

Erstes Praxissemester:  
Foo-Pool, FH-Furtwangen

Textverarbeitung mit L<sup>A</sup>T<sub>E</sub>X am Beispiel der  
Foo-Bibel

Torben Nehmer

*Torben.Nehmer@gmx.net, <http://www.nathan-syntronics.de>*

Erstellt am 6. Februar 2000

## Inhaltsverzeichnis

<b>1</b>	<b>Das Praxissemester im Überblick</b>	<b>3</b>
1.1	LaTeX und die Foo-Bibel . . . . .	3
1.2	Ziel dieser Seminararbeit . . . . .	3
1.3	Weiterführende Literatur . . . . .	4
<b>2</b>	<b>Geschichte und Philosophie von LaTeX</b>	<b>5</b>
2.1	Wie ist LaTeX entstanden? . . . . .	5
2.2	Woher ist LaTeX zu beziehen? . . . . .	5
2.3	Von der Eingabe zum Dokument . . . . .	5
2.4	Warum LaTeX? . . . . .	6
<b>3</b>	<b>LaTeX-Grundlagen</b>	<b>8</b>
3.1	Die Eingabedatei . . . . .	8
3.2	Grundlagen . . . . .	8
3.3	Das Dokument . . . . .	9
3.4	Gliederungsbefehle . . . . .	11
3.5	LaTeX und seine Hilfsprogramme . . . . .	12
<b>4</b>	<b>Fazit</b>	<b>14</b>
	<b>Literatur</b>	<b>15</b>

---

# 1 Das Praxissemester im Überblick

Ich habe vom 1. März 1999 bis zum 31. August 1999 mein Praxissemester im Foo-Pool der FH-Furtwangen absolviert. Haupt-Grund für diese Entscheidung war es, ein heterogenes Netzwerk auf UNIX-Basis kennenzulernen. Im Gegensatz zu den meisten anderen Netzen der FH Furtwangen ist der Foo-Pool dezentral organisiert. Dies bedeutet, daß die verschiedenen Dienste nicht alle auf einem oder zwei zentralen Servern liegen, sondern auf den verschiedenen Rechnern im Pool verteilt sind. Diese Konfiguration ermöglicht eine hohe Ausfall-Sicherheit, da der Ausfall eines Rechners nicht mehrere vitale Dienste auf einmal betrifft. Weiterhin spricht die bessere Auslastung der vorhandenen Ressourcen für ein dezentrales Netz. Herkömmliche Workstations (vor allem auch Intel PCs) besitzen heute relativ große Leistungsreserven, durch die diese Dezentralisierung erst ihr ganzes Leistungsspektrum entfalten kann. Eine Dezentralisierung dieser Art birgt natürlich einige Hürden, sowohl auf der Seite der Administration und auf der Seite der Nutzer. Ziel ist es, eine Umgebung dieser Art für die Nutzer so transparent wie möglich zu halten.

Meine Aufgaben liessen sich in mehrere Bereiche untergliedern: Der grösste Punkt war die Erstellung der Foo-Bibel, die seit diesem Semester als Einsteigerwerk für den Foo-Pool dient. Ein weiteres Projekt war es, die Benutzeroberfläche zu aktualisieren und weiter zu vereinheitlichen. Dazu kamen verschiedene Aufgaben der allgemeinen System-Administration, wie beispielsweise die Betreuung der Nutzer des Foo-Pools oder die Aktualisierung der Webserver-Software.

## 1.1 $\LaTeX$ und die Foo-Bibel

Der Schwerpunkt dieser Seminararbeit liegt in den Möglichkeiten, die  $\LaTeX$  als Textsatzsystem bietet. Es ist meiner Meinung nach für Studenten besonders interessant, da es einerseits ein wirklich plattformunabhängiges Format anbietet und andererseits äusserst stabil läuft. Die Foo-Bibel beispielsweise besteht aus über 60 Seiten, bei denen viele herkömmliche Textverarbeitungs-Systeme oft schon die Flügel strecken, während  $\LaTeX$  sie ohne Probleme selbst auf den etwas älteren Ultrix-Systemen des Foo-Pools verarbeitet. Und das Ergebnis braucht einen Vergleich mit professionellen Produktionen nicht zu scheuen.

