LaTeX

— Art der Arbeit —

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

Vorgelegt von: Arne Westphal
E-Mail-Adresse: 3westpha@informatik.uni-hamburg.de
Matrikelnummer: 6525702
Studiengang: B.Sc. Informatik

Betreuer: Nathanael Hübbe

Hamburg, den 03.09.2014

Inhaltsverzeichnis

1 Einleitung

LaTeX ist ein Softwarepaket, welches von Leslie Lamport entwickelt wurde. Es basiert auf TeX, welches selbst ein Textsatzsystem ist. TeX wurde entwickelt, um einheitliche und schöne Layouts einfacher gestalten zu können. Im Gegensatz zu anderen Textverarbeitungssystemen baut es nicht auf das WYSIWYG-System („What you see is what you get“) auf. Häufig wird das TeX-Verhalten auch als WYSIWYM („What you see is what you mean“) bezeichnet.

LaTeX liefert Makros für TeX und erleichtert dadurch den Umgang mit TeX für den Nutzer. Der wohl wichtigste Teil von LaTeX ist der Umgang mit mathematischen Formeln. Die Darstellung von komplexen mathematischen Formeln wird durch LaTeX sehr vereinfacht. [LaT14a]

Stärken sind zum einen, dass es auch bei langen Texten durch LaTeX zu konsistenten Layouts kommt. Es wird einem, mit einigen generellen Einstellungen zum Dokument, die Arbeit der Layoutgestaltung abgenommen.
Außerdem lassen sich leicht Abschnitte eines Dokuments in einzelne Dateien gliedern. Dadurch wird es einem erleichtert, in einem Dokument den richtigen Abschnitt zu finden, wenn man diese in Dateien eingeteilt hat.
LaTeX liefert zudem die Möglichkeit, Makros zu erstellen. Diese können vorhandene Makros erweitern, indem sie sie wiederverwenden. Sie können aber auch ganz neue Ergebnisse liefern.

2 Arbeiten mit LaTeX

2.1 Anfang des Dokuments

Damit LaTeX all das für einen tun kann, wofür es gedacht ist, muss eine bestimmte Struktur im Dokument eingehalten werden. Dazu gehört zum Beispiel, dass jedes LaTeX-Dokument mit dem „`\documentclass[...]{...}`“ Befehl beginnen muss. Durch diesen Befehl werden dem Dokument die groben Einstellungen des Layouts vorgegeben.
Wichtige Einstellungen hier sind:

- Schriftgröße (z.B. 12pt)
- Seitengröße (z.B. Din A4)
- Ein- oder doppelseitiger Druck

Es gibt noch viel mehr Einstellungsmöglichkeiten, aber diese gehören zu den Wichtigsten. Die hier genannten Einstellungen gehören in die eckigen Klammern. In LaTeX bedeuten eckige Klammern, dass die Parameter in ihnen optional sind. Man muss also nicht alle angeben. Gibt man die Parameter nicht an, gibt es Standardwerte die genutzt werden.

Die geschweiften Klammern hingegen sind Pflichtparameter. In jede geschweifte Klammer eines Befehls muss ein Parameter. Bei dem „`\documentclass[...]{...}`“ Befehl stehen die geschweiften Klammern für die Dokumentenklasse.
Die Standardklassen sind: [Lam94]

- `article` - für kleinere Schriftwerke meist ohne Titelblatt und nur mit einseitigem Druck
- `report` - für mittlere Schriftwerke meist mit Titelblatt und einseitigem Druck
- `book` - für große Schriftwerke meist mit Titelblatt und zweiseitigem Druck
- `letter` - für Briefe

Hieraus würde beispielsweise folgender Code entstehen:

Listing 2.1: LaTeX-Dokumentenklasse

```
1 \documentclass[12pt, oneside,
2     paper=a4]{report}
```

2.2 Pakete

Neben der Dokumentenklasse stehen im Kopfbereich des Dokuments noch andere Dinge, die nicht zum eigentlichen Inhalt gehören.

