

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

**Institut für Informatik  
TU Clausthal**

**WS 08/09**

**Dozent:** Prof. Dr. Sven Hartmann

**Zeit und Ort:** Vorlesung 14.+21. 11., jeweils 13-15 Uhr

Homepage

[http://zach.in.tu-clausthal.de/teaching/  
werkzeuge/index.html](http://zach.in.tu-clausthal.de/teaching/werkzeuge/index.html)

Dort befinden sich wichtige Informationen über die Vorlesung, Dokumente, Übungsserien etc.

# Übersicht

1. Von T<sub>E</sub>X zu L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>
2. Arbeiten mit L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>
3. Weiterführendes

# Literatur

-  L. Lamport: *Das L<sup>A</sup>T<sub>E</sub>X-Handbuch*. Addison-Wesley Deutschland (1995), ISBN 3-89319-826-1 .
-  M. Goossens, F. Mittelbach und A. Samarin: *Der L<sup>A</sup>T<sub>E</sub>X-Begleiter*. Addison Wesley Longman, 2. korr. Nachdruck (1996), ISBN 3-89319-646-3 .
-  D. E. Knuth: *Computers & Typesetting, Vol. A: The T<sub>E</sub>X Book*. Addison-Wesley (1991), ISBN 0-201-13447-0 .
-  N. Schwarz: *Einführung in T<sub>E</sub>X–incl. Version 3.0*. Oldenbourg, 3. Aufl. (1991), ISBN 3-486-24349-7 .

# Kapitel 1. Von T<sub>E</sub>X zu L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

Eine kurze Geschichte von T<sub>E</sub>X

Was braucht man?

Aufbau eines L<sup>A</sup>T<sub>E</sub>X Dokumentes

## Der Inhalt dieses Abschnittes:

- ▶ Unterschiede von  $\text{T}_{\text{E}}\text{X}$  und  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$ .
- ▶ **Markup-** versus WYSIWYG-Systeme.
- ▶ Grundsätzlicher Aufbau eines  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$  Dokumentes.

## Woher kommt T<sub>E</sub>X?

T<sub>E</sub>X: (sprich „Tech“, geschrieben `\TeX`) ist ein rechnerunabhängiges, public domain Textsystem (1977) von Donald Knuth [3, 4]. Es dient zum Setzen von Texten und mathematischen Formeln.

L<sup>A</sup>T<sub>E</sub>X : (sprich „Lah-tech“, geschrieben `\LaTeX`) ist ein Makropaket (1980), das auf T<sub>E</sub>X aufbaut. Es wurde von Leslie Lamport [1, 2] geschrieben.

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> : (sprich „Lah-tech zwei e“, geschrieben `\LaTeXe`) ist die aktuelle Variante von L<sup>A</sup>T<sub>E</sub>X seit dem 1. Juni 1994. (Vorherige hieß L<sup>A</sup>T<sub>E</sub>X 2.09.)

# Markup vs WYSIWYG

**Autor:** Sie! Die **logische Struktur des Textes** muss angegeben werden. **Nicht die Details von Gestaltung und Satz.**

**Designer:**  $\text{\LaTeX 2}_{\epsilon}$  . **Designen ist sehr schwierig!** Man überläßt es besser Leuten, die was davon verstehen.

**Setzer:**  $\text{\TeX}$  . Setzen eines Textes ist ebenfalls schwierig. **Da lassen wir ganz die Hände davon.**  $\text{\TeX}$  ist eine relativ komplizierte Programmiersprache.

Der typische Ablauf beim Arbeiten mit  $\text{\LaTeX}$  ist:

1. Ein Eingabefile `document.tex` schreiben.
2. Dieses File mit  $\text{\LaTeX}2_{\epsilon}$  bearbeiten; dabei **wird eine Datei erzeugt**, die den gesetzten Text in einem geräteunabhängigen Format (dvi, pdf, ps) enthält.
3. Einen „Probeausdruck“ davon auf dem Bildschirm anzeigen (Preview).
4. **Die Eingabe korrigieren** und zurück zu Schritt 2.
5. Die Ausgabedatei drucken.

Texteditor und Preview-Programme sind oft aufeinander abgestimmt.

## Editor, T<sub>E</sub>X-System, Previewer

**Editor:** Zum Beispiel **WinEdt**, **TeXnicCenter**, **XEmacs**, **Kile**. Wichtig ist, daß Dateien als ASCII Dateien gespeichert werden: **keine Steuerzeichen oder andere interne Zeichen**.

**T<sub>E</sub>X-System:** Zum Beispiel **MikTeX** (unter Windows), **TeX** (Linux, Mac).

**Previewer:** Zum Beispiel **xdvi** (dvi-files), **ghostview** (ps-files), **acrobat** (pdf-files).

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

- ▶ Eine L<sup>A</sup>T<sub>E</sub>X Datei ist eine **ASCII Datei**.
- ▶ Man unterscheidet zwischen normalem Text und **L<sup>A</sup>T<sub>E</sub>X-Befehlen**.
- ▶ **Jeder String der mit einem „\“ anfängt, ist ein L<sup>A</sup>T<sub>E</sub>X- (oder ein T<sub>E</sub>X-) Befehl.**

---

```
\documentclass{article}
\begin{document}
  Small is beautiful.
\end{document}
```

---

Abbildung: Eine minimale L<sup>A</sup>T<sub>E</sub>X-Datei

---

```
\documentclass[11pt,a4paper]{article}
\date{14. November 2008}
\author{Tom und Jerry}
\title{Warum?}

\begin{document}
\maketitle
\begin{abstract}
Einige Gedanken                zu aktuellen Textsystemen.
\end{abstract}
\tableofcontents
\section{Start}\label{sect:Start}
Word ist nicht    die Antwort.\\
Word ist die     Frage.\\
Und die Antwort lautet "‘Nein’’! \dots\\
\end{document}
```

---

Abbildung: Aufbau eines Artikels

## Leerzeichen, Absätze

- ▶ **Mehrere** Leerzeichen werden wie **ein** Leerzeichen behandelt. „\ “ steht für ein Leerzeichen, das auf jeden Fall gesetzt wird.
- ▶ Eine Leerzeile zwischen Textzeilen bedeutet das Ende eines **Absatzes**. **Mehrere** Leerzeilen werden wie **eine** Leerzeile behandelt.
- ▶ Ein **Absatz** steht für einen **zusammenhängenden Gedanken** bzw. eine von anderen abgegrenzte Idee.

## Leerzeichen nach $\LaTeX$ -Befehlen

- ▶ `\today` gibt das momentane Datum aus. **Nach einem  $\LaTeX$ -Befehl wird kein Leerzeichen gesetzt** (man muss es explizit anzeigen).

Heute ist der 1. Mai 2008.

Oder: Heute ist der 1. Mai 2008.

