

Fachschaft Physik

# Handout zum L<sup>A</sup>T<sub>E</sub>X-Kurs

Januar/Februar 2007

## Teil 3: weitere Spezialgebiete

---

Dieses Skript dient als Handout zum L<sup>A</sup>T<sub>E</sub>X-Kurs der Fachschaft Physik an der Universität Konstanz. Der Kurs und das Skript sind über mehrere Generationen engagierter Fachschaftler entstanden und weiterentwickelt worden. Bisher haben folgende Personen zu diesem Werk beigetragen: VOLKER DOBLER, BERND RINN, FRANK BICKENDORF, TOBIAS MÜTHER, JÖRG WERNER, ROLAND HACKL, JENS DORFMÜLLER, OLIVER GRÄSER, TIMO BÖHM, DANIEL TRÄUTLEIN, CLAUDIUS RIEK, FRANZISKA MAIER UND MARCEL WUNRAM. Die Autoren erheben keinen Anspruch auf Vollständigkeit und Fehlerfreiheit. Lob, Kritik und Anregungen bitte per Mail an: [Fachschaft.Physik@uni-konstanz.de](mailto:Fachschaft.Physik@uni-konstanz.de). Als pdf-File darf dieses Skript frei kopiert werden. **Viel Spaß mit L<sup>A</sup>T<sub>E</sub>X!**

## Inhaltsverzeichnis

<b>1</b>	<b>Bilder &amp; Tabellen</b>	<b>3</b>
1.1	Tabellen erstellen . . . . .	3
1.2	Bilder einbinden . . . . .	4
1.3	Gleitobjekte . . . . .	4
1.4	Das floatflt-Paket . . . . .	6
1.4.1	floatingfigure . . . . .	6
1.4.2	floatingtable . . . . .	7
1.5	Aufgaben . . . . .	8
<b>2</b>	<b>Literaturverzeichnisse und Indizes</b>	<b>9</b>
2.1	Literaturverzeichnisse . . . . .	9
2.2	BibT <sub>E</sub> X . . . . .	9
2.3	Indizes . . . . .	10
<b>3</b>	<b>L<sup>A</sup>T<sub>E</sub>X konfigurieren</b>	<b>11</b>
3.1	Neue Befehle definieren . . . . .	11
3.2	Neue Umgebungen definieren . . . . .	12
3.3	Eigene Pakete erstellen . . . . .	12
3.4	Zähler . . . . .	12
3.5	Längen . . . . .	13
3.6	Horizontaler/Vertikaler Leerraum . . . . .	14
3.7	Stretch . . . . .	14
3.8	Seitenlayout . . . . .	15
3.9	Aufgaben . . . . .	17
<b>4</b>	<b>Andere Dokumentklassen</b>	<b>18</b>
4.1	Die $\mathcal{A}\mathcal{M}\mathcal{S}$ T <sub>E</sub> X-Klassen . . . . .	18
4.1.1	Der <code>\newtheorem</code> -Befehl . . . . .	18
4.2	Briefe mit <code>scr<sub>l</sub>ttr2</code> . . . . .	20
<b>5</b>	<b>Das beamer-Paket</b>	<b>23</b>
5.1	Der Kopf eines beamer-Dokuments . . . . .	23
5.2	Eine einzelne Folie – die <code>frame</code> -Umgebung . . . . .	25
5.3	Ein Beispiel für eine beamer-Präsentation . . . . .	25
<b>6</b>	<b>pdfL<sup>A</sup>T<sub>E</sub>X</b>	<b>26</b>
6.1	hyperref . . . . .	26
6.2	Beispiel . . . . .	27
<b>7</b>	<b>Installation zusätzlicher Pakete in L<sup>A</sup>T<sub>E</sub>X</b>	<b>27</b>
7.1	Der einfache Weg.... . . . . .	28
7.1.1	<code>apt-get</code> unter debian . . . . .	28
7.1.2	Der MikTeX Update Wizard . . . . .	28
7.2	Installationsbäume . . . . .	28
7.2.1	MikTeX . . . . .	28
7.2.2	TeX <sub>E</sub> X . . . . .	29

## *Inhaltsverzeichnis*

---

7.2.3 Benutzerspezifischer T <sub>E</sub> X-Pfad . . . . .	29
7.3 Installation der Pakete . . . . .	29
<b>8 Dokumentation</b>	<b>30</b>
<b>Literatur</b>	<b>31</b>

# 1 Bilder & Tabellen

## 1.1 Tabellen erstellen

Tabellen werden in L<sup>A</sup>T<sub>E</sub>X mit der `tabular`-Umgebung erstellt. Die `tabular`-Umgebung benötigt ein weiteres Argument, das das Format der Tabelle beschreibt. Dabei steht ein `l` für eine linksbündig ausgerichtete Spalte, ein `r` für eine rechtsbündig ausgerichtete Spalte und ein `c` für eine zentriert ausgerichtete Spalte. L<sup>A</sup>T<sub>E</sub>X bestimmt selber die Breite der einzelnen Spalten. Mit `p{Breite}` kann man eine Spalte mit einer bestimmten *Breite* und Text in Blocksatz erzeugen. Vertikale Linien erzeugt man mit einem `|`.

Innerhalb der Tabellenumgebung springt man mit `&` in die nächste Spalte und mit `\\` in die nächste Zeile. Horizontale Linien können mit dem Befehl `\hline` erzeugt werden, diese befinden sich immer *oberhalb* der Zeile, in der der Befehl aufgerufen wird.

links	zentriert	rechts
l	z	r

```
\begin{tabular}{|l|c|r|}
\hline
links & zentriert & rechts \\
\hline
l & z & r \\
\hline
\end{tabular}
```

Dies ist eine einfache Zelle mit einer Breite von 5cm. Bei sehr schmalen Zellen hat L <sup>A</sup> T <sub>E</sub> X manchmal Probleme die Zeilen umzubrechen!
---

```
\begin{tabular}{|p{5cm}|}
\hline
Dies ist eine einfache Zelle mit einer Breite von 5cm. Bei sehr schmalen Zellen hat \LaTeX{} manchmal Probleme die Zeilen umzubrechen!\\
\hline
\end{tabular}
```

Für Linien, die nur von Spalte *i* bis Spalte *j* gehen, gibt es den Befehl `\cline{i-j}`. Mit dem Befehl `\multicolumn{n}{Format}{Text}` kann man *n* Spalten zu einer zusammenfassen.

Name		
Vorname	Nachname	Tel. Nr
Jens	Dorf Müller	0123/1234
Roland	Hackl	0123/1234

```
\begin{tabular}{|l|l|l|}
\cline{1-2}
\multicolumn{2}{|c|}{Name} \\
\hline
Vorname & Nachname & Tel. Nr \\
\hline \hline
Jens & Dorf Müller & 0123/1234 \\
\hline
Roland & Hackl & 0123/1234 \\
\hline
\end{tabular}
```

Der Befehl `@{Zeichen}` in der Format-Beschreibung unterdrückt den Spaltenzwischenraum und fügt stattdessen ein *Zeichen* ein.

Auf folgende Art und Weise kann man Zahlen an einem Komma ausrichten:

$\pi$ -Ausdruck	Wert	
$\pi$	3,1415	<code>\begin{tabular}{cr@{,}l}</code>
$\pi^\pi$	36,46	<code>  \$\pi\$-Ausdruck &amp;</code>
$\pi^{\pi^\pi}$	8066,7	<code>  \multicolumn{2}{c}{Wert} \\</code>
		<code>  \hline</code>
		<code>  \$\pi\$ &amp; 3&amp;1415 \\</code>
		<code>  \$\pi^\pi\$ &amp; 36&amp;46 \\</code>
		<code>  \$\pi^{\pi^\pi}\$ &amp; 8066&amp;7 \\</code>
		<code>\end{tabular}</code>

D1AF20BFA4563F3E4AB2359F11

## 1.2 Bilder einbinden

Grafiken und Bilder werden meistens mit einer externen Software erstellt und später in das L<sup>A</sup>T<sub>E</sub>X-Dokument eingebunden. Hierfür gibt es in L<sup>A</sup>T<sub>E</sub>X mehrere Möglichkeiten, die vor allem davon abhängen, ob man `latex` oder `pdflatex` verwendet. Ersteres erzeugt einen Output im Vektorformat, letzteres nicht. Daher verlangt `latex` auch zwingend Vektorgrafiken im EPS-Format (= Encapsulated PostScript), `pdflatex` dagegen akzeptiert verschiedene Formate wie PNG, JPG, PDF oder GIF, *EPS aber nicht mehr*. Wir empfehlen aus einer Reihe von Gründen, `pdflatex` zu verwenden. Sollte man noch Restbestände an EPS-Grafiken haben, können diese u.a. mit `ps2pdf` umgewandelt werden. Gegebenenfalls muss man anschließend aber die Ränder der Figur neu einstellen bzw. beschneiden.