Ein Teil davon sind die Pakete. LaTeX bietet nicht alle Befehle, die man verwenden möchte, direkt an. Viele davon werden in einzelnen Paketen geliefert. Und nicht nur das: Einige Pakete liefern keine direkten Befehle, sondern tragen zum Beispiel passiv zur Layoutgestaltung bei durch korrekte Silbentrennung und Ähnliches.

Möchte man nun also ein solches Paket nutzen, muss dieses zunächst auf dem Computer installiert und in das Dokument eingebunden werden. Einige TeX-Distributionen installieren die Pakete automatisch, wenn man sie einbinden möchte. Bei anderen muss man die Pakete manuell installieren. Ist das gewünschte Paket, installiert kann man es in ein spezifisches Dokument per „`\usepackage[...]{...}`“ einbinden.
Dabei gilt wie vorher: Eckige Klammern sind optionale Parameter und in den Geschweiften steht in diesem Fall der Name des einzubindenden Pakets.

Die häufigsten Pakete sind:

- `inputenc` - lädt den richtigen Zeichensatz (meist `UTF8`)
- `babel` - lädt den richtigen Wortschatz. Z.B. zur Silbentrennung (meist `ngerman`)
- `graphicx` - zum Einbinden von Grafiken

Die in den Klammern genannten häufig verwendeten Einstellungen werden einfach in eckigen Klammern eingefügt. So ergeben diese Pakete z.B.:

Listing 2.2: LaTeX-Dokumentenklasse

```
1 \usepackage[utf8]{inputenc}
2 \usepackage[ngerman]{babel}
3 \usepackage{graphicx}
```

2.3 Dokumentenrumpf

Der Rumpf des LaTeX-Dokuments ist für den Inhalt gedacht. Hier steht nun der Code, der wirklich das erzeugt, was später sichtbar ist. Gekennzeichnet ist er durch „`\begin{document}`“ und „`\end{document}`“. Letzterer Befehl ist dabei die letzte Zeile des LaTeX-Dokuments. Alles, was vor diesem Bereich kommt, zählt zum Kopfbereich.

Ein Textdokument kann nun innerhalb dieses Rumpfes weiter gegliedert werden. Dazu gibt es folgende Unterteilungen:

- `chapter` - Oberste Hierarchieebene. Gibt es nur für Dokumentenklassen Berichte und Bücher.
- `section` - Eine Hierarchieebene tiefer. Bilden Unterpunkte eines Kapitels.
- `subsection` - Bilden Unterpunkte der vorher definierten `section`.
- `subsubsection` - Tiefste im Inhaltsverzeichnis sichtbare Ebene.
- `paragraph` - Standardmäßig im Inhaltsverzeichnis nicht mehr sichtbar.
- `subparagraph` - Tiefste Standardebene.

Diesen Bereichen muss jeweils ein Name gegeben werden. Der Befehl dazu ist beispielsweise „`\section{Graphentheorie}`“, wobei „Graphentheorie“ hier der Name des Bereichs ist. Standardmäßig werden alle Bereiche durchnummeriert. Die höchste Ebene stellt die erste Zahl dar: „1.“. Die tieferen Ebene ergeben die Unterpunkte:, z.B.: „1.1 - 1.12“.

Ein Beispiel für eine solche Gliederung ist folgendes:

Listing 2.3: Struktur eines LaTeX-Dokuments

```
1 \chapter{Geometrie}
2
3     \section{Dreiecke}
4
5         \subsection{Pythagoras}
6
7     \section{Vierecke}
```

Ergebnis des Quellcodes ist:

1. Geometrie
1.1. Dreiecke
1.1.1 Pythagoras
1.2. Vierecke

2.4 Befehle

LaTeX arbeitet sehr viel mit Befehlen. Dabei gibt es verschiedene Typen von Befehlen, die sich nicht nur unterschiedlich auf das Ergebnis auswirken. Auch ihre Verwendung ist unterschiedlich.