Falsch: Am 1. Mai 2008regnet es.

Richtig: Am 1. Mai 2008 scheint  
die Sonne.

Oder: Am 1. Mai 2008 schneit es.

Heute ist der `\today`.

Oder: Heute ist der `\today` .

Falsch: Am `\today` regnet es.

Richtig: Am `\today{}` scheint\  
die Sonne.

Oder: Am `\today\` schneit es.

## Kommentare in T<sub>E</sub>X Dateien

- ▶ Alles, was hinter einem Prozentzeichen % steht (bis zum Ende der Eingabezeile), wird von L<sup>A</sup>T<sub>E</sub>X ignoriert. Dies kann für **Notizen** des Autors verwendet werden.

Das ist einBeispiel.

```
Das ist ein% dummes  
% Besser: ein lehrreiches!  
Beispiel.
```

# Präambel

- ▶ Die erste Zeile einer L<sup>A</sup>T<sub>E</sub>X Datei:

```
\documentclass[optionen]{klasse}
```

- ▶ *klasse* **muss**,
- ▶ *optionen* **kann**

vorhanden sein. Siehe Tabelle 1 auf Seite 40 und Tabelle 2 auf Seite 41.

- ▶ Alles was jetzt kommt gehört zur **Präambel**.
- ▶ Hier hört die Präambel auf:

```
\begin{document}
```

- ▶ Die letzte Zeile einer L<sup>A</sup>T<sub>E</sub>X Datei:

```
\end{document}
```

## Tabelle: Dokumentklassen

---

**article** für Artikel in wissenschaftlichen Zeitschriften, kürzere Berichte u. v. a.

**report** für längere Berichte, die aus mehreren Kapiteln bestehen, Diplomarbeiten, Dissertationen u. ä.

**book** für Bücher.

**scrartcl, scrreprt, scrbook** Die sog. KOMA-Klassen sind Varianten der o. g. Klassen mit besserer Anpassung an DIN-Papierformate und „europäische“ Typographie.

**letter** für Briefe.

---

Tabelle: Klassenoptionen (Mehrere Mögl. sind durch | getrennt)

---

**10pt|11pt|12pt:** wählt die normale Schriftgröße des Dokuments aus. 10pt hohe Schrift ist die Voreinstellung.

**a4paper:** für Papier im DIN A4-Format. Ohne diese Option nimmt L<sup>A</sup>T<sub>E</sub>X amerikanisches Papierformat an.

**titlepage|notitlepage:** legt fest, ob Titel und Zusammenfassung auf einer eigenen Seite erscheinen sollen. `titlepage` ist die Voreinstellung für die Klassen `report` und `book`.

**onecolumn|twocolumn:** für ein- oder zweispaltigen Satz. Die Voreinstellung ist immer `onecolumn`. Die Klassen `letter` und `slides` kennen *keinen* zweispaltigen Satz.

**oneside|twoside:** legt fest, ob die Seiten für ein- oder zweiseitigen Druck gestaltet werden sollen. `oneside` ist die Voreinstellung für alle Klassen außer `book`.

---

## Makropakete (packages)

Mit dem Befehl

```
\usepackage[optionen]{pakete}
```

können in der Präambel **ergänzende Makropakete (packages)** geladen werden. Sie bestimmen das Layout der Dokumentklasse oder stellen zusätzliche Funktionalität bereit. Siehe Tabelle 3 auf Seite 43.

Ein Beispiel:

```
\usepackage{latexsym,alltt,  
            graphicx,hyperref}  
\usepackage[ngerman]{babel}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}
```

## Tabelle: Pakete (eine Auswahl)

---

- alltt** Definiert eine Variante der `verbatim`-Umgebung
- amsmath, amssymb** Mathematischer Formelsatz
- babel** Anpassungen für viele verschiedene Sprachen. Die gewählten Sprachen werden als Optionen angegeben.
- fontenc** Erlaubt die Verwendung von Schriften mit unterschiedlicher Kodierung
- german, ngerman** Anpassungen für die deutsche Sprache in traditioneller und neuer Rechtschreibung.
- graphicx** Einbindung von extern erzeugten Graphiken.
- hyperref** Ermöglicht Hyperlinks.
- inputenc** Deklaration der Zeichenkodierung im Eingabefile.
- latexsym** Erlaubt einige besondere Symbole wie  $\square$ , die mit  $\text{\LaTeX}$  2.09 standardmäßig verfügbar waren.
- longtable** für Tabellen über mehrere Seiten mit automatischem Seitenumbruch.
-

# Normale und Sonderzeichen

**Normale Zeichen:** Folgende Zeichen können zur Eingabe von Text verwendet werden:

*a...z A...Z 0...9*  
. : ; , ? ! ' ' ( ) [ ] - / \* @  
+ =

**Sonderzeichen:** Die folgenden Zeichen haben für  $\text{\LaTeX}$  eine **Spezialbedeutung** oder sind nur innerhalb von mathematischen Formeln erlaubt:

$\$ \& \% \# \_ \{ \} \sim \wedge \" \backslash | < >$

## Und was ist mit Umlauten?

Mit `\usepackage[codepage]{inputenc}` kann man auch Sonderzeichen **direkt in das Eingabefile** schreiben.

Mögliche Angaben für *codepage* sind u. a.:

**latin1** Latin-1 (ISO 8859-1), gebräuchlich unter Unix und VMS

**latin9** Latin-9 (ISO 8859-15), Erweiterung von Latin-1, u. a. mit Eurozeichen

**ansinew** Microsoft Codepage 1252 für Windows

**cp850** IBM Codepage 850, üblich unter OS/2

**applemac** Macintosh-Kodierung

Die häßliche Straße muß  
schöner werden.

Die häßliche Straße  
muß schöner werden.

## Darstellbare Zeichen

- ▶ Der in der **Ausgabe** darstellbare Zeichenvorrat hängt nicht vom **Eingabe** Zeichenvorrat ab.
- ▶ Für jedes überhaupt darstellbare Zeichen – also auch diejenigen, die nicht im Zeichensatz des jeweiligen Betriebssystems enthalten sind – gibt es einen  $\LaTeX$ -Befehl oder eine Ersatzdarstellung, die ausschließlich mit ASCII-Zeichen auskommt.

## Kompilieren einer `.tex` Datei

Der Aufruf von  $\text{\LaTeX}$  ( $\text{\PDF\LaTeX}$ ) für eine Datei `sample.tex` erzeugt

- .dvi:** eine Datei `sample.dvi` bei Aufruf von  $\text{\LaTeX}$ ,
- .pdf:** eine Datei `sample.pdf` bei Aufruf von  $\text{\PDF\LaTeX}$ ,
- .aux:** eine Datei `sample.aux`, mit Informationen über die Struktur der Datei,
- .log:** eine Datei `sample.log`, mit Fehlermeldungen etc.

