

# Ein zweites Beispiel-Dokument in LaTeX

Elke Wilkeit

2003-05-29

## 1 Einleitung

Wer das erste Beispiel verstanden hat, sollte jetzt umgehen können mit

- Gliederungspunkten wie `\section`, `\subsection`, `\subsubsection`
- Verweisen auf die Nummern dieser Gliederungspunkte mittels `\ref` und `\pageref`
- Schriftarten
- Schriftgrößen

In diesem Text sollen nun in erster Linie *Aufzählungen* und *Befehle* gelernt werden. Wieder wird das Prinzip verfolgt, das zu Lernende bereits im Lehrtext zu benutzen.

## 2 Aufzählungen

Oben steht schon eine erste Aufzählung und im Quelltext ist zu sehen, wie sie gemacht ist: Es handelt sich um eine *Umgebung*, die mit `\begin{itemize}` eingeleitet und mit `\end{itemize}` beendet wird. Solche Aufzählungen können auch geschachtelt werden; dann bekommen die einzelnen Punkte in den tieferen Ebenen andere Symbole zugeordnet. Auch kann man die Symbole selbst wählen, aber für derartige Feinheiten verweise ich auf die Literatur: [Kop02] ist sehr verständlich und erklärt genau das, was man braucht; [GMS02] geht mehr auf spezielle Pakete ein und ist daher als Zusatz-Text für solche, die es genau wissen wollen, gut geeignet. Zum Durchlesen steht zu viel drin.

Wenn man in der Umgebung oben `itemize` durch `enumerate` ersetzt, erhält man eine *nummerierte Liste*. Auch hierzu sei ein Beispiel gegeben:

1. erster Punkt: Es ist gut, zwischen den *items* einer solchen Liste im Quelltext Leerzeilen zu lassen, das erleichtert spätere Korrekturen.
2. zweiter Punkt: Es ist Geschmacksache, ob man die einzelnen Einträge mit einem großen oder einem kleinen Buchstaben beginnen lässt, man sollte dabei aber konsistent sein. Wenn es sich um kurze Einträge handelt, bevorzuge ich die Form ohne Punkt am Ende und klein am Anfang, so wie hier:
  - (a) erster Unterpunkt des zweiten Punktes
  - (b) zweiter Unterpunkt des zweiten Punktes mit Verweis auf Punkt 3 mit den schönen Listen
3. dritter Punkt: Das schöne an diesen Listen ist, dass man die Reihenfolge ihrer Einträge ändern kann, ohne sich um die Nummerierung kümmern zu müssen.

Im folgenden dieselbe Liste (schau den Quelltext an), nur der letzte Punkt ist nach vorne kopiert worden:

1. dritter Punkt: Das schöne an diesen Listen ist, dass man die Reihenfolge ihrer Einträge ändern kann, ohne sich um die Nummerierung kümmern zu müssen.
2. erster Punkt: Es ist gut, zwischen den *items* einer solchen Liste im Quelltext Leerzeilen zu lassen, das erleichtert spätere Korrekturen.
3. zweiter Punkt: Es ist Geschmacksache, ob man die einzelnen Einträge mit einem großen oder einem kleinen Buchstaben beginnen lässt, man sollte dabei aber konsistent sein. Wenn es sich um kurze Einträge handelt, bevorzuge ich die Form ohne Punkt am Ende und klein am Anfang, so wie hier:
  - (a) erster Unterpunkt des zweiten Punktes
  - (b) zweiter Unterpunkt des zweiten Punktes mit Verweis auf Punkt 1 mit den schönen Listen

Hier merkt man gleich, dass es dumm war, die Reihenfolge der Punkte verbal fest zulegen, weil dann die Worte bei einem Vertauschen angepasst werden müssen. Der Verweis auf Punkt 1 kann einfach so bleiben, weil er automatisch angepasst wird, wenn wir das zweite Mal kompilieren. Für die Ober-Punkte ist es bequemer, die Nummern der automatischen Nummerierung zu benutzen, also noch einmal:

1. Punkt: Das schöne an diesen Listen ist, dass man die Reihenfolge ihrer Einträge ändern kann, ohne sich um die Nummerierung kümmern zu müssen.
2. Punkt: Es ist gut, zwischen den *items* einer solchen Liste im Quelltext Leerzeilen zu lassen, das erleichtert spätere Korrekturen.
3. Punkt: Es ist Geschmacksache, ob man die einzelnen Einträge mit einem großen oder einem kleinen Buchstaben beginnen lässt, man sollte dabei aber konsistent sein. Wenn es sich um kurze Einträge handelt, bevorzuge ich die Form ohne Punkt am Ende und klein am Anfang, so wie hier:
  - (a) Unterpunkt von Punkt 3
  - (b) Unterpunkt von Punkt 3 mit Verweis auf Punkt 1 mit den schönen Listen

