

Einführung in das wissenschaftliche Schreiben und Layouten mit \LaTeX

Markus Haslinger

HTL Leonding – \LaTeX Workshop

Inhaltsverzeichnis

1	Einleitung	1
1.1	WYSIWYG vs. L ^A T _E X	1
1.2	Der Compiler	2
2	Vorstellung der Editoren	3
2.1	VS Code + L ^A T _E X Workshop Extension	3
2.2	LyX	4
2.3	Microsoft Word & Co.	5
3	Arbeiten mit L^AT_EX	7
3.1	Der header	7
3.1.1	documentclass	8
3.1.2	packages	8
3.1.3	Dokumentenweite Einstellungen	8
3.1.4	Einsatz einer Vorlage	9
3.2	Strukturbefehle	9
3.2.1	Abschnitte	9
3.2.2	Aufzählungen	10
3.3	Schriftgröße & Effekte	10
3.4	Tabellen	11
3.5	Grafiken	12
3.6	Codeausschnitte	12
3.7	Referenzen	12
3.7.1	Verzeichnisse	13
3.7.2	Zitate (BibTeX)	13
3.7.3	Fußnoten	13
3.7.4	Querverweise	14
3.8	Formeln	14
3.9	Sonderzeichen und Konventionen	15
4	Mendeley Literaturverwaltung	16
5	Präsentationen mit L^AT_EX	17

Tabellenverzeichnis

0	eine Beispieltabelle	11
2	Tabelle mit einigen Sonderzeichen und Konventionen	15

Abbildungsverzeichnis

1	Einfaches Vergleichsbeispiel \LaTeX vs. WYSIWYG	2
2	Ein Beispiel eines LyX-Dokuments	4
3	\LaTeX Code in LyX	5
4	Ein Beispiel eines Worddokuments	6
0	Eine Beispielgrafik	12
6	Ausschnitt aus der Mendeley Web-Version	16
7	Schlechte Quellen nicht in den Catalog eintragen	16
8	Fortschrittsanzeige der Präsentation	18

1 Einleitung

Die elektronische Textverarbeitung hat die analoge in unseren Breiten mittlerweile größtenteils abgelöst. Insofern ist es wenig überraschend, dass sich die Meisten, ob im Beruf oder privat, damit auseinandersetzen. Häufig hört man in diesem Zusammenhang von Problemen bei der Formatierung von Texten. Dabei hat sich gezeigt, dass Textverarbeitung fast immer mit Microsoft Word gleichgesetzt wird, während L^AT_EX den wenigsten ein Begriff ist. Klarerweise trifft dies im akademischen Bereich nicht unbedingt zu. Dennoch ist auch hier, zumindest bei den Studenten die nicht an einer technischen Fakultät studieren, L^AT_EX als kompliziert und unnötig verschrien. Andererseits trauern diejenigen die die Anfangshürden genommen haben und nun auf L^AT_EX setzen, Word & Co. nicht nach. Um den Abbau von Vorurteilen zu unterstützen sollen im Folgenden sowohl die Vor- als auch die Nachteile, im Sinne von 'there is no tool to rule them all', der jeweiligen Systeme beschrieben werden.

In diesem Skript sollen nur die wichtigsten Befehle von L^AT_EX kompakt zusammengefasst werden, für einen genaueren – wenn auch immer noch oberflächlichen – Einblick sei auf die L^AT_EX 2 ϵ Kurzeinführung im Anhang verwiesen¹.

Zusätzlich wird ein Editor vorgestellt, der das Arbeiten mit L^AT_EX stark vereinfacht.

1.1 WYSIWYG vs. L^AT_EX

Als 'what you see is what you get – WYSIWYG' Editor bezeichnet man ein Programm, dass bereits während der Eingabe von Daten die jeweils aktuelle und endgültige Ausgabe anzeigt. Ein Beispiel hierfür ist Word. Das 'Eingabefeld' stellt – zumindest in den Standardeinstellungen – eine A4 Seite dar. Schreibt man auf diese 'Seite' Text oder fügt Bilder ein, sind diese nach dem Speichern und auch auf der gedruckten Seite an (ziemlich) genau der Stelle an der die Elemente zuvor am Bildschirm dargestellt wurden. Über das Format entscheidet der Benutzer. Das hat Vor- aber auch Nachteile:

- Es ist keine Vorstellungskraft nötig und der Benutzer kann entscheiden an welcher Stelle ein Element – z. B. ein Bild – positioniert werden soll.
- WYSIWYG Editoren sind üblicherweise sehr einsteigerfreundlich konzipiert.
- Leider sind die Fähigkeiten dieser Programme begrenzt, weshalb es schwierig ist unübliche Formatoptionen umzusetzen.
- Gerade bei großen Dokumenten kann bereits eine kleine Änderung wie das Verschieben einer Grafik um wenige Millimeter das Layout des ganzen Dokumentes zerstören, da sich alle anderen Elemente automatisch an diese Änderung 'anpassen'.

L^AT_EX geht einen anderen Weg. Dazu kurz zur Herkunft: Donald Knuth² hat sich geärgert, da die Layouter bei den Verlagen den Inhalt seiner Bücher nicht verstanden und diese – insbesondere die mathematischen Formeln und Funktionen – falsch setzten. Daher hat er sich ein gutes Jahrzehnt lang hingesetzt und die Programmiersprache T_EX entwickelt, die es ihm ermöglichte seine Bücher selbst zu setzen und den Verlagen nur noch das Drucken zu überlassen. Kennt man Don's Arbeit ist es nicht verwunderlich, dass außer ihm niemand klug genug war um mit T_EX zu arbeiten. Deshalb

¹Umfangreiche Hilfestellung auch bei Detailfragen und Problemen bietet <http://tex.stackexchange.com/>

²<http://www-cs-faculty.stanford.edu/~uno/>

wurde L^AT_EX entwickelt. Dieses abstrahiert die komplexen T_EX Anweisungsblöcke mit (relativ) einfachen Kommandos. L^AT_EX richtet sich also ganz klar an Spezialisten die ein fehlerfreies Layout langer Dokumente, welche Formeln, Tabellen und Grafiken enthalten, benötigen. Zumeist sind dies Wissenschaftler die Publikationen verfassen. Dementsprechend implementiert L^AT_EX auch die IEEE Zitierweise, doch dazu später.

1.2 Der Compiler

Ja! L^AT_EX ist eine Programmiersprache.

Der Autor verfasst L^AT_EX Code, bestehend aus zwei Teilen:

1. Den Inhalten des Autors: Text, Grafiken, Tabellen, Formeln
2. L^AT_EX Befehlen und Steuerzeichen

Dieser Code wird dem Compiler übergeben³. Doch da wir mit L^AT_EX und nicht mit T_EX arbeiten, müssen zuerst die L^AT_EX Befehle von einem Präprozessor in T_EX Code umgewandelt werden. Der Benutzer bemerkt von diesem Zwischenschritt üblicherweise nichts. Anschließend generiert der Compiler eine Ausgabedatei (etwa *.dvi oder *.pdf).

Hier wird auch der essentielle Unterschied zu WYSIWYG deutlich: der Compiler erzeugt die Ausgabe und trifft die endgültigen Layout-Entscheidungen, nicht der Benutzer! Denn der Autor kann nur 'Wünsche' äußern an welcher Stelle er welches Element sehen möchte. Die finale Entscheidung an welcher Stelle welches Element positioniert wird trifft das T_EX System. Dies erfordert einige Gewöhnung bis man akzeptiert, dass das automatisierte System zwar vielleicht kein so kreatives aber jedenfalls ein besser lesbares und konsistenteres Layout erzeugt als es der Mensch könnte. Wichtig ist auch, dass die Ausgabe (z. B. als *.pdf) ein recht statisches und nicht leicht zu veränderndes Kompilat darstellt. Dieses soll auch gar nicht verändert werden. Notwendige Korrekturen werden im Code vorgenommen und dieser anschließend rekompiliert.

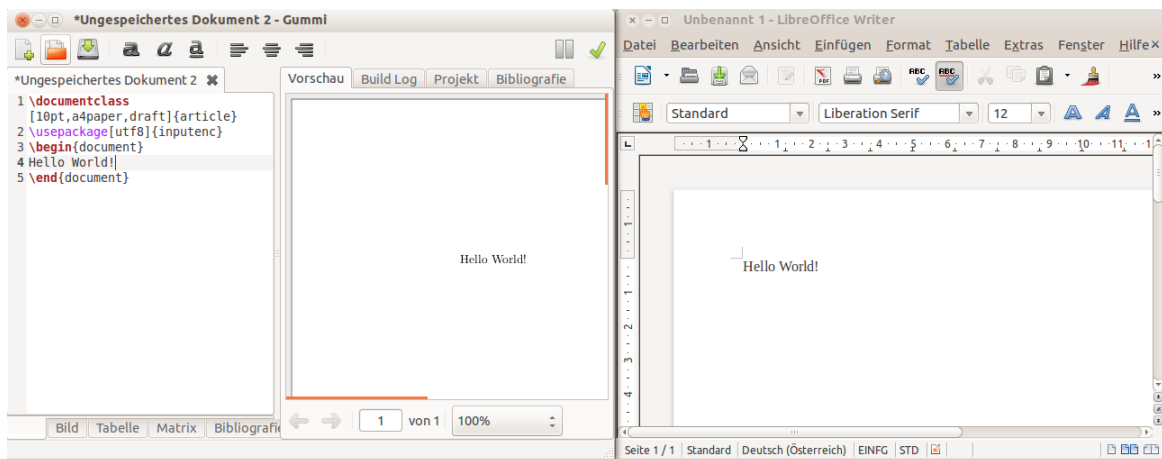


Abbildung 1: Einfaches Vergleichsbeispiel L^AT_EX vs. WYSIWYG

³üblicherweise MikTeX für Windows und TexLive für unixoide

2 Vorstellung der Editoren

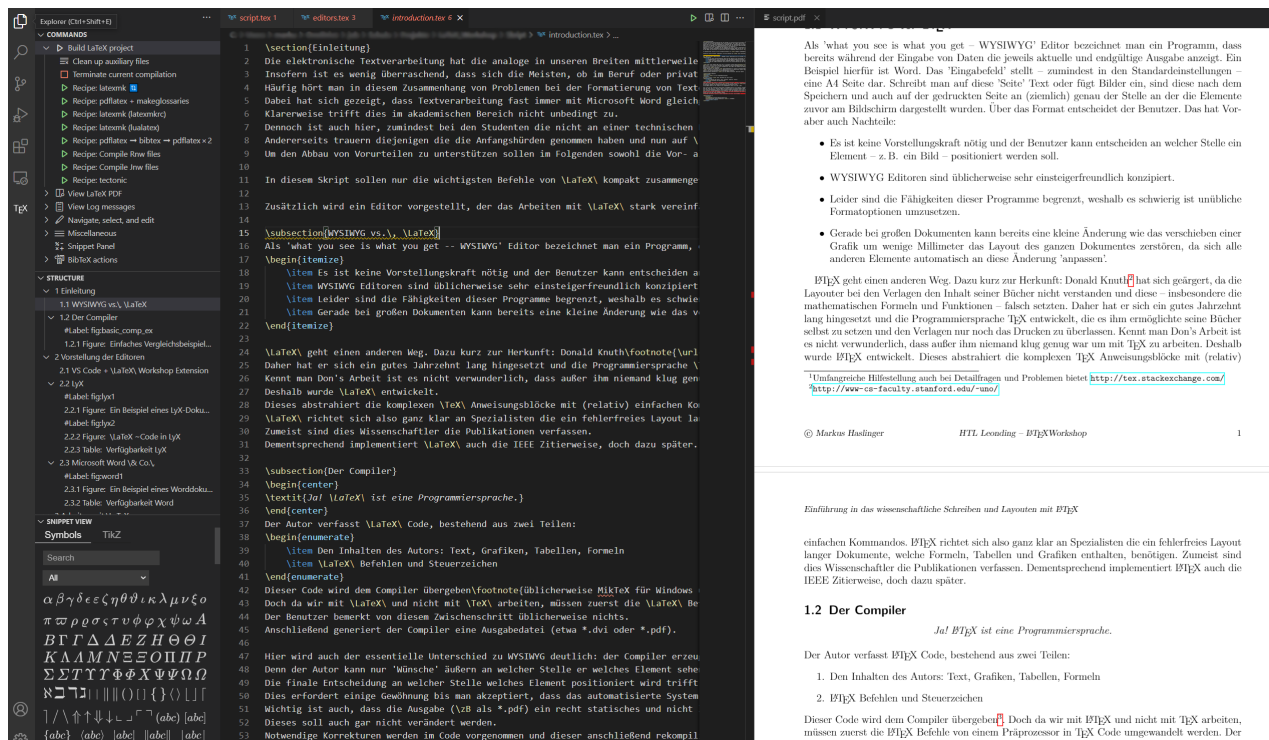
Auch für L^AT_EX gibt es Editoren die das Arbeiten unterstützen. Sei es mit einfachem Syntax-Highlighting und Autocomplete Funktionen oder mit mächtigen Assistenten für Routineaufgaben. Eine Beschreibung von Funktionen der Editoren erfolgt, wo sinnvoll, in Abschnitt 3.

2.1 VS Code + L^AT_EX Workshop Extension

Die derzeit beste 'L^AT_EX Erfahrung' bietet das weitverbreitete VisualStudio Code gemeinsam mit der L^AT_EX Workshop⁴ Erweiterung von James Yu.

Die Ansicht ist in drei Bereiche aufgeteilt:

1. Eine 'Tool' Area mit Build-Werkzeugen, der Dokumentenstruktur und eine Toolbox für Symbole
2. In der Mitte der Editor in dem der L^AT_EX Code geschrieben wird
3. Die Anzeige des gerenderten PDFs



Der Editor bietet Autocompletion und (große Empfehlung) mit einem der verfügbaren Rechtschreibkontrolle Plugins für VS Code aus eine solche. Die Anzeige wird dabei aktualisiert wenn die Datei gespeichert wird. Zudem ist es möglich mit **Ctrl+Click** zur jeweiligen Codestelle zu springen.

⁴<https://github.com/James-Yu/LaTeX-Workshop>

Vorteile:

- breite Verfügbarkeit
- alle notwendigen Features verfügbar
- nicht überladen
- gute Vorschaufunktion inkl. synctex
- aktive open source Entwicklung

Nachteile:

- kein WYSIWYG oder WYSIWYM – \LaTeX Kenntnisse erforderlich
- bei großen Dokumenten kann es lange dauern das PDF zu bauen
- kein Table-Generator
manche recipes müssen manuell ergänzt werden

Tipps:

- Aktuell nicht bearbeitete Kapitel auskommentieren (via `input`) reduziert die Compilezeit enorm
- Tabellen händisch schreiben ist umständlich – <https://www.tablesgenerator.com/> ist gratis und funktioniert gut

2.2 LyX

Der Editor LyX setzt bereits auf \LaTeX verfolgt jedoch einen WYSIWYM⁵ Ansatz. Dabei soll sich der Autor auf den Inhalt konzentrieren können während das Programm die \LaTeX Formatierung übernimmt, sprich die Markups transparent erzeugt. Die Eingabe entspricht also im Wesentlichen bereits der Ausgabe (wie bei WYSIWYG) wobei jedoch ausgabeformatabhängige Aspekte wie etwa Zeilenumbrüche oder die Position von Grafiken nicht final dargestellt werden (können).

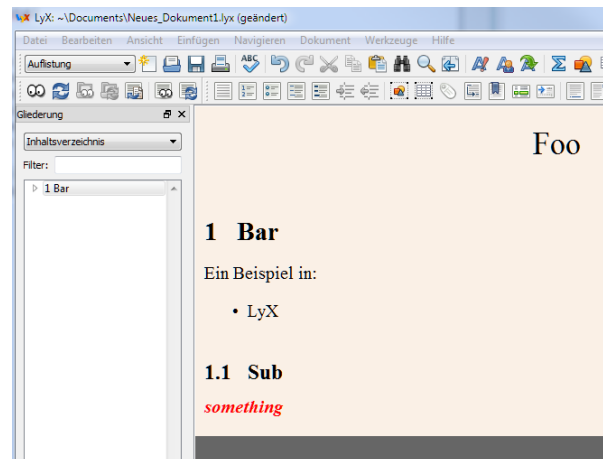


Abbildung 2: Ein Beispiel eines LyX-Dokuments

Ein Beispiel für die Bearbeitung eines Dokument in LyX ist in Abb. 2 ersichtlich. Wie in Word kann Text formatiert werden, indem dieser markiert wird und eine Formatvorlage wie zum Beispiel Titel, Auflistung oder Section ausgewählt wird – per Dropdown, Button, Kontextmenü oder Shortcut.