## Kompilieren einer .tex Datei (2)

Nach dem Aufruf von  $\text{\LaTeX}$  ( $\text{PDF}\text{\LaTeX}$ ) kann die Datei `sample.aux` von weiteren Programmen benutzt werden:

**BIB $\text{\TeX}$** :  $\text{BIB}\text{\TeX}$  erzeugt (mit entsprechenden Anweisungen in `sample.tex`) Dateien `sample.blg` und `sample.bbl` mit der zitierten Literatur.

**Makeindex**: `Makeindex` erzeugt Dateien `sample.idx`, `sample.ilg` und `sample.ind`: die letzte stellt den erzeugten Index der ursprünglichen Datei dar.

**Danach muss  $\text{\LaTeX}$  ( $\text{PDF}\text{\LaTeX}$ ) noch zweimal aufgerufen werden**, denn die erzeugten Dateien (`sample.bbl` bzw. `sample.idx`) werden berücksichtigt.

# Kapitel 2. Arbeiten mit L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

Seitenaufbau

Tabellensatz

Math-Modus

Graphiken

Referenzieren

## Der Inhalt dieses Abschnittes:

- ▶ Seitenaufbau und Tabellensatz.
- ▶ Wie setzt man **mathematische Formeln** in  $\text{\LaTeX} 2_{\epsilon}$  ?
- ▶ Graphikeinbindung (pdf, ps, etc) in  $\text{\LaTeX} 2_{\epsilon}$  Dokumente.
- ▶ **Referenzieren** auf Seiten, Definitionen etc.

**Blocksatz:** Normaler Text wird im **Blocksatz**, d. h. mit Randausgleich gesetzt.  $\text{\LaTeX}$  führt den Zeilen- und Seitenumbruch **automatisch** durch. Dabei wird für jeden **Absatz** die bestmögliche Aufteilung der Wörter auf die Zeilen bestimmt.

**Absatz:** Wie die Absätze gesetzt werden, hängt von der Dokumentklasse ab: Die Klassen `article`, `report` und `book` kennzeichnen Absätze durch **Einrücken** der ersten Zeile; die Klasse `letter` beispielsweise läßt stattdessen zwischen den Absätzen einen **kleinen vertikalen Abstand**.

## Zeilen-, Seitenumbruch

- ▶ Für Ausnahmefälle kann man den Umbruch beeinflussen: Der Befehl `\` oder `\newline` bewirkt einen **Zeilenwechsel ohne neuen Absatz**, der Befehl `\`\* einen Zeilenwechsel, bei dem kein Seitenwechsel erfolgen darf.
- ▶ Der Befehl `\newpage` bewirkt einen **Seitenwechsel**.
- ▶ Mit den Befehlen `\linebreak[n]`, `\nolinebreak[n]`, `\pagebreak[n]` und `\nopagebreak[n]` kann man angeben, ob an bestimmten Stellen ein Zeilen- bzw. Seitenwechsel eher günstig oder eher ungünstig ist, wobei  $n$  die Stärke der Beeinflussung angibt (1, 2, 3 oder 4).

**Zeilenumbruch:** L<sup>A</sup>T<sub>E</sub>X bemüht sich, den Zeilenumbruch besonders schön zu machen. Eventuell sind Zeilen zu lang (**Fehlermeldung** `overfull hbox`). Insbesondere, wenn keine geeignete Stelle für die Silbentrennung gefunden wird.

**sloppy:** Innerhalb der `sloppy`-Umgebung ist L<sup>A</sup>T<sub>E</sub>X generell weniger streng: Wortabstände werden stärker – notfalls auch unschön – vergrößert.

**Silbentrennung:** Der Befehl `\hyphenation` bewirkt, daß die darin angeführten Wörter jedesmal an den und **nur an den mit - markierten Stellen** abgeteilt werden können. Eignet sich **nur** für Wörter, die **keine** Umlaute, scharfes s, Ziffern oder sonstige Sonderzeichen enthalten.

# Trennungen

```
\hyphenation{ Eingabe-file  
Eingabe-files FORTRAN }
```

Der Befehl \- innerhalb eines Wortes bewirkt, daß dieses Wort dieses eine Mal nur an den mit \- markierten Stellen oder unmittelbar nach einem Bindestrich abgeteilt werden kann. Dieser Befehl eignet sich für **alle** Wörter, auch für solche, die Umlaute, scharfes s, Ziffern oder sonstige Sonderzeichen enthalten.

Eingabe, L<sup>A</sup>T<sub>E</sub>X-Eingabe-  
file, ein langes Wort, Häß-  
lichkeit

Eingabe,  
\LaTeX-Eingabe\ -file,  
ein langes Wort,  
H"a\ss{\}\ -lich\ -keit

## `\mbox{...}` und `\fbox{...}`

`\mbox{...}` bewirkt, daß das Argument überhaupt nicht abgeteilt werden kann.

Die Nummer ist schon lange nicht mehr (02 22) 56 36.

`\mbox{filename}` gibt den Dateinamen an.

Die Nummer ist schon lange nicht mehr

`\mbox{(02\,22) 56\,36}`. `\fbox{\textit{filename}}` gibt den Dateinamen an.

Der Befehl `\fbox{...}` ist wie `\mbox{...}` nur wird ein Rahmen drumherum gesetzt.

## `\makebox{...}` und `\framebox{...}`

Der Befehl `\makebox[width][position]{text}` ist ähnlich wie `\mbox{...}`, nur hat die Box eine feste Länge „*width*“. *position* kann „c“ (centered), „l“ (linksbündig), „r“ (rechtsbündig) oder „s“ (stretch).

Hier kommt 

eine Box, eine Box,
---------------------

 und 

hier eine zweite Box,
-----------------------

 und jetzt ist  
Schluß.

Hier kommt `\framebox[3cm][t]{eine Box, eine Box,}`  
und `\framebox[4cm][r]{hier eine zweite Box, }`  
und jetzt ist Schluß.

Der Befehl `\framebox{...}` ist wie `\makebox{...}` nur wird ein Rahmen drumherum gesetzt.

## Die minipage Umgebung

Der Befehl `\begin{minipage}[position]{width}text`  
`\end{minipage}` erlaubt, im Unterschied zu den boxes auf  
der vorigen Folie, **beliebig viele Zeilen**, die im Blocksatz  
gesetzt werden. Die minipage hat eine feste Länge „*width*“.  
*position* kann „t“ (top), oder „b“ (bottom) sein.

```

                                noch eine Minipage,
                                noch eine Minipage,
                                noch eine Minipage,
                                noch eine Minipage,
Hier kommt eine Minipage, eine Minipage, und noch eine ... und jetzt ist Schluß.
                                eine Minipage, eine Minipage,
                                eine Minipage, eine Minipage,
                                eine Minipage, eine Minipage,
                                ...
```

```
Hier kommt \begin{minipage}[t]{.3\textwidth} eine Minipage,
eine Minipage, \dots \end{minipage} und
\begin{minipage}[b]{.2\textwidth} noch eine Minipage, noch
eine Minipage, \dots \end{minipage} und jetzt ist Schluß.
```

# Abstände

**Variieren:** L<sup>A</sup>T<sub>E</sub>X variiert die Leerstellen zwischen den Wörtern. Nach Punkten, Fragezeichen u. a., wird ein etwas größerer Abstand erzeugt.

