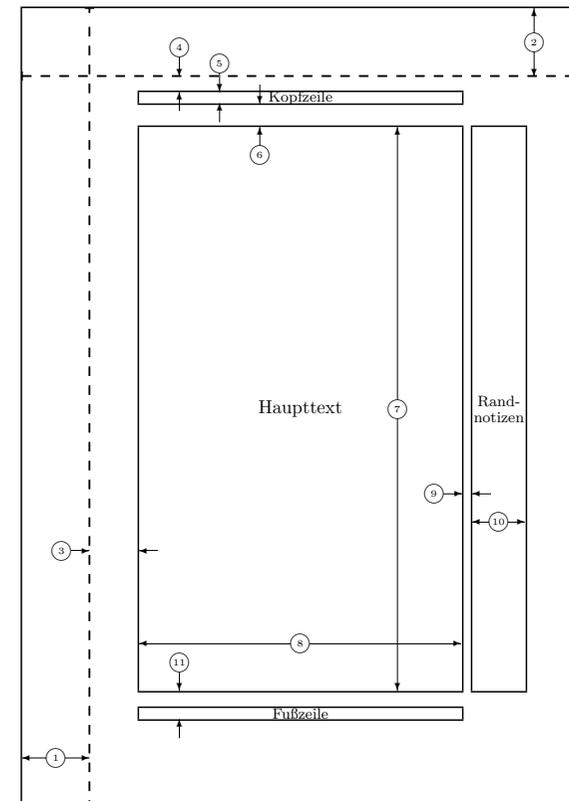
\usepackage[ngerman]{layout}

\begin{document}

\layout

\end{document}

```



- | | |
|-------------------------|---------------------------------------|
| 1 ein Zoll + \hoffset | 2 ein Zoll + \voffset |
| 3 \oddsidemargin = 53pt | 4 \topmargin = 17pt |
| 5 \headheight = 12pt | 6 \headsep = 25pt |
| 7 \textheight = 598pt | 8 \textwidth = 345pt |
| 9 \marginparsep = 11pt | 10 \marginparwidth = 57pt |
| 11 \footskip = 30pt | \marginparpush = 5pt (ohne Abbildung) |
| \hoffset = 0pt | \voffset = 0pt |
| \paperwidth = 597pt | \paperheight = 845pt |

```

\documentclass{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}

\begin{document}

\section{\LaTeX und seine Zähler}

Schauen wir doch mal, ob wir \LaTeX{} beim Zählen in's Handwerk pfuschen
können:
%
\begin{enumerate}
  \item Dieses war der erste Streich\dots
  \item \dots doch der zweite folgt sogleich!
\addtocounter{enumi}{2}
  \item Nanu, fehlt hier nicht etwas?
\setcounter{enumi}{13}
  \item Jetzt schlägt's aber dreizehn!
\addtocounter{enumi}{-20}
  \item Hoppla --- was ist denn jetzt passiert?
\end{enumerate}

\refstepcounter{section}
\section{Das nächste Kapitel}

Moment mal, da haben wir doch etwas übersprungen?

\addtocounter{section}{-2}

\section{Noch ein Kapitel\label{sec:zwei}}

Ach so, hier ist es ja!

\bigskip\noindent
Hurra --- endlich brauche ich nicht mehr selbst bis drei zu zählen: Ich mache
mir jetzt meinen eigenen \LaTeX-Zähler!

\newcounter{meinzaehl}
\setcounter{meinzaehl}{1}

Und los geht's: \arabic{meinzaehl} \dots
%
\stepcounter{meinzaehl}
\arabic{meinzaehl} \dots
%
\stepcounter{meinzaehl}
\arabic{meinzaehl}!!!