Um Grafiken einbinden zu können, wird das `graphicx`-Paket benötigt. Es wird mit dem Befehl `\usepackage[Treiber]{graphicx}` eingebunden. Der *Treiber* hängt vom Ausgabeprogramm ab. Meistens wird `pdftex` (oder bei EPS/latex *dvips*) verwendet. Die Bilder werden mit `\includegraphics[Option=Wert,...]{Datei}` Befehlen eingefügt. Folgende Optionen sind erlaubt:

- `width` um die Breite der Grafik festzulegen
- `height` um die Höhe der Grafik festzulegen
- `angle` um die Grafik gegen den Uhrzeigersinn zu drehen
- `scale` um die Grafik zu Vergrößern

```
\includegraphics[scale=1.5]{hallo}
```

# Hallo Welt!

## 1.3 Gleitobjekte

Bei Grafiken und Tabellen ergibt sich für L<sup>A</sup>T<sub>E</sub>X das Problem, dass beide nicht auf eine neue Seite umgebrochen werden können. Wenn viele Grafiken verwendet werden, entstehen viele nur teilweise gefüllte Seiten. Um dies zu Vermeiden kann L<sup>A</sup>T<sub>E</sub>X Grafiken auf die nächste Seite „fließen“ lassen und den Rest der Seite mit

Text füllen. Dazu bietet L<sup>A</sup>T<sub>E</sub>X zwei Umgebungen an: die `table`-Umgebung für Tabellen und die `figure`-Umgebung für Grafiken.

Beide Umgebungen besitzen einen optionalen Parameter mit dem L<sup>A</sup>T<sub>E</sub>X mitgeteilt wird, welche Positionen für die Grafik erlaubt sind.

Eine `table`-Umgebung könnte z.B. mit dem Befehl `\begin{table} [!hbp]` beginnen. In diesem Fall könnte die Tabelle an der aktuellen Stelle des Textes (**h**), am unteren Ende einer Seite (**b**) oder auf einer speziellen Tabellen-Seite stehen.

<code>h</code>	<i>here</i>	an der Stelle im Text, wo die Umgebung steht. Funktioniert sehr gut bei kleinen Gleitobjekten.
<code>t</code>	<i>top</i>	am Anfang einer Seite
<code>b</code>	<i>bottom</i>	am Ende einer Seite
<code>p</code>	<i>page</i>	auf eine extra Seite
<code>!</code>		unbedingt, L <sup>A</sup> T <sub>E</sub> X soll interne Variablen, die dies verhindern könnten, missachten. Meist ist das Programm allerdings zu renitent, um sich auch daran zu halten.

Beim Verschieben von Gleitobjekten ist L<sup>A</sup>T<sub>E</sub>X mitunter sehr freizügig. Um dem Problem, daß eine Figur aus Kapitel 3 auf einmal in Kapitel fünf auftaucht, abzuhelpfen, verwendet man den Befehl `\clearpage`. Dieser beendet die aktuelle Seite und erzwingt auf den folgenden Seiten die Ausgabe aller noch nicht verarbeiteten Gleitobjekte.

Mit dem Befehl `\caption{...}` können Bildunterschriften erzeugt werden. Außerdem nimmt L<sup>A</sup>T<sub>E</sub>X das Gleitobjekt dann in das Tabellen- bzw. Abbildungsverzeichnis auf. Das Tabellenverzeichnis kann mit dem Befehl `\listoftables`, das Abbildungsverzeichnis mit `\listoffigures` generiert werden.

Lange Bildunterschriften machen das Tabellen- und Abbildungsverzeichnis sehr lang. Mit einem optionalen Argument kann man L<sup>A</sup>T<sub>E</sub>X eine Kurzversion der Bildunterschrift geben:

```
\caption[Kurzversion]{Lange Version einer Bildunterschrift}
```

# Hallo Welt!

Abbildung 1: Grafik in der „Hallo Welt!“ steht.

## Abbildungsverzeichnis

`\listoffigures`

1	„Hallo Welt“-Grafik . . . . .	5
2	Beamer-Beispiel 1 . . . . .	24
3	Beamer-Beispiel . . . . .	24
4	Beamer-Beispiel 3 . . . . .	24
5	Beamer-Beispiel 4 . . . . .	24

## 1.4 Das floatflt-Paket

Mit den standardmäßigen Gleitumgebungen kann man Gleitobjekte lediglich als ganzspaltige Objekte setzen. Dies sieht oft nicht gut aus, denn gerade in einspaltigen Dokumenten nehmen kleine Grafiken oft sehr viel Platz weg. Auf Gleitumgebungen zu verzichten ist aber dennoch nicht ratsam, da nur diese ins Abbildungsverzeichnis aufgenommen werden und auch nur auf Gleitobjekte verwiesen werden kann.

Eine Lösung hier bietet das `floatflt`-Paket. Es stellt die Umgebungen `floatingfigure` und `floatingtable` zur Verfügung, in denen jeweils Grafiken beziehungsweise Tabellen als textumflossene Gleitobjekte gesetzt werden können,

### 1.4.1 floatingfigure

`floatingfigure` unterscheidet sich in der Syntax nahezu überhaupt nicht von der normalen `figure`-Umgebung. Es wird lediglich eine Option und ein weiterer Parameter übergeben. Der Befehl `\floatingfigure[option]{l}` erzeugt ein Gleitobjekt der Breite `l`, welches je nach Ausrichtung rechts oder links von Text umflossen wird. Als Option für die Ausrichtung stehen zur Verfügung

- r** Das Gleitobjekt wird an den rechten Seitenrand gesetzt
- l** Das Gleitobjekt wird an den linken Rand gesetzt
- p** Das Gleitobjekt wird an den äußeren Rand gesetzt. Dies ist bei gerade Seiten der linke Rand, bei ungeraden Seiten der rechte. Diese Einstellung macht nur bei doppelseitigem Druck Sinn.

Wird keine Ausrichtungsoption angegeben, nimmt L<sup>A</sup>T<sub>E</sub>X die bei Aufruf des Paketes angegebene – fehlt diese, wird am äußeren Rand gesetzt.



Dies ist ein großes Phi. Es wurde geT<sub>E</sub>Xt, ausgeschnitten und hier als Grafik eingebunden. Nun fließt Text drum. Dumm nur, daß in unseren Beispielen kein Platz für eine Unterschrift ist

```
\begin{floatingfigure}[l]{3cm}
\includegraphics[scale=6]{phi}
\end{floatingfigure}
Dies ist ein gro"ses Phi. Es wurde
ge\TeX{t}, ausgeschnitten und
hier als Grafik eingebunden.
Nun flie"st Text drum. Dumm nur,
da"s in unseren Beispielen
kein Platz f"ur eine Unterschrift
ist
```

### 1.4.2 floatingtable

`floatingtable` arbeitet im Prinzip genau wie `floatingfigure`, es gibt nur einen wichtigen Unterschied: Und zwar bekommt die Umgebung keine Breite als Parameter übergeben, sondern gleich die ganze Tabelle, von `\begin{tabular}` bis `\end{tabular}`. In der eigentlichen Umgebung selbst findet sich dann nur noch – gegebenenfalls – der `caption`-Befehl für eine Bildunterschrift.

Dies ist kein Phi, sondern eine Tabelle. Sieht doch jeder.

```
lkldfsljhlkd dfsjklkjfd fdsjk-
kjd hnfkdjkljkhdsf jksdfkljldfs
jsdfk jnfkdjklkjfd sdfjksdf
dfsjkjfsd jksdfjlkfd dfsjk sdf-
jkkkjdsf sdfjkl
```

Dies war reichlich sinnloser Text, der war gerade im Angebot

l.o.	r.o.
l.u.	r.u.

Tabelle 1: T1

Dies ist kein Phi, sondern eine Tabelle. Sieht doch jeder.

```
\begin{floatingtable}[r]{
\begin{tabular}{l|r}
l.o. & r.o. \\ \hline
l.u. & r.u. \\
\end{tabular}}
\caption{T1}
\end{floatingtable}
lkldfsljhlkd dfsjklkjfd fdsjkkjd
hnfkdjkljkhdsf jksdfkljldfs jsdfk
jnfkdjklkjfd sdfjksdf dfsjkjfsd
jksdfjlkfd dfsjk sdfjkkkjdsf sdfjkl
Dies war reichlich sinnloser
Text, der war gerade im Angebot
```



## 1.5 Aufgaben

### Aufgabe 1

Erstelle folgende Tabelle:

ITEM	PRICE	
gnats	gram	\$13.65
	each	.01
gnu	stuffed	92.50
emu		33.33
armadillo	frozen	8.99

Dabei sollte auf die Ausrichtung in den einzelnen Spalten, Spalten die über mehrere Spalten gehen, durchgezogene und nichtdurchgezogene Linien geachtet werden. Die Tabelle besteht aus drei Spalten (keine Ausrichtung am Dezimalpunkt nötig).

### Aufgabe 2

Lade die Datei `tigerente.jpg` von der Internet-Webseite des L<sup>A</sup>T<sub>E</sub>X-Kurses (<http://fachschaft.physik.uni-konstanz.de/fachschaft/aktionen/latex/>) herunter und probiere folgende `\includegraphics`-Kommandos aus:

```
\includegraphics[angle=90,height=1cm]{tigerente}
\includegraphics[height=1cm,angle=90]{tigerente}
```

Warum sehen die beiden Bilder so unterschiedlich aus?

### Aufgabe 3

Binde die Tabelle aus Aufgabe 1 und die beiden Grafiken aus Aufgabe 2 in Gleitobjekt-Umgebungen ein und probiere die verschiedenen Positionierungsparameter aus<sup>1</sup>.

Probiere auch das Erstellen von Bildunterschriften und das Generieren von Abbildungs- und Tabellenverzeichnissen aus.

