# ${\ensuremath{\mathbb A}} T_{\ensuremath{\mathbb E}} X$ für blutige Anfänger

Rikka Mitsam

Nov. 2001

# Inhaltsverzeichnis

1	Vor	wort		3
<b>2</b>	Ŀ₽Ţ₽	X mit	KEdit	4
	2.1	KEdit		4
	2.2	Einfac	hes LATEX-Dokument	5
	2.3	Zeiche	m und Formatierung in	6
		2.3.1	Umlaute	6
		2.3.2	Symbole	6
		2.3.3	Anführungszeichen	7
		2.3.4	Absätze	7
		2.3.5	Querverweise	7
	2.4	Komp	lizierteres LAT <sub>E</sub> X-Dokument erstellen	8
		2.4.1	Listen erstellen	8
		2.4.2	Tabellen erstellen	8
		2.4.3	Bilder einfügen	9
		2.4.4	Inhaltsverzeichnis, Tabellen- und Abbildungsverzeichnisse	10
		2.4.5	Literaturverzeichnis	11
		2.4.6	Fußnoten, Randbemerkungen	11

# Tabellenverzeichnis

2.1	Dies ist einen	Tabelle .						•			•																	8	8
-----	----------------	-----------	--	--	--	--	--	---	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---

# Kapitel 1

# Vorwort

Diese kurze Einführung in  $IaT_EX$  habe ich für wirklich blutige Anfänger geschrieben. Es werden hier nur die wichtigsten Grundlagen von  $IaT_EX$  erklärt und zwar unter der Verwendung von KEdit unter Linux.

Anhand dieser kleinen Einführung sollte man in der Lage sein, Dokumente mit IATEX zu verfassen. Dies beinhaltet auch die Verwendung von Listen, Tabellen, Literatur- sowie Inhaltsverzeichnisse, ebenso wie das Einbinden von Bildern. Es werden nur die ersten Schritte mit IATEX erklärt, alles weitere sollte der geneigte Leser in einschlägiger Literatur bzw. im Internet finden.

Ich habe diesen Text eigentlich nur für eine Freundin als kleinen Anreiz, IATEX zu erlernen und für mich zum Nachschlagen verfasst. Wahrscheinlich wimmelt der vorliegende Artikel nur so von Fehlern - wer einen findet, darf ihn gerne behalten. ;-)

Und nun viel Spass allen, die gerne schön gesetzte Dokumente verfassen wollen!

# Kapitel 2

# IAT<sub>E</sub>X mit KEdit

### 2.1 KEdit

LATEX ist ganz einfach. Man nehme einen Editor, zum Beispiel KEdit, den man in einer Shell mit dem Befehl

#### kedit&

öffnet.

KEdit ist ein einfacher Editor, man braucht eigentlich keine Vorkenntnisse, um damit umzugehen. Man öffnet mit File open (strg+o) eine vorhandene Datei, mit File new schafft man eine neue Datei. Mit dem Befehl File save (strg+s) speichert man die gerade geöffnete Datei, mit File save as kann man die gerade offene Datei unter einem anderen Pfad oder Dateinamen speichern. File close (strg+c) schließt das gerade geöffnete Fenster, wenn nur ein Fenster offen war, wird damit KEdit geschlossen. Ansonsten beendet man den Editor mit File exit (strg+q).

Mit Edit copy (strg+c) kann man einen markierten Bereich kopieren, mit Edit cut (strg+x) schneidet man einen markierten Bereich aus und der Befehl Edit paste (strg+v) fügt den kopierten oder ausgeschnittenen Bereich wieder ein. File select all markiert das ganze Dokument, mit File insert file kann man eine ganze Datei einfügen, mit File insert date das aktuelle Datum. Mit File find kann man nach Worten suchen, mit File replace kann man Worte suchen und durch andere Worte ersetzen.

In der Tool-Leiste finden sich von links nach rechts die Befehle:

- Neues Dokument
- Dokument öffnen
- Dokument speichern
- Kopieren
- Einfügen
- Ausschneiden
- Drucken
- Mail
- Hilfe

Im Text selbst navigiert man mit den folgenden Short-cuts:

- Zeilenanfang: strg+a
- Zeilenende: strg+e
- eine Zeile runter: strg+n
- eine Zeile rauf: strg+p
- kopieren: strg+c
- $\bullet$  ausschneiden: strg+x
- einfügen: strg+v