\end{document}

```

1 \LaTeX und seine Zähler

Schauen wir doch mal, ob wir \LaTeX beim Zählen in's Handwerk pfuschen können:

1. Dieses war der erste Streich...
2. ... doch der zweite folgt sogleich!
5. Nanu, fehlt hier nicht etwas?
14. Jetzt schlägt's aber dreizehn!
- 5. Hoppla — was ist denn jetzt passiert?

3 Das nächste Kapitel

Moment mal, da haben wir doch etwas übersprungen?

2 Noch ein Kapitel

Ach so, hier ist es ja!

Hurra — endlich brauche ich nicht mehr selbst bis drei zu zählen: Ich mache mir jetzt meinen eigenen \LaTeX -Zähler!

Und los geht's: 1 ... 2 ... 3!!!

```
\input{bsp-teildokumente-header}  
  
\begin{document}  
  
\include{bsp-teildokumente01-titel}  
\include{bsp-teildokumente01-kap1}  
\include{bsp-teildokumente01-kap2}  
  
\end{document}
```

Titelkopf

Diplomarbeit

Der Titel

Der Author

8. Oktober 2008

Die Herausgeber

Inhaltsverzeichnis

1	Erstes Kapitel	3
---	----------------	---

1 Erstes Kapitel

Dies ist das erste Kapitel,...

2 Zweites Kapitel

Dies ist das zweite Kapitel,...

```
\documentclass[a4paper]{scrreprt}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\parindent=0mm
```