<sup>1</sup>Um die Platzierung der Umgebungen ausprobieren zu können, musst du die Gleitobjekte in einen Text einfügen, z. B. den aus dem Kursteil L<sup>A</sup>T<sub>E</sub>X-Grundlagen.

## 2 Literaturverzeichnisse und Indizes

### 2.1 Literaturverzeichnisse

Wenn man in Texten auf andere Werke verweisen möchte, bietet sich an dies am Ende des Dokuments mit der `thebibliography`-Umgebung zu machen. Diese Umgebung erwartet ein weiteres Argument, mit dem die maximale Breite der Label übergeben wird. Es bietet sich an, einfach den breitesten Eintrag zu nehmen.<sup>2</sup> Jeder Eintrag in der Bibliographie beginnt dort mit dem Befehl `\bibitem{Marker}`. `\bibitem` funktioniert ansonsten genau so wie der Befehl `\item`. Möchte man Einfluß darauf haben wie der Verweis später dargestellt wird, kann man ein optionales Argument voranstellen.

Um sich auf die Werke in der Bibliographie zu beziehen, fügt man im Text den Befehl `\cite{Marker}` ein. Wenn man nur auf einige Seiten eines Werkes beziehen möchte, kann man ein optionales Argument voranstellen.

siehe [1, Seite 21–23].  
siehe [For88].

```
siehe~\cite[Seite 21--23]{Fischer}. \\
siehe~\cite{Forster}.
```

### Literatur

[1] Fischer, *Lineare Algebra*, Vieweg,  
1988, 1. Auflage

[For88] Forster ...

```
% Am Ende des Dokuments:
\begin{thebibliography}{For88}
  \bibitem{Fischer} Fischer,
    \emph{Lineare Algebra}, Vieweg,
    1988, 1.~Auflage
  \bibitem[For88]{Forster} Forster ...
\end{thebibliography}
```

### 2.2 BibT<sub>E</sub>X

Wenn häufiger Literaturverzeichnisse verwendet werden, bietet es sich an das BibT<sub>E</sub>X-Paket zu verwenden. Es wird oft mit T<sub>E</sub>X mitgeliefert. Es ist damit möglich eine Literatur-Datenbank anzulegen. In das Literatur-Verzeichnis werden dann aber nur die wirklich verwendeten Werke übernommen.

Viele Lehrstühle haben sich eigene Datenbanken mit den wichtigsten Werken erstellt. Als Beispiel für so eine Datenbank kommt hier meine Datei `latex.bib`:

```
@BOOK{Kopka:1988,
  LANGUAGE={german},
  AUTHOR={Helmut Kopka},
  TITLE={\LaTeX --- Eine Einf{\u}hrung},
  PUBLISHER={Addison--Wesley},
  ADDRESS={Bonn Reading, Mass.[u. a.]},
  YEAR={1988}}
@BOOK{Kopka:1997-2,
  LANGUAGE={german},
```

<sup>2</sup>Wenn man die Einträge einfach nur durchnummerieren läßt, sollte man „9“ oder „99“ verwenden. Wenn man die Bücher mit den ersten drei Buchstaben des Autors und den letzten beiden Ziffern der Jahreszahl abkürzt, bietet sich an „MMM99“ zu nehmen.

```

AUTHOR={Helmut Kopka},
TITLE={\LaTeX: Erg{\ä}nzungen},
PUBLISHER={Addison--Wesley--Longman},
ADDRESS={Bonn Reading, Mass. [u. a.]},
EDITION={2., {\ü}berarb.},
YEAR={1997}}
@BOOK{Kopka:1997-3,
LANGUAGE={german},
AUTHOR={Helmut Kopka},
TITLE={\LaTeX: Erweiterungen},
PUBLISHER={Addison--Wesley--Longman},
ADDRESS={Bonn Reading, Mass. [u. a.]},
YEAR={1997}}
@BOOKLET{Oetiker:2002,
LANGUAGE={english},
AUTHOR={Tobias Oetiker},
TITLE={The Not So Short Introduction to {\LaTeXe}},
HOWPUBLISHED={\url{http://people.ee.ethz.ch/~oetiker/lshort/}},
ADDRESS={Z{\ü}rich},
MONTH=dec,
YEAR={2002}}

```

So wird die Datei `latex.bib` eingebunden.

```

[?] \cite{Oetiker:2002}

\bibliographystyle{geralpha}
\bibliography{latex}

```

Nach dem ersten Kompilieren der Datei muss der Befehl `bibtex Dateiname.aux` ausgeführt werden. Damit man den Befehl `\bibliographystyle{geralpha}` verwenden kann, muss man das Paket `bibgerm` laden. Es gibt noch andere `\bibliographystyle`-Stile.

Mehr über Bib<sub>T</sub>E<sub>X</sub> findet man in [3] und [2].

## 2.3 Indizes

Bei Büchern und längeren Skripten ist ein Index mit den wichtigsten Begriffen sehr nützlich. Mit L<sup>A</sup>T<sub>E</sub>X und dem Programm `makeindex` kann man auch das sehr einfach machen.

Dazu muss mit dem Befehl `\usepackage{makeidx}` das Index-Paket geladen werden. Mit dem Befehl `\makeindex` im Kopf der T<sub>E</sub>X-Datei wird beim Compilieren eine `.idx`-Datei erzeugt.

Die einzelnen Index-Einträge werden mit dem Befehl `\index{Eintrag}` festgelegt.

Nach dem ersten Compilieren kann dann mit dem Befehl `makeindex Dateiname.idx` der Inhalt des Index sortiert werden. Mit dem Befehl

`\printindex` im T<sub>E</sub>X-Dokument kann dann der Inhalt des Index eingefügt werden. Mit folgenden Befehlen kann man verschiedene Formatierungen des Index erreichen:

Beispiel	Index Eintrag	Kommentar
<code>\index{hallo}</code>	hallo, 1	einfacher Eintrag
<code>\index{hallo!Peter}</code>	Peter, 3	Untereintrag unter „hallo“
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	formatierter Eintrag
<code>\index{Lin@\textbf{Lin}}</code>	<b>Lin</b> , 7	formatierter Eintrag
<code>\index{alpha@\$\alpha\$}</code>	$\alpha$ , 12	mathematisches Zeichen
<code>\index{Jenny textbf}</code>	Jenny, <b>3</b>	formatierte Seitenzahl
<code>\index{Joe textit}</code>	Joe, <i>5</i>	formatierter Seitenzahl
<code>\index{Samuel see{Sam}}</code>	Samuel, <i>siehe</i> Sam	Verweis auf anderen Eintrag

Weitergehende Informationen zum Thema Index finden sich in [3].

### 3 L<sup>A</sup>T<sub>E</sub>X konfigurieren

#### 3.1 Neue Befehle definieren

Machmal benötigt man eine lange Kombination von Befehlen oder Zeichen mehrmals im Dokument. Dabei kann man sich das T<sub>E</sub>X vereinfachen indem man neue Befehle definiert.

Mit `\newcommand{Name}[Zahl]{Definition}` definiert man ein neues Kommando *Name* mit *Zahl* Argumenten. Die *Zahl* ist dabei ein optionales Argument. Wenn sie weggelassen wird, hat der Befehl keine Argumente.

Die einzelnen Argumente können innerhalb der Definition mit `#Nummer` eingesetzt werden.

$\iint_D g(x, y) dx dy$	<pre>\newcommand{\du}{\, \mathrm{d}} \begin{displaymath} \int\!\!\!\int_D g(x, y) \du x \du y \end{displaymath}</pre>
$\iint_D g(x, y) dx dy$	<pre>\newcommand{\dv}[1]{\, \mathrm{d}\#1} \begin{displaymath} \int\!\!\!\int_D g(x, y) \dv{x} \dv{y} \end{displaymath}</pre>

Die Wellenfunktion  $\langle \Psi |$  ist symmetrisch!

```
\newcommand{\bra}[1]{\ensuremath%
{\langle#1|}}
Die Wellenfunktion \bra{\Psi} ist
symmetrisch!
```

`\ensuremath{Argument}` setzt *Argument* im Mathematik-Modus, egal ob dieser vorher aktiv war oder nicht.

Ein weiterer wichtiger Grund für neue Kommandos ist die Möglichkeit, schnell Konventionsänderungen durchzuführen.

### 3.2 Neue Umgebungen definieren

Mit dem Befehl `\newenvironment{Name}[Zahl]{vorher}{nachher}` kann man neue Umgebungen definieren. Der Befehl funktioniert ähnlich wie der Befehl `\newcommand`. Im ersten Argument (*vorher*) wird angegeben was bei `\begin{name}` ausgeführt wird, im zweiten (*nachher*) was bei `\end{name}` ausgeführt wird.

● Diese Umgebung bewirkt...

```
\newenvironment{itemhu}
  {\begin{itemize}\Huge}
  {\end{itemize}}
\begin{itemhu}
  \item Diese Umgebung bewirkt...
\end{itemhu}
```

### 3.3 Eigene Pakete erstellen

Wenn man einzelne selbstdefinierte Befehle und Umgebungen in den meisten seiner Texte verwendet, ist es sehr praktisch, sie in einem Paket zusammenfassen. Zu Beginn eines Dokuments werden diese Befehle dann mit `\usepackage{meinPaket.sty}` in das Dokument eingebunden. Die Grundstruktur eines Paketes `bracket.sty` sähe folgendermaßen aus:

```
\ProvidesPackage{bracket}
\newcommand{\bra}[1]{\langle\bra{#1}\rangle}
\newcommand{\ket}...
```

### 3.4 Zähler

Zähler stecken hinter allem, was in L<sup>A</sup>T<sub>E</sub>X automatisch durchgezählt wird. Folgende Befehle gibt es um Zähler zu beeinflussen.

