

L^AT_EX 2_ε-Kurzbeschreibung

Version 3.0c

8. April 2018

Marco Daniel

Patrick Gundlach

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* [?] in Verbindung mit der Online-Dokumentation.

Autoren: 1998–2016 M. Daniel, P. Gundlach, W. Schmidt, J. Knappen, H. Partl, I. Hyna

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

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 von Mitgliedern der Fachschaft WIAI der Universität Bamberg verändert, um als Arbeitsgrundlage für einen L^AT_EX-Workshop zu dienen. Daher wurden auch bewusst Fehler gegenüber dem Original eingefügt, die in den Aufgaben behoben werden sollen. Dieses Dokument ist nicht dafür bestimmt, ohne die entsprechenden Aufgabenstellungen weitergegeben zu werden. Das Originaldokument ist unten verlinkt.

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

<http://mirror.ctan.org/info/lshort/german/>

Die Änderungen seit Version 2.3 (10. April 2003) sind unter <https://github.com/texdoc/l2kurz> einzusehen.

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

Inhaltsverzeichnis

1. Allgemeines	5
1.1. The Name of the Game	5
1.1.1. T _E X	5
1.1.2. L ^A T _E X	5
1.2. Grundkonzept	5
1.2.1. Autor, Designer und Setzer	5
1.2.2. Layout-Design	6
1.2.3. Vor- und Nachteile	6
1.2.4. Der Arbeitsablauf	7
2. Eingabefile	9
2.1. Leerstellen	9
2.2. L ^A T _E X-Befehle und Gruppen	9
2.3. Kommentare	10
2.4. Aufbau	10
2.5. Dokumentklassen	11
2.6. Pakete	13
2.7. Eingabezeichensatz	14
3. Setzen von Text	16
3.1. Deutschsprachige Texte	16
3.2. Zeilen- und Seiten-Umbruch	16
3.2.1. Blocksatz	16
3.2.2. Silbentrennung	17
3.3. Wortabstand	18
3.4. Spezielle Zeichen	18
3.4.1. Anführungszeichen	18
3.4.2. Binde- und Gedankenstriche	19
3.4.3. Punkte	20
3.4.4. Ligaturen und Unterschneidungen	20
3.4.5. Symbole, Akzente und besondere Buchstaben	21
3.5. Kapitel und Überschriften	22
3.6. Fußnoten	23
3.7. Hervorgehobener Text	23
3.8. Hochgestellter Text	24
3.9. Umgebungen	24
3.9.1. Zitate (quote, quotation, verse)	24
3.9.2. Listen (itemize, enumerate, description)	25

3.9.3.	Flattersatz (flushleft, flushright, center)	25
3.9.4.	Direkte Ausgabe (verbatim, verb)	27
3.9.5.	Tabulatoren (tabbing)	27
3.9.6.	Tabellen (tabular)	28
4.	Setzen von mathematischen Formeln	29
4.1.	Allgemeines	29
4.2.	Elemente in mathematischen Formeln	31
4.3.	Nebeneinander Setzen	34
4.4.	Übereinander Setzen	35
5.	Setzen von Bildern	37
6.	Seitenaufbau	38
6.1.	Kopf- und Fußzeilen	38
6.2.	Gleitobjekte	38
6.2.1.	Abbildungen (figure)	38
6.2.2.	Tabellen (table)	40
7.	Schriften	41
7.1.	Schriftgrößen	41
7.2.	Schriftstil	42
7.3.	Andere Schriftfamilien	43
7.4.	Die „europäischen“ Zeichensätze	43
8.	Spezialitäten	46
8.1.	Abstände	46
8.1.1.	Zeilenabstand	46
8.1.2.	Spezielle horizontale Abstände	46
8.1.3.	Spezielle vertikale Abstände	47
8.2.	Literaturangaben	49
A.	Mit dem Paket textcomp verfügbare Symbole	51
B.	Liste der mathematischen Symbole	52

1. Allgemeines

1.1. The Name of the Game

1.1.1. T_EX

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

1.1.2. L^AT_EX

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

L^AT_EX 2_ε ist die aktuelle Version und mit dem Fokus auf Stabilität werden derzeit nur noch Fehler behoben. Eine Weiterentwicklung findet im L^AT_EX 3-Projekt statt, einige Zusatzmodule (Pakete) für L^AT_EX benutzen schon die neue Version, für den Benutzer ist dies jedoch in der Regel unsichtbar.

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. L^AT_EX ist in diesem Sinne der Buch-Designer, das Programm T_EX 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. L^AT_EX 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 L^AT_EX 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 designt. 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 Nummerierung 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. L^AT_EX 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 L^AT_EX 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 L^AT_EX hingegen aufgrund seiner Arbeitsweise weniger geeignet.

1.2.3. Vor- und Nachteile

Gegenüber anderen Textverarbeitungs- oder DTP-Programmen zeichnet sich L^AT_EX 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.

L^AT_EX-Dokumente sind zwischen verschiedenen Installationen und Rechnerplattformen austauschbar.

Im Gegensatz zu vielen WYSIWYG-Programmen bearbeitet L^AT_EX auch lange oder komplizierte Dokumente zuverlässig, und sein Ressourcenverbrauch (Speicher, Rechenleistung) ist vergleichsweise mäßig.

Ein Nachteil soll freilich auch nicht verschwiegen werden:

Dadurch, dass der Text erst von L^AT_EX nach PDF gewandelt wird, unterscheidet sich der Arbeitsablauf von L^AT_EX stark von den üblichen Textverarbeitungen bzw. DTP-Programmen. Das erfordert ein Umdenken und eine gewisse Einarbeitung.

1.2.4. Der Arbeitsablauf

Der typische Ablauf beim Arbeiten mit L^AT_EX ist:

1. Ein Eingabefile schreiben, das den Text und die L^AT_EX-Befehle enthält.
2. Dieses File mit L^AT_EX 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 L^AT_EX-Bearbeitung des Textes muss noch von Hand angestoßen werden und läuft ebenfalls in einem eigenen Fenster ab.

