

Tips und Tricks für  
***LaTeX***

Dipl. Inf. Meik Teßmer  
Angewandte Informatik/Wirtschaftsinformatik  
Fakultät für Wirtschaftswissenschaften  
Universität Bielefeld

## Was ist L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X ist ein sog. *Satzsystem*, mit dem Texte *gesetzt* werden können. Im Vergleich zu Textverarbeitungsprogrammen wie OpenOffice Writer werden Formatierungen wie Fettschrift oder Schriftart als expliziter Befehl angegeben.

Um ein Dokument zu erzeugen, muss die erstellte Quelldatei zuerst durch L<sup>A</sup>T<sub>E</sub>X verarbeitet werden:

```
latex mein_dokument.tex
```

bzw. für die direkte Erzeugung von PDF-Dateien:

```
pdflatex mein_dokument.tex
```

Erst dann kann die erzeugte Datei (`mein_dokument.dvi` bzw. `mein_dokument.pdf`) betrachtet werden.

## Aufbau einer Quelldatei — 1

Eine minimale Quelldatei sieht folgendermaßen aus:

```
\documentclass{article}  
\begin{document}  
Hallo Welt!  
\end{document}
```

Das Grundlayout eines Dokuments wird durch seine *Klasse* festgelegt. Die gebräuchlichsten Klassen sind:

**article** — Artikel für wissenschaftl. Zeitschriften, kürzere Berichte usw.

**book** — Klasse für das Setzen von Büchern

**report** — Klasse für längere Berichte, die mehrere Kapitel beinhalten

## Aufbau einer Quelldatei — 2

Zusätzlich zur Klasse können sog. *Klassenoptionen* angegeben werden:

`10pt|11pt|12pt` — Schriftgröße. L<sup>A</sup>T<sub>E</sub>X verwendet normalerweise eine Schrift der Größe 10pt.

`a4paper` — Seitengröße. Wird keine Seitengröße angegeben, wird die amerikanische `letter`-Größe verwendet.

`titlepage|notitlepage` — Titelseite und Zusammenfassung können getrennt oder gemeinsam auf einer Seite erscheinen.

`oneside|twoside` — Einseitiger oder zweiseitiger Druck.

`landscape` — Seitenorientierung wird auf quer gestellt.

## Aufbau einer Quelldatei — 3

*Pakete* können zusätzliche Befehle bereitstellen, das Layout verändern usw. Sie werden üblicherweise direkt nach dem `\documentclass`-Befehl angefügt.

Im deutschsprachigen Raum ist bspw. die direkte Eingabe von Umlauten über die Tastatur üblich. Dazu bindet man das Paket `inputenc` ein:

```
\usepackage[latin1]{inputenc}
```

Für die deutsche Silbentrennung nach neuer Rechtschreibung nimmt man `ngerman` (für die alte Rechtschreibung `german`):

```
\usepackage{ngerman}
```

Weitere nützliche Pakete sind `pdftex`, `graphicx` und `fontenc`. Die Verfügbarkeit von Paketen ist allerdings abhängig von der jeweiligen  $\text{\TeX}$ -Installation.

## Aufbau einer Quelldatei — 4

Die minimale Quelldatei sieht mit den bisherigen Erweiterungen nun so aus:

```
\documentclass[a4paper]{article}  
\usepackage[latin1]{inputenc}  
\usepackage{ngerman}
```

```
\begin{document}  
Hallo Welt! Wir können jetzt auch Umlaute benutzen.  
\end{document}
```

## Absätze und Einrückungen

Normaler Text wird im Blocksatz gesetzt. Eine oder mehrere Leerzeilen führen zu einem neuen Absatz.

Absätze werden durch eine zusätzliche Einrückung gekennzeichnet. Ist dies nicht erwünscht, kann *vor* dem Befehl `\begin{document}` die Einrücktiefe auf 0mm gestellt werden:

```
\setlength{\parindent}{0mm}
```

```
...
```

```
\begin{document}
```

```
...
```

# Leerzeichen und Kommentare

## Leerzeichen

„Unsichtbare“ Zeichen wie das Leerzeichen, Tabulatoren und das Zeilenende werden von L<sup>A</sup>T<sub>E</sub>X einheitlich als Leerzeichen behandelt. Mehrere aufeinanderfolgende Leerzeichen werden dabei als *ein* Leerzeichen verstanden.

## Kommentare

Das Prozentzeichen % dient als Kommentarbefehl. Alles, was sich bis zum Zeilenende hinter einem Prozentzeichen befindet, wird von L<sup>A</sup>T<sub>E</sub>X als Kommentar interpretiert und erscheint nicht im fertigen Dokument:

```
% Dies ist eine Kommentarzeile.
```



## Silbentrennung

L<sup>A</sup>T<sub>E</sub>X versucht immer, automatisch die passenden Silbentrennungen zu finden. Sollte dies einmal nicht gelingen, kann man an den entsprechenden Stellen manuell nachhelfen.

Der Befehl `\-` innerhalb eines Wortes bewirkt, dass dieses Wort dieses eine Mal nur an den mit `\-` markierten Stellen getrennt werden darf:

`Ein\-gabe\-datei`

## Kapitel und Unterkapitel

Ein Kapitel wird mit dem Befehl `\section{...}` erzeugt, ein Unterkapitel mit `\subsection{...}` und ein „Unter“-Unterkapitel mit `\subsubsection{...}`. Die Dokumentklassen `book` und `report` erlauben als höchste Gliederungsstufe den Befehl `\chapter{...}`, gefolgt von `\section{...}` usw.

Der Befehl `\tableofcontents` erzeugt an der Stelle des Auftretens automatisch ein Inhaltsverzeichnis.

# Fußnoten

Fußnoten können mit dem Befehl `\footnote{...}` erzeugt werden.

Beispiel:

Hier<sup>a</sup> wird eine Fußnote gesetzt.

Fußnoten werden, abhängig von der Dokumentklasse, automatisch durchnummeriert.

---

<sup>a</sup>Hier? Ja wo denn sonst?

## Listen — 1

### Einfache Listen

Einfache Listen werden mit einer `itemize`-Umgebung erzeugt:

```
\begin{itemize}  
\item Erster Eintrag  
\item Zweiter Eintrag  
\item Dritter Eintrag  
\end{itemize}
```

- Erster Eintrag
- Zweiter Eintrag
- Dritter Eintrag

# Listen — 2

## Nummerierte Listen

Nummerierte Listen werden mittels `enumerate` gesetzt:

```
\begin{enumerate}  
\item Erster Eintrag  
\item Zweiter Eintrag  
\item Dritter Eintrag  
\end{enumerate}
```

1. Erster Eintrag
2. Zweiter Eintrag
3. Dritter Eintrag

## Listen — 3

### Beschreibungslisten

Beschreibungslisten werden mit der `description`-Umgebung gesetzt:

```
\begin{description}
```

```
\item[Computer:] Große, unförmige bis  
kleine elegante Kisten, die außer  
Krach und Elektromog nur bedingt  
viel Nutzen haben.
```

```
\item[Computernutzer:] Meist ebenso  
schlau wie die Kiste, vor der er  
sitzt.
```

```
\end{description}
```

**Computer:** Große, unförmige bis kleine elegante Kisten, die außer Krach und Elektromog nur bedingt viel Nutzen haben.

**Computernutzer:** Meist ebenso schlau wie die Kiste, vor der er sitzt.

Alle beschriebenen Listen können auch verschachtelt werden, d. h. in einer `itemize`-Liste kann eine `enumerate`-Liste oder auch eine zweite `itemize`-Liste eingebettet werden. Die Listenzeichen werden automatisch angepasst.

# Tabellen — 1

Tabellen werden mit Hilfe der `tabular`-Umgebung gesetzt. Dabei wird die Tabellenbreite von  $\text{\LaTeX}$  automatisch ermittelt.

```
\begin{tabular}{|r|l|}  
\hline  
erste Spalte & zweite Spalte\\  
\hline  
neue Zeile & zweite Spalte\\  
\hline\hline  
abgesetzte Zeile & weitere Spalte\\  
\hline  
\end{tabular}
```

ergibt

erste Spalte	zweite Spalte
neue Zeile	zweite Spalte
abgesetzte Zeile	weitere Spalte

## Tabellen — 2

Die Anzahl und Ausrichtung der Spalten einer Tabelle werden in geschweiften Klammern hinter dem Umgebungsbeginn definiert.

`r` — Spalteninhalt rechtsbündig ausrichten

`l` — Spalteninhalt linksbündig ausrichten

`c` — Spalteninhalt zentrieren

`p{breite}` — Spalte wird ab einer Breite von *breite* automatisch umgebrochen.

Der Befehl `\hline` definiert eine horizontale Linie. Die Inhalte einer Zeile werden durch `&` auf die Spalten aufgeteilt. Jede Zeile der Tabelle muss dabei mit `\\` abgeschlossen werden.



## Positionierung von Tabellen — 1

L<sup>A</sup>T<sub>E</sub>X versucht immer, eine passende Aufteilung von Text und Tabelle zu finden. Dadurch kann es passieren, dass im fertigen Dokument der zu einer Tabelle gehörende Fließtext nicht immer direkt vor oder hinter derselben zu finden ist. Mit der `table`-Umgebung kann auf die Positionierung ein gewisser Einfluss ausgeübt werden. Dazu werden hinter dem Umgebungsbeginn in eckigen Klammern Positionsparameter definiert:

```
\begin{table}[t]
\begin{tabular}{l|l}
\hline
Hallo & Welt\\
\hline
\end{tabular}
\caption{Beispiel-Tabelle.}
\end{table}
```

## Positionierung von Tabellen — 2

Die Parameter haben folgende Bedeutung:

**h** — here, „hier“

**t** — top; Seitenbeginn

**b** — bottom; Seitenende

**p** — page; eigene Seite

Werden keine Parameter (und damit keine eckigen Klammern) angegeben, wird automatisch `[tbp]` angenommen.  $\text{\LaTeX}$  schiebt eine Tabelle solange nach hinten, bis sie in den Text passt.

Der Befehl `\caption{...}` definiert eine Tabellenunterschrift, die zudem automatisch nummeriert wird. Mit `\listoftables` kann schließlich ein Tabellenverzeichnis ausgegeben werden.

## Zitate

Um längere Zitate kenntlich zu machen, fasst man sie in eine `quote`-Umgebung ein:

```
\begin{quote}  
\emph{Zwei Dinge sind unendlich: das Universum und die menschliche  
Dummheit. Beim Universum bin ich mir aber noch nicht ganz  
sicher.}
```

```
\hfill Albert Einstein  
\end{quote}
```

So sieht das Zitat dann aus:

*Zwei Dinge sind unendlich: Das Universum und die menschliche  
Dummheit. Beim Universum bin ich mir aber noch nicht ganz sicher.*

Albert Einstein

# Sonderzeichen und Verwandtes — 1

## Befehlszeichen

Um die Zeichen # \$ & ~ \_ ^ % { } (üblicherweise L<sup>A</sup>T<sub>E</sub>X-Befehle) verwenden zu können, muss ein Backslash (\) vorangestellt werden:

`\# = #`   `\$ = $`   `\& = &`   `\~ = ~`   `\_ = _`   `\^ = ^`   `\% = %`   `\{ = {`   `\} = }`

## Sonderzeichen

Folgende Sonderzeichen sind nicht auf der Tastatur zu finden:

`\S = §`   `\dag = †`   `\ddag = ‡`   `\P = ¶`   `\copyright = ©`   `\pounds = £`

Sonderbuchstaben, wie sie in europäischen Sprachen zu finden sind, stellt L<sup>A</sup>T<sub>E</sub>X ebenfalls zur Verfügung:

`{\OE} = Œ`   `{\AE} = Æ`   `{\AA} = Å`   `{\O} = Ø`   `{\SS} = Š`   `{\L} = Ł`   `! ' = ¡`  
`{\oe} = œ`   `{\ae} = æ`   `{\aa} = å`   `{\o} = ø`   `{\ss} = š`   `{\l} = ł`   `? ' = ¿`

## Sonderzeichen und Verwandtes — 2

### Akzente

Die europäischen Sprachen kennen eine Vielzahl von Akzenten:

$\backslash\prime{o} = \grave{o}$     $\backslash\prime{o} = \acute{o}$     $\backslash\hat{o} = \hat{o}$     $\backslash"o = \ddot{o}$     $\backslash\sim{o} = \tilde{o}$     $\backslash={o} = \bar{o}$     $\backslash\cdot{o} = \dot{o}$   
 $\backslash u{o} = \ddot{o}$     $\backslash v{o} = \ddot{o}$     $\backslash H{o} = \ddot{o}$     $\backslash t{oo} = \hat{o}o$     $\backslash c{o} = \text{ç}$     $\backslash d{o} = \text{ð}$     $\backslash b{o} = \underline{o}$   
 $\backslash r{o} = \ring{o}$

Die Akzente der ersten Zeile können auch vereinfacht geschrieben werden:

$\backslash\prime{o} = \grave{o}$     $\backslash\prime{o} = \acute{o}$     $\backslash\hat{o} = \hat{o}$     $\backslash"o = \ddot{o}$     $\backslash\sim{o} = \tilde{o}$     $\backslash={o} = \bar{o}$     $\backslash\cdot{o} = \dot{o}$

### Trennstriche

Trennstriche gibt es in drei Ausführungen:

- : -, normaler Bindestrich, z. B. Trennstrich-Ausführung
- : --, alte Form von Streckenangaben, bspw. Hamburg–Berlin. Nach heutigem Duden jedoch durch den normalen Bindestrich ersetzt.
- : ---, Gedankenstriche — wie an dieser Stelle — werden so erzeugt.

# Sonderzeichen und Verwandtes — 3

## Anführungsstriche

Ist `ngerman.sty` (oder das ältere `german.sty`) aktiv, können die deutschen Anführungszeichen „ und “ benutzt werden:

$$\text{“ ‘} = \text{„} \qquad \text{” ’} = \text{“}$$

# Leerräume — 1

## Leerzeichen

Hinter einem Punkt wird etwas mehr Abstand gelassen als zwischen zwei Wörtern.  
Bei Abkürzungen wie *Dr.* oder *z. B.* kann man man dies vermeiden, indem statt  
eines einfachen Leerzeichens ein `\,` gesetzt wird:

`Dr. No` = `Dr. No`  
`Dr.\,No` = `Dr. No`

## Horizontale Abstände

Mittels `\hfill` können flexible horizontale Abstände erzeugt werden:

`Wort 1\hfill Wort 2`

Wort 1

Wort 2

Leerräume — 2

Eine mehrfache Angabe führt zu einer entsprechend gewichteten Aufteilung der Leerräume. Im folgenden Beispiel ist daher der Abstand zwischen den Wörtern größer als vor dem ersten Wort:

```
\hfill Wort 1 \hfill\hfill Wort 2
```

Wort 1Wort 2

Anstelle von Leerzeichen können auch Punkte oder Unterstriche verwendet werden:

```
\dotfill Wort 1 \hrulefill Wort 2
```

.....Wort 1 \_\_\_\_\_Wort 2

Für kleinere Abstände eignen sich die Befehle `\quad` bzw. `\qquad`. Ersterer erzeugt einen zeichenbreiten Abstand, der zweite einen zwei Zeichen breiten Abstand:

Wort 1   Wort 2

Wort 1    Wort 2



Leerräume — 3

Vertikale Abstände

Um bei einem erzwungenen Zeilenumbruch (`\``\`) einen anderen Abstand als den normalen zu erhalten, kann über einen in `[ ]` eingefassten Wert der gewünschte Abstand angegeben werden:

```
Zeile\dotfill\[1cm]
nächste Zeile\dotfill

Zeile.....

nächste Zeile.....
```

Ähnlich wie flexible horizontale Abstände sind auch vertikale Abstände möglich; der Befehl lautet `\vfill`. Allerdings sind keine Punkte oder Unterstriche möglich (und wohl auch nicht sinnvoll).

## Schriftformatierung — 1

### Allgemeine Formatierung

Hervorhebungen werden in L<sup>A</sup>T<sub>E</sub>X mittels `\emph{...}` realisiert. Je nach gerade verwendeter Schriftfamilie, -form und -serie wird automatisch die richtige Schrift gewählt:

Dies ist ein *wichtiger* Text.

Dies ist ein *wichtiger* Text.

*Dies ist ein wichtiger Text.*

Daneben kann auch direkt Einfluss auf die Schriftart genommen werden. L<sup>A</sup>T<sub>E</sub>X kennzeichnet seine Zeichensätze durch sog. *Attribute*. Auch die Schriftgröße ist ein solches Attribut.

## Schriftformatierung — 2

### Manuelle Formatierung

**Familie** Das globale Erscheinungsbild der Schrift, meist auch durch ihre Herkunft gekennzeichnet (Times New Roman, Baskerville usw.). In einer Standardinstallation sind folgende Familien vorhanden:

`\rmfamily` Umschaltung auf eine Roman-Schrift

`\ttfamily` Umschaltung auf eine Schreibmaschinenschrift

`\sffamily` Umschaltung auf eine serifenlose Schrift

**Form** Neigung der Schrift oder Kapitälchen-Form:

`\itshape` Umschaltung auf eine *Kurivschrift*

`\slshape` Umschaltung auf eine *geneigte Schrift*

`\scshape` Umschaltung auf eine KAPITÄLCHEN-SCHRIFT

`\upshape` Umschaltung auf eine aufrechte Schrift

## Schriftformatierung — 3

**Serie** Mit diese Attribut wird die Stärke und Weite einer Schrift bestimmt:

`\bfseries` Umschaltung auf eine **Fettschrift**

`\mdseries` Umschaltung auf eine normale Schrift

Für kurze Umschaltungen stehen entsprechende Befehle mit Argumenten bereit:

**Familie** `\textrm{...}`, `\texttt{...}`, `\textsf{...}`

**Form** `\textit{...}`, `\textsl{...}`, `\textsc{...}`, `\textup{...}`

**Serie** `\textbf{...}`, `\textmd{...}`

**Standard** `\textnormal{...}`

Schriftformatierung — 4

Schriftgröße

Anders als in MS Word oder vergleichbaren Produkten erfolgt die Schriftgröße nicht über eine Punkt-Angabe, sondern durch entsprechende Befehle:

<code>\tiny</code>	winzig	<code>\Large</code>	Größer
<code>\scriptsize</code>	sehr klein		
<code>\footnotesize</code>	Fußnote	<code>\LARGE</code>	Noch größer
<code>\small</code>	klein		
<code>\normalsize</code>	normal	<code>\huge</code>	riesig
<code>\large</code>	groß	<code>\Huge</code>	Gigantisch

## Setzen von Bildern — 1

### Das `graphicx`-Paket

Das Paket `graphicx` erlaubt die Einbindung von Bildern verschiedenster Formate:

```
\usepackage[pdftex]{graphicx}  
...  
\includegraphics{image.jpg}
```

Lädt man das Paket mit der Option `[draft]`, so erscheint an Stelle des Bilds nur ein Rahmen entsprechender Größe. Auf diese Weise kann man große Dateien schneller betrachten (das Bild muss nicht aufgebaut werden) und bei Probedrucken Tinte bzw. Toner sparen.

### Historie

Ursprünglich wurden keine Formate wie PostScript/EPS, JPEG oder GIF unterstützt, jedoch können die DVI-Dateien `\special`-Befehle enthalten, die zum Betrachter-Programm weitergeleitet werden. Damit wurde die Formatvielfalt nur noch durch den DVI-Betrachter beschränkt.

## Setzen von Bildern — 2

### Historie

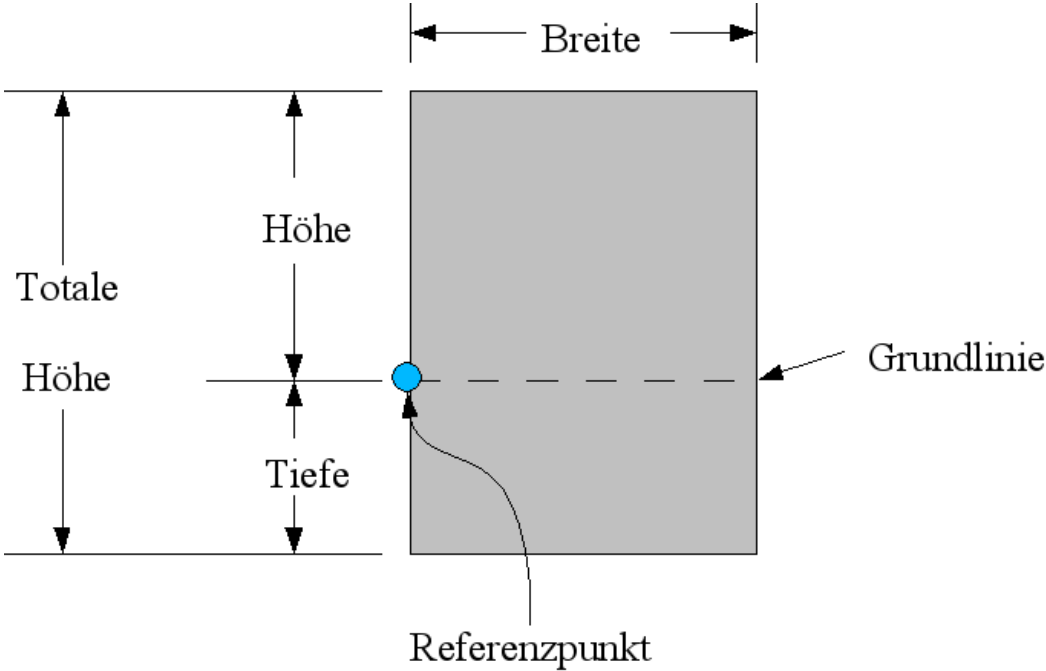
Da DVI-Dateien häufig zu PostScript konvertiert werden, bietet T<sub>E</sub>X eine optimale Unterstützung für EPS-Grafiken (*Encapsulated PostScript*). Um nicht immer mit den `\special`-Befehlen hantieren zu müssen, wurden für L<sup>A</sup>T<sub>E</sub>X2.09 die Pakete `epsf` und `psfig` entwickelt.

Im Zuge der Weiterentwicklung von L<sup>A</sup>T<sub>E</sub>X hat das L<sup>A</sup>T<sub>E</sub>X3-Team beschlossen, das sog. „L<sup>A</sup>T<sub>E</sub>X graphics bundle“ zu schreiben. Es besteht aus dem normalen `graphics`-Paket sowie einer erweiterten `graphicx`-Fassung. Letztere unterstützt die Skalierung und Rotation von Bildern.

# Setzen von Bildern — 3

## Grundlagen

Eine *Box* ist ein Objekt, das von  $\text{\LaTeX}$  als Einheit behandelt wird. Dabei kann es sich um ein Zeichen oder auch um eine Grafik handeln. An der Grundlinie werden alle Zeichen einer Zeile ausgerichtet. Jede Box kann somit durch Angabe von Breite, Höhe und Tiefe definiert werden.





# Setzen von Bildern — 4

## Befehle

Die Einbindung einer Grafik erfolgt mit dem Befehl

`\includegraphics[Optionen]{Dateiname}`

Folgende Optionen sind möglich:

height	Höhe der Grafik
totalheight	Totale Höhe der Grafik
width	Breite der Grafik
angle	Rotationswinkel in Grad, bezogen auf den Referenzpunkt. Positive Werte drehen entgegen dem Uhrzeigersinn.
origin	Gibt einen anderen Bezugspunkt für die Rotation an. Bspw. würde mit <code>origin=c</code> um das Bildzentrum rotiert werden.
bb	Spezifiziert sog. <i>Bounding Box</i> -Parameter (wird selten gebraucht)

## Setzen von Bildern — 5

### Beispiel

Das folgende Bild ist 46,92 Millimeter breit und 52,92 Millimeter hoch. Erfolgt die Einbindung ohne Größenanpassung, erhalten wir:

```
\includegraphics{grafik.jpg}
```



## Setzen von Bildern — 6

### Beispiel

Das Bild ist recht groß und erscheint am linken Rand. Schöner wäre eine etwas kleinere Darstellung und eine Zentrierung:

```
\begin{center}  
  \includegraphics[width=32mm]{grafik.jpg}  
\end{center}
```



## Setzen von Bildern — 7

### Beispiel

Unsere kleine Python-Schlange (Programmiersprache Python ist unter <http://www.python.org> zu finden) soll sich mehr horizontal fortbewegen. Zuerst die Drehung im und dann entgegen dem Uhrzeigersinn:

```
\includegraphics[width=25mm,angle=-40]{grafik.jpg}
```

```
\includegraphics[width=25mm,angle=140]{grafik.jpg}
```



## Setzen von Bildern — 8

### Weitere Optionen

Die Bildgröße kann auch mit Hilfe der `scale`-Option angepasst werden. Hier wird die Bildgröße verdoppelt:

```
\includegraphics[scale=2]{grafik.jpg}
```

### Ausrichtung von Bildern

Um ein Bild horizontal zu zentrieren, wird die `center`-Umgebung benutzt (wie in den vorhergehenden Beispielen). Üblicherweise werden Bilder aber innerhalb einer `figure`-Umgebung gesetzt, damit man neben einer Positionierungsangabe (s. dazu auch den Abschnitt über Tabellen) u. a. auch Bildunterschriften benutzen kann.

## Setzen von Bildern — 9

### Ausrichtung von Bildern

Eine Zentrierung kann dann durch den `centering`-Befehl erfolgen:

```
\begin{figure}  
  \centering  
  \includegraphics[width=25mm]{grafik.jpg}  
\end{figure}
```



## Setzen von Bildern — 10

### Bildunterschriften

Bilder stehen normalerweise nicht allein, sondern werden mit einer Bildunterschrift versehen. Dazu muss die betreffende Grafik in einer `figure`-Umgebung eingebunden sein und mittels `caption`-Befehl eine Unterschrift besitzen:

```
\begin{figure}
  \centering
  \includegraphics[width=25mm]{grafik.jpg}
  \caption{Python-Schlange, entnommen iX-Ausgabe 6/2005
    (\url{http://www.heise.de/ix/artikel/2005/06/128/}).}
\end{figure}
```

Sollte die Unterschrift zu lang sein, kann mit einer optionalen Kurzform ein Eintrag für das Abbildungsverzeichnis erstellt werden:

```
\caption[Python-Schlange]{Python-Schlange, entnommen iX-Ausgabe 6/2005
  (\url{http://www.heise.de/ix/artikel/2005/06/128/}).}
```

## Setzen von Bildern — 11

### Bildunterschriften

Hier nun die vollständige Abbildung:



Abbildung 1: Python-Schlange, entnommen iX-Ausgabe 6/2005 (<http://www.heise.de/ix/artikel/2005/06/128/>).



## Setzen von Bildern — 12

### Bildunterschriften

Abhängig davon, ob der `caption`-Befehl *oberhalb* oder *unterhalb* der Einbindung erfolgt, erscheint eine *Überschrift* oder eine *Legende*.

Um die Breite der Beschriftung anzupassen, kann sie in eine sog. *Parbox* gesteckt werden:

```
\parbox{Breite}{\caption{Beschriftung}}
```

Ist die Beschriftung kürzer als die Zeilenlänge, wird sie horizontal zentriert. Ist sie hingegen länger, wird sie wie ein normaler Absatz mit normaler Textbreite formatiert.

## Setzen von Bildern — 13

### Bildreferenzen

Möchte man später auf ein Bild Bezug nehmen zu können, kann ein Label gesetzt werden, dass dann referenziert werden kann:

```
\begin{figure}  
  \centering  
  \includegraphics[width=25mm]{grafik.jpg}  
  \label{python2005}  
\end{figure}
```

```
... \ref{python2005}
```

An Stelle der Referenz wird dann die Abbildungsnummer gesetzt.

## Setzen von Bildern — 14

### Bildreferenzen

Die vollständige Abbildung sieht dann so aus:



Abbildung 2: Python-Schlange, entnommen iX-Ausgabe 6/2005 (<http://www.heise.de/ix/artikel/2005/06/128/>).

Und noch eine Beispiel-Referenz: „... und in 43 sehen Sie die berühmt-berüchtigte Python-Schlange, die extrem hinterlistig einfache Programme zu schreiben erlaubt.“

## Setzen von Bildern — 15

### Bildreferenzen

Die Nummerierung der Referenz entspricht hier nicht der Abbildungsnummer, weil alle vorigen Bilder nicht innerhalb einer `figure`-Umgebung gesetzt wurden.

Wird das `hyperref`-Paket verwendet, kann an Stelle von `\ref{...}` auch `\autoref{...}` verwendet werden. Vor die Abbildungsnummer wird dann ein entsprechender Begriff gesetzt:

„... und in **Abbildung 43** sehen Sie die berühmt-berüchtigte Python-Schlange, die extrem hinterlistig einfache Programme zu schreiben erlaubt.“

Leider muss die Lokalisierung der Begriffe selbst durchgeführt werden. Dazu ist eine Redefinition notwendig:

```
\renewcommand{\figureautorefname}{Abbildung}
```

Weitere Details dazu sind auf Seite 12 des `hyperref`-Handbuchs von Sebastian Rahtz und Heiko Oberdiek zu finden.

# Literaturverzeichnisse — 1

## Variante 1

In L<sup>A</sup>T<sub>E</sub>X kann auf zwei Wegen ein Literaturverzeichnis erstellt werden. Für kleinere Dokumente, deren Literaturverweise voraussichtlich nicht in späteren Dokumenten wiederverwendet werden sollen, eignet sich die `thebibliography`-Umgebung. Das Literaturverzeichnis erscheint später dort im Text, wo diese Umgebung steht.

## Aufbau

```
\begin{thebibliography}{mustermarke}
  \bibitem[marke]{bezug} Eintrag
  ...
\end{thebibliography}
```

Die `mustermarke` dient zur Bestimmung der Einrücktiefe der Einträge. Sollen bspw. Einträge der Form *Spi89* oder *Teß04* verwendet werden, würde man als Mustermarke ABC12 benutzen (oder XYZ09 usw.).

## Literaturverzeichnisse — 2

### Einträge

Ein Eintrag in das Literaturverzeichnis erfolgt durch den Befehl `\bibitem`:

`\bibitem[marke]{bezug} Eintrag`

Die optionale Marke kann die schon oben genannte Form *Spi89* oder *Tefß04* haben. Wird sie weggelassen, erzeugt L<sup>A</sup>T<sub>E</sub>X eine fortlaufende Nummer. `bezug` ist derjenige Begriff, der für die Referenzierung mittels `\cite{bezug}` benötigt wird. Der Eintragstext besteht üblicherweise aus Autor, Titel, Auflage, Adresse, Jahr etc. Die genauen Vorgaben können den [Anforderungen an Seminararbeiten](#) entnommen werden.

## Literaturverzeichnisse — 3

### Beispiel

```
\begin{thebibliography}{ABC99}  
  \bibitem[Teß04]{tess} Teßmer, M.: Architekturermittlung aus Quellcode:  
    Möglichkeiten und Grenzen anhand einer Fallstudie. Bielefeld, 2004.  
  
  \bibitem[Spi89]{spi} Spitta, T.: Software Engineering und Prototyping.  
    Springer, Berlin-Heidelberg et.\,al, 1989.  
\end{thebibliography}
```

### Literatur

- [Teß04] Teßmer, M.: Architekturermittlung aus Quellcode: Möglichkeiten und Grenzen anhand einer Fallstudie. Bielefeld, 2004.
- [Spi89] Spitta, T.: Software Engineering und Prototyping. Springer, Berlin-Heidelberg et. al, 1989.

## Literaturverzeichnisse — 4

### Variante 2

Die zweite Weg zur Erzeugung eines Literaturverzeichnisses erfordert neben einer separaten Literaturdatenbank auch ein weiteres Programm namens **bibtex**. Dieses Programm erzeugt aus der Literaturdatenbank für **L<sup>A</sup>T<sub>E</sub>X** geeignete Einträge. Der Kompiliervorgang sieht nun, bedingt durch den **bibtex**-Aufruf, wie folgt aus:

```
latex mein_dokument.tex  
bibtex meine_literaturdatenbank  
latex mein_dokument.tex  
latex mein_dokument.tex
```

Man beachte, dass der Aufruf *ohne* die Dateiendung erfolgt!



## Literaturverzeichnisse — 5

### Literaturdatenbank

Die Literaturdatenbank ist eine Datei, in der die Einträge in das Literaturverzeichnis in einem bestimmten Format vorliegen. Beispiel:

```
@book{RoLi04,  
  author="Roock, Stefan and Lippert, Martin",  
  title="Refactoring in großen Softwareprojekt",  
  publisher="dpunkt.verlag",  
  year="2004",  
  edition="1. Auflage"  
}
```

# Literaturverzeichnisse — 6

## Eingabetyp und Felder

Die wichtigsten Eintragstypen sind `article`, `book` und `booklet`. Zu jedem Typ existieren zwingend erforderliche Felder sowie optionale Felder:

Eingabetyp	erforderlich	optional
article	author, title, journal, year	volume, number, pages, month, note
book	author oder editor, title, publisher, year	volume oder number, series, address, edition, month, note
booklet	title	author, howpublished, address, month, year, note, lastchecked, url, urn

## Literaturverzeichnisse — 7

### Eingabetyp und Felder

Der `booklet`-Eingabetyp dient auch zur Einbindung von Online-Ressourcen. Wichtig sind dabei die Felder `url` und `lastchecked`, die zur Angabe der Online-Adresse sowie des Datums dienen, wann diese Ressource zuletzt gelesen wurde.

Ein mit `bibtex` erzeugtes Literaturverzeichnis wird durch den Befehl `\bibliography{literaturdatenbank_datei}` — auch hier *ohne* Dateiendung — in ein `LATEX`-Dokument eingebunden. Das Verzeichnis erscheint dann an der Stelle dieses Befehls.

## Literaturverzeichnisse — 8

### Eingabetype und Felder

Durch den Befehl `\bibliographystyle{stil}` kann ein bestimmter Stil für das Literaturverzeichnis gewählt werden. Der Befehl muss *vor* dem Einfügen des Verzeichnisses gesetzt werden. Mögliche Stile sind:

**plain** Einträge werden alphabetisch nach Autorennamen sortiert und fortlaufend nummeriert.

**unsrt** Die Einträge erfolgen in der Reihenfolge der `\cite`-Befehle und werden fortlaufend nummeriert.

**alpha** Die Anordnung erfolgt wie bei **plain**, allerdings werden an Stelle fortlaufender Nummern Abkürzungen des Autorennamens samt Jahreszahl gesetzt.

**abbrv** Die Anordnung erfolgt wie bei **plain**, die Einträge erscheinen jedoch durch diverse Abkürzungen kompakter.

## Literaturverzeichnisse — 9

### Vorteile und Nachteile beider Varianten

Die erste Variante erfordert keine separat zu pflegende Datei und man kann die Einträge direkt formatieren, ohne sich an Feldvorgaben usw. halten zu müssen. Solange man keine größeren „Werke“ verfasst, deren Literaturverzeichnis auch für andere Dokumente wichtig sein könnten, ist dieser Weg wohl der einfachere.

Sobald ein größeres Dokument verfasst werden soll, das auch als Basis für Papers oder Tagungsbeiträge dient, ist die Einrichtung einer Literaturdatenbank sinnvoll. Zudem bieten viele Online-Quellen — auch die Bibliothek der Universität Bielefeld — die Möglichkeit, zu Büchern `bibtex`-konforme Angaben zu exportieren, die problemlos in die eigene Literaturdatenbank übernommen werden können.

## Wie geht es weiter?

- Für den Einstieg in  $\text{\LaTeX}$  können die Dokumente „ $\text{\LaTeX}$  in 7 Minuten“ ([http://www.wiwi.uni-bielefeld.de/StatCompSci/lehre/material\\_allgemein/latex7/sieben1/sieben1.html](http://www.wiwi.uni-bielefeld.de/StatCompSci/lehre/material_allgemein/latex7/sieben1/sieben1.html)) bzw. „Weitere 7 Minuten für  $\text{\LaTeX}$ “ ([http://www.wiwi.uni-bielefeld.de/StatCompSci/lehre/material\\_allgemein/latex7/sieben2/sieben2.html](http://www.wiwi.uni-bielefeld.de/StatCompSci/lehre/material_allgemein/latex7/sieben2/sieben2.html)) oder die „ $\text{\LaTeX}$ 2e-Kurzbeschreibung“ (<ftp://ftp.dante.de/tex-archive/info/lshort/german/l2kurz.pdf>) verwendet werden.
- Weiterführende Literatur:
  - Helmut Kopka:  $\text{\LaTeX}$  — Eine Einführung (sowie die Folgebände)
  - Michel Goossens, Frank Mittelbach: Der  $\text{\LaTeX}$ -Begleiter

# Ende