

Die neue T_EX FAQ soll Antworten auf die häufigsten Fragen zu T_EX, L^AT_EX und der Software »drumherum« geben. Dieses Dokument wird automatisch aus dem Inhalt der Webseite <http://texfragen.de> erzeugt. Daraus folgen zwei Dinge: die Inhalte der Webseite sind stets auf dem aktuellen Stand und Änderungen müssen im Web vorgenommen werden. Die TeX FAQ ist ein Wiki und kann derzeit von jedem bearbeitet werden. Der Stand dieses PDFs ist vom 24. Juli 2018.

1 Inhalt

1.1 Allgemeines

- ☞ Was ist TeX?
- ☞ Wie installiere ich TeX?
- ☞ Mein erstes LaTeX-Dokument
- ☞ Wie fange ich es an, wenn ich LaTeX erlernen möchte?
- ☞ Worin liegt der Unterschied zwischen TeX, LaTeX und ConTeXt?
- ☞ Was sind PDFTeX, LuaTeX und XeTeX?
- ☞ Was ist ConTeXt?
- ☞ Mein erstes ConTeXt-Dokument
- ☞ Was ist ε-TeX?
- ☞ Wie bearbeite ich LaTeX-Dokumente?

- ☞ Kann ich LaTeX auch online verwenden?
- ☞ Wo bekomme ich Hilfe?
- ☞ Was ist LaTeX3, und wie kann ich es benutzen?
- ☞ Begriffe aus der TeX-Welt erklärt (Glossar)
- ☞ Was ist das Dateiformat DVI und wie kann ich es anzeigen?
- ☞ Wozu dienen Dateien mit der Namensendung .xyz?
- ☞ Was ist CTAN?
- ☞ Was ist TDS, die »TeX Directory Structure«?
- ☞ Wie installiert man am besten ein neues Paket, eine neue Klasse oder eine neue Schrift in einem bestehenden TeX-System?
- ☞ Soll ich lieber KOMA-Script oder Memoir nehmen?

1.2 Äußere Erscheinung der Dokumente

- ☞ Ränder
- ☞ Seitenumbruch
- ☞ Wieso ist mit der Klasse »book« oder mit der Klassenoption »twoside« der innere Rand kleiner als der äußere?
- ☞ Schriftarten
- ☞ Wie hebt man Wörter im Text hervor?

- ☞ Was ist der Unterschied zwischen `\emph{}` und `\textit{}`?
- ☞ Wie verändere ich die Schriftgröße in meinem Dokument?
- ☞ Ich benötige ein spezielles Symbol in meinem Dokument. Wie kann ich es finden?
- ☞ Ich bekomme die Meldung `Overfull \hbox` – was kann ich tun?
- ☞ Worin bestehen die Unterschiede zwischen `\textwidth`, `\linewidth` und `\columnwidth`?
- ☞ Wie kann ich »Hurenkinder« (engl. widows), also einzelne Zeilen eines Absatzes am Kopf von Buchseiten, bzw. »Schusterjungen/Waisenkinder« (engl. clubs), einzelne Absatzzeilen auf der Seite unten, verhindern?
- ☞ Wie kann ich den Abstand zwischen den Aufzählungspunkten verringern?
- ☞ Wie kann ich die Aufzählungspunkte farbig markieren?

1.3 Abbildungen und Gleitumgebungen

- ☞ Wie binde ich Bilder in ein LaTeX-Dokumente ein?
- ☞ Wie kann ich Programmlistings in mein Dokument einbinden?
- ☞ Wie kann ich die Positionierung der Gleitumgebungen (Tabellen und Abbildungen) beeinflussen
- ☞ Wie funktionieren Gleitobjekte überhaupt?

- ☞ Wie kann ich Tabellen und Abbildungen in der Nähe des Aufrufs behalten
- ☞ Wie kann ich ein Gleitobjekt das alleine auf einer Seite steht oben an der Seite ausrichten und nicht vertikal zentriert
- ☞ Wieso werden die meisten meiner Abbildungen an das Ende des Kapitels oder Dokuments verschoben?
- ☞ Wie kann ich in einer Abbildung zwei Bilder nebeneinander setzen?
- ☞ Wie kann ich Bilder absolut positionieren?

1.4 *LaTeX-Makros*

- ☞ Wie kann ich mir mit Makros das Schreiben erleichtern?
- ☞ Wie kann ich innerhalb des optionalen Arguments einer Anweisung zusätzliche eckige Klammern verwenden?
- ☞ Gibt es einen einfachen Weg, die Seitennummer als »Seite 1 von 16« anzugeben?
- ☞ Was machen die Befehle `\makeatletter` und `\makeatother`
- ☞ Ich erhalte die Fehlermeldung »! You can't use `\spacefactor` in vertical mode.« oder »Command `\@` already defined.«. Mache ich etwas falsch?
- ☞ Warum verschluckt LaTeX ein Leerzeichen nach manchen Befehlen?

☞ Wie erzeuge ich automatisch Großbuchstaben?

☞ Wie kann ich Wörter wie »Inhaltsverzeichnis«, »Abbildung« etc. umbenennen?

☞ Wie kann ich festlegen, welche Gliederungsebenen im Inhaltsverzeichnis erscheinen sollen?

1.5 Bibliographien und Literaturverweise

☞ Was ist BibTeX?

☞ Was ist biber?

☞ Wie verwende ich biber in meinem Editor?

☞ Was sind .bib-Dateien?

☞ Was ist biblatex?

☞ Wie kann ich alle Einträge aus meinem Literaturverzeichnis ausgeben lassen?

☞ Gibt es bequeme Editoren für BibTeX-Literaturdatenbankdateien?

1.6 Spezialitäten

☞ Wie schreibe ich einen Brief mit LaTeX

☞ Wie schreibe ich Zahlen mit Einheiten richtig?

