

# L<sup>A</sup>T<sub>E</sub>X - G<sup>E</sup>B<sub>R</sub>A<sup>U</sup>C<sub>H</sub>S<sup>A</sup>N<sup>L</sup>E<sup>I</sup>T<sub>U</sub><sub>N</sub>G

Release 1.01

Annette Huck

**`annette@hopfenwiesen.de`**

Heidelberg, Lyon, Berlin

4. Juli 2007

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>3</b>
<b>1 Einführung</b>	<b>5</b>
Wie erstellt man ein Dokument?	5
Die Syntax	6
Die Vorspannzeilen	8
<b>2 Textformatierungen</b>	<b>9</b>
Besondere Zeichen	9
Umbrüche	9
Feste Maße	10
Elastische Maße	11
Schriftgröße	11
Quelltext ausgeben	12
Zeichensätze	12
Hervorhebungen	13
URL	13
Silbentrennung	14
Ligaturen und Kerning	15
Akzente	16
<b>3 Textanordnung</b>	<b>17</b>
Aufzählung und Nummerierung	17
Ausrichtung	18
Tabulatoren	19
Tabellen	20
Boxen	22
Einzeilige Boxen (genannt: LR-Boxen)	22
Einzeilige Boxen speichern	22
Einzeilige Boxen verschieben	22
Absatzboxen (genannt: Parbox)	23
Balkenboxen	23
Box-Stilparameter	23
Rahmen	24
Das boxedminipage-Paket	24
Das shadow-Paket	24
Das fancybox-Paket	24
Das color-Paket	25
Drehen	25
Trennlinie	26
Regelsätze	26
Das theorem-Paket	26

Absatzeinzug . . . . .	28
Das shapepar-Paket . . . . .	28
<b>4 Mathemodus</b> . . . . .	<b>29</b>
Mathematische Umgebungen . . . . .	29
Schriftgrößen im Mathemodus . . . . .	29
Formelelemente . . . . .	30
Einfache Formelelemente . . . . .	30
Fortsetzungspunkte . . . . .	31
Mathematische Akzente . . . . .	31
Funktionsnamen . . . . .	31
Klammern . . . . .	32
Matrizen und Felder . . . . .	32
Über-/Unterstreichen und Aufgestockte Symbole . . . . .	33
Änderung der Schrift in Formeln . . . . .	33
Formeln anordnen . . . . .	34
gerahmte Formeln . . . . .	35
Formelnummern und Referenzen . . . . .	35
Formatierung von Zählern . . . . .	36
Mathematische Symbole . . . . .	36
Vergleichssymbole . . . . .	36
Binäre Operationssymbole . . . . .	37
Griechische Kleinbuchstaben . . . . .	37
Griechische Großbuchstaben . . . . .	37
Pfeile . . . . .	38
Sonderzeichen . . . . .	38
<b>5 Farben</b> . . . . .	<b>39</b>
Farbige Schrift und Boxen . . . . .	39
Farbtabellen . . . . .	39
model: [gray]{0.xx} . . . . .	39
model: [rgb]{0..1,0..1,0..1} . . . . .	39
model: [cmyk]{0..1,0..1,0..1,0..1} . . . . .	40
model: [named] . . . . .	40
<b>Literaturverzeichnis</b> . . . . .	<b>41</b>

# Kapitel 1

## Einführung

Herzlich Willkommen zu meinem Langzeitprojekt  $\text{\LaTeX}$ -GEBRAUCHSANLEITUNG. Es erwartet dich im Folgenden viel Nützliches und Trickreiches für den Gebrauch von  $\text{\LaTeX}$ . Mit dieser Ausarbeitung verfolge ich zwei Ziele; sie soll als Einstieg in  $\text{\LaTeX}$  funktionieren, gleichzeitig aber auch als Nachschlagewerk und Kuriositätensammlung dienen. Ich werde dafür ganz pragmatisch vorgehen und auf jederlei Theorie und Philosophie verzichten. Wer sich für die Geschichte und Entwicklung von  $\text{\LaTeX}$  oder Textsatz im allgemeinen interessiert, wird im Literaturverzeichnis fündig. Für den Einstieg ist es wahrscheinlich am Sinnvollsten, die Beispiele noch einmal selbst in eine Experimentaldatei abzutippen, abzuändern und die Grenzen des Compilers auszutesten.  $\text{\TeX}$  wird übrigens wie „Tech“ ausgesprochen, da sich das von der griechischen Schreibweise  $\tau\epsilon\chi$  ableitet, ... meint КОПКА [1].

*Aber jetzt ersteinmal ... tatatataaaaaa! ... das erste Release!*

### Wie erstellt man ein Dokument?

1. Quelltext in eine Text-Datei mit der Endung `.tex` schreiben, am besten in einem geeigneten Editor, z. B. WinEdt unter Windows, Kile unter KDE,  $\text{\TeX}$ Shop unter Mac OS X.
2. Die  $\text{\TeX}$ -Datei dann zu einer DVI-Datei kompilieren:  $\text{\LaTeX}$ -Button im Editor oder mit dem Befehl "`latex <dateiname>`" in der Konsole; DVI ist die Abkürzung für *device independent*.
3. Die DVI-Datei kann man sich mit einem DVI-Viewer (YAP!, kdvi, xgdvi) ansehen.
4. Alternativ kann man die DVI-Datei dann weiter in eine PS- oder PDF-Datei umwandeln. Oder man lässt mit `pdflatex` eine PDF-Datei direkt aus dem Quelltext erstellen.

Der Quelltext enthält den Text, den man formatieren will, sowie alle dazu nötigen  $\text{\LaTeX}$ -Befehle, welche beispielsweise sagen *Hier Leerzeile einfügen!*, *Überall Schriftgröße 11 setzen!* oder auch *Ab jetzt alles fett drucken!*. Die DVI-Datei/PDF-Datei/PS-Datei ist dann das fertige Dokument, so wie man es zum Lesen, Ausdrucken und Herzeigen braucht.  $\text{\LaTeX}$  funktioniert also im Prinzip genauso, wie zu Fuß eine HTML-Seite bauen.

In der Datei `vorlage.tex` findest du einige brauchbare Vorspannzeilen (die sogenannte Präambel), welche im Wesentlichen das Seitenlayout festlegen. Zwischen `\begin{document}` und `\end{document}` schreibt man dann den eigentlichen Text, für den man ursprünglich mal ein Textprogramm gesucht hat, und die  $\text{\LaTeX}$ -Befehle. Dabei gilt die folgende Syntax:

## Die Syntax

- ◇ **Groß- und Kleinschreibung** wird grundsätzlich unterschieden!
- ◇ **Normaler 08/15-Text**, zwischen `\begin{document}` und `\end{document}` geschrieben, wird eins zu eins ausgegeben. Jedenfalls fast. Es werden dabei Zeilenumbrüche ignoriert und mehrere Leerzeichen hintereinander auf genau ein Leerzeichen gekürzt. (Ein Leerzeichen, auf das besonders hingewiesen sei, wird durch `\`  symbolisiert.)

der Quelltext `Ich\bin\ein\08/15-Text.`

erzeugt Ich bin ein 08/15-Text.

08/15-Text = Buchstaben  $\cup$  Leerzeichen  $\cup$  { 1234567890!;.,'""+-\*=@\$/()| } }

- ◇ Ein Wort, das mit einem `\` (Backslash, Alt Gr + ?) beginnt, ist für  $\text{\LaTeX}$  schon kein normales 08/15-Wort mehr. Vielmehr kennzeichnet der **Backslash** einen  $\text{\LaTeX}$ -**Befehl** oder eine  $\text{\LaTeX}$ -**Variable**. Befehle selbst enthalten nur Buchstaben und Symbole, aber niemals Ziffern.

der Quelltext `Es gibt einen Befehl für das tolle \LaTeX -Symbol.`

erzeugt Es gibt einen Befehl für das tolle  $\text{\LaTeX}$ -Symbol.

- ◇ Wie du siehst, werden **Leerzeichen** hinter Befehlen grundsätzlich ignoriert. Man kann Leerzeichen aber mit `~` (für einen großen Abstand) oder `\`  (für einen normalen Abstand) erzwingen. Oder mit `{}` den Befehl für definitiv zu Ende erklären.

der Quelltext `\LaTeX\rangeklitscht,\LaTeX~\weit\weg,\LaTeX\sehr\gut,\LaTeX{\}\sehr\gut`

erzeugt  $\text{\LaTeX}$ rangeklitscht,  $\text{\LaTeX}$  weit weg,  $\text{\LaTeX}$  sehr gut,  $\text{\LaTeX}$  sehr gut

- ◇ Benötigt ein Befehl **zwingende Argumente**, so übergibt man diese innerhalb von geschweiften Klammern `{ }` direkt hinter diesem Befehl.

der Quelltext `\c{x} \H{x} \H{o} \r{a}` erzeugt

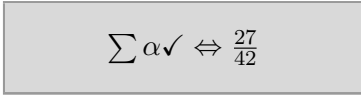
$\text{\c{x}}$   $\text{\H{x}}$   $\text{\H{o}}$   $\text{\r{a}}$

- ◇ In eckigen Klammern [ ] stehen **optionale Argumente** in der Form `\befehl[optional]{zwingend}`, das heißt der Befehl würde auch so `\befehl{zwingend}` funktionieren, wenngleich etwas anderes tun.
- ◇ **Geschweifte Klammern** selbst sind auch fast L<sup>A</sup>T<sub>E</sub>X-Befehle, und kein normaler Text. Sie müssen **immer paarweise** verwendet werden und können beliebige Umgebungen definieren, auf deren Bereich dann die Wirkung von gewissen Befehlen beschränkt ist.

der Quelltext `ottos {mops \LARGE kotzt} ogottogott`  
 erzeugt 

Beachte: in diesem Beispiel hat der Befehl `\LARGE` keine Argumente. `\LARGE` verstellt nur die Schriftgröße ab dem Punkt seines Auftretens bis zum Ende seines Wirkungsbereiches (die Klammer `}`).

- ◇ Ein Dollarzeichen \$ öffnet und beendet jeweils **die Matheumgebung**, was zunächst für die Benutzung der vielen Symbole, die WinEdt/Kile/etc. anzeigen, wichtig ist.

der Quelltext `$$\sum \alpha \checkmark \Leftrightarrow \frac{27}{42}$$`  
 erzeugt 

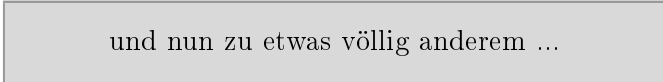
- ◇ **Umgebungen** braucht man für spezielle Layouteinheiten, in denen eigene Befehle wirken, die außerhalb der Umgebung völlig unbekannt sind bzw. andere Bedeutung haben. Umgebungen werden mit `\begin{umgebung}` `\end{umgebung}` geöffnet und geschlossen. Zum Beispiel steht `\begin{tabular}` am Anfang und `\end{tabular}` am Ende einer Tabelle.

Soweit ich weiß (aus dem Buch, in dem ich die folgende Liste gerade gefunden habe ☺), gibt es u.a. folgende Umgebungen, die ich teilweise auch noch vorstellen werde:

<code>abstract</code>	<code>eqnarray</code>	<code>flushright</code>	<code>picture</code>	<code>tabular*</code>
<code>appendix</code>	<code>eqnarray*</code>	<code>fussypar</code>	<code>quotation</code>	<code>thebibliography</code>
<code>array</code>	<code>equation</code>	<code>itemize</code>	<code>quote</code>	<code>theindex</code>
<code>center</code>	<code>figure</code>	<code>letter</code>	<code>sloppypar</code>	<code>titlepage</code>
<code>description</code>	<code>figure*</code>	<code>list</code>	<code>tabbing</code>	<code>trivlist</code>
<code>displaymath</code>	<code>filecontents</code>	<code>lrbox</code>	<code>table</code>	<code>verbatim</code>
<code>document</code>	<code>filecontents*</code>	<code>math</code>	<code>table*</code>	<code>verbatim*</code>
<code>enumerate</code>	<code>flushleft</code>	<code>minipage</code>	<code>tabular</code>	<code>verse</code>

Im allgemeinen sind die Sternchen-Umgebungen einfach Erweiterungen von den ohne-Sternchen-Varianten. Es gibt übrigens auch Befehle mit Sternchen, einige davon sind beispielsweise „energischer“, das heißt erzwingen Formatierungen, wo der ohne-Sternchen-Befehl schon längst zu Gunsten von ästhetischen Layoutansprüchen aufgegeben hätte.

- ◇ Alles, was hinter einem %-Zeichen steht, ist bis zum Zeilenende **Kommentar** und wird nicht kompiliert.

der Quelltext `und nun zu etwas völlig anderem ... % cooler Spruch`  
 erzeugt 

## Die Vorspannzeilen

```
..... vorlage.tex
\documentclass[a4paper,11pt]{article}

\usepackage{amssymb,amsmath,amsthm,amsfonts}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin9]{inputenc}

\parindent0cm
\pagestyle{empty}
\textheight23cm \textwidth16cm
\topmargin-2cm \oddsidemargin0cm

\begin{document}
\end{document}
.....
```

```
\documentclass[a4paper,11pt]{article}    DIN A4, Standardgröße aus {10pt, 11pt, 12pt};
                                         andere Klassen, die man laden könnte wären
                                         report, book, letter, proc, slides

\usepackage{ams...}                    mathematische Sondersymbole, z. B. \mathbb{N}

\usepackage[ngerman]{babel}            deutschsprach. Stil (Umlaute etc.) nach neuer deutscher
                                         Rechtschreibung, mit german statt ngerman hat man die alte

\usepackage[T1]{fontenc}               wählt die aktuelleren T1-Schriftarten aus; könnte aber auch
                                         weggelassen werden

\usepackage[latin9]{inputenc}          Umlaute und andere Nicht-ASCII-Zeichen
```

Für **SUSE** oder fast alle anderen Distribution, die als Standard den **UTF-8-Zeichensatz** verwendet (also Unicode): Nicht `latin9` laden! Damit auch LaTeX Unicode versteht, muss das Paket `latex-ucs` installiert werden (standardmäßig nicht der Fall) und stattdessen folgendes geladen werden: `\usepackage{ucs}`

`\usepackage[utf8]{inputenc}`

Informationsquelle: <http://www.kofler.cc/forum/forumthread.php?rootID=3368>

```
\parindent0cm                          setzt den Abstand auf 0 cm, um den das erste Wort im Absatz
                                         eingerückt wird

\pagestyle{empty}                       empty    ohne Kopf und Fußzeile, keine Seitennummern
                                         plain    Kopf leer, mit Seitennummern – ist Standard
                                         falls \pagestyle nicht im Vorspann auftaucht

\textheight23cm                         Höhe des Textblockes

\textwidth16cm                          mmmh, rat mal!

\topmargin-2cm                          oberer (Standard-)Textrand wird um -2 cm nach unten verschoben

\oddsidemargin0cm                       linker (Standard-)Textrand wird um 0 cm nach rechts verschoben
```



## Kapitel 2

# Textformatierungen

### Besondere Zeichen

<b>vorbelastete Zeichen:</b>	<code>\\$</code>	<code>\#</code>	<code>\%</code>	<code>\_</code>	<code>\&amp;</code>	<code>\{</code>	<code>\}</code>	
	\$	#	%	_	&	{	}	
<b>Anführungszeichen:</b>	<code>\glq</code>	<code>\glqq</code>	<code>\flq</code>	<code>\flqq</code>	<code>\grq</code>	<code>\grqq</code>	<code>\frq</code>	<code>\frqq</code>
	,	”	<	«	‘	“	>	»
	wegen <code>[ngerman]</code>	“	”	“	“	“	»	»
	auch kürzer:	“	”	“	“	“	»	»
<b>Striche:</b>	-	-	Trennstrich, für Worttrennungen (macht L <sup>A</sup> T <sub>E</sub> X automatisch)					
	--	—	Bindestrich, z. B. für Bereichsangaben (3–8); deutscher Gedankenstrich, z. B. für Texte					
	---	—	englischer Gedankenstrich, z. B. für Texte					
	\$-\$	—	ein Mathe-Minus					

### Umbrüche

- ♠ **Zeilenbruch** mit `\\` oder `\\[5mm]` oder `\linebreak` kann man Zeilen umbrechen. Mit `\linebreak` wird die Zeile vor dem Umbruch rechts- und linksbündig gesetzt.
- ♠ **Neuer Absatz** mit `\par` oder `\par\smallskip` oder `\par\medskip` oder `\par\bigskip`  
Besonderheiten: 1.) `\par\par` wirkt wie `\par`, 2.) eine Leerzeile im Quellcode (also 2 mal Enter) wird als `\par` kompiliert, was mitunter zu schwer auffindbaren Fehlern führen kann.
- ♠ **Seitenbruch** `\newpage` lässt den Rest der Seite hinter `\newpage` leer.  
`\pagebreak` passt die Seite optisch an, durch Strecken von Absatzabständen etc.
- ♠ **Abstände** `\vspace{1cm}` `\hspace{1cm}` vertikaler und horizontaler Abstand.  
`\vspace*{1cm}` `\hspace*{1cm}` werden im Gegensatz zu ohne Sternchen auf jeden Fall ausgeführt, zum Beispiel auch nach einem Seitenbruch. Sparsam verwenden.
- ♠ **Sequenzen** `\hfill` `\vfill` füllen den Platz bis zum Rand auf.  
`\dotfill` füllt den Platz mit Punkten und `\hrulefill` mit einem langen Strich.

```

Zeilenumbruch\\
Zeilenumbruch mit 5mm\\[5mm]
Neuer Absatz mit \texttt{\bs par} fügt keinen Abstand ein\par
%%%\bs erzeugt einen Backslash, selbstdefinierter Befehl, siehe vorlage.tex
Neuer Absatz mit \texttt{\bs par\bs smallskip}
           fügt einen elastischen kleinen Zwischenraum ein\par\smallskip
Neuer Absatz mit \texttt{\bs par\bs medskip}
           fügt einen elastischen mittelgroßen Zwischenraum ein\par\medskip
Neuer Absatz mit \texttt{\bs par\bs bigskip}
           fügt einen elastischen großen Zwischenraum ein\par\bigskip
Man kann auch mehrfach skip-en, z.B. so: \texttt{\bs par\bs
           bigskip\bs smallskip}\par\bigskip\smallskip
\hspace{1cm} \texttt{\bs texttt{\}}
           ruft übrigens die Schriftart Typewriter auf \par
\vspace{0.5cm} 1 \dotfill 2 \hfill 3 \hrulefill 4

```

```

Zeilenumbruch
Zeilenumbruch mit 5mm

Neuer Absatz mit \par fügt keinen Abstand ein
Neuer Absatz mit \par\smallskip fügt einen elastischen kleinen Zwischenraum ein
Neuer Absatz mit \par\medskip fügt einen elastischen mittelgroßen Zwischenraum ein
Neuer Absatz mit \par\bigskip fügt einen elastischen großen Zwischenraum ein

Man kann auch mehrfach skip-en, z.B. so: \par\bigskip\smallskip

\texttt{\}} ruft übrigens die Schriftart Typewriter auf

1 .....2
3 _____4

```

## Feste Maße

Wie bei den Umbrüchen gerade benutzt, setzen sich Maßangaben aus einem möglichen Vorzeichen + oder - (optional) einer Zahl (zwingend) und danach der Maßeinheit (zwingend) zusammen, z. B. -2cm. Dezimalzahlen können mit Komma oder mit Punkt geschrieben werden.

<b>Einheiten:</b>	mm	Millimeter	pt	Punkte (1 in = 72.27 pt)
	cm	Zentimeter	pc	Picas (1 pc = 12 pt)
	in	Inch (Zoll = 2.54 cm)	bp	Big Point (1 in = 72 bp)
	dd	Didot (1157 dd = 1238 pt)	cc	Cicero (1 cc = 12 dd)
	sp	Scaled Point (1 pt = 65536 sp)		
	em	Breite des Gedankenstrichs — im aktiven Zeichensatz		
	ex	Höhe des kleinen x im aktiven Zeichensatz		

<b>Befehle für waage-</b>	\,	\:	\;	\!	\quad	\quad	\/	\_	\@	~
<b>rechte Abstände:</b>	$\frac{3}{18}$ em	$\frac{4}{18}$ em	$\frac{5}{18}$ em	$\frac{-3}{18}$ em	1em	2em	wahrsch. dynamisch			

## Elastische Maße

Mit dem Befehl `\setlength{Längenvariable}{Wert}` kann man einer definierten Längenvariable einen neuen Wert zuordnen. Beispiel für eine Längenvariable ist `\parskip`, welche den Abstand zwischen zwei Absätzen festlegt.

Man könnte diesen Abstand zum Beispiel mit `\setlength{\parskip}{2ex}` auf den festen Wert `2ex` umdefinieren. Besser ist, man wählt ein elastisches Maß `\setlength{\parskip}{2ex plus 1ex minus 0.5ex}`, damit L<sup>A</sup>T<sub>E</sub>X die Seite so schön wie möglich einrichten kann ☺.

## Schriftgröße

Die Schriftgröße kann, ausgehend von der in `\documentclass[11pt]{article}` eingestellten Größe `11pt` als „normaler“ Schriftgröße, mit den folgenden Befehlen verändert werden:

<code>\tiny</code>	Schrift ist Winzig	<code>\Large</code>	Schrift ist Größer
<code>\scriptsize</code>	Schrift ist in Indexgröße	<code>\LARGE</code>	Schrift ist Noch Größer
<code>\footnotesize</code>	Schrift ist in Fußnotengröße	<code>\huge</code>	Schrift ist Riesig
<code>\small</code>	Schrift ist Klein	<code>\Huge</code>	Schrift ist Gigantisch
<code>\normalsize</code>	Schrift ist Normal		
<code>\large</code>	Schrift ist Groß		

Dabei wirkt solch ein Befehle wie ein „Schalter“ d.h. ohne Argumente, und bleibt ab seinem Auftreten solange gültig bis die Umgebung endet oder bis ein neuer, konkurrierender Befehl „umschaltet“.

In Ausnahmefällen kann es ganz nützlich<sup>1</sup> sein, die Größe der Schrift relativ zur aktuell eingestellten Schriftgröße  $\in \{\text{\tiny}, \text{\scriptsize}, \dots, \text{\Huge}\}$  ändern zu können. Dazu muss man jedoch `\usepackage{relsize}` im Vorspann ergänzen. Dann kann man auch mit `\larger` und `\smaller` bzw. `\relsize{n}` die Schriftgröße ändern.

```
{\tiny Die Schrift ist jetzt winzig, {doch das kann sich \huge schnell ändern.}
Ganz einfach.}\par
```

```
normal - \larger 1 Stufe größer - \larger 1 Stufe größer -
\larger 1 Stufe größer\par
```

```
\normalsize normal \relsize{5} fünf Stufen größer \relsize{-5} fünf Stufen
kleiner \relsize{-3} nochmal drei Stufen kleiner\par
```

Die Schrift ist jetzt winzig, doch das kann sich **schnell ändern.** Ganz einfach.

normal - 1 Stufe größer - 1 Stufe größer - 1 Stufe größer

normal **fünf Stufen größer** fünf Stufen kleiner nochmal drei Stufen kleiner

<sup>1</sup>Definiert man sich beispielsweise einen eigenen Befehl, innerhalb dessen man die Schriftgröße ändert, so sorgt man mit relativem Schriftgrößen-Wechseln dafür, dass der Befehl auch in allen Schriftgrößen verwendbar ist.

## Quelltext ausgeben

```
\begin{verbatim} \end{verbatim},
\begin{verbatim*} \end{verbatim*},
\verb+ + und \verb** +
```

sind Befehle, um Quelltext auszugeben. Die Befehle mit Stern stellen zusätzlich Leerzeichen durch `\_` dar. Die `verbatim`-Umgebungen setzen den Quelltext in einen neuen Absatz. Die `\verb`-Befehle geben den Quelltext in der gleichen Zeile aus, der Quelltext kommt dabei zwischen die beiden `+`-Zeichen; das `+`-Zeichen kann dabei auch durch ein beliebiges anderes Zeichen außer Stern und Leerzeichen ersetzt werden und wirkt für den Befehl wie eine Klammer.

Die `verbatim`-Umgebung und der `\verb`-Befehl können NICHT als Argument in irgendeinem anderen Befehl benutzt werden!!! Funktionieren aber wunderbar in Umgebungen à la `\begin{irgendwas} \end{irgendwas}`.

## Zeichensätze

Schriftfamilie, Schriftstärke und Schriftform kann man mit `\text{xx}` verändern. Wobei `xx` durch das jeweiligen Kürzel in der Tabelle ersetzt werden muss.

`\textsl{\textsf{Test}}` erzeugt z. B. *Test*.

Roman `rm`, medium `md` und aufrecht `up` sind die Standardeinstellungen.

Es sind nur die Kombinationen aus der Tabelle möglich.

Familie	Stärke	Form	Beispiel
<code>rm</code>	<code>md</code>	<code>up</code>	Roman medium aufrecht
<code>rm</code>	<code>md</code>	<code>sl</code>	<i>Roman medium geneigt</i>
<code>rm</code>	<code>md</code>	<code>it</code>	<i>Roman medium kursiv</i>
<code>rm</code>	<code>md</code>	<code>sc</code>	ROMAN MEDIUM KAPITÄLCHEN
<code>rm</code>	<code>bf</code>	<code>up</code>	<b>Roman fett aufrecht</b>
<code>rm</code>	<code>bf</code>	<code>sl</code>	<b><i>Roman fett geneigt</i></b>
<code>rm</code>	<code>bf</code>	<code>it</code>	<b><i>Roman fett kursiv</i></b>
<code>sf</code>	<code>md</code>	<code>up</code>	Sans medium aufrecht
<code>sf</code>	<code>md</code>	<code>sl</code>	<i>Sans medium geneigt</i>
<code>sf</code>	<code>bf</code>	<code>up</code>	<b>Sans fett aufrecht</b>
<code>tt</code>	<code>md</code>	<code>up</code>	Typewriter medium aufrecht
<code>tt</code>	<code>md</code>	<code>sl</code>	<i>Typewriter medium geneigt</i>
<code>tt</code>	<code>md</code>	<code>it</code>	<i>Typewriter medium kursiv</i>
<code>tt</code>	<code>md</code>	<code>sc</code>	TYPEWRITER MEDIUM KAPITÄLCHEN

Es gibt für das Umstellen der Zeichensätze auch „Schalter“-Befehle, welche (wie bei den Schriftgrößenbefehlen) zwischen den Zeichensätzen „hin- und herschalten“:

1. Engel:

```
\rmfamily \sffamily \ttfamily \mdseries \bfseries
\upshape \itshape \slshape \scshape \normalfont
```

2. Teufel:

```
\bf \rm \tt \sl \it \sc \sf
```

In Büchern wird empfohlen diese Befehle nicht mehr zu verwenden, da sie erstens aus alten Versionen stammen und nur noch aus Gutmütigkeit im „Kompatibilitätsmodus“ weitergeführt werden, und zweitens da sie sich nicht ineinander schachteln bzw. kombinieren lassen. Wenn man allerdings mal extrem schreibfaul ist (beispielsweise in Tabellenzellen: `&\it Text&` statt `&\textit{Text}&` oder `&\itshape Text&`, die Kaufmanns-Und wirken in Tabellen als Umgebungsbegrenzung) ...

`\large` Normaler Text wird zu `\textit{Kursivtext}` und dann weiter zu `\textbf{kursivem, fettem Text.}`\\

Jedoch nicht mit den `{\sf Zweibuchstaben}`-Schriftumschaltbefehlen.\\

Dort schaltet man von `rm` `{\it auf it` und von dort `{\bf auf bf um.}`\\

Deshalb muss man dann `\bf` eigentlich `\rm` auch `\tt` keine Klammern setzen.\\

`\normalfont` Die guten, langen, aktuellen Umschaltbefehle `\itshape` sind allerdings im Rahmen `\bfseries` ihrer Möglichkeiten ineinander schachtelbar.

Normaler Text wird zu *Kursivtext* und dann weiter zu ***kursivem, fettem Text.***

Jedoch nicht mit den Zweibuchstaben-Schriftumschaltbefehlen.

Dort schaltet man von *rm auf it und von dort **auf bf um.***

Deshalb muss man dann **eigentlich** auch keine Klammern setzen.

Die guten, langen, aktuellen Umschaltbefehle *sind allerdings im Rahmen ihrer Möglichkeiten ineinander schachtelbar.*

## Hervorhebungen

zu <code>\emph{markierender} Text</code>	}	ergeben beide	zu <i>markierender Text</i>
zu <code>{\em markierender} Text</code>			
zu <code>\underline{markierender} Text</code>		ergibt	zu <u>markierender Text</u>
zu <code>\fbox{markierender} Text</code>		ergibt	zu <span style="border: 1px solid black; padding: 2px;">markierender</span> Text
zu <code>\shabox{markierender} Text</code>		ergibt	zu <span style="border: 2px solid black; padding: 2px;">markierender</span> Text

Für die `shabox` muss jedoch noch das Paket `shadow` mit `\usepackage{shadow}` in die Vorspannzeilen eingebunden werden (siehe 3. Kapitel, Abschnitt RAHMEN).

## URL

Mit dem Paket `url` (d.h. `\usepackage{url}` in die Vorspannzeilen ergänzen) kann man beispielsweise Internetadressen unkompliziert darstellen.

`\url{http://www.impfgrimpf.de/~tanteemma}`

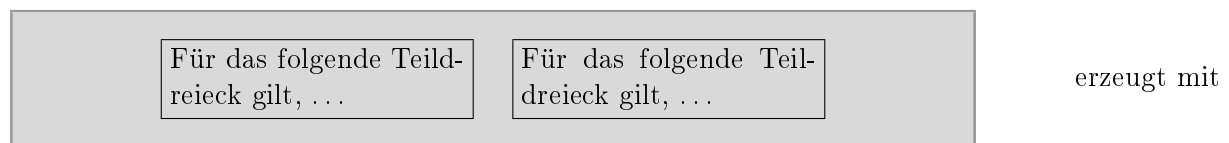
ergibt

<http://www.impfgrimpf.de/~tanteemma>

ACHTUNG: GEKLAUT !!! Die folgenden zwei Abschnitte übernehme ich bis auf „neue Rechtschreibung-“ und Layoutanpassungen vollständig aus dem Skript des L<sup>A</sup>T<sub>E</sub>X-Kurses vom Wurzel-Verein Jena.

## Silbentrennung

Der Trennalgorithmus, der in T<sub>E</sub>X verwendet wird, ist recht schnell, macht wenig Fehler und ist für viele Sprachen anwendbar. Nichtsdestotrotz gibt es Wörter, die falsch getrennt werden, bzw. die man nicht überall trennen sollte, wo es möglich wäre (z. B. Urinstink sollte nicht als Urin-stinkt getrennt werden)<sup>2</sup>. Um solche Fehler zu vermeiden, gibt es die Befehle \- und "-. ("- steht erst durch die Einbindung des Paketes **babel** zur Verfügung.)

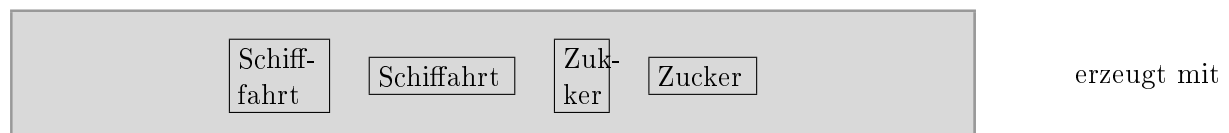


erzeugt mit

```
\fbox{\parbox{39mm}{Für das folgende Teildreieck gilt, \ldots}}
\quad
\fbox{\parbox{39mm}{Für das folgende Teil\ -dreieck gilt, \ldots}}
```

Mit \- kann man die Stellen angeben, an denen ein Wort getrennt werden darf und zwar nur an diesen. Mit "- kann man zusätzliche Trennstellen in einem Wort angeben.

Mit dem Befehl `discretionary{}{}{}` lassen sich mögliche Trennstellen angeben, bei denen sich [nach alter Rechtschreibung] Buchstaben verändern, z. B. Trennung von „ck“ als k-k [n. a. R.]<sup>3</sup> oder Wörter bei denen drei Konsonanten zusammentreffen [n. a. R.].



erzeugt mit

```
\fbox{\parbox{11mm}{Schi\discretionary{ff-}{f}{ff}ahrt}}\quad
\fbox{\parbox{17mm}{Schi\discretionary{ff-}{f}{ff}ahrt}}\quad
\fbox{\parbox{5mm}{Zu\discretionary{k-}{k}{ck}er}}\quad
\fbox{\parbox{12mm}{Zu\discretionary{k-}{k}{ck}er}}
```

Kommen in einem Dokument Wörter vor, welche mehrfach falsch getrennt werden, sollte man die Trennung dieser Wörter in der Präambel festhalten. Dazu gibt es den Befehl `\hyphenation{}`. Beispiel: `\hyphenation{Teil-drei-eck Teil-drei-ecke Ur-instinkt}`

<sup>2</sup>oder Spar-gelder (besser als Spargel-der), Blut-egel (besser als Blute-gel), be-inhalten (besser als bein-halten)

<sup>3</sup>n. n. R. Die Konsonantenverbindungen **ch**, **ck**, **sch**, in Fremdwörtern auch **ph**, **rh**, **sh**, **th**, bleiben ungetrennt, wenn sie für einen einfachen Laut stehen: Bü-cher, Zu-cker, ba-cken, Fla-sche, Ma-che-te, Pro-phet, Myr-rhe, Ca-shew-Nuss, ka-tho-lisch

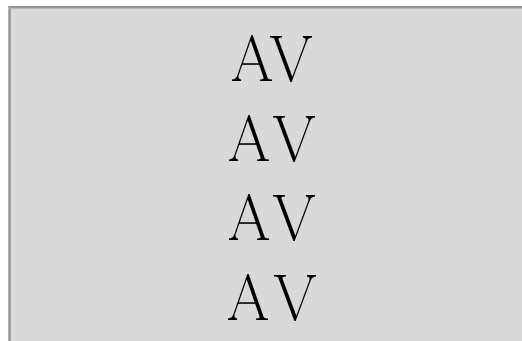
## Ligaturen und Kerning

Im Buchsatz hat es sich eingebürgert, gewisse Buchstaben ineinander zu schieben, um das Schriftbild zu verbessern. Würde man die Buchstaben „A“ und „V“ normal nebeneinander setzen, würde ein zu großer Zwischenraum entstehen.

Dieses Zusammenschieben nennt man *kerning* und T<sub>E</sub>X kümmert sich selbstständig darum.

Zum Verhindern dieses automatischen Kernings ist `\/` der Originalbefehl. Dagegen gilt `"|` speziell für die deutsche Umgebung (`babel`-Paket); es ermöglicht gleichzeitig eine Silbentrennung an dieser Stelle.

`\Huge AV \par A{}V \par A\/V \par A"|V`  
erzeugt folgende Ausgabe:



Wichtiger sind die sogenannten *Ligaturen*. Dabei werden bestimmte Buchstabenkombinationen zu einem Zeichen zusammengefasst.

mit Ligaturen:	ff	fi	fl	ffl	ffi	erzeugt durch
ohne Ligaturen:	ff	fi	fl	ffl	ffi	

```
mit Ligaturen: \quad {\Huge ff \quad fi \quad fl \quad ffl \quad ffi}
\par\medskip ohne Ligaturen: \quad
{\Huge f{}f \quad f{}i \quad f{}l \quad f{}f{}l \quad f{}f{}i}
```

Beim Satz von Ligaturen ist folgende Regel anzuwenden (siehe Duden, Vorschriften für den Satz):

*Ligaturen dürfen die sprachliche Richtigkeit nicht stören; die zu einer Ligatur vereinigten Buchstaben müssen im Wortstamm zusammengehören. Bei Endungen auf -fig und -fisch wird dagegen die Ligatur **fi** benutzt. Beispiele: hoffen, Pffiff, Pflaume*

*Ligaturen dürfen nicht angewendet werden, wenn die Buchstaben, die eine Ligatur bilden können, in Zusammensetzungen unmittelbar vor oder hinter der Wortfuge stehen. Beispiele: auffangen, begrifflich, Briefinhalt*

Um solche verbotenen Ligaturen aufzulösen, setzt man an die entsprechende Stelle mit `{}` eine leere Gruppe, z. B. `Brief{}inhalt`, oder benutzt den `"|`-Befehl aus dem `babel`-Paket, z. B. `Brief"|inhalt`.

## Akzente

Akut	á é ó ẋ	\'a} \'e} \'o} \'x}
Brevis	ă ă ǫ ǫ	\u{a} \u{e} \u{o} \u{x}
Cedille	ç ę ȳ ȳ	\c{a} \c{e} \c{o} \c{x}
Gravis	à è ò x̂	\'a} \'e} \'o} \'x}
Háček	ǎ ě ǒ x̣	\v{a} \v{e} \v{o} \v{x}
Makron	ā ē ō x̄	\={a} \={e} \={o} \={x}
Ogonek	ą ę ȳ ȳ	\k{a} \k{e} \k{o} \k{x}
Punktakzent	á é ó ẋ	\.a} \.e} \.o} \.x}
Ringakzent	â ê ô x̆	\r{a} \r{e} \r{o} \r{x}
Tilde	ã ã õ x̃	\~{a} \~{e} \~{o} \~{x}
Trema	ä ë ö ẍ	\"a} \"e} \"o} \"x}
Ungarischer Doppelakut	á ě ó ẍ	\H{a} \H{e} \H{o} \H{x}
Unterpunktakzent	ạ ẹ ọ ọ	\d{a} \d{e} \d{o} \d{x}
Unterstreichungsakzent	ā ē ō x̄	\b{a} \b{e} \b{o} \b{x}
Verbindungsakzent	āē ōx̄	\t{ae} \t{ox}
Zirkumflex	â ê ô x̆	\^a} \^e} \^o} \^x}
i,j ohne Punkt	ı ȷ	\i \j
Akzente auf i,j	ı̇ ȷ̇	\H{j} \v{i} \H{j} \v{i}
Sonderzeichen	ø Æ Å œ €	\o \O \aa \AA \oe \OE
Sonderzeichen	æ Æ Æ ℒ ℒ ℒ ℒ	\ae \AE \ss \l \L ! ' ?



## Kapitel 3

# Textanordnung

### Aufzählung und Nummerierung

`\begin{itemize} \end{itemize}` ..... Umgebung für Aufzählung  
`\begin{enumerate} \end{enumerate}` ..... Umgebung für Nummerierung  
`\begin{description} \end{description}` ..... beliebiger Text als Markierung  
`\item` ..... die Markierung für einen neuen Unterpunkt (innerhalb dieser 3 Umgebungen)  
`\item[zeich]` ..... `zeich` kann aus beliebigen Zeichen bestehen, ersetzt Standardmarkierung

So sieht normaler Text aus.

- Der Punkt ist Standard ...
  - Unterpunkte werden durch ineinandergeschachtelte Umgebungen erzeugt.
  - Dabei werden der Reihe nach diese Zeichen genutzt: • – \*

♣ ... aber nicht Pflicht

1. erste Stufe: arabische Ziffern

- (a) zweite Stufe: Kleinbuchstaben
  - i. danach: römische Ziffern
  - ii. kleine römische Ziffern !

**erstens** Hier ist das optionale Argument nach `item` unbedingt nötig!

**zweitens** Kann man das hier z.B. für ein Literaturverzeichnis verwenden.

Hier der Quelltextwust:

So sieht normaler Text aus.

```

\begin{itemize} \item Der Punkt ist
Standard \ldots \begin{itemize}
\item Unterpunkte werden durch ineinander
geschachtelte Umgebungen erzeugt.
\item Dabei werden der Reihe nach
diese Zeichen genutzt: \quad $\bullet$
-- $\ast$ $\cdot$ \end{itemize}
\item[$\clubsuit$]\ldots\ aber nicht
Pflicht \end{itemize} \begin{enumerate}
\item erste Stufe: arabische Ziffern
\begin{enumerate}\item zweite Stufe:
Kleinbuchstaben \begin{enumerate}
\item danach: römische Ziffern
\item kleine römische Ziffern!
\end{enumerate} \end{enumerate}
\end{enumerate} \begin{description}
\item[erstens] Hier ist das optionale
Argument nach \texttt{item} unbedingt
nötig! \item[zweitens] Kann man das
hier z.B. für ein Literaturverzeichnis
verwenden. \end{description}
  
```

## Ausrichtung

<code>\begin{center}</code> <code>\end{center}</code>	zentriert zeilenweise, Zeilen mit <code>\</code> oder z.B. <code>\\[abst]</code> (z.B. <code>abst=1ex</code> ) trennen, fügt vor und nach der <code>\center</code> -Umgebung extra Zwischenraum ein, keine Worttrennung am Zeilenende	Erkennungsdienstliche Behandlung  Das ist nicht Dante  Das ist eine Photographie von Dante.  Das ist ein Film, in dem ein Schauspieler auftritt, der vorgibt Dante zu sein.
<code>\centering</code>	zentriert alles innerhalb der Umgebung, in der <code>\centering</code> auftritt	Das ist ein Film, in dem Dante Dante spielt.
<code>\centerline{}</code>	einzelne Zeile zentrieren, als Argument dem Befehl übergeben	Das ist ein Mann, der von Dante träumt.
<code>\begin{flushleft}</code> <code>\end{flushleft}</code> <code>\begin{flushright}</code> <code>\end{flushright}</code>	linksbündig bzw. rechtsbündig, Zeilen mit <code>//</code> trennen, keine Worttrennung am Zeilenende, zusätzlicher Abstand vor/nach der Umgebung	Das ist ein Mann,  der Dante heißt,  aber nicht Dante ist.
<code>\raggedright</code> <code>\raggedleft</code>	<code>ragged = zerlumpt/struppig</code> , wirkt innerhalb einer Umgebung	Das ist ein Mann, der Dante nachäfft. Das ist ein Mann, der...
<code>\begin{quote}</code> <code>\end{quote}</code> <code>\begin{quotation}</code> <code>\end{quotation}</code>	rückt den Text beidseitig ein, in <code>quote</code> wird nach Absätzen (nach <code>\par</code> ) zusätzlichen Abstand eingefügt, in <code>quotation</code> wird das erste Wort im Absatz eingerückt (durch <code>\parindent0cm</code> sonst immer unterdrückt)	Das ist ein Mann, der sich für Dante aus gibt. Das ist ein Mann, der träumt, er sei Dante. Das ist ein Mann, der träumt, er sei Dante.  Das ist ein Mann, der Dante zum Verwechseln ähnlich sieht.
<code>\begin{verse}</code> <code>\end{verse}</code>	beidseitiges einrücken von Gedichten, Reimen, Versen, ... <code>\</code> bricht Zeilen um, bei zu langer Zeile wird nochmals eingerückt	Das ist eine Wachsfigur von Dante. Das ist ein Wechselbalg, ein Zwilling, ein Doppelgänger.

## Tabulatoren

`\begin{tabbing}` `\end{tabbing}` erlaubt es, innerhalb dieser Umgebung Tabulatoren zu setzen und zu nutzen. Zeilen werden innerhalb der `tabbing`-Umgebung mit `\\` getrennt. `tabbing`-Umgebungen können nicht ineinander geschachtelt werden.

- `\=` Setzt einen Tabstop.  
Mehrere Tabstops kann man sich als von links nach rechts durchnummeriert denken, wobei der 0. Tabstop der normale Zeilenanfang ist. Geg.: Seien schon  $n$  ( $n \geq 0$ ) Tabstops gesetzt und man ist hinter dem  $m$ -ten Tabstop ( $0 \leq m \leq n$ ). Ges.: Was, wenn ich einen neuen Tabstop mit `\=` definiere? Antw.: wenn  $m = n$ : ein  $(m + 1)$ -ter Tabstop wird gesetzt, sonst: der  $(m + 1)$ -te Tabstop wird überschrieben.
- `\>` Springt im Text zum nächsten gesetzten Tabstop, ohne Rücksicht auf Verluste.
- `\kill` Kann anstelle von `\\` stehen und unterdrückt die vorangegangene Zeile.
- `\+` Setzt den Zeilenbeginn allgemein um einen Tabstop nach vorn (rechts).
- `\-` Setzt den Zeilenbeginn allgemein um einen Tabstop zurück (nach links).
- `\<` Springt zum vorherigen gesetzten Tabstop.
- `\'` Setzt den davor stehenden Text mit einem kleinen Abstand (Veränderung dieses Abstandes z. B. mit `\setlength{\tabbingsep}{1cm}`) rechtsbündig zum augenblicklichen Tabstop.
- `\‘` Richtet alles folgende rechtsbündig zum Seitenrand aus; dahinter keine weiteren Tabstop-Befehle setzen!

Hier ein paar Beispiele mit Tabulatoren:

Das	erste	<code>hspace</code>	hat einen Stern,
um zu erzwingen,	was <code>\LaTeX</code> nicht freiwillig	tut	nämlich
SORRY!,	... was <code>\LaTeX</code> nicht freiwillig	tut	nämlich
		Leerraum am Zeilenbeginn.	

Die Befehle `\-`, `\=`, `\'` und `\‘` haben außerhalb von `tabbing` andere Bedeutungen.

Der erste ist die selbst definierte Worttrennung und die anderen erzeugen Akzente auf Vokalen.

In `tabbing` werden diese Akzente bspw. auf einem o mit `\a={o}`, `\a' {o}` und `\a' {o}` erzeugt.

So, nun beginnt diese Zeile wieder ganz normal am Anfang.

Tabstop	1	ist	neu definiert!
Tabstop	4	ist	ganz neu.
		linke Seite	rechte Seite
			und ganz am Rand.

```
\begin{tabbing} Hier ein paar Beispiele mit Tabulatoren: \\
\hspace*{4cm} \= \hspace{4cm} \= \hspace{4cm} \= Ällibätsch \kill
Das \> erste \> \texttt{hspace} \> hat einen Stern, \\
um zu erzwingen, \> was \LaTeX nicht freiwillig tut \> nämlich \\
SORRY!, \> \ldots was \LaTeX nicht freiwillig tut \> \> nämlich \\
\> \> Leerraum am Zeilenbeginn.\\[1ex]
\+ Die Befehle \texttt{\bs -}, \texttt{\bs =}, \texttt{\bs ' }
und \texttt{\bs ‘} haben außerhalb von \texttt{tabbing} andere Bedeutungen. \\
Der erste ist die selbst definierte Worttrennung und die anderen \\
erzeugen Akzente auf Vokalen. \\
\< In \texttt{tabbing} werden diese Akzente bspw. auf einem o mit \texttt{\bs a={o}},
\texttt{\bs a' {o}} und \texttt{\bs a' {o}} erzeugt.\- \\[1ex]
So, nun beginnt \= diese Zeile wieder ganz normal am Anfang.\\
Tabstop \> 1 \> ist \> neu definiert! \= \\ Tabstop \> 4 \> ist \> ganz \> neu. \\[1ex]
\> \> linke Seite \\' rechte Seite \‘ und ganz am Rand. \\[-5ex] \end{tabbing}
```

## Tabellen

Eine weitere Umgebung, die es erlaubt Text tabellenartig anzuordnen, ist die Umgebung

```
\begin{tabular} \end{tabular}.
```

Bei einfachen Beispielen kann man mit Tabellen (`tabular`) und Tabulatoren (`tabbing`) die gleichen Ergebnisse erzielen. Vorteil von `tabular` ist, dass sich Tabellen ineinander schachteln lassen und diese Umgebung insgesamt sehr viel funktionaler ist. Außerdem formatiert  $\text{\LaTeX}$  in der `tabular`-Umgebung an den Abständen noch sehr viel rum, wo es unter Layout-Aspekten günstig erscheint; `tabbing` ist eher dazu gedacht, exakt so zu formatieren, wie es angegeben wurde.

### Die Syntax einer einfachen Tabelle:

```
\begin{tabular}[pos]{spalten}
**** & **** & ... & **** \\
**** & **** & ... & **** \\
**** & **** & ... & **** \\
\end{tabular}
```

Die Sternchen stehen für einen beliebigen, normalerweise einzeiligen Text. Nach dem `&`-Zeichen wird der Text in der gleichen Zeile aber nächsten Spalte fortgeführt. Und `\\` beendet die Tabellenzeile, bei Bedarf kann der Abstand zur nächsten Zeile (wie immer) mit bspw. `\\[1ex]` variiert werden.

### Der zwingende Parameter `spalten`: ..... Spaltenanzahl und -formatierung

<code>l</code>	Inhalt der Spalte linksbündig.
<code>c</code>	Inhalt der Spalte zentriert.
<code>r</code>	Inhalt der Spalte rechtsbündig.
<code>p{breite}</code>	<code>breite</code> = feste Spaltenbreite. Die Spalte wird als <code>\parbox</code> formatiert, siehe Abschnitt <code>BOXEN</code> . Zeilenumbruch in dieser „Parbox“ natürlich nur mit <code>\newline</code> oder <code>\par</code> , denn <code>\\</code> würde die nächste Tabellenzeile beginnen!
<code>*{n}{typ}</code>	<code>typ</code> $\in \{l,c,r,p{breite}\}$ und $n \in \mathbb{N}$ . Abkürzung für <code>n</code> gleiche Spalten mit der Formatierung <code>typ</code> .
<code> </code>	Ein vertikaler Strich zwischen zwei Spalten.
<code>  </code>	Zwei dicht benachbarte vertikale Striche zwischen den beiden Spalten.
<code>@{text}</code>	Statt des normalen Spaltenzwischenraumes wird <code>text</code> gesetzt.

### Der optionale Parameter `pos`: ..... vertikale Positionierung zum Rest des Textes

<code>ohne</code>	Zentriert die Tabelle vertikal zum Text, der die Tabelle umgibt.
<code>t</code>	top. Obere Tabellenzeile auf gleicher Höhe zum Text, der die Tabelle umgibt.
<code>b</code>	bottom. Untere Tabellenzeile auf gleicher Höhe zum Text, der die Tabelle umgibt.

### Befehle im Tabellentext: ..... zum Erzeugen von Linien und verbundenen Spalten

<code>\hline</code>	Horizontale Linie über die gesamte Tabellenbreite. Darf nur vor der ersten Zeile oder direkt hinter <code>\\</code> verwendet werden.
<code>\cline{i-j}</code>	Horizontale Linie von Spalte <code>i</code> zu Spalte <code>j</code> . Darf ebenfalls nur vor der ersten Zeile oder direkt hinter <code>\\</code> verwendet werden. Es dürfen mehrere <code>\cline</code> -Befehle direkt hintereinander stehen.
<code>\vline</code>	Vertikale Linie über die Zeilenhöhe.
<code>\multicolumn{n}{typ}{text}</code>	Fasst die nächsten <code>n</code> Spalten zu einer Spalte der Formatierung <code>typ</code> zusammen und enthält <code>text</code>

```

erstens: \quad \begin{tabular}[t]{|l|l|}
\quad zweitens: \quad \begin{tabular}{c|c}
\quad drittens: \quad \begin{tabular}[b]{r{~::~}l}
calligraphy & \mathcal{T} \\
\hline \multicolumn{2}{|l|}{\textbf{Änderung von Tabellenvariablen (immer außerhalb
der Tabelle)}} \\
\hline \tt\bs setlength{\bs arrayrulewidth}\{0.5mm\} & Setzt {\tt
\bs arrayrulewidth} auf 0.5mm. \\
\hline \tt\bs arrayrulewidth0.5mm & Setzt auch
{\tt\bs arrayrulewidth} auf 0.5mm. \\
\hline \multicolumn{2}{|l|}{\textbf{Standardvariablen zur Formatierung von Tabellen}} \\
\hline {\tt\bs tabcolsep} & Halbe Breite des Zwischenraumes zwischen zwei benachbarten Spalten. \\
\hline {\tt\bs arrayrulewidth} & Dicke von vertikalen und horizontalen Tabellenlinien. \\
\hline {\tt\bs doublerulesep} & Abstand von Doppellinien. \\
\hline \multicolumn{2}{|p{15.5cm}|}{\arrayrulewidth0.3ex \centering} \\
\begin{tabular}[p{9cm}|}\hline \textbf{Zeilenabstand} & Mit \tt\bs
{renewcommand{\bs arraystretch}\{1.4\}} werden die Zeilenabstände um das 1.4-fache
gestreckt. \\
\hline \end{tabular} } \\
\hline \end{tabular} } \\

```

erstens:	einerlei	keins	zweitens:	schwarz	weiß	drittens:	bold : $\mathcal{T}$
	ab	und zu		♠	♥		calligraphy : $\mathcal{T}$
				♣	◇		
<b>Änderung von Tabellenvariablen (immer außerhalb der Tabelle)</b>							
\setlength{\arrayrulewidth}{0.5mm}				Setzt \arrayrulewidth auf 0.5mm.			
\arrayrulewidth0.5mm				Setzt auch \arrayrulewidth auf 0.5mm.			
<b>Standardvariablen zur Formatierung von Tabellen</b>							
\tabcolsep				Halbe Breite des Zwischenraumes zwischen zwei benachbarten Spalten.			
\arrayrulewidth				Dicke von vertikalen und horizontalen Tabellenlinien.			
\doublerulesep				Abstand von Doppellinien.			
	<b>Zeilenabstand</b>	Mit \renewcommand{\arraystretch}{1.4} werden die Zeilenabstände um das 1.4-fache gestreckt.					

**weitere nützliche Umgebungen:**

- table**            \begin{table} *überschrift* \begin{tabular}... \end{tabular} *unterschrift* \end{table}  
Eine tabular-Umgebung wird genau an die Stelle seines Auftretens im Text gesetzt, falls zu lang wird eine neue Seite angefangen. Das table ist eine sogenannte gleitende Tabelle und wird an die nächst mögliche Stelle gesetzt, wobei der nachfolgende Text ggf. vorgezogen wird, um keine halbleeren Seiten zu hinterlassen.
- tabular\***        \begin{tabular\*}{breite}[pos]{spalten} ... \end{tabular\*}     Durch breite wird die Gesamtlänge der Tabelle festgelegt. Empfohlen bei spalten nach dem ersten Eintrag ein @{\extracolsep\fill} einzufügen.
- longtable**       Unterstützt Tabellen, die länger als eine Seite sind.
- supertabular**    Wie longtable.
- dcolumn**        Richtet nach Dezimalpunkt von Dezimalzahlen aus.

Noch mehr Tabellen gibt es auf [www.dante.de](http://www.dante.de) und wenn man einfach im Internet danach sucht.

**Boxen**

Boxen sind Kästen, die beliebigen Text enthalten und als eine Einheit betrachtet zum restlichen Text angeordnet werden.

**Einzeilige Boxen (genannt: LR-Boxen)**

<code>\mbox{text}</code>	Box mit Text. Breite vom Text bestimmt.
<code>\makebox[breite][pos]{text}</code>	<ul style="list-style-type: none"> <li>• <code>pos</code> <math>\in \{l, r, s\}</math>, <code>l</code> linksbündig, <code>r</code> rechtsbündig, <code>s</code> stretched, <i>ohne</i> zentriert</li> <li>• <code>breite</code> hier und im folgenden eine Maßangaben, d.h. Zahl mit Einheit, z. B. <code>1cm</code>, <code>0.5ex</code>, siehe 2. Kapitel</li> </ul>
<code>\fbox{text}</code>	Eingerahmte Box mit Text. Breite vom Text bestimmt.
<code>\framebox[breite][pos]{text}</code>	<code>pos</code> $\in \{l, r, s\}$ , <code>breite</code> Maßangaben
<code>\width</code> , <code>\height</code> , <code>\depth</code> <code>\totalheight</code>	Fertige Maße, welche die Breite, die Höhe über und unterhalb der Grundlinie und die Gesamthöhe des Textes zurückgeben.

<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid gray; padding: 2px;">Jena</div> <div style="border: 1px solid gray; padding: 2px;">Nordosten</div> </div> <p>des Thüringer Waldes am mittleren Lauf der Saale gelegen, <span style="border: 1px solid gray; padding: 2px;">verdankt</span> der <span style="border: 1px solid gray; padding: 2px;">Gründung</span> seiner Universität im Jahre 1548 eine bedeutende Entwicklung. ...</p>	<pre> \framebox[3\width]{Jena}, im \framebox[3.5cm][r]{Nordosten} des Thüringer Waldes am mittleren Lauf der Saale gelegen, \framebox[1.5\width][s]{verdankt der Gründung} seiner Universität \makebox[2\width][s]{im Jahre 1548} eine bedeutende Entwicklung. \ldots\par \parbox[t]{2cm}{\framebox[2mm]{zentriert}}\par \framebox[1mm][l]{links}\par \makebox[0mm][r]{rechts}} \parbox[t]{5.5cm}{Ist die Box zu klein, dann ragt der Text raus; das ist nützlich für Randnotizen.} </pre>
<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid gray; padding: 2px; margin-bottom: 2px;">zentriert</div> <div style="border: 1px solid gray; padding: 2px; margin-bottom: 2px;">links</div> <div style="border: 1px solid gray; padding: 2px;">rechts</div> </div>	<p>Ist die Box zu klein, dann ragt der Text raus; das ist nützlich für Randnotizen.</p>

**Einzeilige Boxen speichern**

- `\newsavebox{\boxname}` Führt einen neuen Box-Name ein. Keinen  $\LaTeX$ -Befehl nehmen!
- `\sbox{\boxname}{text}`  
`\savebox{\boxname}[breite][pos]{text}` Erzeugen/speichern beide eine Box mit Inhalt `text` unter dem Namen `\boxname`.
- `\begin{lrbox}{\boxname} text \end{lrbox}` Macht das gleich wie `\sbox` als Umgebung (wichtig, wenn man *verbatim* benutzen will).
- `\usebox{\boxname}` Druckt die Box an beliebige Stelle im Text.

**Einzeilige Boxen verschieben**

$$\text{\raisebox{lift}[oben][unten]{text}}$$

Erzeugt eine `\mbox`, die um `lift` (Längenmaß) oberhalb der momentanen Grundlinie liegt. Mit `oben` und `unten` kann man  $\LaTeX$  die Längen des „ungeliftet“ übergebenen Textes oberhalb und unterhalb der Grundlinie, unabhängig vom tatsächlichen Wert, übergeben.

Hat man mehrere optionale Parameter der gleichen Art, beispielsweise zwei Längen, möchte aber nur den zweiten auf `1cm` setzen, so übergibt man `[][1cm]`.

Hat man mehrere `\raisebox{1ex}{optionale}` Parameter der gleichen Art, beispielsweise zwei Längen, möchte aber nur den `\raisebox{-1ex}{zweiten}` auf `1cm` setzen, so übergibt man `{\ttfamily}[][1cm]`.

**Absatzboxen (genannt: Parbox)**

```
\parbox[pos][höhe][ipos]{breite}{text}
\begin{minipage}[pos][höhe][ipos]{breite} text \end{minipage}
```

Beide Befehle erzeugen eine Box mit gegebenenfalls mehrzeiligem Text. `Parbox` ist der allgemein bekanntere und kürzere Befehl. Die `Minipage` erlaubt jedoch mehr Textformatierungen, wie z. B. `verbatim`, Textausrichtung, Aufzählungen und Listen.

**pos** `pos`  $\in \{t, b, c\}$ , für top, bottom, centered. Kein Argument angeben heißt centered. Bezeichnet die Stelle der Box, die mit dem Nachbartext rechts und links von der Box auf gleicher Höhe ist.

**höhe** Legt die Höhe der Box explizit fest.

**ipos** Sollte beispielsweise die mit `höhe` festgelegte Boxhöhe größer sein als der Text ausfüllen kann, dann sagt `ipos`, wo der Text in der Box steht.  
`ipos`  $\in \{t, b, c, s\}$ , abgekürzt für top, bottom, centered, stretched.

**Balkenboxen**

```
\rule[lift]{breite}{höhe}
```

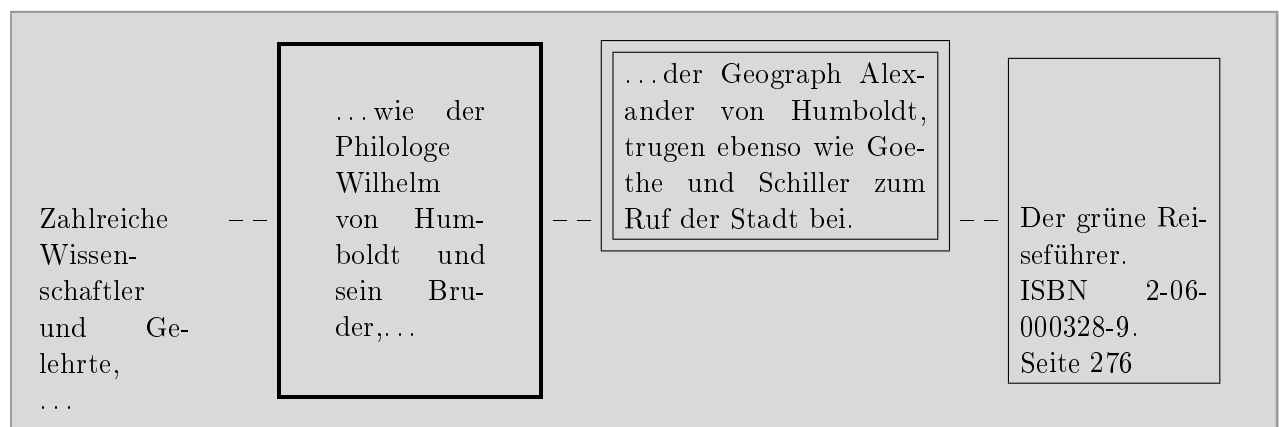
Erzeugt eine schwarze Box der Ausmaße `höhe` und `breite`, die um `lift` nach oben über die Grundlinie versetzt wurde. `\rule{2ex}{1ex}` erzeugt . Setzt man eine der beiden Ausdehnungen auf `0ex` so erhält man eine unsichtbare „Stütze“. Das ist z. B.

dafür  ganz nützlich  
 (Quelltext:`\fbox{\rule[-4mm]{0cm}{1cm}dafür}`).

**Box-Stilparameter**

`\fboxrule` Linienstärke der Boxen mit Rahmen. Standard: 0.4pt

`\fboxsep` Leerraum zwischen Rahmen und einbeschriebenem Text. Standard: 4pt



```
\parbox[t]{2cm}{Zahlreiche Wissenschaftler und Gelehrte, \ldots}
\quad-- -- \fboxsep20pt\fboxrule1.5pt
\fbox{\parbox{2cm}{\ldots wie der Philologe Wilhelm von Humboldt
und sein Bruder,\ldots}} -- -- \fboxsep4pt\fboxrule0.4pt
\fbox{\fbox{\parbox[b]{4cm}{\ldots der Geograph Alexander von
Humboldt, trugen ebenso wie Goethe und Schiller zum Ruf der Stadt bei.}}}
-- -- \fbox{\parbox[] [4cm] [b]{2.5cm}{Der grüne Reiseführer.\par
ISBN 2-06-000328-9.\par Seite 276}}
```

..... mehr dazu im Abschnitt RAHMEN

## Rahmen

### Das boxedminipage-Paket

Wie eine richtige minipage-Umgebung. Einstellungen mit `\fboxrule` und `\fboxsep`

```
\usepackage{boxedminipage}
\begin{boxedminipage}[t]{6cm}
Wie eine richtige ...
\end{boxedminipage}
```

### Das shadow-Paket

So ähnlich wie `\fbox` mit den Parametern `\sboxrule` (default 0.4pt), `\sboxsep` (default 10pt) und für die Breite des Schattens: `\sdim` (default 4pt).

```
\usepackage{shadow}
\shabox{
\parbox[t]{7cm}{So ähnlich wie ...}
}
```

### Das fancybox-Paket

So ähnlich wie `\fbox` mit den Parametern `\fboxrule` (default 0.4pt), `\fboxsep` (default 3pt) und für die Breite des Schattens: `\shadowsize` (default 4pt).

```
\usepackage{fancybox}
\shadowbox{
\parbox[t]{7cm}{So ähnlich wie ...}
}}
```

Breite des äußeren Rahmens: `1.5\fboxrule`, Breite des inneren Rahmens: `0.75\fboxrule`, Abstand zwischen beiden: `1.5\fboxrule plus 0.5pt`

```
\usepackage{fancybox}
\doublebox{
\parbox[t]{7cm}{Breite des äußeren ...}
}}
```

Mit dem Befehl `\cornersize` wird der Durchmesser der Eckbögen bestimmt. Mit `\cornersize{num}` wird der Durchmesser auf  $num \cdot \min\{hoehe, breite\}$  gesetzt, mit `\cornersize*{num}` auf  $num$ . Standard-einstellung ist `\cornersize{0.5}`.

```
\usepackage{fancybox}
\fboxsep10pt
\ovalbox{
\parbox[t]{7.5cm}{Mit dem Befehl ...}
}}
```

`\thinlines` legt die Dicke der `ovalbox` fest und `\thicklines` die Dicke der `ovalbox` (genauerer siehe `picture`-Umgebung).

```
\usepackage{fancybox}
\fboxsep10pt
\Ovalbox{\parbox[t]{7.5cm}{
\texttt{\bs thinlines} legt ...}}
```



Um in einer `fbox` oder in den Fancyboxen mehrzeilig arbeiten zu können, muss man innerhalb dieser Rahmenboxen eine `parbox` bzw. eine `minipage`-Umgebung erstellen. Mit den folgenden im `fancybox`-Paket vordefinierten Umgebungen kann man sich das Leben jedoch leichter machen.

In einer Fancybox kann man folgende Umgebungen

verwenden:

- `Bitemize`
- `Benumerate`
- `Bdescription`

```
\usepackage{fancybox}
```

```
In einer Fancybox ...
\doublebox{
\begin{Bitemize}
\item \texttt{Bitemize} ...
\end{Bitemize}}
```

In einer Fancybox kann man folgende Umgebungen verwenden:

`Bcenter`, `Bflushleft`, `Bflushright`  
In diesen Umgebungen werden  
Zeilen mit `\\` getrennt.

```
\usepackage{fancybox}
```

```
In einer Fancybox ...\par\medskip
\doublebox{
\begin{Bcenter}
\texttt{Bcenter, Bflushleft,
Bflushright}\\ In diesen ...
\end{Bcenter}}
```

Im Kapitel 4 Abschnitt FORMELN ANORDNEN wird erklärt, wie man Formeln anordnen kann, hier schonmal mit Rahmen:

$$\begin{array}{l} x = y \quad (3.1) \\ z = 2,718281 \quad (3.2) \end{array}$$

```
\usepackage{fancybox}
```

```
Im Kapitel 4 ...\par\medskip
\doublebox{
\begin{Beqnarray}
x &=& y\\
z &=& 2,718281
\end{Beqnarray}}
```

## Das color-Paket

In Kapitel 5 Farben wird gezeigt, wie man farbig gerahmte und farbig hinterlegte Boxen mit `\fcolorbox` erstellen kann.

## Drehen

```
\usepackage{rotating}
```

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

Notwendigerweise in den Vorspann einbinden.

Text wird gedreht. `Winkel` hat Wert zwischen -180 und 180. Es wird horizontaler und vertikaler Abstand eingefügt.

```
\begin{turn}{Winkel} Text \end{turn}
```

Text wird gedreht. `Winkel` hat Wert zwischen -180 und 180. Es wird <sup>ein</sup> horizontaler und vertikaler Abstand eingefügt.

```
\begin{sideways} Text \end{sideways}
```

Drehung ins Querformat um 90.

## Trennlinie

`\underline{\centerline{}}` erzeugt übrigens ganz tricky eine waagerechte Linie.



## Regelsätze

```
\newtheorem{internerName}{Begriff}[Zusatz-Zähler]
\begin{internerName}[Zusatz-Text] Text \end{internerName}
```

Ohne lange Worte wird an folgendem Beispiel klar, wofür man Regelsätze gebrauchen kann:

**Axiom 1 (Axiom vom Blödsinn)** *Die Existenz von reinem Blödsinn wird axiomatisch gefordert. Was natürlich reiner Blödsinn ist, und somit per Definition existiert.*

**Definition 3.1** *Absoluter Schwachsinn ist der Schnitt aus reinem Blödsinn und völliger Idiotie.*

**Definition 3.2** *Blödfug ist eine Menge, deren Elemente dem reinen Blödsinn angehören.*

(Da wir uns hier gerade im Kapitel (chapter) «3. Textanordnung» befinden, wurde dieser Zähler auch für die Nummerierung der Definition verwendet.)

Ohne lange Worte wird an folgendem Beispiel klar, ...

```
\newtheorem{meineAxiome}{Axiom}
\newtheorem{meineDefs}{Definition}[chapter]

\begin{meineAxiome}[Axiom vom Blödsinn]
  Die Existenz von reinem Blödsinn ... \end{meineAxiome}

\begin{meineDefs}
  Absoluter Schwachsinn ist der Schnitt aus reinem ... \end{meineDefs}

\begin{meineDefs}
  Blödfug ist eine Menge, deren Elemente dem reinen Blödsinn angehören.
\end{meineDefs}
```

(Da wir uns hier gerade im Kapitel (chapter) \flqq 3. Textanordnung\frqq\ ...

## Das theorem-Paket

Das `theorem`-Paket erweitert die Möglichkeiten der gerade vorgestellten Regelsätze.

- Die Paket `theorem` und `amsthm` blockieren sich, man kann also nur eins davon einbinden. (Die  $\LaTeX$ -Klasse `beamer` lädt automatisch `amsthm`!)
- Der `\newtheorem`-Befehl funktioniert wie bisher, muss für das Paket `theorem` allerdings zwingend in die Präambel (d.h. in die Vorspannzeilen) geschrieben werden, so wie auch alle folgenden Deklarationen und Definitionen.

`\newtheorem{internerName2}[internerName]{Begriff2}`

Es wird für die Umgebung `internerName2` derselbe Zähler verwendet wie für die vorher definierte Umgebung `internerName`

`\theorempreskipamount` und `\theorempostskipamount`

Variablen, welche die vertikalen Abstände vor und nach der Umgebung festlegen. Ihre Standardwerte sind:

```
\setlength{\theorempreskipamount}{12pt plus 5pt minus 3pt}
\setlength{\theorempostskipamount}{8pt plus 3pt minus 1.5pt}
```

`\theoremstyle{stil}`

Ist ein Schalter-Befehl, der das Layout bis auf weiters festlegt.

`stil`  $\in$  `{plain, break, marginbreak, changebreak, change, margin}`

`\theorembodyfont{schriftart}`

Wahl des Zeichensatzes für den Theoremkörper, z. B. `\rmfamily`, `\itshape`, ...

`\theoremheaderfont{schriftart}`

Wahl des Zeichensatzes für den Theoremkopf, z. B. `\rmfamily`, `\itshape`, ...

Verwendet zunächst den Zeichensatz des Textkörpers und addiert die Einstellungen von `\theoremheaderfont{...}`; notfalls `schriftart` mit `\normalfont` beginnen.

**Bemerkung 1 (Optional)** *Testtext Testtext  
Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext Testtext Testtext Testtext*

**Bemerkung 1 (Test)**  
*Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext*

**1 Bemerkung (Test)**  
*Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext*

**1 Bemerkung (Test)**  
*Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext Testtext Testtext Testtext  
Testtext Testtext*

**1 Bemerkung (Test)** *Testtext Testtext Test-  
text Testtext Testtext Testtext Testtext Test-  
text Testtext Testtext Testtext Testtext*

**1 Bemerkung (Test)** *Testtext Testtext Test-  
text Testtext Testtext Testtext Testtext Test-  
text Testtext Testtext Testtext Testtext*

`\theoremstyle{plain}`  
entspricht der Originaldefinition von L<sup>A</sup>T<sub>E</sub>X

`\theoremstyle{break}`  
extra Zeilenumbruch  
Theorembodyfont: `\slshape`

`\theoremstyle{marginbreak}`  
extra Zeilenumbruch  
Theorembodyfont: `\slshape`  
Nummerierung: im linken Rand

`\theoremstyle{changebreak}`  
extra Zeilenumbruch  
Theorembodyfont: `\slshape`  
Nummerierung: zuerst

`\theoremstyle{change}`  
Theorembodyfont: `\slshape`  
Nummerierung: zuerst

`\theoremstyle{margin}`  
Theorembodyfont: `\slshape`  
Nummerierung: im linken Rand

## Absatzeinzug

`\parindent0cm` ist in der Präambel deklariert. `Indent` heißt soviel wie Einrückung oder Einzug. Mit den Befehlen `\indent` und `\noindent` kann man Ausnahmen vom allgemein definierten Standard machen. Folgendes Beispiel:

Das ist ein Mann, der sich für Dante hält. Das ist ein Mann, den alle, außer Dante, für Dante halten.

Das ist ein Mann, den alle für Dante halten, nur er selbst glaubt nicht daran.

Das ist ein Mann, den niemand für Dante hält außer Dante. Das ist Dante.

Hans-Magnus Enzensberger, *Der Untergang der Titanic*, Suhrkamp, 1978

`\parindent1em` Das ist ein Mann, der sich für Dante hält. Das ist ein Mann, den alle, außer Dante, für Dante halten.`\par`

Das ist ein Mann, den alle für Dante halten, nur er selbst glaubt nicht daran.

`\par \noindent` Das ist ein Mann, den niemand für Dante hält außer Dante. Das ist Dante.

`\par \hfill` Hans-Magnus Enzensberger, `\textit{Der Untergang der Titanic}`, Suhrkamp, 1978

## Das shapepar-Paket

◇  
 Und  
 jetzt zum  
 Schluss mal  
 noch einige lustige Sachen, die ich hier beim Schreiben unter anderem auch gefunden habe. Um diese Boxform hier zu bekommen muss man zuerst `\usepackage{shapepar}` in die Vorspann-Zeilen einbinden.  
 ◇

Dann  
 stehen einem unter anderem die Befehle `\diamondpar{}` und `\heartpar{}` und `\squarepar{}` oder auch `\shapepar{\nutshape}` zur Verfügung. In einer Shapebox kann kein Mathe-Modus etc. verwendet werden. Wichtig! Man muss mehrmals kompilieren, damit das optimale Ergebnis erzielt wird. Das macht die Shapebox letztendlich auch zu behäbig für größere Dokumente.  
 ♥

Mehr Informationen in der Dokumentation dieses Paketes oder z. B. unter [www.ntg.nl/maps/pdf/11\\_36.pdf](http://www.ntg.nl/maps/pdf/11_36.pdf)

## Kapitel 4

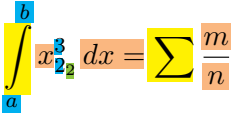
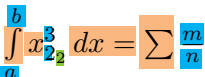
# Mathemodus

### Mathematische Umgebungen

Die mathematische Umgebung lässt sich mit folgenden Befehlen erzeugen:

Langform: <code>\begin{math}\end{math}</code> Kurzform: <code>\$. . . \$</code> oder <code>\( . . . \)</code>	}	Textformeln, d.h. Formeln innerhalb der Textzeile, Zeilenumbruch mit <code>\\</code> möglich.
Langform: <code>\begin{displaymath}\end{displaymath}</code> Kurzform: <sup>1</sup> <code>\[ . . . \]</code>		abgesetzte Formeln, d.h. Formeln in einem extra Absatz, Zeilenumbruch nicht möglich
<code>\begin{equation}\end{equation}</code> <code>\begin{equation*}\end{equation*}</code>	}	auch abgesetzt vom Text, mit Nummerierung, * unterdrückt Nummerierung, kein Zeilenumbruch

### Schriftgrößen im Mathemodus

<code>\displaystyle</code>	Größe der Symbole in abgesetzten Formeln	<i>abgesetzte Formel:</i>
<code>\textstyle</code>	Größe der Symbole in Textformeln, Größe der Schrift in beiden Formeltypen	
<code>\scriptstyle</code>	Größe der Schrift in einfacher Indizierung in beiden Formeltypen	<i>Textformel:</i>
<code>\scriptscriptstyle</code>	Größe der Schrift in zweifacher Indizierung in beiden Formeltypen	

Schalte ich auf eine der vier Schriftgrößen, indem ich den entsprechenden Befehl (`\displaystyle`, `\textstyle` etc.) aufrufe, so lege ich damit die **größte verwendete Schrift** in den darauffolgenden Formel fest. Tiefer- bzw. Höher-Gestelltes, Brüche, etc. werden soweit wie möglich auch kleiner dargestellt.

Im `scriptstyle` werden normale Variablen auf Größe `scriptstyle` ausgegeben, alle Indizes und Brüche im `scriptscriptstyle`. Wird schließlich `\scriptscriptstyle` vor die Formel gesetzt, so erscheinen alle Formelelemente in der kleinsten Schriftgröße `scriptscriptstyle`.

<sup>1</sup>\$ ist Plain-TeX. Es existiert auch der Plain-TeX-Befehl `$$` als Abkürzung zum Öffnen und zum Schließen der Displaymath-Umgebung; hierbei funktioniert allerdings u.a. die `fleqn`-Klassenoption nicht mehr.

Folgende Symbole existieren in zwei Größen, unterscheiden sich deshalb im Displaystyle und im Textstyle:

$\Sigma$	$\sum$	<code>\sum</code>	$\cap$	$\bigcap$	<code>\bigcap</code>	$\odot$	$\bigodot$	<code>\bigodot</code>
$\int$	$\int$	<code>\int</code>	$\cup$	$\bigcup$	<code>\bigcup</code>	$\otimes$	$\bigotimes$	<code>\bigotimes</code>
$\oint$	$\oint$	<code>\oint</code>	$\sqcup$	$\bigsqcup$	<code>\bigsqcup</code>	$\oplus$	$\bigoplus$	<code>\bigoplus</code>
$\prod$	$\prod$	<code>\prod</code>	$\vee$	$\bigvee$	<code>\bigvee</code>	$\uplus$	$\biguplus$	<code>\biguplus</code>
$\coprod$	$\coprod$	<code>\coprod</code>	$\wedge$	$\bigwedge$	<code>\bigwedge</code>			

Sei  $P$  ein Punkt und  $x = \sum a_i$ .

$$x = \sum a_i$$

Sei  $P$  ein Punkt und  $(x = \sum a_i)$ .

`\[x = \sum a_i\]`

Auch bei bloßem Nennen einer Variablen in einem Text, sollte man diese mit einer einfachen Matheumgebung einklammern, damit sie sofort als Variable erkannt wird.

## Formelelemente

### Einfache Formelelemente

$x dy$	<code>\\$x dy\\$</code>	$a^b a_b$	<code>\\$a^b\, , a_b\\$</code>	$a_{bcd}^{ef}$	<code>\\$a_{bcd}^{ef}\\$</code>
$x \, dy$	<code>\\$x\, , dy\\$</code>	$a^{b^c}$	<code>\\$a^{b^c}\\$</code>	$a_{bcd}^{ef}$	<code>\\${a_{bcd}}^{ef}\\$</code>

$\frac{m}{n}$	<code>\\$\frac{m}{n}\\$</code>	$\sqrt{x}$	<code>\\$\sqrt{x}\\$</code>
$\binom{n+1}{k}$	<code>\\$\binom{n+1}{k}\\$</code>	$\sqrt[3]{x^2}$	<code>\\$\sqrt[3]{x^2}\\$</code>
$\binom{n+1}{k}$	<code>\\$\binom{n+1}{k}\\$</code>	$\sqrt[n]{\frac{5}{x}}$	<code>\\$\sqrt[n]{\frac{5}{x}}\\$</code>

$$\frac{a}{b} \frac{a}{b}$$

`\$tfrac{a}{b}\dfrac{a}{b}\$` im Textstyle und im Displaystyle

$$\sqrt[p]{x^2 + y^2} \sqrt[p]{x^2 + y^2}$$

`\$\sqrt[p]{x^2+y^2}\sqrt[\leftroot{-2}\uproot{2}]{p}{x^2+y^2}\$`

$\int x^2 dx$	<code>\\$\int x^2\, dx\\$</code>	$\sum x^2 dx$	<code>\\$\sum x^2\, dx\\$</code>
$\int_a^b x^2 dx$	<code>\\$\int_a^b x^2\, dx\\$</code>	$\sum_a^b x^2 dx$	<code>\\$\sum_a^b x^2\, dx\\$</code>
$\int_a^b x^2 dx$	<code>\\$\int\limits_a^b x^2\, dx\\$</code>	$\sum_a^b x^2 dx$	<code>\\$\sum\limits_a^b x^2\, dx\\$</code>

**Fortsetzungspunkte**

$\vdots$	<code>\vdots</code>	<code>\quad</code>	<code>\ddots</code>	% vertical/diagonal dots
$a_1+a_2+\cdots+a_k+\cdots+a_n$	<code>\$a_1+a_2+\cdots +a_n\$</code>	% center dots		
$a_1, a_2, \dots, a_n$	<code>\$a_1, a_2, \ldots , a_n\$</code>	% low dots		

... \ldots funktionieren auch in normalem Text.

**Mathematische Akzente**

$\hat{a}$	<code> \$\hat{a}\$ </code>	$\check{a}$	<code> \$\check{a}\$ </code>	$\breve{a}$	<code> \$\breve{a}\$ </code>
$\acute{a}$	<code> \$\acute{a}\$ </code>	$\grave{a}$	<code> \$\grave{a}\$ </code>	$\tilde{a}$	<code> \$\tilde{a}\$ </code>
$\bar{a}$	<code> \$\bar{a}\$ </code>	$\vec{a}$	<code> \$\vec{a}\$ </code>	$\dot{a}$	<code> \$\dot{a}\$ </code>
$\ddot{a}$	<code> \$\ddot{a}\$ </code>	$\vec{i}$	<code> \$\vec{\imath}\$ </code>	$\tilde{j}$	<code> \$\tilde{\jmath}\$ </code>
$\widehat{1-x}$	<code> \$\widehat{1-x}\$ </code>	$\widetilde{xyz}$	<code> \$\widetilde{xyz}\$ </code>		

**Funktionsnamen**

Den Standard, Variablenamen immer *Italic* zu setzen, erledigt L<sup>A</sup>T<sub>E</sub>X automatisch in der Matheumgebung. Namen von Funktionen, Operatoren usw. sollten jedoch in aufrechten Buchstaben wiedergegeben werden. Dazu stehen die folgenden Befehle innerhalb der Matheumgebung bereit:

- `\arccos` `\cosh` `\deg` `\gcd` `\ker` `\limsup` `\min` `\sin` `\tanh`
- `\arcsin` `\cot` `\det` `\hom` `\lg` `\ln` `\Pr` `\sinh` `\varlimsup`
- `\arctan` `\coth` `\dim` `\inf` `\lim` `\log` `\projlim` `\sup` `\varliminf`
- `\arg` `\coth` `\dim` `\inf` `\lim` `\log` `\projlim` `\sup` `\varinjlim`
- `\cos` `\csc` `\exp` `\injl` `\liminf` `\max` `\sec` `\tan` `\varprojlim`

Mit `\DeclareMathOperator{\internerName}{ausgabe}` kann man sich dazu noch seine eigenen Operatoren im Vorspann definieren, und genauso wie die anderen Befehle verwenden, z. B. `\DeclareMathOperator{\res}{res}` für das Residuum oder `\DeclareMathOperator{\GL}{GL}`.

<p><code> \$\lim_{x \to \infty}\$ </code> erscheint in ...</p> <p>... Textformeln als: <span style="border: 1px solid black; padding: 2px;"><math>\lim_{x \rightarrow \infty}</math></span></p> <p>... abgesetzten Formeln als: <span style="border: 1px solid black; padding: 2px;"><math>\lim_{x \rightarrow \infty}</math></span></p>	<p>Solche (untergestellte) Grenzangaben funktionieren bei den Befehlen:</p> <p><code>\det</code>, <code>\gcd</code>, <code>\inf</code>, <code>\lim</code>, <code>\liminf</code>, <code>\limsup</code>, <code>\max</code>, <code>\min</code>, <code>\Pr</code>, <code>\sup</code></p>
--	--

Um untergestellte Grenzangaben auch im `textstyle` zu erhalten, kann man (wie bei Integral, Summe, ...) den Befehl `\limits` zwischen Operator und Underline bzw. Dach anwenden.

`$\max\limits_{x \to \infty}$`  erzeugt:  $\max_{x \rightarrow \infty}$

`$a \bmod b$`  erzeugt  $a \bmod b$   `$y \pmod{a+b}$`  erzeugt  $y \pmod{a+b}$

### Klammern

Klammern können sich an die Größe des eingeschlossenen Textes anpassen. Dazu schreibt man `\left` und `\right` unmittelbar vor die beiden entsprechenden Klammern (z. B. `\left[ ... \right]`). Sollte man beispielsweise nur die linke Klammer brauchen, so muss die Klammer dennoch mit `\right.` geschlossen werden. `array` ist übrigens eine Tabellenumgebung im Mathemodus.

$\vec{a} + \vec{b} + \vec{c} = \begin{pmatrix} x \\ y \end{pmatrix}$	<code>\vec{a}+\vec{b}+\vec{c} = \left( {x \atop y}\right)</code>
$y = \begin{cases} -1 & : x < 0 \\ 0 & : x = 0 \\ 1 & : x > 0 \end{cases}$	<code>y= \left\{ \begin{array}{r@{\quad}\quad} -1 &amp; x &lt; 0 \\ 0 &amp; x = 0 \\ 1 &amp; x &gt; 0 \end{array} \right.</code>

`\left` und `\right` können im Mathemodus auf folgende 22 Klammern angewendet werden:

<code>(</code>	<code>)</code>	<code>[</code>	<code>]</code>	<code>{</code>	<code>}</code>	<code> </code>	<code>  </code>	<code>/</code>	<code>\</code>	<code>&lt;</code>	<code>&gt;</code>
<code>(</code>	<code>)</code>	<code>[</code>	<code>]</code>	<code>\{</code>	<code>\}</code>	<code> </code>	<code>\ </code>	<code>/</code>	<code>\backslash</code>	<code>\langle</code>	<code>\rangle</code>
<code>\lfloor</code>	<code>\rfloor</code>	<code>\lceil</code>	<code>\rceil</code>	<code>\uparrow</code>	<code>\downarrow</code>	<code>\updownarrow</code>	<code>\Uparrow</code>	<code>\Downarrow</code>	<code>\Updownarrow</code>		

Manuelle Größenwahl der Klammern:

<code>normal {a b}</code> <code>\big {a b}</code> <code>\Big {a b}</code> <code>\bigg {a b}</code> <code>\Bigg {a b}</code>
---

```

$ \text{\tt normal } \{ a\lfloor b \} \quad \text{\tt\bs big } \big\{ a\big\lfloor b \big\}
\quad \text{\tt\bs Big } \Big\{ a\Big\lfloor b \Big\} \quad \text{\tt\bs bigg }
\bigg\{ a\bigg\lfloor b \bigg\} \quad \text{\tt\bs Bigg } \Bigg\{ a\Bigg\lfloor b \Bigg\}

```

### Matrizen und Felder

Entweder man baut sich Matrizen und Felder mit der `array`-Umgebung (funktioniert genau wie die `tabular`-Umgebung) und klammert diese dann entsprechend ein (siehe das zweite Beispiel Abschnitt KLAMMERN), oder man benutzt die fertigen Umgebungen `matrix` (keine Klammern), `pmatrix` (runde Klammern), `bmatrix` (eckige Klammern), `Bmatrix` (geschweifte Klammern), `vmatrix` (gerade Klammern), `Vmatrix` (doppelte gerade Klammern).

$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$	<code>\begin{Bmatrix}</code> <code>a &amp; b \ \ c &amp; d</code> <code>\end{Bmatrix}</code>
--	--

$\begin{vmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} = 1$	<code>\begin{vmatrix}</code> <code>a &amp; b \ \ c &amp; d</code> <code>\end{vmatrix}</code>
---	--



Über-/Unterstreichen und Aufgestockte Symbole

$\underline{abcdef}$ $\underbrace{abc}_{123} \overbrace{def\ ghi\ jkl}^{456}$	$\backslash\text{underline}\{abc\}$ $\backslash\text{overline}\{def\}\backslash\text{quad}$ $\backslash\text{underbrace}\{abc\}$ $\backslash\text{underbrace}\{def\}_{123}$ $\backslash\text{overbrace}\{ghi\}$ $\backslash\text{overbrace}\{jkl\}^{\{456\}}$
Aufgestockt: $\overset{\text{oben}}{\underset{\text{unten}}{\text{Def.}}} \Longrightarrow$	Aufgestockt: $\backslash\text{stackrel}\{\text{oben}\}\{\text{unten}\}$ $\backslash\text{quad}$ $\backslash\text{stackrel}\{\text{Def.}\}\{\backslash\text{Longrightarrow}\}$
Das Paket <code>amsmath</code> ermöglicht: $\overset{\text{oben}}{\underset{\text{unten}}{\text{normal}}} \xrightarrow{\text{Def.}} \text{normal}$	Das Paket <code>\texttt{amsmath}</code> ermöglicht: $\backslash\text{overset}\{\text{oben}\}\{\text{normal}\}$ $\backslash\text{quad}$ $\backslash\text{overset}\{\text{Def.}\}\{\backslash\text{Longrightarrow}\}$ $\backslash\text{quad}$ $\backslash\text{underset}\{\text{unten}\}\{\text{normal}\}$

Änderung der Schrift in Formeln

$\$ABCDEF G hijklmno\dots\$$	$ABCDEF G hijklmno1234$ $\alpha\beta\gamma\Pi\Xi + - * / \mapsto\rightarrow\Rightarrow\in\subseteq\llcorner$	normal für Variablen, Leerzeichen werden ignoriert
$\backslash\text{mathnormal}\{ABC\dots\}$	$ABCDEF G hijklmno1234$ $\alpha\beta\gamma\Pi\Xi + - * / \mapsto\rightarrow\Rightarrow\in\subseteq\llcorner$	Mediävalziffern, große griechische Buchstaben schräg
$\backslash\text{mathbf}\{ABC\dots\}$	<b>ABCDEF G hijklmno1234</b>	
$\backslash\text{boldsymbol}\{\backslash\alpha\dots\}$	$\alpha\beta\gamma\Pi\Xi + - * / \mapsto\rightarrow\Rightarrow\in\subseteq\llcorner$	kann aber leider nicht alle Symbole
$\backslash\text{pmb}\{\backslash\alpha\dots\}$	$\alpha\beta\gamma\Pi\Xi + - * / \mapsto\rightarrow\Rightarrow\in\subseteq\llcorner$	poor man's bold, die letzte Rettung
$\backslash\text{mathbb}\{ABC\dots\}$	ABCDEFGHIJKLMN OPQRSTUVWXYZ	
$\backslash\text{mathfrak}\{ABC\dots\}$	𐒐𐒑𐒒𐒓𐒔𐒕𐒖𐒗𐒘𐒙𐒚1234	Euler-Fraktur-Alphabet
$\backslash\text{mathcal}\{ABC\dots\}$	<i>ABCDEFGHIJKLMN OPQRST UV</i>	
$\backslash\text{mathrm}\{ABC\dots\}$	ABCDEF G hijklmno1234	
$\backslash\text{mathsf}\{ABC\dots\}$	ABCDEF G hijklmno1234	
$\backslash\text{mathit}\{ABC\dots\}$	<i>ABCDEF G hijklmno1234</i>	kleinerer Buchstabenzwischenraum
$\backslash\text{mathtt}\{ABC\dots\}$	ABCDEF G hijklmno1234	
$\backslash\text{text}\{ABC\dots\}$	ABCDEF G hijklmno 1234	erhält Leerzeichen
$\backslash\text{boldmath } \$ABC \dots\$$ $\backslash\text{unboldmath}$	<b><i>ABCDEF G hijklmno1234</i></b> $\alpha + * \subseteq$	Schalter außerhalb Matheumgebung

## Formeln anordnen

$$\begin{aligned} f(x, y, z) &= x^2 + y^2 \\ &+ \frac{x + y}{z} \\ &= \frac{x + y + (x^2 + y^2)z}{z} \end{aligned} \quad (4.1)$$

wie eine zweispaltige Tabelle `{rl}`, nummeriert alles als eine einzige Gleichung, `equation*` unterdrückt die Nummerierung

```
\begin{equation}\begin{split}
f(x,y,z) &= x^2+y^2\\
&\quad + \frac{x+y}{z}\\
&= \frac{x+y+(x^2+y^2)z}{z}
\end{split}\end{equation}
```

$$\begin{aligned} f(x, y, z) &= x^2 + y^2 \\ &+ \frac{x + y}{z} \\ &= \frac{x + y + (x^2 + y^2)z}{z} \end{aligned} \quad (4.2)$$

wie eine dreispaltige Tabelle `{rcl}`, nummeriert jede Zeile, `eqnarray*` unterdrückt die Nummerierung

```
\begin{eqnarray}
f(x,y,z) &=& x^2+y^2\\
&\notag &+ \frac{x+y}{z}\\
&=& \frac{x+y+(x^2+y^2)z}{z}
\end{eqnarray}
```

$$\begin{aligned} f(x, y, z) &= x^2 + y^2 \\ &+ \frac{x + y}{z} \\ &= \frac{x + y + (x^2 + y^2)z}{z} \end{aligned} \quad (4.4)$$

wie eine zweispaltige Tabelle `{rl}`, nummeriert jede Zeile, `align*` unterdrückt die Nummerierung, formatiert schöner als `eqnarray`

```
\begin{align}
f(x,y,z) &= x^2+y^2\\
&\notag &+ \frac{x+y}{z}\\
&= \frac{x+y+(x^2+y^2)z}{z}
\end{align}
```

$$\begin{aligned} f(x, y, z) &= x^2 + y^2 + \frac{x + y}{z} \\ &= \frac{x + y + (x^2 + y^2)z}{z} \end{aligned} \quad (4.6)$$

zentriert, nummeriert jede Zeile, `gather*` unterdrückt die Nummerierung

```
\begin{gather}
f(x,y,z) = x^2+y^2 + \frac{x+y}{z}\\
= \frac{x+y+(x^2+y^2)z}{z}
\end{gather}
```

$$y = \begin{cases} -1 & : x < 0 \\ 0 & : x = 0 \\ 1 & : x > 0 \end{cases}$$

bastelt die Klammer selbst mit einem `array`

```
$y= \left\{ \begin{array}{r@{\quad:\quad}l}
-1 & x<0 \\
0 & x=0 \\
1 & x>0 \end{array} \right\}$
```

$$y = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

fertige Umgebung

```
$y= \begin{cases}
-1 & x<0 \\
0 & x=0 \\
1 & x>0 \end{cases}$
```

gerahmte Formeln

$$\boxed{\sum_{n=0}^{\infty} f(n) = \dots} \quad (4.8)$$

```
\begin{equation}\fbox{
$\displaystyle \sum_{n=0}^{\infty}
f(n) = \dots$
}\end{equation}
```

Siehe auch Kapitel 4, Abschnitt RAHMEN, für das fancybox-Paket.

## Formelnummern und Referenzen

Für die Verwendung von Referenzen muss man zweimal mit latex kompilieren. Die Label werden beim ersten Kompilieren in eine .aux-Datei geschrieben und dort von \ref ausgelesen.

- `\label{internerName}` kennzeichnet eine Stelle im Text, zur späteren Weiterverwendung; für `interner Name` können Buchstaben, Zahlen und Zeichen außer `\ # $ % & ~ ^ _ { }` verwendet werden.
- `\ref{internerName}` greift auf das Label zurück, um im Text die Referenz zu setzen.
- `\eqref{internerName}` wirkt genauso, aber hier wird die Nummer eingeklammert.
- `\pageref{internerName}` gibt die Seitenzahl zurück, auf der das Label im fertigen Dokument steht.

$$\sum_{n=0}^{\infty} f(n) = \dots \quad (4.9)$$

Wie in 4.9 gezeigt wurde, ist ...  
 Aus (4.9) folgt weiterhin, dass ...  
 Wie auf Seite 35 bereits erwähnt ...

```
\begin{equation}\label{Rechenregel}
\sum_{n=0}^{\infty} f(n) = \dots
\end{equation}
Wie in \ref{Rechenregel} gezeigt ...\par
Aus \eqref{Rechenregel} folgt ...\par
Wie auf Seite \pageref{Rechenregel} ...
```

- `\tag{zeichen}` setzt die Formelnummer auf `zeichen`.
- `\tag*{zeichen}` wirkt genauso, lässt aber außerdem die Klammern weg.
- `\notag` unterdrückt die Formelnummer (äquivalent zu `\nonumber`).

$$\sum_{n=0}^{\infty} g(n) = \dots \quad (*)$$

$$\sum_{n=0}^{\infty} h(n) = \dots \quad *$$

$$x^4 + y^5 = r^7 \quad (4.10)$$

$$x^3 + y^2 = s^3$$

$$\left( \sum_{n=0}^{\infty} f(n) \right) - x^2 = \dots \quad (4.9')$$

Wie in \* gezeigt wurde, ist ...  
 Aus (\*) folgt weiterhin, dass ...

```
\begin{equation}\tag{${\ast}$}
\label{Rechenregel2}
\sum_{n=0}^{\infty} g(n) = \dots\
\end{equation}
\begin{gather}
\tag*{${\divideontimes}$}
\label{Rechenregel3}
\sum_{n=0}^{\infty} h(n) = \dots\
x^4+y^5=r^7\
x^3+y^2=s^3\notag\
\left(\sum_{n=0}^{\infty}
f(n)\right) - x^2 = \dots
\tag{\ref{Rechenregel}$'$}$
\end{gather}
Wie in \ref{Rechenregel2} gezeigt...\par
Aus \eqref{Rechenregel3} folgt ...\par
```

<code>\renewcommand{\theequation}{Formatierung}</code>	setzt die Formatierung der Formelnummer, kann somit bspw. die Formelnummer an die <code>section</code> -Nummer ( <code>\thesection</code> ) koppeln.
<code>\numberwithin{equation}{Gliederungsebene}</code>	koppelt die Gliederungseben unkomplizierter an die Formelnummer; setzt die Formelnummer am Ende eines Abschnittes wieder zurück; <code>Gliederungsebene</code> $\in$ <code>{part, chapter, section, subsection, ...}</code> .

$$\sum_{n=0}^{\infty} f(n) = \dots \quad (\text{XI})$$

Wie in (XI) gezeigt wurde, ist ...

```
\begin{equation}\label{Rechenregel4}
\sum_{n=0}^{\infty} f(n) = \dots
\end{equation}
Wie in \ref{Rechenregel4} gezeigt ...
```

### Formatierung von Zählern

Die Manipulation von  $\LaTeX$ -Zählern und die Definition eigener Befehle und Zähler wird in einem späteren Kapitel ausführlich erklärt werden. An dieser Stelle deshalb nur eine kurze Übersicht über mögliche Formatierungen, um den `renewcommand`-Befehl sinnvoll nutzen zu können:

<code>\arabic{zaehler}</code>	<code>zaehler</code> als arabische Zahl
<code>\roman{zaehler}</code>	<code>zaehler</code> als kleine römische Zahl
<code>\Roman{zaehler}</code>	<code>zaehler</code> als große römische Zahl
<code>\alph{zaehler}</code>	<code>zaehler</code> als Kleinbuchstabe
<code>\Alph{zaehler}</code>	<code>zaehler</code> als Großbuchstabe
<code>\fnsymbol{zaehler}</code>	<code>zaehler</code> als Symbol; nur im Mathemodus möglich, der Wert von <code>zaehler</code> darf nicht größer als 9 sein
<code>\thezaehler</code>	Befehl für die Darstellungsform von <code>zaehler</code>

## Mathematische Symbole

### Vergleichssymbole

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

**Binäre Operationssymbole**

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

**Griechische Kleinbuchstaben**

$\alpha$	<code>\alpha</code>	$\beta$	<code>\beta</code>	$\gamma$	<code>\gamma</code>	$\delta$	<code>\delta</code>
$\epsilon$	<code>\epsilon</code>	$\varepsilon$	<code>\varepsilon</code>	$\zeta$	<code>\zeta</code>	$\eta$	<code>\eta</code>
$\theta$	<code>\theta</code>	$\vartheta$	<code>\vartheta</code>	$\iota$	<code>\iota</code>	$\kappa$	<code>\kappa</code>
$\lambda$	<code>\lambda</code>	$\mu$	<code>\mu</code>	$\nu$	<code>\nu</code>	$\xi$	<code>\xi</code>
$\omicron$	<code>\omicron</code>	$\pi$	<code>\pi</code>	$\varpi$	<code>\varpi</code>	$\rho$	<code>\rho</code>
$\varrho$	<code>\varrho</code>	$\sigma$	<code>\sigma</code>	$\varsigma$	<code>\varsigma</code>	$\tau$	<code>\tau</code>
$\upsilon$	<code>\upsilon</code>	$\phi$	<code>\phi</code>	$\varphi$	<code>\varphi</code>	$\chi$	<code>\chi</code>
$\psi$	<code>\psi</code>	$\omega$	<code>\omega</code>				

**Griechische Großbuchstaben**

$\Gamma$	<code>\Gamma</code>	$\Delta$	<code>\Delta</code>	$\Theta$	<code>\Theta</code>	$\Lambda$	<code>\Lambda</code>
$\Xi$	<code>\Xi</code>	$\Pi$	<code>\Pi</code>	$\Sigma$	<code>\Sigma</code>	$\Upsilon$	<code>\Upsilon</code>
$\Phi$	<code>\Phi</code>	$\Psi$	<code>\Psi</code>	$\Omega$	<code>\Omega</code>		

## Pfeile

	<code>\longrightarrow</code>		<code>\longleftarrow</code>		<code>\rightarrow</code>		<code>\leftarrow</code>
	<code>\Longrightarrow</code>		<code>\Longleftarrow</code>		<code>\Rightarrow</code>		<code>\Leftarrow</code>
	<code>\downarrow</code>		<code>\Downarrow</code>		<code>\uparrow</code>		<code>\Uparrow</code>
	<code>\leftrightarrow</code>		<code>\Leftrightarrow</code>		<code>\updownarrow</code>		<code>\Updownarrow</code>
	<code>\longlefttrightarrow</code>		<code>\Longlefttrightarrow</code>		<code>\longmapsto</code>		<code>\mapsto</code>
	<code>\rightleftharpoons</code>		<code>\hookrightarrow</code>		<code>\hookleftarrow</code>		<code>\leadsto</code>
	<code>\rightharpoondown</code>		<code>\leftharpoondown</code>		<code>\rightharpoonup</code>		<code>\leftharpoonup</code>
	<code>\nearrow</code>		<code>\searrow</code>		<code>\swarrow</code>		<code>\nwarrow</code>
	<code>\twoheadrightarrow</code>		<code>\twoheadleftarrow</code>		<code>\rightarrowtail</code>		<code>\leftarrowtail</code>
	<code>\circlearrowright</code>		<code>\circlearrowleft</code>		<code>\dashrightarrow</code>		<code>\dashleftarrow</code>
	<code>\rightrightarrows</code>		<code>\leftrightharpoons</code>		<code>\rightleftarrows</code>		<code>\leftleftarrows</code>
	<code>\downdownarrows</code>		<code>\looparrowright</code>		<code>\looparrowleft</code>		<code>\upuparrows</code>
	<code>\curvearrowright</code>		<code>\curvearrowleft</code>		<code>\Lsh</code>		<code>\Rsh</code>
	<code>\downharpoonright</code>		<code>\downharpoonleft</code>		<code>\upharpoonright</code>		<code>\upharpoonleft</code>
	<code>\leftrightsquigarrow</code>		<code>\rightsquigarrow</code>		<code>\nLeftrightarrow</code>		<code>\multimap</code>
	<code>\nrightarrow</code>		<code>\nRrightarrow</code>		<code>\nleftarrow</code>		<code>\nLeftarrow</code>

## Sonderzeichen

	<code>\clubsuit</code>		<code>\spadesuit</code>		<code>\diamondsuit</code>		<code>\heartsuit</code>
	<code>\blacksquare</code>		<code>\blacktriangledown</code>		<code>\blacktriangle</code>		<code>\blacklozenge</code>
	<code>\bigstar</code>		<code>\Join</code>		<code>\infty</code>		<code>\triangle</code>
	<code>\forall</code>		<code>\exists</code>		<code>\nexists</code>		<code>\Finv</code>
	<code>\mho</code>		<code>\Game</code>		<code>\nabla</code>		<code>\partial</code>
	<code>\hslash</code>		<code>\hbar</code>		<code>\Bbbk</code>		<code>\aleph</code>
	<code>\imath</code>		<code>\jmath</code>		<code>\ell</code>		<code>\wp</code>
	<code>\Re</code>		<code>\Im</code>		<code>\prime</code>		<code>\backprime</code>
	<code>\surd</code>		<code>\neg</code>		<code>\circledS</code>		<code>\emptyset</code>
	<code>\flat</code>		<code>\natural</code>		<code>\sharp</code>		<code>\varnothing</code>
	<code>\top</code>		<code>\bot</code>		<code>\eth</code>		<code>\complement</code>
	<code>\angle</code>		<code>\measuredangle</code>		<code>\sphericalangle</code>		<code>\diagup</code>
	<code>\diagdown</code>		<code>\parallel</code>		<code>\l</code>		<code>\not&lt;</code>
	<code>\not=</code>		<code>\not\approx</code>		<code>\not\supset</code>		<code>\notin</code>

## Kapitel 5

# Farben

### Farbige Schrift und Boxen

<code>\usepackage{color}</code>	muss in die Präambel.
<code>\color[model]{farbe}</code>	setzt den nachfolgenden Text auf <code>farbe</code> . Schalterbefehl.
<code>\textcolor[model]{farbe}{Text}</code>	setzt den übergebenen Text auf <code>farbe</code> . Argumentbefehl.
<code>\pagecolor[model]{seitenfarbe}</code>	setzt die Hintergrundfarbe der nachfolgenden Seiten auf <code>seitenfarbe</code> .

`\colorbox[model]{hintergrundfarbe}{Text}`

erstellt eine LR-Box mit der gewählten Hintergrundfarbe.

`\fcolorbox[model]{rahmenfarbe}{hintergrundfarbe}{Text}`

erstellt eine Fbox mit der gewählten Hintergrund- und Rahmenfarbe. Parameter lassen sich wie immer mit `\fboxsep` und `\fboxrule` einstellen.

`\definecolor{neuerfarbname}{modell}{farbeImModell}{Text}`

definiert eine neue Farbe, die bspw. mit `\color{neuerfarbname}` ohne Angabe von `[modell]` verwendet werden kann, z. B. `\definecolor{hellgrau}{gray}{0.85}`.

### Farbtabellen

model: `[gray]{0.xx}`

00 05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

model: `[rgb]{0..1,0..1,0..1}`

0.0,0.0	0.2,0.0	0.4,0.0	0.5,0.0	0.7,0.0	0.9,0.0	1.0,0.0
1,0.0,0	1,0.2,0	1,0.4,0	1,0.5,0	1,0.7,0	1,0.9,0	1,1.0,0
1.0,1,0	0.9,1,0	0.7,1,0	0.5,1,0	0.4,1,0	0.2,1,0	0.0,1,0
0,1,0.0	0,1,0.2	0,1,0.4	0,1,0.5	0,1,0.7	0,1,0.9	0,1,1.0
0,1,0,1	0,0.9,1	0,0.7,1	0,0.5,1	0,0.4,1	0,0.2,1	0,0.0,1
0.0,0,1	0.2,0,1	0.4,0,1	0.5,0,1	0.7,0,1	0.9,0,1	1.0,0,1
1,0.0,1	1,0.2,1	1,0.4,1	1,0.5,1	1,0.7,1	1,0.9,1	

model: [cmyk]{0..1,0..1,0..1,0..1}

0,0,0,0.0	1,0,0,0.0	1,1,0,0.0	0,1,0,0.0	0,1,1,0.0	0,0,1,0.0	1,0,1,0.0
0,0,0,0.2	1,0,0,0.2	1,1,0,0.2	0,1,0,0.2	0,1,1,0.2	0,0,1,0.2	1,0,1,0.2
0,0,0,0.4	1,0,0,0.4	1,1,0,0.4	0,1,0,0.4	0,1,1,0.4	0,0,1,0.4	1,0,1,0.4
0,0,0,0.6	1,0,0,0.6	1,1,0,0.6	0,1,0,0.6	0,1,1,0.6	0,0,1,0.6	1,0,1,0.6

model: [named]

GreenYellow	Farbe: GreenYellow	RoyalPurple	Farbe: RoyalPurple
Yellow	Farbe: Yellow	BlueViolet	Farbe: BlueViolet
Goldenrod	Farbe: Goldenrod	Periwinkle	Farbe: Periwinkle
Dandelion	Farbe: Dandelion	CadetBlue	Farbe: CadetBlue
Apricot	Farbe: Apricot	CornflowerBlue	Farbe: CornflowerBlue
Peach	Farbe: Peach	MidnightBlue	Farbe: MidnightBlue
Melon	Farbe: Melon	NavyBlue	Farbe: NavyBlue
YellowOrange	Farbe: YellowOrange	RoyalBlue	Farbe: RoyalBlue
Orange	Farbe: Orange	Blue	Farbe: Blue
BurntOrange	Farbe: BurntOrange	Cerulean	Farbe: Cerulean
Bittersweet	Farbe: Bittersweet	Cyan	Farbe: Cyan
RedOrange	Farbe: RedOrange	ProcessBlue	Farbe: ProcessBlue
Mahogany	Farbe: Mahogany	SkyBlue	Farbe: SkyBlue
Maroon	Farbe: Maroon	Turquoise	Farbe: Turquoise
BrickRed	Farbe: BrickRed	TealBlue	Farbe: TealBlue
Red	Farbe: Red	Aquamarine	Farbe: Aquamarine
OrangeRed	Farbe: OrangeRed	BlueGreen	Farbe: BlueGreen
RubineRed	Farbe: RubineRed	Emerald	Farbe: Emerald
WildStrawberry	Farbe: WildStrawberry	JungleGreen	Farbe: JungleGreen
Salmon	Farbe: Salmon	SeaGreen	Farbe: SeaGreen
CarnationPink	Farbe: CarnationPink	Green	Farbe: Green
Magenta	Farbe: Magenta	ForestGreen	Farbe: ForestGreen
VioletRed	Farbe: VioletRed	PineGreen	Farbe: PineGreen
Rhodamine	Farbe: Rhodamine	LimeGreen	Farbe: LimeGreen
Mulberry	Farbe: Mulberry	YellowGreen	Farbe: YellowGreen
RedViolet	Farbe: RedViolet	SpringGreen	Farbe: SpringGreen
Fuchsia	Farbe: Fuchsia	OliveGreen	Farbe: OliveGreen
Lavender	Farbe: Lavender	RawSienna	Farbe: RawSienna
Thistle	Farbe: Thistle	Sepia	Farbe: Sepia
Orchid	Farbe: Orchid	Brown	Farbe: Brown
DarkOrchid	Farbe: DarkOrchid	Tan	Farbe: Tan
Purple	Farbe: Purple	Gray	Farbe: Gray
Plum	Farbe: Plum	Black	Farbe: Black
Violet	Farbe: Violet	White	

\usepackage[dvipsnames]{color} ... mögliche Lösung, falls in Verbindung mit graphics-Paketen oder pdflatex die Namen der Farben nicht erkannt werden.



# Literaturverzeichnis

- [1] Helmut Kopka, *Einführung Band 1*, Addison-Wesley Verlag, München, 2000
- [2] Roland Willms, *L<sup>A</sup>T<sub>E</sub>X - echt einfach*, Franzis' Verlag GmbH, München, 2001
- [3] Michel Goossens, Frank Mittelbach, Alexander Samarin, *Der L<sup>A</sup>T<sub>E</sub>X-Begleiter*, korrigierter Nachdruck, Addison-Wesley Verlag, München, 2002
- [4] Diverse L<sup>A</sup>T<sub>E</sub>X-Kurs-Mitschriften des Wurzel-Vereins Jena e.V., bis 2001

**Vorschläge für weitere Kapitel:**

- GRAPHIKEN UND BILDER einbinden, Gleitobjekte
- SEITENSTIL: Fußnoten und Randnotizen, verschiedene Längenmaße einer Seite, mehrere Spalten
- DOKUMENTSTIL: Untergliederung eines Textes (mit Inhaltsverzeichnis etc.), Literaturverzeichnis, Kapitel, Verweise, `\input` und `\include`
- BENUTZEREIGENE STRUKTUREN: eigene Befehle definieren, Variablen definieren, Variablen manipulieren, Umgebungen definieren  
  
Dort auch: Zur Verallgemeinerung der Umgebungen `itemize`, `enumerate` und `description` existiert die Umgebung `list`. Man kann sich hier alle Parameter für vom Standard abweichende, aufzählungsähnliche Textgestaltung definieren.
- KOMA-SCRIPT: Briefe, ...
- Pakete mit außergewöhnlichen Symbolen, Zeichen und Schriftarten (Wasyfamily, Zapfdingbats)
- Zeichnen und konstruieren in L<sup>A</sup>T<sub>E</sub>X (Metapost, pictex, PSTricks)
- Pakete fancyunits, ziffer

⌘ \_\_\_\_\_

Wünsche für weitere Kapitel: .....

.....

Bugreports: .....

.....

Sonstiges: .....

.....