Zunächst gibt es Befehle, welche an der Stelle, an der sie stehen, einen anderen Text einfügen. Ein Beispiel dafür ist „`\LaTeX`“. Es werden keinerlei Parameter benötigt und es verändert sich dadurch nichts, abgesehen davon, dass ein LaTeX in den Text eingefügt wird.

Befehle können sich außerdem auch wie ein Umschalter verhalten. Benutzt man einen Befehl wie „`\itshape`“, wird der gesamte folgende Text kursiv ausgegeben. Wieder ausschalten lässt sich dies durch den Befehl „`\upshape`“, welcher wieder auf die reguläre Schrift umschaltet.

Da man häufig gerade bei diesem Beispiel nur einen kleinen Teil des Textes kursiv ausgeben möchte, gibt es Befehle mit Parametern. Statt „`\itshape`“ kann man auch „`\textit{...}`“ verwenden, welcher nur den Text in den geschweiften Klammern in kursiv ausgibt. Wie in Abschnitt 2.1 bereits erklärt, gilt auch hier, dass diese Befehle die geschweiften Klammern benötigen. Sie sind also Pflicht.

Wie ebenfalls bereits gezeigt, gibt es also auch Befehle mit optionalen Parametern. Ein Beispiel hier ist „`\textcolor[rgb]{1,0,0}{TEXT}`“. Bei diesem Befehl sorgt der optionale Parameter dafür, dass die erste Klammer nicht als Name der zu verwendenden Farbe, sondern als deren RGB-Werte interpretiert wird.

Da allerdings nicht alle Befehle bereits existieren, kann man sich auch welche selber erstellen. Hierfür gibt es den Befehl „`\newcommand{\NAME}{AUSGABE}`“. Dieser Befehl hat keine Parameter, er ersetzt lediglich das „`\NAME`“ im Text durch die *Ausgabe*. Dies kann praktisch sein, um häufig verwendete Texte in einen kurzen Befehl zusammenzufassen.

Möchte man nun einen Pflichtparameter in seinem Befehl, nutzt man „`\newcommand{\NAME}[1]{...#1...}`“. Über den Zugriff auf *#1* wird der Pflichtparameter in die Ausgabe miteingebunden. „`\newcommand{\zitat}[1]{"‘#1"’}`“ würde den Pflichtparameter in die richtigen Anführungszeichen setzen. So wird aus „`\zitat{Mailand oder Madrid, hauptsache Italien"}`“: „Mailand oder Madrid, hauptsache Italien“. Möglich ist dies auch mit mehreren Parametern. Dazu muss nur die 1 in den eckigen Klammern des `\newcommand`-Befehls durch die gewünschte Zahl ersetzt werden. Der Zugriff auf die Parameter erfolgt mit *#1, #2* und so weiter. Bei der Nutzung reiht man dann mehrere geschweifte Klammern hintereinander.

Zur Erstellung von Befehlen mit optionalem Parameter nutzt man „`\newcommand{\NAME}[2][Standardbelegung]{...#1..#2.}`“. Dabei steht die *2* wieder für zwei Parameter. Dass der Erste optional ist, liegt daran, dass dahinter in den eckigen Klammern die Standardbelegung mitgeliefert wird. Sollen also mehrere optional sein, müssen in der eckigen Klammer den Nummern die Standardwerte zugewiesen werden. Ein Beispiel hierfür ist:
„`\newcommand{\bsp}[3][1=x, 3=z]{#1#2#3}`“

Listing 2.4: Struktur eines LaTeX-Dokuments

```
1 Nutzung                I         Ausgabe
2 ----------------------I---------------------------
3 \bsp{y}                I         xyz
4 \bsp[a]{y}             I         ayz
5 \bsp[a]{y}[b]          I         ayb
6 \bsp{y}[b]             I         xyb
```