- ☞ Wie schreibe ich griechische Buchstaben aufrecht?
- ☞ Formelzeichenverzeichnis
- ☞ Wie kann ich meinen TeX-Lauf automatisieren?
- ☞ Warum werden Umlaute in meinem Dokument nicht richtig dargestellt?
- ☞ Wie funktioniert der Absatzumbruch in TeX?
- ☞ Was ist SyncTeX bzw. wie kann ich sinnvoll zwischen Texteditor und PDF-Betrachter wechseln?
- ☞ Was ist `-shell-escape` und wie aktiviere ich das im Editor?
- ☞ Wie kann ich einen Text klickbar machen und ihn zu einem Link auf einen anderen Text machen?
- ☞ Was sollen die Kommentar-Zeichen (%) am Ende von Zeilen?
- ☞ Wie kann man Ligaturen verhindern?

1.7 Silbentrennung

- ☞ Wie kann ich LaTeX an die deutsche Rechtschreibung anpassen?
- ☞ Warum trennt LaTeX Worte mit Umlauten nicht oder nicht korrekt?
- ☞ Wie kann ich Trennungsausnahmen mit Umlauten in `\hyphenation` verwenden?

☞ Wieso werden Texte in »typewriter«-Schriften nicht getrennt?

☞ Warum trennt LaTeX Wörter mit Bindestrich nicht bzw. nur am Bindestrich, auch wenn die Trennausnahmeliste diese enthält?

1.8 Mathematiksatz

☞ Warum soll ich $\backslash[...]$ anstelle von $$$...$$$ benutzen

☞ Gibt es einen Unterschied zwischen $...$$ und $\backslash(...)$?

☞ Wie kann ich einen aufrechten Differentialoperator erzeugen

☞ Warum fügt TeX in einer mathematischen Formel hinter einem Dezimalkomma einen kleinen Zwischenraum ein?

☞ Wie kann man Symbole in mathematischen Formeln fett drucken?

1.9 LuaTeX

☞ Was ist LuaTeX und kann ich es anstelle von LaTeX benutzen?

☞ Was muss ich beim Umstieg auf LuaTeX beachten?

☞ Erste Schritte mit LuaTeX

☞ Wie binde ich OpenType-Schriften ein?

1.10 Häufig gesehene Fehlermeldungen

- ☞! LaTeX Error: There's no line here to end.
- ☞! LaTeX Error: Option clash for package <name>.
- ☞! No room for a new \dimen.

1.11 DANTE

- ☞ Was ist DANTE?
- ☞ Warum soll ich Mitglied werden?
- ☞ Wie werde ich Mitglied?
- ☞ Die TeXnische Komödie
- ☞ Tagungen und Veranstaltungen

1.12 Über diese FAQ

Ergänzungen und Korrekturen direkt hier einarbeiten (Link unten) – auch wenn ihr euch unsicher seid. Im Zweifelsfall wird es von den nächsten Besuchern ergänzt. Lückenhafte Stellen könnt ihr mit FIXME bezeichnen, das wird dann sehr schnell gesehen. Ansonsten eine E-Mail an stefan@texblog.net senden. Die offizielle

Seite dieser FAQ ist <https://texfragen.de> und wird betrieben mit freundlicher Unterstützung durch DANTE.

1.13 Autoren

Der aktuelle Maintainer der FAQ ist Stefan Kottwitz.

Mitwirkende bitte immer selbst ergänzen. Aktuell in alphabetischer Reihenfolge (Nachname):

Johannes Böttcher, Patrick Gundlach, Martin Hehn, Stefan Kottwitz, Robin Krahl, Henri Menke, Clemens Niederberger, Georg Pfeiffer, Jonas Stein, Dominik Waßenhoven, Uwe Ziegenhagen. Besonderer Dank geht an die Autoren der alten FAQ, aus der die ein oder andere Frage übernommen wurde: Bernd Raichle, Rolf Niepraschk und Thomas Hafner.

1.14 Lizenz dieser FAQ

Die Textbeiträge dieser FAQ unterliegen der Lizenz [CC-BY-SA 3.0](#). Dadurch ist es erlaubt, die Texte frei zu verwenden, sofern sie weiterhin unter denselben Bedingungen verbreitet und die Autoren genannt werden. Die genauen Bestimmungen finden sich [auf der Seite von Creative Commons](#).

2 Was ist TeX?

TeX (gesprochen: Tech) ist eine Software zur *Formatierung von Texten*. Der Unterschied zu anderen Textverarbeitungssystemen wie

Microsoft Word und Apache OpenOffice Writer ist, dass aus einer TeX-Datei, die vom Anwender erstellt wird, eine PDF-Datei generiert wird. Die ursprüngliche TeX-Datei wird mit einem Texteditor bearbeitet. Hierbei können verschiedene Befehle zur Gestaltung des Textes verwendet werden. Die Ausgabedatei im PDF-Format kann dann mit jedem Rechner problemlos betrachtet werden.

Außerdem kann man TeX auch als Programmiersprache verwenden und bestimmte Dinge vereinfachen. Zum Beispiel unterstützt LaTeX von Haus aus automatische Gliederungsbefehle, die dann bei Bedarf individuell angepasst werden können.

TeX und LaTeX sind freie Software, und sie sind frei erhältlich, d.h. es fallen keine Lizenzgebühren an. Eine Installation besteht meist aus dem Binary (z.B. PDFTeX oder XeTeX), dem Format (z.B. LaTeX) und vielen Zusatzpaketen und Schriftarten. Es gibt für viele Betriebssysteme fertig konfigurierte Distributionen, die eine Installation sehr vereinfachen.

2.1 Einsatzzwecke TeX und Co.

LaTeX, ConTeXt und auch Plain TeX eignen sich für viele Dokumente. Während es früher hauptsächlich im universitären Bereich benutzt wurde (TeXs Fähigkeit mathematische Formeln zu setzen ist bis heute unübertroffen), wird es heutzutage in sehr vielen Anwendungsfällen eingesetzt. Folgende Dokumente lassen sich erzeugen: Bücher, kleinere und größere Aufsätze, Briefe, Präsentationen, Handbücher, Flyer und Poster, Fahrpläne, Produktkataloge, Datenblätter, Notenbücher und vieles andere mehr. Durch die Flexibilität sind dem System keine Grenzen gesetzt.