## 1.2 Ziel dieser Seminararbeit

Dieser Text soll Ihnen  $\LaTeX$  als Alternative zu den gebräuchlichen Office-Textverarbeitungen näherbringen. Die wichtigsten Eigenschaften von  $\LaTeX$  werden anhand von Beispielen kurz erläutert. Leider kann auf so ein riesiges Paket wie  $\LaTeX$  in einer Seminararbeit nicht detailliert eingegangen werden. Ich beschränke mich deshalb auf die Konzepte, denen  $\LaTeX$  zugrundeliegt.

Im ersten Teil der Arbeit gehe ich zunächst auf die Geschichte und die Philosophie von  $\LaTeX$  ein. Danach möchte ich einige der grundlegenden Hilfsmittel zum Erstellen eines Dokuments ansprechen. Weiterhin möchte ich einen kleinen Einblick in die speziellen Funktionen für größere Dokumente geben. Ab-

gerundet wird dies durch einen kurzen Überblick über einige Hilfsprogramme rund um  $\LaTeX$ .

### **1.3 Weiterführende Literatur**

Sollten Sie nach diesem Dokument auf den Geschmack von  $\LaTeX$  gekommen sein, kann ich Ihnen besonders das  $\LaTeX$  Handbuch[4] empfehlen, welches sich sowohl als Einsteigerwerk als auch als Referenz eignet. Der  $\LaTeX$  Begleiter behandelt fortgeschrittene Themengebiete rund um  $\LaTeX$ . Sollten Sie  $\LaTeX$  regelmässig verwenden, sollten Sie sich die Kurz-Referenz  $\LaTeX$  kurz & gut[1] einmal näher ansehen.

---

## 2 Geschichte und Philosophie von L<sup>A</sup>T<sub>E</sub>X

### 2.1 Wie ist L<sup>A</sup>T<sub>E</sub>X entstanden?

L<sup>A</sup>T<sub>E</sub>X (gesprochen „latech“, das X steht für den griechischen Buchstaben Chi) entstand 1985 als Erweiterung zu Donald Knuth's Textsatzprogramm T<sub>E</sub>X. Dieses wurde von ihm zur Erstellung von wissenschaftlichen Texten konzipiert und ist frei erhältlich. Allerdings hat T<sub>E</sub>X einen gravierenden Nachteil: Seine Verwendung ist ein wenig komplex, so dass eine lange Einarbeitungszeit notwendig ist. Um dieses Problem aus dem Weg zu schaffen, wurde T<sub>E</sub>X von Leslie Lamport um das Makropaket L<sup>A</sup>T<sub>E</sub>X ergänzt. L<sup>A</sup>T<sub>E</sub>X verbirgt die T<sub>E</sub>X-Kommandos vor dem Benutzer und stellt Befehle für alle nur denkbaren Anwendungszwecke zur Verfügung. Heutzutage werden die elementaren T<sub>E</sub>X-Befehle in der Regel nur zur Erweiterung von L<sup>A</sup>T<sub>E</sub>X benötigt.

L<sup>A</sup>T<sub>E</sub>X ist in der aktuellen Version 2<sub>ε</sub> genauso fehlerfrei wie T<sub>E</sub>X. Schon seit Jahren wurden beispielsweise in T<sub>E</sub>X keine Fehler mehr gefunden. Werte, von denen Programme wie M\$ Word nur träumen können.

### 2.2 Woher ist L<sup>A</sup>T<sub>E</sub>X zu beziehen?

Die Deutschsprachige Awendervereinigung T<sub>E</sub>X e.V. stellt auf ihrem Web-Server[2] umfangreiche Informationen über L<sup>A</sup>T<sub>E</sub>X zur Verfügung. Hier sind unter anderem auch Verweise zu Distributionen für (Win)DOS, UNIX und Macintosh. Weiterhin gehört L<sup>A</sup>T<sub>E</sub>X mitsamt allen notwendigen Hilfsprogrammen zum Standardumfang aller großen Linux-Distributionen<sup>1</sup>.

### 2.3 Von der Eingabe zum Dokument

Wie wird ein gedrucktes Dokument erstellt? Der Autor übergibt ein getipptes Manuskript dem Verlag. Ein Designer legt die typographische Gestaltung und das allgemeine Layout, also das endgültige Erscheinungsbild des Dokuments, fest. Danach wird die eigentliche Druckvorlage vom Setzer erstellt. Dieser entscheidet, an welcher Stelle einer Seite die Worte und Symbole des Autors erscheinen.