Wenn der Texteditor keine Schnittstelle anbietet, um L^AT_EX direkt aus einem Menüpunkt heraus aufzurufen, dann ist der übliche Weg über die Kommandozeile bzw. Eingabeaufforderung. Dort wird dann das Kommando `pdflatex` aufgerufen und als Parameter wird der Name der Datei angegeben, unter der das Dokument auf der Festplatte gespeichert ist: `pdflatex masterarbeit.tex`

Das Ergebnis des Aufrufs ist eine PDF-Datei, die wie die Eingabedatei heißt, nur mit der Endung `.pdf`. L^AT_EX gibt einige Meldungen auf der Konsole aus, die beispielsweise Auskunft über die Anzahl der Seiten des Dokuments geben.

2. Eingabefile

Das Eingabefile für L^AT_EX ist ein Textfile mit der Endung `.tex`. Es wird mit einem Editor erstellt und enthält sowohl den Text, der gedruckt werden soll, als auch die Befehle, aus denen L^AT_EX erfährt, wie der Text gesetzt werden soll. Als Editor bietet sich ein spezieller L^AT_EX-Editor an wie beispielsweise Texmaker (www.xm1math.net/texmaker). Diese Editoren bieten neben Syntaxhervorhebung und -überprüfung auch vordefinierte Arbeitsabläufe, so dass der Benutzer sich auf die Erstellung des Texts konzentrieren kann. Es ist aber auch möglich und gängige Praxis, *normale* Texteditoren wie emacs, vim oder notepad++ zu benutzen.

2.1. Leerstellen

„Unsichtbare“ Zeichen wie das Leerzeichen, Tabulatoren und das Zeilenende werden von L^AT_EX 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 L^AT_EX-Befehlen.

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

2.2. L^AT_EX-Befehle und Gruppen

Die meisten L^AT_EX-Befehle haben eines der beiden folgenden Formate: Entweder sie beginnen mit einem Backslash (`\`) und haben dann einen nur aus Buchstaben bestehenden Namen, der durch ein oder mehrere Leerzeichen oder durch ein nachfolgendes Sonderzeichen beendet wird; oder sie bestehen aus einem Backslash und genau einem Sonderzeichen. 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 41. Mai 2019.

Oder: Heute ist der 41. Mai 2019.

Falsch ist:

Am 41. Mai 2019regnet es.

Richtig ist:

Am 41. Mai 2019 scheint die Sonne.

Oder: Am 41. Mai 2019 schneit es.

Heute ist der `\today`.

Oder: Heute ist der `\today` .

Falsch ist:

Am `\today` regnet es.

Richtig ist:

Am `\today{ }` scheint die Sonne.

Oder: Am `\today\` 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 L^AT_EX ignoriert. Dies kann für Notizen des Autors verwendet werden, die nicht oder noch nicht ausgedruckt werden sollen.

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

2.4. Aufbau

Der erste Befehl in einem L^AT_EX-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 bzw. deklariert werden, 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 L^AT_EX-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 L^AT_EX ignoriert.

Listing 1 zeigt ein *minimales* L^AT_EX-File. Ein etwas komplizierteres File ist in Listing 2 skizziert.

Listing 1: Ein minimales L^AT_EX-File

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

Listing 2: Aufbau eines Artikels

```
\documentclass[11pt,a4paper,ngerman]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{babel}
\date{29. Februar 1998}
\author{H.~Partl}
\title{Über kurz oder lang}

\begin{document}
\maketitle
\tableofcontents

\section{Start}
Hier beginnt mein schönes Werk ...

\section{Ende}
... und hier endet es.

\end{document}
```

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 der nächsten Seite führt Klassen auf, die in jeder vollständigen L^AT_EX-Installation existieren.

Zwischen den eckigen Klammern *können*, durch Kommas getrennt, eine oder mehrere Optionen für das Klassenlayout angegeben werden. Die wichtigsten Op-

tionen für die Standardklassen sind in der Tabelle 2 angeführt. Das Eingabefile für diese Beschreibung beginnt z. B. mit:

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

Tabelle 1: Dokumentklassen

<code>article</code>	für Artikel in wissenschaftlichen Zeitschriften, kürzere Berichte u. v. a.
<code>report</code>	für längere Berichte, die aus mehreren Kapiteln bestehen, Diplomarbeiten, Dissertationen u. ä.
<code>book</code>	für Bücher
<code>scrartcl</code> , <code>scrreprt</code> , <code>scrbook</code>	Die sog. KOMA-Klassen sind Varianten der o. g. Klassen mit besserer Anpassung an DIN-Papierformate und „europäische“ Typographie. Zudem erlauben sie mittels Optionen die Anpassung des Layouts. Entgegen den Standardklassen existiert eine deutsche und englische Dokumentation (siehe [?])
<code>beamer</code>	für Präsentationen

Tabelle 2: Klassenoptionen für die Standardklassen

<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 nehmen die Standardklassen das amerikanische Papierformat.
<code>fleqn</code>	für linksbündige statt zentrierte mathematische Gleichungen
<code>leqno</code>	für Gleichungsnummern links statt rechts von jeder nummerierten Gleichung

Forsetzung nächste Seite

Tabelle 2: Klassenoptionen für die Standardklassen

<code>titlepage</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>notitlepage</code>	
<code>onecolumn</code>	für ein- oder zweispaltigen Satz. Die Voreinstellung ist immer <code>onecolumn</code> .
<code>twocolumn</code>	
<code>oneside</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> .
<code>twoside</code>	

2.6. Pakete

Mit dem Befehl

```
\usepackage[<optionen>]{<paket>}
```

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. Der Befehl `\usepackage` erlaubt das Laden mehrerer Pakete, wobei eine Optionenübergabe nicht möglich ist, bzw. die Optionen für alle Pakete gelten.

Tabelle 3: Pakete (eine Auswahl)

<code>amsmath</code> , <code>amssymb</code>	Mathematischer Formelsatz mit erweiterten Fähigkeiten, zusätzliche mathematische Schriften und Symbole; Beschreibung siehe [?].
<code>array</code>	Verbesserte und erweiterte Versionen der Umgebungen <code>array</code> , <code>tabular</code> und <code>tabular*</code> .
<code>babel</code>	Anpassungen für viele verschiedene Sprachen. Die gewählten Sprachen werden als Optionen angegeben.
<code>csquotes</code>	Mächtiges Paket zum korrekten Setzen (fremdsprachlicher) Anführungszeichen
<code>fontenc</code>	Erlaubt die Verwendung von Schriften mit unterschiedlicher Kodierung (Zeichenvorrat, Anordnung).

