

Kurzes L^AT_EX Tutorial für Physik-Studenten

Holger Jorra

5. Februar 2009

Dieses Tutorial richtet sich an Menschen, die beim Lesen des Wortes L^AT_EX nicht ausschließlich an Kautschukprodukte denken und sich schon Abstrakt mit dem Thema auseinander gesetzt haben, wissen wofür das Akronym WY-MIWYG in diesem Kontext steht und deshalb nicht erstaunt darüber sind, wenn sich L^AT_EX nicht wie die gewohnte Office Suite verhält, sondern „kryptische“ Tags über einen Editor in eine Datei geschrieben müssen und diese anschließend kompiliert wird. Sie müssen auch nicht mehr von den Vorteilen des Zeichensetzprogrammes, vor allem im naturwissenschaftlichen Bereich, überzeugt werden. Die Ergebnisse sprechen für sich.

Andererseits soll dies auch kein Fachbuch zum Thema sein, nicht mal eine Einführung! Dies ist ein Tutorial, d.h. anhand des sofort nutzbaren Beispiels eines mit [Blindtext](#) gefüllten Protokolls werden hoffentlich alle notwendigen Dinge erklärt, die ein Student *unbedingt* für die Anfertigung eines solchen benötigt, um einen einfachen Einstieg in das Thema zu ermöglichen und die Angst vor der Komplexität zu nehmen. Daher wird hier auf Layout-Funktionen möglichst verzichtet. Alles was für das Protokoll nützlich aber nicht essentiell ist, habe ich in den Teil 8 ausgelagert. Dort beginnt sozusagen der Anhang des Tutorials. Für weitergehende und tiefgründige Informationen ist das Internet oder das angesprochene Fachbuch die Quelle an die man sich wenden sollte. Im Teil 9 dieses Schriftstückes sind dafür schon mal ein paar Links angegeben. Ich verzichte ebenfalls darauf, auf bestimmte Literatur zu verweisen. Einerseits besitze ich selbst keine, daher kann ich keine empfehlen. Andererseits hat es sich für mich als effizient erwiesen, gesuchte Funktionen über eine Suchmaschine zu recherchieren, selbst damit zu experimentieren und für mich selbst zu dokumentieren.

Wer einen plattformunabhängigen Editor sucht, dem sei an dieser Stelle die freie Software [T_EXMaker](#) empfohlen. Ansonsten findet sich eine Anleitung zu Bezugsquellen eines L^AT_EX Systems in der [DE-T_EX-FAQ](#).

Disclaimer: Verwendete Produkt- und Firmennamen sind möglicherweise gesetzlich geschützt. Ein fehlender ausdrücklicher Hinweis darauf kann nicht zu der Annahme führen, dass keine Schutzrechte bestehen.

Inhaltsverzeichnis

| | | |
|-----------|-----------------------------------|-----------|
| 1 | Der Header | 3 |
| 1.1 | Dokumentenklassen | 3 |
| 1.2 | Pakete einbinden | 3 |
| 1.3 | Dokumentformatierungen | 4 |
| 1.4 | Zeichenkodierung | 4 |
| 2 | Titelseite | 4 |
| 3 | Textformatierungen | 5 |
| 4 | Umgebungen | 5 |
| 4.1 | Minipages | 6 |
| 4.2 | Tabellen | 6 |
| 4.3 | Bilder | 7 |
| 4.4 | Aufzählungen | 8 |
| 4.5 | Mathematische Umgebung | 8 |
| 5 | Referenzen | 9 |
| 6 | Abschluß | 10 |
| 7 | Kompilieren | 10 |
| 8 | Tricks und Kniffe | 10 |
| 8.1 | Dateien einbinden | 10 |
| 8.2 | Seitenränder | 11 |
| 8.3 | Nummerierte Tabellen | 11 |
| 8.4 | Darstellung von Formeln | 12 |
| 8.5 | Absätze | 12 |
| 9 | Weiterführende Links | 13 |
| 10 | Befehlsreferenz | 13 |
| 11 | Lizenz | 13 |

1 Der Header

Zu Beginn muß \LaTeX mitgeteilt werden, welche Formatierungen und zusätzlichen Pakete es benutzen soll. Dies steht im Kopf (Head) des Dokuments. Unter dem Dokumentenkopf folgt der Textkörper (Body) in dem der eigentliche Inhalt unseres Schriftstückes untergebracht ist. Für diejenigen mit Erfahrung in HTML ist das alles nichts neues. Um dies schon einmal vorweg zu nehmen, der Body befindet sich zwischen den zwei Tags $\text{\begin\{document\}}$ und $\text{\end\{document\}}$.