Für Sie übernimmt L<sup>A</sup>T<sub>E</sub>X die typographische Gestaltung, das Design. T<sub>E</sub>X wiederum agiert als L<sup>A</sup>T<sub>E</sub>X's Setzer. Ihre Anweisungen werden von L<sup>A</sup>T<sub>E</sub>X in elementare T<sub>E</sub>X-Befehle übersetzt. T<sub>E</sub>X erstellt daraus dann eine geräteunabhängige Datei. Diese DVI-Datei (*device-independent*) kann dann mittels verschiedener Hilfsprogramme in andere Formate, beispielsweise PostScript, umgewandelt werden.

Die Gestaltung eines Textes soll die Ideen des Autors verdeutlichen, so daß das Dokument leichter lesbar ist. Die logische Struktur soll durch die optische Struktur ergänzt werden. Für diese Gestaltung ist es allerdings essentiell wichtig, die logische Struktur des Dokuments zu verstehen. Woher weiß L<sup>A</sup>T<sub>E</sub>X nun,

---

<sup>1</sup>Zu fast allen aktuellen Linux-Distributionen wird die freie teT<sub>E</sub>X Distribution[3] hinzugepackt, mit der auch diese Arbeit erstellt wurde.

wie es ein Dokument formatieren soll. Da  $\LaTeX$  weder Englisch noch Deutsch kann, ist es nicht in der Lage, solche Entscheidungen selbstständig zu treffen. Aus diesem Grunde müssen diese Strukturelemente  $\LaTeX$  durch bestimmte Befehle zu verstehen gegeben werden. Fast alle  $\LaTeX$ -Anweisungen dienen dazu, diese logische Struktur zu beschreiben. Beispielsweise existieren Kommandos für Zitate und Computerprogramme.

Dies ist die eigentliche Stärke von  $\LaTeX$ , die es von den herkömmlichen Textverarbeitungen unterscheidet: Der Autor soll sich nicht auf die typographische Gestaltung des Textes konzentrieren, sondern sich lediglich mit der logischen Strukturierung des Dokumentes befassen. Sie konzentrieren sich also auf den Inhalt des Dokumentes und nicht auf die visuelle Gestaltung.

## 2.4 Warum $\LaTeX$ ?

Als  $\LaTeX$  1985 erschien, galt es als revolutionär, da nur wenige Autoren ihre Dokumente selbst setzen konnten. Heute gehört Desktop-Publishing zum Alltag. „WYSIWYG“-Programme<sup>2</sup> haben sich etabliert. Sie zeigen Ihnen das fertige Dokument bereits während des Bearbeitens. Warum also  $\LaTeX$  verwenden, bei dem Sie für jede Passage angeben müssen, um was es sich denn nun handelt, wenn Sie es in einem WYSIWYG-Programm setzen können, wie es Ihnen gerade in den Sinn kommt?

Um den Unterschied der beiden Konzepte einmal zu verdeutlichen möchte ich auf einen kleinen Auszug aus einer Beispieldatei<sup>3</sup> der Linux beiliegenden  $\TeX$ -Distribution kurz eingehen. In dieser taucht der mathematische Ausdruck  $\langle A, B \rangle$  auf. In einem WYSIWYG-Programm geben Sie in der Regel direkt  $\langle A, B \rangle$  ein. Dies wäre in  $\LaTeX$  ohne weiteres ebenfalls möglich. Benötigt man diese mathematische Struktur – das innere Produkt von  $A$  und  $B$  – jedoch mehrmals, wird der routinierte  $\LaTeX$ -Benutzer ein eigenes Makro für diese Anwendung schreiben. `sample2e.tex` definiert am Anfang das Makro `\ip`, das genau diese Funktion übernimmt. `\ip{A}{B}` ergibt dadurch  $\langle A, B \rangle$ .

Die Vorteile des  $\LaTeX$ -Konzepts werden deutlich, wenn das innere Produkt plötzlich in der Form  $\langle A|B \rangle$  dargestellt werden soll. In einer herkömmlichen Textverarbeitung muß nun das komplette Dokument durchsucht werden. Jedes einzelne innere Produkt muß von Hand ersetzt werden. In  $\LaTeX$  genügt es, die Definition des (eigenen) Makros `\ip` zu ändern. Der geringfügig größere Tippaufwand wird also nicht nur durch die einheitliche Struktur sondern auch durch einfachere Änderungsmöglichkeiten mehr als wettgemacht.