Fortsetzung nächste Seite

Tabelle 3: Pakete (eine Auswahl)

<code>geometry</code>	Manipulation des Seitenlayouts.
<code>graphicx</code>	Einbindung von extern erzeugten Graphiken. Die umfangreichen Möglichkeiten dieses Pakets werden in [?] und [?] beschrieben.
<code>hyperref</code>	Ermöglicht Hyperlinks zwischen Textstellen und zu externen Dokumenten; besonders sinnvoll einsetzbar, wenn mit $\text{\TeX}$ eine Ausgabedatei im PDF- oder HTML-Format erzeugt wird.
<code>imakeidx</code>	Unterstützt das Erstellen eines Index.
<code>listings</code>	Flexible Erweiterung der <code>verbatim</code> -Umgebung.
<code>longtable</code>	für Tabellen über mehrere Seiten mit automatischem Seitenumbruch.
<code>microtype</code>	Verändert den Absatzumbruch leicht, um bessere Zeilenumbrüche zu erzeugen; optischer Randausgleich.
<code>multicol</code>	Mehrspaltiger Satz mit Kolumnenausgleich.
<code>selinput</code>	Deklaration der Zeichenkodierung im Eingabefile.
<code>siunitx</code>	Formatierung von Zahlen und physikalischen Einheiten
<code>tabularx</code>	für Tabellen mit automatisch an den vorhandenen Platz angepasster Breite der Spalten.
<code>textcomp</code>	Bindet Schriften mit zusätzlichen Textsymbolen ein.
<code>xcolor</code>	Unterstützung für Farbausgabe; Beschreibung siehe [?] und [?].

2.7. Eingabezeichensatz

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

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

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

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

Für Zeichen, die über obige Liste hinausgehen, beispielsweise die Umlaute, sind unterschiedliche Kodierungen in Gebrauch. Die Kodierung ist hierbei vom Betriebssystem sowie vom Editor abhängig. Um Umlaute direkt zu nutzen ohne auf spezielle Kodierung einzugehen, bietet sich das Paket `selinput` an. Es ermittelt während des Kompilierungsvorgangs die verwendete Kodierung. Um dies sicher zu stellen, muss das Paket wie folgt in der Präambel eingebunden werden:

Listing 3: Bestimmung der Kodierung für die Nutzung von Umlauten

```
\usepackage{selinput}
\SelectInputMappings{
  adieresis={ä},
  germandbls={ß},
}
% Wörter mit Umlauten werden getrennt:
\usepackage[T1]{fontenc}
```

Neben dem Paket `selinput` existiert auch das Paket `inputenc`. Hier gibt man die zu verwendende Kodierung als Paketoption vor, diese muss man jedoch dafür in Erfahrung bringen, was nicht immer einfach ist. Im Falle von UTF-8 als Eingabekodierung ist das:

```
\usepackage[utf8]{inputenc}
```

Falls L^AT_EX 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 L^AT_EX 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 L^AT_EX-Befehl oder eine Ersatzdarstellung, die ausschließlich mit ASCII-Zeichen auskommt. Näheres darüber erfahren Sie in Abschnitt 3.4.

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 `babel`

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

oder mit der Option `german` für traditionelle Rechtschreibung. Der Grund für diese Unterscheidung ist die unterschiedliche Silbentrennung. Eine ausführliche Beschreibung dieses Pakets findet man in [?].

3.2. Zeilen- und Seiten-Umbruch

3.2.1. Blocksatz

L^AT_EX setzt Text standardmäßig im Blocksatz, d. h. mit Randausgleich, wobei der Zeilen- und Seitenumbruch automatisch durchgeführt wird. 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.

Üblicherweise werden in L^AT_EX Absätze mit Einrückungen gekennzeichnet. Bei den Standardklassen kann dies mittels des Paketes `parskip` deaktiviert werden und Absätze mit Abstand dazwischen kenntlich machen. Die sogenannten KOMA-Klassen bieten hierfür extra Optionen (siehe [?]).

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 L^AT_EX-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.)

L^AT_EX 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 Warnmeldung aus (`overfull hbox`). Das tritt insbesondere dann auf, wenn keine geeignete Stelle für die Silbentrennung gefunden wird. Innerhalb der `sloppypar`-Umgebung ist L^AT_EX 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. Eine weitere Möglichkeit um *schönere* Absätze zu erzeugen ist das Paket `microtype` für PDF_TE_X und neuere Programme. Es verändert einerseits die Breite der Buchstaben in einem so geringen Maß, dass das dem Leser nicht auffällt. Weiterhin erlaubt es gewisse Zeichen etwas über den rechten Rand hinaus zu ragen (Trennstrich, Punkt, Komma, ...), so dass L^AT_EX bessere Umbruchpunkte findet.

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.

Der Befehl `\hyphenation` bewirkt, dass die darin angeführten Wörter jedes Mal 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 indirekt kodierten Umlaute wie "a 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. Mit dem Paket `babel[?]` steht der Befehl `"-`

zur Verfügung, der auch Trennungen an anderen (nicht markierten) Stellen im Wort erlaubt.

Eingabefile,	L ^A T _E X-Eingabefile,	Ein"-gabe"-file,
Hässlichkeit		\LaTeX=Eingabe"-file,
		Häss"-lich"-keit

Der Befehl `\mbox` bewirkt, dass das Argument überhaupt nicht abgeteilt werden kann.

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

Innerhalb des von `\mbox` eingeschlossenen Textes 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.

3.3. Wortabstand

In einigen Fällen kann man sich nicht auf die Automatik von L^AT_EX verlassen: manchmal wird ein Umbruchpunkt eingefügt, an dem keiner sein soll und manchmal möchte man den Abstand zwischen zwei Wörtern (oder Buchstaben) verändern. Dafür werden unter anderem folgende Befehle bereit gestellt:

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.

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.

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.

"No," he said, "I don't know!"

Bei Benutzung des Paketes `babel` (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.

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

Er sagte „Da rief ich ‚Hallo‘“	Er sagte \enquote{Da rief ich \enquote{Hallo}}
--------------------------------	---

Eine gute Einführung in die Typographieregeln für Textsatz findet sich in der PDF-Datei *typokurz* von Christoph Bier[?].