1.1 Dokumentenklassen

Generell gibt es in \LaTeX 4 Klassen von Dokumenten zur Drucklegung:

- Article
- Book
- Letter
- Report

plus jene für Präsentationen (slides bzw. beamer). Diese Klassen bestimmen die Art und Weise, wie \LaTeX den Text für uns formatiert, z.B. ob es als größte Unterteilung nun Paragraphen, Kapitel oder gar Teile gibt. [Markus Kohm](#) hat uns für den deutschsprachigen Raum mit Makros versorgt, die an unsere regionalen Konventionen angepasst sind (z.B. Blätter der Größe DIN A4 statt letter). Die Bezeichnungen dieser so genannten *KoMa-Scripts* setzen sich aus einem gekürzten Namen¹ der 4 ursprünglichen Klassen und dem Präfix **scr** zusammen. Für kurze wissenschaftliche Schriften sind offensichtlich nur die Klassen Article und Report interessant. Im Beispiel wird Report benutzt, aber es steht natürlich jedem frei mit den Klassen zu experimentieren.²

Um \LaTeX nun unseren Wunsch mitzuteilen wird die erste Zeile in unserer neuen .tex Datei folgende sein

Listing 1: Protokoll.tex

```
1 \documentclass[a4paper,11pt]{scrreprt}
```

Außerdem haben wir dem Programm auch noch die Papier- und Standardschriftgröße mitgegeben.

1.2 Pakete einbinden

Normalen Fließtext kann man nun zwar schreiben, aber als Naturwissenschaftler will man natürlich mehr. Um mathematische Symbole usw. nutzen zu können müssen zusätzliche Pakete eingebunden werden. Dies erfolgt über das Kommando \usepackage . Die dazugehörigen Optionen sind paketspezifisch. Sinnvoll erscheint folgende Liste:

Listing 2: Pakete

```
2 \usepackage[ngerman]{babel} %Neue deutsche Rechtschreibung
3 \usepackage[dvips]{graphicx} %Fuer Bilder
4 \usepackage{amsmath} %Mathe
5 \usepackage{amssymb} %Mathesymbole
6 \usepackage{longtable} %Seitenubergreifende Tabellen
```

¹Warum im KoMA-Script article scrartcl, report scrreprt und letter scrlltr2 heißen, weiß wohl nur allein der Markus. Sonderlich intuitiv mag das nicht sein, aber manche Dinge muß man einfach so hinnehmen. Zur Not lässt sich alles im [scrguide](#) nachlesen.

²Hier wird die report nur deshalb verwendet, weil ich auch die Einteilung in Kapitel zeigen will und außerdem ein Deckblatt auf einer eigenen Seite erhalte. Wesentlich sinnvoller für Praktikumsprotokolle ist die Dokumentenklasse article, während report sich eher für umfangreichere Dokumente eignet wie Diplom- oder Bachelorarbeiten.

Kommentare und nicht interpretierte Sonderzeichen: Hier kommen wir auch zum ersten Mal mit Kommentaren in Berührung. Kommentare werden beim Kompilieren nicht beachtet und dienen, so wie hier, zur Erklärung für den Autor. Alles was dem % folgt ist ein Kommentar. Was uns zu dem Problem bringt, wie man das % im Text benutzen kann. Ganz einfach: Man schreibt \%. Das Zeichen nach dem Backslash wird von L^AT_EX nicht interpretiert. Auch bei anderen Sonderzeichen (wie &, \$) und dem Leerzeichen ist das so.

1.3 Dokumentformatierungen

Listing 3: Formatierung 1

```
7 \pagenumbering{arabic} %arabische Seitenzahlen
8 \usepackage{ae} %Fuer skalierbare Schriften (in PDF)
```

In Zeile 7 unseres Dokumentes haben wir nun die Art der Seitennummerierung festgelegt. Arabische Zahlen sind sicher Standard, aber auch Buchstaben (Alph, alph) oder römische Zahlen (Roman, roman) sind möglich. Zeile 8 wird unbedingt benötigt, wenn man das Dokument nicht nur drucken, sondern auch in ein PDF umwandeln, am Bildschirm ausgeben³ und plattformunabhängig bleiben möchte.⁴

1.4 Zeichenkodierung

Deutsche Umlaute und Sonderzeichen sind in der englischen Sprache auch nicht vorgesehen gewesen. Da diese Zeichen im ASCII-Zeichensatz nicht vorkommen, muß noch die Kodierung der Datei (bzw. des Betriebssystems oder Editors) mitgegeben werden, um sie im Fließtext nutzen zu können. Bei mir sieht das z.B. so aus:

Listing 4: Zeichensatz

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

Achtung: Ältere Betriebssysteme und Microsoft®WindowsTM benutzen noch kein UTF, sondern eine ISO-8859-X Codierung. Hier muß man dem Encoder dann *latin1* mitgeben, um keinen Schriftsalat zu bekommen, MacTMUser brauchen wiederum manchmal *latin2*.

2 Titelseite

Bevor es jetzt richtig los geht, übergeben wir dem Dokument noch Namen von Praktikanten, Versuch und Betreuer sowie das Datum.