## 2.2 Einfaches LATEX-Dokument

So schaut ein ganz normales LATEX-Dokument aus, es beginnt immer mit der Angabe der Dokumentklasse, (report, article, book oder letter sind die gängingen Klassen) dann wird angegeben, welche Pakete mit eingebunden werden sollen. Also in den meisten Fällen german für Deutsch und a4 für das Papierformat. Dann folgen die Überschriften von chapter bis subparagraph, häufig reichen die ersten 3 Stufen der Überschriften aus.

```
\documentclass{report}
\usepackage{german,a4}
\begin{document}
 \chapter{bla}
 text
  \section{blub}
   text
   \subsection{gnlpf}
    text
    \subsubsection{argl}
     text
     \paragraph{urks}
      text
      \subparagraph{Das war's}
       text
\end{document}
```

Wenn man nun ein neues Dokument geöffnet hat, ein bißchen Text eingegeben hat, speichert man das ganze mit der Dateiendung :tex; also. File - save as - dateiname.tex. Den Dateinamen gibt man bei Location ein, anstelle von .. hinter dem Dateipfad.

So, nach dem Speichern wechselt man in ein Konsol-Fenster. Dort geht man (wenn man das noch nicht getan hat) mit dem Befehl cd (change directory) in das Verzeichnis, in dem man das eben erstellte Dokument gespeichert hat. Also, das Dokument wurde unter Text/latex gespeichert, dann heißt der Befehl

```
cd Text/latex
```

(Mit ls -la schnell nachschauen, ob das gespeicherte Dokument auch wirklich da ist – wenn ja, weitermachen, ansonsten ist man entweder im falschen Verzeichnis oder man hat das Dokument wo anders gespeichert.)

Dann muß das eben erstellte Dokument (das mit der Endung \*.tex) übersetzt werden, das geht mit dem Befehl:

latex dokumentname.tex

das kann je nach Textlänge etwas dauern. Wenn ein  $\[mathbb{L}^{AT}\[mathbb{E}X\]$ -fehler im Dokument ist, wird er im Konsol-Fenster angezeigt, am besten hier gleich wieder zu KEdit wechseln, den Fehler beheben und im Konsol-Fenster mit der Returntaste weiter schauen, bis zum nächsten Fehler oder bis es fehlerlos übersetzt ist. Bei  $\[mathbb{L}^{AT}\[mathbb{E}X\]$ -fehlern, das Dokument verbessern, speichern und nochmal mit latex dokumentname.tex übersetzen.

Wenn keine Fehler mehr existieren, kann man sich das Dokument mit dem Befehl

xdvi dokumentname

(ohne Dateiendung!!) ansehen. Schaut doch schon ganz gut aus, man sollte stolz auf sich sein, wenn man es soweit geschafft hat.

Nun wird das Dokument weiter bearbeitet, wenn man fertig ist, möchte man vielleicht das Dokument, das im Moment als ps-Datei gespeichert ist, als pdf-Datei speichern, damit man es mit dem Acrobat-Reader ansehen kann. Das geht folgendermaßen:

Nachdem man mit dem cd-Befehl in das Verzeichnis gewechselt ist, in dem das Dokument liegt, tippt man den Befehl:

dvips -o dateiname.ps dateiname.dvi

danach noch den Befehl:

ps2pdf dateiname.ps dateiname.pdf

Nun kann man sich das Dokument im Acrobat-Reader ansehen. Das geht mit dem Befehl:

acroread dateiname.pdf

Nun hat man das Dokument in einem allgemeingültigem Standard gespeichert.

# 2.3 Zeichen und Formatierung in $PT_EX$

#### 2.3.1 Umlaute

Die deutschen Umlaute (ä, ö, ü) werden so eingegeben, wenn das Paket german eingebunden ist:

"andern, "ofters, "ubrigens, mu"s

#### 2.3.2 Symbole

Die Symbole %, \$, &, {, }, # erzeugt man so

\% \\$ \& \{ \} \#

#### 2.3.3 Anführungszeichen

Die deutschen Anführungszeichen erzeugt man mit

\glqq

Das waren linke Anführungsstrich, die rechten gehen mit

\grqq

Achtung! Nach einem solchen Ausdruck muß entweder ein Leerzeichen folgen, das unterdrückt wird, oder ein Satzzeichen. Das Leerzeichen ist notwendig, damit  $IAT_EX$  erkennen kann, wann der Befehl glqq abgeschlossen ist.

#### 2.3.4 Absätze

Solange man nichts anderes angibt, werden Texte im Blocksatz gesetzt. Einen Absatz erzeugt man mit \\, doppelte Leerzeichen oder Leerzeilen werden einfach verschluckt.

