3 Eigene Befehle und Umgebungen

Befehle und Umgebungen spielen eine entscheidende Rolle bei der Eingabe eines Textes. Jede Struktur, die im Dokument (wiederholt) auftaucht, sollte durch einen eigenen Befehl oder eine eigene Umgebung erzeugt werden. LaTeX kennt bereits eine große Zahl von Befehlen und Umgebungen. Trotzdem kann es vorkommen, daß in Ihrem Text eine Struktur vorkommt, auf die LaTeX nicht vorbereitet ist. In solchen Situationen kann man sich helfen, indem man eigenen Befehle und Umgebungen definiert. Im allgemeinen stehen solche Definitionen in der Präambel.

Im Vergleich zu der bisher vorgestellten Texteingabe ist es relativ kompliziert, eigene Befehle und Umgebungen zu definieren. Daher tasten wir uns ganz vorsichtig an dieses Thema heran.

Wir werden in den folgenden Abschnitten noch Gelegenheit haben, genauer auf eigene Befehle einzugehen und können weitere Beispiele vorstellen.

3.1 Definieren von Befehlen

Die einfachsten Befehle sind solche, die nur einen Text ausgeben. Ein Beispiel ist der Befehl `\pounds`, der das Pfundzeichen £ erzeugt. Einen *neuen* Befehl kann man mit dem Befehl `\newcommand` definieren, der zwei Argumente hat. Das erste ist der Name, den der neue Befehl haben soll. Das zweite die *Befehlsdefinition,* in der man angibt, was LaTeX tun soll, sobald es in Ihrer Eingabedateidatei auf den neuen Befehl trifft.

```
\newcommand{\vhom}{Vektorraumhomomorphismus}
Der \vhom\
ist injektiv.
```

Der Vektorraumhomomorphismus ist injektiv.

Der Befehl `\vhom` ist somit eine Abkürzung für das lange und fehlerträchtige Wort *Vektorraumhomomorphismus.*

Etwas komplizierter sind Befehle, die zwingende Argumente haben sollen. Solche Befehle werden verwendet, wenn man sie in Situationen benutzen will, die sich nur wenig voneinander unterscheiden. Die Teile, die unterschiedlich sind, werden dann als Argumente angegeben. Die Befehle aus Tabelle 3 zur Erzeugung von Akzenten gehören zu dieser Art (mit einem zwingenden Argument).

Ein Beispiel für einen neuen Befehl mit Argument wäre einer, der die Telefonnummer eines Mitarbeiters des Fachbereiches ausgibt. Alle Rufnummern von Mitarbeitern fangen mit der 4123 an, der übrige Teil ist variabel. Unser neuer Befehl – wir nennen ihn `\tel` – soll daher *ein* zwingendes Argument haben, nämlich die Nummer des Anschlusses.

```
\newcommand{\tel}[1]{4123-#1}
Herr Nitschke hat die
Telefonnummer~\tel{4929},
Herr Ripken~\tel{5108}.
```

Herr Nitschke hat die Telefonnummer 4123-4929, Herr Ripken 4123-5108.

Hier sieht die Verwendung des `\newcommand`-Befehls etwas anders aus als zuvor. Und zwar tritt ein optionales Argument (das in eckigen Klammern) auf. Dieses gibt an, wieviele Argumente der neue Befehl `\tel` haben soll. Bis zu neun Argumente sind möglich.

In der Befehlsdefinition taucht das Zeichen `#` auf. Es wurde bereits erwähnt, daß es eine besondere Bedeutung für LaTeX hat. Es kennzeichnet einen *Parameter,* der von LaTeX bei Verwendung des Befehls durch das entsprechende Argument des Befehls ersetzt wird: `#1` wird durch das erste Argument ersetzt, `#2` durch das zweite und so weiter.

Die Eingabezeile

```
\newcommand{\tel}[1]{4123-#1}
```

besagt also:

> Definiere einen neuen Befehl `\tel`, der ein Argument hat. Wenn dieser Befehl benutzt wird, so gebe `4123-` aus und anschließend das, was als erstes Argument angegeben wurde.

Zusätzlich zu `\newcommand` gibt es noch den Befehl `\renewcommand`, den man fast genauso verwendet wie `\newcommand`. Der Unterschied besteht darin, daß man mit `\renewcommand` einen bereits vorhandenen Befehl verändern kann.

3.2 Definieren von Umgebungen

Bisher kennen wir nur *eine* Umgebung: `document`. Daher können wir hier nur ein langweiliges Beispiel geben.

Mit dem Befehl

`\newenvironment{`⟨*Name*⟩`}{`⟨*Anfang*⟩`}{`⟨*Ende*⟩`}`

können Sie eine neue Umgebung ⟨*Name*⟩ definieren. Sie haben dadurch die Möglichkeit, in Ihrem Text

`\begin{`⟨*Name*⟩`}` ... `\end{`⟨*Name*⟩`}`

einzugeben. Das zweite Argument gibt an, was LaTeX tun soll, sobald es in Ihrer Eingabedatei auf `\begin{`⟨*Name*⟩`}` trifft. Das dritte Argument sagt LaTeX, was bei `\end{`⟨*Name*⟩`}` zu tun ist.

Man kann auch Umgebungen definieren, die Argumente haben. Dazu verwendet man

`\newenvironment{`⟨*Name*⟩`}[`⟨*Arg*⟩`]{`⟨*Anfang*⟩`}{`⟨*Ende*⟩`}`.

Das optionale Argument ⟨*Arg*⟩ gibt wie beim Befehl `\newcommand` an, wieviele Argumente die neue Umgebung haben soll. Auch hier sind maximal neun Argumente möglich, auf die man durch die Parameter `#1`, ... , `#9` zugreifen kann.

Die Parameter `#1`, ... , `#9` können *nur* in ⟨*Anfang*⟩ verwendet werden, *nicht* in ⟨*Ende*⟩!

Als Beispiel soll eine neue Umgebung `erkl` definiert werden, die man benutzt, wenn man ein Wort erklärt. Das Wort soll in Gänsefüßchen eingeschlossen werden und ein Doppelpunkt soll sich anschließen.

Die neue Umgebung `erkl` muß somit *ein* Argument haben – nämlich das zu erklärende Wort. Die eigentliche Erklärung ist der Text *zwischen* `\begin{erkl}` und `\end{erkl}`.

Im Argument ⟨*Anfang*⟩ des `\newenvironment`-Befehls legen wir fest, daß das Argument der neuen Umgebung `erkl` zwischen Gänsefüßchen gesetzt wird. Außerdem fügen wir einen Doppelpunkt ein. Am Ende der Umgebung ist nichts zu tun.

```
\newenvironment{erkl}[1]{"`#1"':}{}
\begin{erkl}{Gnu}
  S"ud- und ostafrikanische Antilope.
\end{erkl}
```

„Gnu“: Süd- und ostafrikanische Antilope.

Bestehende Umgebungen lassen sich mit dem Befehl `\renewenvironment` verändern.

4 Strukturen

4.1 Titelinformationen

Am Anfang eines Textes stehen im allgemeinen sein Titel, die Namen der Verfasser(innen) und das Erstellungs- oder Erscheinungsdatum.

LaTeX stellt Ihnen zur Angabe dieser Informationen die Befehle

```
\title{⟨Titel⟩}
\author{⟨Verfasser⟩}
\date{⟨Datum⟩}
```

zur Verfügung. Mehrere Autoren werden im Argument des `\author`-Befehls durch `\and` voneinander getrennt.

In der Eingabedatei zu diesem Skript haben wir beispielsweise folgendes geschrieben:

```
\title{Kompaktkurs zu \LaTeXe}
\author{Dirk Nitschke \and Wolfgang Ripken}
```

Wir haben also den `\date`-Befehl weggelassen. In diesem Fall tut LaTeX so, als hätten wir den `\date`-Befehl mit dem Datum des letzten LaTeX-Durchlaufs als Argument aufgerufen.

Keiner der Befehle `\title`, `\author` und `\date` erzeugt irgendwelchen Text. Ausgegeben werden die Titelinformationen erst durch den Befehl `\maketitle`, der nicht vor `\begin{document}` aufgerufen werden darf.

Auf den ersten Blick erscheint dieses Vorgehen vielleicht etwas kompliziert: Wieso soll man bei der Angabe mehrerer Autoren den `\and`-Befehl benutzen und nicht einfach „und" oder „and" oder „," schreiben? Wieso erzeugen die Befehle `\title`, `\author` und `\date` keinen Text? Der Grund besteht im *logischen* Design: Weil die Befehle `\title`, `\author` und `\date` keinen Text erzeugen, können wir sie in beliebiger Reihenfolge aufrufen, ohne daß dies Einfluß auf das Aussehen der Titelseite hätte. Und ob der Befehl `\and` auf der Titelseite das Wörtchen „und" oder „and", ein Komma oder einfach nur einen etwas größeren Abstand erzeugen soll, braucht uns nicht zu kümmern. Dies ist in der jeweiligen Dokumentenklasse festgelegt. (Eigentlich doch ganz praktisch, oder?)

4.2 Der Abstract

Eine weitere meist zu Beginn des Textes auftretende Struktur ist der *Abstract.* Dieser läßt sich mit der Umgebung `abstract` verwirklichen. Der Abstract erhält automatisch eine Überschrift, die durch `\abstractname` erzeugt wird. Das `german`-Paket definiert `\abstractname` als „Zusammenfassung".

```
\begin{abstract}
 In diesem Kurs soll eine
 Einf"uhrung in \LaTeXe\
 gegeben werden.
\end{abstract}
```

Zusammenfassung

In diesem Kurs soll eine Einführung in LaTeX 2_ε gegeben werden.

4.3 Gliederungseinheiten

In einem Text werden Sätze zu Absätzen zusammengefaßt und diese wiederum in eine Abschnittsstruktur eingefügt. Wir kennen zum Beispiel *Teile, Kapitel, Abschnitte* und *Paragraphen.* Wir nennen all diese Einheiten *Gliederungseinheiten.* In diesem Unterabschnitt (sic!) des Skriptes soll beschrieben werden, wie man LaTeX diese Strukturen mitteilt.

Gliederungseinheiten sind meistens mit einer Überschrift versehen, häufig zusätzlich mit einer Nummer. In LaTeX beginnt man jede Gliederungseinheit mit einem Befehl, der als Argument die zugehörige Überschrift hat. Die Nummer wird automatisch erzeugt.

Die Überschrift dieses Unterabschnittes und der erste Satz wurden so erzeugt:

```
\subsection{Gliederungseinheiten}

In einem Text werden S"atze zu Abs"atzen
zusammengefa"st und diese wiederum in eine
Abschnittsstruktur eingef"ugt.
```

Welche Gliederungsbefehle zur Verfügung stehen und wie die Numerierung der Gliederungseinheiten vorzunehmen ist, wird durch die verwendete Dokumentenklasse bestimmt. In den Standardklassen gibt es die folgenden Befehle:

`\chapter`	(nicht in der Klasse `article`)
`\section`	
`\subsection`	
`\subsubsection`	
`\paragraph`	
`\subparagraph`	

Zusätzlich gibt es den Befehl `\part`, mit dem Sie Ihr Dokument in *Teile* unterteilen können.

Nicht immer ist die Numerierung einer Gliederungseinheit gewünscht. Die oben genannten Gliederungsbefehle können auch in der Sternform verwendet werden. Sie erzeugen dann *keine* Nummer.

```
\subsection*{Noch einer}

Diesmal erzeugt \LaTeX\ keine
Nummer.
```

Noch einer

Diesmal erzeugt LaTeX keine Nummer.

Der Sinn der Numerierung von Gliederungseinheiten besteht im allgemeinen darin, an anderer Stelle des Dokumentes auf eine bestimmte Einheit verweisen zu können. Problematisch ist dabei, daß sich im Laufe des Schreibens die Numerierungen häufig ändern und man dann alle Verweise anpassen muß. In Abschnitt 5 (hier haben wir so einen Verweis) erfahren Sie, wie man LaTeX diese Arbeit aufbürden kann.

4.4 Das Inhaltsverzeichnis

Beim Verfassen eines Textes gehört die Erstellung eines Inhaltsverzeichnisses normalerweise zu den aufwendigeren Aufgaben. LaTeX nimmt Ihnen diese Arbeit ab. Wenn Sie ein Inhaltsverzeichnis erzeugen wollen, brauchen Sie lediglich den Befehl `\tableofcontents` an die entsprechende Stelle der Eingabedatei zu schreiben.

In das Inhaltsverzeichnis werden automatisch alle mit den Gliederungsbefehlen aus Unterabschnitt 4.3 erzeugten Überschriften aufgenommen – allerdings nicht diejenigen, die mit den jeweiligen Sternformen angegeben worden sind. Soll eine derartige Überschrift dennoch im Inhaltsverzeichnis erscheinen, müssen Sie dies explizit verlangen, indem Sie direkt *hinter* dem zugehörigen Gliederungsbefehl

`\addcontentsline{toc}{`⟨*Gliederungseinheit*⟩`}{`⟨*Überschrift*⟩`}`

eingeben. Dabei setzen Sie für ⟨*Gliederungseinheit*⟩ die Art der Gliederungseinheit (also beispielsweise `chapter` oder `subsection`) ein.

Das Inhaltsverzeichnis erhält automatisch eine Überschrift, die durch den Befehl `\contentsname` erzeugt wird. Im `german`-Paket wird `\contentsname` als „Inhaltsverzeichnis" festgelegt.

Um ein korrektes Inhaltsverzeichnis zu erhalten, müssen Sie LaTeX mindestens dreimal durchlaufen lassen!

Das liegt an der Art, wie LaTeX das Inhaltsverzeichnis erzeugt: Wenn LaTeX in der Eingabedatei auf einen Gliederungsbefehl stößt, schreibt es einen Eintrag in eine spezielle Datei (die `toc`-Datei), der im allgemeinen aus der Numerierung, der Überschrift und der aktuellen Seitenzahl besteht. Beim nächsten Durchlauf wird diese Datei durch den Befehl `\tableofcontents` eingelesen und das Inhaltsverzeichnis ausgegeben. Da die `toc`-Datei stets nur die Informationen des vorherigen LaTeX-Laufs enthält, das Inhaltsverzeichnis aber unter Umständen die Anzahl der Seiten vergrößert und damit den nachfolgenden Text verschiebt, müssen Sie LaTeX nun noch ein weiteres Mal durchlaufen lassen.

5 Querverweise

Gliederungseinheiten, Abbildungen, Tabellen und auch mathematische Formeln werden häufig numeriert, damit man im Text auf sie verweisen kann. Dies ist nicht sonderlich kompliziert, wenn man seinen Text nicht mehr verändert und sich die Numerierung nicht mehr ändert. In der Praxis kommt es aber immer wieder vor, daß ein Abschnitt, eine Tabelle oder Formel eingefügt wird. Hierdurch verändert sich die Numerierung nachfolgender Objekte und die Verweise müssen angepaßt werden.

LaTeX bietet die Möglichkeit, die Numerierungen und Verweise automatisch richtig vorzunehmen. Das funktioniert so: Man ordnet einem Objekt, auf das man verweisen möchte, ein sogenanntes *Schlüsselwort* zu. Über dieses Schlüsselwort kann man sich dann im Text auf das Objekt beziehen. LaTeX wandelt diese Referenz in die entsprechende Nummer um.

Die Vergabe der Schlüsselworte wird mit dem Befehl `\label` vorgenommen, die Ausgabe der Nummer durch den Befehl `\ref`. Zusätzlich gibt es noch den Befehl `\pageref`, mit dem Sie auf die Seite verweisen können, auf der sich das Objekt befindet. Ein Schlüsselwort darf aus Groß- und Kleinbuchstaben, Ziffern und Satzzeichen bestehen. Da Groß- und Kleinschreibung von LaTeX unterschieden wird, sind die Schlüsselworte `sec:inhalt` und `sec:Inhalt` verschieden.

Im normalen Text wird dem Schlüsselwort durch den Befehl `\label` die Nummer der aktuellen Gliederungseinheit zugewiesen. In einer numerierten Umgebung die Nummer dieser Umgebung.

```
\section{Querverweise}
\label{sec:refer}
...
Wir befinden uns in
Abschnitt~\ref{sec:refer}, der
auf Seite~\pageref{sec:refer}
begonnen hat.
```

Wir befinden uns in Abschnitt 5, der auf Seite 21 begonnen hat.

6 Literaturangaben

Bei einer Literaturangabe handelt es sich um einen Querverweis auf eine andere Veröffentlichung, eine *Quelle*. Im allgemeinen werden alle in einem Text verwendeten Quellen am Ende des Textes in einem Literaturverzeichnis gesammelt und auf bestimmte Weise numeriert. Zusätzlich versieht man die Quellen mit einem Schlüssel, durch den man auf ihre jeweiligen Nummern zugreifen kann.

6.1 Literaturverzeichnis

Um ein Literaturverzeichnis zu erstellen, wird die `thebibliography`-Umgebung verwendet. Wie diese Umgebung benutzt wird, läßt sich am einfachsten anhand eines Beispiels erläutern:

```
\begin{thebibliography}{2}
\bibitem{higham:1993}
  Nicholas~J. Higham.
  "`Handbook of Writing for the
  Mathematical Sciences"'.
  SIAM, Philadelphia, 1993.

\bibitem{lamport:1995}
  Leslie Lamport.
  "`Das \LaTeX-Handbuch"'.
  Addison-Wesley, Bonn, 1995.
\end{thebibliography}
```

Literatur

[1] Nicholas J. Higham. „Handbook of Writing for the Mathematical Sciences“. SIAM, Philadelphia, 1993.

[2] Leslie Lamport. „Das LaTeX-Handbuch“. Addison-Wesley, Bonn, 1995.

Das Argument der `thebibliography`-Umgebung bestimmt, wieviel Platz für die Nummern der einzelnen Literaturangaben gelassen wird. Innerhalb der Umgebung beginnt jeder Eintrag mit einem `\bibitem`-Befehl, der als Argument den Schlüssel der Literaturangabe hat. Die Schlüsselworte dürfen hierbei aus Groß- und Kleinbuchstaben, Ziffern und den Satzzeichen mit Ausnahme des Kommas `,` bestehen.

Das Literaturverzeichnis erhält automatisch eine Überschrift, die in der Klasse `article` durch den Befehl `\refname`, in den Klassen `report` und `book` durch `\bibname` erzeugt wird. Das `german`-Paket definiert `\refname` und `\bibname` als „Literatur“ bzw. „Literaturverzeichnis“.

Das Literaturverzeichnis wird *nicht* automatisch in das Inhaltsverzeichnis übernommen. (Erinnern Sie sich noch an den Befehl `\addcontentsline`?)

Normalerweise werden Literaturangaben mit Zahlen numeriert. Wenn Sie eine andere Markierung (beispielsweise Anfangsbuchstaben der Autoren zusammen mit Erscheinungsjahr) wünschen, können Sie diese als optionales Argument des `\bibitem`-Befehls angeben. Denken Sie auch daran, das Argument der `thebibliography`-Umgebung anzupassen, damit LaTeX ausreichend Platz für die längste Markierung läßt.

```
\begin{thebibliography}{Lam95}
\bibitem[Hig93]{higham:1993}
  Nicholas~J. Higham.
  "‘Handbook of Writing for the
  Mathematical Sciences"’.
  SIAM, Philadelphia, 1993.

\bibitem[Lam95]{lamport:1995}
  Leslie Lamport.
  "‘Das \LaTeX-Handbuch"’.
  Addison-Wesley, Bonn, 1995.
\end{thebibliography}
```

Literatur

[Hig93] Nicholas J. Higham. „Handbook of Writing for the Mathematical Sciences“. SIAM, Philadelphia, 1993.

[Lam95] Leslie Lamport. „Das LaTeX-Handbuch“. Addison-Wesley, Bonn, 1995.

6.2 Referenzen auf Literaturangaben

Für eine Referenz auf eine Literaturangabe gibt es den Befehl `\cite`. Wie der Befehl `\ref` hat `\cite` ein Argument, nämlich den Schlüssel der gewünschten Literaturangabe. `\cite` kann mehr als `\ref`: Es ist möglich, mehrere Quellen gleichzeitig zu zitieren, indem man mehrere Schlüssel durch Kommata voneinander trennt. (Dies ist der Grund, warum das Komma nicht im Schlüsselwort einer Quelle vorkommen darf.) Außerdem hat `\cite` ein optionales Argument, durch das der Querverweis um weitere Informationen wie zum Beispiel Seitenangaben ergänzt werden kann.

```
Die B"ucher~\cite{higham:1993,lamport:1995}
m"ochten wir Ihnen besonders ans Herz legen.
In \cite[S.~81ff.]{lamport:1995}
wird ausf"uhrlich auf die Erstellung des
Literaturverzeichnisses eingegangen.
```

Die Bücher [5, 9] möchten wir Ihnen besonders ans Herz legen. In [9, S. 81ff.] wird ausführlich auf die Erstellung des Literaturverzeichnisses eingegangen.

7 Listen

Listen treten meist in einer der folgenden drei Formen auf:

- Aufzählungen;
- Auflistungen, in denen die Reihenfolge der Einträge nicht wichtig ist (die Einträge sind dann nicht numeriert);
- Beschreibungen.

Für diese Arten von Listen bietet LaTeX die Umgebungen `enumerate`, `itemize` und `description` an. Jeder neue Punkt einer solchen Liste wird durch den Befehl `\item` eingeleitet und automatisch mit einer Standardmarkierung versehen. Man kann solche Listen auch ineinander schachteln, wobei mindestens vier Ebenen zulässig sind. Möchte man für einen Listenpunkt eine andere Markierung verwenden, so gibt man den gewünschten Text als optionales Argument des entsprechenden `\item`-Befehls an.

```
\begin{itemize}
\item Eine Auflistung.
  \begin{enumerate}
  \item Der erste Punkt einer
    Aufz"ahlung.
  \item Eine Liste sollte mindestens
    zwei Punkte enthalten.
  \end{enumerate}
\item[*] Ein Punkt mit anderer
  Markierung.
\end{itemize}
```

- Eine Auflistung.
    1. Der erste Punkt einer Aufzählung.
    2. Eine Liste sollte mindestens zwei Punkte enthalten.

* Ein Punkt mit anderer Markierung.

Die Standardmarkierung der einzelnen Listenpunkte wird – je nach Schachtelungstiefe – bei der `enumerate`-Umgebung durch die Befehle `\labelenumi`, `\labelenumii`, `\labelenumiii` und `\labelenumiv` erzeugt. In der `itemize`-Umgebung geschieht dies durch die Befehle `\labelitemi`, `\labelitemii`, `\labelitemiii` und `\labelitemiv`.

Es lassen sich nur Listenpunkte einer `enumerate`-Umgebung, die mit der Standardmarkierung versehen sind, zufriedenstellend referenzieren.

```
\begin{enumerate}
\item\label{en:def1} Der erste Punkt.
\item[*]\label{en:stern} Ein Stern.
\item\label{en:def2} Der letzte Punkt.
\end{enumerate}
Die Punkte~\ref{en:def1},
\ref{en:stern} (das klappt
leider nicht) und~\ref{en:def2}.
```

1. Der erste Punkt.

* Ein Stern.

2. Der letzte Punkt.

Die Punkte 1, 1 (das klappt leider nicht) und 2.

Die `description`-Umgebung ist für Beschreibungen gedacht. Als Markierung eines Listenpunktes verwendet man meist das Wort, das in diesem Punkt beschrieben wird. Da es daher keine sinnvolle Standardmarkierung geben kann, ist sie leer und man gibt den zu beschreibenden Begriff als optionales Argument des `\item`-Befehls an.

```
\begin{description}
\item[enumerate] Aufz"ahlungen
\item[itemize] Auflistungen
\item[description] Beschreibungen
\item Eine leere Markierung
\end{description}
```

enumerate Aufzählungen

itemize Auflistungen

description Beschreibungen

Eine leere Markierung

8 Theoreme und ähnliches

In mathematischen Texten treten häufig die Strukturen *Satz, Lemma* oder *Beweis* auf. Aber auch in nicht-mathematischen Texten kann es etwas in dieser Art geben: *Regeln, Vermutungen* oder *Schlußfolgerungen.* Diese Strukturen haben etwas gemein: Es handelt sich um *Umgebungen,* denen ein Wort vorangestellt wird, welches angibt, um welche Struktur es sich handelt. Häufig werden diese Strukturen auch numeriert.

Da es unmöglich ist, für alle denkbaren Möglichkeiten eine Umgebung bereitzustellen, gibt es im `amsthm`-Paket, den Befehl `\newtheorem`. Dieser hat zwei Argumente: den Namen der Umgebung und den Text, der als *Kopf* erscheinen soll.

```
\newtheorem{satz}{Satz}
\begin{satz}
  Die Menge der rationalen
  Zahlen ist abz"ahlbar.
\end{satz}
```

Satz 1. *Die Menge der rationalen Zahlen ist abzählbar.*

LaTeX nimmt automatisch eine Numerierung vor und gibt hinter der Nummer einen Punkt aus. Verwendet man die Sternform `\newtheorem*`, so wird die Umgebung *nicht* numeriert.

```
\newtheorem*{cor}{Korollar}
\begin{cor}
  Die Menge der ganzen
  Zahlen ist abz"ahlbar.
\end{cor}
```

Korollar. *Die Menge der ganzen Zahlen ist abzählbar.*

Häufig werden unterschiedliche Arten von solchen theoremähnlichen Strukturen unterschiedlich gesetzt. Bei Theoremen und Sätzen ist es üblich, das Wort *Theorem* bzw. *Satz* fett zu setzen und die Formulierung kursiv. *Definitionen* hingegegen werden aufrecht gesetzt und das Wort *Definition* fett. (Das zu definierende Wort wird im allgemeinen hervorgehoben. Hierzu dient der Befehl `\emph{`⟨*Wort*⟩`}`.) Um deutlich zu machen, zu welcher Sorte von theoremähnlichen Strukturen eine neue Umgebung gehört, gibt es den Befehl `\theoremstyle`. Dieser hat ein Argument, in dem man den entsprechenden Stil angibt. Mögliche Stile sind `plain`, `definition` und `remark`. Wird `\theoremstyle` nicht angegeben, so wird automatisch `plain` gewählt.

```
\theoremstyle{definition}
\newtheorem{defi}{Definition}
\theoremstyle{plain}
\newtheorem{satz}{Satz}
\theoremstyle{remark}
\newtheorem{rem}{Bemerkung}
\begin{satz}
  Das Einselement ist
  eindeutig bestimmt.
\end{satz}
\begin{defi}
  Eine \emph{Gruppe} ist ein
  Monoid, in dem jedes
  Element invertierbar ist.
\end{defi}
\begin{rem}
  Das Einselement ist zu
  sich selbst invers.
\end{rem}
```

Satz 2. *Das Einselement ist eindeutig bestimmt.*

Definition 1. Eine *Gruppe* ist ein Monoid, in dem jedes Element invertierbar ist.

Bemerkung 1. Das Einselement ist zu sich selbst invers.

Im `amsthm`-Paket ist bereits eine Umgebung `proof` definiert, die für *Beweise* verwendet werden kann. Als Kopf wird der durch den Befehl `\proofname` definierte Text ausgegeben. Standardmäßig ist dies „Proof". Für deutsche Texte müssen Sie diesen Befehl mithilfe von `\renewcommand` ändern. Außerdem wird am Ende einer `proof`-Umgebung ein *Beweisendezeichen* `\qedsymbol` gesetzt. Standardmäßig ist dies „□".

```
\renewcommand{\proofname}{Beweis}
\begin{proof}
  Der Beweis wird als
  "Ubungsaufgabe gestellt.
\end{proof}
```

Beweis. Der Beweis wird als Übungsaufgabe gestellt. □

Wenn der Beweis zum Beispiel mit einer Liste endet, so wird das Beweisendezeichen an der falschen Stelle gesetzt. In diesem Fall muß man das Zeichen mit dem Befehl `\qedsymbol` oder `\qed` selber an der geeigneten Stelle setzen und anschließend den Befehl `\qed` zum Nichtstun verurteilen.

```
\begin{proof}
  Offensichtlich gilt:
  \begin{itemize}
  \item Eins ist klar!
  \item Zwei auch!\qed
  \end{itemize}
  \renewcommand{\qed}{}
\end{proof}
```

Beweis. Offensichtlich gilt:

- Eins ist klar!
- Zwei auch! □

9 Fußnoten

Im laufenden Text erzeugt man eine Fußnote durch den Befehl `\footnote`. Als Argument gibt man den gewünschten Text der Fußnote an.

```
Man kann auch Verweise auf
Fu"snoten\footnote{\label{fn:diese}%
Wie diese.} verwenden. Das war gerade
Fu"snote~\ref{fn:diese}.
```

Man kann auch Verweise auf Fußnoten[1] verwenden. Das war gerade Fußnote 1.

[1] Wie diese.

Der Befehl `\footnote` funktioniert nicht überall!