<code>\setcounter{Zähler}{Zahl}</code>	setzt einen Zähler auf einen bestimmten Wert
<code>\addtocounter{Zähler}{Zahl}</code>	addiert etwas zu einem Zähler hinzu
<code>\newcounter{Zähler}</code>	erstellt einen neuen Zähler
<code>\stepcounter{Zähler}</code>	addiert 1 zum Zähler hinzu

Ein paar Standard-Zähler:

<code>page</code>	Seitenzahl
<code>equation</code>	Gleichung
<code>section</code>	Abschnitt
<code>subsection</code>	Unterabschnitt
<code>subsubsection</code>	Unterunterabschnitt

$a^2 + b^2 = c^2$	(1)	<pre>\begin{equation}   a^2 + b^2 = c^2 \end{equation}</pre>
$a^2 + b^2 = c^2$	(12)	<pre>\addtocounter{equation}{10} \begin{equation}   a^2 + b^2 = c^2 \end{equation}</pre>

### 3.5 Längen

Viele Kommandos benötigen Längen d.h. Abstände als Parameter. Folgende Längeneinheiten können in L<sup>A</sup>T<sub>E</sub>X verwendet werden:

mm	Millimeter $\approx 1/25$ Inch	□	
cm	Zentimeter	┌───┐	
in	Inch = 25.4 mm	┌──────────┐	
pt	Punkt $\approx 1/72$ Inch $\approx \frac{1}{3}$ mm	¶	
em	Breite eines „M“	┌──┐	
ex	Höhe eines „x“	┌┐	

Mit dem Befehl `\newlength{Name}` kann man eine neue Länge definieren. Mit den Befehlen `\setlength{Name}{Länge}` und `\addtolength{Name}{Länge}` kann man den Längen Werte zuweisen. Ein *Name* einer Länge könnte z.B. `\Beispiel` lauten.

Mit diesen Befehlen ist es auch Möglich interne Längen von L<sup>A</sup>T<sub>E</sub>X zu verändern.

<p>Dies ist ein Absatz in dem die erste Zeile eingerückt ist.</p> <p>Dies ist ein Absatz in dem die erste Zeile nicht eingerückt ist.</p>	<pre>\setlength{\parindent}{3em} Dies ist ein Absatz in dem die erste Zeile einger\ "uckt ist.</pre> <pre>\setlength{\parindent}{0cm} Dies ist ein Absatz in dem die erste Zeile nicht einger\ "uckt ist.</pre>
---	---

Mit den Befehlen `\indent` und `\noindent` kann man ebenfalls das Einrücken der ersten Zeile eines Absatzes ein- bzw. ausschalten.

Es ist auch möglich L<sup>A</sup>T<sub>E</sub>X einen Bereich anzugeben, in dem eine Länge liegen soll.

<p>Absatz 1</p> <p>Absatz 2</p>	<pre>Absatz 1 \setlength{\parskip} {1ex plus 0.5ex minus 0.2ex}  Absatz 2</pre>
---------------------------------	---

Je nachdem wie die Absätze auf die Seite passen kann L<sup>A</sup>T<sub>E</sub>X den Abstand zwischen den einzelnen Absätzen zwischen 0.8ex und 1.5ex variieren.

Man kann Längen auch mit Zahlen multiplizieren. So setzt der Befehl `\hspace{0.5\textwidth}` also einen horizontalen Leerraum von einer halben Zeilenlänge.

Am besten ist es natürlich, wenn man absolute Längenangaben vermeidet. Mit den Befehlen `\settoheight{Name}{Text}`, `\settodepth{Name}{Text}` und `\settowidth{Name}{Text}` kann die Maße eines *Textes* in ein Längenkommando zu speichern.

Dazu ein Beispiel aus [1]:

$a^2 + b^2 = c^2$ <p>Where: <math>a</math>, <math>b</math> – are adjunct to the right angle of a right-angled triangle.</p> <p><math>c</math> – is the hypotenuse of the triangle and feels lonely.</p> <p><math>d</math> – finally does not show up here at all. Isn't that puzzling?</p>	<pre> \flushleft \newenvironment{vardesc}[1]   {#1:\ \settowidth{\parindent}   {#1:\ }}{}  \begin{displaymath} a^2+b^2=c^2 \end{displaymath}  \begin{vardesc}{Where}\$a\$, \$b\$ -- are adjunct to the right angle of a right-angled triangle.  \$c\$ -- is the hypotenuse of the triangle and feels lonely.  \$d\$ -- finally does not show up here at all. Isn't that puzzling? \end{vardesc} </pre>
--	--

### 3.6 Horizontaler/Vertikaler Leerraum

Mit den Befehlen `\hspace{Länge}`, `\hspace*{Länge}`, `\vspace{Länge}` und `\vspace*{Länge}` kann man einen horizontalen bzw. vertikalen Leerraum erzeugen. Die \*-Formen erzeugen den Leerraum auch wenn der Befehl an einem Zeilen- bzw. Seitenumbruch auftritt.

<p>Jetzt kommt ein Abstand 2 cm. Mit Stern geht's über Zeilenende hinweg.</p>	<p>von Jetzt kommt ein Abstand <code>\hspace{2cm}</code> von 2~cm. Mit Stern geht's <code>\hspace*{2cm}</code> "über ein Zeilenende hinweg.</p>
---	---

### 3.7 Stretch

Mit dem Befehl `\stretch{n}` erzeugt man einen Abstand der Länge 0 und einer Dehnbarkeit von  $\infty$ . Es wird also solange gedehnt, bis die Zeile voll ist. Wenn mehrere `\stretch`-Befehle in einer Zeile verwendet werden, geben die  $n$ 's das

Verhältnis der Streckung an.

x

x

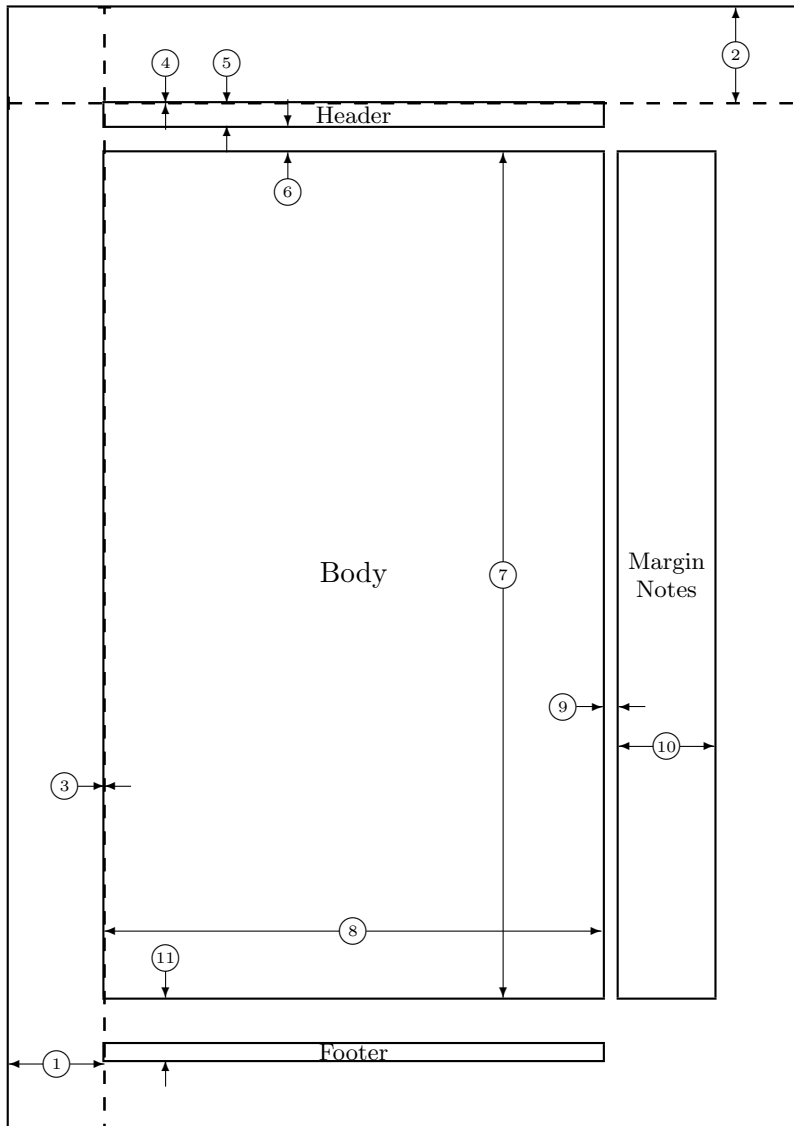
x `x\hspace{\stretch{1}}`  
x `x\hspace{\stretch{3}}`x

### 3.8 Seitenlayout

In der `\documentclass` wird angegeben, welche Papiergröße verwendet wird.  $\text{\LaTeX}$  stellt danach automatisch die Textränder ein. Mit den `\setlength`-Befehlen kann man die verschiedenen Längen verändern.