⁵what you see is what you mean

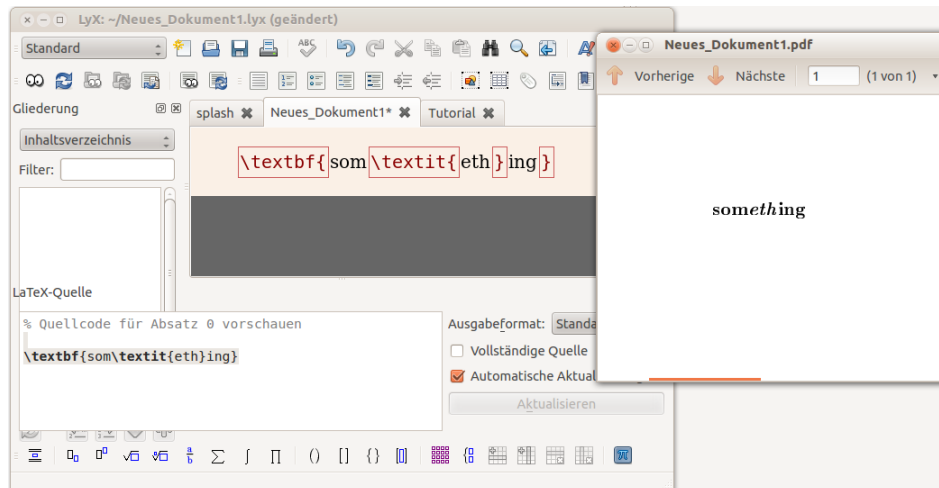


Abbildung 3: L^AT_EX Code in LyX

Zusätzlich ist es möglich direkt L^AT_EX Code einzugeben (siehe Abb. 3) sowie das erzeugte Dokument als L^AT_EX File zu exportieren und in einem anderen (lower Level) Editor die Feineinstellungen durchzuführen – der Rückweg zu LyX ist dann allerdings möglicherweise versperrt, da ein eigenes Dateiformat verwendet wird (*.lyx) und das Importieren nicht immer reibungslos klappt. Grundsätzlich sollten also alle Möglichkeiten von L^AT_EX erhalten bleiben während häufige Standardaufgaben entweder automatisiert oder stark GUI-unterstützt werden.

Vorteile:

- relativ Einsteigerfreundlich
- bei Standardaktionen einfach
- starke GUI Unterstützung
- bietet ungefähre Vorschau

Nachteile:

- eigenes Dateiformat
- Bearbeitung des (generierten) L^AT_EX Quellcodes nicht direkt möglich
- oft wäre man schneller wenn man einen Befehl schreibt als zur Maus zu greifen und die Option in LyX auszuwählen
- manuell eingegebener L^AT_EX Code stört die Darstellung wieder

2.3 Microsoft Word & Co.

Natürlich gibt es auch zahlreiche verschiedene WYSIWYG Editoren wie etwa LibreOffice oder Microsoft Word. In den meisten Betrieben wird hierzulande Word eingesetzt, viele verwenden dieses Programm also beruflich. Es sollte auch erwähnt werden, dass das Office Paket und somit auch Word essentieller Bestandteil des ECDL, in Österreich durch die Österreichische Computer Gesellschaft vergeben, ist. Generell kann behauptet werden, dass das Programm selbst durchaus benutzerfreundlich ist – sofern keine exakte Formatierung erforderlich ist und die Dokumente nicht zu lang werden. All die zuvor erwähnten Punkte tragen zur Verbreitung und Akzeptanz von Word bei.

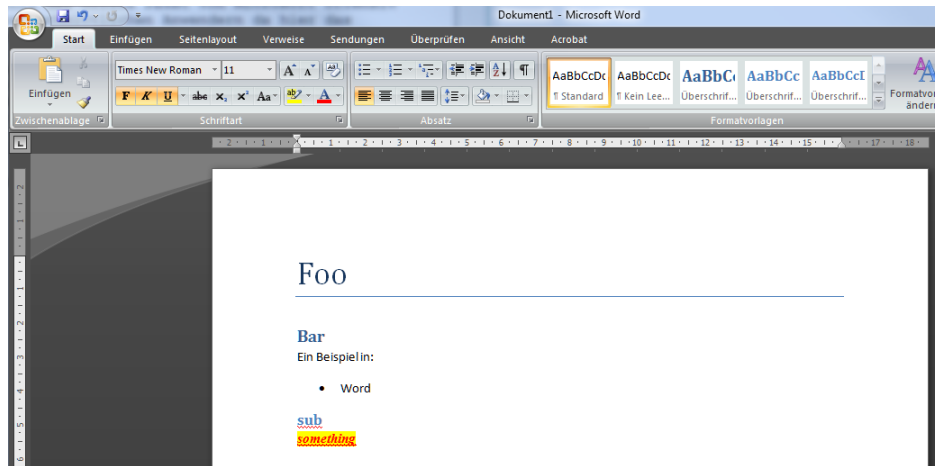


Abbildung 4: Ein Beispiel eines Worddokuments

Grundsätzlich soll bei WYSIWYG Editoren die Eingabe, bzw. die Repräsentation der Eingabe während der Bearbeitung, möglichst (genau) der Ausgabe, zum Beispiel als PDF oder auf einem Drucker, entsprechen. Formatierungen erfolgen dabei durch Auswahl (markieren) des zu formatierenden Textes und der Auswahl einer Formatierungsoption wie zum Beispiel Bold oder Italics entweder per Klick auf einen Button oder per Shortcut. Ein Beispiel für ein Worddokument während der Bearbeitung in Microsoft Word 2007 ist in Abb. 4 ersichtlich.

Vorteile:

- einsteigerfreundlich
- einfach
- weit verbreitet
- erfordert keine Vorstellungskraft
- kein Kompilieren erforderlich

Nachteile:

- nur einfach wenn Standardvorlagen verwendet werden
- Formatierung bei großen Dokumenten sehr schwierig und instabil
- Erstellen von Verzeichnissen nicht optimal
- Literaturverwaltung nicht optimal
- Formeleditor vorhanden aber nicht optimal
- Einsatz von VCS praktisch nicht möglich

3 Arbeiten mit \LaTeX

Die Ausgabe, also das Ergebnis, kann zum Beispiel eine pdf-Datei sein. Dieser Ansatz erfordert, sofern bisher ausschließlich mit WYSIWYG Editoren gearbeitet wurde, ein gewisses Umdenken. Obwohl das allgemeine Layout sehr exakt und mit vielfältigen Optionen und Paketen definiert werden kann, übernimmt die tatsächliche Positionierung der Elemente, innerhalb der vorgegebenen Richtlinien, das System ohne, dass der Benutzer wirklichen Einfluss darauf hat. Ein klassisches Beispiel dafür sind Grafiken. Hierbei kann der Autor zwar einen 'Wunsch' äußern an welcher Stelle eine Grafik eingebunden werden soll, die tatsächliche Position bestimmt jedoch das System um ein einheitliches Layout zu gewährleisten. Im Gegensatz zu etwa Word, bilden bei \LaTeX Editor und Compiler keine Einheit. Es muss also auf dem System \LaTeX installiert und eingerichtet sein um es zu verwenden, allerdings kann das selbe System mit verschiedensten Editoren oder auch ganz ohne einen solchen benutzt werden, während bei Word eine solche Trennung weder vorhanden noch sinnvoll ist. \LaTeX nutzt die Dateiendung *.tex.

Vor allem im akademischen Bereich wird \LaTeX gerne eingesetzt, da sich hierbei einige Vorteile wie die gute Einbindung von Literatur und Zitaten oder die, im Vergleich zu Word, problemlose Formatierung von, auch sehr großen, Dokumenten wie etwa Diplomarbeiten oder Dissertationen. Des Weiteren stellen die Verlage wie ACM oder IEEE üblicherweise ein Template zur Verfügung um den Vorgaben entsprechende Paper schreiben zu können. Ein weiteres Einsatzgebiet ist die Mathematik bzw. Informatik und Technik allgemein, da das Setzen von auch sehr komplexen Formeln nativ ausgezeichnet unterstützt wird.

Vorteile:

- Setzen von Formeln
- Literatur und Zitate
- einfaches Erzeugen von Verzeichnissen
- problemloses Formatieren auch großer Dokumente
- sehr vielseitig (Paper, Präsentation, Brief, Rechnung,...)
- zahlreiche Konfigurationsmöglichkeiten durch Zusatzpakete
- Einsatz von VCS möglich

Nachteile:

- erfordert Einarbeitungszeit
- jedenfalls komplexer zu bedienen als Word

3.1 Der header

Zumindest bei längeren Dokumenten lagert man die Präambel üblicherweise in eine header.tex Datei aus. Diese wird dann mit `\input{header}` in das Dokument eingebunden. In der Präambel werden die generellen Einstellungen für das ganze Dokument festgelegt, Pakete importiert und sofern notwendig Variablen und Befehle definiert. Die Möglichkeiten gehen weit über das Ziel dieser Einführung hinaus, weshalb im Folgenden nur die wichtigsten Elemente kurz beschrieben werden.

3.1.1 documentclass

Die documentclass beschreibt um welche Art von Dokument es sich grundsätzlich handelt (z. B. Artikel, Report, Brief, Rechnung, Präsentation, ...). Dieses Skript verwendet folgende documentclass: `\documentclass[12pt,a4paper,final,titlepage]{scrartcl}`.

- Die Basisschriftgröße ist auf 12pt eingestellt. Alle weiteren Größen wie etwa für Überschriften werden in Relation zu dieser Basis vom System berechnet.
- Die Ausgabe soll auf A4 Format erfolgen.
- final wird bei der Fertigstellung vergeben, während des Schreibens verwendet man draft.
- titlepage ist ein Befehl, der eine ganze Titelseite erzeugt – Standard ist nur ein Titelpopf.
- scrartcl beschreibt die class selbst. Es handelt sich um eine vordefinierte class aus dem KOMA-Skript Paket die für Artikel gedacht ist. Eine Alternative wäre z. B. scrreprt für einen Report (also etwa eine Diplomarbeit). Dies verändert etwa die Anzahl der Gliederungsebenen. In einem Artikel gibt es nur drei (z. B. 1.1.1) während in einem Report noch Kapitel erstellt werden können (z. B. 1.1.1.1). Der Autor muss entscheiden welche class er verwendet. Im Internet sind zahllose solcher classes mit den unterschiedlichsten Eigenschaften verfügbar.

3.1.2 packages

Pakete stellen weitere Funktionen zur Verfügung. Im Grunde handelt es sich um Methoden die jemand programmiert hat und die durch Angabe des Pakets in das eigene Dokument importiert und dort verwendet werden können.

Ein einfaches Beispiel ist `\usepackage{url}`. Wird dieser Befehl in der Präambel angegeben kann im Text der Befehl `\url{www.example.com}` verwendet werden. Die `\url{}` Funktion sorgt dann dafür, dass der übergebene Parameter, also die URL, entsprechend hervorgehoben wird: `www.example.com`. Es gibt eine enorm große Anzahl von Paketen im CTAN⁶ Archiv. Der MikTeX Compiler lädt Pakete bei der ersten Verwendung automatisch herunter, bei TeXLive muss dies manuell erledigt werden.

3.1.3 Dokumentenweite Einstellungen

Ebenfalls in der Präambel werden Einstellungen die das ganze Dokument betreffen angegeben. Beispiele hierfür sind etwa Kopf- und Fußzeilen oder die Seitenränder:

```
\geometry{a4paper,left=2cm,right=1.5cm,bottom=3cm,top=2.5cm}
\pagestyle{scrheadings}
\ihead[]{\footnotesize{Einfuehrung in das wissenschaftliche
  Schreiben und Layouten mit \LaTeX}}
\ohead[]{}
\chead[]{}

```

- Dem geometry Befehl wird das Seitenformat (A4) sowie die gewünschte Breite der Ränder übergeben. Dies wird Für das gesamte Dokument angewendet.

⁶<http://www.ctan.org/>

- Dem `pagestyle` Befehl wird die Anweisung `scrheadings` des KOMA-Skript Pakets übergeben, dadurch werden die Standard Kopf- und Fußzeile überschrieben und es kann jeweils ein rechter, linker und zentrierter Eintrag angegeben werden.
- Dem Befehl `ihead` wird der Text übergeben, der links in der Kopfzeile stehen soll, die Mitte (`chead`) und der rechte Teil (`ohead`) bleiben leer.

3.1.4 Einsatz einer Vorlage

Bei der Verwendung einer Vorlage wird häufig einfach ein header File importiert (wie zuvor beschrieben). Damit werden die Layoutvorgaben automatisch für das Dokument übernommen. Natürlich machen weitere oder geänderte Befehle des Autors diesen Effekt wieder zunichte. Die großen Verlage wie IEEE oder ACM prüfen bei der verpflichtenden elektronischen Einreichung automatisch ob sämtliche Layoutvorschriften exakt so eingehalten werden wie es die Vorlage vorsieht – ist dies nicht der Fall wird die Publikation sofort und ungelesen abgewiesen.

Teilweise erfordert die Vorlage, dass der Benutzer Daten ergänzt z.B. seinen Namen oder den Titel der Arbeit. Ist eine solche Eingabe erforderlich, wird darauf normalerweise mittels eines Kommentars hingewiesen. Ein L^AT_EX Kommentar wird mit einem vorgestellten % gekennzeichnet. Alles weitere in dieser Zeile folgende wird vom Compiler ignoriert. Beispiel:

```
%enter author name
\author{Markus Haslinger}
%enter paper title
\title{Einfuehrung in das wissenschaftliche Schreiben und Layouten
      mit \LaTeX\}
```

3.2 Strukturbefehle

Generell beginnt ein L^AT_EX Dokument mit der, zuvor beschriebenen, Präambel. Anschließend wird der Beginn des 'body' Teils mittels `\begin{document}` gekennzeichnet. Nun wird der eigentliche Inhalt geschrieben. Beendet wird das Dokument mit dem Befehl `\end{document}` – alles was nach diesem Kommando noch steht wird vom Compiler ignoriert.

3.2.1 Abschnitte

Ein wissenschaftliches Dokument ist in Abschnitte (und Kapitel) unterteilt. In L^AT_EX erfolgt dies mit den folgenden Befehlen:

```
\chapter{Name des Kapitels} (nicht in allen classes verfügbar!)
...\section{Name des Abschnitts}
.....\subsection{Name des Unter-Abschnitts}
.....\subsubsection{Name des Unter-Unter-Abschnitts}
```

```
\subsubsection{Abschnitte}
```

Ein wissenschaftliches Dokument ist in Abschnitte (und Kapitel) unterteilt. In \LaTeX\ erfolgt dies mit den folgenden Befehlen:

Wird hinter das Abschnittskommando noch ein '*' geschrieben (z. B. `\section*{don't show me}`), so erscheint dieser Eintrag nicht im Inhaltsverzeichnis, siehe Abschnitt 3.7.1. Die Nummerierung erfolgt automatisch durch das System, auch wenn ein Abschnitt hinzukommt oder gelöscht wird, wird die Nummerierung automatisch beim Kompilieren aktualisiert.

3.2.2 Aufzählungen

L^AT_EX unterstützt zwei Arten von Aufzählungen: automatisch aufsteigend nummerierte (`enumerate`) und einfache (`itemize`) Listen. Standard sind arabische Ziffern bzw. Punkte. Diese Aufzählungssymbole können grundsätzlich beliebig verändert werden, dafür ist jedoch zumindest minimaler Programmieraufwand notwendig, weshalb diese Einführung das nicht weiter behandelt. Jeder Aufzählungspunkt wird dabei durch den Befehl `\item` gekennzeichnet.

- a
- b
- c

```
1 \begin{itemize}
2   \item a
3   \item b
4   \item c
5 \end{itemize}
```

1. a
2. b
3. c

```
1 \begin{enumerate}
2   \item a
3   \item b
4   \item c
5 \end{enumerate}
```

Sehr nützlich sind die Umgebungen `compactitem` & `compactenum` aus dem `paralist` Package. Diese reduzieren die, oft unnötig großen, Abstände zwischen den Aufzählungspunkten.

Das Verschachteln (Nesting) von Aufzählungen ist ebenfalls möglich. Dazu wird als 'Inhalt' eines `item` eine neue Aufzählungsumgebung begonnen.

3.3 Schriftgröße & Effekte

Wie zuvor bereits erwähnt werden alle Schriftgrößen im Dokument auf Basis der in der documentclass angegebenen Basis berechnet. Daher kann man nicht einstellen 'dieses Wort soll 18pt groß sein'. Es kann, um die Einheitlichkeit zu garantieren, nur mit Relationen gearbeitet werden. Anmerkung: `\\` erzwingt einen Zeilenumbruch.

Text
Text
Text
Text
Text

```
1 Text \\ % Standard
2 \tiny{Text} \\
3 \small{Text} \\
4 \large{Text} \\
5 \Large{Text} % l vs. L!
```

Des weiteren ist es möglich Text fett oder kursiv zu gestalten.

Text
Text
Text

```
1 \textbf{Text} \\ % fett
2 \textit{Text} \\ % kursiv
3 \textbf{\textit{Text}} % kombiniert
```

Manchmal ist es, aufgrund eines bekannten aber nie behobenen Fehlers, erforderlich diese Syntax zu verwenden: `\tiny <beliebiger Text>` (anstelle von *tiny* kann jeder zuvor genannte Befehl genutzt werden).