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

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

O-Beine
 10--18~Uhr
 Paris--Dakar
 Schalke 04 -- Hertha BSC
 ja -- oder nein?
 yes---or no?
 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 und Unterscheidungen

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 `babel` steht zusätzlich der Befehl `"|` zur Verfügung, der gleichzeitig eine Trennhilfe darstellt.

Auflage (Auf-lage)

Auf`"|`lage (Auf-lage)

Das Paket `babel`[?] macht noch einige weitere Befehle verfügbar, die bestimmte Besonderheiten der deutschen Sprache berücksichtigen. Die wichtigsten von ihnen sind: `"~` für einen Bindestrich, an dem nicht umbrochen werden darf und `"=` für einen Trennstich, an dem ein Umbruch stattfinden darf, beispielsweise bei zusammengesetzten Hauptwörtern.

x-beliebig	x~beliebig
bergauf und -ab	bergauf und ~ab
Breisgau-Hochschwarzwald	Breisgau="Hochschwarzwald

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:

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

L^AT_EX 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.

Hôtel, naïve, smørebrød.	H\^otel, na\" \i ve, sm\o rebr\o d. \\
Die hässliche Straße.	Die h\"assliche Stra\ss{}e.\\
¡Señorita!	!`Se\~norita!\\
⊗	\textcircled{x}

Benutzt man das Paket `selinput` (siehe Abschnitt 2.7 auf Seite 14), dann darf man diese Zeichen – soweit sie im Zeichensatz des Betriebssystems existieren – auch direkt in das Eingabefile schreiben.

Mit dem Paket `babel` und der Option `ngerman` bzw `german` können Umlaute auch durch einfaches Voranstellen eines doppelten Anführungszeichens (") geschrieben werden, also z. B. "o für „ö“; für scharfes s darf man "s schreiben:

Die hässliche Straße muss schöner werden.	Die h"assliche Stra"se muss sch"oner werden.
---	--

Tabelle 4: Akzente und spezielle Buchstaben

ò	<code>\`o</code>	ó	<code>\'o</code>	ô	<code>\^o</code>	õ	<code>\~o</code>	ō	<code>\=o</code>	ö	<code>\.o</code>
œ	<code>\u o</code>	ő	<code>\v o</code>	ő	<code>\H o</code>	ö	<code>\"o</code>	q	<code>\c o</code>	q	<code>\d o</code>
q	<code>\b o</code>	ő	<code>\r o</code>	ôo	<code>\t oo</code>						
œ	<code>\oe</code>	Œ	<code>\OE</code>	æ	<code>\ae</code>	Æ	<code>\AE</code>	å	<code>\aa</code>	Å	<code>\AA</code>
ø	<code>\o</code>	Ø	<code>\O</code>	l	<code>\l</code>	L	<code>\L</code>	i	<code>\i</code>	j	<code>\j</code>
ß	<code>\ss</code>										

Tabelle 5: Symbole

†	<code>\dag</code>	™	<code>\texttrademark</code>
‡	<code>\ddag</code>	·	<code>\textperiodcentered</code>
¶	<code>\P</code>	•	<code>\textbullet</code>
£	<code>\pounds</code>	©	<code>\copyright</code>
		®	<code>\textregistered</code>

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.

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`:

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

Bei den Klassen `report` und `book`:

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

Artikel können also relativ einfach als Kapitel in ein Buch eingebaut werden. Die Abstände zwischen den Kapiteln, die Nummerierung und die Schriftgröße der Überschrift werden von L^AT_EX 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 11).

Der Befehl `\tableofcontents` bewirkt, dass ein Inhaltsverzeichnis ausgedruckt wird. L^AT_EX 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 Programm L^AT_EX also mindestens zweimal aufrufen, damit man die richtigen Angaben erhält.

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

Mit den Befehlen `\label` und `\ref` ist es möglich, die von L^AT_EX automatisch vergebenen Kapitelnummern im Text anzusprechen. Für `\ref{...}` setzt L^AT_EX 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 nummeriert 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. Im L^AT_EX Begleiter[?] werden Möglichkeiten aufgezählt, diese Einschränkung zu umgehen.

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

3.7. Hervorgehobener Text

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

```
\emph{text}
```

¹Das ist eine Fußnote.

(emphasize) setzt seinen Parameter in einem auffälligen Stil. L^AT_EX verwendet für den hervorgehobenen Text *kursive* Schrift.

Werden innerhalb eines hervorgehobenen Textes nochmals Passagen hervorgehoben, so setzt L^AT_EX 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 12) 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 1 auf der nächsten Seite). Die Umgebung `enumerate` eignet sich für nummerierte Aufzählungen (siehe Abbildung 2 auf der nächsten Seite). Die Umgebung `description` eignet sich für Beschreibungen (siehe Abbildung 3 auf Seite 27). Mit dem Paket `enumitem` können die Umgebungen leicht den eigenen Bedürfnissen angepasst werden.

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

Die Umgebungen `Center`, `FlushLeft` und `FlushRight` aus dem Paket `ragged2e` bewirken zentrierten, links-, und rechtsbündigen Satz. Die Varianten dieser Umgebungen (`center`, `flushleft` und `flushright`), die ohne ein Zusatzpaket zur Verfügung stehen, bewirken im Prinzip dasselbe, nur schaltet L^AT_EX die Trennung fast vollständig aus. Somit ergeben die letztgenannten Umgebungen einen sehr unruhigen Satz.

Listen:	<pre> 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} </pre>
---------	---

Abbildung 1: Beispiel für `itemize`

Nummerierte Listen:	<pre> Nummerierte Listen: \begin{enumerate} \item Bei \texttt{enumerate} werden die Elemente ... \item Die Nummerierung ... \item Listen kann man auch verschachteln: \begin{enumerate} \item Die maximale ... \item Bezeichnung und ... \end{enumerate} \item usw. \end{enumerate} </pre>
---------------------	---

Abbildung 2: Beispiel für `enumerate`

Kleine Tierkunde:	<code>Kleine Tierkunde:</code>
Gelse: ein kleines Tier, das ...	<code>\begin{description}</code>
Gemse: ein großes Tier, das ...	<code>\item[Gelse:]</code>
Gürteltier: ein mittelgroßes Tier, das ...	<code>ein kleines Tier, das ...</code>
	<code>\item[Gemse:]</code>
	<code>ein großss es Tier, das ...</code>
	<code>\item[Gürteltier:]</code>
	<code>ein mittelgro"ses Tier, das ...</code>
	<code>\end{description}</code>

Abbildung 3: Beispiel für `description`

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an.	<code>% \usepackage{ragged2e}</code>
	<code>\begin{FlushLeft}</code>
	Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an.
	<code>\end{FlushLeft}</code>

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 L^AT_EX-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.