**Vorsicht:** es hat seine Gründe, dass eine normale  $\text{\LaTeX}$ -Seite so aussieht, wie sie aussieht!

Auf der nächsten Seite wird einmal dargestellt, welche Größen es auf einer  $\text{\LaTeX}$ -Seite gibt. Die Längenangaben beziehen sich hier auf die Dokumentenklasse `scrartcl`!





```
1 one inch + \hoffset      2 one inch + \voffset
3 \oddsidemargin = 0pt    4 \topmargin = 0pt
5 \headheight = 17pt      6 \headsep = 20pt
7 \textheight = 636pt     8 \textwidth = 375pt
9 \marginparsep = 12pt    10 \marginparwidth = 72pt
11 \footskip = 47pt        \marginparpush = 6pt (not shown)
    \hoffset = 0pt         \voffset = 0pt
    \paperwidth = 597pt    \paperheight = 845pt
```

### 3.9 Aufgaben

#### Aufgabe 1

Definiere Kommandos für:

<code>\bra</code>	$\langle \Psi  $	ein Argument; zwischen <code>\langle</code> und <code>\mid</code>
<code>\ket</code>	$  \Psi \rangle$	ein Argument; zwischen <code>\mid</code> und <code>\rangle</code>
<code>\bk</code>	$\langle \Psi   \Phi \rangle$	zwei Argumente: $\Psi$ und $\Phi$
<code>\mbra</code>	$\langle \Psi_{\alpha_1} \cdots \Psi_{\alpha_N}  $	vier Argumente: $\Psi$ , $\alpha$ , 1 und $N$
<code>\mket</code>	$  \Psi_{\beta_1} \cdots \Psi_{\beta_N} \rangle$	vier Argumente: $\Psi$ , $\beta$ , 1 und $N$
<code>\mbk</code>	$\langle \Psi_{\alpha_1} \cdots \Psi_{\alpha_N}   \Phi_{\beta_2} \cdots \Phi_{\beta_{N+1}} \rangle$	acht Argumente

Damit du diese neuen Befehle sowohl im mathematischen Modus, wie auch im Textmodus verwenden kannst, ohne jedes mal explizit den mathematischen Modus aktivieren zu müssen, schreibe die Befehlsfolge für das neue Kommando in ein `\ensuremath{...}`-Kommando.

#### Aufgabe 2

Definiere eine Umgebung mit der man folgendes abkürzen kann:

- Dies ist ein Test ...
- `\rule{1ex}{1ex}\hspace{\stretch{1}}`  
Dies ist ein Test `\ldots`  
`\hspace{\stretch{1}}\rule{1ex}{1ex}`

#### Aufgabe 3

Erstelle ein neues Dokument `layout.tex`. Schreibe in den Kopf des Dokuments den Befehl `\usepackage{layout}`. Schreibe nach `\begin{document}` den Befehl `\layout` und schau dir das Ergebnis an. Verändere nun verschiedene Layout-Parameter **im Dokumentkopf** und **vor dem usepackage-Befehl** und überprüfe den Erfolg.

## 4 Andere Dokumentklassen

In L<sup>A</sup>T<sub>E</sub>X steckt das Wissen eines Buchdruckers, heißt es – genau genommen steckt allerdings in T<sub>E</sub>X das Wissen eines Setzers und in L<sup>A</sup>T<sub>E</sub>X das Wissen eines Layouters. Das Layout steckt hierbei in der gewählten Dokumentklasse. So erzeugen die Standardklassen `article`, `report`, `book` jeweils ein solches Layout im Amerikanischen Textsatz – Überschriften sind wie auch Strukturierungen in Serifenschriftarten gesetzt. Im europäischen Textsatz werden hingegen für alle Überschriften serifenlose Schriftarten verwendet. Hierzu kann das KOMA-Script-Paket verwendet werden, dieses stellt die Standarddokumentklassen in europäischem Textsatz zur Verfügung<sup>3</sup>. Man erhält die Namen der KOMA-Script-Klassen, indem man dem Namen ein `scr` voranstellt, hat die Klasse dann mehr als acht Buchstaben, lässt man die Vokale weg – `scrartcl`, `scrprft`, `scrbook`.

Da auch alle wissenschaftlichen Fachverlage heute ihre Texte in L<sup>A</sup>T<sub>E</sub>X setzen, wird auch das Design z.B. eines Springer-Buches schlicht und ergreifend durch die entsprechende Dokumentklasse erzeugt. Man findet mitunter auch diese Klassen manchmal im Internet – falls das AP-Protokoll mal wie der Gertshen aussehen soll, braucht man nur diese Klasse verwenden.

### 4.1 Die $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ -Klassen

Soll das AP-Protokoll nicht wie ein Buch, sondern gleich wie eine wissenschaftliche Veröffentlichung aussehen, sind die Klassen der American Mathematical Society die erste Wahl. In dieser Klasse werden die wichtigsten physikalischen Publikationen gesetzt – zum Beispiel die Physical Review Letters und die Applied Physical Letters. Diesen Zweck merkt man der Klasse auch an – sie ist nicht dazu gedacht, durch extravagantes Design Aufmerksamkeit zu erregen, sondern so leicht lesbar und platzsparend wie möglich zu sein. Überschriften sind nur minimal größer als normaler Text und auch nicht so weit abgesetzt, so daß mehr Text auf die Seite passt. Außerdem sind alle Erweiterungen, die die `amsmath`, `amsthm` und `amssymb`-Pakete mitbringen, auch hier enthalten.

Außer der geänderten Formatierung bringen die AMS-Klassen auch noch ein paar nützliche Befehle mit. Viele beziehen sich dabei auf Details des Satzes von mathematischen Formeln und sind hier eher nicht so wichtig, deswegen wollen wir nur eine Besonderheit hier vorstellen.

#### 4.1.1 Der `\newtheorem`-Befehl

Speziell beim Setzen von Skripten, aber auch bei den Ausarbeitungen von Seminaren braucht man Umgebungen, die referenzierbar sind, gleichzeitig aber Aussage, Beweisführung und ähnliches zusammenfassen. Hierzu kann man mit `\newtheorem` Umgebungen definieren, zum Beispiel für Lemmata, Sätze, Korollare – aber auch z.B. in einem IK-Skript die Experimente so strukturieren.

<sup>3</sup>Auch wenn Überschriften serifenlos besser aussehen, sollte vermieden werden, ganze Dokumente in serifenloser Schrift zu setzen, da dies in Mengen deutlich schlechte zu lesen ist

Die Syntax ist dabei `\newtheorem{name}[Zähler(optional)]{Bezeichnung}`. NAME ist dabei der Name der neu erzeugten Umgebung, beim Aufruf dieser Umgebung wird als Titel des Theorems dann *Bezeichnung* gedruckt.

Normalerweise hat jedes Theorem seinen eigenen Zähler, definiere ich mir also ein Theorem *Lemma* und ein Theorem *Korollar*, so steht dort nach n- bzw. m-maligen Aufrufen *Lemma n* bzw. *Korollar m*. Mit dem optionalen *Zähler*-Argument benutzt ein theorem dann den Zähler eines anderen mit.

**Lemma 1.**

$$a = b \wedge b = c \Rightarrow a = c$$

**Satz 2.**

$$a \Rightarrow b \Leftrightarrow \bar{b} \Rightarrow \bar{a}$$

*Korollar 1.*

$$a > b \wedge b < c \neq A > b$$

```

\theoremstyle{plain}
\newtheorem{lem}{Lemma}
\theoremstyle{definition}
\newtheorem{satz}[lem]{Satz}
\theoremstyle{remark}
\newtheorem{corr}{Korollar}
\begin{lem}\[a=b \land b=c \rightarrow%
a=c]\end{lem}
\begin{satz}\[a\rightarrow b%
\leftarrow \overline{b}%
\rightarrow \overline{a}]\end{satz}
\begin{corr}\[ a>b \land b<c %
\neq A>b]\end{corr}

```

## 4.2 Briefe mit `scr1ttr2`

Auch wenn es dem einen oder anderen vielleicht etwas übertrieben erscheinen mag, aber es gibt gute Gründe (und gute Hilfsmittel), auch Briefe in L<sup>A</sup>T<sub>E</sub>X zu setzen. Unter anderem

- Blocksatz!
- Passend vorgegebene Faltung
- Kompatibel zu Fensterumschlägen
- optisch stimmiges Layout
- sehr leicht für Serienbriefe nutzbar

Hier zunächst einmal ein kleines Beispiel:



OTTO WAALKES  
AM DEICH 5  
3141 NORDDEICH MOLE  
E-MAIL: GABESDAMALS@NOCHNICHT.DE

Otto Waalkes, Am Deich 5 . 3141 Norddeich Mole

Deutsche Bank AG  
Frau von Ackermann  
Goetheplatz 5  
66734 Frankfurt

Kontostand	Bankverbindung	Datum
42€	Deutsche Bank Frijsland, 21070024, Konto 16021019	18. Dezember 2006

Sehr geehrte Frau von Ackermann

mit Freude nehme ich Ihre regelmäßigen Mitteilungen, mein Konto sei überzogen, zur Kenntnis. Auch ist es mir stets ein Vergnügen, Ihren Aufforderungen, dies zu ändern, aufs eiligste nachzukommen.

Da momentan der außerordentliche Zustand, daß mein Konto im Plus ist, vorherrscht, möchte ich es nicht versäumen, diese Höflichkeit zu erwidern und Ihnen dies mitzuteilen. Auch ich wäre Ihnen verbunden, wenn Sie dies binnen der nächsten acht Tage ausgleichen könnten.