3.4 Tabellen

Tabellen gehören zu den anspruchsvollsten Aspekten in L^AT_EX.

eine	einfache	Tabelle
mit		Inhalt
und einer verbundenen Zelle		

Tabelle 0: eine Beispieltabelle

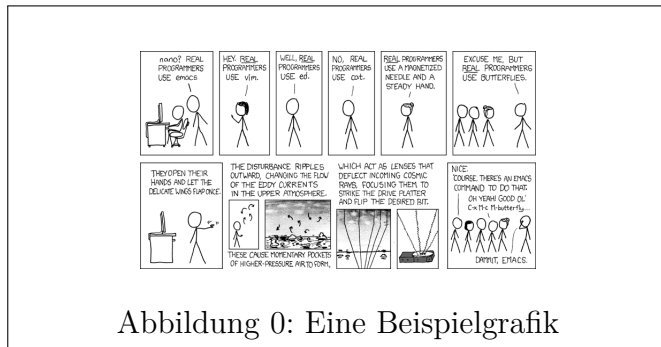
```
1 \begin{table}
2 \begin{center}
3 \begin{tabular}{|l|c|r|}
4 \hline
5 eine & einfache & Tabelle \\
6 \hline
7 mit & ~ & Inhalt \\
8 \hline
9 \multicolumn{3}{|l|}{und einer
10 verbundenen Zelle} \\
11 \hline
12 \end{tabular}
13 \end{center}
14 \caption{eine Beispieltabelle}
15 \end{table}
```

Zuerst muss mit `\begin{table}` eine Tabellenumgebung erzeugt werden. Anschließend schreibt man `\begin{center}` um alles Folgende auf der Seite zentriert darzustellen (in diesem Beispiel repräsentiert das Rechteck in dem die Ausgabe zu sehen ist die Seite, die Tabelle wird also im Rechteck zentriert). Nun beginnt erst die eigentliche Tabelle mittels `\begin{tabular}`. Der Ausdruck `'|l|c|r|'` definiert die Spalten, die Randlinien und die Zellausrichtung. Jeder vertikale Strich `'|'` bewirkt, dass hier eine vertikale Linie gezeichnet wird. Würde man `'|l|cr|'` schreiben, so wären es immer noch 3 Spalten, zwischen zweiter und dritter würde jedoch keine Trennlinie gezeichnet. Die Ausrichtung wird durch die Buchstaben `'l'` für linksbündig, `'c'` für zentriert und `'r'` für rechtsbündig definiert. Jede Reihe von Zellen wird mit `\hline` abgeschlossen, dadurch wird eine horizontale Trennlinie gezeichnet. Zu beachten ist auch, dass auch die allererste und die letzte horizontale Linie explizit angegeben werden müssen.

Eine Reihe wird definiert durch `'< Text > & < Text > & ... & < Text > \\'`. Noch komplexer sind verbundene Zellen die mit dem Befehl `\multicolumn{}{}` erzeugt werden können. Die Tabelle wird beendet mit dem Befehl `\end{tabular}`. Anschließend wird auch die center Umgebung geschlossen. Nun folgt die Beschriftung mit dem Befehl `\caption{}` – diese Beschriftung wird auch in ein etwaige Tabellenverzeichnis übertragen. Die Nummerierung erfolgt wiederum automatisch durch das System. Schließlich muss noch die Tabellenumgebung mittels `\end{table}` geschlossen werden. Generell ist sehr zu empfehlen für das Erstellen einer Tabelle einen Assistenten zu verwenden.

3.5 Grafiken

Grafiken werden in einer figure Umgebung eingefügt. Der Autor kann einen Wunsch äußern wo diese Grafik platziert werden soll: 'h' = genau hier, 't' = am Anfang der Seite, 'b' = am Ende der Seite. Dabei können auch mehrere Präferenzen angegeben werden, entschieden wird jedoch in jedem Fall der Compiler. Möglicherweise wird die Grafik auch auf einer eigenen oder ganz anderen Seite positioniert.



```

1 \begin{figure}[h t]
2 \centering
3 \includegraphics[scale=0.2]{pics/
   xkcd1.png}
4 \caption{Eine Beispielgrafik}
5 \end{figure}

```

Das funktioniert ähnlich wie bei einer Tabelle, ist jedoch etwas einfacher. Statt `\begin{center}` und `\end{center}` kann auch `\centering` verwendet werden. Die Option `scale=(0.0-1.0)` gibt an in welcher Relation zur Originalgröße das Bild dargestellt werden soll.

3.6 Codeausschnitte

Arbeiten zu Informatikthemen enthalten üblicherweise auch Codeausschnitte. Dabei ist darauf zu achten, dass nicht seitenweise Code eingefügt wird, sondern nur die minimale Menge an Zeilen ausgewählt wird, die für das Verständnis eines besonderen wichtigen Aspekts erforderlich ist. Dieses Feature ist nicht standardmäßig vorhanden und muss mittels `listings` package hinzugefügt werden.

```
\begin{lstlisting}[numbers=left]
var result = await Service.GetData<CalculationResult>(1234);
return result.Sum;
\end{lstlisting}
```

Umfangreiche Konfigurationsmöglichkeiten (Keywords, Farben, Zeilennummerierung, Überschriften etc.) sind mittels `\lstset` möglich.

3.7 Referenzen

Die Referenzen sind eine große Stärke von \LaTeX . Da die Nummerierung und Anordnung von Abschnitten, Grafiken, Einträgen im Literaturverzeichnis etc. automatisch erfolgt, müsste man Verweise im Text darauf ständig manuell aktuell halten (wie es z. B. bei Word auch der Fall ist). Das ist nicht nur sehr zeitaufwändig, sondern auch fehleranfällig. Aus diesem Grund nimmt einem \LaTeX praktisch die gesamte Arbeit des Referenzierens ab.

*Wichtig: Das Dokument muss mindestens zwei Mal unverändert kompiliert werden damit L^AT_EX die Referenzen richtig zuordnen kann!*⁷

⁷Editoren wie VS Code haben das im recipe üblicherweise bereits so hinterlegt.

3.7.1 Verzeichnisse

Verzeichnisse kann L^AT_EX selbstständig erstellen und vor allem auch aktuell halten.

- `\tableofcontents` erzeugt an dieser Stelle das Inhaltsverzeichnis, welches automatisch alles Abschnitte (=section & chapter) enthält die nicht durch einen '*' ausgenommen wurden. Ändert sich Reihenfolge, Bezeichnung oder Anzahl der Abschnitte werden diese Änderungen selbstständig im Index übernommen. Gleiches gilt für die angegebene Seitenzahl.
- `\listoftables` erzeugt das Tabellenverzeichnis.
- `\listoffigures` erzeugt das Abbildungsverzeichnis.

3.7.2 Zitate (BibTeX)

Für das Literaturverzeichnis nutzt L^AT_EX die Technologie BibTeX. Dabei wird eine *.bib Datei erstellt, die die Metadaten jeder Literaturquelle enthält. Auf den Aufbau dieser Datei wird in dieser Einführung nicht weiter eingegangen, da die *.bib Datei von Mendeley, siehe Abschnitt 4, automatisch erzeugt werden kann.

Relevant ist allerdings, dass zu jeder Quelle eine BibTeX-Referenz hinterlegt ist z. B. Knuth96. Auch diese Referenz auf eine Quelle kann aus Mendeley entnommen werden. Im L^AT_EX Dokument muss jedoch, wenn auf einen Quelle verwiesen werden soll, an der entsprechenden Stelle im Dokument der Befehl `\cite{}` angegeben werden – z. B. `\cite{Knuth96}`. Dadurch wird von L^AT_EX automatisch eine IEEE konforme Literaturreferenz auf den entsprechenden Eintrag im Literaturverzeichnis erzeugt.

Das Literaturverzeichnis selbst kann mit folgenden Befehlen an der entsprechenden Stelle im Dokument eingefügt werden:

```
\bibliographystyle{plain} % oder: IEEEtran
\bibliography{bib}
```

Hinweis: Im Literaturverzeichnis scheinen der Übersichtlichkeit wegen nur die Einträge auf, die im Dokument bereits referenziert wurden und nicht alle im *.bib File definierten.

3.7.3 Fußnoten

Fußnoten werden einfach an der gewünschten Stelle im Text an der die Referenz (eine hochgestellte Zahl, die Fußnoten werden automatisch nummeriert) aufscheinen soll mit dem Befehl `\footnote{Text}` definiert.

3.7.4 Querverweise

Möchte man innerhalb eines Dokuments einen Querverweis auf einen Abschnitt oder eine Grafik etc. anlegen, verwendet man dazu das `\label{type:reference}`. L^AT_EX fügt dann automatisch an dieser Stelle die richtige Referenz ein und hält diese auch bei Änderungen aktuell.

```
... fuer Verzeichnisse siehe Abschnitt \ref{sec:verzeichnisse} und
    Abbildung \ref{fig:bsp_grafik} ...
[...]
\subsubsection{Verzeichnisse}\label{sec:verzeichnisse}
\begin{figure}[h t]
\centering
\includegraphics[scale=0.2]{pics/xkcd1.png}
\caption{Eine Beispielgrafik}
\label{fig:bsp_grafik}
\end{figure}
```

Als Konvention verwendet man das Schema:

- für Grafiken `fig:xyz`
- für Tabellen `tab:xyz`
- für Abschnitte `sec:xyz`

Natürlich muss jedes Label einzigartig sein.

3.8 Formeln

Zur Eingabe mathematischer Formeln können drei unterschiedliche Methoden gewählt werden.

1. `$...$` für kurze Formeln oder Gleichungen im Fließtext.
2. `displaymath` Umgebung: erzeugt einen eigenen zentrierten Bereich für die Formel.
3. `equation` Umgebung: wie `displaymath`, erstellt jedoch zusätzlich eine Referenz.

Das ist ein $y = ax^2 + bx + c$ Satz.

$$y = ax^2 + bx + c$$

$$y = ax^2 + bx + c \quad (1)$$

```
1 Das ist ein $ y=ax^2+bx+c $ Satz. \\
2 \begin{displaymath}
3 y = ax^2 + bx + c
4 \end{displaymath}
5 \begin{equation}
6 y = ax^2 + bx + c
7 \end{equation}
```

3.9 Sonderzeichen und Konventionen

Ausgabe	\LaTeX Code
\LaTeX	<code>\LaTeX\</code>
z.B.	<code>z.\,B.\,</code>
bzw.	<code>bzw.\,</code>
Abb.	<code>Abb.\,</code>
Gedankenstrich ‘-’	<code>--</code>
$\&$	<code>\&</code>
Zeilenumbruch	<code>\\</code>
neue Zeile	<code>\newline</code>
neue Seite	<code>\newpage</code>
hochstellen	<code>abc</code>
tiefgestellt	<code>\textsubscript{abc}</code>

Tabelle 2: Tabelle mit einigen Sonderzeichen und Konventionen

4 Mendeley Literaturverwaltung

Mendeley ist ein kostenloses Programm zur Literaturverwaltung. Es kann entweder die Web-Version benutzt oder eine Applikation für den PC heruntergeladen werden⁸. Alle Einträge eines Benutzers werden über die Cloud auf allen Geräten synchronisiert. Des weiteren können Dokumente (z. B. ein *.pdf) bei einem Eintrag hinterlegt und ebenfalls in der Cloud gespeichert werden (2GB Speicherplatz kostenlos). Quellen können mit anderen geteilt und gemeinsam verwaltet werden (Stichwort: Arbeit im Team). Literatureinträge können entweder manuell, durch Eingabe aller verfügbaren Daten wie Titel, Autor, etc. erfasst oder automatisch importiert werden. [1]

Die Importfunktion wird von allen großen Datenbanken (IEEE, ACM, Springer, ...) unterstützt. Da für Schüler leider noch kein Zugang zu echten Uni-Bibliotheken besteht werden in vielen Fällen nur Webquellen und Bücher aus physischen Bibliotheken übrig bleiben (eventuell noch Google Scholar). Daher werden die Einträge in vielen Fällen manuell angelegt werden müssen. Bitte bei schlechten Quellen (wie es Webquellen immer sind) darauf achten, dass der Eintrag *nicht* im Catalog veröffentlicht wird, siehe Abb. 7. Damit blamiert man sich in der akademischen Welt bloß. Erfasste Einträge können natürlich jederzeit bearbeitet werden. Über das Kontextmenü jedes Eintrages kann man die BibTeX Referenz in die Zwischenablage kopieren und dann im L^AT_EX Code einfügen.

Markiert man mehrere Einträge kann man ebenfalls über das Kontextmenü ein *.bib File exportieren, das dann (im richtigen Ordner platziert⁹) von L^AT_EX verwendet werden kann.

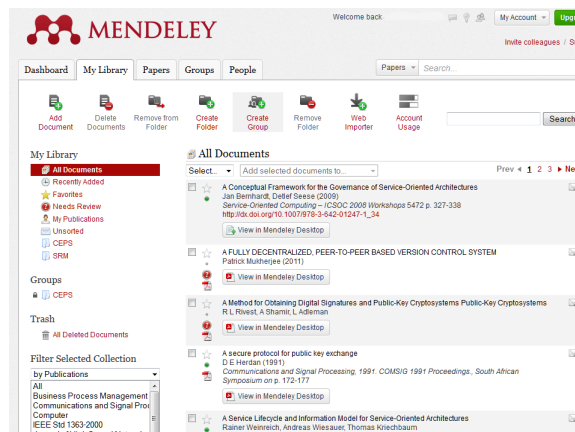


Abbildung 6: Ausschnitt aus der Mendeley Web-Version

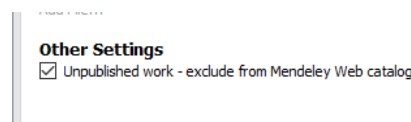


Abbildung 7: Schlechte Quellen nicht in den Catalog eintragen

⁸siehe <http://www.mendeley.com/>

⁹Grundsätzlich sollten sich alle Dateien die L^AT_EX benötigt im selben Ordner befinden. Verwendet man z. B. einen Unterordner für Grafiken, muss man im Code natürlich den entsprechenden Pfad angeben.

5 Präsentationen mit L^AT_EX

Es ist auch möglich professionelle Präsentationen mit L^AT_EX zu erstellen. Dabei wird als Ausgabe ebenfalls wieder eine *.pdf Datei erstellt, wobei jede Folie auf einer Seite gedruckt wird. Aktuelle PDF-Anzeigeprogramme haben meistens einen entsprechenden Präsentationsmodus integriert, der die Seiten als Vollbild und ohne Werkzeugleisten darstellt. Auf das Einbinden von animierten Grafiken sowieso Multimediadateien sollte man im Allgemeine verzichten. Ist es jedoch unbedingt notwendig so existieren Pakete die derartige Effekte auch im *.pdf ermöglichen.

Um nun eine Präsentation zu erstellen kann die documentclass **beamer** verwendet werden. Es gibt eine große Anzahl an 'Designs' für die Folien, hier wird jedoch nicht näher auf diese eingegangen. In der header Datei sind einige Einstellungen notwendig, daher werden hier recht allgemein einsetzbare angeführt, wobei Titel, Autor und Datum natürlich entsprechend angepasst werden müssen:

```
\documentclass[hyperref={pdfpagelabels=false}]{beamer}
\usepackage{lmodern}
\usepackage[utf8]{inputenc}
\usepackage[naustrian, ngerman]{babel}
\title{presentation title}
\author{author name}
\date{\today}
\usepackage{beamerthemebars}
\beamerunsetcovermixins{\opaqueness<1>{25}}{\opaqueness<2->{15}}
\setbeamertemplate{footline}
{%
\begin{beamercolorbox}[wd=0.5\textwidth,ht=3ex,dp=1.5ex,leftskip=.5
  em,rightskip=.5em]{author in head/foot}%
\usebeamerfont{author in head/foot}%
\insertframenum{} von \inserttotalframenum\hfill\
  insertshortauthor%
\end{beamercolorbox}%
\vspace*{-4.5ex}\hspace*{0.5\textwidth}%
\begin{beamercolorbox}[wd=0.5\textwidth,ht=3ex,dp=1.5ex,left,
  leftskip=.5em]{title in head/foot}%
\usebeamerfont{title in head/foot}%
\insertshorttitle%
\end{beamercolorbox}%
}
```

Sind diese Anfangseinstellungen erledigt wird wie immer mit `\begin{document}` der body Teil eröffnet. Anschließend wird jede Folie einzeln als Bereich mittels **frame** Umgebung definiert. Dies sieht z. B. so aus:

```
\begin{frame}
\frametitle{Anforderungen III}
Des Weiteren muss dieses System
\begin{itemize}
\item dynamisch
```

```
\item erweiterbar  
\end{itemize}  
sein.  
\end{frame}
```

- Mittels `\frametitle{}` wird der Titel der Folie angegeben.
- Anschließend kann Text eingefügt werden. Bei Präsentationen üblich ist der Einsatz von Auflistungen. Auch Grafiken können wie immer mittels `figure` Umgebung eingefügt werden.

Eine Titelfolie kann mit folgendem Befehl erzeugt werden:

```
\begin{frame}  
\titlepage  
\end{frame}
```

Auch der Einsatz von `\section{}` und tieferen Gliederungsebenen ist möglich und sinnvoll. Denn für jede einem Abschnitt zugeordnete Folie wird ein 'Punkt' im Titelbereich der Folien angelegt wodurch der Zuhörer den Fortschritt gut mitverfolgen kann, siehe Abb. 8. Der Name des Abschnitts wird ebenfalls angezeigt.

Anforderungen	Status
○● ○○○○○	○ ○○○○
Anforderungen Service Repository Browsing	
Anforderungen II	

► Bewertung der Services

Abbildung 8: Fortschrittsanzeige der Präsentation

Verwendet man Abschnitte, kann natürlich auch ein Index generiert werden. Folgender Befehl erzeugt eine 'Agenda' Folie die den Inhalt der Präsentation enthält.

```
\begin{frame}  
\frametitle{Agenda}  
\tableofcontents  
\end{frame}
```

Literatur

- [1] Mendeley Ltd. Stay on top of your research. Internet: <http://mendeley.com>. last retrieved May 30, 2021.