Listing 5: Formatierung 2

```
10 \title{Lorem ipsum}
11 \author{Max Musterstudent}
12 \date{19. Januar 2008}
13 \publishers{Betreuer: Dr. Hilfreich}
14 \subject{Grundpraktikum I}
```

Für das aktuelle Datum gibt es da noch das Makro \today. Allerdings empfiehlt sich das für unser Protokoll nicht, da wir festhalten wollen wann der Versuchstag war und nicht, wann wir das Protokoll geschrieben haben. Anzumerken ist hier, dass noch diverse Dinge

³TeX-Fonts werden bei der Formatierung eines Textes als Bitmaps in die erzeugte Postscript- bzw. PDF-Datei eingebettet. Diese Bitmaps sind aber für Auflösung des Ausgabegeräts berechnet (in der Regel sind das 600 dpi). Wenn ein solches PDF am Bildschirm betrachtet werden soll, müssen die Bitmaps auf 72 dpi skaliert werden, was zu einem hässlichen Schriftbild mit ausgefransten Zeichenformen führt. Dieses Paket realisiert die automatische Anpassung an skalierbare Fonts.

⁴Lösungen mit proprietären Schriftarten gibt es zwar auch, diese setzen aber die Installation dieser Schriftarten auch beim Betrachter zwingend voraus und sind deshalb für den wissenschaftlichen Bereich und wegen fehlender Vor- und Rückwärtskompatibilität ungeeignet.

in den Titel geschrieben werden können. Dies soll aber nicht Inhalt unserer Betrachtung sein, daher verweise ich auf den [scrguide](#).

Listing 6: Anfang

```
15 \begin{document}
16 \maketitle
17 \tableofcontents %Das Inhaltsverzeichnis
18 \chapter{Grundlagen}
```

Mit `\begin{document}` startet, unser Text. In Zeile 16 wird aus den im Listing 5 angegebenen Informationen die Titelseite automatisch formatiert und erstellt. Daher ist es wichtig, dass der Aufruf `\maketitle` hinter diesen Informationen auftaucht. Alles was danach kommt, wird nicht mehr berücksichtigt. In Zeile 17 wird auch gleich noch das Inhaltsverzeichnis auf der nächsten Seite erstellt. Das Kapitel 1 wird anschließend begonnen und trägt den Namen *Grundlagen*.⁵

3 Textformatierungen

Ab hier beginnt nun das tatsächliche Dokument. Wie bekannt sein sollte, formatiert \LaTeX den Text selbst. Man muß lediglich die Unterteilung⁶ festlegen. Manchmal möchte man das formatieren aber selbst übernehmen. Insbesondere Absatz- und Seitenumbrüche oder Tabulatoren und Leerzeichen. Wie das geht, sieht man, wenn man etwas mit Fließtext herumspielt. Eine Mini-Referenz der möglichen Befehle findet sich im Abschnitt 10 dieses Tutorials.

Manuell im Editor vorgenommene Umbrüche haben keine Auswirkungen auf das Dokument. So etwas würde man mit den Befehlen `\newline` bzw. `\\` oder `\newpage` erreichen. Bei einem doppelten Zeilenumbruch entsteht allerdings ein neuer, eingerückter Absatz. Eine andere sichtbare Trennung gibt es nicht. Wer diese will, sollte sich über den Befehl `\vfill` informieren oder einfach mal ausprobieren.

4 Umgebungen

Umgebungen werden zur Platzierung eines Objektes auf einer Seite benutzt. Sie alle haben gemeinsam, dass man durch sie die Art, Eigenschaften und Position eines Objektes festlegen kann. Im Allgemeinen tut man das während der `\begin` Anweisung: `\begin{Art}[Eigenschaft]{Position}`.

Als eine der Eigenschaften, kann die Größe in einem beliebigen Längenmaß [`width=2.5cm`], bei Bildern auch in absoluten Prozent [`scale=0.3`] oder relativen Prozent zur Textbreite [`width=0.5\textwidth`] angegeben werden. Eine weitere Eigenschaft stellt die Beschreibung des eingefügten Objektes dar, welche mit [`caption=Beschreibungstext`] eingefügt werden kann. Mehrere Eigenschaften (außer den in Abschnitt 5 erklärten Referenzen) werden mit Kommata separiert, [`caption=Bild 1, width=12cm`] ist also kein Problem.

Die Position wird durch die Buchstaben h (here), b (bottom), t (top) und p (page) festgelegt. Für \LaTeX ist das allerdings nur ein Vorschlag. Mit einem Ausrufezeichen kann diesem Vorschlag mehr Nachdruck verliehen werden, ob sich \LaTeX daran hält, steht auf einem ganz anderen Blatt. Im Folgenden werden hier die wichtigsten Umgebungen gezeigt.

⁵Es gehört sich schließlich, dass man zu Beginn des Protokolls die Grundlagen des Versuchs erläutert, nicht wahr?