Der <code>\dots</code> -Befehl ...	Der <code>\verb \dots </code> -Befehl <code>\dots</code>
------------------------------------	--

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. Entgegen der im nächsten Abschnitt vorgestellten Umgebungen `array` und `tabular` erlaubt die `tabbing`-Umgebung einen Seitenumbruch.

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>\> \> Teddybär</code>
			<code>\end{tabbing}</code>

3.9.6. Tabellen (`tabular`)

Tabellen lassen sich mit der `tabular`-Umgebung erzeugen. Dort kann man mit einer Tabellenpräambel bestimmen, wie die Spalten dargestellt werden. Die Spaltenbreite passt sich automatisch dem Inhalt der Tabelle an und muss nicht (außer bei p-Spalten) angegeben werden.

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, `\` oder `\tabularnewline` trennt die Zeilen, `\hline` (an Stelle einer Zeile) setzt einen waagrechten Strich. Anstelle von `\hline` bietet das Paket `booktabs` unterschiedliche Strichstärken, um den Tabellenkopf vom Tabellenkörper deutlicher zu trennen: `\toprule`, `\midrule` und `\bottomrule`.

		<code>% \usepackage{booktabs}</code>
		<code>\begin{tabular}[t]{rl}</code>
		<code>\toprule</code>
	Wert & Zahlensystem	<code>Wert & Zahlensystem \\</code>
		<code>\midrule</code>
	7C0 hexadezimal	<code>7C0 & hexadezimal \\</code>
	3700 oktal	<code>3700 & oktal \\</code>
	11111000000 binär	<code>11111000000 & binär \\</code>
	1984 dezimal	<code>1984 & dezimal \\</code>
		<code>\bottomrule</code>
		<code>\end{tabular}</code>

4. Setzen von mathematischen Formeln

4.1. Allgemeines

L^AT_EX verfügt über einen eigenen Modus für mathematische Formeln, der sich in einigen Punkten deutlich vom Textmodus unterscheidet. Formeln können sowohl im Fließtext als auch als eigenständige Elemente auf der Seite vorkommen. Trotz der unterschiedlichen Darstellung unterscheidet sich die Eingabe der Formeln nur wenig voneinander.

Die Fülle der Möglichkeiten ist u. a. in der Dokumentation „mathmode“ zusammengefasst und soll hier als Referenz dienen [?].

Obwohl L^AT_EX ohne Zusatzpakete viele Möglichkeiten bietet, wird häufig das Paket `amsmath` eingebunden. Es erweitert die in L^AT_EX vorhandenen Möglichkeiten um eine Vielzahl von Makros und Umgebungen. Im weiteren Verlauf wird daher auch vorausgesetzt, dass das Paket geladen ist. In dem genannten Dokument mit dem Titel „mathmode“ wird `amsmath` ausführlich beschrieben.

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).

T_EX spricht man wie $\tau\epsilon\chi$ aus.
Mit ♡-lichen Grüßen

`\TeX` spricht man wie
`$\tau\epsilon\chi$` aus.
Mit `\heartsuit`-lichen
Grü`ss`en

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 \dots
```

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 `\text{...}`² 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}
```

²Durch `amsmath` bereitgestellt

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 Anhang B auf Seite 52.

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`

Des weiteren gibt es eine Vielzahl von **mathematischen Symbolen**: von $\in$ über $\Rightarrow$ bis ∞ (siehe Anhang B auf Seite 52).

Neben der voreingestellten Kursivschrift für die Variablen bietet L^AT_EX 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 `amssymb` [?] stehen auch Alphabete für Mengenzeichen und Frakturschrift zur Verfügung. Lokal können noch weitere installiert sein.

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

$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$ `\[`
`a_{1} \quad \quad x^{2} \quad \quad e^{-\alpha t} \quad \quad a_{ij}^3`
`\]`

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

$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$ `\[`
`\sqrt{x} \quad \quad \sqrt{x^2 + \sqrt{y}} \quad \quad \sqrt[3]{2}`
`\]`

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

$$\overline{m+n} \quad \$\overline{\text{m+n}}\$$$

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

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

Um mathematische **Akzente** wie Pfeile oder Schlangen auf Variablen zu setzen, gibt es die in Tabelle 12 auf Seite 52 angeführten Befehle. Längere Tilden und Dacherln, die sich über mehrere (bis zu 3) Zeichen erstrecken können, erhält man mit `\widetilde` bzw. `\widehat`. Ableitungszeichen werden mit ' (Apostroph) eingegeben.

$$y = x^2 \quad y' = 2x \quad y'' = 2 \quad \begin{array}{l} \backslash[\\ y=x^{\text{2}} \quad \backslash\!\!\backslash\!\!\backslash y'=2x \quad \backslash\!\!\backslash\!\!\backslash y''=2 \\ \backslash] \end{array}$$

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:

```
\begin{tabular}{@{}*{8}{>{\$}c<{\$}}@{}}
\arccos & \cos & \csc & \exp & \ker & \limsup & \min & \sinh \\
\arcsin & \cosh & \deg & \gcd & \lg & \ln & \Pr & \sup \\
\arctan & \cot & \det & \hom & \lim & \log & \sec & \tan \\
\arg & \coth & \dim & \inf & \liminf & \max & \sin & \tanh \\
\end{tabular}
```

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

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

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

```
\[
\lim_{x \to 0} \frac{\sin x}{x}
=1
\]
```

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

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

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

```
$1\frac{1}{2}$~Stunden
\[
\frac{x^{2}}{k+1} \qquad
x^{\frac{2}{k+1}} \qquad
x^{1/2}
\]
```

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

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

```
\[
{n \choose k} \qquad
{x \atop y+2}
\]
```

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.