Die Zahlen sind dabei immer die Reihenfolge der Parameter bei der Nutzung: „`\NAME[#1]{#2}[#3]{#4}`“. Maximal sind neun Parameter möglich. [Pos08]

2.5 Referenzen

In längeren Dokumenten ist es häufig sinnvoll, auf andere Bereiche zu verweisen. Möchte man also beispielsweise erwähnen, dass man an einer Stelle auf eine Definition aus einem bestimmten Bereich zurückgreifen kann, so ist dies in LaTeX leicht möglich. Man kann überall im Text ein `label` setzen, mit dem man sich immer wieder auf diese Stelle beziehen kann. Das `label` ist dabei eine kurze Bezeichnung hierfür.
Im Normalfall wird den in 2.3 beschriebenen Bereichen je ein `label` zugewiesen. So kann man sich mit `\ref{Bezeichnung}` einfach auf die Nummer des Abschnitts beziehen.
Man kann sich allerdings auch direkt auf die Seitenzahl beziehen, auf der sich das Referenzierte befindet. Diese Seitenzahl orientiert sich nicht an der Seite der Bereichsüberschrift, sondern auf die Seite, an der tatsächlich das `label` sein würde, wenn es im Text sichtbar wäre.
Auf diese Weise, kann man sich auch auf bestimmte Dinge beziehen, die mitten in einem größeren Bereich stehen. Die Referenz zeigt also nicht zwangsläufig auf den Anfang des Bereichs.
Eine beispielhafte Verwendung wäre:

Quelltext:

Listing 2.5: Quelltextbeispiel - Referenzen

```
1  \section{Wie funktioniert \LaTeX ?}
2  \label{latex:funktion}
3
4      [...]
5
6  \section{Umgang mit \LaTeX ?}
7
8      Wie bereits in \ref{latex:funktion} auf Seite
9      \pageref{latex:funktion} erkl"art,
10
11     [...]
```

Ausgabe:

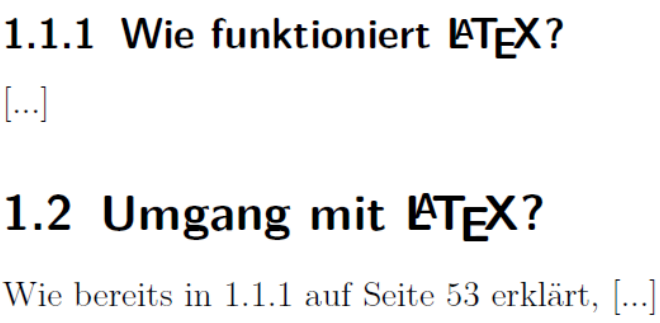

Abbildung 2.1: Quelltextausgabe Referenzen

2.6 Whitespace

Wichtig, gerade im Umgang mit Befehlen, ist zu wissen, wie LaTeX mit dem Quelltext umgeht. Generell gilt: An jeder Stelle, an der mehrerer Leerzeichen direkt aufeinanderfolgen, wird nur ein Leerzeichen dargestellt. Möchte man das nun unbedingt umgehen, weil man mehrere Leerzeichen darstellen möchte, muss man vor jedes zusätzliche Leerzeichen ein „\“ setzen. Es ergibt sich also zum Beispiel: „`A \ \ \ B`“, was nun vier Leerzeichen zwischen A und B darstellt.
Außerdem werden Leerzeichen nach einem Befehl wie „`\LaTeX`“ nur als Ende des jeweiligen Befehls erkannt. Möchte man also ein Leerzeichen zwischen der Befehlsausgabe und dem nächsten Zeichen, schreibt man „`\LaTeX\ `“.
Ähnliches wie für Leerzeichen gilt für Absätze. Allerdings wird hier ein reiner Zeilenumbruch komplett ignoriert. Eine Leerzeile bewirkt, dass in der Ausgabe ein Absatz erscheint. Der Anfang des neuen Absatzes wird dabei standardmäßig eingerückt, was sich durch „`\parindent0pt`“ für das gesamte Dokument deaktivieren lässt. Möchte man nun einen reinen Zeilenumbruch haben, helfen einem die Befehle „`\\`“ und „`\newline`“, welche beide das gleiche bewirken. Da mehrere solcher Zeilenumbrüche allerdings von LaTeX nicht gerne gesehen sind, sollte man für Leerräume „`\vspace{Abstand}`“ verwenden. Der Abstand kann in allen LaTeX bekannten Längeneinheiten angegeben werden. Am besten eignen würde sich hier wohl eine Zentimeter- (cm) oder Millimeterangabe (mm). Mehrere Leerzeilen werden, ebenso wie mehrere Leerzeichen, ignoriert.