Anhang

L^AT_EX 2_ε-Kurzbeschreibung

Version 2.3
10. April 2003

Walter Schmidt*
Jörg Knappen
Hubert Partl
Irene Hyna

L^AT_EX ist ein Satzsystem, das für viele Arten von Schriftstücken verwendet werden kann, von einfachen Briefen bis zu kompletten Büchern. Besonders geeignet ist es für wissenschaftliche oder technische Dokumente. L^AT_EX ist für praktisch alle verbreiteten Betriebssysteme verfügbar.

Die vorliegende Kurzbeschreibung bezieht sich auf die Version L^AT_EX 2_ε in der Fassung vom Juni 2001 und sollte für den Einstieg in L^AT_EX ausreichen. Eine vollständige Beschreibung enthält das *L^AT_EX-Handbuch* [1] in Verbindung mit der Online-Dokumentation.

*<w-a-schmidt@arcor.de>

Copyright © 1998–2003 W. Schmidt, J. Knappen, H. Partl, I. Hyna

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Die in dieser Publikation erwähnten Software- und Hardware-Bezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Dieses Dokument wurde mit L^AT_EX gesetzt. Es ist als Quelltext und im PDF-Format online erhältlich:

`<ftp://dante.ctan.org/tex-archive/info/lshort/german/>`

Die Autoren bedanken sich bei Luzia Dietsche, Michael Hofmann, Peter Karp, Rolf Niepraschk, Heiko Oberdiek, Bernd Raichle, Rainer Schöpf und Stefan Steffens für Tips, Anmerkungen und Korrekturen.

Inhaltsverzeichnis

1	Allgemeines	3
1.1	The Name of the Game	3
1.1.1	TeX	3
1.1.2	L ^A T _E X	3
1.1.3	L ^A T _E X 2 _ε	3
1.2	Grundkonzept	3
1.2.1	Autor, Designer und Setzer	3
1.2.2	Layout-Design	4
1.2.3	Vor- und Nachteile	4
1.2.4	Der Arbeitsablauf	5
2	Eingabefile	5
2.1	Leerstellen	6
2.2	L ^A T _E X-Befehle und Gruppen	6
2.3	Kommentare	6
2.4	Aufbau	7
2.5	Dokumentklassen	7
2.6	Pakete	8
2.7	Eingabezeichensatz	9
3	Setzen von Text	12
3.1	Deutschsprachige Texte	12
3.2	Zeilen- und Seiten-Umbruch	12
3.2.1	Blocksatz	12
3.2.2	Silbentrennung	13
3.3	Wortabstand	14
3.4	Spezielle Zeichen	15
3.4.1	Anführungszeichen	15
3.4.2	Binde- und Gedankenstriche	15
3.4.3	Punkte	15
3.4.4	Ligaturen	16
3.4.5	Symbole, Akzente und besondere Buchstaben	16
3.5	Kapitel und Überschriften	18
3.6	Fußnoten	18
3.7	Hervorgehobener Text	19
3.8	Hochgestellter Text	19
3.9	Umgebungen	19
3.9.1	Zitate (quote, quotation, verse)	19
3.9.2	Listen (itemize, enumerate, description)	20
3.9.3	Flattersatz (flushleft, flushright, center)	22
3.9.4	Direkte Ausgabe (verbatim, verb)	22
3.9.5	Tabulatoren (tabbing)	22
3.9.6	Tabellen (tabular)	23

4	Setzen von mathematischen Formeln	24
4.1	Allgemeines	24
4.2	Elemente in mathematischen Formeln	25
4.3	Nebeneinander Setzen	28
4.4	Übereinander Setzen	29
4.5	Liste der mathematischen Symbole	30
5	Setzen von Bildern	34
6	Seitenaufbau	34
6.1	Kopf- und Fußzeilen	34
6.2	Gleitobjekte	35
6.2.1	Abbildungen (figure)	35
6.2.2	Tabellen (table)	36
7	Schriften	37
7.1	Schriftgrößen	37
7.2	Schriftstil	38
7.3	Andere Schriftfamilien	39
7.4	Die „europäischen“ Zeichensätze	39
8	Spezialitäten	41
8.1	Abstände	41
8.1.1	Zeilenabstand	41
8.1.2	Spezielle horizontale Abstände	41
8.1.3	Spezielle vertikale Abstände	42
8.2	Briefe	43
8.3	Literaturangaben	44
8.4	Zerbrechliche Befehle	44
A	Mit dem Paket textcomp verfügbare Symbole	45

1 Allgemeines

1.1 The Name of the Game

1.1.1 \TeX

\TeX (sprich „Tech“, kann auch „TeX“ geschrieben werden) ist ein Computerprogramm von Donald E. Knuth [17, 18]. Es dient zum Setzen von Texten und mathematischen Formeln.

1.1.2 \LaTeX

\LaTeX (sprich „Lah-tech“ oder „Lej-tech“, kann auch „LaTeX“ geschrieben werden) ist ein auf \TeX aufbauendes Computerprogramm und wurde von Leslie Lamport [1, 3] geschrieben. Es vereinfacht den Umgang mit \TeX , indem es entsprechend der logischen Struktur des Dokuments auf vorgefertigte Layout-Elemente zurückgreift.

1.1.3 $\text{\LaTeX 2}_{\epsilon}$

$\text{\LaTeX 2}_{\epsilon}$ (sprich „ \LaTeX zwei e“) ist die aktuelle Variante von \LaTeX seit dem 1. Juni 1994. (Die vorherige hieß \LaTeX 2.09 .) Wenn hier von \LaTeX gesprochen wird, so ist normalerweise dieses $\text{\LaTeX 2}_{\epsilon}$ gemeint.

Neue Versionen von $\text{\LaTeX 2}_{\epsilon}$ (z. B. mit Fehlerberichtigungen oder Ergänzungen) erscheinen jährlich im Juni; die vorliegende Beschreibung setzt mindestens diejenige vom Juni 2001 voraus.

1.2 Grundkonzept

1.2.1 Autor, Designer und Setzer

Für eine Publikation übergab der Autor dem Verleger traditionell ein maschinengeschriebenes Manuskript. Der Buch-Designer des Verlages entschied dann über das Layout des Schriftstücks (Länge einer Zeile, Schriftart, Abstände vor und nach Kapiteln usw.) und schrieb dem Setzer die dafür notwendigen Anweisungen dazu. \LaTeX ist in diesem Sinne der Buch-Designer, das Programm \TeX ist sein Setzer.

Ein menschlicher Buch-Designer erkennt die Absichten des Autors (z. B. Kapitel-Überschriften, Zitate, Beispiele, Formeln ...) meistens aufgrund seines Fachwissens aus dem Inhalt des Manuskripts. \LaTeX dagegen ist „nur“ ein Programm und benötigt daher zusätzliche Informationen vom Autor, die die logische Struktur des Textes beschreiben. Diese Informationen werden in Form von sogenannten „Befehlen“ innerhalb des Textes angegeben. Der Autor braucht sich also (weitgehend) nur um die logische Struktur seines Werkes zu kümmern, nicht um die Details von Gestaltung und Satz.

Im Gegensatz dazu steht der visuell orientierte Entwurf eines Schriftstückes mit Textverarbeitungs- oder DTP-Programmen wie z. B. WORD. In diesem Fall legt der Autor das Layout des Textes gleich bei der interaktiven Eingabe fest. Dabei sieht er am Bildschirm das, was auch auf der gedruckten Seite stehen

wird. Solche Systeme, die das visuelle Entwerfen unterstützen, werden auch WYSIWYG-Systeme („what you see is what you get“) genannt.

Bei \LaTeX sieht der Autor beim Schreiben des Eingabefiles in der Regel noch nicht sofort, wie der Text nach dem Formatieren aussehen wird. Er kann aber jederzeit einen „Probe-Ausdruck“ seines Schriftstücks auf dem Bildschirm machen und danach sein Eingabefile entsprechend korrigieren und die Arbeit fortsetzen.

1.2.2 Layout-Design

Typographisches Design ist ein Handwerk, das erlernt werden muß. Ungeübte Autoren machen dabei oft gravierende Fehler. Fälschlicherweise glauben viele Laien, daß Textdesign vor allem eine Frage der Ästhetik ist – wenn das Schriftstück vom künstlerischen Standpunkt aus „schön“ aussieht, dann ist es schon gut „designed“. Da Schriftstücke jedoch gelesen und nicht in einem Museum aufgehängt werden, sind die leichtere Lesbarkeit und bessere Verständlichkeit wichtiger als das schöne Aussehen.

Beispiele: Die Schriftgröße und Numerierung von Überschriften soll so gewählt werden, daß die Struktur der Kapitel und Unterkapitel klar erkennbar ist. Die Zeilenlänge soll so gewählt werden, daß anstrengende Augenbewegungen des Lesers vermieden werden, nicht so, daß der Text das Papier möglichst schön ausfüllt.

Mit interaktiven visuellen Entwurfssystemen ist es leicht, Schriftstücke zu erzeugen, die zwar „gut“ aussehen, aber ihren Inhalt und dessen Aufbau nur mangelhaft wiedergeben. \LaTeX verhindert solche Fehler, indem es den Autor dazu zwingt, die logische Struktur des Textes anzugeben, und dann automatisch ein dafür geeignetes Layout verwendet.

Daraus ergibt sich, daß \LaTeX insbesondere für Dokumente geeignet ist, wo vorgegebene Gestaltungsprinzipien auf sich wiederholende logische Textstrukturen angewandt werden sollen. Für das – notwendigerweise – visuell orientierte Gestalten etwa eines Plakates ist \LaTeX hingegen aufgrund seiner Arbeitsweise weniger geeignet.

1.2.3 Vor- und Nachteile

Gegenüber anderen Textverarbeitungs- oder DTP-Programmen zeichnet sich \LaTeX vor allem durch die folgenden Vorteile aus:

- Der Anwender muß nur wenige, leicht verständliche Befehle angeben, die die logische Struktur des Schriftstücks betreffen, und braucht sich um die gestalterischen Details (fast) nicht zu kümmern.
- Das Setzen von mathematischen Formeln ist besonders gut unterstützt.
- Auch anspruchsvolle Strukturen wie Fußnoten, Literaturverzeichnisse, Tabellen u. v. a. können mit wenig Aufwand erzeugt werden.
- Routineaufgaben wie das Aktualisieren von Querverweisen oder das Erstellen des Inhaltsverzeichnisses werden automatisch erledigt.

- Es stehen zahlreiche vordefinierte Layouts zur Verfügung.
- \LaTeX -Dokumente sind zwischen verschiedenen Installationen und Rechnerplattformen austauschbar.
- Im Gegensatz zu vielen WYSIWYG-Programmen bearbeitet \LaTeX auch lange oder komplizierte Dokumente zuverlässig, und sein Ressourcenverbrauch (Speicher, Rechenleistung) ist vergleichsweise mäßig.

Ein Nachteil soll freilich auch nicht verschwiegen werden:

- Innerhalb der von \LaTeX unterstützten Dokument-Layouts können zwar einzelne Parameter leicht variiert werden, grundlegende Abweichungen von den Vorgaben sind aber nur mit größerem Aufwand möglich (Design einer neuen Dokumentklasse, siehe [10, 13, 14, 15].)

1.2.4 Der Arbeitsablauf

Der typische Ablauf beim Arbeiten mit \LaTeX ist:

1. Ein Eingabefile schreiben, das den Text und die \LaTeX -Befehle enthält.
2. Dieses File mit \LaTeX bearbeiten; dabei wird eine Datei erzeugt, die den gesetzten Text in einem geräteunabhängigen Format (DVI, PDF oder auch PostScript) enthält.
3. Einen „Probeausdruck“ davon auf dem Bildschirm anzeigen (Preview).
4. Wenn nötig, die Eingabe korrigieren und zurück zu Schritt 2.
5. Die Ausgabedatei drucken.

Zeitgemäße Betriebssysteme machen es möglich, daß der Texteditor und das Preview-Programm gleichzeitig in verschiedenen Fenstern „geöffnet“ sind; beim Durchlaufen des obigen Zyklus brauchen sie also nicht immer wieder von neuem gestartet werden. Nur die wiederholte \LaTeX -Bearbeitung des Textes muß noch von Hand angestoßen werden und läuft ebenfalls in einem eigenen Fenster ab.

Wie man auf die einzelnen Programme – Editor, \LaTeX , Previewer, Druckertreiber – in einer bestimmten Betriebssystemumgebung zugreift, muß in einem *Local Guide* [7] beschrieben sein.

2 Eingabefile

Das Eingabefile für \LaTeX ist ein Textfile. Es wird mit einem Editor erstellt und enthält sowohl den Text, der gedruckt werden soll, als auch die Befehle, aus denen \LaTeX erfährt, wie der Text gesetzt werden soll.

2.1 Leerstellen

„Unsichtbare“ Zeichen wie das Leerzeichen, Tabulatoren und das Zeilenende werden von \LaTeX einheitlich als Leerzeichen behandelt. *Mehrere* Leerzeichen werden wie *ein* Leerzeichen behandelt. Wenn man andere als die normalen Wort- und Zeilenabstände will, kann man dies also nicht durch die Eingabe von zusätzlichen Leerzeichen oder Leerzeilen erreichen, sondern nur mit entsprechenden \LaTeX -Befehlen.

Eine Leerzeile zwischen Textzeilen bedeutet das Ende eines Absatzes. *Mehrere* Leerzeilen werden wie *eine* Leerzeile behandelt.

2.2 \LaTeX -Befehle und Gruppen

Die meisten \LaTeX -Befehle haben eines der beiden folgenden Formate: Entweder sie beginnen mit einem Backslash (\backslash) und haben dann einen nur aus Buchstaben bestehenden Namen, der durch ein oder mehrere Leerzeichen oder durch ein nachfolgendes Sonderzeichen oder eine Ziffer beendet wird; oder sie bestehen aus einem Backslash und genau einem Sonderzeichen oder einer Ziffer. Groß- und Kleinbuchstaben haben auch in Befehlsnamen *verschiedene* Bedeutung. Wenn man nach einem Befehlsnamen eine Leerstelle erhalten will, muß man $\{ \}$ zur Beendigung des Befehlsnamens oder einen eigenen Befehl für die Leerstelle verwenden.

Heute ist der 35. Mai 1998. Oder:
Heute ist der 35. Mai 1998. Falsch
ist: Am 35. Mai 1998 regnet es.
Richtig: Am 35. Mai 1998 scheint
die Sonne. Oder: Am 35. Mai 1998
schneit es.

Heute ist der \backslash today.
Oder: Heute ist der \backslash today .
Falsch ist:
Am \backslash today regnet es.
Richtig:
Am \backslash today $\{ \}$ scheint die Sonne.
Oder: Am \backslash today \backslash schneit es.

Manche Befehle haben Parameter, die zwischen geschwungenen Klammern angegeben werden müssen. Manche Befehle haben Parameter, die weggelassen oder zwischen eckigen Klammern angegeben werden können. Manche Befehle haben Varianten, die durch das Hinzufügen eines Sterns an den Befehlsnamen unterschieden werden.

Geschwungene Klammern können auch dazu verwendet werden, Gruppen (groups) zu bilden. Die Wirkung von Befehlen, die innerhalb von Gruppen oder Umgebungen (environments) angegeben werden, endet immer mit dem Ende der Gruppe bzw. der Umgebung. Im obigen Beispiel ist $\{ \}$ eine leere Gruppe, die außer der Beendigung des Befehlsnamens `today` keine Wirkung hat.

2.3 Kommentare

Alles, was hinter einem Prozentzeichen (%) steht (bis zum Ende der Eingabezeile), wird von \LaTeX ignoriert. Dies kann für Notizen des Autors verwendet werden, die nicht oder noch nicht ausgedruckt werden sollen.

Das ist ein Beispiel.

Das ist ein % dummes
% Besser: ein lehrreiches <----
Beispiel.

2.4 Aufbau

Der erste Befehl in einem \LaTeX -Eingabefile muß der Befehl

```
\documentclass
```

sein. Er legt fest, welche Art von Schriftstück überhaupt erzeugt werden soll (Bericht, Buch, Brief usw.). Danach können weitere Befehle folgen, die für das gesamte Dokument gelten sollen. Dieser Teil des Dokuments wird auch als *Vorspann* oder *Präambel* bezeichnet. Mit dem Befehl

```
\begin{document}
```

endet der Vorspann, und es beginnt das Setzen des Schriftstücks. Nun folgen der Text und alle \LaTeX -Befehle, die das Ausdrucken des Schriftstücks bewirken. Die Eingabe muß mit dem Befehl

```
\end{document}
```

beendet werden. Falls nach diesem Befehl noch Eingaben folgen, werden sie von \LaTeX ignoriert.

Abbildung 1 zeigt ein *minimales* \LaTeX -File. Ein etwas komplizierteres File ist in Abbildung 2 skizziert.

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

Abbildung 1: Ein minimales \LaTeX -File

2.5 Dokumentklassen

Die am Beginn des Eingabefiles mit

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

definierte „Klasse“ eines Dokumentes enthält Vereinbarungen über das Layout und die logischen Strukturen, z. B. die Gliederungseinheiten (Kapitel etc.), die für alle Dokumente dieses Typs gemeinsam sind.