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

```
\[
\sum_{i=1}^n \qquad
\int_0^{\frac{\pi}{2}} \qquad
\int \limits_{-\infty}^{+\infty}
\]
```

Für **Klammern** und andere Begrenzer gibt es in TeX viele verschiedene Symbole (z.B. `[` `<` `||` `\uparrow`). Runde und eckige Klammern können mit den entsprechenden Tasten eingegeben werden, geschwungene mit `\{`, die anderen mit speziellen Befehlen (z.B. `\updownarrow`).

Setzt man den Befehl `\left` vor öffnende Klammern und den Befehl `\right` vor schließende, so wird automatisch die richtige Größe gewählt.

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

```
\[
1 + \left( \frac{1}{1-x^2} \right)^3
\]
```

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

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

```
\[
\Bigl( (x+1) (x-1) \Bigr)^2
\]
```

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

$$x_1, \dots, x_n \qquad x_1 + \cdots + x_n$$

```
\[
x_{\{1\}}, \ldots, x_{\{n\}} \quad \quad \quad x_{\{1\}} + \cdots + x_{\{n\}}
\]
```

4.3. Nebeneinander Setzen

Wenn man mit den von T_EX gewählten **Abständen** innerhalb von Formeln nicht zufrieden ist, kann man sie mit expliziten Befehlen verändern. Die wichtigsten sind `\,` für einen sehr kleinen Abstand, `\;` für einen mittleren, `\quad` und `\qquad` für große Abstände sowie `\!` für die Verkleinerung eines Abstands. Die Vergrößerung des Abstandes durch die Anweisungen `\quad` und `\qquad` ist gleichbedeutend zu den Breiten 1 em und 2 em (siehe 8.1.2 auf Seite 46).

$$F_n = F_{n-1} + F_{n-2} \qquad n \geq 2$$

```
\[
F_{\{n\}} = F_{\{n-1\}} + F_{\{n-2\}}
\qquad n \geq 2
\]
```

$$\iint_D dx \, dy \quad \text{statt} \quad \int \int_D dx dy$$

```
\[
\int\!\!\int\!\!\int_D \mathrm{d}x \, \mathrm{d}y \quad \text{statt} \quad \int \int_D dx dy
\]
```

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.

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

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

Für **mehrzeilige** Formeln oder Gleichungssysteme bietet L^AT_EX die Umgebung `eqnarray` an. Auf Grund verschiedener Defizite dieser Umgebung sei an dieser Stelle nur gesagt, dass diese Umgebung nicht genutzt werden soll. Das in der Einleitung zu diesem Kapitel genannte Paket `amsmath` bietet die wesentlich verbesserte Umgebung `align`.

Bei `align` erhält jede Zeile eine eigene Gleichungsnummer, bei `align*` wird ebenso wie bei `\[...\]` *keine* Gleichungsnummer hinzugefügt.

Für Gleichungssysteme, die *eine* gemeinsame Gleichungsnummer erhalten sollen, kann auf die `array`-Umgebung oder die durch `amsmath` bereitgestellte Umgebung `gather` zurückgegriffen werden.

Die Umgebungen `align` arbeitet nach dem Prinzip einer Tabelle, wobei `&` als Ausrichtungssymbol verwendet wird. Entgegen einer Tabelle, muss die Anzahl der Spalten nicht vorher angegeben werden. Das Ausrichtungssymbol sollte stets *vor* dem Relationszeichen stehen, so dass die korrekten Abstände ermittelt werden können.

Der Befehl `\\` trennt die Zeilen.

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

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

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 folgende Variante:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

Die Dokumentation mit dem Titel „mathmode“ wurde bereits erwähnt, soll aber an dieser Stelle noch einmal ins Gedächtnis gerufen werden. Neben Zahlreichen Beispielen kann die Dokumentation als Nachschlagewerk für den mathematischen Satz genutzt werden.

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. Das häufig benutzte PDF_TE_X kann Grafiken in den Formaten PDF, PNG JPEG und JBIG2 verarbeiten.



Hier ist ein Bild.

```
Hier \includegraphics  
      [width=3cm]{ozean}  
ist ein Bild.
```

Wird das Paket `graphicx` mit der Option `[draft]` geladen, dann erscheint anstelle des Bildes nur ein Rahmen entsprechend der tatsächlichen Bildgröße mit dem Namen des Grafikfiles, was die Bearbeitung beschleunigt und für Probeausdrucke nützlich ist.

Weitere Informationen zum Einbinden von Bildern finden Sie in der Online-Dokumentation [\[?\]](#), im *Graphics Companion* [\[?\]](#) und in K. Reckdahls empfehlenswertem Tutorium [\[?\]](#).

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* [?] ist angegeben, wie man das Aussehen der Kopf- und Fußzeilen außerdem mit dem Seitenstil `myheadings` und den Befehlen `\markboth`, `\markright` und `\pagenumbering` beeinflussen kann. Zur Gestaltung der Kopf- und Fußzeilen stehen die Pakete `scrpage2` oder `fancyhdr` zur Verfügung, die dem Nutzer die Anpassungen erleichtern.

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 Nummerierung nicht! Wie bereits in der Einführung zum Inhaltsverzeichnis erläutert, benötigt L^AT_EX mindestens zwei Durchläufe für das korrekte setzen der Nummern und des Verweises.

Im folgenden Beispiel wird einfach mit dem Befehl `\vspace` (siehe Abschnitt 8.1.3) leerer Raum für ein später einzusetzendes Bild gelassen:

Abbildung 4 auf S. 39 zeigt ein Beispiel aus der Minimal art. Abbildung 4: Landschaft im Nebel	<pre> Abbildung~\ref{weiss} auf S.~\pageref{weiss} zeigt ein Beispiel aus der Minimal art. \begin{figure}[!htb] \centering \vspace*{1cm} \caption{Landschaft im Nebel} \label{weiss} \end{figure> </pre>
--	---

L^AT_EX kann eine Abbildung nach verschiedenen Kriterien platzieren: **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 Platzierung der Abbildung auf die angegebenen Orte *einzuschränken*. Durch Angabe von z. B. **tb** wird L^AT_EX angewiesen, nur eine Platzierung oben oder unten auf der Seite zu versuchen, je nachdem, wo *zuerst* eine passende Stelle gefunden wird. Werden keine Parameter (und keine eckigen Klammern!) angegeben, ist die Voreinstellung **tbp**, also ohne **h**.