2.7 Sonderzeichen

LaTeX verwendet viele Symbole für Kennzeichnungen im Quelltext. Zum Beispiel ein „\“ kennzeichnet den Anfang eines Befehls. Möchte man nun diese Zeichen aber im Text wirklich abbilden, so muss man häufig etwas anderes schreiben als das jeweilige Zeichen.

Häufig gebrauchte Zeichen sind:

Command	Symbol	Command	Symbol
\%	%	\#	#
\$	$	\&	&
\{	{	\}	}
_	_	\S	§
\P	¶	\dag	†
\ddag	‡	\textbackslash	\
\textbar	|	\textless	<
\textgreater	>	\textemdash	—
\textendash	–	\textregistered	®
\texttrademark	™	\textquestiondown	¿
\textexclamdown	¡	\textcircled{a}	ⓐ
^{a}	ª	\copyright	©
\pounds	£		

Abbildung 2.2: LaTeX Sonderzeichen [LaT14b]

Gerade mathematische Symbole können aber in Formeln auch angepasst dargestellt werden. In der höheren Mathematik werden dazu oft Symbole über Variablen verwendet.

Die folgenden Befehle, welche in der Informatik viel verwendet werden, können dabei nur in den Matheumgebungen, auf die in 3.1 noch genauer eingegangen wird, verwendet werden. Falls es eine zugehörige Variante für den Gebrauch im normalen Text gibt, steht diese in der Spalte „Im Text“: [LaT14b]

Befehl	**Ergebnis**	**Im Text**
`\hat{a}`	$\hat{a}$	`\^`
`\tilde{a}`	$\tilde{a}$	`\~`
`\dot{a}`	$\dot{a}$	`\.`
`\bar{a}`	$\bar{a}$	`\=`
`\vec{a}`	$\vec{a}$	

3 Umgebungen

Umgebungen sind Bereiche im Quelltext. In diesen Umgebungen können zum Beispiel andere Befehle nutzbar sein, sie sorgen für eine andere Textformatierung oder ähnliches. Die häufigste verwendete Umgebung ist wohl die `document` Umgebung. Diese befindet sich in jedem LaTeX-Dokument.

3.1 Matheumgebung

Aber es gibt noch viele andere Umgebungen, die häufig genutzt werden. Da in LaTeX mathematische Formeln eine sehr große Rolle spielen, gibt es natürlich auch hierfür die jeweiligen Umgebungen. Diese heißen `math` und `displaymath`. Weil diese so oft verwendet werden, muss man sie nicht mit „`\begin{Name}`“ einleiten und mit „`\end{Name}`“ beenden, wie für andere Umgebungen üblich, sondern sie besitzen Kurzformen. So leitet „`\(`“ die `math`-Umgebung ein und „`\)`“ beendet sie. Gleiches gilt mit „`\[`“ und „`\]`“ für `displaymath`. Abgesehen von „`\( \)`“ und „`\[ \]`“ werden häufig auch die TeX-Varianten „`$ $`“ für `math` und „`$$ $$`“ für `displaymath` verwendet. Allerdings sind diese häufige Fehlerquellen und werden nicht von allen Compilern akzeptiert.
Diese Matheumgebungen sorgen nun dafür, dass z.B. ein „`x^2`“ in der Umgebung als x^2 ausgegeben wird. Auf diese Weise kann man mit Befehlen, die nur in der Matheumgebung funktionieren, schöne mathematische Formeln in ein Dokument einbinden.
`math` und `displaymath` unterscheiden sich nur in ihrer Darstellung im Dokument. `math` stellt die mathematische Formel innerhalb einer Textzeile dar. So kann man z.B. ein $e^{\frac{x^2}{5}}$ mitten in einem Satz durch den Befehl „`\( e^{\frac{x^2}{5}} \)`“ positionieren.
Möchte man die Formel aber hervorheben, eignet sich `displaymath`, da dieses die Formel in einer eigenen Zeile zentriert darstellt. Auf diese Weise werden Formeln mit größerer Zeilenhöhe weder gestaucht noch vergrößern sie den Zeilenabstand. Dies eignet sich also für komplexe Formeln besser. Der selbe Befehl in der `displaymath`-Umgebung ergibt also:

$$e^{\frac{x^2}{5}}$$

Ein paar einfache Befehle hier sind:

- `\frac{Zähler}{Nenner}` - stellt einen Bruch dar.
- `\sqrt[Wurzelexponent]{Formel}` - erzeugt ein Wurzelzeichen um die Formel.
- `{Formel}` - Die Formel fungiert als ein Block
  - ist z.B. bei „2x“ als Exponenten nötig, um beide hochzustellen

3.2 Aufzählungen

Natürlich gibt es extrem viele Umgebungen, die man nutzen kann.
Auch wichtig sind die Aufzählungsumgebungen `itemize`, `enumerate` und `description`. Überall gilt, dass die Elemente in der Aufzählung jeweils durch ein „`\item`“ eingeleitet werden. Alles in der Zeile hinter diesem „`\item`“ wird nun als ein Element der Aufzählung erkannt. `itemize` ist dabei die nichtnummerierte Aufzählung, die z.B. einen Punkt als Aufzählungszeichen vor das jeweilige Element setzt.
`enumerate` nummeriert die Elemente der Aufzählung durch. Welche Art der Nummerierung vorgenommen wird, lässt sich durch das Paket `paralist` einstellen. Standardmäßig werden in der ersten Ebene Zahlen verwendet in der Form „1.“ und in der zweiten Ebene Buchstaben wie „(a)“.
Mit `description` kann man den Elementen eine vorangestellte, fettgedruckte Beschreibung zuordnern. Ansonsten werden hier keine weiteren Aufzählungszeichen verwendet.
So sehen diese Umgebungen standardmäßig aus:

Quelltext	*Ausgabe*

Listing 3.1: Struktur eines LaTeX-Dokuments

```
1  \begin{description}
2      \item[itemize]~
3          \begin{itemize}
4              \item Element
5              \item Element
6          \end{itemize}
7      \item[enumerate]~
8          \begin{enumerate}
9              \item Element
10             \item Element
11         \end{enumerate}
12     \item[description]~
13         \begin{description}
14             \item[a] Element
15             \item[b] Element
16         \end{description}
17 \end{description}
```

itemize

- Element
- Element

enumerate

1. Element
2. Element

description

a Element

b Element

Das Symbol „~“ steht für ein geschütztes Leerzeichen und sorgt hier für den richtigen Zeilenumbruch. Beginnt man in einer dieser Umgebungen eine weitere Aufzählungsumgebung, wird eine zweite Aufzählungsebene erstellt, in der die Elemente weiter eingerückt sind. Gegebenenfalls sind auch die verwendeten Symbole in den einzelnen Ebenen unteschiedlich. Ich werde hier nicht weiter darauf eingehen, wie man diese Symbole verändern kann.