2.2 Beispiel für eine LaTeX-Datei

Es folgt ein vollständiges Beispiel für eine LaTeX-Eingabedatei. Man erkennt schön, dass es keinerlei Formatierungsanweisungen gibt, nur *logische* Auszeichnungen wie `\emph` oder `\tableof-contents`

```
\documentclass[a4paper]{article}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\usepackage{graphicx}
\usepackage[T1]{fontenc}

\begin{document}
\title{Eine Welt voller Wasser}
\author{N. Autilus}
\maketitle

\section{Einleitung}
```

Im Meer gibt es `\emph{viele}` Pflanzen- und Tierarten. Das wird in Abbildung `\ref{abb:ocean}` gezeigt.

```
\begin{figure}[hb]
\centering
\includegraphics[width=4cm]{ocean}
\caption{Ozean}
\label{abb:ocean}
\end{figure}
```

Man sollte nur aufpassen, die richtigen Zeilen zu schreiben, sonst hat man schnell die falschen. Oder man vergisst ganz und gar etwas zu schreiben -- ein leeres Blatt wäre die Folge.

```
\end{document}
```

Eine Welt voller Wasser

N. Autilus

10. März 2013

1 Einleitung

Im Meer gibt es *viele* Pflanzen- und Tierarten. Das wird in Abbildung 1 gezeigt.



Abbildung 1: Ozean

Man sollte nur aufpassen, die richtigen Zeilen zu schreiben, sonst hat man schnell die falschen. Oder man vergisst ganz und gar etwas zu schreiben – ein leeres Blatt wäre die Folge.

2.3 Beispiel für eine TeX-Datei

Im Gegensatz zu dem LaTeX-Beispiel gibt es in „plain TeX“ wenige solcher Strukturbefehle, dafür eignet sich plain TeX für Layouts, bei der man die volle Kontrolle benötigt.

```
\hspace=115mm
```

```
``Good morning!'' he said at last. ``We don't want any
adventures
here, thank you! You might try over The Hill or across
The Water.''
By this he meant that the conversation was at an end.
```

```
``What a lot of things you do use {\it Good morning\}/}
  for!''
said Gandalf. ``Now you mean that you want to get rid
  of me,
and that it won't be good till I move
off.''\footnote*{J. R. R. Tolkien, {\it The Hobbit.}}
\bye
```

Ruft man nun (auf der Kommandozeile oder aus dem Editor heraus) `pdftex` auf, ergibt das folgende Ausgabe:

“Good morning!” he said at last. “We don’t want any adventures here, thank you! You might try over The Hill or across The Water.” By this he meant that the conversation was at an end.

“What a lot of things you do use *Good morning* for!” said Gandalf. “Now you mean that you want to get rid of me, and that it won’t be good till I move off.”*

und weiter unten die Fußnote und Seitenzahl:

* J. R. R. Tolkien, *The Hobbit*.

2.4 Formelsatz

Wie oben erwähnt, ist Formelsatz einer der Stärken von TeX (und damit auch LaTeX und ConTeXt). Formelnotationen anderer mathematischer Programme (z.B. MathJax) lehnen sich an die von LaTeX an. Außerdem ist die Art und Weise, wie TeX Formeln

zusammenbaut, Vorbild für Microsoft Word und OpenType Math. Nachfolgend sind einige Beispiele für Formelsatz mit LaTeX:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

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

```
\documentclass{article}
\begin{document}
\[ x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]
\[ \int_a^b f(x) dx = F(b) - F(a) \]
\end{document}
```

MAXWELL-GLEICHUNGEN IN DIFFERENTIALFORM:

$$\nabla \cdot \mathbf{D} = \rho$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

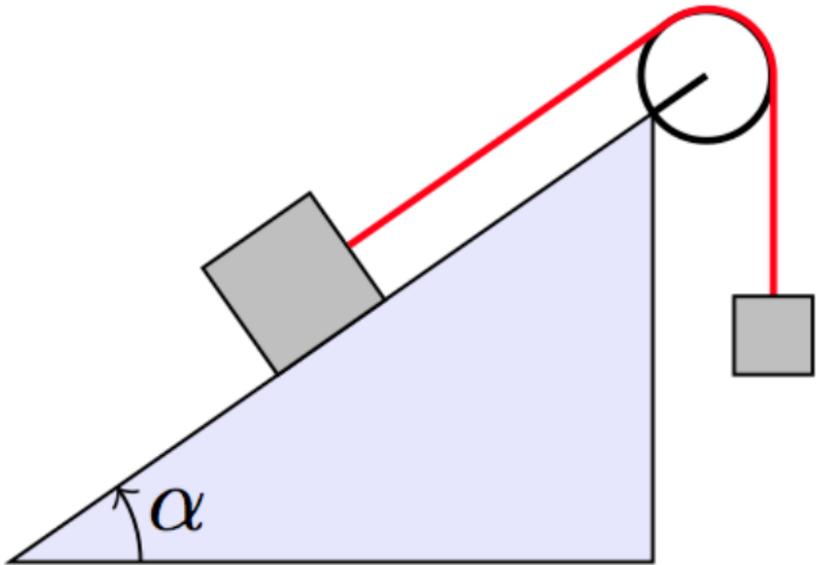
```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\textsc{Maxwell-Gleichungen in Differentialform:}

\[ \nabla \cdot \mathbf{D} = \rho \]
\[ \nabla \cdot \mathbf{B} = 0 \]
\[ \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \]
\end{document}
```

```
\[ \nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \]  
\end{document}
```