**Spaces:** L<sup>A</sup>T<sub>E</sub>X nimmt an, daß Punkte, die auf einen Großbuchstaben folgen, eine Abkürzung bedeuten. Ausnahmen:

„\ “: Leerstelle darf nicht verbreitert werden darf.

„~“: Leerstelle ohne Zeilenwechsel.

„\,“: kurzer Abstand (z. B. in Abkürzungen).

„\:“: mittlerer Abstand.

„\!“: kurzer negativer Abstand.

„\@“: Wenn ein Punkt einen Satz beendet, obwohl davor ein Großbuchstabe steht.

Das betrifft u.a. auch die wissenschaftl. Mitarbeiter.

Das betrifft u.\,a.\ auch die wissenschaftl.\ Mitarbeiter. \\

Noch immer wohnt Dr. Partl im 1. Stock.

Noch immer wohnt Dr.~Partl im 1.~Stock. \\

Noch immer wohnt Dr. Partl im 1. Stock.

Noch immer wohnt Dr.\ Partl im 1.~Stock. \\

... 5cm breit.

\dots\ 5\,cm breit. \\

Ich brauche Vitamin C. Du nicht?

Ich brauche Vitamin~C\@. Du nicht?

# Anführungszeichen

- ▶ Für Anführungszeichen ist **nicht** das auf Schreibmaschinen übliche Zeichen (") zu verwenden.
- ▶ Im Buchdruck werden für öffnende und schließende Anführungszeichen jeweils **verschiedene** Zeichen bzw. Zeichenkombinationen gesetzt.
- ▶ **Öffnende** Anführungszeichen, wie sie im amerikanischen Englisch üblich sind, erhält man durch Eingabe von **zwei Grave-Akzenten**, **schließende** durch **zwei Apostrophe**.

“No,” he said, „I really don't know!”

‘ ‘No, ’ ’ he said,  
‘ ‘I really don't know! ’ ’

Bei Benutzung des Paketes `german` stehen die folgenden Befehle für deutsche Anführungszeichen zur Verfügung:

`"‘` Doublequote und Grave-Akzent,

`"’` Doublequote und Apostroph,

„Nein,“ log er uns an, „ich weiß nichts!“      `"‘Nein,“’` log er uns an,  
`"‘ich wei\ss{} nichts!“’`

## Binde-, Gedanken-strich, minus

O-Beine

10-18 Uhr

Paris-Dakar

Schalke 04 – Hertha BSC

ja – oder nein?

yes—or no?

0, 1 und –1

0-Beine

10--18~Uhr

Paris--Dakar

Schalke 04 -- Hertha BSC

ja -- oder nein?

yes---or no?

0, 1 und \$-1\$

## Punkte und Ligaturen

Nicht so ... sondern so:  
Wien, Graz, ...

Nicht so ... sondern so: \\  
Wien, Graz, \dots

Nicht Auflage (Au-fl-age)  
sondern Auflage (Auf-lage)

Nicht Auflage (Au-fl-age) \\  
sondern Auf\\/lage (Auf-lage)

## Sonderzeichen

\$	\\$
&	\&
%	\%
#	\#
_	\_
{	\{
}	\}
~	\textasciitilde
^	\textasciicircum
\	\textbackslash
	\textbar
<	\textless
>	\textgreater

Tabelle: Akzente und spezielle Buchstaben

ò	\'o	ó	\'o	ô	\^o
õ	\~o	ō	\=o	ò	\.o
õ	\u o	ǒ	\v o	ố	\H o
ö	\"o	ơ	\c o	ọ	\d o
ơ	\b o	ơ	\r o	ồ	\t oo
œ	\oe	Œ	\OE	æ	\ae
Æ	\AE	å	\aa	Å	\AA
ø	\o	Ø	\O	ı	\l
Ł	\L	ı	\i	Ј	\j
ß	\ss				

### Table: Symbole

i	!‘	†	\dag	™	\texttrademark
¿	?‘	‡	\ddag	.	\textperiodcentered
§	\S	¶	\P	•	\textbullet
£	\pounds	©	\copyright	®	\textregistered

## Textaufbau

Der Beginn eines Kapitels bzw. Unterkapitels und seine Überschrift werden mit Befehlen der Form `\section...` angegeben. Dabei muß die logische Hierarchie eingehalten werden.

Bei der Klasse `article`:

```
\section \subsection \subsubsection
```

Bei den Klassen `report` und `book`:

```
\chapter \section \subsection \subsubsection
```

Artikel können also relativ einfach als Kapitel in ein Buch eingebaut werden. Die Abstände zwischen den Kapiteln, die Numerierung und die Schriftgröße der Überschrift werden von  $\text{\LaTeX}$  automatisch bestimmt.

# Titelseite

- Titelseite:** Die Überschrift des gesamten Artikels bzw. die Titelseite des Schriftstücks wird mit dem Befehl `\maketitle` gesetzt.
- Autor etc.:** Der Inhalt muß vorher mit den Befehlen `\title`, `\author` und `\date` vereinbart werden (vgl. Abbildung 2 auf Seite 26).
- Inhaltsverzeichnis:** Der Befehl `\tableofcontents` bewirkt, daß ein Inhaltsverzeichnis ausgedruckt wird. Es wird dabei eine Datei mit der Endung `.toc` generiert.

# Fußnoten

Fußnoten<sup>1</sup> werden automatisch numeriert und am unteren Ende der Seite ausgedruckt.

```
Fußnoten\footnote{Das ist  
eine Fußnote.} werden ...
```

---

<sup>1</sup>Das ist eine Fußnote.

## Hervorhebungen

*Werden innerhalb eines hervorgehobenen Textes nochmals Passagen hervorgehoben, so setzt L<sup>A</sup>T<sub>E</sub>X (beachte L<sup>A</sup>T<sub>E</sub>X ) diese in einer aufrechten Schrift.*

```
\emph{Werden innerhalb eines  
hervorgehobenen Textes  
\emph{nochmals} Passagen  
hervorgehoben, so setzt  
\LaTeX{ } (beachte  
\emph{\LaTeX}) diese  
in einer  
\emph{aufrechten} Schrift.}
```

## Hochgestellter Text

Hochgestellten Text in passender Größe generiert folgender Befehl:

```
\textsuperscript{text}
```

le 2<sup>ième</sup> régime

```
le 2\textsuperscript{i\'eme}  
r\'egime
```

## Nicht-Blocksatz Umgebungen

Die Kennzeichnung von speziellen Textteilen, die anders als im normalen Blocksatz gesetzt werden sollen, erfolgt mittels sogenannter Umgebungen (environments):

```
\begin{name}  
  text  
\end{name}
```

Umgebungen sind *Gruppen*. Sie können auch ineinander geschachtelt werden:

```
\begin{aaa}  
  ... \begin{bbb} ... \end{bbb} ...  
  
\end{aaa}
```

## quote-Umgebung

Die quote-Umgebung eignet sich für kürzere Zitate, hervorgehobene Sätze und Beispiele. Der Text wird links und rechts eingerückt.