In allen mathematisch auch nur angehauchten Texten nimmt Ihnen  $\LaTeX$ 's unschlagbarer Formelsatz ( $\pi \sum_{i=1}^n \frac{a_i x^i}{i}$ ) sehr viel Arbeit ab. Keine andere, mir bekannte Software beherrscht mathematische Konstrukte so ausgereift, wie  $\LaTeX$ .

Nicht zuletzt dürfen Sie nicht die Zeit unterschätzen, die Sie durch  $\LaTeX$ 's Konzept der logischen Textstrukturierung gewinnen. Sie können sich voll auf den zu schreibenden Text konzentrieren und werden nicht durch überflüssiges Brimborium abgelenkt.

<sup>2</sup>„What you see is what you get“ – man bekommt, was man sieht

<sup>3</sup>Es handelt sich hier um die Datei `sample2e.tex`.

Wie bereits in der Einleitung kurz angesprochen wurde, sind die Ergebnisse, die  $\LaTeX$  produziert exzellent. Dafür spricht, dass der Verlag Addison Wesley viele seiner Bücher mit  $\LaTeX$  erstellt, während die Bank 1822 direkt  $\LaTeX$  für ihre Korrespondenz mit ihren Kunden (Kontoeröffnungen, Vergabe von PIN-Nummern u.ä.) verwendet.

Für Programmierer aus allen möglichen Sachgebieten kann  $\LaTeX$  eine enorme Arbeitserleichterung darstellen: Anstelle eine PostScript-Datei selbst zu erzeugen und sich mit vielen kleinen Details herumzuärgern, generiert man eine  $\LaTeX$ -Quelle und lässt diese übersetzen und drucken.

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

### 3.1 Die Eingabedatei

Als Eingabe für L<sup>A</sup>T<sub>E</sub>X dient eine Textdatei, die mit jedem gewöhnlichen Editor erstellt werden kann. In der Regel besitzen L<sup>A</sup>T<sub>E</sub>X-Quellen die Endung `.tex`. In dieser Eingabedatei sind alle Informationen vorhanden, die L<sup>A</sup>T<sub>E</sub>X zur Erstellung des Dokumentes benötigt.

Welchen Editor hierfür verwendet wird, liegt allein am persönlichen Geschmack des Benutzers. Der in der Unix-Welt sehr verbreitete Editor Emacs verfügt zum Beispiel über einen sehr leistungsfähigen L<sup>A</sup>T<sub>E</sub>X-Modus.

### 3.2 Grundlagen

Ein Großteil der L<sup>A</sup>T<sub>E</sub>X-Befehle beschreiben die logische Strukturierung des Textes. Als ersten Schritt in Richtung L<sup>A</sup>T<sub>E</sub>X sollte man die Art und Weise verstehen, in der L<sup>A</sup>T<sub>E</sub>X eine solche Struktur aufbaut. Grundsätzlich kann ein Dokument mehrere Strukturen mit unterschiedlicher Größe enthalten. Dies beginnt bei einzelnen Worten, die zu Absätzen zusammengefasst werden. Mehrere Absätze bilden wiederum einen Abschnitt. Das gesamte Dokument kann dann in Vorwort, Hauptteil und Anhänge aufgegliedert werden.

#### Sätze und Absätze

Den eigentlichen Text geben Sie wie gewohnt fortlaufend ein. Wichtig ist hier die Tatsache, dass Leerräume, egal aus wievielen Leerzeichen sie bestehen, grundsätzlich wie ein einziger Leerraum behandelt. Soll ein neuer Absatz beginnen, gibt man dies L<sup>A</sup>T<sub>E</sub>X durch eine oder mehrere Leerzeilen zu verstehen.

Wieviele Leerzeichen zwischen den Worten oder stehen, ist irrelevant.

Für diesen Absatz hätte auch eine Leerzeile genügt.

Wieviele Leerzeichen zwischen den Worten oder stehen, ist irrelevant.

Für diesen Absatz hätte auch eine Leerzeile genügt.