Im Übrigen möchte ich darauf hinweisen, daß Ihre Briefe stets nur im Flattersatz gesetzt sind. Finden Sie nicht auch, daß es im Blocksatz deutlich besser aussieht?

Mit freundlichem Grusse

O. Waalkes

Beispiel für einen `scrlettr2`-Brief

```

\documentclass[DIN, pagenumber=false, parskip=half, fromalign=right,
fromphone=false, fromemail=true, fromurl=false, fromlogo=true,
  fromrule=false]{scrlettr2}
\usepackage[applemac]{inputenc}
\usepackage{ngerman}
\RequirePackage{graphicx}
\usepackage{eurosym}

\setkomavar{fromname}{Otto Waalkes}
\setkomavar{fromaddress}{Am Deich 5 \\ 3141 Norddeich Mole}
\setkomavar{fromemail}{gabesdamals@nochnicht.de}
\setkomavar{signature}{0. Waalkes}
\setkomavar{frombank}{Deutsche Bank Frijsland, 21070024, Konto 16021019}
\setkomavar{fromlogo}{\includegraphics[scale=10]{phi.pdf}}
\newkomavar*[Kontostand]{mybalance}
\setkomavar{mybalance}{42\euro}
\addtoeffields{frombank}
\setkomafont{fromaddress}{\scshape}
\setkomafont{fromname}{\scshape}

\begin{document}
\begin{letter}{Deutsche Bank AG\\Frau von Ackermann
\\ Goetheplatz 5 \\ 66734 Frankfurt}
\opening{Sehr geehrte Frau von Ackermann}

mit Freude nehme ich Ihre regelmäßigen Mitteilungen, mein Konto sei
überzogen, <snip>
.....
<snap>Sie nicht auch, daß es im Blocksatz deutlich besser aussieht?

\closing{Mit freundlichem GruÙe}
\end{letter}

\end{document}

```

Wichtig sind im Quelltext vor allem zwei neue Konzepte: Optionen und Variablen. KOMA kennt eine ganze Reihe solcher Optionen und Variablen. Eine Variable ist dabei ein – mögliches – Element des Dokuments, dem bei jedem Dokument ein neuer Inhalt gegeben werden kann. Eine Option hingegen bestimmt, was z.B. mit einer Variablen gemacht werden soll.

Optionen können entweder als Option der Dokumentklasse angegeben werden oder anschließend mit `\KOMAOption{option=wert}\verb` gesetzt werden.

**parskip=half** Während in normalen Dokumenten ein neuer Absatz stets durch eine Einrückung gekennzeichnet wird, ist es bei Briefen eher noch üblich, Leerzeilen einzufügen. Die gewählte Option fügt eine halbe Leerzeile ein, andere Optionen sind **full** für eine ganze Zeile und **false** für eine Absatzeinrückung

**pagenumber=false** Schaltet die Seitennummerierung aus

**from\*** Diese Optionen beziehen sich jeweils auf eine Absendervariable und teilen mit , ob die entsprechende Variable im Text erscheinen soll oder nicht. Leider funktioniert dies nicht mit allen Absendervariablen. So wird die Absenderbankverbindung **frombank** auch bei **frombank=true** weggelassen.

Die wichtigsten Variablen sollten leicht aus dem Quelltext ersichtlich sein, es gibt derer sehr viele. Eine genauere Auflistung findet sich in [6]. Allerdings kommt man doch oft in die Situation, ein Feld für einen Extra-Zweck in der Referenzzeile zu benötigen. Man kann sich solche Felder mit `\newkomavar[Bezeichner]{variablenname}` definieren. In eckigen Klammern steht dabei der Bezeichner, der im Brief über den Inhalt geschrieben wird, in geschweiften Klammern steht der Variablenname, unter dem die Variable im Quelltext aufgerufen werden kann. Um der Variable einen Wert zuzuweisen, muß dieser auf jeden Fall erst mit `\setkomavar{variablenname}{Inhalt}` gesetzt werden.

Möchte man nun seine neue Variable auch in der Referenz-Zeile erscheinen lassen, muß man sie mit `\addtoeffields{variablenname}` hinzufügen. Dies ist auch für bereits vordefinierte Variablen möglich, für selbstdefinierte kann man dies auch mit `\newkomavar*` abkürzen.

Eine Reihe von Variablen fristen ihr Dasein quasi im Geheimen – man findet von Ihnen nichts im gedruckten Dokument. Dies dient dazu, diese dann an anderen Stellen aufrufen zu können (Dies ist auch in allen anderen **scr**-Klassen möglich und dort auch evtl. sinnvoller). Der Aufruf geschieht mit `\usekomavar{variablenname}`.



## 5 Das beamer-Paket

Das Paket `beamer` wurde von Till Tantau entwickelt und im März 2003 veröffentlicht. Es ist seitdem sehr populär geworden, und man findet von vielerlei Autoren zusätzliche Stildateien, um das Aussehen der Präsentationen zu verändern.

`beamer` erzeugt Präsentationen, die für die Darstellung mit einem Videoprojektor optimiert sind<sup>4</sup>. Das heißt, die Folien werden im 4:3-Querformat erzeugt und haben eine sehr geringe (Druck-)Auflösung von 72dpi. Ausgabeformat ist auch hier PDF, das dann (mit dem Acrobat Reader) aufgerufen und im Vollbildmodus präsentiert werden kann.

### 5.1 Der Kopf eines beamer-Dokuments

`beamer` ist eine Dokumentklasse wie `scratcl` oder `book`. Im Kopf einer solchen Präsentation werden vor allem die grundlegenden Layout-Richtlinien festgelegt.

Grundsätzlich unterstützt der Kopf einer Präsentation die gleichen Kommandos wie andere L<sup>A</sup>T<sub>E</sub>X-Dokumente – `author`, `title`, `subtitle`, `date` können gesetzt werden. Hauptunterschied ist der `\mode`-Befehl, in dem die Layouteinstellungen der Präsentation festgelegt werden. Als Beispiel:

```
\mode<presentation>{
% es existieren auch noch andere Modi, die z.B. für die
%Ausgabe der Folien als Handout angepasst sind
  \usetheme[hideallsubsections]{Warsaw}
%das generelle Erscheinungsbild der Folien
  \usecolortheme{Albatros}
%Farbmodifikation des Erscheinungsbildes
  \usefonttheme{structuresmallcapsserif}
  %Gliederungen werden in Kapitälchen gesetzt
}
```

`\usetheme{}` Mit diesem Befehl wird das generelle Layout festgelegt – zum Beispiel, wie Kopf- und Fußzeile aussehen, wie die Titel einzelner Folien formatiert werden und auch wo, wie groß und ob überhaupt der momentane Gliederungspunkt angezeigt wird. Diese Themen sind alle nach Städten benannt, jeweils in deren englischem Namen (Warsaw, Antibes, Darmstadt, Berlin, Illmenau, Copenhagen...).

`\usecolortheme{}` Dieser Befehl legt die in der Präsentation verwendeten Farben für Schrift, Rahmen, Hervorhebungen usw. fest. Vorteil der Verwendung dieser Schemata ist, daß man um alte Powerpoint-Sünden, wie zum Beispiel hellgelben Text auf weißem Grund, ohne weiteres Nachdenken herkommt. Farbthemen sind nach fliegenden Tieren (seagull, albatros, crane...) benannt.

<sup>4</sup>Der korrekte englische Ausdruck lautet so auch *video projector*, *beamer* ist lediglich Slang für einen BMW

`\usefonttheme{}` Auch Schriftarten können bei `beamer` manipuliert werden. Die wichtigste Methode ist hierbei, die Strukturschriftart (Titel, Gliederungen) zu verändern. So erzeugt `\usefonttheme{structuresmallcapsserif}` Folien, bei denen Strukturbefehle in KAPITÄLCHEN, während die Inhalte einfach serifenlos gesetzt werden.