2.5 Abseits von Text

Mit TeX kann man aber nicht nur Text erzeugen, sondern auch Diagramme und andere graphische Elemente. Nachfolgend ein Beispiel¹, das mit TikZ/PGF erstellt wurde:



¹ Besten Dank an Andrew Stacey für das Beispiel. Auf der Seite <http://texample.net> ist es – neben vielen anderen – mit Quellcode und Ausgabe zu finden

Der Quellcode ist auf den ersten Blick sicherlich etwas abschreckend. Mit ein wenig Übung sind solche Dokumente aber schneller erstellt, als man vermutet. Ein großer Vorteil von Diagrammen, die in TeX erstellt werden, soll erwähnt werden: Die Schriftart und die Farben, Strichstärken und ähnliche Dinge lassen sich für alle Diagramme in einem Dokument zentral einstellen. Somit harmonisieren diese Grafiken perfekt mit dem restlichen Dokument.

```

\documentclass[border=3mm]{standalone}
\usepackage{tikz}
\begin{document}

\def\iangle{35} % Angle of the inclined plane
\def\arcr{0.5cm} % Radius of the arc used to indicate
                 angles

\begin{tikzpicture}[
  M/.style={rectangle,draw,fill=lightgray,minimum size
             =0.5cm,thin},
  m/.style={rectangle,draw=black,fill=lightgray,minimum
             size=0.3cm,thin},
  plane/.style={draw=black,fill=blue!10},
  string/.style={draw=red,thick},
  pulley/.style={thick},
]

\draw[plane] (0,-1) coordinate (base)
             -- coordinate[pos=0.5] (mid) ++(\iangle
             :3) coordinate (top)
             |- (base) -- cycle;
\path (mid) node[M,rotate=\iangle,yshift=0.25cm] (M)
        {};
\draw[pulley] (top) -- ++(\iangle:0.25) circle (0.25
        cm)
             ++ (90-\iangle:0.5) coordinate (pulley);
\draw[string] (M.east) -- ++(\iangle:1.5cm) arc
             (90+\iangle:0:0.25)
             -- ++(0,-1) node[m] {};

\draw[->] (base)++(\arcr,0) arc (0:\iangle:\arcr);
\path (base)++(\iangle*0.5:\arcr+5pt) node {\$\alpha}

```

```
$};
```

```
\end{tikzpicture}
```

```
\end{document}
```

3 *Wie installiere ich TeX?*

Um TeX, LaTeX und Co. verwenden zu können, braucht man eine sogenannte TeX-Distribution. Bekannte TeX-Distributionen sind [TeX Live](#) (plattformübergreifend) und [MiKTeX](#) (Windows). Diese TeX-Distributionen sind freie Software und können daher kostenlos heruntergeladen und installiert werden.

3.1 *Linux*

Die meisten Linux-Distributionen bieten über die Paketverwaltung eine TeX-Live-Installation an. (Das entsprechende Paket heißt in allen gängigen Distributionen `texlive`.) Für den Einstieg ist diese vollkommen ausreichend. Erfahrene Benutzer können TeX Live manuell von <http://tug.org/texlive> installieren, um immer die jeweils aktuelle Version zu haben.

3.2 *Windows*

Die einfachste Möglichkeit, TeX unter Windows zu installieren, ist MiKTeX. Auf der [MiKTeX-Homepage](#) steht ein Installer zum Download bereit. MiKTeX hat die nette Eigenschaft, benötigte,

aber nicht lokal vorhandene Pakete automatisch aus dem Netz nachzuladen.

Der *basic*-Installer ist ca. 150 MB groß und installiert (s.o.) Pakete auf Anforderung nach. Der *net*-Installer ist etwa 7-10 MB groß und installiert aus dem Netz ein vollständiges System. Daneben gibt es noch die *portable* Version, die ohne Administrationsrechte und ohne Spuren zu hinterlassen auf einem Windows Rechner ausgeführt werden kann.

3.3 Mac OS X

Für Mac OS X gibt es eine Distribution, die auf TeX Live aufbaut: [The MacTeX-2012 Distribution](#). Erfahrene Anwender können auch über die Kommandozeile TeX Live installieren.

Achtung: Es gibt noch vereinzelt Hinweise im Netz, die den den i-Installer empfehlen. Dieser wird aber nicht mehr weiterentwickelt.

Hilfe für TeX unter Mac OS X gibt es in der englischsprachigen Mailingliste »MacOSX-TeX«, die auf der Hilfe Seite unter <http://tug.org/mactex/help/> verlinkt ist.

4 Mein erstes LaTeX-Dokument

Ein einfaches LaTeX-Dokument sieht so aus:

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

```
\begin{document}
```

```
Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine  
Familie lebt dagegen in  
Hamburg. Und dazwischen ich, aber ganz cool.  
Vollbremskombination aus Ampel  
oder Einfädeln oder beides auf einmal\ldots
```

```
\end{document}
```

und wird mit dem Befehl `pdflatex <dateiname.tex>` übersetzt. Spezielle  Editoren haben dafür meist eine Schaltfläche oder einen Kurzbefehl. Das erzeugte PDF sieht so aus:

```
Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen  
in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus  
Ampel oder Einfädeln oder beides auf einmal...
```

Jedes LaTeX-Dokument hat diese Grundstruktur:

```
\documentclass{<Dokumentklasse>
```

```
% Präambel
```

```
\begin{document}
```

```
% Text
```

```
\end{document}
```

Das %-Zeichen ist ein Kommentarzeichen. Alles nach dem % wird ignoriert.

Für jedes Dokument muss man eine **Dokumentklasse** angeben. Es gibt viele verschiedene Dokumentklassen, die mit LaTeX installiert sind wie `article`, `report`, `book`, `scrartcl`, `scrre-`

prt, scrbook oder beamer. In der LaTeX-Kurzanleitung `l2kurz.pdf` werden diese erläutert.