⁶Wie bereits in 1.1 erwähnt, gibt es verschiedene Einteilungen eines Dokuments. Je nach benutzter Klasse gibt es *part* und *chapter*, außerdem *section*, *subsection*, *paragraph* und *subparagraph*. Diese wird dann sogar bis zu einer voreingestellten Tiefe automatisch in einem Inhaltsverzeichnis referenziert. Hängt man an diese Umgebung noch ein Sternchen (*), dann wird keine Nummerierung angezeigt und der Abschnitt erscheint auch nicht im Inhaltsverzeichnis.

4.1 Minipages

Eine Minipage ist eine Seite auf einer Seite. Sie wird meistens dazu benutzt andere Umgebungen nebeneinander zu setzen oder zu zentrieren. Sie kann aber auch einfach nur Fließtext enthalten. Da sie in den anderen Beispielen auftaucht, sei einfach auf diese verwiesen. Setzt man keine Minipages ein, dann werden die folgenden Umgebungen standardmäßig an den linken Rand gesetzt.

4.2 Tabellen

Listing 7: Tabelle

```
19 \begin{table}[hbt]
20 \begin{minipage}[hbt]{14cm}
21 \centering
22 \begin{tabular}{l|c||c|r}
23 1 & 2 & 3 & 4\\\hline
24 5 & 6 & 7 & 8\\\hline
25 9 & 10 & & \\\hline
26 11 & 12 & 13 & 14\\\hline
27 15 & 16 & 17 & 18\\
28 \end{tabular}
29 \caption{\small Ut wisi enim ad minim veniam}
30 \end{minipage}
31 \end{table}
```

Durch die *table* Umgebung wird hier ein Tabellenobjekt definiert, in der eine Tabelle und eine Beschreibung (`\caption{}`) untergebracht wurde. Dies kann man innerhalb eines größeren Objektes so machen. Die Position ist egal [hbt], die Positionsangabe wird durch die *table* Umgebung möglich. Durch die eingebundene Minipage und das `\centering` Kommando wird die Tabelle innerhalb des Objektes auf der Seite zentriert, ansonsten wäre sie linksbündig. Die Breite der Umgebung beträgt 14cm.

In der *tabular* Umgebung wird das Aussehen der Tabelle definiert. Der Text in den linken Zellen ist linksbündig (l), in den mittleren beiden Spalten zentriert (c) und rechts rechtsbündig (r). Es gibt zwischen den Spalten einfache (|) und doppelte (||) senkrechte Trenner. Spalten werden durch den Tabulator & voneinander getrennt. Am Ende jeder Spalte (außer der letzten) ist ein Zeilenumbruch und eine horizontale Trennlinie (`\hline`) zu finden. Das Ergebnis kann in Abbildung 1 betrachtet werden.

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |
| 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 |

Tabelle 1.1: Ut wisi enim ad minim veniam

Abbildung 1: Die Tabelle aus Listing 7

Zu Tabellen lässt sich noch sagen, dass sich senkrecht durchnummerierte Tabellen leichter warten und zum Beispiel in einem Plotter weiterverarbeiten lassen. Bei wenig ver-

schiedenen, abhängigen Werten kann das natürlich Platzverschwendung sein. \LaTeX ist die Formatierung jedenfalls egal und es lässt uns freie Hand.

4.3 Bilder

Eins gleich vorweg: Mit der von uns verwendeten Einstellung im Header (dvips) benutzen wir `latex` und nicht `pdftex` zum kompilieren. `latex` verwendet am liebsten Grafiken im Postscript Format, also `.ps` oder `.eps` Dateien. Darauf sollte geachtet werden, da sonst Fehler bei der Übersetzung ausgegeben werden (die meist aber nicht weiter schlimm sind). `pdftex` arbeitet dagegen mit `.jp(e)g` und `.tiff` und kann keine Postscript Dateien einbinden. Näheres dazu [hier](#). Wir gehen jetzt davon aus, dass unsere Grafiken `wave.eps` und `delta.eps` heißen und schreiben den Code für beide Programme kompatibel. Die Datei liegt im gleichen Verzeichnis wie unser `.tex` Dokument.

Listing 8: Grafik 1

```

32 \section{At vero eos}
33 \begin{figure}[htb]
34 \centerline{\includegraphics[width=0.3\textwidth]{wave}}
35 \caption{\small Wellen}
36 \end{figure}

```

In unserem neuen Abschnitt haben wir nun also die Grafik mit der `figure` Umgebung eingebunden und zentriert. Außerdem ist sie auf 30% der Seitenbreite skaliert worden. Ebenso wie bei `table` kann natürlich auch hier auf die Umgebungen verzichtet und die Grafik einfach direkt via `\includegraphics` eingebunden werden. Wie das aussieht, kann jeder selbst ausprobieren.

Als nächstes Beispiel setzen wir die Grafik und die Tabelle von oben nebeneinander.