#### Befehle und Umgebungen

Jede L<sup>A</sup>T<sub>E</sub>X-Anweisung wird mit dem Backslash `\` eingeleitet. Dabei wird zwischen Befehlen und Umgebungen unterschieden. Befehle bestehen aus dem Backslash, dem Kommando und Parametern, die in die geschweiften Klammern `{ }` eingeschlossen werden. Umgebungen werden mit dem Kommando `\begin{name}` eingeleitet und enden mit `\end{name}`. Auf den hierdurch eingegrenzten Textteil werden die Modifikationen der Umgebung wirksam.

Umgebungen werden in der Regel für die Formatierung von einem oder mehreren Absätzen verwendet. Sie können so Zentrierte Bereiche, Zitate, Liste-



numgebungen, Tabellen und vieles mehr erzeugen. Befehle hingegen sind für kleinere Textpassagen gedacht. Hervorhebungen, Abschnittsüberschriften oder Fußnoten werden beispielsweise mit Befehlen erzeugt.

*Hervorgehobene Textteile* können leicht erzeugt werden. Der durch ... eingeschlossene Teil stellt hier das hervorzuhebende Element, den Parameter, dar.

```
\emph{Hervorgehobene Textteile} können leicht erzeugt werden. Der durch {...} eingeschlossene Teil stellt hier das hervorzuhebende Element, den Parameter, dar.
```

Auch zentrierte Umgebungen werden von  $\LaTeX$  beherrscht. Sie sind besonders für Abbildungen und Tabellen nützlich.

```
\begin{center}
Auch zentrierte Umgebungen werden von \LaTeX beherrscht. Sie sind besonders für Abbildungen und Tabellen nützlich.
\end{center}
```

### Fußnoten

Die in diesem Text bereits mehrfach verwendeten Fußnoten werden mit dem Befehl `\footnote` erzeugt, dessen Argument den Fußnoten-Text enthält. Dieser Befehl ist ein schönes Beispiel für eine komplexen  $\LaTeX$ -Anweisung. Sie muss zum einen für die automatische Numerierung sorgen, die zwischen Fußnoten innerhalb von Abbildungen und Fußnoten innerhalb des Fliesstextes unterscheidet und zum anderen die Fussnoten korrekt formatieren und plazieren. In Grösseren Dokumenten beginnt die automatische Numerierung in jedem Kapitel von neuem.

28.63 Punkt<sup>a</sup> entspricht in etwa einem Zentimeter.  $\LaTeX$  nutzt diese Masseinheit für alle Längenangaben, die während des Übersetzens ausgegeben werden.

<sup>a</sup>Punkt (pt) ist die Einheit der typographischen Spezialisten.

```
28.63 Punkt\footnote{Punkt (pt) ist die Einheit der typographischen Spezialisten.} entspricht in etwa einem Zentimeter. \LaTeX nutzt diese Masseinheit für alle Längenangaben, die während des Übersetzens ausgegeben werden.
```

## 3.3 Das Dokument

Bevor aus einer Eingabedatei ein Dokument erzeugt werden kann, muß zunächst die Art des Textes näher spezifiziert werden. Der eigentliche Textteil des Dokumentes beginnt mit dem Befehl `\begin{document}` und endet mit `\end{document}`.  $\LaTeX$  ignoriert sämtliche Eingaben nach `\end{document}`. Alle Anweisungen vor `\begin{document}` dienen zur Steuerung des Layouts. Man nennt diesen Bereich *Präambel*. Hier wird der Stil des Dokumentes festgelegt.

Der allererste Befehl in jedem  $\LaTeX$ -Dokument lautet `\documentclass`. Er erhält als Argument den Namen der zu verwendenden Dokumenten-Klasse. Die Dokumenten-Klasse legt das allgemeine Aussehen des Dokumentes und

dessen Eigenschaften fest. Für diese Ausarbeitung verwendete ich `article` als Dokumenten-Klasse. Dazu kommt `report` für einfache Dokumente und `book` für ganze Bücher. Das Aussehen des gesamten Dokuments kann mit Hilfe einer anderen Dokumenten-Klasse geändert werden, ohne dass auch nur eine weitere Zeile Text geändert werden müsste.

Die gewählte Dokumenten-Klasse Hilfe von Optionen modifiziert werden. Diese Optionen sind spezifisch zur Dokumenten-Klasse und nicht unbedingt auf andere Klassen anwendbar. Einige Beispiele für Dokumentenklassenoptionen von L<sup>A</sup>T<sub>E</sub>X's Standardklassen sind:

**11pt** erhöht die Basisschriftgröße von 10 auf 11 Punkt.

**twoside** formatiert den Text für eine zweiseitige Ausgabe.

**a4paper** aktiviert die Randeinstellungen für DIN A4-Papier.

**titlepage** erzeugt eine eigene Titelseite.

Für dieses Dokument habe ich folgende Einstellungen verwendet<sup>4</sup>:

```
\documentclass[titlepage,twoside,a4paper]{article}
```

### Makropakete

In der Präambel können weitere Makropakete geladen werden. Für deutsche Texte sind folgende Pakete empfehlenswert:

```
\usepackage{isolatin1}
\usepackage{german}
\usepackage[T1]{fontenc}
```

Das Paket `isolatin1` ermöglicht die direkte Eingabe von Umlauten, die sonst nur durch Kombinationen mit dem Steuerzeichen `"` erzeugt werden können. Das `german`-Paket aktiviert unter anderem dieses Steuerzeichen und sorgt für eine komplette Anpassung an den Deutschen Sprachraum<sup>5</sup>. Das `fontenc`-Paket sorgt dafür, dass die verwendeten Schriften die Deutschen Umlaute direkt verwenden können. Dies ist insbesondere für L<sup>A</sup>T<sub>E</sub>X's automatische Silbentrennung eine grosse Hilfestellung.

Für L<sup>A</sup>T<sub>E</sub>X sind Unmengen an Zusatzpaketen verfügbar. Die Distribution `teX` liefert bereits mehr als 150 Pakete mit, beispielsweise Pakete für lange Tabellen, viele neue Dokumenten-Klassen oder angepasste Syntaxumgebungen zum Setzen von Programmcode. Dazu sind im Internet viele weitere Pakete erhältlich, wie Kommandos für die Erstellung von Fluss- oder Organisationsdiagrammen.

<sup>4</sup>Die durch die eckigen Klammern eingefassten Parameter eines Befehls sind optional.

<sup>5</sup>Es übersetzt zum Beispiel die englischen Bezeichnungen für Kapitel, Inhaltsverzeichnisse usw.

### 3.4 Gliederungsbefehle

Absätze werden in eine hierarchischen Struktur, genannt Gliederung, unterteilt. Sie befinden sich derzeit im Abschnitt 3.4 „Gliederungsbefehle“, der dem Kapitel 3 „*LaTeX-Grundlagen*“ untergeordnet ist. Jede dieser Gliederungseinheiten beginnt mit einem Gliederungsbefehl, dessen Argument den Titel der Einheit beinhaltet. *LaTeX* kennt folgende Gliederungsbefehle, mit denen auch die Überschriften in einem Dokument erzeugt werden:

<code>\part</code>	<code>\subsection</code>	<code>\paragraph</code>
<code>\chapter</code>	<code>\subsubsection</code>	<code>\subparagraph</code>
<code>\section</code>		

Die Kapitelüberschrift wurde mit `\section{\LaTeX-Grundlagen}` und die Abschnittsüberschrift mit `\subsection{Gliederungsbefehle}` erzeugt. Soll aus diesen Befehlen automatisch ein Inhaltsverzeichnis erzeugt werden, muß lediglich an der Stelle, an der es erscheinen soll, der Befehl `\tableofcontents` geschrieben werden.

Zu beachten ist, daß nicht alle Dokumenten-Klassen auch jeden Gliederungsbefehl unterstützen. Die Klasse `article` kennt den Befehl `\chapter` nicht. Dadurch können sehr leicht mehrere Artikel zu einem Bericht (`report`) oder einem Buch (`book`) zusammengefasst werden. Benötigen Sie einen Anhang, kann dieser mit dem Befehl `\appendix` eingeleitet werden. Dieser sorgt für die korrekte Numerierung der Kapitel. Im Anhang können die gleichen Gliederungsbefehle wie im Hauptteil verwendet werden.

#### Zitate und Listen

*LaTeX* stellt viele Umgebungen zur Verfügung. Zwei der wichtigeren Elemente sind Zitate und Listen, die ich hier einmal exemplarisch vorstellen will.

Zitate sind abgesetzte Textzeile mit vergrößertem Rand. Die von *LaTeX* zur Verfügung gestellten Umgebungen `quote` und `quotation` können sehr vielseitig verwendet werden.