Zwischen dem Laden der Dokumentenklasse und `\begin{document}` ist die sogenannte **Präambel**, in der zusätzliche Pakete geladen und eigene Definitionen formuliert werden können. Der eigentliche Text steht zwischen `\begin{document}` und `\end{document}`.

In unserem Beispiel werden drei Pakete geladen. Die Anweisung `\usepackage[ngerman]{babel}` bewirkt, dass das Paket `babel` geladen wird und die Option `ngerman` übergeben bekommt. Was mit der Option passiert, ist paketabhängig. Im Fall von `babel` wird auf deutsche Trennmuster umgeschaltet und einige interne LaTeX-Namen wie »chapter« und »figure« werden übersetzt, damit diese in Querverweisen in der ausgewählten Sprache erscheinen.

Das Paket `fontenc` mit der Option `T1` bewirkt, dass LaTeX intern auf eine 256-Bit-Fontkodierung umschaltet. Das ist deshalb wichtig, weil sonst Wörter mit Umlauten nicht richtig getrennt werden. Ohne diese Zeile würde LaTeX keine Trennung im Wort »Mädchen« finden.

Zum Schluss bleibt noch die Zeile mit der Eingabekodierung (Paket `inputenc`). Da der Computer einer Textdatei nicht ansieht, in welcher Kodierung sie gespeichert ist, muss man sie explizit mit angeben. Die bekannteste Kodierung heutzutage ist UTF-8. Auf Unix-Systemen wie Linux war lange Zeit ISO 8859-1 (was der Angabe von `latin1` beim Paket `inputenc` entspricht) oder ISO-8859-15 (`latin9`) üblich, auf Windows eher die Codepage 1252 (`ansinew`). Im Normalfall wird man heutzutage UTF-8

(`utf8` bei `inputenc`) benutzen, es sei denn, der Editor kann mit der Kodierung nicht umgehen (wie es derzeit bei TeXnicCenter der Fall ist, aber die Version 2 wird auch mit UTF-8-kodierten Dokumenten zurecht kommen).

Siehe hierzu den eigenen Abschnitt über [☞ Wie kann ich die Kodierung meines Dokuments bestimmen? Oder: warum funktionieren meine Umlaute nicht?](#) in dieser FAQ.

5 Wie fange ich es an, wenn ich LaTeX lernen möchte?

Wenn man nicht sowieso schon eine TeX-Distribution [☞ installiert](#) hat, kann man erst einmal auf ein kostenloses [☞ Online-Angebot](#) zurückgreifen, um die ersten Gehversuche zu unternehmen.

Mit jeder TeX-Distribution wird eine große Menge an Dokumentation mitgeliefert. Diese ist jedoch oft sehr speziell und daher nicht wirklich geeignet, um LaTeX zu lernen. Zum Einstieg bietet sich das kostenlose Dokument [»☞ l2kurz.pdf«](#) an, das auch in der Distribution über den Kommandozeilenbefehl `texdoc l2kurz` aufgerufen werden kann.

Weiterhin gibt es eine schier unüberschaubare Menge an Büchern zu LaTeX. In der alten FAQ gibt es [☞ eine ausführliche Literaturliste](#).

Zum Einstieg bieten sich folgende kostenlose PDFs an:

- [☞ LaTeX2ε-Kurzbeschreibung](#) Marco Daniel, Patrick Gundlach und andere (oben erwähnt, `texdoc l2kurz`, Stand 2012)

- [☞ Bilder einfügen in LaTeX: Ein How-To von Dominik Bischoff \(texdoc 12picfaq.pdf, Version 1.50 vom 29. August 2010\)](#)
- [☞ Die LaTeX-Sündensammlung von Marc Ensenbach, Mark Trettin \(texdoc 12tabu.pdf, Version 2.3 vom 20. September 2011\)](#)
- [☞ LaTeX – eine Einführung und ein bisschen mehr ... von Manuela Jürgens und Thomas Feuerstack, aktualisiert 2012.](#)

5.1 Und dann?

Wer die obigen Anleitungen durchgearbeitet hat, sollte sich den LaTeX-Begleiter (bzw. besser die englische Ausgabe: The LaTeX Companion) anschaffen. Der ist zwar nicht mehr auf dem neusten Stand, aber eine bessere Referenz wird man wohl nicht mehr finden. Und wenn man damit *durch* ist, ist der nächste Schritt das originale TeXbook von Knuth. Damit versteht man die ganzen Hintergründe.

Siehe [☞](#) die Buchliste.

6 *Worin liegt der Unterschied zwischen TeX, LaTeX und ConTeXt?*

TeX ist die Grundlage aller Erweiterungen. Es bietet nur recht einfache Programmierbefehle, um Texte zu setzen.

LaTeX und ConTeXt nutzen diese einfachen Befehle, um für den Benutzer eine wesentlich bequemere Schnittstelle zur Verfügung zu stellen. So bieten die sogenannten »Formate« LaTeX und ConTeXt einen Befehl `\section{...}`, um einen neuen Abschnitt einzuleiten. Intern werden dann TeX-Befehle ausgeführt um die Schriftart beispielsweise auf 12 Punkt Fett umzuschalten, einen Abstand vorher und nachher einzufügen, gegebenenfalls eine neue Seite zu beginnen, eine Abschnittsnummer vor die Überschrift zu setzen sowie einen Eintrag in das Inhaltsverzeichnis einzufügen. Das erspart dem Anwender viel mühselige Arbeit und ermöglicht es, das Layout zentral zu steuern.

Anfänger greifen in der Regel auf das Format »LaTeX« zurück, da es dafür mehr deutschsprachige Literatur gibt. Auch behandelt diese FAQ größtenteils dieses Format. Sofern nicht anders angegeben, beziehen sich alle Fragen darauf.