Listing 9: Grafik 2

```

37 \begin{figure}[hbt]
38 \begin{minipage}[hbt]{7.2cm}
39 \centering
40 \includegraphics[width=6cm]{delta}
41 \caption{\small Eine Delta-Distribution}
42 \end{minipage}
43 \hfill
44 \begin{minipage}[hbt]{7.2cm}
45 \centering
46 \begin{tabular}{l|c|c|r}
47 1 & 2 & 3 & 4\\\hline
48 5 & 6 & 7 & 8\\\hline
49 9 & 10 & & \\\hline
50 11 & 12 & 13 & 14\\\hline
51 15 & 16 & 17 & 18\\
52 \end{tabular}
53 \caption{\small Ut wisi enim ad}
54 \end{minipage}
55 \end{figure}

```

Die Summe der Breiten beider Minipages beträgt 14,4cm. Das ist der Platz zwischen den Seitenrändern. Auch daran muß man sich nicht halten, wenn man zu groß wird gibt das Programm eine Warnung aus. Diesmal habe ich der Grafik eine feste Breite von 6cm zugewiesen. Zwischen Tabelle und die Grafik wurde mit `\hfill` ein horizontaler Zwischenraum eingefügt. Fließtext ließe sich außerdem einfach zwischen die Umgebungen schreiben. Das Ergebnis unserer Bemühungen zeigt die Abbildung [2](#).

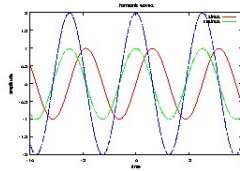


Abbildung 1.1: Wellen

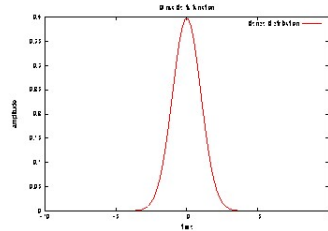


Abbildung 1.2: Eine Delta-Distribution

| | | | |
|----|----|----|----|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |
| 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 |

Abbildung 1.3: Ut wisi enim ad

Abbildung 2: Die Ergebnis aus Listings 8 und 9

Dem aufmerksamen Leser fällt darin vielleicht auf, dass die Tabelle nun als Abbildung bezeichnet wird. Grund dafür ist, dass wir sie in eine *figure* Umgebung eingebaut haben, statt *table* zu benutzen.

4.4 Aufzählungen

Dazu muß man nicht viel sagen. So wird es gemacht:

Listing 10: Aufzählung

```

56 \begin{itemize}
57 \item Lorem ipsum
58 \item Duis autem vel
59 \item Ut wisi enim
60 \end{itemize}

```

Für Aufzählungen gibt es noch die *enumerate* Umgebung. Die Tiefe der Schachtelung beträgt maximal 4. Die Aufzählungszeichen lassen sich verändern und auch weitere Listenumgebungen sind vorhanden, das würde hier aber den Rahmen sprengen.

4.5 Mathematische Umgebung

Es gibt mehrere davon. Im Fließtext benutzt man am besten $\$$. Man schreibt also z.B. $\mu \cdot a = 2\phi$ und erhält $\mu \cdot a = 2\phi$. Will man in einer mathematischen Umgebung normalen Text einbauen, so benutzt man $\text{\mbox{}}$. Das ist vor allem bei der Verwendung von Einheiten sinnvoll, die man schließlich nicht mit Variablen verwechseln soll.

Dann gibt es noch die Gleichungsumgebungen *align*, die man für lange Rechnungen benutzt und *equation*, mit der man eine abgesetzte Zeile für die Rechnung erhält. Beide setzen an den Rand jeder Zeile eine Zahl als Referenz. Das kann man, wie bereits in der Fußnote auf Seite 5 erklärt durch die Benutzung eines Sternchens vermeiden.

Man vergleiche die Ergebnisse der beiden Umgebungen in Abbildung 3:

Listing 11: Gleichungen

```

61 \begin{equation}
62 \mu \cdot a = 2 \phi
63 \\\ = \frac{b}{2}

```



```

64 \\\Rightarrow \phi = \frac{b}{4}
65 \end{equation}
66
67 \begin{align*}
68 \mu \cdot a &\stackrel{2}{=} \phi \\
69 &= \frac{b}{2} \\
70 \\\Rightarrow \phi &= \frac{b}{4}
71 \end{align*}

```

$$\mu \cdot a = 2\phi = \frac{b}{2} \Rightarrow \phi = \frac{b}{4} \quad (1.1)$$

$$\begin{aligned} \mu \cdot a &= 2\phi \\ &= \frac{b}{2} \\ &\Rightarrow \phi = \frac{b}{4} \end{aligned}$$

$$\mu \cdot a = 2\phi = \frac{b}{2} \Rightarrow \phi = \frac{b}{4}$$

Abbildung 3: Die Ergebnisse aus den Listings 11 und 12

Die Zeilenumbrüche in der *equation* Umgebung werden ignoriert und bei *align** findet besagte Nummerierung nicht statt. Wer sich fragt, was die Kaufmanns-Und vor dem Gleichheitszeichen bedeuten: Das sind Tabulatoren, genau wie in Listing 7, mit denen die Gleichungen untereinander ausgerichtet werden können.