Die *Labels*, mit denen die Verweise gemacht werden, müssen alle verschiedene Namen haben. Obwohl ich hier die Aufzählung zweimal (fast) wörtlich kopiert habe, musste ich den Labels, die im Text vor „Das schöne an diesen Listen ...“ stehen, verschiedene Namen (schoen1, schoen2, schoen3 — siehe Quelltext) geben.

## 3 Befehle

Wir können in  $\text{\LaTeX}$  verschiedene Arten von Befehlen unterscheiden:

- Befehle, die etwas einstellen
- Befehle, die auf etwas wirken
- Befehle, die eine Wirkung haben

### 3.1 Befehle, die etwas einstellen

Wir haben schon Befehle kennengelernt, die eine bestimmte Schrift einstellen, z.B. `\it` für italic-Schrift. Die geschweiften Klammern geben den Bereich an, in dem ein solcher Befehl wirken soll, ohne sie wirkt er von seinem Auftreten bis zum Ende des Dokuments. *Hier ist ein Stück Text in italic gesetzt.*

### 3.2 Befehle, die auf etwas wirken

Das sind Befehle mit Parametern. Ihnen wird etwas (ein oder mehrere Parameter) übergeben und sie tun etwas damit. Der Befehl `\subsection{}` z.B. bekommt den Text für die überschrift eines Abschnitts übergeben und setzt in in Fettdruck und ordnet ihm eine passende Nummer zu.

Solche Befehle werden auch oft von BenutzerInnen definiert. Zum Beispiel war es mir zu mühsam, jedesmal die notwendigen Befehle einzugeben, um so etwas wie `\subsection` oder `\pageref` im Text stehen zu haben, da wird nämlich eine andere Schrift benutzt und der Backslash (`\`) ist nur im mathematischen

Modus verfügbar, daher müssen  $\$$ -Zeichen verwendet werden, wenn man ihn aufruft. Im Quelltext sieht das so aus:

```
{\tt $\backslash$subsection}
{\tt $\backslash$page}ref}
```

Diese umständliche Schreibweise wird vermieden durch einen benutzereigenen Befehl `\befehl`, der wie folgt definiert wird:

```
\newcommand{\befehl}[1]{\tt $\backslash$#1}
```

Diese Definition steht am Anfang des Dokuments und wo immer im Text dieser Befehl aufgerufen wird, wird der formale Parameter `#1` ersetzt durch den aktuellen Wert, der zwischen geschweiften Klammern übergeben wird. Es können auch Befehle mit mehreren Parametern `#1`, `#2`, ... definiert werden. Die werden dann mit vielen geschweiften Klammern aufgerufen, z.B.

```
\newcommand{\matrix}[9]{
  $\left(
    \begin{array}{ccc}
      #1&#2&#3\\
      #4&#5&#3\\
      #7&#8&#3\\
    \end{array}
  \right)$
}
```

$$\begin{pmatrix} 1 & 3 & 6 \\ -8 & 12 & 6 \\ 0 & 0 & 6 \end{pmatrix}$$

Die Definition eines Befehls muss vor seiner Anwendung erfolgen, jedoch empfiehlt es sich, alle benutzereigenen Befehle am Anfang des Dokuments oder sogar in einer gesonderten Datei zu definieren, um sie leicht wiederfinden zu können.

Im Befehl `\Matrix`, der hier als Beispiel dienen soll, ist in mancherlei Hinsicht im Stoff vorgegriffen worden, ich erkläre daher hier seine Bestandteile:

- Die `$` umschließen ein Textstück, das im mathematischen Modus steht. Dazu später mehr.
- Ein `array` ist eine Tabelle im mathematischen Modus.
- Die Parameterliste (`ccc`) gibt an, dass die Tabelle drei Spalten hat, alle drei zentriert.
- Die Zellen einer Tabelle werden durch `&` voneinander getrennt.
- Die Zeilen einer Tabelle werden mit

beendet.

- `\right(` erzeugt eine öffnende runde Klammer, die ihre Größe dem Inhalt anpasst.
- `\left)` erzeugt eine schließende runde Klammer, die ihre Größe dem Inhalt anpasst.

Beim Aufruf des Befehls wie oben angegeben (der Name ist groß geschrieben, also muss er auch groß geschrieben aufgerufen werden) werden die Zahlen in geschweiften Klammern der Reihe nach den Parametern zugeordnet. In der Definition ist zu sehen, dass nicht alle Parameter verwendet werden, einer dafür gleich dreimal. Das mag für eine Matrix nicht sinnvoll sein, es ist aber erlaubt und ich verwende es z.B. für den Befehl `\silent`, definiert und aufgerufen wie folgt:

```
\newcommand{\silent}[1]{#1}
\silent{Hier steht ein Text, vielleicht seitenlang. Der Befehl
silent gibt den Text und alles, was dazugeh"ort, einfach aus.}
```

Wenn nun die Definition des Befehls so geändert wird:

```
\newcommand{\silent}[1]{}
```

dann sind alle `silent`-Textteile auf einen Schlag aus der Ausgabe verschwunden. Ein nützlicher Trick, um z.B. Kommentare verschiedener Autoren während der Arbeit an einem Dokument ein- und auszublenden.

### 3.3 Befehle, die eine Wirkung haben

Hier haben wir parameterlose Befehle. Einige nützliche Beispiele:

```
\quad          etwas Leerraum
\quad          etwas mehr Leerraum
\%             ein Prozent-Zeichen
\\            eine neue Zeile
\par          ein Absatz
\par\smallskip ein Absatz mit kleinem Leerraum
\par\medskip  ein Absatz mit mittlerem Leerraum
\par\bigskip  ein Absatz mit gro"sem Leerraum
```

## 4 Inhaltsverzeichnis

Weil dieses Dokument schon einige Gliederungspunkte hat, versuchen wir es mal mit einem Inhaltsverzeichnis.  $\LaTeX$  merkt sich in der `toc`-Datei (`table-of-contents`), welche Punkte die Gliederung hat und auf welcher Seite sie stehen. Der Befehl `\tableofcontents` sorgt dafür, dass ein Inhaltsverzeichnis ausgegeben wird. Hier kommt es gleich, an einer recht ungewöhnlichen Stelle im Dokument. Seine Überschrift wird automatisch erzeugt, die Seitenzahlen werden ebenfalls von  $\LaTeX$  ermittelt und jeweils angepasst — ein weiterer Grund, warum mehrfach kompiliert werden muss: Im ersten Durchgang sucht das Programm die Seitenzahlen heraus, im zweiten schreibt es sie ins Inhaltsverzeichnis. Wenn das Schreiben des Inhaltsverzeichnisses die Seitenzahlen noch mal ändert, was meistens der Fall ist, **muss sogar ein drittes Mal kompiliert werden**, bis alle Verweise stimmen. — Besser als dreimal Korrektur-Lesen und sie stimmen noch immer nicht ...

## Inhaltsverzeichnis

1	Einleitung	1
2	Aufzählungen	1
3	Befehle	2
3.1	Befehle, die etwas einstellen . . . . .	2
3.2	Befehle, die auf etwas wirken . . . . .	2
3.3	Befehle mit Wirkung . . . . .	4
4	Inhaltsverzeichnis	4

Auch hier sind die Einflussmöglichkeiten nahezu unbegrenzt, eine sei zur Appetit-Anregung erwähnt: Wenn eine Abschnitts-Überschrift in geschweiften Klammern angegeben wird und davor eine weitere (meist Kurz-Version) in eckigen Klammern, dann erscheint die geschweifte im Text und die eckige im Inhaltsverzeichnis. Zu sehen am Beispiel des Abschnitts 3.3.

## Literatur

[GMS02] GOOSSENS, MICHEL, FRANK MITTELBACH UND ALEXANDER SAMARIN: *Der  $\LaTeX$ -Begleiter*. Pearson Studium, 2002, 560 Seiten, 39.95 EUR (1. Auflage Addison-Wesley, 1994)

[Kop02] KOPKA, HELMUT:  *$\LaTeX$  Band 1: Einführung*. Addison-Wesley, 2002 (3. Auflage), 39.95 EUR.

[KPS99] SCHMIDT, WALTER, JÖRG KNAPPEN, HUBERT PARTL UND IRENE HYNA:  *$\LaTeX_2\epsilon$ -Kurzbeschreibung. 1999, 46 Seiten, online unter <http://parsys.informatik.uni-oldenburg.de/~ewi/softskills/latex/l2kurz.{ps,pdf}>*