Eine Platzierungsbeschränkung *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.

Eine Abbildung, die nicht platziert werden konnte, wird von L^AT_EX 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 Platzierungsparameter, **!** (bang), der L^AT_EX anweist, gewisse eingebaute Beschränkungen zu ignorieren, z. B., dass bei der Platzierung 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* [?] und in der Online-Dokumentation [?]. Für X_EL^AT_EX und LuaL^AT_EX steht das Paket `fontspec` zur Verfügung, mit dem beliebige OpenType und TrueType Schriftarten eingebunden werden können, ohne dass sie für L^AT_EX angepasst werden müssen.

```
\documentclass{article}
\usepackage[ngerman]{babel}
% Nur XeLaTeX und LuaLaTeX:
\usepackage{fontspec}
\setmainfont{Linux Libertine O}
\begin{document}
Das ist ein Text in der Schriftart Linux Libertine mit den Schnitten
\textit{Kursiv}, \textbf{\textit{Fett Kursiv}} und \textsc{Kapitälchen}.
\end{document}
```

Ergibt einen Text in der Schriftart Linux Libertine:

Das ist ein Text in der Schriftart Linux Libertine mit den Schnitten *Kursiv*, **Fett Kursiv** und KAPITÄLCHEN.

7.1. Schriftgrößen

Die in der Tabelle 6 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.

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.

Tabelle 6: 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

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 7 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.

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}
Römer
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. L^AT_EX verwendet als Voreinstellung die sog. Computer-Modern-Schriftfamilien (CM), siehe Tabelle 8; 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 8 führt einige derartige Pakete auf.

Die Dokumentation der T_EX-Distributionen 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 L^AT_EX-System verwendbar sein [?].

7.4. Die „europäischen“ Zeichensätze

L^AT_EX 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 L^AT_EX 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 T_EX 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 [?] 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.

Tabelle 7: 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

Tabelle 8: 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	≈ CM Roman
<code>courier</code>			Courier	
<code>droid</code>	Droid Serif *	Droid Sans	Droid Sans Mono	
<code>fourier</code>	Utopia Regular *			Fourier
<code>helvet</code>		Helvetica		
<code>inconsolata</code>			Inconsolata	
<code>libertine</code>	Linux Libertine *	Linux Biolinum		
<code>lmodern</code>	LM Roman *	LM Sans Serif	LM Typewriter	≈ LM Roman
<code>mathptmx</code>	Times *			≈ Times
<code>mathpazo</code>	Palatino *			≈ Palatino

8. Spezialitäten

Das komplette Menü der Spezialitäten, die von L^AT_EX serviert werden, ist im *L^AT_EX-Handbuch* [?] 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 L^AT_EX 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}
```

Häufig wird ein anderthalbfacher Zeilenabstand gewünscht, wobei bspw. Fußnoten ausgenommen sein sollen. Der Befehl `\linespread` macht diese Unterscheidung nicht. Zur Änderung des Zeilenabstandes sollte daher stets auf das Paket `setspace` zurückgegriffen werden.

8.1.2. Spezielle horizontale Abstände

Die Abstände zwischen Wörtern und Sätzen werden von L^AT_EX automatisch gesetzt. Sonstigen horizontalen Abstand kann man mit den Befehlen

```
\hspace{länge}  
\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 9 angeführt.

Die Befehle in Tabelle 10 sind Abkürzungen zum Einfügen besonderer horizontaler Abstände.

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

Tabelle 9: Einheiten für Längenangaben

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

Tabelle 10: 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>\qquad</code>	doppelt so breit wie ein <code>\quad</code>
<code>\hfill</code>	ein Abstand, der sich von 0 bis ∞ ausdehnen kann.

Schafft mir	Raum!	<code>\raggedright</code>
$\triangleleft$	$\triangleright$	Schafft mir <code>\hspace{1.5cm}</code> Raum! <code>\\</code>
		<code>\$\triangleleft\$\hfill</code>
		<code>\$\triangleright\$</code>

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

`\vspace{länge}`

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 11 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:

Tabelle 11: 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

`\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. 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 Nummerierung 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 ...