Weitere Formatvorlagen wie auch Informationen zu Detailmanipulationen findet man in [5]

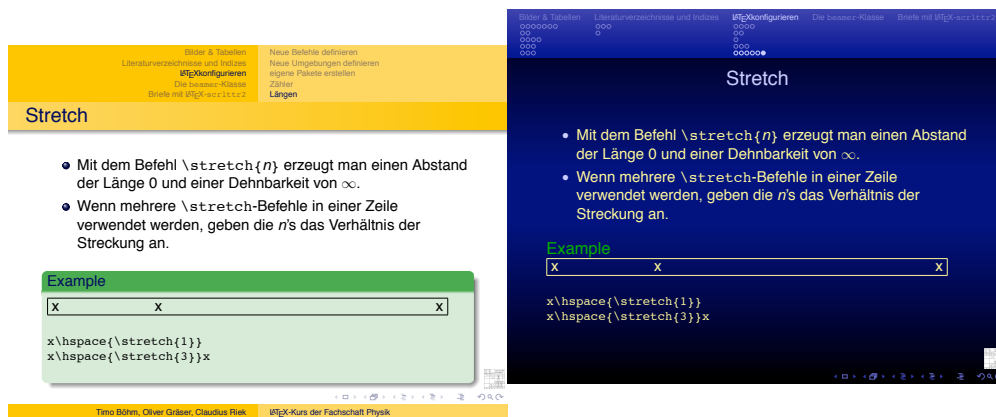


Abbildung 2: `\setheme{Warsaw}`,  
`\usecolortheme{crane}`

Abbildung 3: `\setheme{Singapore}`,  
`\usecolortheme[overlystylish]{albatross}`,  
`\usepackage{lily}`

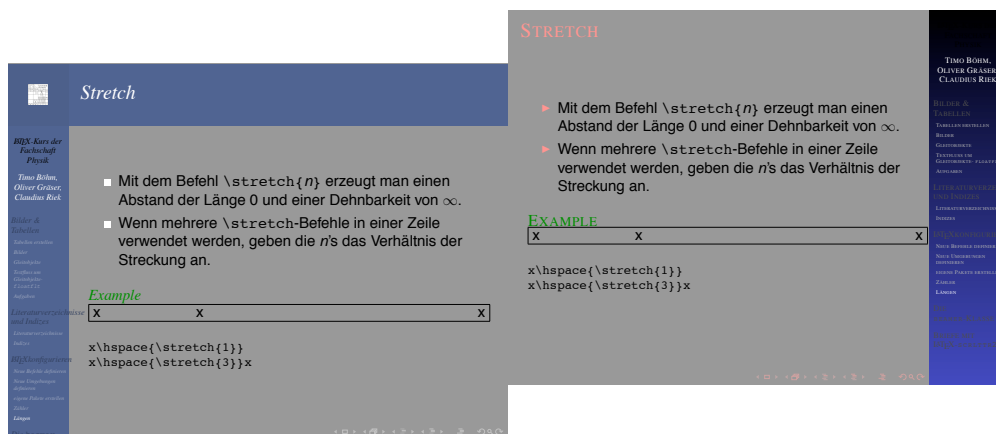


Abbildung 4: Strukturvorlage Berkeley, Farbschema beetle, Schriftschema structureritalicserif

Abbildung 5: Strukturvorlage Marburg, Farbschema fly, weiteres Farbschema [RGB={255,128,128}] {structure} Schriftschema structuresmallcapsserif

## 5.2 Eine einzelne Folie – die frame-Umgebung

Ein Frame ist eine Folie im Sinne von Powerpoint – ein Hintergrund mit festem Titel, auf der allerdings nacheinander verschiedene Dinge erscheinen können. Im Ergebnis sind dies natürlich dann mehrere Seiten in der PDF-Datei, da PDF das Hinzufügen solcher Argumente nicht kennt.

Jeder Frame wird mit `\begin{frame}` geöffnet und mit `\end{frame}` abgeschlossen. Jeder Frame kann genau einen Titel haben, der in diesem Fenster angezeigt wird, allerdings nicht in die Gliederung mit aufgenommen wird. Folgende Befehle sind innerhalb des Frames wichtig:

`\frametitle{.}` Mit diesem Befehl wird dem Frame ein Titel gegeben – dieser steht dann quasi als Überschrift auf jeder Ausgabeseite dieses Frames. Er ist allerdings *kein* Gliederungsbestandteil, man findet ihn also nicht im Inhaltsverzeichnis.

`\pause` Jeglicher Text und alle Befehle, die innerhalb dieses Frames, aber nach diesem Befehl stehen, werden erst auf der nächsten Seite ausgegeben – eine sogenannte Überlagerung. In Abhängigkeit von der Formatvorlage werden sie allerdings eventuell blass im Hintergrund dargestellt

`\item<a-b>` in `itemize`-Umgebungen kann jedem Spiegelpunkt mit diesem Befehl mitgeteilt werden, auf der wievielten Seite dieser Folie er erscheinen soll – hier erscheint er auf Seite a und verschwindet wieder ab Seite b. Hierdurch können auch mehrere Punkte gleichzeitig erscheinen – oder die untersten Punkte zuerst.

`\uncover{<cdot><n->` Dieser Befehl wirkt grundsätzlich wie die Positionierung bei den `\item`-Befehlen, allerdings wird hier das Argument von `uncover` ab der n-ten Folie gesetzt. `\uncover{\footnote{Schön, oder}}` `<4->` erzeugt also ab der vierten Ausgabeseite dieses Frames eine Fußnote.

Für Präsentationen ist es hilfreich, ab und an die Gliederung wieder anzuzeigen. Für solch wiederkehrende Seiten bietet `beamer` das Kommando `\AtBeginSection[] {Inhalt}`. Als Inhalt bietet sich zum Beispiel ein einzelner Frame mit dem Titel *Gliederung* an, der dann ein Inhaltsverzeichnis des aktuellen Abschnitts enthält. Ein solches setzt man mit `\tableofcontents [currentsection, currentsubsection]`

Im `beamer`-Paket sind sowohl Beispieldateien als auch ein schönes Tutorial dabei. Allerdings ist das Paket noch sehr jung, sodaß man manchmal noch mit kleinen Problemen leben muss. So klappt das Zusammenspiel von `beamer` mit `verbatim`-Umgebungen (Umgebungen, die den Inhalt wörtlich setzen) nur, wenn man als Option des Frames `[fragil]` angibt – das  $\beta$  kennt `beamer` dann aber auch mit `inputenc` nicht mehr.

## 5.3 Ein Beispiel für eine beamer-Präsentation

... haben wir in dem Dokument *Eine Handvoll Beamer-Features* zusammengestellt. Ihr könnt dieses unter

<http://fachschaft.physik.uni-konstanz.de/fachschaft/aktionen/latex/kurs2.html> zusammen mit dem Quelltext und allen anderen Dateien herunterladen.

## 6 pdfL<sup>A</sup>T<sub>E</sub>X

pdfL<sup>A</sup>T<sub>E</sub>X bietet außer dem neuen Dateiformat *.pdf* auch Zugriff auf die erweiterten Funktionen dieses Formates, die es im ursprünglich von T<sub>E</sub>X verwendeten DVI-Format nicht gab:

### 6.1 hyperref

Das `hyperref`-Paket stellt diese erweiterten Funktionen zur Verfügung. Mit `hyperref` werden das Inhaltsverzeichnis, Verweise auf Abbildungen, Tabellen, Formeln und Seiten verlinkt. Hier eine Aufstellung der Optionen des `hyperref`-Paketes:

<code>pdftex</code>	benötigte Option, damit <code>hyperref</code> mit pdf-L <sup>A</sup> T <sub>E</sub> X zusammenarbeitet
<code>pdftitle={}</code>	Text, der als Titel im Informationsfenster des Acrobat-Readers erscheint
<code>pdfauthor={}</code>	Text, der als Autor im Informationsfenster erscheint
<code>pdfsubject={}</code>	Text, der als Thema im Informationsfenster erscheint
<code>pdfproducer={}</code>	Text für den PDF-Hersteller
<code>pdfkeywords={}</code>	Text, der als Stichwörter im Informationsfenster erscheint
<code>bookmarks=true/false</code>	legt fest, ob Lesezeichen für den Acrobat Reader erstellt werden; die Lesezeichen sind genau wie das Inhaltsverzeichnis aufgebaut
<code>colorlinks=true/false</code>	legt fest, ob die Verknüpfungen im Dokument farbig dargestellt werden oder nicht
<code>citecolor=green</code>	Farbe der Links auf das Literaturverzeichnis
<code>filecolor=magenta</code>	Farbe der Links auf Dateien
<code>linkcolor=red</code>	Farbe der dokumentinternen Links
<code>urlcolor=cyan</code>	Farbe der Links auf Webseiten; wir bevorzuge die Farbe „blue“; auch möglich: „black“
<code>pdfpagemode=</code>	legt fest wie das Dokument im Acrobat Reader geöffnet wird: <code>None</code> – das Dokument wird ohne alles geöffnet; <code>UseThumbs</code> – wird mit der Seitenspalte auf der linken Seite geöffnet; <code>UseOutlines</code> – mit Lesezeichen geöffnet; <code>FullScreen</code> – wird im Vollbildmodus geöffnet
<code>pdfstartpage</code>	legt die Seite feste, die beim Öffnen des Dokuments angezeigt wird

Das `hyperref`-Paket sollte immer als letztes geladen werden. Es erweitert die Definition vieler L<sup>A</sup>T<sub>E</sub>X-Befehle und es könnte sonst passieren, dass die `hyperref`-Optionen wieder überschrieben werden. Die Optionen für das `hyperref`-Paket müssen nicht zwingend angegeben werden. Bei den meisten

Optionen sind sinnvolle Voreinstellungen vorhanden.