Eine Faustregel für die Zeilenlänge lautet:

*Keine Zeile soll mehr als ca. 66 Buchstaben enthalten.*

Daher werden in Zeitungen mehrere Spalten nebeneinander verwendet.

Eine Faustregel für die Zeilenlänge lautet:

```
\begin{quote}
Keine Zeile soll mehr als
ca. \ 66~Buchstaben
enthalten.
```

```
\end{quote}
```

Daher werden in Zeitungen mehrere Spalten nebeneinander verwendet.

# Listen

---

- |   |  |
|---|--|
| ▶ Bei <code>itemize</code> werden die Elemente durch Punkte/Symbole gekennzeichnet. | <pre>\begin{itemize}</pre>             |
| ▶ Listen kann man auch verschachteln:   | <pre>\item Bei \texttt{itemize}</pre>  |
| ▶ Maximale Tiefe ist 4.   | <pre>\item Listen kann man</pre>       |
| ▶ Bezeichnung und Einrückung wechseln automatisch.                                  | <pre>  \begin{itemize}</pre>           |
| ▶ usw.  | <pre>  \item Die maximale ...</pre>    |
|   | <pre>  \item Bezeichnung und ...</pre> |
|   | <pre>  \end{itemize}</pre>             |
|   | <pre>\item usw.</pre>                  |
|   | <pre>\end{itemize}</pre>               |
- 

Abbildung: Beispiel für `itemize`

## Listen: numeriert

---

- |   |   |
|---|---|
| 1. Bei <code>enumerate</code> wird numeriert. | <code>\begin{enumerate}</code>            |
| 2. Numerierung ist automatisch.               | <code>\item Bei \texttt{enumerate}</code> |
| 3. Verschachtelte Listen:                     | <code>\item Numerierung ...</code>        |
| 3.1 Maximale Tiefe ist 4.                     | <code>\item Verschachtelte Listen:</code> |
| 3.2 Bezeichnung/Einrückung ist automatisch.   | <code>\begin{enumerate}</code>            |
| 4. usw.                                       | <code>\item Die maximale ...</code>       |
|   | <code>\item Bezeichnung und ...</code>    |
|   | <code>\end{enumerate}</code>              |
|   | <code>\item usw.</code>                   |
|   | <code>\end{enumerate}</code>              |
- 

Abbildung: Beispiel für `enumerate`

## description-Umgebung

---

Kleine Tierkunde:

**Gelse:** ein kleines Tier,  
das

**Gemse:** ein großes Tier,  
das

**Gürteltier:** ein mittleres  
Tier, das

Kleine Tierkunde:

```
\begin{description}
\item[Gelse:] ein kleines
Tier, das ...
\item[Gemse:] ein gro\sses
Tier, das ...
\item[G"urteltier:] ein
mittleres Tier, das ...
\end{description}
```

---

Abbildung: Beispiel für description

## Links, rechts, Mitte

links  
Backbord

```
\begin{flushleft}  
links \\  
Backbord  
\end{flushleft}
```

rechts  
Steuerbord

```
\begin{flushright}  
rechts \\  
Steuerbord  
\end{flushright}
```

Im  
Reich  
der  
Mitte

```
\begin{center}  
Im \\ Reich \\ der \\ Mitte  
\end{center}
```

## verbatim und verb

Zwischen `\begin{verbatim}` und `\end{verbatim}` stehende Zeilen werden **genauso ausgedruckt, wie sie eingegeben wurden** (mit allen Leerzeichen, Zeilenwechselln und **ohne Interpretation** von  $\text{\LaTeX}$ -Befehlen).

Innerhalb eines Absatzes kann man dies auch mit `\verb|` und `|` tun. Mit diesen Befehlen wurden z. B. alle  $\text{\LaTeX}$ -Befehle in der vorliegenden Beschreibung gesetzt.

Der `\dots`-Befehl ...

Der `\verb|\dots|`-Befehl `\dots`

Die `verbatim`-Umgebung und der Befehl `\verb` dürfen *nicht* innerhalb von Parametern von anderen Befehlen verwendet werden.

In der `tabbing`-Umgebung kann man Tabulatoren setzen. Der Befehl `\=` setzt eine Tabulatorposition, `\kill` bedeutet, daß die „Musterzeile“ nicht ausgedruckt werden soll, `\>` springt zur nächsten Tabulatorposition, und `\\` trennt die Zeilen.

links	Mittelteil	rechts	<code>\begin{tabbing}</code>
Es			<code>war einmal\quad \=</code>
war einmal	und ist	nicht mehr	<code>Mittelteil\quad \= \kill</code>
ein		ausgestopfter	<code>links \&gt; Mittelteil \&gt; rechts\\</code>
		Teddybär	<code>Es \\</code>
			<code>war einmal \&gt; und ist</code>
			<code>\&gt; nicht mehr\\</code>
			<code>ein \&gt; \&gt; ausgestopfter\\</code>
			<code>\&gt; \&gt; Teddyb"ar</code>
			<code>\end{tabbing}</code>

## Horizontale Abstände

*`\hspace{länge}`*

- ▶ Wenn der Abstand auch am Beginn oder Ende einer Zeile erhalten bleiben soll, muß `\hspace*` statt `\hspace` geschrieben werden.
- ▶ Die Längenangabe besteht im einfachsten Fall aus einer Zahl und einer Einheit. Die wichtigsten Einheiten sind in Tabelle 6 angeführt.

**Tabelle:** Einheiten für Längenangaben

---

mm	Millimeter
cm	Zentimeter = 10 mm
in	inch = 25.4 mm
pt	point = $(1/72.27)$ in $\approx 0.351$ mm
bp	big point = $(1/72)$ in $\approx 0.353$ mm
em	Geviert (doppelte Breite einer Ziffer der akt. Schrift)
ex	Höhe des Buchstabens x der aktuellen Schrift

---

Die Befehle in Tabelle 7 sind Abkürzungen zum Einfügen besonderer horizontaler Abstände.