```
\titlehead{Titelkopf}
```

```
\subject{Diplomarbeit}
```

```
\title{Der Titel}
```

```
\author{Der Author}
```

```
\date{\today}
```

```
\publishers{Die Herausgeber}
```

```
\maketitle
```

```
\tableofcontents
```

```
\chapter{Erstes Kapitel}
```

```
Dies ist das erste Kapitel,\dots
```

`\chapter{Zweites Kapitel}`

`Dies ist das zweite Kapitel,\dots`

Einführung in \LaTeX – Übungen –

Felix Widrig

2008

1 Übung: grundeigenschaften von L^AT_EX

1. Im Verzeichnis ueb-basics sind fehlerhafte L^AT_EX-Quelltexte. Korrigieren Sie alle Fehler.
2. Setzen Sie dieses Dokument:

Erst etwas Text, dann 5em Abstand und dann wieder Text. Dann mehr Text, nur um einen Zeilenumbruch zu erzwingen. Dann 5ex vertikaler Abstand:

Dann geht es weiter mit „Anführungszeichen“, und zwar testen wir «einige». Immer gut sind die "doublequotes". Dann ein bigskip vor dem nächsten Absatz.

Hier wollen wir nun etwas besonderes testen, das hfill Kommando:
"A" 'B' <C> 'D' "E"

2 Übung: Dokumentklassen

1. Im Verzeichnis ueb-dokumentklassen sind fehlerhafte \LaTeX -Quelltexte. Korrigieren Sie alle Fehler.
2. Setzen Sie das auf der Rückseite sich befindende Dokument (6 Seiten!):

Studienarbeit

Umgang mit L^AT_EX

Marcel Mustermann Camille Musterfrau

23.08.2006

Fax Ingwer Verlag

Zusammenfassung

Diese Arbeit dient zur Übung im Umgang mit L^AT_EX. Ich hoffe, es bringt was. Wir üben, wie eine Titelseite gesetzt wird, und wie die Grundbefehle zum Gliedern eines Dokumentes aufgebaut sind. Achten wir nun auf diese Verweise, vorsicht, nicht einfach die Zahlen schreiben, sondern immer mit Referenzen arbeiten: Im Kapitel 1 befinden sich Informationen, insbesondere im Abschnitt 1.1.1 auf Seite V. Der Anhang beginnt auf Seite VI.

Inhaltsverzeichnis

Vorwort	IV
1. Erstes Kapitel	V
1.1. Der Anfang	V
1.1.1. Der Rest	V
A. Erster Anhang	VI

Vorwort

Dem folgte ein Kapitel.

1. Erstes Kapitel

1.1. Der Anfang

Gefolgt von dem Rest.

1.1.1. Der Rest

Dann fertig.

V

A. Erster Anhang

VI

3 Übung: Textformatierung

1. Im Verzeichnis ueb-textformatierung sind fehlerhafte \LaTeX -Quelltexte. Korrigieren Sie alle Fehler.
2. Setzen Sie das auf der Rückseite sich befindende Dokument (2 Seiten):

1 Schriftarten und Textausrichtung

Große Überschrift zentriert in Fettschrift

Dann etwas Text. Ich zitiere nun aus einem Buch, es ist ein Text, der sich zu lesen lohnt:

Der Held kam mit seinem Schwert, erledigte den Drachen, und befreite die Prinzessin. Diese war von seiner Heldentat begeistert und verliebte sich in ihn.

Das war also die Passage aus jenem Buch.

TITEL IN KAPITÄLCHEN¹

Dann ein kurzer Abschnitt mit SansSerif Schrift, leicht geneigt. Hier steht nichts interessantes, nur Fülltext. Dies ist noch wichtig:

1. nichts
 - auch nichts
 - und wieder nichts
2. alles

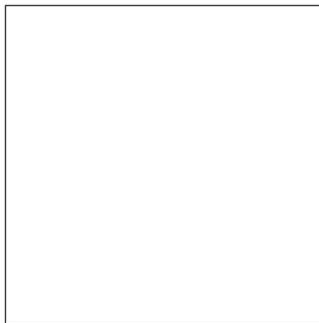
2 Ein paar Boxen

Nun folgt eine volle Zeile Text, damit wir sehen, wie das, was danach unten steht, in Wirklichkeit aussieht.

Etwas Text in einer Box, um zu demonstrieren, wie zwei Boxen nebeneinander gesetzt werden können, und das ganze noch umrahmt! hier noch etwas Fülltext, um die Box groß genug zu machen.

Etwas Text in einer Box, um zu demonstrieren, wie zwei Boxen nebeneinander gesetzt werden können, und das ganze noch umrahmt! hier noch etwas Fülltext, um die Box groß genug zu machen.

Hier setzten wir noch eine Box:



Die Leere

¹sogar mit Fußnote

3 Tabulatoren

WYSIWYG-System: Vorteile: Leicht zu lernen

Schnelle Ergebnisse

Weit verbreitet

Nachteile: Eingeschränkte Gestaltungsmöglichkeiten

Formelsatz problematisch