Zwischen den geschwungenen Klammern *muß* genau eine Dokumentklasse angegeben werden. Tabelle 1 auf S. 9 führt Klassen auf, die in jeder \LaTeX -Installation existieren sollten. Im *Local Guide* [7] können weitere verfügbare Klassen angegeben sein.

```

\documentclass[11pt,a4paper]{article}
\usepackage[latin1]{inputenc}
\usepackage{ngerman}
\date{29. Februar 1998}
\author{H.~Partl}
\title{Über kurz oder lang}

\begin{document}
\maketitle
\begin{abstract}
Beispiel für einen wissenschaftlichen Artikel
in deutscher Sprache.
\end{abstract}
\tableofcontents

\section{Start}

Hier beginnt mein schönes Werk ...

\section{Ende}

... und hier endet es.

\end{document}

```

Abbildung 2: Aufbau eines Artikels

Zwischen den eckigen Klammern *können*, durch Kommata getrennt, eine oder mehrere Optionen für das Klassenlayout angegeben werden. Die wichtigsten Optionen sind in der Tabelle 2 auf S. 10 angeführt. Das Eingabefile für diese Beschreibung beginnt z. B. mit:

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

2.6 Pakete

Mit dem Befehl

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

können im Vorspann ergänzende Makropakete (packages) geladen werden, die das Layout der Dokumentklasse modifizieren oder zusätzliche Funktionalität bereitstellen. Eine Auswahl von Paketen findet sich in der Tabelle 3 auf S. 11. Das Eingabefile für diese Beschreibung enthält beispielsweise:

```
\usepackage{german,latexsym,alltt,
             graphicx,textcomp,hyperref}
```

Tabelle 1: 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. (Nicht überall vorhanden, siehe <i>Local Guide</i> [7].)
letter	für Briefe (siehe auch Abschnitt 8.2)
foils	für Folien oder Präsentationen. (Nicht überall vorhanden, siehe <i>Local Guide</i> [7].)

2.7 Eingabezeichensatz

Bei jedem \LaTeX -System dürfen mindestens die folgenden Zeichen zur Eingabe von Text verwendet werden:

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

Die folgenden Eingabezeichen haben für \LaTeX eine Spezialbedeutung oder sind nur innerhalb von mathematischen Formeln erlaubt:

\$ & % # _ { } ~ ^ " \ | < >

Für Zeichen, die über obige Liste hinausgehen, beispielsweise die Umlaute, sind je nach Betriebssystem des verwendeten Computers unterschiedliche Kodierungen in Gebrauch. Damit auch diese Zeichen im Eingabefile benutzt werden dürfen, muß man das Paket `inputenc` laden und dabei die jeweilige Kodierung als Option angeben: `\usepackage[codepage]{inputenc}`. 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

Tabelle 2: Klassenoptionen (Alternativen sind durch | getrennt)

<code>10pt 11pt 12pt</code>	wählt die normale Schriftgröße des Dokuments aus. 10 pt hohe Schrift ist die Voreinstellung; diese Beschreibung benutzt 11 pt.
<code>a4paper</code>	für Papier im DIN A4-Format. Ohne diese Option nimmt \LaTeX amerikanisches Papierformat an.
<code>fleqn</code>	für linksbündige statt zentrierte mathematische Gleichungen
<code>leqno</code>	für Gleichungsnummern links statt rechts von jeder nummerierten Gleichung
<code>titlepage notitlepage</code>	legt fest, ob Titel und Zusammenfassung auf einer eigenen Seite erscheinen sollen. <code>titlepage</code> ist die Voreinstellung für die Klassen <code>report</code> und <code>book</code> .
<code>onecolumn twocolumn</code>	für ein- oder zweispaltigen Satz. Die Voreinstellung ist immer <code>onecolumn</code> . Die Klassen <code>letter</code> und <code>slides</code> kennen <i>keinen</i> zweispaltigen Satz.
<code>oneside twoside</code>	legt fest, ob die Seiten für ein- oder zweiseitigen Druck gestaltet werden sollen. <code>oneside</code> ist die Voreinstellung für alle Klassen außer <code>book</code> .

Falls \LaTeX ein eingegebenes Zeichen nicht darstellen kann, was meist für die sogenannten „Pseudografik-Zeichen“ gilt, bekommt man eine entsprechende Fehlermeldung. Auch sind manche Zeichen nur im Text, andere nur in mathematischen Formeln erlaubt.

Man beachte, daß der in der *Ausgabe* darstellbare Zeichenvorrat von \LaTeX nicht davon abhängt, welche Zeichen als *Eingabe* erlaubt sind: 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. Näheres darüber erfahren Sie in Abschnitt 3.4.

Tabelle 3: Pakete (eine Auswahl)

<code>alltt</code>	Definiert eine Variante der <code>verbatim</code> -Umgebung
<code>amsmath</code> , <code>amssymb</code>	Mathematischer Formelsatz mit erweiterten Fähigkeiten, zusätzliche mathematische Schriften und Symbole; Beschreibung siehe [5].
<code>babel</code>	Anpassungen für viele verschiedene Sprachen. Die gewählten Sprachen werden als Optionen angegeben.
<code>color</code>	Unterstützung für Farbausgabe; Beschreibung siehe [11] und [6].
<code>dcolum</code>	Unterstützt auf Dezimaltrennzeichen ausgerichtete Spalten in den Umgebungen <code>array</code> und <code>tabular</code>
<code>fontenc</code>	Erlaubt die Verwendung von Schriften mit unterschiedlicher Kodierung (Zeichenvorrat, Anordnung).
<code>fancyhdr</code>	Flexible Gestaltung von Kopf- und Fußzeilen.
<code>geometry</code>	Manipulation des Seitenlayouts.
<code>german</code> , <code>ngerman</code>	Anpassungen für die deutsche Sprache in traditioneller und neuer Rechtschreibung.
<code>graphicx</code>	Einbindung von extern erzeugten Graphiken. Die umfangreichen Möglichkeiten dieses Pakets werden in [11] und [6] beschrieben.
<code>hyperref</code>	Ermöglicht Hyperlinks zwischen Textstellen und zu externen Dokumenten; besonders sinnvoll einsetzbar, wenn mit \LaTeX eine Ausgabedatei im PDF- oder HTML-Format erzeugt wird.
<code>inputenc</code>	Deklaration der Zeichenkodierung im Eingabefile.
<code>latexsym</code>	Erlaubt einige besondere Symbole wie \square , die mit \LaTeX 2.09 standardmäßig verfügbar waren.
<code>longtable</code>	für Tabellen über mehrere Seiten mit automatischem Seitenumbruch.
<code>makeidx</code>	Unterstützt das Erstellen eines Index.
<code>multicol</code>	Mehrspaltiger Satz mit Kolumnenausgleich.
<code>textcomp</code>	Bindet Schriften mit zusätzlichen Textsymbolen ein.

3 Setzen von Text

3.1 Deutschsprachige Texte

L^AT_EX wurde ursprünglich für den englischen Sprachraum entwickelt. Für Texte, die in einer anderen Sprache als (amerikanischem) Englisch verfaßt sind, muß deshalb ein zusätzliches Paket (siehe Abschnitt 2.6) zur Sprachanpassung geladen werden. Für deutschsprachige Texte ist das normalerweise das Paket `german` oder `ngerman`:

```
\usepackage{german}
```

oder

```
\usepackage{ngerman}
```

Für Texte in *traditioneller* Rechtschreibung ist `german` zu benutzen, für Texte in der *neuen* deutschen Rechtschreibung `ngerman`. Der Grund für diese Unterscheidung ist die unterschiedliche Silbentrennung. Eine ausführliche Beschreibung dieser Pakete findet man in [20]. Wenn im folgenden vom Paket `german` die Rede ist, so bezieht sich das normalerweise auch auf `ngerman`.

3.2 Zeilen- und Seiten-Umbruch

3.2.1 Blocksatz

Normaler Text wird im Blocksatz, d. h. mit Randausgleich gesetzt. L^AT_EX führt den Zeilen- und Seitenumbruch automatisch durch. Dabei wird für jeden Absatz die bestmögliche Aufteilung der Wörter auf die Zeilen bestimmt, und wenn notwendig werden Wörter automatisch abgeteilt.

Das Ende von Wörtern und Sätzen wird durch Leerzeichen gekennzeichnet. Hierbei spielt es keine Rolle, ob man ein oder 100 Leerzeichen eingibt.

Eine oder mehrere Leerzeilen kennzeichnen das Ende von Absätzen.

Das Ende von Wörtern und Sätzen wird durch Leerzeichen gekennzeichnet.
Hierbei spielt es keine Rolle,
ob man ein oder 100
Leerzeichen eingibt.

Eine oder mehrere Leerzeilen
kennzeichnen das Ende von
Absätzen.

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.

Mit Hilfe der in Abschnitt 3.9 beschriebenen Umgebungen ist es möglich, spezielle Textteile jeweils anders zu setzen.

Für Ausnahmefälle kann man den Umbruch außerdem mit den folgenden Befehlen 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).

Mit dem \LaTeX -Befehl `\enlargethispage{Länge}` läßt sich eine gegebene Seite um einen festen Betrag verlängern oder verkürzen. Damit ist es möglich, noch eine Zeile mehr auf eine Seite zu bekommen. (Zur Schreibweise von Längenangaben siehe Abschnitt 8.1.2.)

\LaTeX bemüht sich, den Zeilenumbruch besonders schön zu machen. Falls es keine den strengen Regeln genügende Möglichkeit für einen glatten rechten Rand findet, läßt es eine Zeile zu lang und gibt eine entsprechende Fehlermeldung aus (`overfull hbox`). Das tritt insbesondere dann auf, wenn keine geeignete Stelle für die Silbentrennung gefunden wird. Innerhalb der `sloppypar`-Umgebung ist \LaTeX generell weniger streng in seinen Ansprüchen und vermeidet solche überlange Zeilen, indem es die Wortabstände stärker – notfalls auch unschön – vergrößert. In diesem Fall werden zwar Warnungen gemeldet (`underfull hbox`), das Ergebnis ist aber meistens durchaus brauchbar.

3.2.2 Silbentrennung

Falls die automatische Silbentrennung in einzelnen Fällen nicht das richtige Ergebnis liefert, kann man diese Ausnahmen mit den folgenden Befehlen richtigstellen. Das kann insbesondere bei zusammengesetzten oder fremdsprachigen Wörtern notwendig werden; außerdem findet \LaTeX in Wörtern mit Umlauten oder akzentuierten Buchstaben nicht alle zulässigen Trennstellen.

Der Befehl `\hyphenation` bewirkt, daß die darin angeführten Wörter jedesmal an den und nur an den mit – markierten Stellen abgeteilt werden können. Er sollte im Vorspann stehen und eignet sich *nur* für Wörter, die keine Umlaute, scharfes s, Ziffern oder sonstige Sonderzeichen enthalten.

```
\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.

Eingabefile, \LaTeX -Eingabe-
file, Häßlichkeit

```
Ein\ -gabe\ -file,
\LaTeX-Eingabe\ -file,
Häß\ -lich\ -keit
```

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

Die Telefonnummer ist nicht mehr
(02 22) 56 01-36 94.
filename gibt den Dateinamen an.

```
Die Telefonnummer ist nicht mehr
\mbox{(02\,22) 56\,01-36\,94}. \\\
\mbox{\textit{filename}} gibt den
Dateinamen an.
```

Innerhalb des von `\mbox` eingeschlossenen Text können Wortabstände für den notwendigen Randausgleich bei Blocksatz nicht mehr verändert werden. Ist dies nicht erwünscht, sollte man besser einzelne Wörter oder Wortteile in `\mbox` einschließen und diese mit einer Tilde `~`, einem untrennbaren Wortzwischenraum (siehe Abschnitt 3.3), verbinden.

Das Paket `german` macht noch einige weitere Befehle verfügbar, die bestimmte Besonderheiten der deutschen Sprache berücksichtigen. Die wichtigsten von ihnen sind: `"ck` für „ck“, das als „k-k“ abgeteilt wird, `"ff` für „ff“, das als „ff-f“ abgeteilt wird (und ebenso für andere Konsonanten) und `"~` für einen Bindestrich, an dem kein Zeilenumbruch stattfinden soll.

Drucker bzw. Druk-ker	Dru"cker
Rolladen bzw. Roll-laden	Ro"lladen
x-beliebig	x"~beliebig
bergauf und -ab	bergauf und "~ab

3.3 Wortabstand

Um einen glatten rechten Rand zu erreichen, variiert \LaTeX die Leerstellen zwischen den Wörtern etwas. Nach Punkten, Fragezeichen u. a., die einen Satz beenden, wird dabei ein etwas größerer Abstand erzeugt, was die Lesbarkeit des Textes erhöht. \LaTeX nimmt an, daß Punkte, die auf einen Großbuchstaben folgen, eine Abkürzung bedeuten, und daß alle anderen Punkte einen Satz beenden. Ausnahmen von diesen Regeln muß man \LaTeX mit den folgenden Befehlen mitteilen:

Ein Backslash (`\`) vor einem Leerzeichen bedeutet, daß diese Leerstelle nicht verbreitert werden darf.

Eine `~` (Tilde) bedeutet eine Leerstelle, an der kein Zeilenwechsel erfolgen darf.

Mit `\,` läßt sich ein kurzer Abstand erzeugen, wie er z. B. in Abkürzungen vorkommt oder zwischen Zahlenwert und Maßeinheit.

Der Befehl `\@` vor einem Punkt bedeutet, daß dieser Punkt einen Satz beendet, obwohl davor ein Großbuchstabe steht.

Das betrifft u. a. auch die	Das betrifft u.\,a.\ auch die \\\
wissenschaftl. Mitarbeiter.	wissenschaftl.\ Mitarbeiter. \\\
Dr. Partl wohnt im 1. Stock.	Dr.~Partl wohnt im 1.~Stock. \\\
... 5 cm breit.	\dots\ 5\,cm breit. \\\
Ich brauche Vitamin C. Du nicht?	Ich brauche Vitamin~C\@.
	Du nicht?

Außerdem gibt es die Möglichkeit, mit dem Befehl `\frenchspacing` zu vereinbaren, daß die Abstände an Satzenden nicht anders behandelt werden sollen als die zwischen Wörtern. Diese Konvention ist im nicht-englischen Sprachraum verbreitet. In diesem Fall brauchen die Befehle `\,` und `\@` nicht angegeben werden. Mit dem Paket `german` ist `\frenchspacing` automatisch gewählt; das kann durch `\nonfrenchspacing` wieder rückgängig gemacht werden – so wie durchgängig im vorliegenden Dokument!

3.4 Spezielle Zeichen

3.4.1 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 don’t know!”	‘‘No,’’ he said,
	‘‘I don’t know!’’

„Deutsche Gänsefüßchen“ sehen anders aus als “amerikanische Quotes”. Bei Benutzung des Paketes `german` (siehe 3.1) stehen die folgenden Befehle für deutsche Anführungszeichen zur Verfügung: `"` (Doublequote und Grave-Akzent) für Anführungszeichen unten, und `"` (Doublequote und Apostroph) für Anführungszeichen oben.

„Nein,“ sagte er, „ich weiß nichts!“	"Nein," sagte er,
	"ich weiß nichts!"

In den Zeichensätzen mancher Rechner (z.B. Macintosh) sind die deutschen Anführungszeichen enthalten. Das Paket `inputenc` (siehe Abschnitt 2.7) erlaubt dann, sie auch direkt einzugeben.

3.4.2 Binde- und Gedankenstriche

Im Satz werden unterschiedliche Striche für Bindestrich, Gedankenstrich und Minus-Zeichen verwendet. Die verschieden langen Striche werden in `LATEX` durch Kombinationen von Minus-Zeichen angegeben. Der ganz lange Gedankenstrich (—) wird im Deutschen nicht benutzt, im Englischen wird er ohne Leerzeichen eingefügt.

O-Beine	O-Beine
10–18 Uhr	10--18~Uhr
Paris–Dakar	Paris--Dakar
Schalke 04 – Hertha BSC	Schalke 04 -- Hertha BSC
ja – oder nein?	ja -- oder nein?
yes—or no?	yes---or no?
0, 1 und –1	0, 1 und \$-1\$

3.4.3 Punkte

Im Gegensatz zur Schreibmaschine, wo jeder Punkt und jedes Komma mit einem der Buchstabenbreite entsprechenden Abstand versehen ist, werden Punkte und Kommata im Buchdruck eng an das vorangehende Zeichen gesetzt. Für Fortsetzungspunkte (drei Punkte mit geeignetem Abstand) gibt es daher einen eigenen Befehl `\ldots` oder `\dots`.

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

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

3.4.4 Ligaturen

Im Buchdruck ist es üblich, manche Buchstabenkombinationen anders zu setzen als die Einzelbuchstaben.

fi fl AV Te ... statt fi fl AV Te ...

Mit Rücksicht auf die Lesbarkeit des Textes sollten diese Ligaturen und Unterscheidungen (kerning) unterdrückt werden, wenn die betreffenden Buchstabenkombinationen nach Vorsilben oder bei zusammengesetzten Wörtern zwischen den Wortteilen auftreten. Dazu dient der Befehl `\/`.

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

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

Mit dem Paket **german** steht zusätzlich der Befehl `"|` zur Verfügung, der gleichzeitig eine Trennhilfe darstellt.

Auflage (Auf-lage)

Auf"|lage (Auf-lage)

3.4.5 Symbole, Akzente und besondere Buchstaben

Einige der Zeichen, die bei der Eingabe eine Spezialbedeutung haben, können durch das Voranstellen des Zeichens `\` (Backslash) ausgedruckt werden:

`$ & % # - { }`

`\$ \& \% \# _ \{ \}`

Für andere gibt es besondere Befehle. Sie gelten nur für normalen Text; wie derartige Symbole innerhalb von mathematischen Formeln gesetzt werden, erfahren Sie im Kapitel 4:

<code>~</code>	<code>\textasciitilde</code>
<code>^</code>	<code>\textasciicircum</code>
<code>\</code>	<code>\textbackslash</code>
<code> </code>	<code>\textbar</code>
<code><</code>	<code>\textless</code>
<code>></code>	<code>\textgreater</code>

\LaTeX ermöglicht darüber hinaus die Verwendung von Akzenten und speziellen Buchstaben aus zahlreichen verschiedenen Sprachen, siehe die Tabellen 4 und 5. Akzente werden darin jeweils am Beispiel des Buchstabens o gezeigt, können aber prinzipiell auf jeden Buchstaben gesetzt werden. Wenn ein Akzent auf ein i oder j gesetzt werden soll, muß der i-Punkt wegbleiben. Dies erreicht man mit den Befehlen `\i` und `\j`. Es steht auch ein Befehl `\textcircled` für eingekreiste Zeichen zur Verfügung.

Tabelle 4: Akzente und spezielle Buchstaben

ò	\‘o	ó	\’o	ô	\^o
õ	\~o	ō	\=o	ô	\.o
ö	\u o	ö	\v o	ő	\H o
ö	\"o	q	\c o	q	\d o
o	\b o	ö	\r o	öo	\t oo
œ	\oe	Œ	\OE	æ	\ae
Æ	\AE	å	\aa	Å	\AA
ø	\o	Ø	\O	l	\l
L	\L	ı	\i	J	\j
ß	\ss				

Tabelle 5: Symbole

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

Hôtel, naïve, smørebrød.

H\^otel, na\"i ve,
sm\o rebr\o d.

Die häßliche Straße.

Die h\"a\ss{}liche
Stra\ss{}e.

¡Señorita!

!‘Se\~norita!

ⓧ

\textcircled{x}

Benutzt man das Paket `inputenc` mit der passenden Option für das jeweilige Betriebssystem, siehe Abschnitt 2.7, dann darf man diese Zeichen – soweit sie im Zeichensatz des Betriebssystems existieren – auch direkt in das Eingabefile schreiben.

Mit dem Paket `german` können Umlaute auch durch einfaches Voranstellen eines Doublequotes geschrieben werden, also z. B. "o für „ö“; für scharfes s darf man "s schreiben (ohne Probleme mit nachfolgenden Leerstellen):

Die häßliche Straße muß schöner
werden.

Die h"a"sliche Stra"se
mu"s sch"oner werden.

Diese Notation wurde eingeführt, als die direkte Eingabe und Anzeige von Umlauten auf vielen Rechnersystemen noch nicht möglich war. Als Quasi-Standard zum plattformübergreifenden Austausch von T_EX- und L^AT_EX-Dokumenten ist sie aber nach wie vor nützlich und für deutschsprachige Texte weit verbreitet; sie wird auch in einigen Beispielen dieser Kurzbeschreibung benutzt.

3.5 Kapitel und Überschriften

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 \LaTeX automatisch bestimmt.

Die Überschrift des gesamten Artikels bzw. die Titelseite des Schriftstücks wird mit dem Befehl `\maketitle` gesetzt. Der Inhalt muß vorher mit den Befehlen `\title`, `\author` und `\date` vereinbart werden (vgl. Abbildung 2 auf Seite 8).

Der Befehl `\tableofcontents` bewirkt, daß ein Inhaltsverzeichnis ausgedruckt wird. \LaTeX nimmt dafür immer die Überschriften und Seitennummern von der jeweils letzten vorherigen Verarbeitung des Eingabefiles. Bei einem neu erstellten oder um neue Kapitel erweiterten Schriftstück muß man das Programms \LaTeX also mindestens zweimal aufrufen, damit man die richtigen Angaben erhält.

Es gibt auch Befehle der Form `\section*{...}`, bei denen keine Numerierung und keine Eintragung ins Inhaltsverzeichnis erfolgen.

Mit den Befehlen `\label` und `\ref` ist es möglich, die von \LaTeX automatisch vergebenen Kapitelnummern im Text anzusprechen. Für `\ref{...}` setzt \LaTeX die mit `\label{...}` definierte Nummer ein. Auch hier wird immer die Nummer von der letzten vorherigen Verarbeitung des Eingabefiles genommen. Beispiel:

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

3.6 Fußnoten

Fußnoten¹ werden automatisch numeriert und am unteren Ende der Seite ausgedruckt. Innerhalb von Gleitobjekten (siehe Abschnitt 6.2), Tabellen (3.9.6) oder der `tabbing`-Umgebung (3.9.5) ist der Befehl `\footnote` nicht erlaubt.

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

¹Das ist eine Fußnote.

3.7 Hervorgehobener Text

In maschinengeschriebenen Texten werden hervorzuhebende Texte unterstrichen, im Buchdruck wird stattdessen ein auffälliger Schriftschnitt verwendet. Der Befehl