Übrigens gibt es für eine *equation** Umgebung auch eine Abkürzung. Benötigt man die Referenz nicht, kann diese auch einfach folgendermaßen geschrieben werden:

Listing 12: Gleichungen

```

72 \[
73 \mu \cdot a = 2 \phi
74 \stackrel{2}{=} \frac{b}{2}
75 \\\Rightarrow \phi = \frac{b}{4}
76 \]

```

Eine linksbündige *align** Umgebung bekommt man mit $\left(\dots \right)$. Einfach damit rumexperimentieren.

Was die einzelnen Befehle bedeuten erklärt die Tabelle 3 im Abschnitt 10. Für eine ausführlichere Beschreibung mathematischer Symbole würde ich allerdings eher auf [Wikibooks](#) verweisen.

5 Referenzen

Referenzpunkte werden mit $\text{\label{key}}$ definiert, wobei *key* für ein bestimmtes Schlüsselwort steht. Dies kann in den Eigenschaften eines Objektes oder im Fließtext geschehen. Sinnvoller weise sollte man, wenn man sich darauf beziehen möchte, nummerierte Abschnitte und Objekte benutzen, damit der Leser diese auch wiederfindet. Es gibt dann zwei Möglichkeiten. Man kann sich entweder auf einen Abschnitt oder ein Objekt beziehen, dafür nutzt man $\text{\ref{key}}$. Die zweite Möglichkeit ist die Referenzierung einer Seite, dafür gibt es dann $\text{\pageref{key}}$. Hier ein Beispiel:

Listing 13: Referenz

```

77 \chapter{Neues Kapitel}
78 \label{neues} Hier steht ein sinnvoller Text auf den wir uns
79 noch beziehen werden.
80 ...
81 \section{Einige Seiten weiter}
82 Hier beziehen wir uns auf Kapitel \ref{neues} auf
83 Seite \pageref{neues}.

```

L^AT_EX setzt dann die zu ergänzenden Informationen automatisch ein. Insbesondere bei langen, sich stark verändernden Texten sehr angenehm, da man sich dann nicht mehr darum kümmern muß. Existiert ein *label* nicht, oder wurde falsch geschrieben, werden beim kompilieren Warnungen ausgegeben und im Ergebnis ein paar Fragezeichen angezeigt.

6 Abschluß

Damit sind wir am Ende unseres Beispielprotokolls. Um das hier noch mal herauszustellen: Die Diskrepanz zwischen Meß- vom Sollwert war ganz klar insignifikant. Nachdem das auch noch geklärt ist, schließen wir das Dokument ab.

Listing 14: Ende

```

84 \end{document}

```

7 Kompilieren

Wie das Kompilieren des Quellcodes vor sich geht, hängt meist vom benutzten System ab. Wenn man einen speziellen L^AT_EX-Editor verwendet, übernimmt dieser das. Für hartnäckige Kommandozeilenbenutzer genügt auch der Aufruf von `latex Protokoll.tex` im Verzeichnis, in dem sich die Datei `Protokoll.tex` befindet. L^AT_EX wandelt diese in eine DVI Datei um, welche aber nicht portabel ist.⁷ Eine sinnvolle Weiterverarbeitung bietet das Tool `dvips` an, das im T_EX Paket enthalten ist. Dieses Programm wandelt das DVI in eine Postscript Datei um, die sich auch auf anderen Rechnern mit Hilfe von [Ghostscript](#) problemlos betrachten und ausdrucken lässt. Für die Umwandlung in PDF Dateien sind die Tools `dvipdf` und `dvipdfm` ebenfalls vorhanden.

Wichtig zu wissen ist, dass bei der Verwendung eines Inhaltsverzeichnisses oder von Referenzen die Dateien mit der Endung `.toc` und `.aux` im `latex`-Lauf neu geschrieben werden, deren Änderungen aber noch nicht in die auszugebene DVI-Datei einfließen. Daher ist es in diesen Fällen wichtig, den Lauf ein zweites mal zu wiederholen.

8 Tricks und Kniffe

8.1 Dateien einbinden

Den Header muß man nicht jedes mal neu schreiben. Es reicht auch diesen über das `\input` Kommando in eine leere Datei einzubinden, d.h. man schreibt ihn z.B. in eine Datei `header.tex`. Dann bindet man ihn in einer leeren Datei immer folgendermaßen ein:

```

1 \input{/Pfad/zur/Datei/header.tex}
2 ...
3 dein Text

```

⁷Beispielsweise sind Grafiken nicht fest eingebunden und es wird auch die gleiche Version des T_EX Paketes auf einem anderen System vorausgesetzt, damit dieser die Datei lesen kann.

Kopiert man die Datei ins lokale Verzeichnis, kann man auf den Pfad verzichten.