**Tabelle:** Befehle für horizontale Abstände

---

<code>\,</code>	ein sehr kleiner Abstand
<code>\enspace</code>	so breit wie eine Ziffer
<code>\quad</code>	so breit, wie ein Buchstabe hoch ist
<code>\qqquad</code>	doppelt so breit wie ein <code>\quad</code>
<code>\hfill</code>	Abstand, der sich von 0 bis $\infty$ ausdehnen kann.

---

Der Befehl `\hfill` kann dazu dienen, einen vorgegebenen Platz auszufüllen.

Schafft mir    Raum!



Schafft

mir`\hspace{.5cm}`Raum! `\\`



`(\triangleleft)\hfill`

`(\triangleright)`

## Vertikale Abstände

`\vspace{länge}`

Dieser Befehl sollte immer zwischen zwei Leerzeilen angegeben werden. Wenn der Abstand auch am Beginn oder Ende einer Seite erhalten bleiben soll, muß `\vspace*` statt `\vspace` geschrieben werden. Die Befehle in Tabelle 8 sind Abkürzungen für bestimmte vertikale Abstände.

**Tabelle:** Befehle für vertikale Abstände

---

<code>\smallskip</code>	etwa $\frac{1}{4}$ Zeile
<code>\medskip</code>	etwa $\frac{1}{2}$ Zeile
<code>\bigskip</code>	etwa 1 Zeile
<code>\vfill</code>	Abstand, der sich von 0 bis $\infty$ ausdehnt

---

Die tabular-Umgebung dient zum Setzen von Tabellen.  
Im **Parameter** des Befehls `\begin{tabular}{...}` wird das **Format der Tabelle** angegeben. Dabei bedeutet

- ▶ l eine Spalte mit linksbündigem Text,
- ▶ r eine mit rechtsbündigem,
- ▶ c eine mit zentriertem Text,
- ▶ `p{breite}` eine Spalte der angegebenen Breite mit mehrzeiligem Text,
- ▶ | einen senkrechten Strich.

Innerhalb der Tabelle bedeutet

- ▶ & den Sprung in die nächste Tabellenspalte,
- ▶ `\\` trennt die Zeilen,
- ▶ `\hline` (an Stelle einer Zeile) setzt einen waagrechten Strich.

7C0	hexa
3700	oktal
11111000000	binär
1984	dezimal

```
\begin{tabular}{|r|l|}  
\hline  
7C0 & hexa \\  
3700 & oktal \\  
11111000000 & bin"ar \\  
\hline\hline  
1984 & dezimal \\  
\hline  
\end{tabular}
```

## Math-Modus

Mathematische Textteile innerhalb eines Absatzes werden zwischen `\(` und `\)` oder zwischen `$` und `$` oder zwischen `\begin{math}` und `\end{math}` eingeschlossen.

## Was sind mathematische Texte?

Seien  $a$  und  $b$  die Katheten und  $c$  die Hypotenuse, dann gilt  $c^2 = a^2 + b^2$  (Satz des Pythagoras).

Seien `\(a\)` und `\(b\)` die Katheten und `\(c\)` die Hypotenuse, dann gilt `\(c^2=a^2+b^2\)` (Satz des Pythagoras).

$\TeX$  spricht man wie  $\tau\epsilon\chi$  aus.

Mit ♥-lichen Grüßen

`\TeX\` spricht man wie `\(\tau\epsilonpsilon\chi\)` aus.\\ Mit `\(\heartsuit\)`-lichen Gr"u"sen

Größere mathematische Formeln setzt man in eigene Zeilen.

**Mit Gleichungsnummer:** zwischen `\begin{equation}` und `\end{equation}`.

**Ohne Gleichungsnummer:** zwischen `\begin{displaymath}` und `\end{displaymath}` oder zwischen `\[` und `\]`.

Seien  $a$  und  $b$  die Katheten und  $c$  die Hypotenuse, dann gilt

$$c = \sqrt{a^2 + b^2} \quad (1)$$

(Satz des Pythagoras).

Seien  $a$  und  $b$  die Katheten und  $c$  die Hypotenuse, dann gilt

```
\begin{equation}
c = \sqrt{ a^{2}+b^{2} }
\end{equation}
```

(Satz des Pythagoras).

Hier einer der kürzesten  
(Mathematiker-) Witze  
überhaupt:

$$\varepsilon \not\leq 0 \quad (2)$$

Aus (2) folgt ...

Hier einer der k"urzesten  
(Mathematiker-) Witze  
"uberhaupt:

```
\begin{equation} \label{eps}
\varepsilon \lneqq 0
\end{equation}
```

Aus (\ref{eps}) folgt \dots

Das Setzen im mathematischen Modus unterscheidet sich vom Text-Modus vor allem durch folgende Punkte:

- ▶ **Leerzeilen sind verboten** (Mathematische Formeln müssen innerhalb eines Absatzes stehen).
- ▶ **Leerstellen und Zeilenwechsel haben bei der Eingabe keine Bedeutung**, alle Abstände werden nach der Logik der mathematischen Ausdrücke automatisch bestimmt oder müssen durch spezielle Befehle wie `\`, oder `\quad` angegeben werden.

$$\forall x \in \mathbb{R} : x^2 \geq 0 \quad (3)$$

```
\begin{equation}
\forall x \in \mathbb{R} :
\quad x^2 \geq 0
\end{equation}
```

- ▶ Jeder einzelne Buchstabe wird als Name einer Variablen betrachtet und entsprechend gesetzt (kursiv mit zusätzlichem Abstand). Will man innerhalb eines mathematischen Textes normalen Text (in aufrechter Schrift, mit Wortabständen) setzen, muß man diesen in `\textnormal{...}` einschließen.

$$x^2 \geq 0 \quad \text{für } x \in \mathbb{R} \quad (4)$$

```
\begin{equation}
x^{2} \geq 0\quad
\textnormal{für }
x \in \mathbb{R}
\end{equation}
```

Kleine **griechische Buchstaben** werden als `\alpha`, `\beta`, `\gamma`, `\delta`, usw. eingegeben, große griechische Buchstaben als `\Gamma`, `\Delta`, usw.

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

`(\lambda, \xi, \pi, \mu,`  
`\Phi, \Omega \)`

Es gibt eine Fülle von **mathematischen Symbolen**: von  $\in$  über  $\Rightarrow$  bis  $\infty$ .

Neben der voreingestellten Kursivschrift für die Variablen bietet  $\text{\LaTeX}$  eine Auswahl von mathematischen **Alphabeten** an:

$ABCabc$	$\backslash(\backslashmathrm{ABCabc})\backslash$
<b><math>ABCabc</math></b>	$\backslash(\backslashmathbf{ABCabc})\backslash$
$ABCabc$	$\backslash(\backslashmathsf{ABCabc})\backslash$
$ABCabc$	$\backslash(\backslashmathtt{ABCabc})\backslash$
<i><math>ABC</math></i>	$\backslash(\backslashmathcal{ABC})\backslash$

Die kalligraphischen Buchstaben ( $\backslash\mathcal$ ) gibt es nur als Großbuchstaben. Mit dem Paket `amsymb` stehen auch Alphabete für Mengenzeichen und Frakturschrift zur Verfügung.