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}
```

Werden viele Literatureinträge zitiert bzw. verwendet, bietet sich die Nutzung einer Datenbank an. Die Datenbank besitzt ihre eigene Syntax, um die benötigten Literatureinträge zu verwalten. Für die komfortable Verwaltung von Literaturdatenbanken existieren viele Programme wie beispielsweise JabRef (frei) oder Endnote (kommerziell). Die Datenbank ist im eigentlichen Sinne eine Textdatei mit Endung `bib`.

Für die Verarbeitung dieser Literaturdatenbanken bieten sich zwei verschiedene Hilfsmittel für L^AT_EX an. Die klassische Variante ist *BibT_EX* in Verbindung mit einem Literaturverzeichnisstil. Die Anpassung an die eigenen Bedürfnisse

gestaltet sich mehr als schwierig. Daher wurde in den letzten Jahren das L^AT_EX Makropaket *biblatex* entwickelt, das alternativ zu *BibT_EX* das mächtigere Programm *biber* nutzen kann. Das Makropaket *biblatex* erlaubt die Manipulation des Literaturverzeichnisses auf L^AT_EX-Ebene. Auf CTAN ist eine deutsche Übersetzung der Dokumentation verfügbar [?].

A. Mit dem Paket textcomp verfügbare Symbole³

,	<code>\textquotestraightbase*</code>	„	<code>\textquotestraightdblbase*</code>
—	<code>\texttwelveudash*</code>	—	<code>\textthreequartersemdash*</code>
←	<code>\textleftarrow</code>	→	<code>\textrightarrow</code>
␣	<code>\textblank</code>	\$	<code>\\$*</code>
'	<code>\textquotesingle*</code>	*	<code>\textasteriskcentered*</code>
=	<code>\textdblhyphen</code>	/	<code>\textfractionsolidus*</code>
<	<code>\textlangle</code>	—	<code>\textminus*</code>
>	<code>\textrangle</code>	∪	<code>\textmho</code>
○	<code>\textbigcircle</code>	Ω	<code>\textohm</code>
⌈	<code>\textlbrackdbl</code>	⌋	<code>\textrbrackdbl</code>
↑	<code>\textuparrow</code>	↓	<code>\textdownarrow</code>
`	<code>\textasciigrave*</code>	★	<code>\textborn</code>
∅	<code>\textdivorced</code>	†	<code>\textdied</code>
🍃	<code>\textleaf</code>	∞	<code>\textmarried</code>
♪	<code>\textmusicalnote</code>	~	<code>\texttildelow*</code>
=	<code>\textdblhyphenchar</code>	˘	<code>\textasciibreve*</code>
˘	<code>\textasciicaron*</code>	”	<code>\textacutedbl*</code>
“	<code>\textgravedbl*</code>	†	<code>\dag*</code>
‡	<code>\ddag*</code>	‖	<code>\textbardbl*</code>
‰	<code>\textperthousand*</code>	•	<code>\textbullet*</code>
°C	<code>\textcelsius*</code>	\$	<code>\textdollaroldstyle</code>
c	<code>\textcentoldstyle</code>	f	<code>\textflorin*</code>
Ⓒ	<code>\textcolonmonetary</code>	₩	<code>\textwon</code>
₦	<code>\textnaira</code>	₧	<code>\textguarani</code>
₱	<code>\textpeso</code>	₯	<code>\textlira</code>
℞	<code>\textrecipe</code>	‽	<code>\textinterrobang</code>
‽	<code>\textinterrobangdown</code>	₫	<code>\textdong</code>
™	<code>\texttrademark*</code>	‰	<code>\textpertenthousand</code>
¶	<code>\textpilcrow</code>	฿	<code>\textbaht</code>
№	<code>\textnumero</code>	%	<code>\textdiscount</code>
ℰ	<code>\textestimated</code>	◦	<code>\textopenbullet</code>
SM	<code>\textservicemark</code>	{	<code>\textlquill</code>
}	<code>\textrquill</code>	¢	<code>\textcent*</code>
£	<code>\pounds*</code>	₪	<code>\textcurrency*</code>
¥	<code>\textyen*</code>		<code>\textbrokenbar*</code>
§	<code>\S*</code>	¨	<code>\textasciidieresis*</code>
©	<code>\copyright*</code>	ª	<code>\textordfeminine*</code>
Ⓒ	<code>\textcopyleft</code>	¬	<code>\textlnot*</code>
Ⓓ	<code>\textcircledP</code>	®	<code>\textregistered*</code>
—	<code>\textasciimacron*</code>	°	<code>\textdegree*</code>
±	<code>\textpm*</code>	²	<code>\texttwosuperior</code>
³	<code>\textthreesuperior</code>	´	<code>\textasciacute*</code>

³Schriften, die nicht speziell für die Verwendung mit T_EX entworfen wurden, enthalten normalerweise nur die mit * markierten Zeichen.

μ	<code>\textmu*</code>	$\P$	<code>\P*</code>
$\cdot$	<code>\textperiodcentered*</code>	‰	<code>\textreferencemark</code>
1	<code>\textonesuperior</code>	♂	<code>\textordmasculine*</code>
$\sqrt{}$	<code>\textsurd</code>	$\frac{1}{4}$	<code>\textonequarter</code>
$\frac{1}{2}$	<code>\textonehalf</code>	$\frac{3}{4}$	<code>\textthreequarters</code>
€	<code>\textsf{\texteuro}</code>	$\times$	<code>\texttimes*</code>
$\div$	<code>\textdiv*</code>		

B. 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. Mit den Paketen `amssymb`, `mathrsfs` oder `wasysym` stehen weitere Zeichen zur Verfügung. Die in einer T_EX-Distribution üblicherweise vorhandene Übersicht The Comprehensive L^AT_EX Symbol List [?] zeigt viele Symbole und wie sie mit L^AT_EX zu erreichen sind.

Tabelle 12: 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 13: 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 14: 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 15: Verschiedene sonstige Symbole (* benötigt Paket `latexsym`)

$\aleph$	<code>\aleph</code>	$\prime$	<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>	$\bot$	<code>\bot</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 16: „Große“ Operatoren

Σ	$\sum$	<code>\sum</code>	$\cap$	$\bigcap$	<code>\bigcap</code>	$\odot$	$\bigodot$	<code>\bigodot</code>
$\prod$	$\prod$	<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 17: 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>	$\uplus$	<code>\uplus</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>	$\trianglelefteq$	<code>\unlhd^*</code>	$\amalg$	<code>\amalg</code>
$\bigcirc$	<code>\bigcirc</code>	$\trianglerighteq$	<code>\unrhd^*</code>	$\wr$	<code>\wr</code>
$\triangle$	<code>\bigtriangleup</code>	∇	<code>\bigtriangledown</code>		

Tabelle 18: 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>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupset$	<code>\sqsupset</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 19: Negierte Relationen

$\not<$	<code>\not<</code>	$\not>$	<code>\not></code>	$\neq$	<code>\neq</code>
$\not\leq$	<code>\not\leq</code>	$\not\geq$	<code>\not\geq</code>	$\not\equiv$	<code>\not\equiv</code>
$\not\prec$	<code>\not\prec</code>	$\not\succ$	<code>\not\succ</code>	$\not\sim$	<code>\not\sim</code>
$\not\preceq$	<code>\not\preceq</code>	$\not\succeq$	<code>\not\succeq</code>	$\not\simeq$	<code>\not\simeq</code>
$\not\subset$	<code>\not\subset</code>	$\not\supset$	<code>\not\supset</code>	$\not\approx$	<code>\not\approx</code>
$\not\subseteq$	<code>\not\subseteq</code>	$\not\supseteq$	<code>\not\supseteq</code>	$\not\cong$	<code>\not\cong</code>
$\not\sqsubset$	<code>\not\sqsubset</code>	$\not\sqsupset$	<code>\not\sqsupset</code>	$\not\bowtie$	<code>\not\bowtie</code>

Tabelle 20: 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>
$\Leftarrow$	<code>\Leftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\rightarrow$	<code>\rightarrow</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</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>\right...</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>			$\leadsto$	<code>\leadsto</code> *

Tabelle 21: 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>		