Einen Absatz (oder ein Bild oder eine Tabelle) kann man mit

```
\begin{center}
  text, text, text
\end{center}
```

zentrieren. Um einen Absatz linksbündig mit rechtem Flatterrand zu erzeugen, tippt man statt center einfach flushleft, für rechtsbündigen Text mit linkem Flatterrand nimmt man flushright.

#### 2.3.5 Querverweise

Um einen Verweis auf eine bestimmte Stelle weiter oben (oder weiter unten) im Text zu erhalten, fügt man Markierungen ein:

\label{marke}
\ref{marke}
\pageref{marke}

Die Markierung heißt label, in den geschweiften Klammern benennt man eine Marke (Bsp.: elch) und mit ref greift man dann (weiter unten im Text) darauf zu und erzeugt damit einen Verweis auf die Kapitelnummer, in der die label-Marke steht, zum Beispiel:

```
...der Elch\label{elch}, blah blubber...
...wie oben in Kapitel \ref{elch} bereits erw"ahnt, hat der Elch...)
```

Mit pageref erzeugt man einen Verweis auf die Seitenzahl, in der die label-Marke ist.

## 2.4 Komplizierteres LATEX-Dokument erstellen

### 2.4.1 Listen erstellen

Eine einfach Aufzählung von Elementen macht man so:

```
\begin{itemize}
  \item Ein Punkt
  \item Noch ein Punkt
  \item[$\circ$] Ein anderer Punkt
  \end{itemize}
```

Itemize ist eine einfache Aufzählung und item sind die einzelnen Punkte. Dann gibt es noch die nummerierte Aufzählung:

```
\begin{enumerate}
  \item Erster Punkt
  \item Zweiter Punkt
  \end{enumerate}
```

Das schaut übersetzt so aus:

- 1. Erster Punkt
- 2. Zweiter Punkt

Das war noch ganz einfach, somit wird der Rest des komplizierten Dokuments auch nicht viel schwieriger werden...

#### 2.4.2 Tabellen erstellen

Tabellen sind eigentlich fürchterlich einfach zu erstellen. Man benötigt nur den folgenden Befehl – hier ein Beispiel für eine Tabelle mit 3 Spalten und 3 Reihen:

```
\begin{table}[h]
 \begin{center}
 \begin{tabular}{ccc}
  text & text & text\\
  text & text & text\\
  text & text & text\\
        text & text & text\\
        text & text & text\\
        \end{tabular}
        \caption{Dies ist eine Tabelle\label{testtabelle}}
        \end{table
```

Das Ganze schaut dann in echt so aus wie Tabelle 2.1. Zur Erklärung: table heißt,

$\operatorname{text}$	$\operatorname{text}$	$\operatorname{text}$
$\operatorname{text}$	$\operatorname{text}$	$\operatorname{text}$
$\operatorname{text}$	$\operatorname{text}$	$\operatorname{text}$

Tabelle 2.1: Dies ist einen Tabelle

daß jetzt etwas kommt, was eine Tabelle sein soll, das [h] erzwingt, daß die Tabelle

auch genau hier im Text erscheinen soll, ohne dem h<br/> kann es passieren, daß die Tabelle da im Text erscheint, wo<br/>  ${\rm LATEX}$  meint, daß sie schön aussieht. (Das Ganze hat mit Gleitobjekten zu tun...) Durch center wird die Tabelle zentriert.

Dann kommt endlich die richtige Tabelle: tabular bedeutet Tabelle, das ccc heißt, daß die Spalten zentriert werden. Alternativ dazu gibt es l für linksbündig und r für rechtsbündig. Die Anzahl der c's entsprechen der Anzahl der Spalten, diese werden dann im Text mit & getrennt. Die einzelnen Zeilen oder Reihen werden mit \\ getrennt.

Um einen Rahmen um die einzelnen Zellen zu machen, setzt man einen | vor und zwischen die c's: also  $\{|c|c|c|\}$ , das erzeugt vertikale Linien. Um horizontale Linien zu bekommen, muß man nur den Befehl

\hline

einfügen, und zwar immer dort, wo man eine Linie haben will. Damit die Tabelle später im Tabellenverzeichnis erscheinen kann, tippt man unter die Tabelle wie oben im Beispiel diesen Befehl:

```
\caption{Das ist eine Tabelle\label{testtabelle}}
```

Die Sache mit labeltext hat zur Folge, daß man sich im fortlaufenden Text auf genau diese Tabelle 2.1 beziehen kann. Der Bezug wird durch

```
\ref{testtabelle}
```

hergestellt, man tippt ihn im fortlaufenden Text. Und?! War doch gar nicht so schwierig, oder?