```
\emph{text}
```

(emphasize) setzt seinen Parameter in einem auffälligen Stil. \LaTeX verwendet für den hervorgehobenen Text *kursive* Schrift.

Werden innerhalb eines hervorgehobenen Textes nochmals Passagen hervorgehoben, so setzt \LaTeX diese in einer aufrechten Schrift.

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

3.8 Hochgestellter Text

Hochgestellten Text in passender Größe generiert folgender Befehl:

```
\textsuperscript{text}
```

le 2^{ième} régime

```
le 2\textsuperscript{i\‘eme}
r\’egime
```

3.9 Umgebungen

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

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

Umgebungen sind *Gruppen*. Sie können auch ineinander geschachtelt werden, dabei muß aber die richtige Reihenfolge beachtet werden:

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

3.9.1 Zitate (quote, quotation, verse)

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

Eine typographische Faustregel für die Zeilenlänge lautet:

Keine Zeile soll mehr als
ca. 66 Buchstaben ent-
halten.

Deswegen werden in Zeitungen meh-
rere Spalten nebeneinander verwen-
det.

Eine typographische Faustregel
für die Zeilenlänge lautet:

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

Deswegen werden in Zeitungen
mehrere Spalten nebeneinander
verwendet.

Die **quotation**-Umgebung unterscheidet sich in den Standardklassen (vgl. Tabelle 1 auf Seite 9) von der **quote**-Umgebung dadurch, daß Absätze durch Einzüge gekennzeichnet werden. Sie ist daher für längere Zitate, die aus mehreren Absätzen bestehen, geeignet.

Die **verse**-Umgebung eignet sich für Gedichte und für Beispiele, bei denen die Zeilenaufteilung wesentlich ist. Die Verse (Zeilen) werden durch `\\` getrennt, Strophen durch Leerzeilen.

3.9.2 Listen (**itemize**, **enumerate**, **description**)

Die Umgebung **itemize** eignet sich für einfache Listen (siehe Abbildung 3). Die Umgebung **enumerate** eignet sich für numerierte Aufzählungen (siehe Abbildung 4). Die Umgebung **description** eignet sich für Beschreibungen (siehe Abbildung 5).

Listen:

- Bei **itemize** werden die Elemente durch Punkte und andere Symbole gekennzeichnet.
- Listen kann man auch verschachteln:
 - Die maximale Schachtelungstiefe ist 4.
 - Bezeichnung und Einrückung der Elemente wechseln automatisch.
- usw.

Listen:

```
\begin{itemize}

\item Bei \texttt{itemize}
werden die Elemente ...

\item Listen kann man auch
verschachteln:
  \begin{itemize}
    \item Die maximale ...
    \item Bezeichnung und ...
  \end{itemize}

\item usw.

\end{itemize}
```

Abbildung 3: Beispiel für **itemize**

Numerierte Listen:	Numerierte Listen:
1. Bei <code>enumerate</code> werden die Elemente mit Ziffern oder Buchstaben numeriert.	<code>\begin{enumerate}</code>
2. Die Numerierung erfolgt automatisch.	<code>\item Bei \texttt{enumerate} werden die Elemente ...</code>
3. Listen kann man auch verschachteln:	<code>\item Die Numerierung ...</code>
(a) Die maximale Schachteltiefe ist 4.	<code>\item Listen kann man auch verschachteln:</code>
(b) Bezeichnung und Einrückung der Elemente wechseln 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 4: Beispiel für `enumerate`

Kleine Tierkunde:	Kleine Tierkunde:
Gelse: ein kleines Tier, das östlich des Semmering Touristen verjagt.	<code>\begin{description}</code>
Gemse: ein großes Tier, das westlich des Semmering von Touristen verjagt wird.	<code>\item[Gelse:]</code>
Gürteltier: ein mittelgroßes Tier, das hier nur wegen der Länge seines Namens vorkommt.	ein kleines Tier, das ...
	<code>\item[Gemse:]</code>
	ein großes Tier, das ...
	<code>\item[Gürteltier:]</code>
	ein mittelgroßes Tier, das ...
	<code>\end{description}</code>

Abbildung 5: Beispiel für `description`

3.9.3 Linksbündig, rechtsbündig, zentriert (`flushleft`, `flushright`, `center`)

Die Umgebungen `flushleft` und `flushright` bewirken links- bzw. rechtsbündigen Satz ohne Randausgleich („Flattersatz“) und ohne Trennungen, `center` setzt den Text in die Mitte der Zeile. Die einzelnen Zeilen werden durch `\\` getrennt. Wenn man `\\` nicht angibt, bestimmt \LaTeX automatisch die Zeilenaufteilung.

links	<code>\begin{flushleft}</code>
Backbord	<code>links \\</code>
	<code>Backbord</code>
	<code>\end{flushleft}</code>
rechts	<code>\begin{flushright}</code>
Steuerbord	<code>rechts \\</code>
	<code>Steuerbord</code>
	<code>\end{flushright}</code>
Im	<code>\begin{center}</code>
Reich	<code>Im \\ Reich \\ der \\ Mitte</code>
der	<code>\end{center}</code>
Mitte	

3.9.4 Direkte Ausgabe (`verbatim`, `verb`)

Zwischen `\begin{verbatim}` und `\end{verbatim}` stehende Zeilen werden genauso ausgedruckt, wie sie eingegeben wurden, d. h. mit allen Leerzeichen und Zeilenwechseln und ohne Interpretation von Spezialzeichen und \LaTeX -Befehlen. Dies eignet sich z. B. für das Ausdrucken eines (kurzen) Computer-Programms.

Innerhalb eines Absatzes können einzelne Zeichenkombinationen oder kurze Textstücke ebenso „wörtlich“ ausgedruckt werden, indem man sie zwischen `\verb|` und `|` einschließt. Mit diesen Befehlen wurden z. B. alle \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.

3.9.5 Tabulatoren (`tabbing`)

In der `tabbing`-Umgebung kann man Tabulatoren ähnlich wie an Schreibmaschinen setzen und verwenden. 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 \> Mittelteil \> rechts\\</code>
		Teddybär	<code>Es \\</code>
			<code>war einmal \> und ist</code>
			<code>\> nicht mehr\\</code>
			<code>ein \> \> ausgestopfter\\</code>
			<code>\> \> Teddyb"ar</code>
			<code>\end{tabbing}</code>

3.9.6 Tabellen (tabular)

Die `tabular`-Umgebung dient zum Setzen von Tabellen, bei denen \LaTeX automatisch die benötigte Spaltenbreite bestimmt, und bei der auch spezielle Eigenschaften wie Rechtsbündigkeit und Hilfslinien vereinbart werden können.

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	hexadezimal
3700	oktal
11111000000	binär
1984	dezimal

```

\begin{tabular}{|rl|}
\hline
7C0 & hexadezimal \\
3700 & oktal \\
11111000000 & bin"ar \\
\hline\hline
1984 & dezimal \\
\hline
\end{tabular}

```


4 Setzen von mathematischen Formeln

4.1 Allgemeines

Mathematische Textteile innerhalb eines Absatzes werden zwischen `\(` und `\)` oder zwischen `$` und `$` oder zwischen `\begin{math}` und `\end{math}` eingeschlossen. Als mathematische Texte gelten sowohl komplette mathematische Formeln als auch einzelne Variablennamen, die sich auf Formeln beziehen, griechische Buchstaben und diverse Sonderzeichen.

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.

`\TeX` spricht man wie

Mit ♡-lichen Grüßen

`\(\tau\epsilon\chi\)` aus.
Mit `\(\heartsuit\)`-lichen

`\(\heartsuit\)`-lichen
Grüßen

Größere mathematische Formeln oder Gleichungen setzt man besser in eigene Zeilen. Wenn sie *keine* Gleichungsnummer erhalten sollen, stellt man sie dazu zwischen `\begin{displaymath}` und `\end{displaymath}` oder zwischen `\[` und `\]`; wenn sie eine Gleichungsnummer erhalten sollen, stellt man sie zwischen `\begin{equation}` und `\end{equation}`.

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).

Mit `\label` und `\ref` kann man die Gleichungsnummern im Text ansprechen.

$$\varepsilon > 0 \quad (2)$$

Aus (2) folgt ...

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

Aus `(\ref{eps})` folgt ...

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

1. Leerzeilen sind verboten (Mathematische Formeln müssen innerhalb eines Absatzes stehen).

2. 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.

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

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

3. 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.

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

```
\begin{equation}
x^2 \geq 0 \quad \text{für alle } x \in \mathbf{R}
\end{equation}
```

4.2 Elemente in mathematischen Formeln

In diesem Abschnitt werden die wichtigsten Elemente, die in mathematischen Formeln verwendet werden, kurz beschrieben. Eine Liste aller verfügbaren Symbole enthält Kapitel 4.5.

Kleine **griechische Buchstaben** werden als `\alpha`, `\beta`, `\gamma`, `\delta`, usw. eingegeben, große griechische Buchstaben als `\mathrm{A}`, `\mathrm{B}`, `\Gamma`, `\Delta`, usw.

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

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

Weiters gibt es eine Fülle von **mathematischen Symbolen**: von \in über \Rightarrow bis ∞ (siehe Kapitel 4.5).

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

ABCabc	<code>\(\mathrm{ABCabc}\)</code>
ABCabc	<code>\(\mathbf{ABCabc}\)</code>
ABCabc	<code>\(\mathsf{ABCabc}\)</code>
ABCabc	<code>\(\mathtt{ABCabc}\)</code>
<i>ABC</i>	<code>\(\mathcal{ABC}\)</code>

Die kalligraphischen Buchstaben (`\mathcal`) gibt es nur als Großbuchstaben. Mit dem Paket `amsymb` [5] stehen auch Alphabete für Mengenzeichen und Frakturschrift zur Verfügung. Lokal können noch weitere installiert sein – siehe *Local Guide* [7].

Exponenten und Indizes können mit den Zeichen `^` und `_` hoch- bzw. tiefgestellt werden.

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

Das **Wurzelzeichen** wird mit `\sqrt`, n -te Wurzeln werden mit `\sqrt[n]` eingegeben. Die Größe des Wurzelzeichens wird von L^AT_EX automatisch gewählt.

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

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

$\overline{m+n}$	<code>\(\overline{m+n}\)</code>
------------------	---------------------------------

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

$\underbrace{a+b+\cdots+z}_{26}$	<code>\(\underbrace{a+b+\cdots+z}_{26}\)</code>
----------------------------------	---

Um mathematische **Akzente** wie Pfeile oder Schlangen auf Variablen zu setzen, gibt es die in Tabelle 6 auf Seite 30 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.

	<code>\begin{displaymath}</code>
$y = x^2$	<code>y=x^{2}</code> \quad
$y' = 2x$	<code>y'=2x</code> \quad
$y'' = 2$	<code>y''=2</code>
	<code>\end{displaymath}</code>

Mathematische **Funktionen** werden in der Literatur üblicherweise nicht kursiv (wie die Namen von Variablen), sondern in „normaler“ Schrift dargestellt. L^AT_EX stellt die folgenden Befehle für mathematische Funktionen zur Verfügung:

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

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.

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

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

Binomial-Koeffizienten können in der Form `{...}\choose{...}` gesetzt werden. Mit dem Befehl `\atop` erhält man das Gleiche ohne Klammern.

$$\binom{n}{k} \quad x \atop y+2$$

```
\begin{displaymath}
\{ n \choose k \} \quad x \atop y+2
\end{displaymath}
```

Das **Integralzeichen** wird mit `\int` eingegeben, das **Summenzeichen** mit `\sum`. Die obere und untere Grenze wird mit `^` bzw. `_` wie beim Hoch-/Tiefstellen angegeben.

Normalerweise werden die Grenzen neben das Integralzeichen gesetzt (um Platz zu sparen), durch Einfügen des Befehls `\limits` wird erreicht, daß die Grenzen oberhalb und unterhalb des Integralzeichens gesetzt werden.

Beim Summenzeichen hingegen werden die Grenzen bei der Angabe von `\nolimits` oder im laufenden Text neben das Summenzeichen gesetzt, ansonsten aber unter- und oberhalb.

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

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


4.4 Übereinander Setzen

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}
```

Für **mehrzeilige** Formeln oder Gleichungssysteme verwendet man die Umgebungen `eqnarray` und `eqnarray*` statt `equation`. Bei `eqnarray` erhält jede Zeile eine eigene Gleichungsnummer, bei `eqnarray*` wird ebenso wie bei `displaymath` *keine* Nummer hinzugefügt. Für Gleichungssysteme, die *eine* gemeinsame Nummer erhalten sollen, kann man eine `array`-Umgebung innerhalb der `equation`-Umgebung verwenden.