Mit dem `hyperref`-Paket ist es auch möglich Links zu Seiten im Internet zu erstellen:

[Fachschaft Physik](http://www.uni-konstanz.de/studis/fs/fs-physik/)

```
\href{http://www.uni-konstanz.de/%  
studis/fs/fs-physik/}{Fachschaft Physik}
```

## 6.2 Beispiel

Hier noch ein Vollständiges Beispiel für ein pdfL<sup>A</sup>T<sub>E</sub>X-Dokument:

```
\documentclass[12pt,a4paper]{scrartcl}  
  
\usepackage[latin1]{inputenc}  
\usepackage{ngerman}  
\usepackage[T1]{fontenc}  
\usepackage{mathptmx,helvet,courier|}  
  
\usepackage[pdftitle={Testdokument},%  
    pdfsubject={nur ein simpler Text},%  
    pdfauthor={Fachschaft Physik},%  
    pdfproducer={pdfLaTeX},%  
    pdfkeywords={Test, Test, Test},%  
    bookmarks=true,%  
    colorlinks=true,%  
%    colorlinks=false,%  
    urlcolor=blue,%  
    pdfpagemode=UseOutlines,%  
    pdfstartview=Fit,%  
    pdftex]{hyperref}  
  
\begin{document}  
  
\end{document}
```

## 7 Installation zusätzlicher Pakete in L<sup>A</sup>T<sub>E</sub>X

Wie im ersten Teil bereits gesagt, ist L<sup>A</sup>T<sub>E</sub>X Open Source und für den Anwender in verschiedenen Distributionen verfügbar. Allerdings steckt hinter diesen Distributionen etwas mehr als einfach nur die entsprechende Version für Windows, Linux oder OS X. Eine Distribution bedeutet darüber hinaus auch eine bestimmte Zusammenstellung von L<sup>A</sup>T<sub>E</sub>X-Paketen – manche Pakete sind bei manchen Distributionen dabei, bei anderen nicht. Da sich mittlerweile im wesentlichen zwei Distributionen herauskristallisiert haben (MikTeX für Windows und TeTeX für die Unix-Welt, einschließlich OS X), ist dies im wesentlichen

eine Frage der Version. Fehlt also das Paket in einer Version und ist in einer anderen Verfügbar, hilft sehr einfach ein Update. Allerdings sind sehr neue Pakete – **beamer** ist ein solches – oftmals nur als solche und nicht innerhalb einer Distribution verfügbar. Sie müssen daher manuell nachinstalliert werden, und gerade an diesem schon nicht ganz trivialen Punkt unterscheiden sich die Distributionen teilweise recht deutlich.

## 7.1 Der einfache Weg....

Auch ohne größere Kenntnisse lassen sich viele Pakete mit einer neuen MikTeX-Installation bzw. einer debian-basierten Linuxinstallation installieren

### 7.1.1 apt-get unter debian

*apt-get* ist der Paketmanager von debian, er wird auch in vielen anderen Distributionen (Ubuntu, Yellow Dog) verwendet. Liegt ein L<sup>A</sup>T<sub>E</sub>X-Paket als .deb vor (das ist für erstaunlich viele der Fall), kann man es über diesen Paketmanager installieren, so man denn eine Verbindung zum Internet hat. Zunächst muss man den exakten Namen des Pakets herausfinden. Mittels

```
apt-get update
```

```
apt-cache search string
```

kann man sich alle Pakete anzeigen lassen, die den entsprechenden String im Namen führen. Mittels

```
apt-get install Paketname
```

wird das Paket dann mit allen anderen Paketen, die zur Ausführung benötigt werden, installiert. Unter Ubuntu hilft auch das GUI-Utility *Synaptic*.

### 7.1.2 Der MikTeX Update Wizard

## 7.2 Installationsbäume

Eine normale L<sup>A</sup>T<sub>E</sub>X-Distribution kennt drei Installationsbäume: Den benutzereigenen Installationsbaum, den lokalen Installationsbaum und den globalen Installationsbaum. Dies liegt ein wenig an der Unix-Herkunft der ersten Installationen und bezieht sich auf Komponenten, die dem ganzen Netzwerk, bestimmten Rechnern beziehungsweise einem bestimmten Nutzer zur Verfügung stehen. So gibt es zum Beispiel in der MiK<sub>T</sub>E<sub>X</sub>-Distribution keinen benutzer-spezifischen Zweig mehr.

Beim setzen eines Dokuments benötigt L<sup>A</sup>T<sub>E</sub>X zumindest die entsprechende Dokumentklasse (.cls), oft auch noch zusätzliche Pakete (.sty). Nach diesen sucht LaTeX im aktuellen Verzeichnis sowie in den Installationsbäumen. Wenn LaTeX die entsprechende Datei nicht findet, stoppt es den Textsatz und zeigt eine Fehlermeldung an.

### 7.2.1 MikTeX

MikTeX fragt die Pfade für den globalen und lokalen Installationspfad bei der Installation ab – die Vorschläge hierfür sind `C:\texmf` bzw. `C:\localtexmf`, man kann Sie allerdings ändern, zum Beispiel als Unterordner von `C:\Programme`

### 7.2.2 TeTeX

TeX die Bürde der vielen verschiedenen \*X-Distributionen zu tragen – wo man sich bei Linux schon nicht auf bestimmte Anordnungen einigen kann, kann man das dann für die TeX-Distribution sicher nicht. Generell ist es allerdings so, daß die Installationsbäume im allgemeinen in Ordnern namens *texmf* oder *texmf.local* liegen, diese kann man gegebenenfalls einfach suchen. Typische Installationsordner sind zum Beispiel

**Suse- und debian-Linux** /usr/share/texmf und /etc/share/texmf

**Fink-basierte Installation auf OSX** Fink legt für alle Pakete das neue Verzeichnis /sw auf der Bootpartition an. Man findet den Hauptzweig unter /sw/share/texmf, den lokalen Pfad unter /sw/etc/texmf.local. Außerdem weißt ein Link von /sw/share/texmf-local dorthin

**installer-basierte OS X Installation** Ein wenig ausgefallener, wie es sich für einen Mac gehört: /usr/local/TeX/share/texmf.tetex/tex/latex/ ;-)

**andere Linux-Installationen** Während /usr/share/texmf offenbar der globale Standardpfad ist, gibt es verschiedenen Quellen zufolge einen lokalen TeX-Pfad meist unter /usr/local/share/texmf (den wir aber bei den uns zur Verfügung stehenden Installationen nicht gefunden haben)

### 7.2.3 Benutzerspezifischer TeX-Pfad

Dies ist der Zweig, unter dem man ohne weitere Rechte L<sup>A</sup>T<sub>E</sub>X-Pakete installieren kann. Er existiert nur auf Unix-Systemen und lautet dort im allgemeinen /texmf/ oder /Library/texmf. Er ist meist nicht angelegt, daher ist es wohl ein wenig Glückssache, den richtigen Pfad zu finden. Im Zweifelsfall sollte man sich an die Administratoren des entsprechenden Rechnerpools wenden

## 7.3 Installation der Pakete

Die meisten Pakete kommen heute in einem Ordner gemeinsam mit Dokumentation, Beispielen, Makros für Emacs/Lyx und anderen Erweiterungen. Damit das Paket nutzbar wird, muß es lediglich an die passende Stelle im Installationsbaum kopiert werden. Hierzu bietet sich vor allem der lokale Installationsbaum an – der benutzerspezifische ist oft schwer aufzufinden und beim globalen besteht die Gefahr, daß die Änderungen bei einem Update der Installation einfach überschrieben werden. Welchen Zweig man dann auch immer wählt, innerhalb dieses Zweiges müssen die Pakete immer in einen Unterordner von  $\$texpath\$/tex/latex/$  kopiert werden. Da die meisten Pakete in einem Ordner kommen, einfach diesen Ordner dorthin kopieren (es werden lokale root-Rechte bzw. Administratorrechte benötigt). Auch für selbst erstellte Pakete sollte man der Übersichtlichkeit halber einen Unterordner erzeugen.

Sind die Pakete an der richtigen Stelle, muß das Programm *texhash* gestartet werden, das die Paketdatenbank auf den neuesten Stand bringt. Es befindet sich bei Linux-Systemen meist in /usr/bin, bei Fink-Installationen in /sw/bin

(als Link auf `/sw/bin/mktextlsr`), beim iinstaller unter `/usr/local/teTeX/bin/powerpc-apple-darwin-current/texhash` und auf Windows-Systemen unter `C:\Programme\MikTeX`. Anschließend sollte L<sup>A</sup>T<sub>E</sub>X die so installierten Pakete finden.

## 8 Dokumentation

Weitere Informationen zu den installierten L<sup>A</sup>T<sub>E</sub>X-Paketen gibt es im Verzeichnis `/usr/share/texmf/doc/`. Dort gibt es auch ein Unterverzeichnis `/usr/share/texmf/doc/help/Catalogue/` mit einer Datei `catalogue.html` die Informationen über die meisten erhältlichen L<sup>A</sup>T<sub>E</sub>X-Pakete enthält.

Außerdem erhält man viele Informationen über die Homepage der „Deutschsprachigen Anwendervereinigung T<sub>E</sub>X e. V.“ auf <http://www.dante.de/>.



## Literatur

- [1] Tobias Oetiker, Hubert Partl, Irene Hyna, Elisabeth Schlegl.  
*The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*. PDF-Datei, 2001  
(wird ständig aktualisiert)  
(<http://people.ee.ethz.ch/~oetiker/lshort/lshort.pdf>).
- [2] Helmut Kopka. *L<sup>A</sup>T<sub>E</sub>X— Eine Einführung*. Addison–Wesley, 2000.
- [3] Michael Goossens, Frank Mittelbach, Alexander Samarin.  
*Der L<sup>A</sup>T<sub>E</sub>X–Begleiter*. Addison–Wesley, 1996.
- [4] Leslie Lamport. *Das L<sup>A</sup>T<sub>E</sub>X–Handbuch*. Addison–Wesley, 1995.
- [5] Till Tantau  
*User’s Guide to the Beamer Class*, PDF  
(<http://latex-beamer.sourceforge.net/>)
- [6] Frank Neukam, Markus Kohn, Axel Kielhorn  
*SCRGUIDE–Anleitung zu KOMA–Script*