6.1 Gängige Formate

Plain TeX

Donald E. Knuth hat für TeX ein Makropaket namens Plain-TeX erstellt und diese Makros im Buch »The TeXBook« beschrieben. Dabei stellt Plain-TeX einen minimalen Satz an Makros dar, um TeX sinnvoll verwenden zu können. Zusätzlich enthält Plain-TeX einige Makros, wie `\beginsection` oder `\proclaim`, die man als Beispiel für komplexere Anweisungen ansehen kann. Wenn jemand sagt, dass er »TeX programmiert«, so meint er normalerweise, dass er in TeX und den Plain-TeX-Makros programmiert.

LaTeX

LaTeX ist das bekannteste Format. Viele  Editoren unterstützen von Hause aus nur LaTeX, manche auch plain TeX. Bücher behandeln fast ausschließlich dieses Format.

LaTeX wurde ursprünglich Anfang der 1980er Jahre von Leslie Lamport geschrieben und wird derzeit von Freiwilligen weiter entwickelt. Die aktuelle Version ist LaTeX2 ϵ , die die Version 2.09 im Jahr 1994 abgelöst hat. Ein längerfristiges Ziel ist die Entwicklung von  LaTeX 3.

ConTeXt

ConTeXt ist ein von Hans Hagen geschriebenes Makropaket für TeX. Es erlaubt, ähnlich wie LaTeX, das Erstellen von Texten mit Hilfe von Strukturmakros (Überschrift u. ä.). Bei der Erstellung wurde auf eine leichte Benutzbarkeit und Flexibilität geachtet, zudem wird die Erstellung interaktiver Texte unterstützt. Eine Besonderheit ist die multilinguale Schnittstelle, die z. B. niederländisch, englisch und deutsch unterstützt, sowie das Setzen von multilingualen Unterschriften (»Tabelle« statt »Table« u. ä.).

6.2 IniTeX

Das Programm `initex` ist quasi ein *rohes* TeX, in dem kein Format geladen wurde. In diesem Modus besitzt TeX nur ca. 300 primitive »control sequences«. Davon sind ungefähr die Hälfte Anweisungen, die andere Hälfte Parameter. Für einen Autor

sind diese Anweisungen alleine zu einfach und dürftig, so dass es ratsam ist, immer eines der oben genannten Formate zu benutzen.

7 Was sind PDFTeX, LuaTeX und XeTeX?

PDFTeX, LuaTeX und XeTeX sind sogenannte »engines«, also die Maschinen, die die sogenannten TeX-Formate ausführen. Beispielsweise läuft LaTeX sowohl unter PDFTeX, als auch unter LuaTeX und XeTeX. Jedes dieser Programme hat seine Vor- und Nachteile.

PDFTeX wurde von Hàn Thế Thành entwickelt und Ende Februar 1997 in ersten Testversionen veröffentlicht. PDFTeX ist der direkte Nachfolger des originalen TeX von D. E. Knuth (dem ursprünglichen Entwickler von TeX), aber um die Möglichkeit erweitert, direkt PDF anstelle dem TeX eigenen DVI-Format auszugeben. PDFTeX ist vermutlich derzeit die am meisten eingesetzte »engine«.

Es gibt mehrere Nachfolger von PDFTeX, von denen XeTeX und LuaTeX die wichtigsten sind. Beide können mit OpenType-Schriftart und Unicode (UTF-8) problemlos umgehen. Unterschiedlich ist, wie diese beiden Programme intern arbeiten. LuaTeX enthält einen eingebetteten Interpreter der Programmiersprache Lua, der Zugriff auf viele TeX-Internals hat, was die Programmierung von Erweiterungen gegenüber der Programmierung in TeX stark vereinfacht. Derzeit ist XeTeX wohl etwas stabiler als LuaTeX, die Erweiterbarkeit von LuaTeX wird aber vermutlich XeTeX mittelfristig ablösen.

8 Was ist ConTeXt?

8.1 Grundlagen

ConTeXt ist, genau wie LaTeX, ein Makropaket für das TeX Textsatzprogramm. Da die beiden aber vollkommen unabhängig voneinander entwickelt werden, haben sie nicht sehr viel miteinander zu tun. Beide haben natürlich gemeinsam, dass sie auf Plain TeX aufbauen und daher dieselben Regeln zutreffen, z. B. dass Makros mit `\` beginnen.

Am Beispiel des folgenden Hallo-Welt-Dokuments sieht man die gravierenden Syntax-Unterschiede.

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

In ConTeXt:

```
\starttext
\startsection[title={Hallo Welt!}]
Hallo Welt!
\stopsection
\stoptext
```

Die aktuelle Version von ConTeXt heißt MkIV (Mark 4) und baut auf LuaTeX auf. Die stabile, eingefrorene Vorgängerversion heißt MkII und unterstützte pdfTeX und XeTeX. Wenn man etwas über ConTeXt lernen möchte sollte man sich ausschließlich mit MkIV beschäftigen, da MkII seit langem nicht mehr weiterentwickelt wird und neue, interessante Features nur in MkIV implementiert werden.

Einen guten Einstieg in die ConTeXt-Syntax liefert »ConTeXt MkIV, an excursion« welches in TeXlive über `texdoc ma-cb-en` zugänglich ist. Das ältere »ConTeXt, an excursion« ist unvollständiger und bezieht sich auf MkII.

8.2 Warum sollte man sich für ConTeXt interessieren?

1. Es gibt keine Pakete. Es existieren ein paar wenige Module, welche man mit `\usemodule[<Modul>]` einbindet, eigentlich ist es aber so gedacht, dass ConTeXt bereits alles mitbringt. Nie wieder `Option clash for package XYZ`.
2. »XML-Unterstützung. Mit ConTeXt kann man XML verarbeiten und setzen, aber auch exportieren um zum Beispiel docbook oder »ePub zu erstellen.
3. »PDF-X. Manchmal ist es nötig standardisiertes PDF zu erzeugen. ConTeXt kann sogar »tagged PDF erstellen, was mit LaTeX als unmöglich gilt.
4. »UTF-8- und Systemschriftunterstützung. ConTeXt MkIV baut auf LuaTeX auf und bringt entsprechend alle Funktionen mit, welche auch standardmäßig aktiviert sind.
5. »Grundlinienraster.
6. Konsistentes Key-Value-Interface. Beinahe alles in ConTeXt funktioniert über das Tripel `\setupsomething[key=value]`, `\startsomething`, `\stopsomething`.