Natürlich lässt sich das auch umgekehrt mit dem Body machen. Dann kann man sogar jedes Kapitel in eine eigene Datei schreiben, was sehr hilfreich ist, wenn man die Übersicht bei größeren Texten behalten will. Also so:

```

4 \documentclass [] {}
5 \usepackage [] {}
6 ...
7 \begin{document}
8 \input{kapitel1.tex}
9 \input{kapitel2.tex}
10 ...
11 \end{document}

```

Der Text in den Dateien wird durch `\input` einfach nur da eingehängt, wo man es hinschreibt. So erhöht sich neben der Übersichtlichkeit auch die Flexibilität.

8.2 Seitenränder

Wem die Formatierung der Seiten nicht gefällt, dem sei das `geometry` Paket ans Herz gelegt. Natürlich existieren noch diverse andere Hacks, diese gelten aber als unsauber. Die Syntax des Paketes ist selbsterklärend. Die Anweisung kann übrigens auf jeder Seite verändert werden. Als globale Einstellung gehört die Anweisung aber in den Header.

```

1 \usepackage{geometry}
2 \geometry{a4paper, left=4cm, right=2cm, top=2cm, bottom=2cm}

```

8.3 Nummerierte Tabellen

Viele Leute schreiben gern laufende Nummern in ihre Tabellen. Um das nicht per Hand machen zu müssen, oder bei maschinell anfallenden Daten, die man automatisch nachbearbeiten möchte, oder einfach nur um nicht Daten mit Formatierungen im Quelltext durcheinander zu bringen, kann \LaTeX das Zählen für den Autor übernehmen. Hierfür legt man einen neuen Zähler an und setzt ihn auf Null. Das Makro `\newcounter{}` übernimmt das für einen Zähler dessen Namen man in der Klammer festlegt.

In einem zweiten Schritt legt man ein neues Kommando fest, dass das Zählen übernimmt. Gezählt wird mit dem Makro `\stepcounter`. Das Neue Kommando wird sinniger Weise mit `\newcommand` angelegt.

```

3 \newcounter{number} %Zaehler initialisieren
4 \newcommand{\n}{\stepcounter{number}\arabic{number}}

```

Wie man sieht, werden hier arabische Zahlen benutzt. Der Name des Kommandos ist frei wählbar. Nachdem eine Tabelle (oder wo auch immer gezählt wird), durchnummeriert wurde und man von nun vorn beginnen möchte, muß man kein neues Kommando definieren. Mit Hilfe von `\setcounter{Zaehler}{Wert}` lässt sich der Zähler nämlich auf einen beliebigen Wert setzen. Bei mehrspaltigen Tabellen kann man so auch mehrere Zähler gleichzeitig laufen lassen, die verschiedene Startwerte hatten.

Nachdem diese Vorarbeit erledigt ist, kann man Tabellen nach folgendem Muster anlegen:

```

5 \begin{table}[hbt]
6 \centering
7 \begin{tabular}{l|r|r|r|r|r|r|r}
8 lfd.Nr. & \n & \n & \n & \n & \n & \n & \n \\\hline
9 a & 1,2 & 2,3 & 3,4 & 4,4 & 5,6 & 6,4 & 7,6\\\hline
10 $\varphi$ & 15 & 30 & 45 & 60 & 75 & 90 & 105

```

```

11 \end{tabular}
12 \end{table}

```

Das `\n` wird in diesem Beispiel durch eine laufende Nummer substituiert und es entsteht eine Tabelle wie in Abbildung 4.

| lfd. Nr. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------|-----|-----|-----|-----|-----|-----|-----|
| a | 1,2 | 2,3 | 3,4 | 4,4 | 5,6 | 6,4 | 7,6 |
| φ | 15 | 30 | 45 | 60 | 75 | 90 | 105 |

Abbildung 4: Automatisch durchnummerierte Tabelle

8.4 Darstellung von Formeln

Außerhalb der mathematischen Umgebungen sehen Formeln in der Regel folgendermaßen aus: $\sum_{k=0}^{\infty} q^k = \frac{1}{1-q}$.⁸ Das macht so auch Sinn, da sonst die Zeilenabstände und die Textformatierung total durcheinander gebracht werden. Trotzdem kann es Situationen geben, in denen man das nicht will. Dafür gibt es die `\displaystyle{}` Umgebung. Formeln sehen damit wieder so aus:
$$\sum_{k=0}^{\infty} q^k = \frac{1}{1-q}.$$

8.5 Absätze

Zur Formatierung von Absätzen wäre noch etwas zu sagen. Wie in Abschnitt 3 beschrieben, werden diese standardmäßig eingerückt werden. Die Tiefe, wieweit eingerückt werden soll, kann man über das Makro `\parindent` einstellen. Möchte man auf ein Einrücken von Absätzen verzichten, so schreibt man in den Header:

```

13 \parindent0pt %Absatz eingerueckt um 0 Punkte