Laufen oft nicht sehr stabil

Schwierigkeiten bei großen Dokumenten

Lizenzgebühren

Einsatzgebiete: Einfache Dokumente

Private Nutzung

L^AT_EX: Vorteile: Enorme Vielfalt an Gestaltungsmöglichkeiten

Hervorragender Formelsatz

Auch bei großen Dokumenten stabil

Kostenlos

Nachteile: Schwieriger zu lernen

Einfache Texte etwas umständlich

Nicht überall installiert

Einsatzgebiete: Wissenschaftlicher Textsatz

Texte mit vielen Formeln

Große Dokumente (Bücher etc.)

Eintrag 1:

Vorname: Johannes

Nachname: Gutenberg

Geburtstag: Um 1400

4 Übung: Gleitobjekte

1. Im Verzeichnis ueb-gleitobjekte sind fehlerhafte \LaTeX -Quelltexte. Korrigieren Sie alle Fehler.

5 Übung: Tabellen

1. Im Verzeichnis ueb-tabellen sind fehlerhafte \LaTeX -Quelltexte. Korrigieren Sie alle Fehler.
2. Setzen Sie das auf der nächsten Seite sich befindende Dokument (2 Seiten):

Tabellenverzeichnis

1	Die Leere	1
2	Misthaufen im Landkreis Kleinkreis	1
3	Der gleiche Mist	2
4	Immer noch der gleiche Mist	2
5	Mist! Mist! Mist!	2

1 Merkwürdige Tabelle

Achtung, das hier ist eine Tabelle, keine Box!

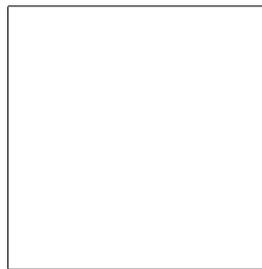


Tabelle 1: Die Leere

2 Anspruchsvolle Tabellen

Erster Versuch:

Erhebung des Bauernverbandes		
Ort	Anzahl der Haufen	Größe [cbm]
Vorderkleinbach	3	1.24
Hintergroßbach	7	11.5
Mittelkuhdorf	5	7.99
Entenhausen	0	—

Tabelle 2: Misthaufen im Landkreis Kleinkreis

So sieht's schöner aus:

Erhebung des Bauernverbandes		
Ort	Anzahl der Haufen	Größe [cbm]
Vorderkleinbach	3	1.24
Hintergroßbach	7	11.5
Mittelkuhdorf	5	7.99
Entenhausen	0	—

Tabelle 3: Der gleiche Mist

Oder probieren wir's mal so:

Erhebung des Bauernverbandes		
Ort	Anzahl der Haufen	Größe [cbm]
Vorderkleinbach	3	1.24
Hintergroßbach	7	11.5
Mittelkuhdorf	5	7.99
Entenhausen	0	—

Tabelle 4: Immer noch der gleiche Mist

Und noch eine Variante:

Erhebung des Bauernverbandes		
Ort	Anzahl der Haufen	Größe [cbm]
Vorderkleinbach	3	1.24
Hintergroßbach	7	11.5
Mittelkuhdorf	5	7.99
Entenhausen	0	—

Tabelle 5: Mist! Mist! Mist!

6 Übung: Mathe

1. Im Verzeichnis `ueb-mathe` sind fehlerhafte \LaTeX -Quelltexte. Korrigieren Sie alle Fehler.
2. Setzen Sie das auf der nächsten Seite sich befindende Dokument (2 Seiten):

$$y_1 = \frac{3 + \frac{2x}{9y}}{\frac{3}{4} + z}$$

$$\int_{\alpha}^{\gamma} f dx = \int_{\alpha}^{\beta} f dx + \int_{\beta}^{\gamma} f dx \quad (1)$$

für alle $\gamma \in [\alpha, \beta]$

Die Meßwerte $x_1 < x_2 < \dots < x_r$ traten p_1, p_2, \dots, p_r -mal in einer Meßreihe auf. Der Mittelwert x und die Streuung s ist dann

$$x = \frac{1}{n} \sum_{i=1}^r p_i x_i, \quad s = \sqrt{\frac{1}{n} \sum_{i=1}^r p_i (x_i - x)^2}$$

mit $n = p_1 + p_2 + \dots + p_r$.

Die Gammafunktion $\Gamma(x)$ ist definiert als:

$$\begin{aligned} \Gamma(x) &\equiv \lim_{n \rightarrow \infty} \prod_{\nu=0}^{n-1} \frac{n! n^{x-1}}{x + \nu} \\ &= \lim_{n \rightarrow \infty} \frac{n! n^{x-1}}{x(x+1)(x+2) \dots (x+n-1)} \\ &\equiv \int_0^{\infty} e^{-t} t^{x-1} dt \end{aligned} \quad (2)$$

Die Integraldefinition gilt nur für $x > 0$ (2. Eulersches Integral).

Der kürzeste Abstand zweier Geraden, deren Gleichungen in der Form

$$\frac{x - x_1}{l_1} = \frac{y - y_1}{m_1} = \frac{z - z_1}{n_1} \quad \text{und} \quad \frac{x - x_2}{l_2} = \frac{y - y_2}{m_2} = \frac{z - z_2}{n_2}$$

gegeben sind, lässt sich nach der Formel

$$\pm \frac{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \end{vmatrix}}{\sqrt{\begin{vmatrix} l_1 & m_1 \\ l_2 & m_2 \end{vmatrix}^2 + \begin{vmatrix} m_1 & n_1 \\ m_2 & n_2 \end{vmatrix}^2 + \begin{vmatrix} n_1 & l_1 \\ n_2 & l_2 \end{vmatrix}^2}}$$