7. Buffer und Setups. Um das zu schätzen muss man sich schon etwas mit ConTeXt beschäftigt haben.
8. \LaTeX -Interface. Alle Befehle, die in ConTeXt als `\befehl` verfügbar sind kann man innerhalb von ConTeXt auch in Lua verwenden mit `context . befehl`.
9. Das `context`-Programm macht alles in einem Aufwasch. Es erkennt selbst wie oft es durchlaufen muss um Querverweise, Indizes und Literaturverzeichnis richtig zu haben. Es werden mit MkIV *keine* Zusatzprogramme wie `makeindex` oder `bibtex` benötigt.
10. Es gibt weniger »Dateimüll«. `context` legt lediglich eine `.log`-Datei und eine `.tuc`-Datei an (Äquivalent zur `.aux`-Datei).

8.3 Quelle

Verfasst von Henri Menke auf www.texwelt.de: [Was ist ConTeXt?](#)

9 Mein erstes ConTeXt-Dokument

ConTeXt funktioniert so ähnlich wie LaTeX: Man schreibt einen Text, angereichert mit Formatierungsbefehlen, speichert ihn in eine Datei und verarbeitet diese mit dem Programm `context`.

```
\mainlanguage[de]
\starttext
Hallo schöne Welt
```

```
\blank[big]
```

```
\externalfigure[ozean][width=0.5\textwidth]
```

```
\stoptext
```

Der Text wird nun mit `context <dateiname>` übersetzt.

1

Hallo schöne Welt



10 Was ist ϵ -TeX?

Unter dem Namen » ϵ -TeX« (extended/enhanced TeX) wird unter der Leitung von Peter Breitenlohner TeX um dringend benötigte Erweiterungen ergänzt. Ein wichtiges Ziel von ϵ -TeX ist dabei, weiterhin 100% kompatibel zu TeX zu bleiben, solange die Erweiterungen nicht benutzt werden. Die Entwicklung basiert auf dem aktuellen TeX-Quellcode in einem durch WEB-Konstrukte erweiterten Standard-Pascal, so dass dies und die genannte Zielsetzung gewisse Beschränkungen, wie der Verzicht auf eine dynamische Speicherverwaltung, auferlegt, was grundlegende Neuerungen in ϵ -TeX anbelangt. Die erste Version von ϵ -TeX wurde im Oktober 1996 auf der Mitgliederversammlung von DANTE e.V. in Hamburg veröffentlicht. ϵ -TeX ist in allen aktuellen  engines enthalten.

11 Wie bearbeite ich (La)TeX-Dokumente?

TeX- oder LaTeX-Quellcode wird in reinen Textdateien gespeichert. Diese können im Prinzip mit jedem Texteditor bearbeitet werden (siehe dazu den  Artikel in der Wikipedia). Es bieten sich aber spezielle Editoren für LaTeX an. Es gibt hier im Prinzip zwei Arten: Einerseits die *normalen* Texteditoren wie Emacs, Vim oder Textmate, die um LaTeX-Fähigkeiten erweitert werden. Andererseits gibt es Editoren wie  Texmaker,  TeXStudio,  TeXShop oder  TeXworks, die speziell für den Einsatz mit LaTeX konstruiert wurden. Als Dateiendung für die Dateien hat sich `.tex` etabliert.

Neben diesen Texteditoren gibt es noch spezielle Anwendungen wie  LyX oder dessen kommerzielles Vorbild Scientific WorkPlace,

die die TeX-Befehle sofort grafisch darstellen, meist aber in einer »So ungefähr könnte es später aussehen«-Vorschau (*what you see is what you mean*).

Einen Vergleich von LaTeX-Editoren gibt es beispielsweise auf tex.stackexchange.com unter [☞ LaTeX Editors/IDEs](#).

12 Kann ich LaTeX auch online verwenden?

Wenn man an einem fremden Rechner ohne LaTeX-Installation arbeiten muss, kann ein web-basierter LaTeX-Editor hilfreich sein. Uns sind die folgenden Angebote bekannt:

- [☞ ShareLaTeX](#) - kostenlos, private und öffentliche Projekte, Live-Kollaboration
- [☞ Overleaf](#) - kostenloser LaTeX Cloud-Dienst, Premium-Abo kostenpflichtig
- [☞ verbosus](#), Standardaccount kostenlos, Premiumaccount kostenpflichtig
- [☞ publications.li](#) - kostenloser(?) Dienst.

13 On- und offline Ressourcen zum Thema TeX / LaTeX

13.1 Anleitungen und Dokumentation

- Die Anleitung (oder zumindest vorhandene Information) zu einem Paket X, welches mit `\usepackage[] {X}` eingebunden wird, kann sowohl unter Windows als auch unter Linux in einer Konsole mit `texdoc X` geöffnet werden.
- Diese Anleitungen können online über <http://texdoc.net/> abgerufen werden. Die Webseite bietet auch einen Direktzugriff auf eine Anleitung über <http://texdoc.net/pkg/...> an, beispielsweise <http://texdoc.net/pkg/babel> für das Paket `babel`.
- Eine Einführung in LaTeX (und ein wenig mehr) gibt es kostenlos im Dokument `l2kurz.pdf`. Siehe den Abschnitt [☞ Wie fange ich an, wenn ich LaTeX lernen möchte?](#)

13.2 Fragen und Antworten

Wer eine spezielle Frage hat, hat viele Möglichkeiten:

Deutsch

- Die »Frage und Antwort« Seite unter <http://texwelt.de/wissen>. Noch recht neu aber schon sehr aktiv.