Mit *LaTeX* können folgende Sonderzeichen direkt erzeugt werden:

`. : ; , ? ! ' ' ( ) [ ] - / * @`

Man sieht hier einen alternativen Anwendungszweck für Zitat-Umgebungen.

Mit `\LaTeX\` können folgende Sonderzeichen direkt erzeugt werden:

```
\begin{quote}
. : ; , ? ! ' ' ( ) [ ] - / * @
\end{quote}
```

Man sieht hier einen alternativen Anwendungszweck für Zitat-Umgebungen.

Mehrere Listen-Umgebungen stehen Ihnen zur Verfügung, die selbstverständlich ineinander geschachtelt werden können. Dabei wird die Art der Numerierung von *LaTeX* automatisch festgelegt. Die Umgebung `itemize` erzeugt eine Aufzählungsliste und `enumerate` eine nummerierte Liste. Jeder Listeneintrag beginnt mit einem `\item`-Befehl. Die spezialisierte Listenumgebung `description` steht für Definitionslisten zur Verfügung. Der zu definierende Begriff wird dabei dem `\item`-Befehl in eckigen Klammern (`\item[Begriff]`) übergeben.

- Bei verschachtelten Listen schleichen sich gerne Syntax-Fehler ein.
  - Bis zu vier Listen können geschachtelt werden.
    1. Erstens
    2. Zweitens
- ```
\begin{itemize}
\item Bei verschachtelten Listen
schleichen sich gerne Syntax-Fehler
ein.
\item Bis zu vier Listen können
geschachtelt werden.
\begin{enumerate}
\item Erstens
\item Zweitens
\end{enumerate}
\end{itemize}
```

### Mathematische Formeln

Ich habe L<sup>A</sup>T<sub>E</sub>X's Formelsatz bereits angesprochen. Eine vollständige Einführung in den mathematischen Modus von L<sup>A</sup>T<sub>E</sub>X würde vermutlich nicht unter zehn Seiten auskommen, deshalb will ich Ihnen hier lediglich ein kurzes Beispiel über die Umgebungen, die den mathematischen Modus aktivieren, vorstellen<sup>6</sup>.

L<sup>A</sup>T<sub>E</sub>X beherrscht sowohl numerierte als auch nicht numerierte Formeln. Diese werden abgesetzt mit erhöhtem Rand, ähnlich den Zitaten, gesetzt. Selbstverständlich brauchen Sie sich nicht um die Numerierung zu kümmern. Die Dokumenten-Klasse legt fest, wie sie auszusehen hat. Nichtnumerierte Formeln werden in die `displaymath`-Umgebung eingefasst, die mit `\[` und `\]` abgekürzt werden kann. Numerierte Formeln hingegen können mit der `equation`-Umgebung gesetzt werden.

Dies ist ein Beispiel für eine nicht numerierte, abgesetzte Formel:

$$a^2 + b^2 = c^2$$

Wenn Sie Verweise auf eine Formel erstellen wollen, sollten Sie numerierende Umgebungen verwenden:

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

Dies ist ein Beispiel für eine nicht numerierte, abgesetzte Formel:

$$\[ a^2 + b^2 = c^2 \]$$

Wenn Sie Verweise auf eine Formel erstellen wollen, sollten Sie numerierende Umgebungen verwenden:

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

### 3.5 L<sup>A</sup>T<sub>E</sub>X und seine Hilfsprogramme

Dreh- und Angelpunkt jedes L<sup>A</sup>T<sub>E</sub>X-Systems sind die beiden Programme `latex` und `tex`. Sie rufen `latex` auf, das alle eigenen Makros durch T<sub>E</sub>X-Befehle ersetzt und diese dann an `tex` übergibt. Auf diese Weise erhalten sie eine geräteunabhängige Datei (engl. *device independent*) mit der Endung `dvi`. Diese Datei muß noch weiterverarbeitet werden, bevor sie beispielsweise auf einem Laserdrucker ausgegeben wird. Dazu kann die DVI-Datei mit dem *DVI-Treiber*

<sup>6</sup>Das *L<sup>A</sup>T<sub>E</sub>X-Handbuch*[4] und der *L<sup>A</sup>T<sub>E</sub>X-Begleiter*[5] geben umfangreiche Informationen zu L<sup>A</sup>T<sub>E</sub>X's mathematischen Modus.