**Exponenten und Indizes** können mit den Zeichen  $\wedge$  und  $\_$  hoch- bzw. tiefgestellt werden.

$a_1$	$x^2$	$e^{-\alpha t}$	$a_{ij}^3$	<code>\(a_{1}\)</code> \quad <code>\(x^{2}\)</code> <code>\quad \(\e^{-\alpha t}\)</code> <code>\quad \(\a^{3}_{ij}\)</code>
-------	-------	-----------------	------------	--

Das **Wurzelzeichen** wird mit `\sqrt`,  $n$ -te Wurzeln werden mit `\sqrt[n]` eingegeben. Die Größe des Wurzelzeichens wird von L<sup>A</sup>T<sub>E</sub>X automatisch gewählt.

$\sqrt{x}$	$\sqrt{x^2 + \sqrt{y}}$	$\sqrt[3]{2}$	<code>\(\sqrt{x}\)</code> \quad <code>\(\sqrt{ x^{2}+\sqrt{y} }\)</code> <code>\quad \(\sqrt[3]{2}\)</code>
------------	-------------------------	---------------	---

Die Befehle `\overline` und `\underline` bewirken **waagrechte Striche** direkt über bzw. unter einem Ausdruck.

$$\overline{m+n}$$

```
\(\overline{m+n}\)
```

Die Befehle `\overbrace` und `\underbrace` bewirken **waagrechte Klammern** über bzw. unter einem Ausdruck.

$$\underbrace{a+b+\cdots+z}_{26}$$

```
\(\underbrace{a+b+\cdots+z}_{26}\)
```

Tabelle: Klammern

(	(	)	)	[	\lceil	]	\rceil
<	\langle	>	\rangle	[	\lfloor	]	\rfloor
[	[	]	]	{	\{	}	\}
			\	\	\		

Um mathematische **Akzente** wie Pfeile oder Schlangen auf Variablen zu setzen, gibt es die in Tabelle 10 auf Seite 169 angeführten Befehle. Längere Tilden und Dacherln, die sich über mehrere (bis zu 3) Zeichen erstrecken können, erhält man mit `\widetilde` bzw. `\widehat`. Ableitungszeichen werden mit ' (Apostroph) eingegeben.

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

```
\begin{displaymath}
y=x^{2} \quad \quad y'=2x
\quad \quad y''=2
\end{displaymath}
```

**Tabelle:** Mathematische Akzente

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

Tabella: Sonstige Symbole (\* benötigt Paket latexsym)

$\aleph$	<code>\aleph</code>	$'$	<code>\prime</code>	$\forall$	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>
$\hbar$	<code>\hbar</code>	$\emptyset$	<code>\emptyset</code>	$\exists$	<code>\exists</code>	$\exists$	<code>\exists</code>
$\imath$	<code>\imath</code>	$\nabla$	<code>\nabla</code>	$\neg$	<code>\neg</code>	$\neg$	<code>\neg</code>
$\jmath$	<code>\jmath</code>	$\surd$	<code>\surd</code>	$\flat$	<code>\flat</code>	$\flat$	<code>\flat</code>
$\ell$	<code>\ell</code>	$\top$	<code>\top</code>	$\natural$	<code>\natural</code>	$\natural$	<code>\natural</code>
$\wp$	<code>\wp</code>	$\perp$	<code>\perp</code>	$\sharp$	<code>\sharp</code>	$\sharp$	<code>\sharp</code>
$\Re$	<code>\Re</code>	$\diamond$	<code>\Diamond*</code>	$\clubsuit$	<code>\clubsuit</code>	$\clubsuit$	<code>\clubsuit</code>
$\Im$	<code>\Im</code>	$\square$	<code>\Box*</code>	$\diamond$	<code>\diamondsuit</code>	$\diamond$	<code>\diamondsuit</code>
$\partial$	<code>\partial</code>	$\triangle$	<code>\triangle</code>	$\heartsuit$	<code>\heartsuit</code>	$\heartsuit$	<code>\heartsuit</code>
$\infty$	<code>\infty</code>	$\angle$	<code>\angle</code>	$\spadesuit$	<code>\spadesuit</code>	$\spadesuit$	<code>\spadesuit</code>
$\mho$	<code>\mho*</code>						

Mathematische **Funktionen** werden nicht kursiv (wie die Namen von Variablen), sondern in „normaler“ Schrift dargestellt. In  $\text{\LaTeX}$  gibt es folgende Befehle:

$\backslash$ arccos	$\backslash$ cos	$\backslash$ csc	$\backslash$ exp	$\backslash$ ker	$\backslash$ limsup	$\backslash$ min
$\backslash$ sinh	$\backslash$ arcsin	$\backslash$ cosh	$\backslash$ deg	$\backslash$ gcd	$\backslash$ lg	$\backslash$ ln
$\backslash$ Pr	$\backslash$ sup	$\backslash$ arctan	$\backslash$ cot	$\backslash$ det	$\backslash$ hom	$\backslash$ lim
$\backslash$ log	$\backslash$ sec	$\backslash$ tan	$\backslash$ arg	$\backslash$ coth	$\backslash$ dim	
$\backslash$ inf	$\backslash$ liminf	$\backslash$ max	$\backslash$ sin	$\backslash$ tanh		

Für die Modulo-Funktion gibt es zwei verschiedene Befehle:  
`\bmod` für den binären Operator  $a \bmod b$  und `\pmod{...}` für  
die Angabe in der Form  $x \equiv a \pmod{b}$ .

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
\begin{displaymath}
\lim_{x \to 0}
\frac{\sin x}{x} = 1
\end{displaymath}
```

Ein **Bruch** (fraction) wird mit dem Befehl `\frac{...}{...}` gesetzt. Für einfache Brüche kann man aber auch den Operator `\/` verwenden.

$1\frac{1}{2}$  Stunden

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

```
\(1\frac{1}{2}\)~Stunden
\begin{displaymath}
\frac{ x^{2} }{k+1} \quad \quad
x^{\frac{2}{k+1}} \quad \quad
x^{ 1/2 }
\end{displaymath}
```



## Summen, Integrale

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \int_{-\infty}^{+\infty}$$

```
\begin{displaymath}
  \sum_{i=1}^n \quad
  \int_0^{\frac{\pi}{2}}
  \quad
  \int_{-\infty}^{+\infty}
\end{displaymath}
```

## Klammern

Runde und eckige Klammern können mit den entsprechenden Tasten eingegeben werden, geschwungene mit `\{`.  
Setzt man den Befehl `\left` vor öffnende Klammern und den Befehl `\right` vor schließende, so wird automatisch die richtige Größe gewählt.

$$1 + \left( \frac{1}{1 - x^2} \right)^3$$