Die Umgebungen `eqnarray` und `eqnarray*` funktionieren wie eine 3-spaltige Tabelle der Form `{rcl}`, wobei die mittlere Spalte für das Gleichheits- oder Ungleichheitszeichen verwendet wird, nach dem die Zeilen ausgerichtet werden sollen. Der Befehl `\\` trennt die Zeilen.

$$\begin{array}{lcl} f(x) & = & \cos x \quad (5) \\ f'(x) & = & -\sin x \quad (6) \\ \int_0^x f(y) dy & = & \sin x \quad (7) \end{array}$$

```
\begin{eqnarray}
f(x) & = & \cos x \\
f'(x) & = & -\sin x \\
\int_0^x f(y) dy & = & \sin x
\end{eqnarray}
```

Zu lange Gleichungen werden von L^AT_EX *nicht* automatisch abgeteilt. Der Autor muß bestimmen, an welcher Stelle abgeteilt und wie weit eingerückt werden soll. Meistens verwendet man dafür eine der beiden folgenden Varianten:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (8)$$

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

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (9)$$

```

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

```

Der Befehl `\nonumber` bewirkt, daß an diese Stelle keine Gleichungsnummer gesetzt wird. Der Befehl `\lefteqn` ermöglicht Ausnahmen von der Spaltenaufteilung innerhalb `eqnarray`. Genauere Informationen enthält das *L^AT_EX-Handbuch* [1].

4.5 Liste der mathematischen Symbole

In den folgenden Tabellen sind alle Symbole angeführt, die standardmäßig im mathematischen Modus verwendet werden können. Die mit * versehenen Symbole werden in L^AT_EX 2_ε nur durch das Paket `latexsym` bereitgestellt. Bei vielen Installationen stehen mit den Paketen `amssymb`, `mathrsfs` oder `wasysym` weitere Zeichen zur Verfügung, näheres steht im *Local Guide* [7].

Tabelle 6: 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>		

Tabelle 7: Kleine griechische Buchstaben

α	<code>\alpha</code>	ι	<code>\iota</code>	ϱ	<code>\varrho</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	σ	<code>\sigma</code>
γ	<code>\gamma</code>	λ	<code>\lambda</code>	ς	<code>\varsigma</code>
δ	<code>\delta</code>	μ	<code>\mu</code>	τ	<code>\tau</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	υ	<code>\upsilon</code>
ε	<code>\varepsilon</code>	ξ	<code>\xi</code>	ϕ	<code>\phi</code>
ζ	<code>\zeta</code>	\omicron	<code>\omicron</code>	φ	<code>\varphi</code>
η	<code>\eta</code>	π	<code>\pi</code>	χ	<code>\chi</code>
θ	<code>\theta</code>	ϖ	<code>\varpi</code>	ψ	<code>\psi</code>
ϑ	<code>\vartheta</code>	ρ	<code>\rho</code>	ω	<code>\omega</code>

Tabelle 8: Große griechische Buchstaben

Γ	<code>\Gamma</code>	Ξ	<code>\Xi</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	Π	<code>\Pi</code>	Ψ	<code>\Psi</code>
Θ	<code>\Theta</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>
Λ	<code>\Lambda</code>	Υ	<code>\Upsilon</code>		

Tabelle 9: Verschiedene sonstige Symbole (* benötigt Paket `latexsym`)

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

Tabelle 10: „Große“ Operatoren

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

Tabelle 11: Binäre Operatoren (* benötigt Paket `latexsym`)

$+$	<code>\+</code>	$-$	<code>\-</code>	\div	<code>\div</code>
\pm	<code>\pm</code>	\cap	<code>\cap</code>	\vee	<code>\vee</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\wedge	<code>\wedge</code>
\setminus	<code>\setminus</code>	\oplus	<code>\oplus</code>	\oplus	<code>\oplus</code>
\cdot	<code>\cdot</code>	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\sqcup	<code>\sqcup</code>	\otimes	<code>\otimes</code>
$*$	<code>\ast</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
\star	<code>\star</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\diamond	<code>\diamond</code>	\triangleleft^*	<code>\lhd^*</code>	\dagger	<code>\dagger</code>
\circ	<code>\circ</code>	\triangleright^*	<code>\rhd^*</code>	\ddagger	<code>\ddagger</code>
\bullet	<code>\bullet</code>	\triangleleft^*	<code>\unlhd^*</code>	\amalg	<code>\amalg</code>
\bigcirc	<code>\bigcirc</code>	\triangleright^*	<code>\unrhd^*</code>	\wr	<code>\wr</code>
\triangleup	<code>\bigtriangleup</code>	∇	<code>\bigtriangledown</code>		

Tabelle 12: Relationen (* benötigt Paket `latexsym`)

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\sqsubset^*	<code>\sqsubset^*</code>	\sqsupset^*	<code>\sqsupset^*</code>	\Join^*	<code>\Join^*</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\notin	<code>\notin</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\smile	<code>\smile</code>	\mid	<code>\mid</code>	\doteq	<code>\doteq</code>
\frown	<code>\frown</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
$:$	<code>:</code>	\propto	<code>\propto</code>		

Tabelle 13: Negierte Relationen

\nless	<code>\not<</code>	\ngtr	<code>\not></code>	\neq	<code>\not=</code>
\nleq	<code>\not\leq</code>	\ngeq	<code>\not\geq</code>	\nequiv	<code>\not\equiv</code>
\nprec	<code>\not\prec</code>	\nsucc	<code>\not\succ</code>	\nsim	<code>\not\sim</code>
\npreceq	<code>\not\preceq</code>	\nsucceq	<code>\not\succeq</code>	\nsimeq	<code>\not\simeq</code>
\nsubset	<code>\not\subset</code>	\nsupset	<code>\not\supset</code>	\napprox	<code>\not\approx</code>
\nsubseteq	<code>\not\subseteq</code>	\nsupseteq	<code>\not\supseteq</code>	\ncong	<code>\not\cong</code>
\nsubsetneq	<code>\not\subsetneq</code>	\nsupsetneq	<code>\not\supsetneq</code>	\nasymp	<code>\not\asymp</code>

Tabelle 14: Pfeile (Vertikale Pfeile werden als Klammerungssymbole behandelt, alle anderen als Relationen. * benötigt Paket `latexsym`.)

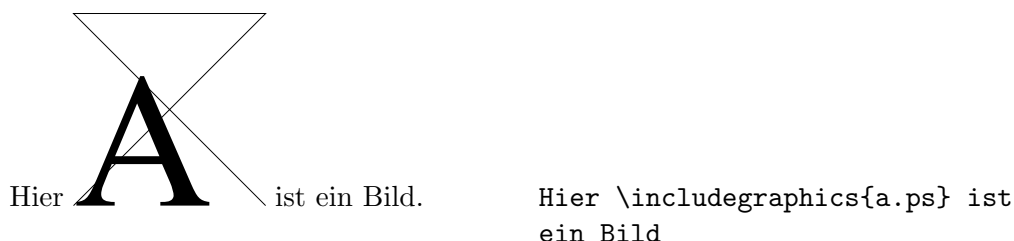
\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Lleftrightarrow	<code>\Lleftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightarrow	<code>\rightarrow</code>	\nwarrow	<code>\nwarrow</code>
\rightrightarrows	<code>\rightrightarrows</code>	\rightsquigarrow	<code>\rightsquigarrow</code>	\leadsto	<code>\leadsto</code> *

Tabelle 15: Klammern

$($	<code>(</code>	$)$	<code>)</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>
$[$	<code>[</code>	$]$	<code>]</code>	$\{$	<code>\{</code>	$\}$	<code>\}</code>
$ $	<code> </code>	$\ $	<code>\ </code>	\backslash	<code>\backslash</code>		

5 Setzen von Bildern

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 T_EX-System ab und muß in dessen Dokumentation beschrieben sein.



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, was die Bearbeitung beschleunigt und für Probeausdrucke nützlich ist.

Weitere Informationen zum Einbinden von Bildern finden Sie in der Online-Dokumentation [11], im *Graphics Companion* [6] und in K. Reckdahls empfehlenswertem Tutorium [16].

6 Seitenaufbau

6.1 Kopf- und Fußzeilen

Der Inhalt von Kopf- und Fußzeilen kann mit dem Befehl

```
\pagestyle{style}
```

festgelegt werden:

Mit `\pagestyle{plain}` steht die Seitennummer zentriert in der Fußzeile; das ist die Voreinstellung und braucht normalerweise nicht explizit angegeben zu werden. Mit dem Stil **headings** stehen Kapitel-Überschrift und Seitennummer in der Kopfzeile. Mit **empty** sind Kopf- und Fußzeile leer. Der Befehl

```
\thispagestyle{style}
```

gilt entsprechend nur für die aktuelle Seite. Einige Befehle, wie etwa `\chapter`, ändern den Stil der aktuellen Seite. Diese Änderungen kann man durch einen nachfolgenden `\thispagestyle`-Befehl aufheben.

Im *L^AT_EX-Handbuch* [1] ist angegeben, wie man das Aussehen der Kopf- und Fußzeilen außerdem mit dem Seitenstil **myheadings** und den Befehlen `\markboth`, `\markright` und `\pagenumbering` beeinflussen kann.

6.2 Gleitobjekte

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^AT_EX die beiden Umgebungen `figure` und `table`.

6.2.1 Abbildungen (`figure`)

Diese Umgebung ist für die Behandlung von Abbildungen gedacht. Tatsächlich spielt es aber keine Rolle, *wie* diese erzeugt wurden: 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 L^AT_EX 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` (siehe Abschnitt 8.1.3) leerer Raum für ein später einzusetzendes Bild gelassen:

Abbildung 6 auf S. 36 zeigt ein Beispiel aus der Minimal art.

```
Abbildung~\ref{weiss} auf
S.~\pageref{weiss} zeigt
ein Beispiel aus der
Minimal art.
\begin{figure}[tb]
\vspace{6cm}
\caption{Landschaft im
Nebel} \label{weiss}
\end{figure}
```

L^AT_EX 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 Parameter in den eckigen Klammern, die wahlweise angegeben werden können, dienen dazu, die Plazierung der Abbildung auf die angegebenen Orte *einzuschränken*. Durch Angabe von z. B. `tb` wird L^AT_EX angewiesen, nur eine Plazierung 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`.

Eine Plazierungsbeschränkung *nur* auf `[h]` ist unsinnig; sie würde das „Gleiten“ ja gerade verhindern. Wenn der Platz „hier“ nicht ausreicht, verschiebt L^AT_EX dann die Abbildung mindestens bis zum Anfang der nächsten Seite, so als hätte man `[ht]` angegeben.

Abbildung 6: Landschaft im Nebel

Eine Abbildung, die nicht plziert werden konnte, wird von \LaTeX immer weiter nach hinten verschoben (und schiebt alle weiteren Abbildungen vor sich her!), bis ein neues Kapitel beginnt, das Dokument zu Ende ist, oder der Befehl `\clearpage` eingegeben wird.

Es gibt noch einen weiteren Plzierungsparameter, ! (bang), der \LaTeX anweist, gewisse eingebaute Beschränkungen zu ignorieren, z. B., daß bei der Plzierung gemäß `h`, `t` oder `b` ein Mindestanteil der Seite für normalen Text übrig bleiben muß. „Bang“ muß immer zusammen mit mindestens einem der vier anderen Parameter benutzt werden.

6.2.2 Tabellen (`table`)

Damit Tabellen nicht auf einen Seitenwechsel fallen, können sie, analog zu Abbildungen, zwischen `\begin{table}` und `\end{table}` gesetzt werden. Die Befehle `\caption`, `\label`, `\ref` und `\pageref` wirken entsprechend. Hier sind beide möglichen Konventionen verbreitet: Die Bezeichnung wird entweder immer *über* oder immer *unter* die Tabelle gesetzt.

Auch hier gilt, daß in der `table`-Umgebung beliebiger Text stehen darf; die Tabelle muß nicht zwangsläufig durch die `tabular`-Umgebung erzeugt worden sein. Der Unterschied zu `figure` besteht nur darin, daß die Bezeichnung mit dem Wort „Tabelle“ versehen wird, und daß die Tabellen unabhängig von den Abbildungen nummeriert werden.

7 Schriften

Normalerweise wählt L^AT_EX die Größe und den Stil der Schrift aufgrund der Befehle aus, die die logische Struktur des Textes angeben: Überschriften, Fußnoten, Hervorhebungen usw. Im folgenden werden Befehle und Makropakete beschrieben, mit denen die Schrift auch explizit beeinflusst werden kann. Ausführlichere Erläuterungen zum Umgang mit Schriften in L^AT_EX findet man im *L^AT_EX-Begleiter* [3] und in der Online-Dokumentation [9].

7.1 Schriftgrößen

Die in der Tabelle 16 angeführten Befehlen wechseln die Schriftgröße. Sie spezifizieren die Größe relativ zu der von `\documentclass` festgelegten Grundschrift. Ihr Wirkung reicht bis zum Ende der aktuellen Gruppe oder Umgebung.

Tabelle 16: Schriftgrößen

<code>\tiny</code>	winzig kleine Schrift
<code>\scriptsize</code>	sehr kleine Schrift (wie Indizes)
<code>\footnotesize</code>	kleine Schrift (wie Fußnoten)
<code>\small</code>	kleine Schrift
<code>\normalsize</code>	normale Schrift
<code>\large</code>	große Schrift
<code>\Large</code>	größere Schrift
<code>\LARGE</code>	sehr große Schrift
<code>\huge</code>	riesig groß
<code>\Huge</code>	gigantisch

Die Größen-Befehle verändern auch die Zeilenabstände auf die jeweils passenden Werte – aber nur, wenn die Leerzeile, die den Absatz beendet, innerhalb des Gültigkeitsbereichs des Größen-Befehls liegt:

zu enger Abstand	<code>{\Large zu enger \\ Abstand}\par</code>
---------------------	---

richtiger Abstand	<code>{\Large richtiger\ Abstand}\par}</code>
----------------------	---

Für korrekte Zeilenabstände darf die schließende geschwungene Klammer also nicht zu früh kommen, sondern erst nach einem Absatzende, das übrigens nicht nur als Leerzeile, sondern auch als Befehl `\par` eingegeben werden kann.

7.2 Schriftstil

Der Schriftstil wird in L^AT_EX durch 3 Merkmale definiert:

Familie Standardmäßig stehen 3 Familien zur Wahl: „roman“ (Antiqua), „sans serif“ (Serifenlose) und „typewriter“ (Schreibmaschinenschrift).

Serie Die Serie gibt Stärke und Laufweite der Schrift an: „medium“ (normale Schrift), „boldface extended“ (fett und breiter).

Form Die Form der Buchstaben: „upright“ (aufrecht), „slanted“ (geneigt), „italic“ (kursiv), „caps and small caps“ (Kapitälchen).

Tabelle 17 zeigt die Befehle, mit denen diese Attribute explizit beeinflußt werden können. Die Befehle der Form `\text{...}` setzen nur ihr Argument im gewünschten Stil. Zu jedem dieser Befehle ist ein Gegenstück angegeben, das von seinem Auftreten an bis zum Ende der laufenden Gruppe oder Umgebung wirkt.

Zu beachten ist, daß Wörter in Schreibmaschinenschrift nicht automatisch getrennt werden.

Tabelle 17: Schriftstile

<code>\textrm{text}</code>	<code>\rmfamily</code>	Antiqua
<code>\textsf{text}</code>	<code>\sffamily</code>	Serifenlose
<code>\texttt{text}</code>	<code>\ttfamily</code>	Maschinenschrift
<code>\textmd{text}</code>	<code>\mdseries</code>	normal
<code>\textbf{text}</code>	<code>\bfseries</code>	fett, breiter laufend
<code>\textup{text}</code>	<code>\upshape</code>	aufrecht
<code>\textsl{text}</code>	<code>\slshape</code>	<i>geneigt</i>
<code>\textit{text}</code>	<code>\itshape</code>	<i>kursiv</i>
<code>\textsc{text}</code>	<code>\scshape</code>	KAPITÄLCHEN
<code>\textnormal{text}</code>	<code>\normalfont</code>	Die Grundschrift des Dokuments

Die Befehle für Familie, Serie und Form können untereinander und mit den Größen-Befehlen kombiniert werden; allerdings muß nicht jede mögliche Kombination tatsächlich als reale Schrift (Font) zur Verfügung stehen.

Die kleinen **fetten** Römer beherrschten
das ganze große *Italien*.

plakativ

```
{\small Die kleinen
\textbf{fetten} R"omer
beherrschten }{\large das
ganze gro"se \textit{Italien}.}
{\Large\sffamily\slshape plakativ}
```

Je *weniger* verschiedene Schriftarten man verwendet, desto lesbarer und schöner wird das Schriftstück!

7.3 Andere Schriftfamilien

Mit den im vorigen Abschnitt eingeführten Befehlen kann man nicht beeinflussen, welche Schriftfamilien tatsächlich als Antiqua, Serifenlose und Maschinenschrift benutzt werden. \LaTeX verwendet als Voreinstellung die sog. Computer-Modern-Schriftfamilien (CM), siehe Tabelle 18; der Stil der mathematischen Zeichensätze paßt dabei zu CM Roman.

Will man andere Schriften benutzen, dann ist der einfachste Weg das Laden eines Pakets, das eine oder mehrere dieser Schriftfamilien komplett ersetzt. Tabelle 18 führt einige derartige Pakete auf.

Die Dokumentation Ihres \TeX -Systems [7] sollte darüber informieren, welche Schriften verfügbar sind und wie Sie weitere installieren und verwenden können. Insbesondere sollte eine Anzahl von verbreiteten PostScript-Schriften mit jedem aktuellen \LaTeX -System verwendbar sein [12].

Tabelle 18: Pakete für alternative Schriftfamilien (Eine leere Tabellenspalte bedeutet, daß das Paket die betreffende Schriftfamilie nicht verändert; * kennzeichnet die jeweils als Grundschrift eingestellte Familie.)

Paket	Antiqua	Serifenlose	Schreibmaschine	math. Formeln
(keines)	CM Roman *	CM Sans Serif	CM Typewriter	\approx CM Roman
<code>ccfonts</code>	Concrete *			\approx Concrete
<code>cmbright</code>		CM Bright *	CM Typewriter Light	\approx CM Bright
<code>mathptmx</code>	Times *			\approx Times
<code>mathpazo</code>	Palatino *			\approx Palatino
<code>helvet</code>		Helvetica		
<code>courier</code>			Courier	

7.4 Die „europäischen“ Zeichensätze

\LaTeX verwendet standardmäßig Schriften mit einem Umfang von 128 Zeichen. Umlaute oder akzentuierte Buchstaben sind darin nicht enthalten; sie werden jeweils aus dem Grundsymbol und dem Akzent zusammengesetzt.

Inzwischen stehen die meisten der mit \LaTeX verwendbaren Schriften auch mit einem erweiterten „europäischen“ Zeichenvorrat bereit. Sie enthalten jetzt 256 Zeichen, welche fast alle europäischen Sprachen abdecken, d. h., jedes benötigte Zeichen ist vorgefertigt in ihnen enthalten. Das hat nicht nur eine höhere typographische Qualität zur Folge; aufgrund der inneren Arbeitsweise von \TeX entfallen damit auch die Einschränkungen im Zusammenhang mit der Silbentrennung, die im Abschnitt 3.2.2 erwähnt wurden: Wörter mit Umlauten werden nun besser getrennt, und im Argument des Befehls `\hyphenation` dürfen auch Umlaute und das scharfe s stehen.

Die europäischen Schriften bestehen aus zwei Teilen: Der T1-Zeichensatz enthält Buchstaben, ASCII-Zeichen sowie verschiedene Anführungszeichen und Striche, während ein ergänzender TS1-Zeichensatz zusätzliche Textsymbole bereitstellt.

L^AT_EX wird veranlaßt, T1-Schriften zu verwenden, indem man das Paket `fontenc` mit der Option `T1` lädt:

```
\usepackage[T1]{fontenc}
```

Das Paket `textcomp` ermöglicht den Zugriff auf die Textsymbole:

```
\usepackage{textcomp}
```

Welche zusätzlichen Zeichen mit den T1-Schriften bereitgestellt werden, ist in [8] zusammengefaßt; Anhang A der vorliegenden Kurzbeschreibung enthält eine Liste aller TS1-Textsymbole. Einige der Textsymbole sind auch ohne das Paket `textcomp` verfügbar, siehe Abschnitt 3.4.5, dann aber nicht immer in einem zur laufenden Schrift passenden Stil.

Beachten Sie, daß in Fonts, die nicht speziell für die Verwendung mit T_EX entworfen wurden, nur ein Teil der TS1-Textsymbole enthalten ist. Das betrifft vor allem die „handelsüblichen“ PostScript-Schriften.

8 Spezialitäten

Das komplette Menü der Spezialitäten, die von \LaTeX serviert werden, ist im *\LaTeX -Handbuch* [1] und in der Online-Dokumentation beschrieben. Hier soll nur auf einige besondere „Schmankerln“ hingewiesen werden.

8.1 Abstände

8.1.1 Zeilenabstand

Um in einem Schriftstück größere Zeilenabstände zu verwenden, als es in der Dokumentklasse vorgesehen ist, gibt es in \LaTeX den Befehl `\linespread`, der im Vorspann stehen sollte und dann auf das gesamte Dokument wirkt. Das kann beispielsweise dann notwendig werden, wenn eine Schrift benutzt wird, die eine größerer x-Höhe hat als die voreingestellte Computer-Modern. Für die Schrift „Palatino“ etwa ist eine Vergrößerung des Zeilenabstandes um ca. 5% angemessen:

```
\usepackage{mathpazo}
\linespread{1.05}
```

8.1.2 Spezielle horizontale Abstände

Die Abstände zwischen Wörtern und Sätzen werden von \LaTeX automatisch gesetzt. Sonstigen horizontalen Abstand kann man mit dem Befehl

```
\hspace{länge}
```

einfügen. 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 19 angeführt. Die Befehle in Tabelle 20 sind Abkürzungen zum Einfügen besonderer horizontaler Abstände. Der Befehl `\hfill` kann dazu dienen, einen vorgegebenen Platz auszufüllen.

Tabelle 19: Einheiten für Längenangaben

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

Tabelle 20: Befehle für horizontale Abstände

<code>\,</code>	ein sehr kleiner Abstand (siehe auch Abschnitt 3.3)
<code>\enspace</code>	so breit wie eine Ziffer
<code>\quad</code>	so breit, wie ein Buchstabe hoch ist („weißes Quadrat“)
<code>\qqquad</code>	doppelt so breit wie ein <code>\quad</code>
<code>\hfill</code>	ein Abstand, der sich von 0 bis ∞ ausdehnen kann.

Tabelle 21: 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>	ein Abstand, der sich von 0 bis ∞ ausdehnen kann

Schafft mir	Raum!	Schafft mir\hspace{1.5cm}Raum! \\ \triangleleft \(\triangleleft\)\hfill \triangleright \(\triangleright\)
-------------	-------	---

8.1.3 Spezielle vertikale Abstände

Die Abstände zwischen Absätzen, Kapiteln usw. werden von L^AT_EX automatisch bestimmt. In Spezialfällen kann man zusätzlichen Abstand *zwischen zwei Absätzen* mit dem Befehl

$$\backslash\mathrm{vspace}\{l\ddot{a}n\mathrm{g}e\}$$

bewirken. 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 21 sind Abkürzungen für bestimmte vertikale Abstände. Der Befehl `\vfill` in Verbindung mit `\newpage` kann dazu dienen, Text an den unteren Rand einer Seite zu setzen oder vertikal zu zentrieren. Beispielsweise enthält der Quelltext für die zweite Seite der vorliegenden Beschreibung:

\vfill

Dieses Dokument wurde mit \LaTeX{} gesetzt.

...

\newpage

Zusätzlichen Abstand zwischen zwei Zeilen *innerhalb* eines Absatzes oder einer Tabelle erreicht man mit dem Befehl `\\[länge]`.

Albano Cesara
 Lindenallee 10
 95632 Pestitz

Albano Cesara \\
 Lindenallee 10 \\[1.5ex]
 95632 Pestitz

8.2 Briefe

Mit der Dokumentklasse `letter` kann man zwischen `\begin{document}` und `\end{document}` einen oder mehrere Briefe schreiben. Abbildung 7 enthält ein Beispiel für einen Brief.

```

\documentclass[12pt,a4paper]{letter}
\usepackage[latin1]{inputenc}
\usepackage{german}
\address{EDV-Zentrum der TU Wien \\
  Abt. Digitalrechenanlage \\
  Wiedner Hauptstraße 8--10 \\
  A-1040 Wien}
\signature{Dr. Hubert Partl}
\begin{document}
\begin{letter}{Frau Mag. Elisabeth Schlegl \\
  EDV-Zentrum der Karl-Franzens-Universität \\
  Attemsgasse 25/II \\
  \textbf{A-8010 Graz}}
\opening{Liebe Frau Schlegl,}
herzlichen Dank für die Zusendung ...

... in etwa 2--3~Wochen fertig zu sein.
\closing{Mit freundlichen Grüßen}
\end{letter}
\end{document}

```

Abbildung 7: Brief von H. P. an E. S.

Mit dem Befehl `\address` definiert man die Adresse des Absenders. `\begin{letter}{...}` beginnt einen Brief an den im Parameter angegebenen Empfänger. `\opening{...}` schreibt die Anrede und `\closing{...}` den abschließenden Gruß, an den automatisch die eingangs mit `\signature` vereinbarte Unterschrift angefügt wird. `\end{letter}` beendet den jeweiligen Brief.

Das von der Dokumentklasse `letter` bewirkte Layout der Briefe orientiert sich an amerikanischen Gepflogenheiten. Mit vielen \LaTeX -Systemen ist die Klasse `dinbrief` verfügbar; sie setzt die Briefe in einer Anordnung gemäß DIN 676, die für die Verwendung von A4-Bogen in Fensterkuverts geeignet ist. Der *Local Guide* [7] sollte Auskunft über diese oder andere Alternativen zu `letter` geben.

8.3 Literaturangaben

Mit der `thebibliography`-Umgebung kann man ein Literaturverzeichnis erzeugen. Darin beginnt jede Literaturangabe mit `\bibitem`. Als Parameter wird ein Name vereinbart, unter dem die Literaturstelle im Text zitiert werden kann, und dann folgt der Text der Literaturangabe. Die Numerierung erfolgt automatisch. Der Parameter bei `\begin{thebibliography}` gibt die maximale Breite dieser Nummernangabe an, also z. B. `{99}` für maximal zweistellige Nummern.

Im Text zitiert man die Literaturstelle dann mit dem Befehl `\cite` und dem vereinbarten Namen als Argument.

Partl [1] hat vorgeschlagen, daß ...

`Partl~\cite{pa}` hat
vorgeschlagen ...

Literatur

- [1] H. Partl: *German T_EX*, TUGboat Vol. 9, No. 1 (1988)

```
\begin{thebibliography}{99}
\bibitem{pa}
H.~Partl: \textit{German TEX},
TUGboat Vol.~9, No.~1 (1988)
\end{thebibliography}
```

8.4 Zerbrechliche Befehle

Manche L^AT_EX-Befehle „verfrachten“ ihre Argumente an eine andere Stelle im Text; beispielsweise kann das Argument von `\section` auch im Inhaltsverzeichnis und möglicherweise in der Kopfzeile auftauchen.

Bestimmte Befehle „überstehen“ diesen Transport nicht, wenn sie ohne besondere Maßnahmen in einem solchen „beweglichen Argument“ auftreten. Derartige Befehle heißen „zerbrechlich“. Damit sie dennoch innerhalb von beweglichen Argumenten benutzt werden dürfen, muß man ihnen einfach den Befehl `\protect` voranstellen.

Zerbrechlich sind insbesondere alle Befehle, die ein optionales Argument kennen, also auch `\` (sic!), außerdem die Befehle `\(, \)` und `\footnote`.

Bewegliche Argumente haben, neben den Gliederungsbefehlen, auch der Befehl `\caption` und die Umgebung `letter`.

A Mit dem Paket textcomp verfügbare Symbole

,	\textquotestraightbase*	"	\textquotestraightdblbase*
—	\texttwelveudash*	—	\textthreequartersemdash*
←	\textleftarrow	→	\textrightarrow
␣	\textblank	\$	\\$*
'	\textquotesingle*	*	\textasteriskcentered*
=	\textdblhyphen	/	\textfractionsolidus*
<	\textlangle	—	\textminus*
>	\textrangle	∪	\textmho
○	\textbigcircle	Ω	\textohm
⌈	\textlbrackdbl	⌋	\textrbrackdbl
↑	\textuparrow	↓	\textdownarrow
`	\textasciigrave*	★	\textborn
∘	\textdivorced	†	\textdied
🍃	\textleaf	∞	\textmarried
♪	\textmusicalnote	~	\texttildelow*
=	\textdblhyphenchar	˘	\textasciibreve*
˘	\textasciicaron*	”	\textacutedbl*
“	\textgravedbl*	‡	\dag*
‡	\ddag*		\textbardbl*
‰	\textperthousand*	●	\textbullet*
°C	\textcelsius*	\$	\textdollaroldstyle
¢	\textcentoldstyle	f	\textflorin*
⌠	\textcolonmonetary	₩	\textwon
₦	\textnaira	₵	\textguarani
₱	\textpeso	£	\textlira
℞	\textrecipe	‽	\textinterrobang
⤵	\textinterrobangdown	₫	\textdong
™	\texttrademark*	‰	\textpertenthousand
¶	\textpilcrow	฿	\textbaht
№	\textnumero	%	\textdiscount
€	\textestimated	○	\textopenbullet
SM	\textservicemark	{	\textlquill
}	\textrquill	¢	\textcent*
£	\pounds*	¤	\textcurrency*
¥	\textyen*		\textbrokenbar*
§	\S*	..	\textasciidieresis*
©	\copyright*	♀	\textordfeminine*
⊕	\textcopyleft	¬	\textlnot*
Ⓟ	\textcircledP	®	\textregistered*
—	\textasciimacron*	°	\textdegree*
±	\textpm*	²	\texttwosuperior
³	\textthreesuperior	´	\textasciiacute*
μ	\textmu*	¶	\P*
·	\textperiodcentered*	※	\textreferencemark
¹	\textonesuperior	♂	\textordmasculine*
√	\textsurd	¼	\textonequarter
½	\textonehalf	¾	\textthreequarters
€	\textsf{\texteuro}	×	\texttimes*
÷	\textdiv*		

Schriften, die nicht speziell für die Verwendung mit T_EX entworfen wurden, enthalten normalerweise nur die mit * markierten Zeichen.

Literatur

- [1] L. Lamport: *Das L^AT_EX-Handbuch*. Addison-Wesley Deutschland (1995). Deutsche Übersetzung von [2].
- [2] L. Lamport: *L^AT_EX, A Document Preparation System*. Addison-Wesley, 2. Aufl. (1994).
- [3] M. Goossens, F. Mittelbach und A. Samarin: *Der L^AT_EX-Begleiter*. Addison Wesley Longman, 2. korr. Nachdruck (1996). Deutsche Übersetzung von [4].
- [4] M. Goossens, F. Mittelbach und A. Samarin: *The L^AT_EX Companion*. Addison-Wesley (1994).
- [5] M. Goossens, F. Mittelbach und A. Samarin: *Higher Mathematics*. Aktualisierte Fassung (1998) von Kapitel 8 aus [4].
<<ftp://dante.ctan.org/tex-archive/info/companion-rev/ch8.pdf>>
- [6] M. Goossens, S. Rahtz und F. Mittelbach: *The L^AT_EX Graphics Companion*. Addison Wesley Longman (1997).
- [7] Zu jedem installierten L^AT_EX-System sollte ein *L^AT_EX Local Guide* vorhanden sein, in dem alle für dieses System spezifischen Angaben – z. B. die für den Aufruf der Programme notwendigen Befehle und die zur Verfügung stehenden Dokumentklassen, Pakete und Schriften – angeführt sind.
- [8] L^AT_EX3 Project Team (Hrsg.): *L^AT_EX 2_ε for authors*. Bestandteil der Online-Dokumentation von L^AT_EX, Datei `usrguide.tex`. Aktuelle Änderungen und Ergänzungen sowie die Unterschiede zum früheren L^AT_EX 2.09 sind hier dokumentiert.
- [9] L^AT_EX3 Project Team (Hrsg.): *L^AT_EX 2_ε font selection*. Bestandteil der Online-Dokumentation von L^AT_EX, Datei `fontguide.tex`.
- [10] L^AT_EX3 Project Team (Hrsg.): *L^AT_EX 2_ε for class and package writers*. Bestandteil der Online-Dokumentation von L^AT_EX, Datei `clsguide.tex`.
- [11] D. P. Carlisle: *Packages in the ‘graphics’ bundle*. Bestandteil der Online-Dokumentation von L^AT_EX, Datei `grfguide.ps`.
- [12] W. Schmidt: *Using common PostScript fonts with L^AT_EX*. Bestandteil der Online-Dokumentation von L^AT_EX (seit Juni 2000), Datei `psnfss2e.pdf`.
- [13] H. Partl und A. Kielhorn: *Layout-Änderungen mit L^AT_EX*. EDV-Zentrum der Technischen Universität Wien (1996).
<<ftp://dante.ctan.org/tex-archive/macros/latex/contrib/supported/refman/>>
- [14] D. F. Langmyhr: *How to make your own document style in L^AT_EX 2_ε*. In: *Proceedings of the Eighth European T_EX Conference* (1994).

- [15] A. Reichert: *Typografie – Gestaltung einer Beispielklasse*. (1999)
<<ftp://dante.ctan.org/info/german/typografie/>>
- [16] K. Reckdahl: *Using Imported Graphics in L^AT_EX 2_ε*. (1997)
<<ftp://dante.ctan.org/tex-archive/info/epslatex.ps>>
- [17] D. E. Knuth: *Computers & Typesetting, Vol. A: The T_EX Book*. Addison-Wesley (1991).
- [18] N. Schwarz: *Einführung in T_EX– incl. Version 3.0*. Oldenbourg, 3. Aufl. (1991).
- [19] H. Partl: *German T_EX*. TUGboat Vol. 9, No. 1 (1988).
- [20] B. Raichle: *Kurzbeschreibung – german.sty*.
<<ftp://dante.ctan.org/tex-archive/language/german/gerdoc.tex>>
- [21] B. Raichle, R. Niepraschk und Th. Hafner: *Fragen und Antworten (FAQ) über das Textsatzsystem T_EX und DANTE e.V.*
<<http://www.dante.de/faq/de-tex-faq/>>