`dvips` in das verbreitete PostScript-Format übersetzt werden. Es existieren einige weitere DVI-Treiber, beispielsweise `dvilj4` für HP-PCL oder `dvifax` für ein Gruppe 2 Fax-Format. Speziell in der Unix-Welt ist das PostScript-Format aber so weit verbreitet, dass sie diese anderen DVI-Treiber dort kaum benötigen werden.

### Spezialisierte $\LaTeX$ -Varianten

Mehrere abgewandelte  $\LaTeX$ -Varianten sind im Moment erhältlich, die anstelle der DVI-Datei ein anderes Format erzeugen. Ein sehr nützliches Tool ist `pdflatex`, mit dem aus einer  $\LaTeX$ -Datei direkt eine PDF-Datei erzeugt werden kann. Man benötigt nicht das teure Adobe Acrobat System, um  $\LaTeX$ 's PostScript Dateien in das PDF-Format umzuwandeln.

Ergänzt wird dies mit `latex2html`, daß  $\LaTeX$ 's Markupanweisungen in HTML-Code übersetzt. Dies funktioniert zwar nur mit Einschränkungen, genügt aber für viele Zwecke bereits. Würden Sie `latex2html` beispielsweise auf diese Seminararbeit ansetzen, würden dieses auf Grund der speziellen Syntaxboxen kapitulieren.

### Sonstige Hilfsprogramme

Für die Erzeugung von Indizes ist bei vielen  $\LaTeX$ -Distributionen das Hilfsprogramm `makeindex` dabei. Es generiert aus `\index`-Befehlen eine Indexdatei, die anschliessend von  $\LaTeX$  ausgelesen und gesetzt wird.

Ähnlich arbeitet das Programm `bibtex`, welches aus einer vorhandenen Literaturdatenbank automatisch Literaturverzeichnisse erstellen kann<sup>7</sup>.

---

<sup>7</sup>Das Literaturverzeichnis dieser Seminararbeit wurde ebenfalls mit `BIB $\TeX$`  erzeugt.

## 4 Fazit

$\LaTeX$  hat sich während meines Praxissemesters mehrfach bewährt. Die grössten Stärken von  $\LaTeX$  liegen in der Plattformunabhängigkeit, der hohen Verarbeitungsgeschwindigkeit und der exzellenten Ergebnisse. Die grösste Schwierigkeit beim Umstieg von einem gewöhnlichen Textprogramm zu  $\LaTeX$  ist es, sich völlig von der eigentlichen Formatierung zu distanzieren. Man gerät gerne in Versuchung, selbst eine visuelle Formatierung durchzuführen. Hat man dieses „Problem“ einmal überwunden, kann man mit  $\LaTeX$  sehr schnell hervorragende Dokumente produzieren.

Ich denke nicht, dass Programme wie Word, StarOffice oder WordPerfect die Foo-Bibel in dieser Qualität hätten erzeugen können. Rechnet man hinzu, dass für  $\LaTeX$  keine Lizenzgebühren gezahlt werden müssen, kann ich  $\LaTeX$  ohne Einschränkung an jeden empfehlen. Vor allem für Studenten in technischen und wissenschaftlichen Bereichen dürfte  $\LaTeX$  besonders gut geeignet sein. Schon alleine, um sich während der Diplomarbeit Ärger auf Grund instabiler Software vom Leibe zu halten.

## Literatur

- [1] Matthias Kalle Dalheimer. *LaTeX kurz & gut*. O'Reillys: ISBN: 3-89721-204-8, 1998. <http://www.nathan-syntronics.de/buecher/165.html>.
- [2] DANTE e.V. Deutschsprachige Anwendervereinigung TeX e.v. WWW. <http://www.dante.de>.
- [3] Thomas Esser. The TeX homepage. WWW. <http://www.tug.org/tetex>.
- [4] Leslie Lamport. *Das LaTeX Handbuch*. Addison-Wesley: ISBN: 3-89319-826-1, 1995. <http://www.nathan-syntronics.de/buecher/159.html>.
- [5] Michel Goossens, Franz Mittelbach und Alexander Samarin. *Der LaTeX Begleiter*. Addison Wesley: ISBN: 3-89319-646-3, 1994. <http://www.nathan-syntronics.de/buecher/166.html>.