berechnen. Verschwindet der Zähler, so schneiden sich die Geraden im Raum.

Laurent-Entwicklung:

Mit $c_n = \frac{1}{2\pi i} \oint (\zeta - a)^{-n-1} f(\zeta) d\zeta$ gilt für jede Funktion $f(z)$ die Darstellung ($n = 0, \pm 1, \pm 2, \dots$)

$$\begin{aligned} f(x) &= \sum_{n=-\infty}^{+\infty} c_n (z-a)^n \\ &= \begin{cases} c_0 & + c_1(z-a) + c_2(z-a)^2 + \dots + c_n(z-a)^n + \dots \\ & + c_{-1}(z-a)^{-1} + c_{-2}(z-a)^{-2} + \dots + c_{-n}(z-a)^{-n} + \dots \end{cases} \\ &= \begin{cases} c_0 & + c_1(z-a) & + c_2(z-a)^2 & + \dots & + c_n(z-a)^n & + \dots \\ & + c_{-1}(z-a)^{-1} & + c_{-2}(z-a)^{-2} & + \dots & + c_{-n}(z-a)^{-n} & + \dots \end{cases} \\ &= \begin{cases} c_0 + c_1(z-a) & + c_2(z-a)^2 & + \dots & + c_n(z-a)^n & + \dots \\ & + c_{-1}(z-a)^{-1} + c_{-2}(z-a)^{-2} & + \dots & + c_{-n}(z-a)^{-n} & + \dots \end{cases} \end{aligned}$$

7 Übung: Literaturverzeichnis setzen mit BibTeX

1. Im Verzeichnis ueb-literatur sind fehlerhafte \LaTeX -Quelltexte. Korrigieren Sie alle Fehler.
2. Setzen Sie dieses Dokument (Legen Sie zunächst die nötige BibTeX-Datei an, laden Sie dann das Paket bibgerm und benutzen sie den stil gerplain):

Verschiedene Arten von Literatur

Wenn wir zum Beispiel eine Arbeit über Schwarze Löcher schreiben wollen, dann kommen wir an den folgenden Literaturstellen nicht vorbei: In dem Buch [1] werden alle Fragen über Schwarze Löcher beantwortet, die Ihnen wahrscheinlich noch nie eingefallen sind. Der Artikel [2] erklärt, wie Schwarze Löcher schwingen können. Die Doktorarbeit [3], veröffentlicht in dem Artikel [4], zeigt eine Unmenge von Rechnungen zu dem Thema.

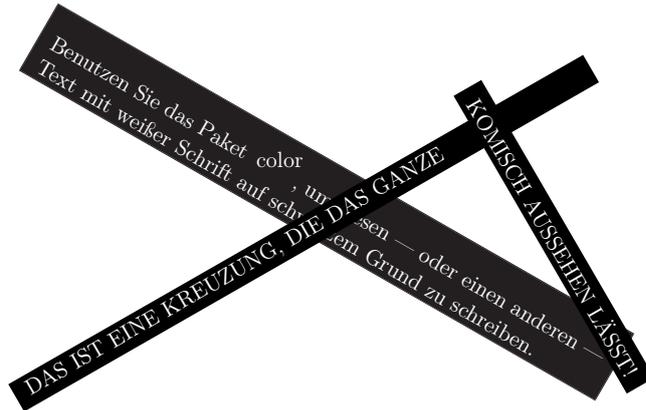
Literatur

- [1] CHANDRASEKHAR, SUBRAHMANYAN: *The Mathematical Theory of Black Holes*. Pergamon Press, 1983.
- [2] FERRARI, VALERIA und BAHRAM MASHHOON: *Oscillations of a Black Hole*. Phys. Rev. Lett., 52:1361 – 1364, 1984.
- [3] LEAVER, EDWARD W.: *The quasi-normal modes of Schwarzschild and Kerr black holes*. Doktorarbeit, University of Utah, Salt Lake City, 1984.
- [4] LEAVER, EDWARD W.: *An Analytic Representation for the Quasi-Normal Modes of Kerr Black Holes*. Proc. R. Soc. Lond. A, 402:285, 1985.

8 Übung: Farbe benutzen mit L^AT_EX

1. Setzen Sie dieses Dokument (überlegen Sie, wie sie verschiedene Boxen ineinander schachteln müssen, um den gewünschten Effekt zu erzielen.):

Erst einmal normaler Text, damit eine Zeile zumindest voll mit Text ist. So sehen wir den Unterschied:



9 Übung: Besonderheiten

1. Bekanntlich kann man im Mathemodus Mengen, wie zB $\{1, 2, 3\}$ mit $\left\{1, 2, 3\right\}$ beschreiben. Definieren Sie ein Makro `\set{}`, mit der Sie im Mathemodus Mengen beschreiben können. Das Makro soll so funktionieren:

$\set{a, b}$ ergibt $\{a, b\}$, oder zB $\set{4, 6}$ ergibt $\{4, 6\}$.

2. Angenommen Sie sollen mit dem Paket *amsmath* viele Matrizen in der Form $\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$, usw schreiben. Wenn Sie jedes Mal die Form `\left(\begin{matrix} 2&3 \\ 4&5 \end{matrix}\right)` benutzen, haben Sie viel zu Tippen.

Definieren Sie deshalb ein Makro `\mat{}`, mit dem Matrizen so gesetzt werden:

$\mat{2&3\4&5}$ für $\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$,

$\mat{1&2&0\2&3&4}$ für $\begin{pmatrix} 1 & 2 & 0 \\ 2 & 3 & 4 \end{pmatrix}$, usw.

3. Definieren Sie ein eigenes Makro namens `\mybox{ }{ }`, mit dem Sie die Breite von mehrzeiligen Boxen anhand einer Musterzeile berechnen können. Hier ein Beispiel, wie das Makro funktionieren soll:

```
\mybox{Das ist so ... und nicht anders,}
{Das ist so ... und nicht anders, nämlich sonst geht es nicht weiter}
```

Das ist so ... und nicht anders,
nämlich sonst geht es nicht
weiter

```
\mybox{Diesmal andere Breite.}
{Diesmal andere Breite. Weil wirs anders haben wollten, vor allem hier...}
```

Diesmal andere Breite.
Weil wirs anders ha-
ben wollten, vor allem
hier...

Hinweis Definieren Sie sich eine eigene Länge, die Sie dann innerhalb des Makros auf die entsprechende Breite setzen.

4. Setzen Sie das auf der nächsten Seite sich befindende Dokument (**Hinweise:** Die Tabelle: denkbar wären 5 Spalten mit dem Paket *dcolumn*, sonst 8 Spalten. Sie benötigen nicht das *multirrow* Paket für den senkrechten Text. Benutzen dafür Sie das Paket *rotating*. Welche der Umgebungen aus diesem Paket bewirkt das gewünschte Ergebnis?

Papierformate in mm gemäß DIN 478				
	Haupt-Reihe A	Zusatzreihen		
		B	C	
Formatbezeichnung	0	841×1189	1000×1414	917×1297
	1	594×841	707×1000	648×917
	2	...×...	...×...	...×...
	3	...×...	...×...	...×...
	4	...×...	...×...	...×...
	5	...×...	...×...	...×...
	6	...×...	...×...	...×...
	7	...×...	...×...	...×...
	8	...×...	...×...	...×...
	9	37×52	44×62	

Anwendungen der B- und C-Reihe:

Die Formate der Zusatzreihen B und C gelten nur für abhängige Papiergrößen wie Briefhüllen (DIN 678) Fensterbriefhüllen (DIN680) Mappen, Aktendeckel, usw.