#### 2.4.3 Bilder einfügen

Um Bilder einfügen zu können , muß erst das Paket epsfig eingebunden werden (was?!?!), nein, so schwierig ist das gar nicht, am Anfang jedes  $IAT_EX$ -Dokuments binden wir mit dem Befehl:

```
\usepackae{german,a4,epsfig}
```

dieses Paket mit ein. (Puh, war ja wirklich nicht schwer!) Dann wollen wir ein Bild - aber nur im \*.ps-Format (also bild.ps) einfügen, dazu speichern wird das gewünschte Bild erstmal in dem Verzeichnis, in dem sich auch das IAT<sub>F</sub>X-Dokument befindet. (Mit cp, wie oben beschrieben!) Dann kommt das:

```
\begin{figure}
  \begin{center}
   \epsfig{file=bild.ps, width=8cm}
   \caption{Ein kleines Bild, mit Label\label{bildchen}
   \end{center}
  \end{figure}
  ...
  wie schon in Abb. \ref{bildchen} zu sehen ist
  ...
```

Zur Erklärung: figure heißt, daß nun ein Bild kommt, center bedeutet, daß das Bild zentriert werden soll (also in der Mitte der Seite), dann kommt das Bild selber mit der Angabe, wie breit es sein soll. Mit dem caption-Befehl erreicht man, daß das Bild im Abbildungsverzeichnis auftaucht und mit labelbildchen erzeugt man ein Label auf das man mit refbildchen in einem Text auf das Bild verweisen kann.

### 2.4.4 Inhaltsverzeichnis, Tabellen- und Abbildungsverzeichnisse

Nachdem man jetzt schon fähig ist, ein einfaches LATEX-Dokument zu erstellen, ist der nächste Schritt, ein etwas komplizierteres Dokument zu erstellen. Als erstes erstellt man ein ganz normales LATEX-Dokument wie oben gelernt. Dann wird ein Inhaltsverzeichnis eingefügt.

Ein Inhaltsverzeichnis wird mit dem Befehl

```
\tableofcontents
```

erstellt. Dieser Befehl steht entweder ganz am Anfang, also nach

```
\begin{document}
   \tableofcontents
```

oder eben am Ende. Wenn man nun schon einige Kapitel Text geschrieben hat, will man sich das Ganze ansehen. Man tippt wie gewohnt die Befehle

```
latex dateiname.tex xdvi dateiname
```

und sieht erstmal nichts. Jetzt nicht wundern oder verzweifeln, das ist ganz normal. Man muß das Dokument einfach nur zweimal mit

#### latex dateiname.tex

übersetzen, und zwar gleich hintereinander, dann erst mit x<br/>dvi ansehen. Und – o $\rm Wunder$  – es funktioniert!

Ein Tabellenverzeichnis funktioniert genauso wie die Inhaltsangabe, hier heißt der Befehl:

#### \listoftables

Wie eben werden auch hier die einzelnen Tabellen (sofern sie alle korrekt sind) aufgelistet.

Fehlt nur noch das Abbildungsverzeichnis, das eine Liste aller Abbildungen erzeugt, hier heißt der Befehl

#### \listoffigures

### 2.4.5 Literaturverzeichnis

Das Literaturverzeichnis erscheint ganz am Schluß des Dokuments (meistens) und schaut so aus:

```
\begin{thebibliography}{99}
\bibitem{Autorenk"urzel}
Autor, \emph{Titel}, Verlag, Jahr, \dots
\bibitem{Autorenk"urzel}
Autor, \emph{Titel}, Verlag, Jahr, \dots
\end{thebibliography}
```

Die 99 steht für die maximale Anzahl der Bücher, auf die man verweisen möchte, also bei 156 Büchern muß man 999 in die Klammer setzen, bei nur 3 Büchern eine 9. Hinter bibitem kommt das Autorenkürzel in geschweifte Klammern, also irgendeine Bezeichnung für den Autor. Im Text verweist man auf ein Buch mit

\cite{Autorenk"urzel}

### 2.4.6 Fußnoten, Randbemerkungen

Fußnoten sind wieder recht einfach, man setzt den Befehl

\footnote{Text, der in die Fu"snote rein soll}

in den fortlaufenden Text.

Randbemerkungen funktionieren ähnlich, der Befehl dazu heißt

\marginpar{Text, der an den Rand soll}

und der soll ebenso im fortlaufenden Text getippt werden.