3.3 Tabellen

Natürlich bietet LaTeX auch die Möglichkeit Tabellen in sein Dokument einzubinden. Hierfür wird im Normalfall die Umgebung `tabular` verwendet. Beim Beginn der Umgebung muss hier als zusätzlicher Parameter die Spaltendefinition mitgegeben werden.
Hier sind folgende Eingaben möglich:

- `|` - Die vertikale Linie erzeugt in der Tabelle eine sichtbare Abgrenzung zwischen den Spalten oder am Rand.
  - auch als „`||`“ möglich, für doppelte Linie
- `l`, `r`, `c` - Eine links-, rechtsbündige oder zentrierte Spalte wird erzeugt.
- `p{Breite}` - Eine Spalte mit fester *Breite* wird erzeugt. Der Text wird linksbündig dargestellt.

Zu beachten ist, dass l,r,c keine feste Spaltenbreite besitzen. Die Spalte wird so breit, wie der breiteste Eintrag. Dies kann sogar breiter sein als der verfügbare Platz, wodurch Teile der Tabelle abgeschnitten werden. Hat man also vor, längere Einträge in die Spalte zu schreiben, sollte man `p` wählen und die Breite entsprechend wählen. Aber auch hier kann die Breite vom Nutzer zu groß gewählt sein, wodurch die Tabelle ebenfalls zu breit ist.
Innerhalb des Tabellen Inhalts verwendet man ein `&` um in die nächste Spalte zu springen. Möchte man in die nächste Zeile springen, verwendet man `\\`. `\hline` erzeugt eine Linie zwischen zwei Zeilen. Diese kann man, wie bei Spalten ebenfalls, doppelt verwenden, um eine doppelte Linie zu erzeugen.
So könnte eine Tabelle aussehen:

Quelltext:

Listing 3.2: Quelltextbeispiel Tabelle

```
1 \begin{tabular}{| c || r | l | p{0.3\textwidth}}
2     zentriert & rechtsb.     & linksb.   & paragraph \\
3     \hline
4     und       & noch einmal  & das Ganze & Hier steht noch
5                                          ein langer Text.\\
6     \hline \hline
7 \end{tabular}
```

Ausgabe:

zentriert	rechtsb.	linksb.	paragraph
und	noch einmal	das Ganze	Hier steht noch ein langer Text.

Tabelle 3.1: Beispiel einer Tabelle

4 Zusammenfassung

Insgesamt lässt sich nach diesem Einblick feststellen, dass LaTeX extrem umfangreich ist, was die zur Verfügung gestellten Möglichkeiten angeht. LaTeX bietet die Grundlage eines unendlich erweiterbaren Programms. Durch die Pakete lassen sich, je nach gewünschter Verwendung, beliebig Funktionen hinzufügen oder rauslassen.
Man muss jedoch auch hinzufügen, dass LaTeX im Gegensatz zu bekannten Textverarbeitungsprogrammen wie Word weniger einsteigerfreundlich ist. Die Einarbeitungszeit ist relativ hoch und man muss sich bei jedem neuen Paket an die jeweilige Verwendung gewöhnen.
Wenn man bereits viel mit anderen Programmen gearbeitet hat, ist die Verwendung von LaTeX auf jeden Fall eine große Umstellung. Möchte man aber lange oder wissenschaftlich orientierte Dokumente verfassen, ist LaTeX die perfekte Wahl um ein schönes und konsistentes Layout zu erhalten.

Literaturverzeichnis

[Lam94] Leslie Lamport. *LaTeX. A Document Preparation System.* Addison-Wesley Professional, zweite edition, 1994.

[LaT14a] LaTeX. `http://en.wikibooks.org/wiki/LaTeX`, 7 2014. [zuletzt abgerufen am 01.09.2014].

[LaT14b] LaTeX - Special Characters. `http://en.wikibooks.org/wiki/LaTeX/Special_Characters`, 8 2014. [zuletzt abgerufen am 01.09.2014].

[Pos08] Matthias Pospiech. Definition von Makros und Umgebungen. `http://www.matthiaspospiech.de/blog/2008/04/16/definition-von-makros-und-umgebungen/`, April 2008. [letztes Abrufdatum: 19.08.2014].

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis