



Grundkurs LaTeX

Release 0.1.0c

Aktualisiert am 02.12.2018

Bernhard Grotz

<http://www.grund-wissen.de>

Dieses Buch wird unter der [Creative Commons License \(Version 3.0, by-nc-sa\)](#) veröffentlicht. Alle Inhalte dürfen daher in jedem beliebigen Format vervielfältigt und/oder weiterverarbeitet werden, sofern die Weitergabe nicht kommerziell ist, unter einer gleichen Lizenz erfolgt, und das Original als Quelle genannt wird. Siehe auch:

[Erläuterung der Einschränkung by-nc-sa](#)
[Leitfaden zu Creative-Commons-Lizenzen](#)

Unabhängig von dieser Lizenz ist die Nutzung dieses Buchs für Unterricht und Forschung (§52a UrhG) sowie zum privaten Gebrauch (§53 UrhG) ausdrücklich erlaubt.

Der Autor erhebt mit dem Buch weder den Anspruch auf Vollständigkeit noch auf Fehlerfreiheit; insbesondere kann für inhaltliche Fehler keine Haftung übernommen werden.

Die Quelldateien dieses Buchs wurden unter [Linux](#) mittels [Vim](#) und [Sphinx](#), die enthaltenen Graphiken mittels [Inkscape](#) erstellt. Der Quellcode sowie die Original-Graphiken können über die Projektseite heruntergeladen werden:

<http://www.grund-wissen.de>

Bei Fragen, Anmerkungen und Verbesserungsvorschlägen bittet der Autor um eine kurze Email an folgende Adresse:

info@grund-wissen.de

Augsburg, den 2. Dezember 2018.

Bernhard Grotz

Inhaltsverzeichnis

Installation von LaTeX	1
Einführung	2
Ein Latex-Minimalbeispiel	2
Aufbau eines LaTeX-Dokuments	2
Erstellen eines PDF-Dokuments aus einer Quelldatei	4
Grundlegende Konzepte in LaTeX	4
Gewöhnliche Textsatzung	7
Abschnitte und Überschriften	7
Schriftfamilien und Texthervorhebungen	10
Schriftgrößen und Längenmaße	12
Spezielle Zeichen	13
Silbentrennung	14
Vertikale und horizontale Abstände	15
Fußnoten und Randnotizen	16
Querverweise	16
Boxen, Balken und Minipages	17
Mathematischer Formelsatz	21
Mathematische Symbole	23
Mathematische Ausdrücke	25
Wichtige Umgebungen	32
center, flushleft, flushright – Textausrichtung	32
itemize, enumerate und description – Aufzählungen	32
figure – Abbildungen	34
tabbing – Ausgerichteter Text	36
tabular und table – Tabellen	36
verbatim – Quelltext	38
quote, quotation und verse – Einrückungen	39
Zusatzpakete	40
array – Bessere Spaltenoptionen für Tabellen	40
blindtext – Blindtext setzen	40
caption – Bildunterschriften ohne Nummerierung	41
cmap – Copy und Paste in PDF-Dateien	41
colortbl – Tabellen mit farbigem Hintergrund	42
geometry – Seitenränder festlegen	42
hyperref – Verlinkungen erstellen	43
longtable – Mehrseitige Tabellen	44
lscap – Bereiche in Querformat setzen	46
multicol – Tabelleneinträge über mehrere Spalten	46
multirow – Tabelleneinträge über mehrere Zeilen	47
paralist – Kompakte Aufzählungen	47
units – Einheiten setzen	47
urls – Links auf Webseiten	48

Seitenlayout	49
Kopf- und Fußzeilen	49
Seitenränder	52
Zähler und Makros	56
Zähler-Variablen	56
Makros definieren	57
Weitere Dokumentklassen	59
scr1ttr2 – Briefe und Serienbriefe	59
Links	64
Stichwortverzeichnis	66

Installation von LaTeX

LaTeX ist definitiv der Standard als wissenschaftliches Textsatzungssystem. Es wurde bereits ab Beginn der 1980er Jahre von [Donald Knuth](#) entwickelt und wurde bzw. wird seitdem beständig weiterentwickelt und ergänzt. Lag die Stärke von LaTeX ursprünglich in der Darstellung von mathematischen Formeln und Quellcode, existieren inzwischen ausgereifte Pakete für Briefe, Präsentationen, Notensatz uvm.

Dokumente werden in LaTeX nicht mittels einer graphischen Bedienoberfläche gesetzt, sondern gewissermaßen „programmiert“. Dies ermöglicht nicht nur ein Aufteilen von umfangreichen Werken in kleinere Bestandteile und eine Wiederverwertbarkeit von Code-Stücken, es macht obendrein das Erstellen von PDF-Dateien „skriptbar“. Einige Wiki-Werkzeuge, beispielsweise [Sphinx](#), können so aus einer eigenen (meist kürzeren) Syntax heraus LaTeX- bzw. PDF-Dokumente erzeugen.

Für wissenschaftliche Publikationen, insbesondere für mathematische und naturwissenschaftliche Formeln, ist ein umfangreiches LaTeX-System empfehlenswert. Sofern genügend Festplattenspeicher vorhanden ist, sollten folgende Pakete installiert werden:

```
sudo aptitude install texlive-base texlive-common texlive-latex-base \  
texlive-latex-recommended texlive-latex-extra texlive-math-extra \  
texlive-science texlive-extra-utils texlive-lang-german
```

Zur Installation der obigen Pakete wird insgesamt etwa 1 GB an Festplattenspeicher benötigt.

Möchte man zusätzlich auf ein umfangreiches Schriftpaket zurückgreifen (siehe *LaTeX Font Catalogue* <<http://www.tug.dk/FontCatalogue/seriffonts.html>>), so kann man zusätzlich das Paket `texlive-fonts-extra` installieren; dieses hat allerdings alleine einen Umfang von rund 800 MB.

CTAN-Zusatzpakete installieren

LaTeX in seiner Grundform kann durch zahlreiche Pakete erweitert werden. Eine ausführliche Übersicht (inklusive der Paket-Dokumentationen) findet sich im so genannten [CTAN](#) („Comprehensive TeX Archive Network“).

Um ein CTAN-Paket zu installieren, lädt man die jeweilige `.zip`-Datei herunter, entpackt sie und kopiert sie mit Superuser-Rechten in den LaTeX-Pfad, unter Linux meist `/usr/share/texmf/tex/latex`. Anschließend muss die Veränderung der LaTeX-Datenbank noch mitgeteilt werden. Hierzu gibt man in einem Shell-Fenster folgendes ein:

```
sudo texhash && sudo mktexlsr
```

Anschließend kann das Paket in der Dokument-Präambel (meist) als `\usepackage{paketname}` eingebunden werden. Nähere Beschreibungen finden sich in den jeweiligen Paket-Dokumentationen.

Einführung

LaTeX-Quellcode wird üblicherweise in Dateien mit der Endung `tex` geschrieben. Mit dem Programm `pdflatex` kann daraus ein PDF-Dokument erstellt werden.

Ein Latex-Minimalbeispiel

Eine minimale LaTeX-Quelldatei sieht beispielsweise folgendermaßen aus:

```
% Datei: hallo-welt.tex

% Papierformat, Schriftgröße und Dokumentklasse festlegen:
\documentclass[a4paper, 12pt]{article}

% Beginn des Dokuments:
\begin{document}

Hallo Welt!

% Ende des Dokuments:
\end{document}
```

Bei allen Zeilen, die mit einem Prozentzeichen (%) beginnen, handelt es sich um Kommentare, die von LaTeX beim Übersetzen des Quellcodes in eine PDF-Datei ignoriert werden.

Wird der obige Quelltext als Datei `hallo-welt.tex` in einem beliebigen Verzeichnis gespeichert, so kann in einer Shell durch einen Aufruf von `pdflatex hallo-welt.tex` in diesem Verzeichnis das entsprechende PDF-Dokument `hallo-welt.pdf` erstellt werden.

Aufbau eines LaTeX-Dokuments

In der ersten Zeile einer `.tex`-Datei wird mit der Anweisung `\documentclass` die Dokumentklasse festgelegt. Neben den auf amerikanische Layout-Formate zugeschnittenen Standard-Klassen `article`, `report` und `book` können auch die europäischen Formate `scrartcl` für kurze Artikel, `scrcrpt` für (längere) Berichte und `scrbook` für Bücher gewählt werden. Zudem gibt es beispielsweise die Dokumentklassen `scrlettr` für Briefe und `beamer` für Präsentationen.

Allgemein hat die `\documentclass`-Anweisung folgende Syntax:

```
\documentclass[optionen]{dokumentklasse}
```

Optionale Argumente werden allgemein in eckige Klammern gesetzt und können weggelassen werden, notwendige Argumente werden in geschweifte Klammern gesetzt.

Mögliche Optionen für Dokumentklassen sind beispielsweise:

Option	Beschreibung
10pt oder 12pt	Festlegung der Standard-Schriftgröße
oneside oder twoside	Erstellen einseitiger oder zweiseitiger Dokumente
twocolumn	Zweispaltige Textausgabe
noparskip oder halfparskip	Festlegung des Abstands zwischen Absätzen

In den darauf folgenden Zeilen, der so genannten „Präambel“ einer LaTeX-Datei, können mit der Anweisung `\usepackage` Zusatzpakete geladen werden. Durch eine passende Auswahl an solchen Paketen kann LaTeX bei bestmöglicher Performance auf die jeweilige Aufgabe angepasst werden.

Die `\usepackage`-Anweisung hat allgemein folgende Syntax:

```
\usepackage[optionen]{paketname}
```

Häufig genutzte Pakete sind beispielsweise `graphicx` zum Einbinden von Bildern oder `color` zum Drucken von farbigem Text. Viele Pakete werden ohne eine Angabe von Optionen geladen. Solche Pakete können auch folgendermaßen importiert werden:

```
\usepackage{paket1, paket2, ...}
```

Anschließend beginnt das eigentliche Dokument. Es wird von `\begin{document}` und `\end{document}` eingeschlossen.

Eine *Gliederung eines Dokuments* in Kapitel (möglich in den Dokumentklassen `scrbook` und `scrreprt`) und Abschnitte (möglich in allen Dokumentklassen) ist durch die Anweisungen `\chapter{}` beziehungsweise `\section{}` an den entsprechenden Stellen möglich. Wahlweise kann anhand dieser Untergliederungen auch ein Inhaltsverzeichnis durch die Anweisung `\tableofcontents` erstellt und an der entsprechenden Stelle in das Dokument eingefügt werden.

Ein etwas ausführlicheres Beispiel eines LaTeX-Dokuments sieht somit folgendermaßen aus:

```
\documentclass[a4paper, 12pt]{scrbook}

\begin{document}

% Inhaltsverzeichnis anzeigen:
\tableofcontents

% Neue Seite beginnen:
\newpage

% Überschrift eines Kapitels:
\chapter{Einleitung}

...

\newpage

% Überschrift eines Kapitels:
\chapter{Theoretische Grundlagen}

% Überschrift eines Abschnittes:
\section{Allgemeines}

...

% Überschrift eines Abschnittes:
\section{Konkreteres}

...

\end{document}
```

Die Anweisung `\newpage` bewirkt hierbei, dass an der entsprechenden Stelle eine neue Seite angefangen wird.

Umlaute und deutsche Sprachunterstützung

LaTeX ist ursprünglich für den englischen Sprachraum entwickelt worden. Es gibt allerdings Zusatzpakete, die eine Erkennung von beispielsweise deutschen Umlauten sowie passender Silbentrennung ermöglichen.

In der Präambel eines deutschsprachigen LaTeX-Dokuments sollten folgende Pakete geladen werden:

```
% deutsche Silbentrennung aktivieren:  
\usepackage[ngerman]{babel}  
  
% deutsche Umlaute erlauben:  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}
```

Sollte die deutsche Version des Sprachpakets `babel` nicht gefunden werden, so muss das Paket `texlive-lang-german` nachinstalliert werden (`sudo aptitude install texlive-lang-german`).

Erstellen eines PDF-Dokuments aus einer Quelldatei

Aus einer `tex`-Quelldatei kann in einer Shell mittels `pdflatex` ein gleichnamiges PDF-Dokument erstellt werden. Die Syntax dabei lautet:

```
pdflatex datei.tex
```

Neben dem fertigen Dokument `datei.pdf` erzeugt der obige Aufruf zusätzlich automatisch weitere Dateien, beispielsweise `datei.aux`, `datei.log` und `datei.toc`. In diesen Dateien werden nötige Hilfsinformationen über interne Sprungstellen (beispielsweise Fußnoten und Zitate), die Ausgabe des letzten Aufrufs von `pdflatex` oder die Seitenangaben des aktuellen Inhaltsverzeichnisses abgelegt.

Hilfsdatei	Bedeutung
<code>.log</code>	Protokoll des letzten LaTeX-Aufrufs
<code>.aux</code>	Hilfsdatei für Querverweise
<code>.toc</code>	Inhaltsverzeichnis
<code>.lof</code>	Abbildungsverzeichnis
<code>.lot</code>	Tabellenverzeichnis
<code>.idx</code>	Index-Register

Üblicherweise kann man die automatisch erzeugten Hilfsdateien weitgehend außer Acht lassen. Eine Besonderheit liegt jedoch darin, dass LaTeX die Seitenzahlen für das Inhaltsverzeichnis erst dann komplett in die `.toc`-Datei schreiben kann, wenn die Quelldatei vollständig übersetzt ist. Um nach einer Veränderung einer `tex`-Quelldatei ein aktualisiertes Inhaltsverzeichnis in der `pdf`-Druckversion zu erhalten, muss `pdflatex` daher ein zweites Mal aufgerufen werden. Gleiches gilt für Änderungen im Index-Register oder im Literaturverzeichnis: Auch hier werden erst im Laufe des Übersetzungsvorgangs die Änderungen in die entsprechenden Hilfsdateien geschrieben, so dass die gemachten Änderungen erst bei einem zweiten Übersetzungsvorgang berücksichtigt werden können.

Grundlegende Konzepte in LaTeX

Im folgenden sollen einige Prinzipien, die für das Arbeiten mit LaTeX von Bedeutung sind, in einer knappen Zusammenfassung vorgestellt werden.

Anweisungen, Deklarationen und Umgebungen

In LaTeX gibt es dreierlei Arten von „Steuerelementen“, die den Ablauf des Übersetzungsvorgangs beeinflussen können:

- **Anweisungen** haben folgende Syntax:

```
\anweisungsname [optionale-argumente] {pflicht-argumente}
```

Alle Anweisungen beginnen also mit einem Backslash-Zeichen, gefolgt vom eigentlichen Namen der Anweisung. Der Anweisungsname wird allgemein durch ein Leerzeichen oder durch ein Sonderzeichen beendet.¹

Der erstere Fall tritt häufig bei Anweisungen auf, die ohne weitere Argumente aufgerufen werden, wie beispielsweise `\newpage`. Der zweite Fall tritt stets ein, wenn der Anweisung beim Aufruf optionale oder obligatorische Argumente übergeben werden, denn auch die Zeichen `[` und `{` sind Sonderzeichen. Ebenso ist es nach dieser Regel allerdings auch möglich, in einer mathematischen Formel beispielsweise `\alpha_1` für ein α -Zeichen mit dem Index 1, also α_1 zu schreiben.

Anweisungen sind in LaTeX elementare Bausteine, sie können also nicht weiter ineinander verschachtelt werden. Daher ist es beispielsweise nicht möglich, innerhalb einer `\textbf{}`-Anweisung, die den in den geschweiften Klammern stehenden Text in Fettdruck ausgibt, mittels `\newline` einen manuellen Zeilenwechsel zu erzwingen. Anweisungen beziehen sich daher meist auf einzelne Textteile innerhalb einer Zeile oder innerhalb eines Absatzes.

- **Deklarationen** sind lokale Bereiche innerhalb eines LaTeX-Dokuments, die innerhalb von geschweiften Klammern gesetzt sind, also von `{` und `}` umschlossen sind.

```
{  
    ... Inhalt ...  
}
```

Der Vorteil von Deklarationen liegt darin, dass es Anweisungen wie `\small` (Verkleinerung der Schriftart) gibt, die sich normalerweise auf den gesamten nachfolgenden Text auswirken – bis zum Ende des Dokuments, oder bis ihre Wirkung durch eine andere Anweisung aufgehoben wird – im obigen Beispiel durch `\normalsize`.

Begrenzt man den gewünschten Textabschnitt, der kleingedruckt erscheinen soll, allerdings mit geschweiften Klammern und schreibt unmittelbar nach der öffnenden Klammer `{` die Anweisung `\small`, so wird deren Wirkung mit der schließenden Klammer `}` wieder aufgehoben.

Mittels Deklarationen kann die Wirkung einzelner Anweisungen somit auch auf mehrere Zeilen oder Absätze ausgeweitet werden.

- **Umgebungen** haben stets folgende Syntax:

```
\begin{umgebung}  
    ... Inhalt ...  
\end{umgebung}
```

Umgebungen schaffen ebenso wie Deklarationen einen lokalen Bereich, in dem bestimmte Bearbeitungsmerkmale wie Schriftgröße, Textbreite, Textausrichtung oder ähnliches von den übrigen Text verschieden sein können. In LaTeX existieren zahlreiche vordefinierte Umgebungen, die eigene Änderungen mit sich bringen; im Kapitel *Wichtige Umgebungen* werden einige davon näher vorgestellt.

¹ Manche Anweisungen verlangen auch mehrere Pflicht-Argumente. Beispielsweise wird bei `\setcounter{}` innerhalb der ersten geschweiften Klammern der Name einer Zählervariablen, und innerhalb der zweiten der zuzuweisende Wert angegeben. Beide Argumente sind beim Aufruf der `\setcounter{}`-Anweisung allerdings notwendig.

Neben den lokalen Änderungen, welche die jeweiligen Umgebungen von sich aus vornehmen, können auch innerhalb von Umgebungen mittels entsprechender Anweisungen zusätzliche lokale Änderungen manuell vorgenommen werden.

Text- und Formelmodus

In LaTeX werden zwischen drei verschiedenen Arten von Text unterschieden:

- Im so genannten „Absatzmodus“ wird Text als eine gewöhnliche Sequenz von Wörtern angesehen, die bei Bedarf automatisch auf mehrere Zeilen aufgeteilt wird (inklusive automatischer Silbentrennung). Dieser Modus ist Standard bei der Eingabe von Text.
- Im so genannten „LR-Modus“ wird Text ebenfalls als Sequenz von Wörtern angesehen, die von links nach rechts abgearbeitet beziehungsweise übersetzt wird; im Gegensatz zum Absatzmodus kann in diesem Modus allerdings kein Zeilenumbruch zwischen den Wörtern stattfinden.
- Im so genannten „Mathematik-Modus“ werden die die eingegebenen Buchstaben und Symbole als Teil einer Formel interpretiert. Hierbei werden beispielsweise Leerzeichen, die zwischen einzelnen Buchstaben stehen, ignoriert; dafür sind bestimmte Syntax-Elemente wie $\hat{}$ oder $_$ zum Hochbeziehungsweise Tiefstellen von Textelementen ausschließlich in diesem Modus erlaubt.

Da sich der Absatz-Modus und der LR-Modus nur hinsichtlich des Zeilenumbruch-Verhaltens unterscheiden, werden beide oftmals gemeinsam als „Textmodus“ bezeichnet, um sie von der Eingabe mathematischer Formeln zu unterscheiden.

Gewöhnliche Textsatzung

Abschnitte und Überschriften

Der Inhalt eines Dokuments kann mittels der folgenden Anweisungen hierarchisch in einzelne Abschnitte untergliedert werden:

- `\part{Text}` kennzeichnet Buchteile. Diese Gliederungsebene existiert nur bei den Dokumentklassen `book` und `scrbook`.
- `\chapter{Text}` kennzeichnet Kapitel. Diese Gliederungsebene existiert nur bei den Dokumentklassen `book` und `scrbook`, `report` und `scrreprt`, nicht aber bei `article` und `scrartcl`.
- `\section{Text}` kennzeichnet Abschnitte. Diese Gliederungsebene ist bei Artikeln die höchste, bei Büchern oder Berichten die zweithöchste.
- `\subsection{Text}` und `\subsubsection{Text}` kennzeichnet Unter- und Unterunterabschnitte.
- `\paragraph{Text}` kennzeichnet einzelne Absätze (Paragraphen) im Text. Die tiefste Ebene stellt schließlich der `\subparagraph{Text}` dar.

In der Präambel kann mittels `\setcounter{secnumdepth}{tiefe}` beeinflusst werden, bis zu welcher Schachtelungstiefe eine Nummerierung der Überschriften erfolgt. Der Standardwert für `tiefe` ist 2, es erhalten also Kapitel-, Abschnitts- und Unterabschnittsüberschriften eine fortlaufende Nummer. Setzt man beispielsweise `\setcounter{secnumdepth}{1}`, so werden nur noch die Kapitel- und Abschnittsüberschriften nummeriert, bei `\setcounter{secnumdepth}{0}` nur noch die Kapitelüberschriften. Setzt man `\setcounter{secnumdepth}{-1}`, so wird überhaupt keine Überschrift nummeriert.

Wenn bei einer Überschrift ein `*` vor die öffnende geschweifte Klammer gesetzt wird (beispielsweise `\section*{Text}`), so erfolgt keine Nummerierung der Überschrift und ebenso kein Eintrag in das Inhaltsverzeichnis.

Titelseite und Abstract

Optional können zu Beginn einer LaTeX-Datei mittels der folgenden Anweisungen Informationen über den Autor und das aktuelle Dokument festgelegt werden:

- `\title{}`: Der Titel des Dokuments
- `\author{}`: Der Autor beziehungsweise die Autoren (Aufzählung mittels `\and`)
- `\date{}`: Das Datum der Veröffentlichung (`\today` für das heutige Datum)

Zu Beginn des Dokuments kann dann mittels folgender Anweisung eine automatische Titelseite erzeugt werden:

```
\maketitle
```

Die Titelseite kann selbstverständlich auch manuell gestaltet werden. Bei Büchern folgt auf der zweiten Seite häufig eine kurze Beschreibung des Inhalts, ein so genannter „Abstract“. Dieser kann folgendermaßen

```
\abstract{Text}
```

Die Ausrichtung des Abstracts auf der zweiten Seite erfolgt dabei – ebenso wie bei `\maketitle` – automatisch.

Inhaltsverzeichnis

Anhand der Gliederung in Kapitel und Abschnitte kann in LaTeX mittels `\tableofcontents` ein automatisch erzeugtes Inhaltsverzeichnis in das Dokument eingebunden werden:

```
\tableofcontents
```

Üblicherweise wird das Inhaltsverzeichnis unmittelbar nach der Titelseite und dem Abstract gesetzt. Die Seiten des Inhaltsverzeichnisses werden von LaTeX automatisch mit kleinen römischen Buchstaben nummeriert, das eigentliche Dokument beginnt dann mit 1 als Seitennummer.

In der Präambel kann mittels `\setcounter{tocdepth}{tiefe}` beeinflusst werden, bis zu welcher Schachtelungstiefe die Überschriften des Dokuments aufgelistet werden sollen. Der Standardwert für `tiefe` ist wiederum 2, so dass Kapitel-, Abschnitts- und Unterabschnittsüberschriften aufgelistet werden. Nummer. Setzt man beispielsweise `\setcounter{secnumdepth}{1}`, so werden nur noch die Kapitel- und Abschnittsüberschriften aufgelistet. Überschriften, bei denen vor die öffnende Klammer ein `*` gesetzt wurde (beispielsweise `\section*{Text}`), werden unabhängig davon im Inhaltsverzeichnis nicht aufgelistet.

Mit folgender Anweisung kann mitten im Dokument manuell ein Eintrag zum Inhaltsverzeichnis hinzugefügt werden, der dann auf die jeweilige Textstelle verweist:

```
% Eintrag manuell zum Inhaltsverzeichnis hinzufügen:  
\addcontentsline{toc}{section}{Name des Eintrags}
```

Anstelle `section` kann selbstverständlich auch eine andere Gliederungsebene wie `chapter` oder `subsection` gewählt werden. Manuelle Einträge sind an Stellen sinnvoll, an denen man den Navigationskomfort des Inhaltsverzeichnisses (inklusive der automatisch generierten Sprungmarken) nutzen mag, ohne im Dokument (beispielsweise aus Platzgründen) tatsächlich eine Überschrift setzen zu wollen.

Werden Änderungen in der Gliederung vorgenommen, so schreibt LaTeX beim Übersetzen des Quellcodes mittels `pdflatex` die Änderungen des Inhaltsverzeichnisses in eine Hilfsdatei mit der Endung `:math:.toc`; erst bei einem nochmaligen Übersetzen des Quellcodes mittels `pdflatex` werden die Änderungen auch im PDF-Dokument sichtbar.

Tabellen- und Abbildungsverzeichnis

Neben einem Inhaltsverzeichnis kann nach Belieben auch mit `\listoftables` eine Übersicht aller Tabellen und mit `\listoffigures` eine Übersicht über alle Abbildungen im Dokument ausgegeben werden. Auch hier ist gegebenenfalls ein zweimaliger Aufruf von `pdflatex` nötig, um die Listen (Hilfsdateien `.lot` und `.lof`) im PDF-Dokument zu aktualisieren.

Anhang

Soll ein Dokument einen Anhang beinhalten, so kann dieser mittels `\appendix` eingeleitet werden. Ab dieser Anweisung werden weitere Kapitel anstelle mit Nummern mit Großbuchstaben A, B, usw. durchnummeriert. Die Seitennummerierung wird unverändert fortgesetzt.

Zeilen- und Seitenumbruch

Einzelne Absätze werden in LaTeX durch Leerzeilen gekennzeichnet. Der Abstand zwischen Absätzen wird als Option in Verbindung mit der Dokumentklasse festgelegt, beispielsweise `\documentclass[halfparskip]{scrbook}`.

Möchte man innerhalb eines Absatzes eine neue Zeile erzeugen, so kann dies mittels `\newline` oder der Kurzform `\\` erfolgen. Unmittelbar im Anschluss an `\\` kann zudem in eckigen Klammern ein Längenmaß angegeben werden, um das der Abstand zur nächsten Zeile verkleinert oder vergrößert wird; beispielsweise bewirkt bei einer Schriftgröße von 12pt ein Zeilenumbruch mittels `\\[6pt]` einen 1,5-fachen Zeilenabstand.

Eine Wechsel auf eine neue Seite kann mittels `\pagebreak` oder `\newpage` manuell erzwungen werden.

Aufteilung eines Dokuments in mehrere Dateien

Umfangreiche LaTeX-Dokumente können, beispielsweise kapitelweise, in mehrere `.tex`-Dateien aufgeteilt werden. In der Hauptdatei kann der Inhalt dieser Dateien dann mittels der Anweisung `\input{kapitelname}` eingefügt werden; die Endung `.tex` wird dabei automatisch ergänzt.

```
% Beispiel einer aufgeteilten Haupt-Datei:
```

```
\input{preamble}  
\input{kapitel1}  
\input{kapitel2}  
\input{kapitel3}  
\input{anhang}
```

Die einzelnen Kapitel können bei umfangreichen Dokumenten auch in verschiedene Unterverzeichnisse abgelegt werden, um dort beispielsweise die zugehörigen Bild- oder Code-Dateien mit abzuspeichern.

Auf eine sehr ähnliche Weise kann die LaTeX-Anweisung `\include{}` verwendet werden. Auch bei dieser Anweisung wird der Name der angegebenen Datei um die Endung `.tex` ergänzt; der Inhalt der Datei wird allerdings erst auf einer neuen Seite (quasi nach einer `\newpage`-Anweisung) eingebettet. Die `\include{}`-Anweisung kann folglich nur einzelne Kapitel-Dateien in das Dokument einbauen; beispielsweise für die Präambel muss hingegen die `\input{}`-Anweisung verwendet werden.

```
\frontmatter, \mainmatter und \backmatter
```

Bücher werden oftmals in Vorspann, Hauptteil und Nachspann aufgeteilt. Auch bei den LaTeX-Dokumentklassen `book` beziehungsweise `scrbook` existieren entsprechende Gliederungsebenen:

```
% Vorspann einleiten:
```

```
\frontmatter
```

```
% Hauptteil einleiten:
```

```
\mainmatter
```

```
% Nachspann einleiten:
```

```
\backmatter
```

- Im Vorspann werden die Seiten mit kleinen römischen Ziffern durchnummeriert. Der Vorspann enthält üblicherweise nur die Titelseite, das Inhaltsverzeichnis (bisweilen auch ein Abbildungs- und Tabellenverzeichnis) sowie möglicherweise ein Vorwort. Meist kommt der Vorspann daher mit einer einzigen Kapitelüberschrift aus, die ohne Nummerierung erscheint.¹

¹ Sollte der Vorspann, was nicht empfehlenswert ist, durch `\section{}`-Anweisungen weiter untergliedert werden, so sollte bevorzugt die `\section*{}`-Anweisung für nicht-nummerierte Abschnitts-Überschriften gewählt werden.

- Im Hauptteil werden die Seiten mit arabischen Zahlen nummeriert, wobei die Nummerierung erneut mit 1 beginnt.
- Im Nachspann wird bisweilen nur das Literatur- und/oder Stichwortverzeichnis gesetzt, manchmal jedoch auch der gesamte Anhang. Der Nachspann gleicht bezüglich der Gliederungsüberschriften dem Vorspann, die Kapitel werden jedoch mit Großbuchstaben gekennzeichnet (beginnend mit A). Eine getrennte Seitennummerierung ist nicht vorgesehen.

Schriftfamilien und Text hervorhebungen

Üblicherweise werden mehrere Schriftarten zu einer Schriftfamilie zusammengefasst, die zwar das gleiche Design aufweisen, sich aber in Neigungsgrad, Fettschrift, Zeichenabstand und weiteren Merkmalen unterscheiden können. Die Standard-Schriftfamilie von LaTeX heißt „Computer Modern“.

Innerhalb einer Schriftfamilie kann die Darstellung von Text folgendermaßen verändert werden:

- *Schriftstärke*: Mit `\textbf{Text}` („bold font“) wird Text fettgedruckt, mit `\textmd{Text}` („medium“, Standard) in normaler Schriftstärke ausgegeben.

<code>\textbf{Ein Blindtext.}</code>	Ein Blindtext.
<code>\textmd{Ein Blindtext.}</code>	Ein Blindtext.

- *Schriftform*: Mit `\textit{Text}` („italic“) wird der angegebene Text kursiv, mit `\textsl{Text}` („slanted“) schräggestellt gedruckt. Als Standard wird die Text aufrecht ausgegeben (entspricht `\textup{Text}`).

<code>\textit{Ein Blindtext.}</code>	<i>Ein Blindtext.</i>
<code>\textsl{Ein Blindtext.}</code>	<i>Ein Blindtext.</i>
<code>\textup{Ein Blindtext.}</code>	Ein Blindtext.

Mit `\emph{Text}` („emphasize“) wird Text hervorgehoben; er wird dann innerhalb einer normalen Textzeile kursiv gedruckt, innerhalb einer kursiven Textzeile jedoch aufrecht dargestellt. Mit `\textsc{Text}` („small caps“) wird Text in unterschiedlich großen Großbuchstaben („Kapitälchen“) ausgegeben; mittels `\underline{Text}` kann Text unterstrichen ausgegeben werden.

<code>\emph{Ein Blindtext.}</code>	<i>Ein Blindtext.</i>
<code>\textsc{Ein Blindtext.}</code>	Ein BLINDTEXT.
<code>\underline{Ein Blindtext.}</code>	<u>Ein Blindtext.</u>

Bindet man in der Präambel das Paket `soul` ein (`\usepackage{soul}`), so kann Text auch durchgestrichen ausgegeben werden:

<code>\st{Ein Blindtext.}</code>	Ein Blindtext.
----------------------------------	---------------------------

- *Schrifttyp*: Standardmäßig wird Text als `\textrm{Text}` („roman“), d.h. mit so genannten Serifen ausgegeben. Serifen sind kleine Füschen und Häkchen an den einzelnen Buchstaben, die eine Schrift für das Auge besser lesbar machen. Serifen-Schriften sollten für längere Texte bevorzugt werden, während beispielsweise für die Erstellung von Plakaten mit `\textsf{Text}` („sans serif“) auch eine serifenlose Schrift gewählt werden kann.

<code>\textrm{Ein Blindtext.}</code>	Ein Blindtext.
<code>\textsf{Ein Blindtext.}</code>	Ein Blindtext.
<code>\texttt{Ein Blindtext.}</code>	Ein Blindtext.

Üblicherweise nehmen die einzelnen Buchstaben einer Schrift beim Druck, da sie unterschiedlich breit sind, unterschiedlich viel Platz ein. Beispielsweise zur Darstellung von Quellcode wird jedoch bevorzugt eine nicht-proportionale Schrift verwendet, um eine optische Absetzung vom restlichen Text zu bewirken. Dies kann mit `\texttt{Text}` („typewriter“) erreicht werden.

Die obigen Anweisungen können verschachtelt auftreten, es kann also beispielsweise ein Text fett *und* kursiv gedruckt werden. Die Anweisungen sind jedoch auf einzelne Textelemente innerhalb eines Absatzes begrenzt.

Farbiger Text

Um Farben nutzen zu können, muss in der *Präambel* des LaTeX-Dokuments das Paket `xcolor` geladen werden:

```
\usepackage{xcolor}
```

Innerhalb des Dokuments lässt sich dann durch die Anweisung `\color{farbname}` die Standardfarbe auf eine gewünschte Farbe ändern. Der Farbname kann einer Standardfarbe (`black`, `white`, `darkgray`, `gray`, `lightgray`, `brown`, `red`, `green`, `blue`, `cyan`, `magenta` oder `yellow`) entsprechen oder eine selbst definierte Farbe bezeichnen.

Um einen Farbwechsel nicht wieder explizit rückgängig machen zu müssen, kann die Wirkung des Farbwechsels mittels einer *Deklaration* auf einen Textblock beschränkt werden, beispielsweise `Text { \color{red} roter Text hier } Text`. Derartige Textblöcke können auch mehrere Absätze umfassen.

Eine andere Möglichkeit ist, einen Textabschnitt mittels `\textcolor{farbname}{Text}` hervorzuheben; hierbei ist die Länge des Textes allerdings auf maximal einen einzelnen Absatz begrenzt.

Neben den Standard-Farben können folgendermaßen auch eigene Farben definiert werden:

Beispiel:

```
\textcolor[rgb]{0.85, 0.00, 0.15}{Rot}          \ \ % rot-gruen-blau (prozentual)
\textcolor[RGB]{ 240, 115,  40}{Orange}       \ \ % Rot-Gruen-Blau (Farbwerte 0..255)
\textcolor[HTML]{FOE10F}{Gelb}               \ \ % HTML-Code      (hexadecimal)
\textcolor[cmymk]{0.25, 0.10, 0.35, 0.08}{Grün} \ \ % CMYK-Anteile   (für Druckereien)
\textcolor[wave]{450}{Blau}                  \ \ % Wellenlänge    (380..750)

\textcolor[gray]{0.75}{Gru}                   \ \ % Grau-Anteil    (0..1)
```

Ergebnis:

Rot
Orange
Gelb
Grün
Blau

Gru

Möchte man eine Farbe nicht nur einmalig verwenden, so kann man ihr einen Namen zuweisen; die Farbe kann anschließend wie eine Standard-Farbe verwendet werden:

```
% Eigenen Farbnamen definieren:
\definecolor{deppurple}{RGB}{50, 25, 150}
```

Mittels des `xcolor`-Pakets kann Text auch mit einem farbigen Rahmen versehen werden:

Beispiel:

```
% Text mit farbigem Hintergrund:
\colorbox{blue}{Text}

% Text mit farbiger Umrandung:
\fcolorbox{blue}{white}{Text}
```

Ergebnis:

Text

Text

Mit beiden Anweisungen werden *LR-Boxen* erzeugt (siehe Abschnitt *Boxen* weiter unten). In einer derartigen Box ist kein allgemein kein Zeilenumbruch möglich, außer man umschließt damit eine *Parbox*. Alle von `xcolor` bereitgestellten Anweisungen lassen sich auch im Mathe-Modus und in Tabellen nutzen; für farbige Texteinträge in Tabellen lohnt es sich allerdings, auch einen Blick auf das *colortbl*-Paket zu werfen.

Allgemein sollten Farben sparsam verwendet werden. Zueinander passende Farben lassen sich beispielsweise mit dem Programm `agave` auffindig machen – dieses ist über das gleichnamige Paket via `apt` installierbar.

Gedrehter Text

Um Text vertikal oder schräg zu setzen, muss zunächst in der Präambel des Dokuments das Paket `graphicx` geladen werden. Innerhalb des Dokuments lässt sich dann durch die Anweisung `\rotatebox{Winkel}{Text}` der angegebene Text um den angegebenen Winkel drehen.

Beispiel:

```
% In der Präambel:
\usepackage{graphicx}

% Innerhalb des Dokuments:
\rotatebox{90}{Hallo Welt!}
```

Der Drehwinkel wird dabei im mathematischen Sinn interpretiert, eine Angabe von 90 entspricht also einer Drehung um 90° gegen den Uhrzeigersinn.

Schriftgrößen und Längenmaße

In LaTeX wird die Standard-Größe für ein Dokument in Verbindung mit der Dokumentklasse festgelegt, beispielsweise `\documentclass[12pt]{scrbook}` für ein Buch mit einer normalen Schriftgröße von 12pt. Die Größe einer Schrift kann dann innerhalb des Dokuments folgendermaßen angepasst werden, wobei die Abstufungen relativ zur Standard-Schriftgröße und in harmonischen Größenverhältnissen erfolgen:

- `\tiny`
- `\scriptsize`
- `\footnotesize`
- `\small`
- `\normalsize`
- `\large`
- `\Large`
- `\LARGE`
- `\huge`
- `\Huge`

Eine Änderung der Schriftgröße kann entweder mittels `\normalsize` beendet oder mittels einer *Deklaration* auf einen Textblock beschränkt werden, beispielsweise `Text { \large großer Text hier } Text`.

Derartige Textblöcke können auch mehrere Absätze umfassen; ebenso kann ein solcher Bereich mittels `\begin{large}` und `\end{large}` begrenzt werden.

Soll innerhalb des Dokuments die Standard-Größe der Schrift verändert werden, so ist dies mit folgender Syntax möglich:

```
\fontsize{12pt}{14.4pt}
\selectfont
```

Der `\fontsize{}`-Anweisung werden hierbei die neue Schriftgröße sowie der neue Standard-Zeilenabstand als Argumente übergeben. Möchte man eine andere Schriftgröße angeben, so sollte das Verhältnis von Schriftgröße sowie Zeilenabstand beibehalten, also beide Werte um den gleichen Faktor skaliert werden.

In LaTeX können allgemein folgende Längenmaße verwendet werden:

- **in**: Ein Zoll („inch“) entspricht 2,54 cm.
- **pt**: Der „point“ ist eine Maßgröße aus der ursprünglichen Textsatzung. Es gilt $1\text{ pt} \approx 0,0351\text{ cm}$
- **em**: Ein em war früher als die Breite des großen ‚M‘ definiert. Bei der LaTeX-Standard-Schrift („Computer Modern“) sind beispielsweise Ziffern `0.5em` breit.
- **ex**: Ein ex ist in etwa die Höhe des kleinen ‚x‘.

Spezielle Zeichen

LaTeX kennt drei Arten von Zeichen: Normale Zeichen, Steuerzeichen und Sonderzeichen:

- Normale Zeichen sind alle „normalen“ Buchstaben (a bis z beziehungsweise A bis Z) sowie die Ziffern und Satzzeichen.
- Steuerzeichen steuern LaTeX. Das Steuerzeichen `\` bedeutet beispielsweise, dass anschließend eine LaTeX-Anweisung folgt; so bewirkt beispielsweise `\textbf{Text}`, dass der Text innerhalb der geschweiften Klammern fett gedruckt werden soll.

Sollen Zeichen, die in LaTeX Sonderbedeutungen als Steuerzeichen haben, als normale Zeichen gedruckt werden, so müssen sie gemäß der folgenden Tabelle im laufenden Text eingegeben werden:

Eingabe	Ausgabe	Eingabe	Ausgabe
<code>\{</code>	{	<code>\}</code>	}
<code>\#</code>	#	<code>\&</code>	&
<code>_</code>	_	<code>\%</code>	%
<code>\\$</code>	\$	<code>\^{}{}</code>	^
<code>\textasciitilde</code>	~	<code>\textbackslash</code>	\

- Umlaute und andere Sonderzeichen sind oftmals länderspezifisch. Für deutschsprachige Dokumente sollten daher, wie bereits im Abschnitt *Umlaute und deutsche Sprachunterstützung* beschrieben, in der Präambel des Dokuments folgende Pakete eingebunden werden:

```
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

Gewöhnliche deutschsprachige Umlaute können damit wie normale Zeichen im laufenden Text eingegeben werden. Weitere Sonderzeichen werden üblicherweise mittels eines `\`-Zeichens eingeleitet.

Eingabe	Ausgabe	Eingabe	Ausgabe
<code>\`a</code>	à	<code>\'a</code>	á
<code>\.a</code>	â	<code>\^a</code>	â
<code>\~a</code>	ã	<code>\={a}</code>	ā
<code>\u{a}</code>	ǎ	<code>\v{a}</code>	ǻ
<code>\k{a}</code>	ą	<code>\c{c}</code>	ç
<code>\"a</code>	ä	<code>\ss</code>	ß

Beispielsweise können also spanisch-sprachige Umlaute mittels `\`` und `\~` vor dem eigentlichen Buchstaben erzeugt werden. So ergibt die Eingabe von `aqu\`i` als Ausgabe `aquí`; eine Eingabe von `sen\~nor` ergibt entsprechend `señor`. Bei Verwendung der beiden obigen Pakete und einer spanisch-sprachigen Tastatur können die Umlaute und Sonderbuchstaben jedoch auch als normaler Text eingegeben werden.

Einige weitere Sonderzeichen sind in der folgenden Tabelle aufgelistet:

Eingabe	Ausgabe	Eingabe	Ausgabe
<code>\oe</code>	œ	<code>\OE</code>	Œ
<code>\ae</code>	æ	<code>\AE</code>	Æ
<code>\aa</code>	å	<code>\AA</code>	Å
<code>\o</code>	ø	<code>\O</code>	Ø
<code>\l</code>	ł	<code>\L</code>	Ł
<code>\P</code>	¶	<code>\S</code>	§
<code>\textexclamdown</code>	¡	<code>\textquestiondown</code>	¿
<code>\pounds</code>	£	<code>\copyright</code>	©

Für manche Sonderzeichen müssen zusätzliche Pakete geladen werden; beispielsweise sollte in der Präambel grundsätzlich das Paket `marvosym` geladen werden, da damit unter anderem mittels `\EUR` das Euro-Zeichen € gesetzt werden kann.

Griechische Buchstaben werden gewöhnlich mit Hilfe des Mathematik-Modus eingegeben; dabei werden sie allerdings als Bezeichnungen für Variablen angesehen und damit kursiv gedruckt. Sollen griechische Buchstaben in Normalschrift in den Text eingebaut werden, so kann beispielsweise in der Präambel das Paket `textgreek` geladen und anschließend die Buchstaben mittels `\textalpha`, `\textbeta` usw. gesetzt werden. Als Alternative kann anstelle des Pakets `babel` das Paket `betababel` mit den gleichen Optionen (beispielsweise `ngerman`) geladen werden, um innerhalb des Dokuments beispielsweise mittels `\bcode{logos}` den Schriftzug $\lambda\gamma\omicron\varsigma$ zu erhalten.

Eine sehr ausführliche Übersicht von LaTeX-Symbolen gibt es im [LaTeX-Wikibook \(Sonderzeichen\)](#) und in der PDF-Datei [Symbols-A4](#).

Silbentrennung

In LaTeX wird eine sprachspezifische Silbentrennung in der Präambel über das Paket `babel` aktiviert, beispielsweise wird mittels `\usepackage[ngerman]{babel}` die Silbentrennung für die neue deutsche Rechtschreibung aktiviert. Die Silbentrennung erfolgt in LaTeX automatisch, kann allerdings manuell angepasst werden.

- Soll an einer Leerstelle ein Zeilenumbruch verhindert werden, kann anstelle eines Leerzeichens das Tilde-Zeichen `~` eingesetzt werden; beispielsweise würde `Seite~9` nicht zwischen `Seite` und `9` getrennt.
- Soll an einer bestimmten Stelle innerhalb eines Wortes ein Zeilenumbruch erzwungen werden, so ist dies mittels `\-` möglich, beispielsweise `Archeo\-``ptenix`. Der Zeilenumbruch an dieser Stelle wird allerdings nur dann durchgeführt, wenn das Wort auch am Ende einer Zeile steht und getrennt werden muss; andernfalls wird die Trenn-Anweisung `\-` von LaTeX ignoriert.

Dank des `babel`-Pakets werden zwar die meisten Wörter der deutschen Sprache bei Zeilenumbrüchen richtig getrennt. Kommen im Text allerdings Wörter vor, für die keine mögliche Silbentrennung bekannt ist, so kann der `\hyphenation{}`-Anweisung am Ende der Präambel eine Liste mit Trenn-Empfehlungen festgelegt werden:²

```
% Trennempfehlungen für folgende Wörter festlegen:
\hyphenation{Ar-choe-pte-rix Stau-becken Nach-kom-ma Stel-len}
```

Die `\hyphenation{}`-Liste kann beliebig lang sein und sollte alle Wörter umfassen, die beim Durchblättern des fertigen PDF-Dokuments am rechten Seitenrand auffallen, weil sie nicht automatisch getrennt werden konnten.

Vertikale und horizontale Abstände

Einzelne Absätze werden in LaTeX durch leere Zeilen voneinander getrennt. Kommen mehrere aufeinander folgende leere Zeilen vor, so werden die folgenden ignoriert, der Abstand zwischen den einzelnen Absätzen bleibt somit gleich.

Um den vertikalen Abstand zwischen einzelnen Zeilen zu verändern, gibt es mehrere Möglichkeiten:

- Mit beispielsweise `\\[6pt]` wird eine neue Zeile eingeleitet mit einem zusätzlichen Abstand von `6pt` (entspricht etwa 2mm).
- Zwischen zwei Absätzen kann mittels `\vspace{Länge}` ein beliebig langer vertikaler Abstand an dieser Stelle eingefügt werden, beispielsweise mittels `\vspace{3cm}` ein 3cm breiter vertikaler Abstand.
- Lädt man in der Präambel mittels `\usepackage{setspace}` das Zusatz-Paket `setspace`, so kann man mittels der Anweisungen `\onehalfspacing` beziehungsweise `\doublespacing` im folgenden Dokumentteil einen eineinhalb-fachen beziehungsweise doppelten Zeilenabstand einstellen. Der ursprüngliche Zeilenabstand kann mittels `\singlespacing` wieder hergestellt werden.
- Möchte man einen anderen, selbst definierten Zeilenabstand wählen, so ist dies mittels der `spacing`-Umgebung möglich:

```
% In der Präambel:

\usepackage{setspace}

% Innerhalb des Dokuments:

\begin{spacing}{Zahl}

    % ... Inhalt ...

\end{spacing}
```

Im diesem Beispiel würde ein angegebener Zahlenwert von 1,5 zu einem eineinhalb-fachen Zeilenabstand führen. Gibt man einen Wert kleiner als Eins an, so wird der Zeilenabstand entsprechend verkleinert.

Innerhalb der einzelnen Zeilen wird das Leerzeichen als Worttrennzeichen verwendet; hierbei werden, wenn mehrere aufeinander folgende Leerzeichen vorkommen, die folgenden ignoriert. LaTeX richtet in Textbereichen *automatisch* die Abstände zwischen den einzelnen Worten (und sogar den Abstand zwischen den Buchstaben innerhalb der Worte) so aus, dass sich – unter Berücksichtigung möglicher Silbentrennungen – innerhalb des jeweiligen Absatzes ein möglichst harmonisches Gesamtbild ergibt.

Um horizontale Abstände einzufügen, gibt es ebenfalls mehrere Möglichkeiten:

² Man kann sich innerhalb des Dokuments auch beispielsweise mittels `\showhyphens{Staubacken}` ausgeben lassen, wie das angegebene Wort von LaTeX automatisch getrennt würde; als Ergebnis erhält man an dieser Stelle als Ausgabe `\tenrm Staub-acken`.

- Mittels `\hspace{}` kann an der jeweiligen Stelle ein horizontaler Abstand mit einer festgelegten Länge erzeugt werden, beispielsweise mittels `\hspace{1 in}` ein Abstand von einem Zoll. Auch beliebige andere *Längeneinheiten* können gewählt werden.
- Mittels `\fill` wird in der aktuellen Zeile so viel Platz eingefügt, dass der anschließend eingegebene Text rechtsbündig am Zeilenrand gedruckt wird. Beispielsweise kann man mittels `\hfill \today` erreichen, dass das aktuelle Datum am rechten Rand der aktuellen Zeile ausgegeben wird.
- Mittels der ``-Anweisung kann man einen horizontalen Abstand einfügen, der ebenso lang ist wie der innerhalb der geschweiften Klammern angegebene Text. Mit dieser Anweisung wird also Platz für den als Argument eingegebenen Text freigehalten, dieser aber nicht gedruckt.

Fußnoten und Randnotizen

Innerhalb eines Textabschnitts kann mit `\footnote{Text}` eine Fußnote erstellt werden. Der angegebene Text wird dabei in einer kleineren Schrift an das Seitenende geschrieben und automatisch mit einer Nummerierung versehen.

Beispiel:

```
Hier ist ein Text.\footnote{Und hier ist die zugehörige Fußnote --
automatisch nummeriert und an der richtigen Stelle platziert!}
```

Standardmäßig werden Fußnoten in den Dokumentenklassen `article` und `scrartcl` durch das gesamte Dokument fortlaufend nummeriert, bei den Dokumentklassen `book`, `scrbook`, `report` und `scrreprt` findet eine Nummerierung kapitelweise statt. Überlange Fußnoten werden von LaTeX automatisch auf mehrere aufeinander folgende Seiten aufgeteilt.

Neben Fußnoten können beispielsweise einzelne Schlagwörter oder Kurzbeschreibungen auch auf den Seitenrändern ausgegeben werden. Die Anweisung hierfür lautet `\marginpar{Text}`, wobei der angegebene Text an der jeweiligen Stelle im Dokument auf den Außenrand der Seite gedruckt wird. Möchte man auf den anderen Rand der Seite drucken, so kann die Standard-Einstellung mittels `\reversemarginpar` geändert werden.

Querverweise

Innerhalb eines Dokumentes können beliebige Stellen mittels so genannten Labels markiert werden. Die Syntax für eine solche Markierung ist:

```
% Label erzeugen:
\label{Stichwort}
```

Von anderen Stellen aus kann auf die markierten Stellen mittels Querverweisen („Referenzen“) Bezug genommen werden. Es ist empfehlenswert, für unterschiedliche Arten von Sprungmarken eigene Label-Präfixe zu verwenden, beispielsweise können mit `eq-` beginnende Labels für *Gleichungen*, mit `fig-` beginnende Labels für *Abbildungen*, und mit `tab-` beginnende Labels *Tabellen* gesetzt werden.

Beim Verweis auf die Sprungmarke kann entweder die Kapitel- oder die Seitennummer angezeigt werden:

```
% Auf Label verweisen:
Siehe Kapitel \ref{Stichwort} auf Seite \pageref{Stichwort}.
```

Innerhalb eines Dokumentes können Querverweise auch auf sich weiter hinten befindende Labels beziehen. Die einzelnen Querverweise werden beim Erzeugen der fertigen PDF-Datei mittels `pdflatex` in eine Hilfsdatei mit der Endung `.aux` gespeichert. Änderungen bei Sprungmarken werden im Allgemeinen erst beim zweiten Durchlauf von `pdflatex` wirksam.

Sprungmarken können nicht nur auf andere Stellen im gleichen Dokument, sondern unter Verwendung des Zusatz-Paketes *hyperref* beispielsweise auch auf Web-Adressen gesetzt werden.

Boxen, Balken und Minipages

Ein Grundprinzip von LaTeX besteht darin, sämtliche Inhalte einer Seite auf verschiedene Boxen aufzuteilen und diese dann zu platzieren. Die wichtigsten Typen von Boxen sind folgende:

- LR-Box:

In einer solchen Box wird Text von links nach rechts *ohne* Zeilenumbruch gesetzt.

LR-Boxen können wahlweise mit oder ohne Rahmen gesetzt werden. Soll sich die Größe der Box nach dem darin enthaltenen Text richten, so können folgende Anweisungen genutzt werden:

```
% LR-Box ohne Rahmen setzen:  
\mbox{Ein Text-Beispiel.}  
  
% LR-Box mit Rahmen setzen:  
\fbox{Ein Text-Beispiel.}
```

Die Höhe einer LR-Box wird von LaTeX automatisch anhand der Größe des Inhalts ermittelt. Enthält beispielsweise eine `fbox` nur einen kurzen Text, so wird die Höhe der Box auf eine Zeilenhöhe festgelegt; enthält sie hingegen eine Abbildung mit Bildunterschrift, so wird die Höhe der Box daran festgelegt. Mittels LR-Boxen können also beliebige Objekte mit einem Rahmen versehen werden.

Weitere LR-Boxen ohne feste Breite können beispielsweise mittels der Anweisungen `\shadowbox{}`, `\doublebox{}` oder `\ovalbox{}` gesetzt werden; diese Boxen sind dann mit einer Schattierung hinterlegt beziehungsweise doppelt oder oval umrandet.

Soll die Breite einer LR-Box explizit vorgegeben werden, so können folgende Anweisungen genutzt werden:

```
% LR-Box mit fester Breite setzen (ohne Rahmen):  
% Optionale Argumente: Breite, Ausrichtung (l,c,r)  
\makebox[8.0cm][c]{Ein Text-Beispiel (zentriert).}  
  
% LR-Box mit fester Breite setzen (mit Rahmen):  
\framebox[8.0cm][c]{Ein Text-Beispiel (zentriert).}
```

Bei LR-Boxen *mit* Umrandung kann die Dicke des Rahmens und der Leerraum zwischen Umrandung und Inhalt über folgende zwei Variablen festgelegt werden:

```
% Rahmenbreite festlegen:  
\setlength{\fboxrule}{0.1cm}  
  
% Abstand zwischen Rahmen und Inhalt festlegen:  
\setlength{\fboxsep}{0.5cm}
```

Um farbige Boxen zu setzen, kann, wie *weiter oben beschrieben*, in der Präambel das Paket `xcolor` geladen werden; innerhalb des Dokuments können dann die Anweisungen `\colorbox{farbname}{text}` beziehungsweise `\fcolorbox{rahmenfarbe}{boxfarbe}{Text}` genutzt werden.

Beispiel:

```
% In der Präambel:  
% \usepackage{xcolor}  
  
% ...  
% Im Dokument:  
  
\fcolorbox{blue}{white}{\makebox[8.0cm][c]{Hallo Welt!}}
```

Ergebnis:

Hallo Welt!

Vom Paket `fancybox` werden weitere LR-Boxen bereitgestellt:

- Mit `\ovalbox{Text}` beziehungsweise `\Ovalbox{Text}` kann der angegebene Text mit einer ovalen Umrandung versehen werden.
- Mit `\doublebox{Text}` wird der angegebene Text mit einem doppelten Rahmen versehen.
- Mit `\shadowbox{Text}` wird der angegebene Text mit einem Rahmen mit Schattierung versehen.

Beispiel:

```
% In der Präambel:  
% \usepackage{fancybox}  
  
% ...  
% Im Dokument:  
  
% Box mit Schattierung:  
\shadowbox{\makebox[8.0cm][c]{Hallo Welt!}}
```

Ergebnis:



Hallo Welt!

Auch das Paket `shadow` stellt mit der Anweisung `\shabox{}` eine schattierte LR-Box bereit; bei dieser sind die Abstände zur Umrandung bewusst groß gewählt, so dass sich diese Box beispielsweise gut für Titelseiten eignet

Beispiel:

```
% In der Präambel:  
% \usepackage{shadow}  
  
% Box mit Schattierung:  
\shabox{\makebox[8.0cm][c]{Hallo Welt!}}
```

Ergebnis:



Hallo Welt!

Da bei LR-Boxen kein Zeilenumbruch erfolgt, muss man selbst darauf achten, dass der Inhalt nicht über den Rand der Box beziehungsweise über den Rand der Seite hinausragt; ein weiterer Trick besteht in der Nutzung der im folgenden beschriebenen Parboxen.

- **Parbox:**

In einer solchen Box wird Text als Blocksatz gesetzt, also gegebenenfalls *mit* Zeilenumbruch. Insgesamt darf der Text, wie der Name der Box schon andeutet, maximal einen Absatz („Paragraph“) umfassen. Die Syntax dafür lautet etwa folgendermaßen:

```
% Parbox mit 8cm Breite erstellen:  
\parbox[c]{8.0cm}{\blindtext}
```

Die `\parbox{}`-Anweisung kann auch mit zusätzlichen Argumenten aufgerufen werden, und zwar mit der Syntax `\parbox[pos][hoehe][ipos]{breite}{Text}`. Hierbei gibt `pos` die vertikale Ausrichtung der Box am aktuellen Absatz an; als mögliche Ausrichtungen sind dabei `t` für „top“, `c` für „center“ oder `b` für „bottom“ möglich. Wird zudem über das optionale Argument `hoehe` die Höhe der Parbox explizit angegeben, so kann mittels des optionalen Arguments `ipos` festgelegt werden, wie die vertikale Positionierung innerhalb der Parbox erfolgen soll. Neben `b`, `c` und `t` gibt es hier auch die Option `s` („stretched“) für eine gleichmäßige Verteilung des Inhalts auf die gesamte Höhe der Box.

Parboxen können auch innerhalb von LR-Boxen verwendet werden, um in diesen indirekt Zeilenbrüche zu ermöglichen. Beispielsweise kann man so einen farbigen Rahmen um eine mehrzeilige Formel zeichnen:

```
% In der Präambel:  
% \usepackage{shadow}  
  
\shabox{  
  \parbox{0.10\textwidth}{  
    Hallo \\  
    Welt!  
  }  
}
```




Hallo
Welt!

- Rule-Box:

Eine solche Box dient vorwiegend zum Setzen von Linien und Balken.

Rule-Boxen werden hauptsächlich benutzt, um einzelne Text-Abschnitte untereinander abzugrenzen.

```
% Horizontale Linie mit 5 cm Breite und 0.5 cm Höhe erstellen:  
% Allgemeine Syntax: \rule[verschiebung]{breite}{hoehe}  
\rule[0cm]{5.0cm}{0.5cm}
```



Über das erste (optionale) Argument kann eine vertikale Verschiebung des Balkens gegenüber der Grundlinie der aktuellen Zeile angegeben werden. Für die Breite oder die Höhe kann bei Bedarf auch der Wert Null angegeben werden, um einen unsichtbaren Balken zu erstellen; hierdurch kann beispielsweise ein vertikaler Versatz an einer Stelle bewirkt werden, an dem die Anweisung `\vspace{}` nicht möglich ist (beispielsweise innerhalb eines Absatzes oder innerhalb einer LR-Box).

Minipages

Eine `parbox` kann nur einen einzelnen Absatz beinhalten. Möchte man allerdings mehrere Absätze oder sogar Abbildungen und Tabellen in eine einzelne Box packen, so bietet sich hierfür die sogenannte „Minipage“-Umgebung an.³

Die Syntax der Minipage-Umgebung ist der einer *parbox* sehr ähnlich:

```
% Minipage mit 8cm Breite erstellen:  
  
\begin{minipage}[c]{8.0cm}  
  
  % ... Inhalt ...  
  
\end{minipage}
```

Auch die `minipage`-Umgebung kann auch mit zusätzlichen Argumenten aufgerufen werden, und zwar mit der Syntax `\begin{minipage}[pos][hoehe][ipos]{breite}`. Hierbei gibt `pos` wiederum die vertikale Ausrichtung der Box zwischen dem vorherigen und dem nächsten Absatz an; als mögliche Ausrichtungen sind dabei `t` für „top“, `c` für „center“ oder `b` für „bottom“ möglich. Wird zudem über das optionale

³ Eine Einschränkung bei Minipages besteht darin, dass sie keine Fließumgebungen beinhalten dürfen; beispielsweise dürfen mittels der `tabular`-Umgebung gesetzte Tabellen enthalten sein; diese darf jedoch nicht von einer `table`-Umgebung umschlossen sein, da bei dieser die konkrete Position nicht unmittelbar festgelegt ist.

Argument `hoehe` die Höhe der Parbox explizit angeben, so kann mittels des optionalen Arguments `ipos` festgelegt werden, wie die vertikale Positionierung innerhalb der Parbox erfolgen soll.

Mathematischer Formelsatz

LaTeX unterstützt wie kaum ein anderes Textsatzungsprogramm das Einfügen mathematischer Formeln. Wahlweise können Formeln innerhalb einer Absatzes („Inline“) oder als eigenständiger Absatz in ein Dokument eingefügt werden:¹

- Mit `$ Formel $` wird eine Formel innerhalb eines Textabsatzes eingefügt.
- Mit `$$ Formel $$` wird eine Formel als eigenständiger Absatz zentriert eingefügt.

Sollen auch Zeilenumbrüche innerhalb einer Formel erlaubt sein, so sollte die Umgebung `align` (beziehungsweise `align*` für nummerierte beziehungsweise nicht nummerierte Formeln) gewählt werden:

```
% Standard-Mathe-Umgebung:  
\begin{align}  
  E \underset{Einstein}{=} m \cdot c^2  
  \underset{Pythagoras}{=} m \cdot (a^2 + b^2)  
\end{align}
```

Ergebnis:

$$E \underset{Einstein}{=} m \cdot c^2 \underset{Pythagoras}{=} m \cdot (a^2 + b^2) \tag{1}$$

Der Gleichungs-Zähler kann bei Bedarf mittels `\setcounter{equation}{1}` wieder auf den Wert 1 (oder einen beliebigen anderen Wert) gesetzt werden. Mit `\numberwithin{equation}{section}` kann zudem festgelegt werden, welcher Gliederungstiefe die Formeln zugerechnet werden sollen, ob also eine Nummerierung in der Art (1), (2), ... oder abschnittsweise als (1.1), (1.2), ... erfolgen soll.

Formeln mit mehreren Nummerierungen

Die Umgebungen `align` und `align*` sind für fast alle abgesetzten Formeln die beste Wahl. Eine Ausnahme bilden mehrere einzeilige Formeln, die zueinander ausgerichtet, aber einzeln nummeriert werden sollen. Für diesen Zweck gibt es die Umgebung `eqnarray`:

```
% Umgebung für mehrere nummerierte Zeilen:  
\begin{eqnarray}  
  x^2 &=& 2 \\  
  \to x &=& \pm\sqrt{2}  
\end{eqnarray}
```

Ergebnis:

$$x^2 = 2 \tag{2}$$

$$\rightarrow x = \pm\sqrt{2} \tag{3}$$

¹ Anstelle mit `$... $` können für Inline-Formeln auch mit `\(... \)` begrenzt werden. Dies wird beispielsweise vom Dokumentationssystem Sphinx für automatisch erzeugten LaTeX-Code genutzt.

Anstelle mit `$$... $$` können eigenständige Formeln auch mit `\[... \]` begrenzt werden.

Sollen bei Verwendung einer `eqnarray`-Umgebung einzelne Zeilen *nicht* nummeriert werden, so kann man am Ende der jeweiligen Zeile die Anweisung `\notag` einfügen, um eine Nummerierung zu verhindern.

Sollen die einzelnen Zeilen eines Formel-Absatzes zwar eigenständig nummeriert werden und eigene Labels erhalten, aber letztlich nur als Teile einer ganzen Formel angesehen werden, so ist dies mittels der `subequations`-Umgebung möglich:

Beispiel:

```
\begin{subequations}
\label{maxwell-gleichungen}
\begin{align}
\text{div }(\vec{D}) &= 4 \cdot \pi \cdot \rho && \text{\label{coulomb-gesetz}} \\
\text{rot }(\vec{H}) &= \frac{4 \cdot \pi}{c} \cdot \vec{j} && \text{\label{ampere-gesetz}} \\
\text{rot }(\vec{E}) &= - \frac{1}{c} \cdot \frac{\partial \vec{B}}{\partial t} && \text{\label{faraday-gesetz-1}} \\
\text{div }(\vec{B}) &= 0 && \text{\label{faraday-gesetz-2}}
\end{align}
\end{subequations}
```

Ergebnis:

$$\text{div }(\vec{D}) = 4 \cdot \pi \cdot \rho \tag{4a}$$

$$\text{rot }(\vec{H}) = \frac{4 \cdot \pi}{c} \cdot \vec{j} \tag{4b}$$

$$\text{rot }(\vec{E}) = -\frac{1}{c} \cdot \frac{\partial \vec{B}}{\partial t} \tag{4c}$$

$$\text{div }(\vec{B}) = 0 \tag{4d}$$

Auf einzelne Gleichungen kann dann via `\eqref{}` wahlweise auf ein Label der Teilgleichung oder auch auf das Label der gesamten Gleichung verwiesen werden.

Besonderheiten im Mathematik-Modus

Der Mathematik-Modus weist gegenüber normalem Text einige Besonderheiten auf:

- Mathematische Formeln dürfen keine leeren Zeilen beinhalten: Jede Formel entspricht einem einzigen Absatz.
- Leerzeichen innerhalb von Formeln werden ignoriert und müssen bei Bedarf manuell mittels Abstands-Anweisungen wie `\,` oder `\;` gesetzt werden. Vor und nach mathematischen Operatoren wie `+` oder `-` wird von LaTeX automatisch etwas Freiraum eingefügt.
- Buchstaben werden in Formeln grundsätzlich als Namen von Variablen interpretiert und daher kursiv gedruckt. Sollen einzelne Buchstaben aufrecht gedruckt werden, so ist dies mittels `\mathrm{}` möglich; normale Textabschnitte (inklusive Leerzeichen) können innerhalb von Formeln mittels `\text{Text}` eingebettet werden.

Mathematische Symbole

Griechisches Alphabet

In Formeln werden sehr häufig griechische Buchstaben als Variablenbezeichnungen verwendet. In der folgenden Liste sind die griechischen Buchstaben sowie die zugehörigen LaTeX-Anweisungen für den Mathe-Modus aufgelistet.

Tab. 1: Griechisches Alphabet

Aussprache	Buchstabe	LaTeX-Code	Aussprache	Buchstabe	LaTeX-Code
Alpha	$A \alpha$	<code>A \alpha</code>	Ny	$N \nu$	<code>N \nu</code>
Beta	$B \beta$	<code>B \beta</code>	Xi	$\Xi \xi$	<code>\Xi \xi</code>
Gamma	$\Gamma \gamma$	<code>\Gamma \gamma</code>	Omikron	$O o$	<code>O o</code>
Delta	$\Delta \delta$	<code>\Delta \delta</code>	Pi	$\Pi \pi$	<code>\Pi \pi</code>
Epsilon	$E \varepsilon$	<code>E \varepsilon</code>	Rho	$P \rho$	<code>P \rho</code>
Zeta	$Z \zeta$	<code>Z \zeta</code>	Sigma	$\Sigma \sigma$	<code>\Sigma \sigma</code>
Eta	$H \eta$	<code>H \eta</code>	Tau	$T \tau$	<code>T \tau</code>
Theta	$\Theta \vartheta$	<code>\varTheta \vartheta</code>	Ypsilon	$\Upsilon \upsilon$	<code>\Upsilon \upsilon</code>
Iota	$I \iota$	<code>I \iota</code>	Phi	$\Phi \varphi$	<code>\varPhi \varphi</code>
Kappa	$K \kappa$	<code>K \kappa</code>	Chi	$X \chi$	<code>X \chi</code>
Lambda	$\Lambda \lambda$	<code>\Lambda \lambda</code>	Psi	$\Psi \psi$	<code>\Psi \psi</code>
My	$M \mu$	<code>M \mu</code>	Omega	$\Omega \omega$	<code>\Omega \omega</code>

Bei manchen griechischen Buchstaben existiert neben den oben angegebenen Varianten auch noch alternative Darstellungen. Beispielsweise wird `\theta` als θ ausgegeben, während `\vartheta` als ϑ gedruckt wird.

Mathematische Schriftarten

Ebenso wie Texte in normalen Textabsätzen mittels `\textbf{}`, `\textit{}` usw. hervorgehoben werden können, existieren im Mathe-Modus verschiedene Möglichkeiten, die Form oder den Typ einer Schrift zu verändern:

- Ohne explizite Angabe wird `\mathnormal{}` als Schrifttyp verwendet. Hierbei werden Buchstaben kursiv dargestellt, Zahlen hingegen aufrecht:

ABCDEF abcdef 123456

- Mit `\mathrm{}` („Math Roman“) werden sowohl Buchstaben als auch Zahlen im Mathe-Modus aufrecht gedruckt. Dieser Schrifttyp wird beispielsweise geometrische Punkte, für Einheiten oder Symbole in Indizes verwendet.

ABCDEF abcdef 123456

- Mit `\mathit{}` („Math Italic“) werden sowohl Buchstaben als auch Zahlen im Mathe-Modus kursiv gedruckt:

ABCDEF abcdef 123456

- Mit `\mathbf{}` („Math Bold Font“) werden Buchstaben und Zahlen im Mathe-Modus aufrecht und fettgedruckt ausgegeben. In manchen Lehrbüchern werden auf diese Weise Vektoren gekennzeichnet.

ABCDEF abcdef 123456

- Mit `\mathsf{}` („Math Sans Serif“) werden Buchstaben und Zahlen im Mathe-Modus aufrecht und ohne Serifen ausgegeben:

ABCDEF abcdef 123456

- Mit `\mathtt{}` („Math Typesetter“) werden Buchstaben und Zahlen im Mathe-Modus aufrecht und nicht-proportional ausgegeben:

ABCDEF abcdef 123456

- Mit `\mathfrak{}` („Math Fraktur“) werden Buchstaben und Zahlen im Mathe-Modus als Frakturschrift ausgegeben:

℔BCDE℔ abcdef 123456

- Mit `\mathbb{}` („Math Blackboard Bold“) werden Großbuchstaben im Mathe-Modus als Mengensymbole ausgegeben. Hierzu muss in der Präambel das Paket `amsmath` mittels `\usepackage{amsmath}` geladen werden.

ABCDEF

- Mit `\mathcal{}` („Math Calligraphy“) werden Großbuchstaben im Mathe-Modus kalligraphisch ausgegeben:

ABCDEF

- Mit `\mathscr{}` („Math Script“) werden Großbuchstaben im Mathe-Modus in einer weiteren Darstellungsvariante ausgegeben. Hierzu muss in der Präambel das Paket `mathrsfs` mittels `\usepackage{mathrsfs}` geladen werden.

A B C D E F

Relationszeichen

Die Relationszeichen =, < und > können direkt mittels der Tastatur eingegeben werden. Weitere Relationszeichen sind in der folgenden Tabelle aufgelistet.

Eingabe	Ausgabe	Eingabe	Ausgabe
=	=	<code>\neq</code>	≠
<code>\stackrel{\wedge}{=}</code>	≐	<code>\stackrel{!}{=}</code>	≑
<code>\equiv</code>	≡	<code>\cong</code>	≅
<code>\geq</code>	≥	<code>\leq</code>	≤
<code>\gg</code>	≫	<code>\ll</code>	≪
<code>\approx</code>	≈	<code>\sim</code>	~
<code>\propto</code>	∝	<code>\simeq</code>	≈
<code>\in</code>	∈	<code>\notin</code>	∉
<code>\subset</code>	⊂	<code>\supset</code>	⊃
<code>\subseteq</code>	⊆	<code>\supseteq</code>	⊇
<code>\cup</code>	∪	<code>\cap</code>	∩
<code>\perp</code>	⊥	<code>\parallel</code>	∥

Allgemein können die obigen Relationssymbole, wie am Beispiel \notin (`\notin`) gezeigt, durch ein Voranstellen von `\not` invertiert werden; beispielsweise ergibt eine Eingabe von `\not\geq` das Zeichen \nlessgtr .

Weitere mathematische Symbole

Eingabe	Ausgabe	Eingabe	Ausgabe
<code>\pm</code>	\pm	<code>\mp</code>	\mp
<code>\div</code>	\div	<code>\setminus</code>	\setminus
<code>\cdot</code>	\cdot	<code>\times</code>	\times
<code>\ast</code>	$*$	<code>\star</code>	\star
<code>\circ</code>	\circ	<code>\bullet</code>	\bullet
<code>\varangle</code>	\varangle	<code>\angle</code>	\angle
<code>\vee</code>	\vee	<code>\wedge</code>	\wedge
<code>\forall</code>	\forall	<code>\exists</code>	\exists
<code>\infty</code>	∞	<code>\emptyset</code>	\emptyset
<code>\partial</code>	∂	<code>\nabla</code>	∇
<code>\oplus</code>	\oplus	<code>\ominus</code>	\ominus
<code>\odot</code>	\odot	<code>\oslash</code>	\oslash
<code>\Box</code>	\square	<code>\checkmark</code>	\checkmark
<code>\clubsuit</code>	\clubsuit	<code>\spadesuit</code>	\spadesuit
<code>\heartsuit</code>	\heartsuit	<code>\diamondsuit</code>	\diamondsuit
<code>\%</code>	$\%$	<code>\permil</code>	\permil

Die Anweisung `\permil` setzt voraus, dass in der Präambel das Paket `wasysym` geladen wird (`\usepackage{wasysym}`).

Mathematische Ausdrücke

Indizes und Exponenten

Soll im Mathe-Modus eine einzelne Ziffer oder ein einzelner Buchstabe als Index einer Variablen gesetzt werden, so ist dies mittels `x_1`, `x_2`, ..., `x_n` möglich; die Ausgabe würde in diesem Fall x_1, x_2, \dots, x_n lauten. Soll der Index aus mehr als einem Zeichen bestehen, so müssen diese in geschweifte Klammern gesetzt werden, also beispielsweise `x_{1,2}` für $x_{1,2}$. Lässt man die geschweiften Klammern weg, so springt LaTeX nach dem ersten Index-Zeichen wieder in den normalen Mathe-Modus zurück und druckt die restlichen Zeichen als normal große Variablenamen.

Um im Mathe-Modus eine einzelne Ziffer oder einen einzelnen Buchstaben als Exponent einer Variablen zu setzen, so ist dies mittels `x^1`, `x^2`, ..., `x^n` möglich; die Ausgabe würde in diesem Fall x^1, x^2, \dots, x^n lauten. Auch bei Exponenten müssen geschweifte Klammern gesetzt werden, sofern diese aus mehr als einem Zeichen bestehen.

In Exponenten werden Buchstaben in LaTeX standardmäßig aufrecht gedruckt, in Indizes hingegen kursiv. Möchte man, wie es in der Textsatzung üblich ist, aufrechte Indizes erhalten, so müssen diese in geschweifte Klammern gesetzt und mittels `\mathrm{}` explizit in aufrechter Form ausgegeben werden (Zahlen werden auch in LaTeX stets aufrecht gedruckt).

Einheiten

In Mathe-Umgebungen können Einheiten – ebenso wie in normalen Text-Bereichen – am einfachsten mittels des `units`-Pakets gesetzt werden. Die Syntax dafür ist denkbar einfach:

Beispiel:

```
% Größe mit Einheit setzen:
\unit[1]{Liter} = \unit[1]{dm^3}
```

Ergebnis:

$$1 \text{ Liter} = 1 \text{ dm}^3$$

Die `units`-Anweisung hat einerseits den Effekt, dass die in den geschweiften Klammern angegebene Einheit nicht wie im Mathe-Modus üblich kursiv, sondern aufrecht gedruckt wird; andererseits wird der Abstand zwischen dem Zahlenwert und der Einheit etwas reduziert. Die `units{}`-Anweisung kann auch ohne Angabe eines Zahlenwerts genutzt werden, beispielsweise um reine Einheits-Gleichungen zu setzen. Innerhalb der geschweiften Klammern sind auch weitere LaTeX-Anweisungen, wie beispielsweise *Brüche* oder *Wurzeln*, erlaubt.

Klammern

Runde und eckige Klammern können in LaTeX-Formeln als gewöhnliche Zeichen gesetzt werden, bei geschweiften Klammern muss ein Backslash-Zeichen vor die öffnende und schließende Klammer gesetzt werden.

Beispiel:

```
% Verschiedene Klammern in LaTeX:  
(a) \quad [b] \quad \{c\} \quad |d| \quad \langle e \rangle
```

Ergebnis:

$$(a) \quad [b] \quad \{c\} \quad |d| \quad \langle e \rangle$$

Möchte man die Größe einer Klammer anpassen, wenn beispielsweise ein Bruch innerhalb der Klammer vorkommt, so kann die Klammergröße automatisch oder manuell festgelegt werden:

- Mit `\left` und `\right` wird die Größe der darauf folgenden öffnenden beziehungsweise schließenden Klammer automatisch an den Inhalt der Klammer angepasst. Für runde Klammern mit automatische Größenanpassung kann man also `\left(` beziehungsweise `\right)` schreiben:

Beispiel:

```
(a^{b^{c^d}}) \neq \left( d^{c^{b^a}} \right)
```

Ergebnis:

$$(a^{b^{c^d}}) \neq \left(d^{c^{b^a}} \right)$$

Das gleiche funktioniert auch mit eckigen und geschweiften Klammern, wobei bei letzteren `\left\{` beziehungsweise `\right\}` geschrieben werden müssen.

- Mit `\big(... \big)`, `\Big(... \Big)`, `\bigg(... \bigg)` oder `\Bigg(... \Bigg)` kann die Größe der öffnenden beziehungsweise schließenden runden Klammern manuell festgelegt werden. Das gleiche funktioniert ebenfalls mit vertikalen Strichen (beispielsweise Betragstrichen), die direkt mittels des Pipe-Zeichens `|` eingegeben werden können.

Bisweilen sind auch „liegende“ geschweifte Klammern nützlich, um beispielsweise eine Erklärung für einen auftretenden Term in die Gleichung mit aufzunehmen. Bindet man in der Präambel das Paket `mathtools` mit ein, so kann man dafür Anweisung die Anweisungen `\underbrace{}` beziehungsweise `\overbrace{}` nutzen:

Beispiel:

```
\overbrace{n \cdot n \cdot n \cdot \ldots \cdot n}^{\text{\$k\$ mal}} = n^k \\ \underbrace{n \cdot n \cdot n \cdot \ldots \cdot n}_{\text{\$k\$ mal}} = n^k
```

Ergebnis:

$$\overbrace{n \cdot n \cdot n \cdot \dots \cdot n}^{k \text{ mal}} = n^k$$

$$\underbrace{n \cdot n \cdot n \cdot \dots \cdot n}_{k \text{ mal}} = n^k$$

Pfeile

Pfeile können entweder über oder zwischen mathematischen Symbolen stehen. Pfeile über mathematischen Symbolen markieren Vektoren oder gerichtete Strecken. Im Fall von Vektoren, wenn sich der Pfeil über ein einzelnes Zeichen erstreckt, kann man die Anweisung `\vec{}` verwenden, für Pfeile über mehreren mathematischen Symbolen muss hingegen `\overrightarrow{}` (oder `\overline{}` für Strecken ohne Richtung) verwendet werden:

Beispiel:

```
\begin{align*}
  \vec{a}
  \overrightarrow{\mathrm{BC}}
  \overline{\mathrm{DE}}
\end{align*}
```

Ergebnis:

$$\vec{a} \quad \overrightarrow{BC} \quad \overline{DE}$$

Für Pfeile zwischen mathematischen Symbolen gibt es mehrere Anweisungen. Ein einfacher waagrechtlicher Pfeil, wie er beispielsweise geschrieben wird, wenn eine Zahl gegen einen bestimmten Grenzwert geht, kann einfach mit `\to` gesetzt werden. Mehr Flexibilität bieten die Anweisungen der Art `\leftarrow` und `\rightarrow`:

LaTeX-Code	Ergebnis	Beschreibung
<code>\leftarrow</code> und <code>\rightarrow</code>	\leftarrow und \rightarrow	waagrechtlicher Pfeil
<code>\Leftrightarrow</code> und <code>\Rrightarrow</code>	\Leftrightarrow und \Rrightarrow	waagrechtlicher Pfeil mit Doppelstrich
<code>\longleftarrow</code> und <code>\longrightarrow</code>	\longleftarrow und \longrightarrow	langer waagrechtlicher Pfeil
<code>\Llongleftarrow</code> und <code>\Llongrightarrow</code>	\Llongleftarrow und \Llongrightarrow	langer waagrechtlicher Pfeil mit Doppelstrich
<code>\leftrightarrow</code> und <code>\Leftrightarrow</code>	\leftrightarrow und \Leftrightarrow	beidseitiger Pfeil
<code>\longlefttrightarrow</code> und <code>\Llonglefttrightarrow</code>	\longlefttrightarrow und \Llonglefttrightarrow	langer beidseitiger Pfeil

Mittels `\uparrow` und `\downarrow` lassen sich entsprechend nach oben (\uparrow) beziehungsweise nach unten (\downarrow) zeigende Pfeile setzen; auch diese können wie `\leftarrow` und `\rightarrow` gemäß der obigen Tabelle modifiziert werden. Für diagonale Pfeile gibt es die Anweisungen `\nwarrow` und `\nearrow` (\nwarrow und \nearrow) beziehungsweise `\swarrow` und `\searrow` (\swarrow und \searrow), wobei die ersten beiden Buchstaben jeweils die Himmelsrichtung angeben, in die der Pfeil zeigen soll.

Mittels `\circlearrowleft` und `\circlearrowright` lassen sich die zur Kennzeichnung von Drehmomenten verwendeten kreisförmige Pfeile (\circlearrowleft und \circlearrowright) setzen, mittels `\curvearrowleft` beziehungsweise `\curvearrowright` gebogene Pfeile (\curvearrowleft und \curvearrowright).

Beschriftete waagrechtliche Pfeile können zudem mittels `\xleftarrow{}` beziehungsweise `\xrightarrow{}` erstellt werden. Die Länge eines solchen Pfeils wird von LaTeX automatisch anhand der Länge des über beziehungsweise unter dem Pfeil stehenden Textes bestimmt.

Beispiel:

```
\mathrm{A} \quad \xleftarrow[\phantom{\text{was sonst?}}]{-1} \quad \mathrm{B} \quad \xrightarrow[\text{was sonst?}]{+1} \quad \mathrm{C}
```

Ergebnis:

$$A \quad \xleftarrow{-1} \quad B \quad \xrightarrow[\text{was sonst?}]{+1} \quad C$$

Eine vollständige Übersicht über die verschiedenen Pfeile gibt es beispielsweise [hier](#).

Wurzeln, Brüche und Binomialkoeffizienten

Wurzeln werden in LaTeX-Formeln mittels `\sqrt{}` gesetzt. Möchte man keine Quadrat-Wurzel ausgeben, sondern eine Wurzel mit einem anderen Wurzelexponenten, so kann für die n -te Wurzel aus einer Zahl `\sqrt[n]{Zahl}` geschrieben werden.

Beispiel:

```
% "Normale" Quadrat-Wurzel :
\sqrt{16} = 4

% Kubische Wurzel :
\sqrt[3]{8} = 2
```

Ergebnis:

$$\sqrt{16} = 4$$

$$\sqrt[3]{8} = 2$$

Innerhalb der `\sqrt{}`-Anweisung können auch andere Mathe-Anweisungen wie Pfeile oder Brüche gesetzt werden; Diese werden in LaTeX mittels `\frac{Zähler}{Nenner}` gesetzt:

Beispiel:

```
\begin{align*}
\frac{a}{b} : \frac{c}{d} &= \frac{a}{b} \cdot \frac{d}{c}
\end{align*}
```

Ergebnis:

$$\frac{a}{b} : \frac{c}{d} = \frac{a}{b} \cdot \frac{d}{c}$$

Innerhalb des Zählers und/oder Nenners können wiederum `\frac{}`-Anweisungen auftreten, so dass sich damit auch verschachtelte Brüche bilden lassen. Die Größe der Schriften sowie die Dicke der Bruchstriche werden dabei automatisch angepasst:

Beispiel:

```
\begin{align*}
\frac{\frac{a}{b}}{\frac{c}{d}} &= \frac{a \cdot d}{b \cdot c}
\end{align*}
```

Ergebnis:

$$\frac{\frac{a}{b}}{\frac{c}{d}} = \frac{a \cdot d}{b \cdot c}$$

Die automatische Anpassung der Schriftgröße bei der `\frac{}`-Anweisung kann umgangen werden, wenn man mit ansonsten gleicher Syntax die `\tfrac{}` oder `\dfrac{}`-Anweisung verwendet:

- Bei der `\tfrac{}`-Anweisung wird automatisch die Option `\textstyle` aktiviert; der Bruch wird dadurch möglichst so dimensioniert, dass er in eine normale Textzeile passt.
- Bei der `\dfrac{}`-Anweisung wird automatisch die Option `\displaystyle` aktiviert; eine automatische Verkleinerung der Schriftgröße findet dabei auch bei verschachtelten Brüchen nicht statt.

In ähnlicher Weise wie Brüche lassen sich auch **Binomial-Koeffizienten** setzen. Auch wenn diese aus mathematischer Sicht eine andere Bedeutung haben, sind sie aus Sicht der Textsatzung den Brüchen ähnlich, nur dass der Bruchstrich zwischen der oberen und der unteren Zahl fehlt.

Beispiel:

```
% Binomialkoeffizient:
\begin{align*}
\binom{n}{k} = \frac{n!}{(n - k)! \cdot k!}
\end{align*}
```

Ergebnis:

$$\binom{n}{k} = \frac{n!}{(n - k)! \cdot k!}$$

Summen, Produkte und Integrale

Das Summenzeichen \sum kann im Mathe-Modus mittels `\sum` gedruckt werden. Üblicherweise wird dabei unterhalb des Summenzeichens die Untergrenze und oberhalb die Obergrenze der zu summierenden Größe angegeben. In LaTeX wird dazu die für Indizes und Exponenten übliche Syntax genutzt und somit `\sum_{~}^{~}` geschrieben:

Beispiel:

```
\begin{align*}
\sum_{i=1}^n i = \frac{n \cdot (n+1)}{2}
\end{align*}
```

Ergebnis:

$$\sum_{i=1}^n i = \frac{n \cdot (n + 1)}{2}$$

In gleicher Weise kann das (seltener vorkommende) Produkt-Zeichen \prod genutzt werden:

Beispiel:

```
\begin{align*}
\sum_{i=1}^n i = \frac{n \cdot (n+1)}{2}
\end{align*}
```

Ergebnis:

$$\prod_{i=1}^n i = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 1 = n!$$

Das Integralzeichen \int kann im Mathe-Modus mittels `\int` gedruckt werden. Üblicherweise wird auch hierbei unterhalb des Integralzeichens die Untergrenze und oberhalb die Obergrenze der zu integrierenden Größe angegeben. In LaTeX wird dazu wiederum die für Indizes und Exponenten übliche Syntax genutzt und somit `\int_{~}^{~}` geschrieben. Ohne weitere Voreinstellungen werden die Integralgrenzen etwas versetzt gedruckt; möchte man dies unterbinden, so kann man dies durch ein Einfügen von `\limits` vor den Integralgrenzen erreichen:

Beispiel:

```

\begin{align*}
\int_a^b f(x) \cdot dx &= F(b) - F(a) \\
\int\limits_a^b f(x) \cdot dx &= F(b) - F(a)
\end{align*}

```

Ergebnis:

$$\int_a^b f(x) \cdot dx = F(b) - F(a)$$

$$\int\limits_a^b f(x) \cdot dx = F(b) - F(a)$$

Mehrfachintegrale (Flächen- und Volumenintegrale) können entsprechend mittels `\iint`, `\iiint` gesetzt werden, geschlossene Linien-Integrale mittels `\oint`.

Beispiel:

```

% Gaußscher Integralsatz:
\iiint_V \text{div } \vec{E} (\vec{r}) \cdot d^3r =
\oint_{S(V)} \vec{E} \cdot d\vec{f}

```

Ergebnis:

$$\iiint_V \operatorname{div} \vec{E}(\vec{r}) \cdot d^3r = \oint_{S(V)} \vec{E} \cdot d\vec{f}$$

Matrizen und Determinanten

Zum Setzen von Matrizen beziehungsweise Determinanten stellt das Paket `amsmath` für den Mathe-Modus mehrere Umgebungen bereit:

<code>matrix</code>	Matrizen ohne Klammern
<code>pmatrix</code>	Matrizen mit runden Klammern
<code>bmatrix</code>	Matrizen mit eckigen Klammern
<code>vmatrix</code>	Matrizen mit eckigen Betragstrichen

Innerhalb einer Matrix-Umgebung werden die einzelnen Spalten, ähnlich wie bei *Tabellen*, durch `&`-Zeichen getrennt; Zeilenumbrüche lassen sich wie gewöhnlich mittels `\\` erreichen.

Beispiel:

```

\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{pmatrix}

```

Ergebnis:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Die `vmatrix`-Umgebung kann insbesondere zur Erzeugung von *Determinanten* genutzt werden; mittels der `pmatrix`-Umgebung hingegen können auch *Vektoren* als einspaltige Matrizen gesetzt werden.

Fallunterscheidungen

Fallunterscheidungen können im Mathe-Modus mittels der `cases`-Umgebung verwirklicht werden. Innerhalb einer solchen Umgebung können die einzelnen Elemente mittels eines `&`-Zeichens ausgerichtet werden; die vertikale Ausrichtung übernimmt LaTeX (bis auf manuell zu setzende Zeilenumbrüche) automatisch.

Beispiel:

```
| a | = \begin{cases}
+a & \& \text{ falls } a > 0 \\
\backslash; \backslash; 0 & \& \text{ falls } a = 0 \\
-a & \& \text{ falls } a < 0
\end{cases}
```

Ergebnis:

$$|a| = \begin{cases} +a & \text{falls } a > 0 \\ 0 & \text{falls } a = 0 \\ -a & \text{falls } a < 0 \end{cases}$$

Die array-Umgebung

Die `array`-Umgebung kann innerhalb von Formeln verwendet werden, um eine horizontale Ausrichtung der einzelnen Formel-Elemente zu erreichen. Die Syntax ist dabei der `tabular`-Umgebung sehr ähnlich:

Beispiel:

```
\begin{align*}
\begin{array}{lcr}
\text{Erste Zahl} & \& x & \& 8 \\
\text{Zweite Zahl} & \& y & \& 15 \\
\text{Summe} & \& x + y & \& 23 \\
\text{Differenz} & \& x - y & \& -7 \\
\text{Produkt} & \& x \cdot y & \& 120
\end{array}
\end{align*}
```

Ergebnis:

Erste Zahl	x	8
Zweite Zahl	y	15
Summe	$x + y$	23
Differenz	$x - y$	-7
Produkt	$x \cdot y$	120

Direkt hinter `\begin{align}` werden in geschweiften Klammern die gewünschten Spalten-Optionen angegeben, wobei `l` für eine Links-Ausrichtung der Spalte, `c` für eine Zentrierung und `r` für eine Rechts-Ausrichtung steht. Die Anzahl an Spaltenoptionen muss mit der tatsächlichen Spaltenanzahl innerhalb der Tabelle übereinstimmen. In die einzelnen Felder der `array`-Umgebung können wiederum beliebige Formel-Elemente gesetzt werden (sogar Matrizen).

Zwischen den einzelnen Spalten-Optionen können nach Belieben `|`-Zeichen eingefügt werden, um (wie bei einer Tabelle) vertikale Striche zwischen den jeweiligen Spalten (oder am Rand der Tabelle) einzuzichnen; horizontale Linien lassen sich jeweils zu Beginn einer neuen Zeile mittels der `\hline`-Anweisung verwirklichen.

Wichtige Umgebungen

So genannte Umgebungen werden in LaTeX verwendet, um einzelne Absätze zu gestalten. Üblicherweise werden Umgebungen mit `\begin{umgebung}` eingeleitet und mit `\end{umgebung}` abgeschlossen.

Die im folgenden Abschnitt aufgelisteten Umgebungen sind standardmäßig in LaTeX definiert; weitere Umgebungen können durch das Einbinden von *Zusatzpaketen* genutzt werden.

center, flushleft, flushright – Textausrichtung

Normalerweise wird Text in Latex als Blocksatz dargestellt. LaTeX optimiert dabei den Abstand zwischen einzelnen Wörtern und nimmt automatisch eine Silbentrennung vor. Mittels der folgenden Umgebungen kann die Ausrichtung eines Absatzes manuell geändert werden:

- *Zentrierung*: Bereiche eines LaTeX-Dokuments, die zwischen `\begin{center}` und `\end{center}` stehen, werden mittig auf der Seite ausgegeben.
- *Flattersatz*: Bereiche eines LaTeX-Dokuments, die innerhalb einer `\begin{flushleft}` und `\end{flushleft}` stehen, werden linksbündig als Flattersatz ausgegeben. Entsprechend werden Bereiche, die zwischen `\begin{flushright}` und `\end{flushright}` stehen, rechtsbündig als Flattersatz ausgegeben.

Die Flattersatz-Umgebungen werden üblicherweise für Text oder mathematische Formeln verwendet; Zentrierungen sind hingegen auch für Abbildungen, Tabellen oder dergleichen üblich.

itemize, enumerate und description – Aufzählungen

In LaTeX gibt es folgende Umgebungen für Aufzählungen:

- Mit einer `itemize`-Umgebung kann eine nicht nummerierte Aufzählung erstellt werden. Die einzelnen aufzuzählenden Punkte werden mittels der Anweisung `\item` gelistet:

```
\begin{itemize}
  \item ...
  \item ...
  \item ...
\end{itemize}
```

Innerhalb der einzelnen Einträge werden Leerzeilen berücksichtigt, zwischen den einzelnen Einträgen werden Leerzeilen ignoriert.

- Mit einer `enumerate`-Umgebung kann eine nummerierte Aufzählung erstellt werden. Die einzelnen aufzuzählenden Punkte werden ebenfalls mittels der Anweisung `\item` gelistet:

```

\begin{enumerate}

  \item ...
  \item ...
  \item ...

\end{enumerate}

```

- Mit einer `description`-Umgebung kann eine Stichwort-Aufzählung erstellt werden. Die einzelnen aufzuzählenden Punkte werden mittels der Anweisung `\item[Stichwort]` gelistet:

```

\begin{description}

  \item[Stichwort 1:] ...
  \item[Stichwort 2:] ...
  \item[Stichwort 3:] ...

\end{description}

```

Aufzählungs-Umgebungen können in LaTeX auch (maximal vierfach) geschachtelt auftreten, beispielsweise kann eine `itemize`-Umgebung innerhalb einer `enumerate`-Umgebung stehen:

```

\begin{enumerate}

  \item ...

  \item

  \begin{itemize}

    \item ...
    \item ...

  \end{itemize}

  \item ...

\end{enumerate}

```

LaTeX passt die Einrückungstiefen der Aufzählungen sowie die Art der Anführungszeichen, soweit nicht manuell vorgegeben, automatisch an. Einrückungen innerhalb des Quellcodes haben keine Auswirkung auf die PDF-Datei, sind aber für eine bessere Lesbarkeit empfehlenswert.

Abstände zwischen Einträgen anpassen

LaTeX lässt zwischen den einzelnen Einträgen einer Aufzählung verhältnismäßig viel Abstand, um die Lesbarkeit zu erhöhen. Bisweilen ist es allerdings aus Platzgründen hilfreich, eine Aufzählung möglichst kompakt zu halten. In diesem Fall sind vor allem zwei Anweisungen nützlich:

- Mit `\vspace{-2mm}` kann der Abstand zwischen dem vorherigen Absatz und dem Beginn der Aufzählung reduziert werden.
- Mit `\setlength\itemsep{0mm}` wird der *zusätzliche* Abstand zwischen den einzelnen Einträgen auf Null reduziert; die einzelnen Einträge haben somit nur einen normalen Zeilenabstand.

Als Alternative zu dieser manuellen Anpassung kann auch das Zusatz-Paket *paralist* verwendet werden, das kompakte Versionen der normalen Aufzählungs-Umgebungen bereitstellt.

figure – Abbildungen

Die `figure`-Umgebung ist zum Einbinden von Abbildungen vorgesehen; hierzu muss in der Präambel des Dokuments das Paket `graphicx` mittels `\usepackage{graphicx}` geladen werden.

Bei der `figure`-Umgebung handelt es sich um eine so genannte „Fließumgebung“ (bisweilen auch „Gleitobjekt“ genannt), bei der LaTeX selbst anhand von angegebenen Optionen entscheidet, an welcher Stelle die Abbildung am besten eingebaut wird:

```
% In der Präambel:

\usepackage{graphicx}

% Innerhalb des Dokuments:

\begin{figure}[htb]
  \centering
  \includegraphics[width=0.8\linewidth]{image-filename.png}
  \caption{Hier kommt die Bildunterschrift hin.}
  \label{fig-image-label}
\end{figure}
```

In diesem Beispiel sind als Optionen für mögliche Positionierungen der Abbildung `htb` angegeben: `h` („here“) steht für die aktuelle Position, `t` („top“) für den Beginn der aktuellen Seite, `b` („bottom“) für das Ende der aktuellen Seite; eine weitere mögliche Option ist `p` („page“), bei der die Abbildung optional auch auf einer separaten Seite gedruckt werden darf. Mittels eines Ausrufezeichens kann eine Position erzwungen werden, beispielsweise bewirkt eine Optionsangabe von `[h!]`, dass die Abbildung unbedingt an der aktuellen Stelle eingebunden werden soll.

Das eigentliche Einfügen der Graphik wird durch die Anweisung `\includegraphics{}` übernommen. Als Optionen kann hierbei mittels `width` oder `height` die Größe der Abbildung im Dokument festgelegt werden, mittels `angle=90` kann die Abbildung zudem bei Bedarf um den angegebenen Winkel (gegen den Uhrzeigersinn) gedreht werden. Als Bildformate können bei Verwendung von `pdflatex` wahlweise `png`, `jpg` oder `bmp` verwendet werden.

Beispiel:

```
\begin{figure}[h!]
  \centering
  \includegraphics[width=0.5\textwidth]{pics/tux.png}
  \caption{Das Linux-Maskottchen `Tux'.}
  \label{fig-tux}
\end{figure}
```

Ergebnis:

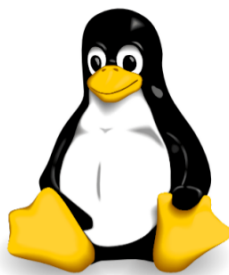


Abbildung 1: Das Linux-Maskottchen ‘Tux’.

Befinden sich die Bilddateien nicht im gleichen Verzeichnis wie die `.tex`-Datei, so kann bei der `\includegraphics{}`-Anweisung auch ein relativer oder absoluter Pfad angegeben werden. Hierbei muss allerdings geachtet werden, dass in dem Pfadnamen keine Leerzeichen oder Unterstriche vorkommen; ist

dies der Fall, so muss vor diese „Sonderzeichen“ je ein Backslash-Zeichen `\` geschrieben werden. Empfehlenswert ist es daher, alle zu einem LaTeX-Dokument gehörenden Bilddateien in einem Unterordner `pics` abzulegen und darauf zu achten, dass in den Dateinamen Leerzeichen und Unterstriche beispielsweise durch Minus-Zeichen ersetzt sind.

Die `\includegraphics{}`-Anweisung kann auch ohne eine umschließende `figure`-Umgebung verwendet werden; in diesem Fall wird die Abbildung genau an der Stelle im Dokument eingebunden, an der die `\includegraphics{}`-Anweisung steht. In diesem Fall ist es allerdings nicht möglich, die Abbildung mit einer Bildunterschrift („Caption“) zu versehen. Umgekehrt können allerdings innerhalb einer `figure`-Umgebung auch mehrere `\includegraphics{}`-Anweisungen vorkommen, wenn beispielsweise mehrere Bilder nebeneinander oder untereinander abgebildet werden sollen. Im letzteren Fall muss zwischen den einzelnen `\includegraphics{}`-Anweisungen eine Neue-Zeile-Anweisung `\\` stehen, zudem können die Anweisungen `\hspace{}` und `\vspace{}` für die Ausrichtungen der Abbildungen nützlich sein.

Bildunterschriften ohne automatische Nummerierung

Wird innerhalb einer `figure`-Umgebung mittels der `\caption{}`-Anweisung eine Bildunterschrift gesetzt, so wird diese automatisch mit einer Nummerierung versehen, beispielsweise „Abbildung 1: Bildunterschrift.“

Möchte man die automatische Nummerierung verhindern, so kann in der Präambel das Zusatz-Paket `caption` eingebunden und anstelle der Anweisung `\caption{}` die Anweisung `\caption*{}` verwendet werden.

Bilder mit `subfigure` nebeneinander anordnen

Eine elegante Methode, um mehrere Bilder nebeneinander anzuordnen, bietet das Zusatz-Paket `subfigure`: Die gleichnamige Anweisung bietet die Möglichkeit, die einzelnen Teil-Abbildungen mit separaten Bild-Unterschriften zu versehen und auf diese an anderer Stelle zu verweisen.

Beispiel:

```
% In der Präambel:

\usepackage{subfigure}
\usepackage{caption}

% Im Dokument:

\begin{figure}[h!]
  \centering
  \subfigure[Bildunterschrift links]{
    \includegraphics[angle=+30, origin=c, width=0.4\textwidth]{pics/tux.png}
    \label{fig-subfig1}
  }
  \subfigure[Bildunterschrift rechts]{
    \includegraphics[angle=-30, origin=c, width=0.4\textwidth]{pics/tux.png}
    \label{fig-subfig2}
  }
  \label{fig:subfigure-beispiel}
  \caption*{Gemeinsame Beschriftung für die beiden Abbildungen}
  \subref{fig-subfig1} und \subref{fig-subfig2}
\end{figure}
```

Ergebnis:



(a) Bildunterschrift links



(b) Bildunterschrift rechts

Gemeinsame Beschriftung für die beiden Abbildungen (a) und (b)

Mittels des Subfigure-Pakets können, sofern die Bildbreiten richtig gewählt werden, beliebig viele Bilder nebeneinander angeordnet werden. Erzwingt man mittels `\` einen Zeilenumbruch, so lassen sich ebenfalls gekachelt angeordnete Bilder-Reihen erstellen.

tabbing – Ausgerichteter Text

In einer `tabbing`-Umgebung werden üblicherweise innerhalb der ersten Zeile mittels `\=` Tabulatorpositionen festgelegt, an denen in den übrigen Zeilen der Text mittels den Sprungmarken `\>` ausgerichtet werden kann:

```

\begin{tabbing}
Name1: \= Text1 \\
Name2: \> Text2 \\
Name3: \> Text3 \\
...
\end{tabbing}

```

Durch die `tabbing`-Umgebung im obigen Beispiel könnte beispielsweise ein Dialog dargestellt werden, der sich auch über mehrere Seiten erstrecken darf; es können allerdings auch mehrere Tabulatoren innerhalb einer Zeile auftreten.

Tabulatoren können jederzeit neu gesetzt werden, wobei die bisherigen überschrieben werden. Üblicherweise möchte man allerdings bereits in der ersten Zeile die Tabulatoren anhand des längsten in einer „Spalte“ vorkommenden Textes festlegen. Dazu gibt es im wesentlichen zwei Möglichkeiten:

- Die erste Textzeile wird an notwendigen Stellen durch unsichtbaren Text aufgefüllt, der mittels `` erzeugt werden kann.
- Die längste Textzeile wird kopiert, am Anfang der `tabbing`-Umgebung mit Tabulator- statt Sprungmarken eingefügt und mittels `\kill` beendet. Die `\kill`-Anweisung bewirkt dabei, dass die soeben abgeschlossene Textzeile nicht gedruckt wird (die Tabulatoren bleiben jedoch gesetzt).

tabular und table – Tabellen

Tabellen werden in LaTeX üblicherweise mit Hilfe der `tabular`-Umgebung dargestellt. Diese hat folgende Syntax:

```

\begin{tabular}{Spaltenoptionen}
...
\end{tabular}

```


Spaltenoptionen

Mit den Spaltenoptionen wird festgelegt, wie die Ausrichtung der einzelnen Spalten erfolgen soll. Dabei sind folgende Angaben üblich:

- `l`: Spalte wird linksbündig ausgerichtet
- `r`: Spalte wird rechtsbündig ausgerichtet
- `c`: Spalte wird zentriert
- `p{4cm}`: Spalte wird linksbündig mit fester Breite ausgerichtet

Die Anzahl an Spalten einer Tabelle wird durch die Anzahl an angegebenen Spaltenausrichtungen festgelegt. Sollen am Rand der Tabelle oder zwischen den einzelnen Spalten vertikale Striche entlang der Tabelle gedruckt werden, so kann zwischen den einzelnen Spaltenausrichtungen ein `|`-Zeichen gesetzt werden; beispielsweise wird mit `\begin{tabular}{|l|c|c|}` eine Tabelle mit einer linksbündigen und zwei zentrierten Spalten eingeleitet, zwischen denen jeweils trennende Striche gezogen werden.

Bei den Optionen `l`, `r` und `c` wird die Spaltenbreite von LaTeX anhand des längsten Eintrags in der jeweiligen Spalte berechnet. Möchte man eine *linksbündige* Spalte mit fester Breite und gegebenenfalls automatischen Zeilenumbrüchen, so kann `p{breite}` verwendet werden. Möchte man auch zentrierte oder rechtsbündige Spalten mit fester Spaltenbreite erzeugen, so muss zusätzlich das Paket *array* verwendet werden.

Für Tabelleneinträge über mehrere Zeilen beziehungsweise Spalten können die Pakete *multirow* und *multicol* genutzt werden, die entsprechende Anweisungen bereitstellen.

Spaltentrennzeichen und neue Zeilen

Innerhalb der `tabular`-Umgebung werden die einzelnen Reihen allgemein mittels der Neue-Zeile-Anweisung `\\` voneinander getrennt. Eine Tabellenspalte kann sich im Quellcode gegebenenfalls also auch über mehrere Zeilen erstrecken, wenn dies einer besseren Lesbarkeit dient.

Innerhalb einer Reihe bewirkt das Zeichen `&` ein Trennen der einzelnen Spalten. Um horizontale Linien am Rand der Tabelle oder zwischen einzelnen Reihen zu ziehen, kann zu Beginn der Tabelle sowie jeweils hinter einem `\\` die Anweisung `\hline` („horizontal line“) geschrieben werden. Soll ein horizontaler Strich nicht über die gesamte Tabelle hinweg gesetzt werden, sondern nun für einzelne Spalten gelten, so kann man nach dem Beenden einer Zeile mittels `\\` auch beispielsweise `\cline{1-3}` schreiben, um einen waagrecht Strich unter die angegebenen Spalten-Einträge zu setzen. Man kann auch beispielsweise `\\ \cline{1-2} \cline{4-5}` schreiben, um horizontale Striche an genau den angegebenen Spalten zu setzen.

Jede Reihe sollte bei n Spalten stets $(n - 1)$ Spaltentrennzeichen `&` beinhalten. Ist dies nicht der Fall, verbleibt bei einer umrandeten Tabelle der rechte Spaltenrand in dieser Reihe.

Nach einer Neuen-Zeile-Anweisung `\\` kann wiederum ein in eckigen Klammern eine Längenangabe geschrieben werden, um die aktuelle Reihe um diesen Wert zu in ihrer Höhe zu verändern. Beispielsweise würde ein Zeilenumbruch

hinter der aktuellen Reihe einen vertikalen Abstand von `6pt` einfügen, was bei einer Schriftgröße von `12pt` einem 1,5-fachen Zeilenabstand entspräche.

Soll der Abstand zwischen den einzelnen Reihen allgemein vergrößert werden, so ist dies folgendermaßen möglich:

```
% Reihenabstand auf den 1,25-fachen Wert festlegen:
\renewcommand{\arraystretch}{1.25}

\begin{tabular}{Spaltenoptionen}

%% Die eigentliche Tabelle %%
```

(continues on next page)

```

\end{tabular}

% Reihenabstand wieder auf den normalen Wert zurücksetzen
\renewcommand{\arraystretch}{1}

```

Die table-Umgebung

Die eigentliche Tabelle, die durch `\begin{tabular}` und `\end{tabular}` begrenzt ist, kann zusätzlich in eine `table`-Umgebung gepackt werden. Dabei handelt es sich, ebenso wie bei `figure`, um eine Fließumgebung, bei der LaTeX anhand von den angegebenen Optionen die Positionierung der Tabelle selbst vornimmt.

```

\begin{table}[htpb]
\centering
\caption{Hier kommt die Tabellenbeschriftung hin.}
\label{tab:table-label}

\begin{tabular}{|c|l|}
\hline
1 & Erste Zeile \\
2 & erste Zeile \\
\hline
\end{tabular}
\end{table}

```

Auch bei der `table`-Umgebung sind folgende Optionen für mögliche Positionierungen der Tabelle möglich: `h` („here“) steht für die aktuelle Position, `t` („top“) für den Beginn der aktuellen Seite, `b` („bottom“) für das Ende der aktuellen Seite; eine weitere mögliche Option ist `p` („page“), bei der die Tabelle optional auch auf einer separaten Seite gedruckt werden darf. Mittels eines Ausrufezeichens kann eine Position erzwungen werden, beispielsweise bewirkt eine Optionsangabe von `[h!]`, dass die Tabelle nur an der aktuellen Stelle eingebunden werden darf.

Mittels der `\caption{}`-Anweisung kann die Tabelle beschriftet werden; in der klassischen Textsatzung wird diese Beschriftung, anders als bei Abbildungen, allerdings *über* die eigentliche Tabelle gesetzt. Mittels der `\label{}`-Anweisung kann die Tabelle zudem mit einem Label versehen werden, so dass auf sie an einer anderen Stelle im Dokument verwiesen werden kann.

Mehrseitige Tabellen

Tabellen, die mittels einer `tabular` bzw. `table`-Umgebung erstellt werden, erlauben keinen Seitenumbruch innerhalb der Tabelle. Bei längeren Tabellen empfiehlt sich die Verwendung des Pakets `longtable`, mit dem sich mehrseitige Tabellen erstellen lassen.

verbatim – Quelltext

Text innerhalb einer `verbatim`-Umgebung wird von LaTeX nicht interpretiert, sondern genau so dargestellt, wie er eingegeben wurde. Es finden also beispielsweise keine automatischen Zeilenumbrüche und keine Silbentrennungen statt; zusätzlich wird der *Schrifttyp* auf `typewriter` umgestellt.¹

¹ Die Quelltext-Umgebung `verbatim` kann auch durch `\begin{verbatim}` und `\end{verbatim}` begrenzt werden, wobei bei dieser Version Leerzeichen durch ein eigenes Symbol () dargestellt werden. Dies kann zur Hervorhebung der Anzahl von Leerzeichen in einer Codestelle nützlich sein.

Innerhalb eines Absatzes können kurze Quelltext-Passagen mittels `\verb| Quelltext |` hervorgehoben werden. Kommt in dem angegebenen Quelltext das Symbol `|` vor, so kann auch ein anderes Begrenzungszeichen für `\verb` verwendet werden, beispielsweise `\verb= Quelltezt =`.

`quote`, `quotation` und `verse` – Einrückungen

- Diese Umgebung wird üblicherweise für kurze Zitate verwendet, die aus einem einzelnen Absatz bestehen. Text innerhalb von `\begin{quote}` und `\end{quote}` wird links und rechts etwas eingerückt, die Zeilenlänge wird also gegenüber dem restlichen Text etwas verringert. Neue Absätze innerhalb einer `quote`-Umgebung beginnen ebenfalls linksbündig.
- Diese Umgebung wird üblicherweise für längere Zitate verwendet, die aus mehr als einem Absatz bestehen. Text innerhalb von `\begin{quotation}` und `\end{quotation}` wird ebenfalls links und rechts etwas eingerückt, die Zeilenlänge wird also gegenüber dem restlichen Text etwas verringert. Auch innerhalb einer `quotation`-Umgebung werden Absätze linksbündig dargestellt, wobei die erste Zeile eines neuen Absatzes leicht eingerückt wird.
- Diese Umgebung wird üblicherweise für Gedichte verwendet. Innerhalb von `begin{verse}` und `\end{verse}` werden einzelne Zeilen durch `\\` explizit beendet; sehr lange Zeilen werden in den nachfolgenden Zeilen zusätzlich eingerückt und so als zusammengehörig gekennzeichnet. Leerzeilen werden zur Trennung von Absätzen verwendet.

Zusatzpakete

In diesem Abschnitt sollen in alphabetischer Reihenfolge einige Pakete aufgelistet werden, die mittels der `\usepackage`-Anweisung eingebunden werden können und weitere hilfreiche Funktionen für spezielle Zwecke bereitstellen.

array – Bessere Spaltenoptionen für Tabellen

In der Standard-Version von LaTeX können, wie im Abschnitt *Spaltenoptionen* beschrieben, Spalten mit fester Breite nur linksbündig ausgerichtet werden.

Das Paket `array` ermöglicht es, Spalten mit fester Breite auch zentriert oder rechtsbündig auszurichten. Hierfür ist es sinnvoll, in der Präambel folgende „neue“ Spaltentypen zu definieren:¹

```
\usepackage{array}

\newcolumnmtype{L}[1]{>{\raggedright\let\newline\\\arraybackslash\hspace{0pt}}m{#1}}
\newcolumnmtype{C}[1]{>{\centering\let\newline\\\arraybackslash\hspace{0pt}}m{#1}}
\newcolumnmtype{R}[1]{>{\raggedleft\let\newline\\\arraybackslash\hspace{0pt}}m{#1}}
```

Damit können beispielsweise `L{5cm}`, `R{5cm}` oder `C{5cm}` als Spaltenoptionen verwendet werden, um linksbündige, rechtsbündige oder zentrierte Spalten mit fester Breite zu definieren. Der Text wird dabei vertikal stets zentriert ausgerichtet, also jeweils mittig in die Reihen der Tabellen gesetzt.

Soll die vertikale Ausrichtung der Zeilen jeweils bündig am unteren Zeilenrand erfolgen, wie dies beispielsweise für das Setzen von sehr langen Spaltennamen mittels `rotatebox{}` vorteilhaft sein kann, so können nach dem obigen Prinzip weitere Spaltentypen definiert werden, wobei lediglich `m{#1}` („middle“) durch `b{#1}` („bottom“) ersetzt werden muss.

Bei Verwendung von Spaltenoptionen mit fester Breite wird der Spaltentext bei Bedarf automatisch in der nächsten Zeile weitergeführt. Bei Verwendung der oben definierten Spaltenoptionen `L`, `C` und `R` kann eine Fortsetzung des Spaltentexts in der nächsten Zeile manuell mittels `\newline` explizit erzwungen werden.

blindtext – Blindtext setzen

Das Paket `blindtext` stellt die gleichnamige Anweisung `\blindtext` bereit, mit deren Hilfe – wie der Name schon sagt – ein Blindtext ins Dokument eingefügt werden kann. Dabei handelt es sich um einen Textabschnitt, der an sich keinen Sinn ergibt, aber als „Füllmaterial“ für Layout-Überlegungen dienen soll.

Das Paket bietet neben der Anweisung `\blindtext`, die einen einzelnen Absatz erstellt, folgende weitere Möglichkeiten:

- `\blindtext[x]` erstellt einen Absatz Mustertext mit `x` Wiederholungen.²

¹ Der Tipp stammt von dieser Seite.

² Da das Argument `x` in eckige Klammern gesetzt wird, kann es also auch weggelassen werden; für `x` wird dann der Standardwert (hier: 1) verwendet.

- `\Blindtext[y][x]` erstellt y Absätze an Mustertext mit x Wiederholungen.
- `\blinddocument` erstellt einen Mustertext mit 3 Seiten Länge.
- `\Blinddocument` erstellt einen Mustertext mit 16 Seiten Länge.

Beispiel:

```
% In der Präambel:
\usepackage{blindtext}
% Im Dokument:
\blindtext[1]
```

Ergebnis:

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Zusätzlich zu den obigen Anweisungen, die „normalen“ Blindtext erzeugen, gibt es mit dem `\blindtext`-Paket auch die Möglichkeit, beispielhafte Aufzählungen zu erzeugen:

- `\blinditemize` erstellt eine Muster-Aufzählung verschiedener Punkte ohne Nummerierung.
- `\blindenumerate` erstellt eine Muster-Aufzählung verschiedener Punkte mit Nummerierung.

Die obigen Anweisungen können beispielsweise beim Testen anderer Dokumentklassen (beispielsweise Brief), Schriftarten und Schriftgrößen genutzt werden.

caption – Bildunterschriften ohne Nummerierung

Das Paket `caption` stellt die Anweisung `\caption*{}` bereit, die beispielsweise innerhalb von `figure`- oder `table`-Umgebungen genutzt werden kann, um Bild- beziehungsweise Tabellen-Unterschriften ohne eine automatische Nummerierung zu setzen.

cmap – Copy und Paste in PDF-Dateien

Als das Textsatzungssystem (La-)TeX von Donald Knuth entwickelt wurde, gab es noch keine graphischen Bedienoberflächen und noch keine Maus als Bedienungs-Werkzeug; als Zielgeräte waren Drucker vorgesehen. Heutzutage hingegen ist es sehr nützlich, wenn aus den erstellten PDF-Dateien der darin enthaltene Text einfach via Copy-and-Paste in eine andere Datei übernommen werden kann. Hierfür genügt es, in der Präambel das Zusatz-Paket `cmap` mittels `\usepacke{cmap}` zu laden:

```
% Möglichst zu Beginn der Präambel:
\usepackage{cmap}
```

Das Paket sollte möglichst **zu Beginn** der Präambel eingebunden werden, um Konflikte mit anderen Zusatz-Paketen zu vermeiden. Innerhalb des Dokuments sind keine weiteren Anweisungen nötig.

colortbl – Tabellen mit farbigem Hintergrund

Das Paket `colortbl` ermöglicht es in Kombination mit dem Paket `xcolor`, in Tabellen auch nur einzelne Zellen mit einer anderen Hintergrund-Farbe hervorzuheben.

Beispiel:

```
% In der Präambel:  
  
\usepackage{xcolor, colortbl}  
\definecolor{grey}{rgb}{0.8,0.8,0.8}  
  
% Innerhalb des Dokuments:  
  
\begin{tabular}{|c|c|c|}  
  \hline  
  Spalte A & Spalte B & Spalte C \\  
  \hline  
  A1 & B1 & C1 \\  
  A2 & \cellcolor{grey} B2 & C2 \\  
  A3 & B3 & C3 \\  
  \hline  
\end{tabular}
```

Ergebnis:

Spalte A	Spalte B	Spalte C
A1	B1	C1
A2	B2	C2
A3	B3	C3

Die `\cellcolor{}`-Anweisung kann auch im Argument der `\multicolumn{}`-Anweisung genutzt werden, um einen Bereich über mehrere Spalten farblich hervorzuheben, beispielsweise `\multicolumn{3}{c}{\cellcolor{grey}Hallo Welt!}`. Dabei muss allerdings genau diese Verschachtelungs-Reihenfolge eingehalten werden, andernfalls erhält man eine Fehlermeldung.

geometry – Seitenränder festlegen

Das Paket `geometry` bietet eine einfache Möglichkeit, die Abstände zwischen dem Seitenrand und dem Textbereich einer Seite manuell einzustellen. Die Syntax hierzu ist folgende:

```
% In der Präambel:  
  
\usepackage[left=2.5cm, right=2.5cm, top=2.5cm, bottom=2.5cm]{geometry}
```

Die Angabe der einzelnen Abstände ist optional, es können somit beispielsweise auch nur die Werte für den oberen und unteren Seitenrand (`top` und `bottom`) angepasst werden; für die seitlichen Ränder gelten dann weiterhin die LaTeX-Standardwerte.

Sollen die so definierten Einstellungen, die Auswirkungen auf das gesamte Dokument haben, zu einem späteren Zeitpunkt innerhalb des Dokuments wieder geändert werden, so ist dies mittels der `\newgeometry{}`-Anweisung möglich:

```
% Innerhalb des Dokuments:  
  
\newgeometry{left=1cm, right=1cm, top=2.5cm, bottom=2.5cm}
```

Diese Anweisung hat Auswirkungen auf das Layout der aktuellen und aller folgenden Seiten beziehungsweise bis eine neue `\newgeometry{}`-Anweisung gesetzt wird.

hyperref – Verlinkungen erstellen

Das Paket `hyperref` dient zur Erstellung von Verlinkungen innerhalb des erstellten PDF-Dokuments. In der Präambel sollte dieses Pakete mittels `\usepackage{hyperref}` möglichst *am Ende* geladen werden, um Konflikte mit anderen Zusatzpaketen zu vermeiden.

Bindet man dieses Paket ohne weitere Optionen ein, so werden beispielsweise automatisch die Überschriften des Inhaltsverzeichnis in der PDF-Datei mit roten Rahmen versehen; sofern der PDF-Betrachter dies unterstützt, kann mit der Maus auf eine solche „Box“ geklickt werden, und man gelangt zur entsprechenden Stelle im Dokument. Die Boxen werden nur am Computer-Bildschirm angezeigt; schickt man die Datei an einen Drucker, so werden diese nicht mit gedruckt.³

Das Paket stellt die folgenden vier Anweisungen bereit:

- Mit `\url{webadresse}` kann eine Web-Adresse anklickbar verlinkt werden. Die als Argument angegebene Adresse wird als Ganzes in Maschinenschrift gedruckt.

```
% Möglichst am Ende der Präambel:  
  
\usepackage{url, hyperref}  
  
% Innerhalb des Dokuments:  
  
\url{http://www.grund-wissen.de}
```

Kommen in der angegebenen Webadresse Unterstriche vor, so müssen diese manuell mit einem vorangehenden Backslash-Zeichen versehen werden; alternativ dazu kann zusätzlich zum Paket `hyperref` das Paket `url` geladen werden, das genau hierfür Abhilfe schafft.

- Mit `\href{webadresse}{text}` kann ebenfalls ein anklickbarer Link auf eine Webadresse gesetzt werden; es wird allerdings nicht die Web-Adresse, sondern der als zweites Argument angegebene Text gedruckt und dieser mit der Adresse verlinkt.

```
\href{http://www.grund-wissen.de}{Grund-Wissen-Webseite}
```

Die `\href{}`-Anweisung kann ebenso verwendet werden, um Email-Adressen zu setzen. Dazu muss man lediglich die Ziel-Adresse mit dem Vermerk `mailto:` versehen werden:⁴

```
\href{mailto:email@adresse.de}{email@adresse.de}
```

In seltenen Fällen ist es auch hilfreich, mit `\href{}` auf lokale Dateien verlinken zu können. Hierzu wird der Ziel-Pfad mit dem Vermerk `run:` versehen.

```
\href{run:/path/to/file.pdf}{Dateiname}
```

Die Datei wird dann mit der für den jeweiligen Dateityp vorgesehenen Standard-Anwendung geöffnet.

- Mit `\ref{labelname}` kann auf eine andere Stelle im aktuellen PDF-Dokument verlinkt werden; diese muss im LaTeX-Quellcode mittels der Anweisung `\label{labelname}` markiert werden. Als `labelname` sollten dabei nur ASCII-Symbole (Groß- und Kleinbuchstaben, Zahlen, Bindestrich, Doppelpunkt) verwendet werden, um mögliche Kodierungsfehler zu vermeiden.⁵

Verlinkt man mit der `\ref{}`-Anweisung auf ein Label, so wird im PDF-Dokument an dieser Stelle der Wert des so genannten `refstepcounters` eingefügt; beispielsweise könnte dieser bei der ersten

³ Möchte man die farbigen Umrandungsboxen auch in der PDF-Version deaktivieren und dafür die Verlinkungen durch eine blaue Schriftfarbe hervorheben, so ist dies mittels `\hypersetup{colorlinks=true, linkcolor=blue}` möglich (siehe Abschnitt *Konfiguration des hyperref-Pakets*).

⁴ Auch für das Setzen von Email-Adressen ist das Zusatzpaket `url` nützlich, um nicht weiter auf mögliche Unterstrich-Zeichen achten zu müssen.

⁵ Die Anweisungen `\label{}`, `\ref{}` und `\pageref{}` können, wie bereits im Abschnitt *Querverweise* beschrieben, auch ohne Einbindung des Pakets `hyperref` verwendet werden; in den PDF-Dateien sind die jeweiligen Stellen dann aber nicht anklickbar.

Abbildung im vierten Abschnitt des ersten Kapitels als Ergebnis 1.4.1 liefern. Möchte man anstelle dieses Zählers die jeweilige Seitenzahl ausgeben, so ist dies mit der Anweisung `\pageref{label}` möglich.⁶

- Mit `\hyperlink{target}{Link-Text}` wird ein Link auf eine Textstelle erzeugt, die sich auch innerhalb eines Absatzes befinden kann – beispielsweise eine Definition eines Fachbegriffs oder einer Abkürzung. Das jeweilige Ziel wird dabei mittels der Anweisung `\hypertarget{target}{Eigentlicher Text}` definiert.

Konfiguration des hyperref-Pakets

Das Paket `hyperref` bietet zudem vielerlei Möglichkeiten, das Aussehen der Links anzupassen; dies ist mittels der Anweisung `\hypersetup{}` am Ende der Präambel möglich. Unter anderem kann damit das Aussehen der Links festgelegt werden:

```
% Am Ende der Präambel:

\hypersetup{

  unicode      = true,      % Umlaute in Links erlauben
  colorlinks   = true,      % Links anstelle von Umrandungsboxen farbig hervorheben
  citecolor    = green,     % Farbe von Links auf Bibliographie-Einträge festlegen
  linkcolor    = blue,      % Farbe von Links auf Sprungmarken festlegen
  urlcolor     = cyan,      % Farbe von Links auf Webseiten festlegen
  % ...
}
```

Mit der `\hypersetup{}`-Anweisung können zudem Meta-Tags in die PDF-Datei geschrieben werden, die Informationen über den Titel des Dokuments, den Autor-Namen usw. beinhalten; in einem PDF-Betrachter können diese als Dokument-Eigenschaften angezeigt werden. Insbesondere bei Veröffentlichungen im Internet ist ein Setzen solcher Informationen empfehlenswert, da Suchmaschinen wie Google diese in ihren Algorithmen mit auswerten.

```
% Am Ende der Präambel:

\hypersetup{

  % ... andere Konfigurationen ...

  pdftitle     = {Dokumentname},
  pdfsubject   = {Kurzbeschreibung},
  pdfauthor    = {Autornamen},
  pdfkeywords  = {Schlagwort 1} {Schlagwort 2},
  pdfnewwindow= true,
}
```

Eine vollständige Liste der `hyperref`-Optionen findet sich auch im [LaTeX-Wörterbuch](#).

longtable – Mehrseitige Tabellen

Das Paket `longtable` stellt eine gleichnamige Umgebung bereit, die zum Erstellen von beliebig langen Tabellen verwendet werden kann. Die Syntax ist dabei im Wesentlichen an die `tabular`-Umgebung angelehnt; es können allerdings unterschiedliche Kopf- und Fußzeilen für die erste Seite der Tabelle sowie für die folgenden Tabellen-Seiten definiert werden.

⁶ Als zusätzliche Alternative kann das Paket `nameref` verwendet werden, das die gleichnamige Anweisung `\nameref{}` bereitstellt. Diese gibt den Inhalt des jeweils verlinkten Labels aus. Steht beispielsweise ein Label `\label{sec:Abschnittname}` hinter einer Überschrift, so wird der Name der Überschrift als Linkbezeichnung ausgegeben; hat man die Überschrift mit `\section[Kurzbezeichnung]{Langer Überschriftsname}` gesetzt, so wird nur die Kurzbezeichnung gedruckt.

Beispiel:

```
% In der Präambel:

\usepackage{multicol}
\usepackage{array, longtable}
\newcolumnmtype{L}[1]{>{\raggedright\let\newline\\\arraybackslash\hspace{0pt}}m{#1}}
\newcolumnmtype{C}[1]{>{\centering \let\newline\\\arraybackslash\hspace{0pt}}m{#1}}
\newcolumnmtype{R}[1]{>{\raggedleft \let\newline\\\arraybackslash\hspace{0pt}}m{#1}}

% Im Dokument:

\begin{longtable}{|C{3cm}|C{7cm}|C{3cm}|}
  \hline
  &&\[-11pt]

  % Kopfzeile zu Beginn der Tabelle:
  Spalte A & Spalte B & Spalte C \\
  \hline \endfirsthead

  % Kopfzeile für die Folgeseiten:
  A & B & C \\
  \hline \endhead

  % Fußzeilen bei Seiten-Umbrüchen mitten in der Tabelle:
  \multicolumn{3}{|c|}{Fortsetzung folgt...} \\
  \hline \endfoot

  % Fußzeile am Ende der Tabelle:
  \multicolumn{3}{|c|}{Ende!} \\
  \hline \endlastfoot

  % Eigentlicher Tabellen-Inhalt:
  A1 & B1 & C1 \\
  A2 & B2 & C2 \\
  A3 & B3 & C3 \\
  \hline
\end{longtable}
```

Ergebnis:

Spalte A	Spalte B	Spalte C
A1	B1	C1
A2	B2	C2
A3	B3	C3
Ende!		

In diesem Beispiel wurden wieder die im Abschnitt *Tabular* beschriebenen Eigen-Definitionen für zentrierte Spalten mit fester Breite genutzt. Da der eigentliche Tabellen-Inhalt im Beispiel nur sehr kurz ist und auf eine Seite passt, wird nur der erste Tabellenkopf sowie die Fußzeile für das Tabellen-Ende gedruckt.

Der Aufbau einer `longtable` unterscheidet sich also nur durch die explizite Definition der Kopf- und Fußzeilen von einer „normalen“ `tabular`-Umgebung.⁷ Der wesentliche Vorteil liegt dabei darin, dass

⁷ Leider können in einer `longtable` waagrechte Linien nur mittels `\hline`, nicht jedoch mittels `\toprule`, `\midrule` und `\bottomrule` gesetzt werden.

Möchte man beispielsweise den Abstand zwischen einer `\hline` und dem nachfolgenden Text vergrößern, so muss bei einer vierspaltigen Tabelle und Schriftgrad 10 auf eine Anweisung der Form `&&&\[-8pt]` zurückgegriffen werden. Sie bewirkt, dass eine weitere (komplette) Zeile mit leeren Spalteneinträgen eingefügt wird; der vertikale Abstand zum folgenden, eigentlichen Tabellen-Inhalt würde somit um 10pt erhöht. Durch die zusätzliche Angabe von `[-8pt]` hinter dem Zeilen-Trennzeichen wird die folgende Zeile allerdings um 8pt nach oben verschoben. Dadurch resultiert schließlich ein Abstand von 2pt zur davor platzierten `\hline`.

man sich keine Gedanken darüber machen muss, an welcher Stelle die Seitentrennung erfolgt – LaTeX übernimmt dies von selbst.

lscapex – Bereiche in Querformat setzen

Das Paket `lscapex` stellt die Umgebung `landscape` bereit, mit deren Hilfe einzelne Bereiche eines Dokuments im Querformat gedruckt werden können. Dies ist beispielsweise für breite Tabellen hilfreich.

Beispiel:

```
% In der Präambel:  
  
\usepackage{lscapex}  
  
% Im Dokument:  
  
\begin{landscape}  
  \begin{tabular}{|c|c|c|}  
    \hline  
    Spalte A & Spalte B & Spalte C \\  
    \hline  
    A1 & B1 & C1 \\  
    A2 & B2 & C2 \\  
    A3 & B3 & C3 \\  
    \hline  
  \end{tabular}  
  
\end{landscape}
```

Ergebnis:

Spalte A	Spalte B	Spalte C
A1	B1	C1
A2	B2	C2
A3	B3	C3

Der entsprechende Bereich muss also lediglich durch `\begin{landscape}` und `\end{landscape}` eingeschlossen werden.

multicol – Tabelleneinträge über mehrere Spalten

Das Paket `multicol` stellt die Anweisung `\multicolumn{}` bereit, mit deren Hilfe Einträge in Tabellen über mehrere Spalten hinweg gesetzt werden können. Die Syntax hierfür ist folgende:

```
% In der Präambel:  
  
\usepackage{multicol}
```

(continues on next page)

```
% Innerhalb des Dokuments:
\multicolumn{AnzahlSpalten}{Ausrichtung}{Text}
```

Die Ausrichtung des angegebenen Textes beginnt dann mit in der Spalte, in der die `\multicolumn{}`-Anweisung steht, und umfasst insgesamt die mit `AnzahlSpalten` angegebene Anzahl an Spalten. Als Ausrichtung kann `l`, `c` oder `r` gewählt werden; Trennstriche am linken beziehungsweise rechten Rand des `multicolumn`-Eintrags können mittels eines `|`-Zeichens vor beziehungsweise nach dem Ausrichtungszeichen gesetzt werden.

multirow – Tabelleneinträge über mehrere Zeilen

Das Paket `multirow` stellt die gleichnamige Anweisung `\multirow{}` bereit, mit deren Hilfe Einträge in Tabellen über mehrere Zeilen hinweg ausgerichtet werden können. Die Syntax hierfür ist folgende:

```
% In der Präambel:
\usepackage{multirow}

% Innerhalb des Dokuments:
\multirow{AnzahlReihen}{Breite}{Text}
```

Die Ausrichtung des angegebenen Textes innerhalb der Spalte, in der die `\multirow{}`-Anweisung steht, beginnt mit der aktuellen Reihe und umfasst insgesamt die mit `AnzahlReihen` angegebene Anzahl an Reihen. Der Inhalt der Reihe wird zentriert zu den mit `AnzahlReihen` angegebenen Reihen ausgerichtet. Als vertikal für den Text zu reservierende Breite wird meist `*` für eine automatische Breite gewählt, die Breite der auszurichtenden Reihe entspricht dann der Breite der angegebenen Anzahl von Reihen.

paralist – Kompakte Aufzählungen

Das Paket `paralist` stellt die Umgebungen `compactitem` und `compactenum` bereit, mit deren Hilfe kompakte Aufzählungen gesetzt werden können:

```
% Kompakte Aufzählung ohne Nummerierung:
\begin{compactitem}
  \item Eintrag 1
  \item Eintrag 2
\end{compactitem}

% Kompakte Aufzählung mit Nummerierung:
\begin{compactenum}
  \item Eintrag 1
  \item Eintrag 2
\end{compactenum}
```

units – Einheiten setzen

In der Typographie ist es allgemein üblich, mathematische oder physikalische Symbole kursiv zu drucken, die zugehörigen Einheiten hingegen in aufrechter Schriftform („roman“). Um insbesondere bei der Eingabe

von Einheiten im Mathe-Modus nicht ständig den Schrift-Typ ändern zu müssen, kann anstelle dessen in der Präambel das Paket `units` mittels `\usepackage{units}` eingebunden werden.

Das Paket `units` stellt die Anweisung `\units[Zahlenwert]{Einheit}` bereit, die sowohl in Text- wie auch in Mathe-Umgebungen genutzt werden kann; sie stellt einerseits sicher, dass die angegebene Einheit aufrecht gedruckt wird, andererseits verringert sie geringfügig den horizontalen Abstand zwischen Zahlenwert und Einheit, so dass diese optisch als zusammengehörig, aber nicht gequetscht erscheinen. Die Angabe des Zahlenwerts ist optional, es kann also auch nur `\units{Einheit}` gesetzt werden, beispielsweise um beim Setzen physikalischer Aufgaben eine Einheiten-Kontrolle vorzunehmen.

Sowohl bei der Angabe des Zahlenwerts wie auch bei der Angabe der Einheit dürfen weitere mathematische Anweisungen gesetzt werden; beispielsweise kann mittels `\unit{\frac{km}{h}}` die Ausgabe $\frac{\text{km}}{\text{h}}$ erzeugt werden. Eine solche Bruch-Schreibweise wird häufig bei Formeln verwendet, die eigene Absätze darstellen; bei Angaben von Einheiten innerhalb einer Zeile kann hingegen die zusätzlich bereitgestellte Anweisung `\unitfrac[Zahlenwert]{Zähler}{Nenner}` verwendet werden, die beispielsweise für den Quellcode `\unitfrac{km}{h}` die Ausgabe km/h liefert.

urls – Links auf Webseiten

Bindet man dieses Paket mittels `\usepackage{url}` in die Präambel ein, so müssen innerhalb von `\url{}`-Anweisungen nicht manuell Backslash-Symbols `\` vor Unterstriche gesetzt werden; diese sind in LaTeX prinzipiell ja nur für den Mathe-Modus vorgesehen, um tiefgestellten Text zu erzeugen.

Das Paket `url` bewirkt, dass Unterstriche in Web-Adressen, auf die mittels der `\url{}`-Anweisung verlinkt wird, keinen Ärger mehr bereiten. Üblicherweise wird das `url`-Paket zusätzlich zum Paket `hyperref` verwendet, welches die mittels der `\url{}`-Anweisung erstellten Links „anklickbar“ macht; das Paket `url` sollte in der Präambel hierbei **vor** dem `hyperref`-Paket geladen werden.

Seitenlayout

Das Standard-Layout eines Dokuments wird in der Präambel der LaTeX-Datei durch die Wahl der Dokumentklasse sowie eines Seiten-Stils festgelegt. Bei der Wahl der Dokumentklasse sind für das Seitenlayout insbesondere die Zusatz-Optionen von Bedeutung:

Beispiel:

```
\documentclass[a4paper, 12pt, halfparskip]{article}
```

Durch die im obigen Beispiel angegebenen Optionen wird als Seitengröße des Dokuments `DinA4` gewählt und die Standard-Schriftgröße auf `\unit[12]{pt}` gesetzt. Durch Verwendung der Option `halfparskip` werden einzelne Absätze, die im LaTeX-Code durch leere Zeilen begrenzt werden, auch in der Druckversion durch eine Leerzeile getrennt.

Einseitige und zweiseitige Dokumente

Als weitere Option kann bei der Wahl der Dokumentklasse beispielsweise mittels `\documentclass[twoside]{report}` festgelegt werden, dass das vorliegende Dokument in Buchform gedruckt werden soll. Dies hat als Konsequenz, dass zwischen ungeradzahligen und geradzahligen Seiten unterschieden werden muss: Ungeradzahlige Seiten befinden sich bei einem aufgeschlagenen Buch stets auf der rechten, geradzahlige entsprechend auf der linken Seite.

Da Bücher meist von vorne nach hinten durchblättert werden, beginnen neue Kapitel meist auf den rechten, also ungeradzahligen Seiten; notfalls wird dabei in Kauf genommen, dass die linke (geradzahlige) Buchseite leer bleibt. Bei der Dokumentklasse `book` beziehungsweise der auf europäische Seitenformate angepassten Version `scrbook` wird ein zweiseitiges Dokumentenlayout als Standard angesehen; man kann dies bei Bedarf mittels `\documentclass[oneside]{scrbook}` ändern.

Standardmäßig gelten für die Standard-Dokumentklassen folgende Einstellung:

- Bei `book` beziehungsweise `scrbook` wird ohne weitere Einstellungen *zweiseitig* gedruckt.
- Bei `report` beziehungsweise `scrreprt` wird ohne weitere Einstellungen *einseitig* gedruckt.
- Bei `article` beziehungsweise `scrartcl` wird ohne weitere Einstellungen *einseitig* gedruckt.

Auch „einseitige“ Dokumente können selbstverständlich beim Drucken beidseitig gedruckt werden; die Bezeichnung bezieht sich lediglich auf das Layout des PDF-Dokuments: Die Festlegung auf `oneside` beziehungsweise `twoside` hat unter anderem auch Auswirkungen auf Kopf- und Fußzeilen sowie auf die jeweiligen Seitenränder.

Kopf- und Fußzeilen

LaTeX druckt mit den Standard-Einstellungen automatisch auf jede Seite unten mittig die jeweilige Seitennummer. Soll dies verhindert oder zusätzlich der jeweilige Abschnittsname als Kopfzeile jeder Seite gedruckt werden, so kann dieses mittels der Funktion `\pagestyle{}` angepasst werden:

- Mit `\pagestyle{empty}` wird festgelegt, dass weder Kopf- noch Fußzeilen gedruckt werden.

- Mit `\pagestyle{plain}` wird festgelegt, dass ausschließlich Fußzeilen, jedoch keine Kopfzeilen gedruckt werden.
- Mit `\pagestyle{headings}` wird festgelegt, dass sowohl Fußzeilen als auch Kopfzeilen mit den jeweiligen Abschnitts-Überschriften gedruckt werden.

Die Anweisung `\pagestyle{}` kann sowohl in der Präambel als auch an jeder anderen Stelle im Dokument genutzt werden; sie hat Auswirkung auf alle folgenden Seiten. Soll nur das Layout der jeweils aktuellen Seite verändert werden, so kann anstelle von `\pagestyle{}` die Funktion `\thispagestyle{}` mit gleicher Syntax verwendet werden.

Mit `\pagestyle{myheadings}` kann das Design der Kopf- und Fußzeilen individuell festgelegt werden; üblicherweise werden dafür allerdings die im Folgenden näher beschriebenen Pakete `fancyhdr` oder `scr-layer-scrpage` verwendet.

Individuelle Kopf- und Fußzeilen mit fancyhdr:

Das Paket `fancyhdr`, das für die Dokumentklassen `article`, `report` und `book` vorgesehen ist, stellt den neuen Pagestyle `fancy` bereit. Das Paket mit den Basis-Einstellungen wird folgendermaßen eingebunden:¹

```
% In der Präambel:
\usepackage{fancyhdr}

% ...

\pagestyle{fancy}
```

Für das Layout der Kopfzeilen steht anschließend die Anweisung `\fancyhead{}`, für die Definition der Fußzeilen die Anweisung `\fancyfoot{}` zur Verfügung. Als Argumente werden in geschweiften Klammern die zu druckenden Informationen angegeben; zudem wird in eckigen Klammern die horizontale Position festgelegt, an der die Informationen gedruckt werden sollen: `L` steht für „left“, `R` für „right“ und `C` für „center“; diese Angaben können wahlweise mit `O` für „odd“ (ungeradzahlig, rechts) oder `E` für „even“ (geradzahlig, links) kombiniert werden:

```
% Bisherige Einstellungen deaktivieren (optional):
\fancyhead{}
\fancyfoot{}

\fancyhead[RO,LE]{\today}    % Aktuelles Datum auf ungeradzahligen Seiten rechts,
                             % auf geradzahligen links ausgeben.

\fancyfoot[RO,LE]{\thepage}  % Seitennummer auf ungeradzahligen Seiten rechts,
                             % auf geradzahligen links ausgeben.
```

Zudem kann festgelegt werden, ob beziehungsweise mit welcher Stärke eine horizontale Linie zwischen dem eigentlichen Text und der Kopf- beziehungsweise Fußzeile gedruckt werden soll:

```
\renewcommand\headrulewidth{2pt}  % Liniendicke zwischen Kopfzeile und Text festlegen
\renewcommand\footrulewidth{0pt}  % Linie zwischen Fußzeile und Text deaktivieren
```

Auch mehrzeilige Kopfzeilen sind möglich; hierfür muss allerdings mittels beispielsweise `\setlength\headheight{24pt}` die Höhe der Kopfzeilen-Box angepasst werden. Bei der Festlegung der Kopfzeile kann die Zeilentrennung dann wie üblich mittels `\\` erreicht werden.

Individuelle Kopf- und Fußzeilen mit sclayer-scrpage:

Das Paket `sclayer-scrpage`, das für die Dokumentklassen `scrtartcl`, `scrrprt` und `scrbook` vorgesehen ist, stellt den neuen Pagestyle `scrheadings` bereit. Die Syntax für das Einbinden des Pakets und die

¹ Sollte die normale Textbreite `\textwidth` manuell geändert werden, so muss `\pagestyle{fancy}` anschließend erneut aufgerufen werden, da sich das Layout der Kopf- und Fußzeilen an dieser Länge orientiert.

Festlegung der einzelnen Informationen sieht für einseitige Dokumente beispielsweise folgendermaßen aus:²

```
% Präambel:

\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}

% Bisherige Einstellungen für Kopf- und Fußzeilen löschen:
\clearpairofpagestyles

% Zentriert auf linken Seiten die aktuelle Kapitelüberschrift,
% auf rechten Seiten die Überschrift des aktuellen Abschnitts ausgeben:
\thead{\headmark}

% Zentriert die Seitenzahl ausgeben (auch beim Seitenstil "scrplain"):
\tfoot*{\pagemark}

\pagestyle{scrheadings}
```

Durch die Anweisung `\headmark` wird bei Verwendung der Voreinstellungen automatisch auf linken Seiten die aktuelle Kapitelüberschrift, und auf rechten Seiten die Überschrift des aktuellen Abschnitts inklusive Nummer ausgegeben. Diese Voreinstellungen können mit den Anweisungen `\automark{}` beziehungsweise `\automark*{}` angepasst werden; erstere überschreibt alle bisherigen Einstellungen, letztere „überlagert“ lediglich die bisherigen Einstellungen.

Möchte man beispielsweise auf den rechten Seiten so lange die Kapitel-Überschrift ausgeben, bis der erste Abschnitt beginnt, so kann dies folgendermaßen erreicht werden:

```
% Sowohl auf rechten Seiten (eckige Klammern) als auch auf linken Seiten
% (geschweifte Klammern) Kapitel-Überschriften ausgeben:
\automark[chapter]{chapter}

% Linke Seiten-Definitionen unverändert lassen, auf rechten hingegen
% -- sofern vorhanden -- Abschnitts-Überschriften ausgeben:
\automark*[section]{}
```

Bei einseitigen Dokumenten existieren für LaTeX nur „rechte“ Seiten; die Seiten werden dann also unabhängig von ihrer Nummer als ungerade gewertet. Für zweiseitige Dokumente können die einzelnen Informationen mittels folgender Anweisungen explizit gesetzt werden:

```
% Inhalte der Kopfzeilen für rechte Seiten:
\lehead[Inhalt plain.scrheadings]{Inhalt scrheadings} % left even
\cehead[Inhalt plain.scrheadings]{Inhalt scrheadings} % center even
\rehead[Inhalt plain.scrheadings]{Inhalt scrheadings} % right even

% Inhalte der Kopfzeilen für linke Seiten:
\lohead[Inhalt plain.scrheadings]{Inhalt scrheadings} % left odd
\cohead[Inhalt plain.scrheadings]{Inhalt scrheadings} % center odd
\rohead[Inhalt plain.scrheadings]{Inhalt scrheadings} % right odd
```

In den eckigen Klammern können optional die Inhalte für den Seitenstil `scrplain` festgelegt werden; soll dieser für eine bestimmte Angabe mit dem Inhalt des Seitenstils `scrheadings` übereinstimmen, so kann beispielsweise anstelle von `\cehead[Inhalt]{Inhalt}` auch `\cehead*{Inhalt}` geschrieben werden. Sollen die gleichen Einstellungen sowohl für ungerade als auch für gerade Seitenzahlen gelten, so kann beispielsweise auch einfacher `\thead{}` als Anweisung genutzt werden.

² In der Original-Anleitung wird empfohlen, das Paket `scrlayer-scrpage` nicht in Kombination mit den Seitenstilen `headings` und `myheadings` zu verwenden; stattdessen sollte unbedingt `scrheadings` verwendet werden.

Das Paket `scrlayer-scrpage` ist mit dem Vorgänger-Paket `scrpage2` weitestgehend abwärts-kompatibel, so dass für eine Aktualisierung in den meisten Dokumenten lediglich `\usepackage{scrpage2}` durch `\usepackage{scrlayer-scrpage}` ersetzt werden muss.

Allgemein müssen nicht alle obigen Anweisungen genutzt werden, sondern nur diese, die für die gewünschten Positionen vorgesehen sind. Für die Fußzeilen existieren die gleichen Anweisungen, wobei lediglich `head` durch `foot` ersetzt werden muss.

Seitenränder

In LaTeX werden bei Verwendung von `a4paper` als Dokumentgröße und Standard-Einstellungen verhältnismäßig breite Seitenränder (1 in $\approx 2,54$ cm) gedruckt. Dies hat damit zu tun, dass aus Sicht der Typographie die einzelnen Zeilen nicht zu viel Text beinhalten sollten (nur circa 70 Zeichen), um die Lesbarkeit zu erhöhen.

Bei zweiseitigen Dokumenten ist der innere Rand schmaler als der äußere; er ist so breit gewählt, dass ein Buch nach der Bindung im aufgeschlagenen Zustand mittig ebenso viel Abstand zwischen den Texten auf der linken und der rechten Seite aufweist wie die Texte zu den jeweiligen Seitenrändern.

Anpassung der Seitenränder mittels `geometry`

Möchte man die Seitenränder allgemein anpassen, so ist dies am einfachsten durch Verwendung des `geometry`-Pakets möglich:

```
% In der Präambel:
\usepackage[left=2.5cm, right=2.5cm, top=2.5cm, bottom=2.5cm]{geometry}
```

Sollen die so definierten Einstellungen, die Auswirkungen auf das gesamte Dokument haben, zu einem späteren Zeitpunkt innerhalb des Dokuments wieder geändert werden, so ist dies mittels der `\newgeometry{}`-Anweisung möglich:

```
% Innerhalb des Dokuments:
\newgeometry[left=1cm, right=1cm, top=2.5cm, bottom=2.5cm]
```

Diese Anweisung hat Auswirkungen auf das Layout der aktuellen und aller folgenden Seiten beziehungsweise bis eine neue `\newgeometry{}`-Anweisung gesetzt wird. Um lediglich die aktuelle Seite etwas zu vergrößern, kann zudem als Anweisung `\enlargethispage{3cm}` o.ä. verwendet werden.

Abstände manuell festlegen

In LaTeX gibt es mehrere Variablen, die das Seitenlayout beeinflussen. Mittels der Pakete `layout` beziehungsweise `layouts` können die Namen dieser Variablen sowie die bei den aktuellen Einstellungen vorherrschenden Werte angezeigt werden.

- Die für das Seitenlayout relevanten Variablen sowie ihre aktuellen Werte können mittels des `layout`-Pakets folgendermaßen angezeigt werden:

```
% Präambel:
\documentclass[halfparskip,12pt,twoside]{scrreprt}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[german]{babel}
\usepackage[top=2cm,bottom=2.5cm]{geometry}
\usepackage{layout}

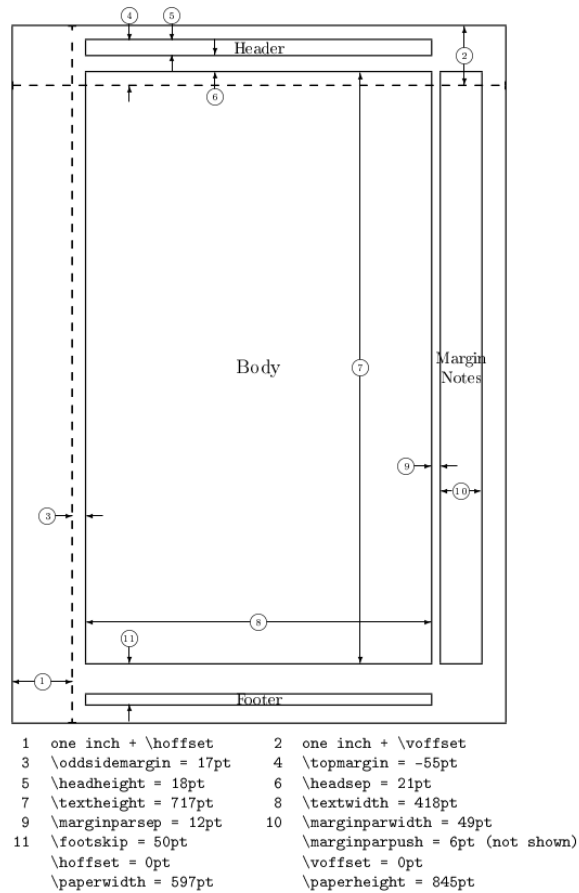
\begin{document}

% Seitenlayout-Variablen und aktuelle Werte ausgeben:
\layout
```

(continues on next page)


```
\end{document}
```

Der obige Code liefert etwa folgendes Ergebnis:



Die einzelnen Längen können mittels `\setlength{laengen-variable}{wert}` gesetzt werden, beispielsweise `\setlength{\oddsidemargin}{36pt}`; selbstverständlich können auch andere *Längenmaße* wie mm, cm oder in genutzt werden. Die Werte der Längen-Variablen `\topmargin`, `\oddsidemargin` und `\evensidemargin` werden dabei zu den beispielsweise mittels des `geometry`-Pakets eingestellten Randwerten hinzuaddiert.

- Weitere beispielsweise für das Layout der einzelnen Absätze relevanten Variablen sowie ihre aktuellen Werte lassen sich mittels des Pakets `layouts` folgendermaßen anzeigen:

```

% Präambel:
\documentclass[halfparskip,12pt,twoside]{scrreprt}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[german]{babel}
\usepackage[top=2cm,bottom=2.5cm]{geometry}
\usepackage{layouts}
\usepackage{caption}

\begin{document}

\setuplayouts

% Allgemeines Absatz-Layout anzeigen:

```

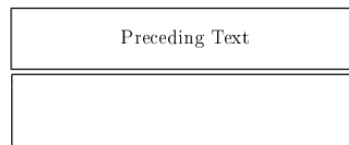
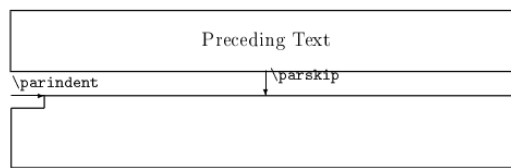
(continues on next page)

```
\begin{figure}
  \paragraphdiagram
  \caption*{Für einzelne Absätze relevante Variablen.}
\end{figure}

% Aktuelles Absatz-Layout anzeigen:
\begin{figure}
  \currentparagraph
  \paragraphdesign
  \caption*{Absatz-Variablen des aktuellen Dokuments.}
\end{figure}

\end{document}
```

Als Ergebnis erhält man beispielsweise:



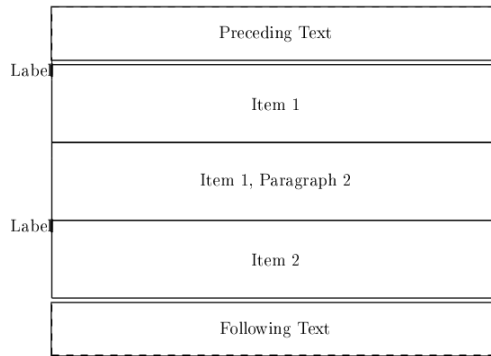
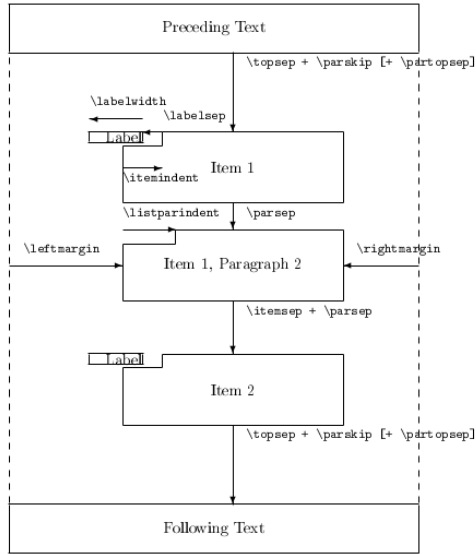
Lengths are to the nearest pt.
\parindent = 0pt \parskip = 7pt
\baselineskip = 15pt \linewidth = 418pt

- Nach dem gleichen Prinzip lassen sich mittels des `layouts`-Pakets auch die für Aufzählungen relevanten Variablen und ihre Werte anzeigen:

```
\begin{figure}
  \listdiagram
  \caption*{Für Listen relevante Variablen.}
\end{figure}

\begin{figure}
  \listdesign
  \caption*{Listen-Variablen des aktuellen Dokuments.}
\end{figure}
```

Ergebnis:



Lengths are to the nearest pt.
 $\leftmargin = \text{Opt}$ $\rightmargin = \text{Opt}$
 $\itemindent = \text{Opt}$ $\labelwidth = \text{Opt}$
 $\labelsep = \text{Opt}$ $\listparindent = \text{Opt}$
 $\topsep = \text{Opt}$ $\parskip = 7\text{pt}$
 $\partopsep = \text{Opt}$ $\parsep = \text{Opt}$
 $\itemsep = \text{Opt}$

In der [Dokumentation des layouts-Pakets](#) sind weitere entsprechende Beispiele aufgelistet, welche die Platzierung von Fußnoten sowie das Aussehen des Inhaltsverzeichnisses näher beschreiben.

Zähler und Makros

In LaTeX können gibt es einige (Zähl-)Variablen, die standardmäßig genutzt werden können; zudem können nach Belieben eigene Variablen definiert werden. Ebenso können eigenen Anweisungen („Makros“) definiert werden, so dass beispielsweise sich häufig wiederholende Text-Formatierungen mit geringerem Aufwand erreichen lassen.

Zähler-Variablen

Alle Zähler-Variablen, die von LaTeX verwendet werden, haben Standard-Namen; diese wiederum haben entsprechende LaTeX-Anweisungen, welche die jeweiligen Zähler-Werte ausgeben. Um beispielsweise innerhalb eines LaTeX-Dokuments den Wert eines Zählers mit dem Namen `number` auszugeben, genügt an dieser Stelle ein Aufruf von `\thenumber`.

Verwendungszweck	Zähler-Variablen
Kapiteileinteilung	<code>part</code> , <code>chapter</code> , <code>section</code> , <code>subsection</code> , <code>subsubsection</code> , <code>paragraph</code> , <code>subparagraph</code>
Seiten-Nummerierung	<code>page</code>
Aufzählungs-Nummerierung	<code>enumi</code> , <code>enumii</code> , <code>enumiii</code> , <code>enumiv</code>
Sonstige Nummerierungen	<code>figure</code> , <code>table</code> , <code>footnote</code> , <code>equation</code>

Sämtliche in der obigen Tabelle aufgelistete Zähler haben zu Beginn eines Dokuments den Wert Null und beim Aufruf der Anweisung, das die Numerierung erzeugt, *vor* der Ausgabe um 1 erhöht. Eine Ausnahme hierbei bildet der Zähler `page`; dieser hat zu Beginn des Dokuments den Wert Eins und wird stets *nach* einer Ausgabe automatisch erhöht.

Die Anpassung der Zähler wird von LaTeX weitgehend automatisch vorgenommen. Beispielsweise wird durch die Anweisung `\section{}` die Zähler-Variable `section` um 1 erhöht; zugleich wird dadurch die Zähler-Variable `subsection` wieder auf 0 gesetzt.

Manuell können folgende Anweisungen genutzt werden, um die jeweiligen Zähler zu beeinflussen:

Anweisung	Beschreibung
<code>\newcounter{name}</code>	Definition eines neuen Zählers
<code>\setcounter{name}{Wert}</code>	Wertzuweisung (bisheriger Wert wird überschrieben)
<code>\addtocounter{name}{wert}</code>	Addition eines Werts zum aktuellen Zählerwert
<code>\stepcounter{name}</code>	Erhöhung des Zählerwerts um 1
<code>\roman{name}</code>	Umschalten des Zählers auf römische Ziffern
<code>\arabic{name}</code>	Umschalten des Zählers auf arabische Ziffern
<code>\alph{name}</code>	Umschalten des Zählers auf Kleinbuchstaben
<code>\Alph{name}</code>	Umschalten des Zählers auf Großbuchstaben

Beispielsweise kann die `\setcounter{}`-Anweisung genutzt werden, um die Seitennummerierung im laufenden Dokument zu ändern (beispielsweise wieder bei 1 beginnen zu lassen).

Makros definieren

Mit `\newcommand{}` kann man in LaTeX eigene Anweisungen („Makros“) definieren. Die allgemeine Syntax hierzu lautet:

```
\newcommand{\Anweisungsname} [Parameter-Anzahl] [Voreinstellungen] {Inhalt}
```

Der Anweisungsname sowie der eigentliche Inhalt sind Pflichtangaben. Kommt die selbst definierte Anweisung ohne weitere Parameter aus, so können die übrigen Angaben weggelassen werden:

Beispiel:

```
% Neue Anweisungen definieren:

\newcommand{\datum}{1. April}
\newcommand{\scherz}{Scherz für den \datum}
\newcommand{\lx}{\fbox{\LaTeX}}

% Anweisung nutzen:

Das ist kein \scherz!

Hallo \lx
```

Ergebnis:

Das ist kein Scherz für den 1. April!

Hallo L^AT_EX

Gibt man bei der Definition eines Makros eine bestimmte Anzahl (maximal 9) an Parametern an, so muss auch beim Aufruf des Makros genau diese Anzahl an Parametern angegeben werden. Innerhalb der Makro-Definition werden die Parameter mit #1, #2 usw. bezeichnet:

Beispiel:

```
% In der Präambel:
% \usepackage{shadow}

% Neue Anweisung mit _einem_ Parameter definieren:
\newcommand{\shac}[1]{\shabox{ \centering \textsc{ #1 } }}

% Anweisung nutzen:
\shac{Hallo Welt!}
```

Ergebnis:

HALLO WELT!

Möchte man ein Makro definieren, das neben einem Pflicht-Parameter auch einen optionalen Parameter hat, so kann man für diesen einen Standard-Wert festlegen. Innerhalb der Makro-Definition wird der optionale Parameter automatisch mittels #1 bezeichnet.

Beispiel:

```
% In der Präambel:
% \usepackage{shadow}

% Neue Anweisung mit Pflicht- und optionalem Parameter definieren:
\newcommand{\shaC}[2] [5cm]{ \shabox{\parbox{#1}{ \centering \textsc{ #2 } }} }
```

(continues on next page)

```
% Anweisung nutzen:  
\shaC{Hallo Welt!} \\  
\shaC[10cm]{Hallo Welt!}
```

Ergebnis:

HALLO WELT!

HALLO WELT!

Sollte eine mathematische Formel innerhalb der Makro-Definition vorkommen, so sollte diese unbedingt mit der Anweisung `\ensuremath{Formel}` umschlossen werden. Hierdurch wird sichergestellt, dass das Makro korrekt abläuft, unabhängig davon, ob das Makro im Mathe-Modus aufgerufen wird oder nicht.

... to be continued ...

Weitere Dokumentklassen

LaTeX kann nicht nur zum Schreiben von Artikeln und Büchern verwendet werden – es gibt einige weitere Dokumentklassen, mit denen auch andere Arten von Textdokumenten erstellt werden können.

scrlettr2 – Briefe und Serienbriefe

Die Dokumentklasse `scrlettr2` ermöglicht es, auf einfache Weise Briefe mit gewohnt eleganten LaTeX-Layout zu verfassen. Wie bei anderen Dokumentklassen können bereits in der `\documentclass{}`-Anweisung optionale Layout-Angaben festgelegt werden:

```
\documentclass[a4paper, 12pt, halfparskip]{scrlettr2}
```

Für die typischen `scrlettr2`-Optionen empfiehlt es sich der Übersichtlichkeit halber, diese separat mittels der `\KOMAOptions{}`-Anweisung festzulegen:

```
\KOMAOptions{
  backaddress,           % Adresse des Absenders im Adressfenster anzeigen
  foldmarks=on,         % Faltungs-Markierungen drucken
  fromalign=left,       % Ausrichtung des Briefkopfes (Absender)
  fromphone,            % Telefonnummer anzeigen
  fromemail,            % Emailadresse des Absenders anzeigen
  fromurl,              % Verweis auf Webseite
  % fromlogo,           % Logo drucken
  % fromfax,            % Faxnummer des Absenders angeben
}
```

Auf diese Weise können einzelne Angaben einfach ausgeblendet werden, indem zu Beginn der jeweiligen Zeile das Kommentarzeichen `%` gesetzt wird. Die einzelnen Angaben können dann folgendermaßen festgelegt werden:

```
\setkomavar{fromname}{Vorname Nachname}
\setkomavar{fromurl}{http://webadresse.de}
\setkomavar{fromaddress}{Straßenname 1 \\\ 123456 Ortsname}
\setkomavar{fromphone}{01234- 56\,78\,90}
\setkomavar{fromemail}{vorname.nachname@email.de}

\setkomavar{backaddressseparator}{
  ~\raisebox{0.25ex}{\textbf{\Large.}}~
}

\setkomafont{backaddress}{\sffamily\tiny}
\setkomafont{fromaddress}{\rmfamily\footnotesize}
```

Neben diesen Festlegungen können in der Präambel ebenso wie bei anderen Dokumenten weitere Zusatzpakete mittels `\usepackage{}` geladen werden. Das eigentliche Textdokument wird dann wie üblich durch die `document`-Umgebung begrenzt.

Da einem Brief zusätzliche Inhalte als Anhang beigefügt sein können, wird das eigentliche Anschreiben mittels der innerhalb der `scrlettr2`-Dokumentklasse definierten `letter`-Umgebung vorgenommen:

```

\begin{letter}{
  Name des Empfängers \\
  Anschrift des Empfängers \\
  PLZ Ort
}

\opening{Sehr geehrte Person(en),}

... Briefftext ...

\closing{Mit freundlichen Grüßen,}

\end{letter}

```

Am Ende der letter-Umgebung kann optional mittels der Anweisung `\ps{}` ein Nachsatz („Post Scriptum“) und/oder mittels der `\encl{}`-Anweisung eine Aufzählung der Anhänge („Enclosures“) angefügt werden.¹

Ein kompletter Brief kann somit als Minimalbeispiel folgendermaßen aussehen: `testbrief.tex`

```

\documentclass[a4paper, 12pt, halfparskip]{scrlettr2}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}
\usepackage{graphicx, blindtext}

\KOMAOptions{
  backaddress,           % Adresse des Absenders im Adressfenster anzeigen
  foldmarks=on,         % Faltungs-Markierungen drucken
  fromalign=left,       % Ausrichtung des Briefkopfes (Absender)
  % fromfax,             % Faxnummer des Absenders angeben
  fromemail,            % Emailadresse des Absenders anzeigen
  fromphone,            % Telefonnummer anzeigen
  % fromlogo,           % Logo drucken
  fromurl,              % Verweis auf Webseite
}

\setkomavar{fromname}{Vorname Nachname}
\setkomavar{fromurl}[Web:\quad]{http://webadresse.de}
\setkomavar{fromaddress}{Straßenname 1 \\ 123456 Ortsname}
\setkomavar{fromphone}{01234- 56\,78\,90}
\setkomavar{fromemail}{vorname.nachname@email.de}

\setkomavar{backaddressseparator}{
  ~\raisebox{0.25ex}{\textbf{\Large.}}~
}

\setkomafont{backaddress}{\sffamily\tiny}
\setkomafont{fromaddress}{\rmfamily\footnotesize}

\begin{document}

\begin{letter}{
  Name des Empfängers \\
  Anschrift des Empfängers \\
  PLZ Ort
}

\opening{Sehr geehrte Person(en),}

```

(continues on next page)

¹ Mit `\setkomavar*{enclseparator}{Anlage}` kann die als Standard definierte Kennzeichnung „Anlage(n):“ durch den angegebenen Text ersetzt werden.


```
\blindtext % Das eigentliche Anschreiben

\closing{Mit freundlichen Grüßen,}

\ps{P.S.: Hier noch ein kurzer Nachsatz, ausschließlich zu Demonstrationszwecken.}
\encl{
  Anhang 1 \\
  Anhang 2
}

\end{letter}
\end{document}
```

Dieses Codebeispiel liefert folgendes Ergebnis: `testbrief.pdf`²

² In Sphinx ist es für die HTML-Version leider nicht möglich, „normalen“ LaTeX-Code als Umgebung zu nutzen, um die jeweiligen Ergebnisse zu rendern; es können nur die Ausgaben des LaTeX-Mathe-Modus genutzt werden, um Formeln innerhalb einer Zeile oder als eigenen Absatz einzubinden.

Im obigen Beispiel musste ich daher die erstellte PDF-Datei in eine PNG-Graphik umwandeln, um diese als normale Abbildung einbinden zu können. Ein Screenshot wäre zwar auch möglich gewesen; ich habe mich für eine bessere Auflösung jedoch lieber für eine Konvertierung der PDF-Datei in eine PNG-Graphik mittels `Imagemagick` entschieden:

```
convert -verbose -density 150 testbrief.pdf -quality 100 -sharpen 0x1.0 -border 2x2 testbrief.png
```

Hierbei wird eine Auflösung von 150 dpi festgelegt; die Ausgabe-Qualität soll bestmöglich sein (100%), zudem wird durch die `sharpen`-Option ein pixeliges Aussehen des Ergebnisses weitgehend vermieden. Mit der Option `border` wird schließlich noch ein Rand mit jeweils 2 Pixeln Abstand erzeugt.

Vorname Nachname
Straßenname 1
123456 Ortsname
Telefon: 01234- 56 78 90
E-Mail: vorname.nachname@email.de
Web: http://webadresse.de

Vorname Nachname • Straßenname 1 • 123456 Ortsname

Name des Empfängers
Anschrift des Empfängers
PLZ Ort

13. Mai 2016

Sehr geehrte Person(en),

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Mit freundlichen Grüßen,

Vorname Nachname

P.S.: Hier noch ein kurzer Nachsatz, ausschließlich zu Demonstrationszwecken.

Anlage(n): Anhang 1
Anhang 2

Werden keine weiteren Layout-Vorgaben getroffen, so werden die Briefseiten bei Verwendung von `srlttr2` ab der zweiten Seite automatisch durchnummeriert.³

³ Ich habe die Erfahrung gemacht, dass die erste Seite der Briefe oftmals schneller beendet wird, als ich dies erwartet hätte (trotz „nur“ 2,5 cm unterem Seitenrand); dies liegt wohl daran, dass die Briefklasse `srlttr2` auch dann Platz für eine Fußzeile freihält, auch wenn ein solcher Inhalt gar nicht angegeben wird. Man kann an dieser Stelle mittels der Anweisung `\enlargethispage{3cm}` im Verlauf der ersten Seite etwas tricksen.

Beispielsweise sollte bei Briefen, die nur eine Seite lang sein sollen, auch das Brief-Ende noch auf dieser Seite stehen; ist keine Fußzeile explizit angegeben, so sieht das Layout durch die manuelle Dehnung der Seite mittels `\enlargethispage{}` auch nicht gequetscht aus. Bei zwei- oder mehrseitigen (Geschäfts-)Briefen mit expliziter Fußzeile kann die erste Seite hingegen bei Bedarf auch schon früher manuell mittels `\newpage` beendet werden, um ein harmonisches Gesamtbild zu erhalten.

Design der ersten Seite gestalten

Bei offiziellen Anschreiben weicht oftmals das Layout der ersten Seite von dem der folgenden Seiten ab. Beispielsweise erscheint nur auf der ersten Seite der Briefkopf, oftmals mit Briefkopf und Logo, bei geschäftlichen Briefen zusätzlich mit Kunden- oder Rechnungsnummer; in der Fußzeile finden sich auf der ersten Seite eines Briefes zudem oftmals Bankverbindungs- und weitere Kontaktdaten.

Offizielle Briefe enthalten häufig zusätzlich zu den oben genannten Angaben folgende Variablen:

```
\setkomavar{fromurl}[Web:\quad]{http://webadresse.de}
\setkomavar{fromlogo}{\includegraphics{testlogo.png}}
\setkomavar{location}{\raggedright Mitglied Nr.~4711\\
                      seit dem 11.09.2001\\
                      Vorsitzender in den Jahren 2003--2005}

\setkomavar{frombank}{
  IBAN: DE\,0000\,0000\,0000\,0000\,0000 \\
  BIC: XXXXXXXXXXXX \\
  Name der Bank
}
```

... to be continued ...

Links

Tutorials:

- [LaTeX-Tutorial für Einsteiger \(de, PDF\)](#)
- [LaTeX-Tutorial für Fortgeschrittene \(de, PDF\)](#)
- [LaTeX-Skript von Martin Schimmels \(PDF\)](#)
- [LaTeX-Grundlagen von Daniel Hommel \(PDF\)](#)
- [LaTeX-Tutorial von Sebastian Stein](#)
- [Einführung in LaTeX von Sascha Frank](#)
- [LaTeX-Einführung 1 \(Fernuni Hagen\)](#)
- [LaTeX-Einführung 2 \(Fernuni Hagen\)](#)
- [LaTeX 2e Kurzbeschreibung](#)
- [LaTeX-Linksammlung](#)
- [LaTeX for Complete Novices \(en\)](#)
- [Getting Started with LaTeX \(en\)](#)
- [The Not So Short Introduction to LaTeX \(en\)](#)

Dokumentationen:

- [KOMA-Script Anleitung \(PDF\)](#)
- [LaTeX Documentation \(en.\)](#)
- [LaTeX-Wikibook \(en.\)](#)

Referenzen:

- [Latex-Befehlsübersicht](#)
- [LaTeX-Wörterbuch](#)
- [Mathematische Symbole \(de\)](#)
- [Mathematische Symbole \(en, 1\)](#)
- [Mathematische Symbole \(en, 2\)](#)
- [Mathematische Symbole \(en, PDF\)](#)
- [LaTeX CheatSheet \(en\)](#)

Tips:

- [LaTeX Stack Exchange](#)
- [TeX-Welt Fragen und Antworten](#)
- [LaTeX-Codeschnipsel \(Uwe Ziegenhagen\)](#)
- [Briefe mit „scrlltr2“](#)

Typographie:

- [Practical Typography \(en\)](#)

Stichwortverzeichnis

Symbols

- `\,` 9
- `\appendix`, 8
- `\arraystretch`{}, 37
- `\chapter`{}, 7
- `\color`{}, 11
- `\documentclass`{}, 2
- `\emph`{}, 10
- `\fontsize`{}, 13
- `\footnote`{}, 16
- `\framebox`{}, 17
- `\href`{}, 43
- `\hspace`{}, 15
- `\hyperlink`{}, 44
- `\hypersetup`{}, 44
- `\hypertarget`{}, 44
- `\hyphenation`{}, 15
- `\includegraphics`{}, 33
- `\int`, 29
- `\label`{}, 16, 43
- `\listoffigures`, 8
- `\listoftables`, 8
- `\makebox`{}, 17
- `\marginpar`{}, 16
- `\mathbb`{}, 24
- `\mathbf`{}, 23
- `\mathcal`{}, 24
- `\mathfrak`{}, 24
- `\mathit`{}, 23
- `\mathnormal`{}, 23
- `\mathrm`{}, 23
- `\mathscr`{}, 24
- `\mathsf`{}, 23
- `\mathtt`{}, 24
- `\multicolumn`{}, 46
- `\multirow`{}, 47
- `\necommand`{}, 56
- `\newgeometry`{}, 42
- `\newline`, 9
- `\newpage`, 9
- `\pagebreak`, 9
- `\pageref`{}, 16, 43
- `\paragraph`{}, 7
- `\parbox`{}, 18
- `\part`{}, 7
- `\phantom`{}, 16
- `\ref`{}, 16, 43
- `\rotatebox`{}, 12
- `\rule`{}, 19
- `\section`{}, 7
- `\subparagraph`{}, 7
- `\subsection`{}, 7
- `\subsubsection`{}, 7
- `\sum`, 29
- `\tableofcontents`, 8
- `\textbf`{}, 10
- `\textit`{}, 10
- `\textrm`{}, 10
- `\textsc`{}, 10
- `\textsf`{}, 10
- `\textsl`{}, 10
- `\texttt`{}, 10
- `\underline`{}, 11
- `\url`{}, 43
- `\usepackage`{}, 3

A

- Abbildung, 33
- Abschnitt, 7
- Abstand, 15
 - horizontal, 15
 - vertikal, 15
- Abstract, 7
- Anhang, 8
- array (Paket), 40
- array (Umgebung), 31
- Aufzählung
 - nicht nummeriert, 32
 - nummeriert, 32

B

- Blindtext, 40
- blindtext (Paket), 40
- Box, 16
 - LR-Box, 17
 - Parbox, 18
 - Rule-Box, 19
- Briefe, 59

C

- caption (Paket), 41
- center (Umgebung), 32
- cmap (Paket), 41
- colortbl (Paket), 41
- Copy-and-Paste, 41
- Counter, 56
- CTAN, 1

D

description (Umgebung), 33

E

Einheiten, 47

em (Einheit), 13

enumerate (Umgebung), 32

ex (Einheit), 13

Exponent, 25

F

Farbiger Text, 11

figure (Umgebung), 33

Fließumgebung, 33

flushleft (Umgebung), 32

flushright (Umgebung), 32

Formel, 20

Fußnote, 16

Fußzeile, 49

G

geometry (Paket), 42

Gleitobjekt, 33

Graphik, 33

Griechische Buchstaben, 14

Griechisches Alphabet, 23

H

Hilfsdatei, 4

Horizontale Abstände, 15

hyperref (Paket), 42

|

Index, 25

Inhaltsverzeichnis, 8

Integralzeichen, 29

itemize (Umgebung), 32

K

Kapitälchen, 10

Kapitel, 7

Klammern, 26

Kommentar, 2

Kopfzeile, 49

L

Längenmaß, 13

Label, 16

landscape (Umgebung), 46

Link, 42

 Datei, 43

 Email, 43

 Label, 43

 Webadresse, 43

longtable (Paket), 44

lscape (Paket), 46

M

Makro, 56

Mathematische Ausdrücke, 25

 Binomialkoeffizienten, 28

 Brüche, 28

 Determinanten, 30

 Fallunterscheidungen, 30

 Matrizen, 30

 Vektoren, 30

 Wurzeln, 28

Mathematische Formel, 20

Mathematische Schriftarten, 23

Mathematische Symbole, 22

Mengensymbol, 24

Meta-Tags, 44

Minipage, 19

mktexlsr, 1

multicol (Paket), 46

multirow (Paket), 47

P

Paket, 1

paralist (Paket), 47

Part, 7

pdflatex, 4

Pfeile, 27

Phantom-Text, 16

Präambel, 3

pt (Einheit), 13

Q

Querformat, 46

Querverweise, 16

quotation (Umgebung), 39

quote (Umgebung), 39

R

Randnotiz, 16

Referenz, 16

Relationszeichen, 24

S

Schriftart, 10

Schriftfamilie, 10

Schriftgröße, 12

scrlltr2 (Dokumentklasse), 59

Seitenränder, 42

Seitenrand, 52

Seitenumbruch, 9

Serifen, 10

Silbentrennung, 14

spacing (Umgebung), 15

Sprachunterstützung, 4

Steuerzeichen, 13

Summenzeichen, 29

T

Tabbing, 36

Tabelle, 36
 mehrseitig, 44
 Spaltenoptionen, 36, 40
table (Umgebung), 38
tabular (Umgebung), 36
texhash, 1
Texthervorhebung, 10
Titelseite, 7

U

Umgebung, 31
 Fließumgebung, 33
Umlaute, 13
units (Paket), 47
Unterstreichung, 11
url (Paket), 48
URLs, 48

V

Variablen, 56
verb (Umgebung), 38
verbatim (Umgebung), 38
Verlinkung, 42
verse (Umgebung), 39
Vertikale Abstände, 15

W

Weblink, 43

Z

Zähler, 56
Zeilenabstand, 15