```

Außerdem wird **immer** Blocksatz benutzt. Es kann natürlich vorkommen, dass Flattersatz vom Betreuer/Prüfer vorgeschrieben wird. Wichtig ist, dass das Schriftbild insgesamt einheitlich ist, daher benutzt man für eine Änderung dieses Parameters folgendes am Dokumentenbeginn und analog mit `\end{}` am Ende:

- `\begin{flushleft}` – nach links ausgerichteten Flattersatz
- `\begin{flushright}` – nach rechts ausgerichteten Flattersatz
- `\begin{center}` – zentrierter Flattersatz

\LaTeX trennt ihm bekannte Wörter für diese Formatierung automatisch. Bei zusammengesetzten oder Fremdwörtern, sowie bei Wörtern die Umlaute enthalten, funktioniert das manchmal nicht. Solche Wörter müssen dann so geschrieben werden, dass \LaTeX weiß, wo es sie zu trennen hat. Das geschieht durch das Einfügen von `\-` an der Stelle im Wort, an der Silbentrennung möglich ist. Man schreibt also z.B. das Wort Phasenbrechungsindex als `Pha\-\sen\-\brech\-\ungs\-\in\-\dex`.

⁸für $|q| < 1$

9 Weiterführende Links

- [Deutschsprachige Anwendervereinigung TeX e.V](#)
- [FAQ der DAnTe e.V.](#)
- [L^AT_EX-Kompendium](#) bei Wikibooks
- [Komascript.de](#)
- [Deutsche T_EXGruppe im Usenet de.comp.text.tex](#)
- [T_EX Users Group \(englisch\)](#)
- [LaTeX - eine Einführung und ein bißchen mehr...](#)
- [OpenOffice Calc2LaTeX](#)
- [Excel2Latex](#) (bis Excel97)

10 Befehlsreferenz

Diese Tabellen sollen einen kurzen Überblick verschaffen. Auf keinen Fall wird Anspruch auf Vollständigkeit erhoben.

| Befehl | Bedeutung | Befehl | Bedeutung |
|--------------------------|----------------------|---------------------------|--------------------|
| <code>\newline</code> | Zeilenumbruch | <code>\newpage</code> | Seitenumbruch |
| <code>&</code> | Tabulator | <code>~</code> | Leerzeichen |
| <code>\$</code> | Math. Umgebung | <code>\underline{}</code> | Unterstreichen |
| <code>\textbf{}</code> | fette Schrift | <code>\textit{}</code> | kursive Schrift |
| <code>\small</code> | kleine Schrift | <code>\large</code> | große Schrift |
| <code>\footnote{}</code> | Fußnote | <code>\caption{}</code> | Beschreibungstext |
| <code>\hline</code> | waagerechter Strich | <code>\hfill</code> | horizontaler Raum |
| <code>\label{key}</code> | setzt Referenz „key“ | <code>\ref{key}</code> | referenziert „key“ |

Tabelle 1: Formatierungsbefehle

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

Tabelle 2: Griechisches Alphabet

Bei den großen griechischen Buchstaben schreibt man den ersten Buchstaben groß, kursive Buchstaben (falls vorhanden) bekommt man durch Benutzung des Präfix **var**. Bei `\phi` wäre das also `\varphi`.

| | | | | | | | |
|------------|-----------------------|---------------|--------------------------|-----------------|----------------------------|-----------------|----------------------------|
| \sum | <code>\sum</code> | \prod | <code>\prod</code> | \hbar | <code>\hbar</code> | ∇ | <code>\nabla</code> |
| ∂ | <code>\partial</code> | \int | <code>\int</code> | \oint | <code>\oint</code> | $\frac{1}{2}$ | <code>\frac{1}{2}</code> |
| \div | <code>\div</code> | \neg | <code>\neg</code> | \wedge | <code>\wedge</code> | \vee | <code>\vee</code> |
| \pm | <code>\pm</code> | \mp | <code>\mp</code> | \cdot | <code>\cdot</code> | \times | <code>\times</code> |
| \sqrt{y} | <code>\sqrt{y}</code> | $\sqrt[n]{x}$ | <code>\sqrt[n]{x}</code> | \equiv | <code>\equiv</code> | \sim | <code>\sim</code> |
| X^Y | <code>X^Y</code> | 360° | <code>360^\circ</code> | \overbrace{z} | <code>\overbrace{R}</code> | \underline{P} | <code>\underline{P}</code> |
| \neq | <code>\neq</code> | \rightarrow | <code>\rightarrow</code> | \Rightarrow | <code>\Rightarrow</code> | ∞ | <code>\infty</code> |
| | | \mathbb{N} | <code>\mathbb{N}</code> | \mathbb{Z} | <code>\mathbb{Z}</code> | \mathbb{R} | <code>\mathbb{R}</code> |

Tabelle 3: Symbole

11 Lizenz

Dieses Werk ist unter einem Creative Commons Namensnennung 2.0 Deutschland Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by/2.0/de/> oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.