```
\begin{displaymath}
1+\left( \frac{1}{1-x^{2}} \right) ^3
\end{displaymath}
```

In manchen Fällen möchte man die Größe der Klammern lieber selbst festlegen, dazu sind die Befehle `\bigl`, `\Bigl`, `\biggl` und `\Biggl` anstelle von `\left` und analog `\bigr` etc. anstelle von `\right` anzugeben.

$$\left( (x+1)(x-1) \right)^2$$

```
\begin{displaymath}
\Bigl( (x+1)(x-1) \Bigr)^{2}
\end{displaymath}
```

Um in Formeln **3 Punkte** (z. B. für  $1, 2, \dots, n$ ) auszugeben, gibt es die Befehle `\ldots` und `\cdots`. `\ldots` setzt die Punkte auf die Grundlinie (low), `\cdots` setzt sie in die Mitte der Zeilenhöhe (centered). Außerdem gibt es die Befehle `\vdots` für vertikal und `\ddots` für diagonal angeordnete Punkte.

		<code>\begin{displaymath}</code>
$x_1, \dots, x_n$	$x_1 + \dots + x_n$	<code>x_{1}, \ldots, x_{n} \quad \backslash\mathrm{quad}</code>
		<code>x_{1}+\cdots+x_{n}</code>
		<code>\end{displaymath}</code>

Für **Matrizen** u. ä. gibt es die `array`-Umgebung, die ähnlich wie die `tabular`-Umgebung funktioniert. Der Befehl `\\` trennt die Zeilen.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

```
\begin{displaymath}
\mathbf{X} = \left(
\begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array}
\right)
\end{displaymath}
```

## Zu lange Formeln

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (5)$$

```
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!}
        + \frac{x^5}{5!} - {}
        \nonumber \\
& & - \frac{x^7}{7!} + \dots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (6)$$

```
\begin{eqnarray}
\lefteqn{\cos x = 1}
        - \frac{x^2}{2!} + {}
        \nonumber \\
& & + \frac{x^4}{4!}
        - \frac{x^6}{6!} + \dots
\end{eqnarray}
```

- ▶ Lädt man im Vorspann des Dokuments das Paket `graphicx`, dann kann man Bilder, die mit einem anderen Programm erzeugt wurden, mit dem Befehl `\includegraphics` in das Dokument einfügen.
- ▶ Der Dateiname ist als Argument des Befehls anzugeben. Welche Dateiformate verarbeitet werden können, hängt vom jeweiligen  $\text{T}_{\text{E}}\text{X}$ -System ab und muß in dessen Dokumentation beschrieben sein.

Hier ist ein Bild:

```
\includegraphics [width=1cm]  
{a.pdf}.
```

Hier ist ein Bild:  .

Wird das Paket `graphicx` mit der Option `[draft]` geladen, dann erscheint anstelle des Bildes nur ein Rahmen entsprechend der tatsächlichen Bildgröße mit dem Namen des Grafikfiles (Probeausdrucke).

Hier ist ein Bild:  .

Hier ist ein Bild:

```
\includegraphics [width=1cm]  
{a.pdf}.
```

**Label:** Mit den Befehlen `\label` und `\ref` ist es möglich, die von  $\text{\LaTeX}$  automatisch vergebenen Kapitelnummern im Text anzusprechen.

```
\section{Algorithmen}
... Der Beweis findet sich
in Abschnitt~\ref{bew}.
```

```
.
.
.
```

```
\section{Beweise} \label{bew}
...
```

**Label:** Mit den Befehlen `\label` und `\pageref` ist es möglich, die von L<sup>A</sup>T<sub>E</sub>X automatisch vergebenen Seitenzahlen im Text anzusprechen.

```
\section{Algorithmen}
... Der Beweis findet sich auf
Seite~\pageref{bew} ...
```

.  
.
.

```
\section{Beweise} \label{bew}
...
```

- ▶ Große Bilder und lange Tabellen lassen sich nicht immer genau dort unterbringen, wo sie inhaltlich hingehören, weil sie nicht mehr vollständig auf die aktuelle Seite passen, aber auch nicht durch einen Seitenwechsel zerrissen werden sollen.
- ▶ Um solche Strukturen automatisch an eine geeignete Stelle „gleiten“ zu lassen, kennt L<sup>A</sup>T<sub>E</sub>X die beiden Umgebungen `figure` und `table`.

## figure

Diese Umgebung ist für die Behandlung von **Abbildungen** gedacht: Alles, was zwischen `\begin{figure}` und `\end{figure}` steht, wird automatisch an eine Stelle gesetzt, wo es komplett hinpaßt, ohne durch einen Seitenwechsel zerrissen zu werden.

- ▶ Mit `\caption{...}` setzt man die Bezeichnung der Abbildung. Dabei ist nur der Text anzugeben, das Wort „Abbildung“ und die fortlaufende Nummer werden von  $\text{\LaTeX}$  hinzugefügt.
- ▶ Bei Abbildungen ist es allgemein üblich, die Bezeichnung *unter* das Bild zu setzen.
- ▶ Mit `\label` und `\ref` kann man die Nummer der Abbildung im Text ansprechen, mit `\pageref` ihre Seitenzahl.
- ▶ Der Befehl `\label` muß dabei *nach* dem `\caption`-Befehl stehen, sonst stimmt die Numerierung nicht!

- ▶ Im folgenden Beispiel wird einfach mit dem Befehl `\vspace` leerer Raum für ein später einzusetzendes Bild gelassen:

Abbildung 7 auf S. 204 zeigt ein Beispiel aus der Minimal art.

Abbildung: Im Nebel

Abbildung~\ref{weiss} auf S.~\pageref{weiss} zeigt ein Beispiel aus der Minimal art.

```
\begin{figure}[tb]
\vspace{1cm}
\caption{Im Nebel}
\label{weiss}
\end{figure}
```

Abbildung: Landschaft im Nebel

- ▶  $\LaTeX$  kann eine Abbildung nach verschiedenen Kriterien plazieren: `h` „here“ (hier), `t` „top“ (oben auf der Seite), `b` „bottom“ (unten auf der Seite) oder `p` „page“ (eigene Seite für Abbildungen).
- ▶ Die optionalen Parameter, dienen dazu, die Platzierung der Abbildung auf die angegebenen Orte *einzuschränken* .
- ▶ Durch Angabe von z. B. `tb` wird  $\LaTeX$  angewiesen, nur eine Platzierung oben oder unten auf der Seite zu versuchen, je nachdem, wo *zuerst* eine passende Stelle gefunden wird.
- ▶ Werden keine Parameter (und keine eckigen Klammern!) angegeben, ist die Voreinstellung `tbp`, also ohne `h`.