- Die [Mailingliste »tex-d-l«](http://www.listserv.dfn.de/archives/tex-d-l.html). Archiv und Eintragen unter <http://www.listserv.dfn.de/archives/tex-d-l.html>
- Das Webforum [goLaTeX](#)
- Das LaTeX-Forum bei [mrunix.de](#)
- Hilfen speziell für KOMA-Script: <http://www.komascript.de/>
- Die [Newsgruppe de.comp.text.tex](#)
- Für Mitglieder von [DANTE e.V.](#) gibt es einen Beraterkreis, der TeX-Fragen kompetent beantwortet.

Englisch

- [TeX.StackExchange.com](#), eine sehr lebhafte Q&A-Webseite mit vielen aktiven Experten (»Q&A« = Frage und Antwort, was bedeutet, dass es sich nicht um klassisches Forum handelt).
- Die [Newsgruppe comp.text.tex](#)
- Das Forum [LaTeX-Community](#)

Mailinglisten:

- [LuaTeX users](#) und [LuaTeX-Entwicklerliste](#)
- [ConTeXt](#)
- [Metapost](#)
- [LaTeX-L](#)
- [Weitere Mailinglisten bei der TUG](#) und [bei der NTG](#).

Weiterhin gibt es noch diverse Blogs über TeX & friends, die unter planet.dante.de gesammelt sind.

13.3 Persönliche Hilfe

- LaTeX Kurse und Tutorien gibt es auf den zwei Mal im Jahr stattfindenden [DANTE-Treffen](#) und an diversen Hochschulen, [beispielsweise der FU Berlin](#). **Ergänzungen hierzu sind herzlich willkommen!**
- In vielen Städten gibt es TeX-Stammtische. Eine Liste findet sich im [„Stammtisch-Wiki“](#) von DANTE.

13.4 Kommerzielle Dienstleister

Es gibt eine Reihe kommerzieller Dienstleister rund um das Thema TeX, LaTeX, ConTeXt – meist mit tiefgehenden Kenntnissen in den Themen drumherum wie Textsatz & Layout, XML/XSLT oder Cross-Media-Publishing.

Ulrike
scher
Martin
Sievers

Fi- Mönchen-
gladbach
Trier

☞troubleshooting-
tex
☞in-
fo@schoen-
erpublizie-
ren.de

Vorlagener-
stellung, Bi-
bliographien,
Mathemati-
scher Satz

Martin
Wilhelm
Leidig

Ladenburg /
Heidelberg

☞mwl@gmx.net Buch- und
Zeitschriften-
satz, (La)TeX-
Programmierung
DTP (Quark
XPress usw.),
Bildbearbei-
tung

Henning
Hraban
Ramm

Limburg

☞fiëé visuelle

Buch- und
Zeitschrif-
tensatz,
ConTeXt,
DTP (In-
Design),
Bildbear-
beitung,
Web- und
Spezialpro-
grammierung

14 Was ist LaTeX3, und wie kann ich es benutzen?

LaTeX wurde ursprünglich von Leslie Lamport geschrieben und durch eine Gruppe Freiwilliger zu dem heute aktuellen LaTeX2 ϵ weiter entwickelt. Dieser Stand ist mehr oder weniger »eingefroren«, um höchstmögliche Kompatibilität und Stabilität von Dokumenten zu erreichen. LaTeX-Dokumente, so die Idee, sollten auch nach etlichen Jahren problemlos übersetzt werden können und exakt dieselben Ergebnisse reproduzieren.

Dem LaTeX-Team ist natürlich bewusst, dass sich die Anforderungen an den Textsatz ändern. Deshalb pflegt das Team einerseits die stabile Version von LaTeX2 ϵ , entwickelt diese aber auch weiter. Der Name dieser Weiterentwicklung ist LaTeX3.

LaTeX3 wird in verschiedenen Modulen (l3kernel, l3packages, l3experimental, l3trial) entwickelt, die auch unter LaTeX2 ϵ schon benutzt werden können. Marco Daniel hat in der \LaTeX DTK 2/2012 in einem Artikel über das LaTeX-Paket xparse gezeigt, wie LaTeX3 mit LaTeX2 ϵ benutzt werden kann.

Ein Beispiel für ein Paket, das auf LaTeX3-Funktionalität zurückgreift, ist fontspec. Es verwendet OpenType-Schriftfamilien mit LaTeX. Übersetzt man das folgende Beispieldokument, sieht man in den Logausgaben viele Verweise auf LaTeX3-Pakete:

```
\documentclass{article}
\usepackage{fontspec}
\setmainfont{Linux Libertine O}
\begin{document}
Dieser Text ist mit Linux Libertine gesetzt.
\end{document}
```

Ausgabe (Auszug)

```
[...]
(/opt/texlive2012/texmf-dist/tex/latex/l3kernel/exp13.
  sty
(/opt/texlive2012/texmf-dist/tex/latex/l3kernel/l3names.
  sty
(/opt/texlive2012/texmf-dist/tex/latex/l3kernel/l3
  bootstrap.sty
[...]
(/opt/texlive2012/texmf-dist/tex/latex/l3packages/
  xparse/xparse.sty)
```

Der *normale Anwender* wird LaTeX3-Pakete benutzen, ohne es zu merken. Er profitiert nur davon, dass viele Dinge mit LaTeX3 einfacher zu entwickeln sind als mit bisherigen Methoden und dadurch die eingebundenen Pakete mehr Möglichkeiten bieten, wie im Beispiel oben mit dem Paket `fontspec`.

14.1 *Wie kann ich es benutzen?*

Die Grundlage der Benutzung von LaTeX3 wird in der Dokumentation `exp13` erörtert. Um stets die neueste Version zu nutzen, bietet sich der Clone der Github-Repo an: [LaTeX3 auf Github](#). Dieses Dokument verweist auf die Art und Weise der LaTeX3-Programmierung. Die Erläuterung der Module ist der Dokumentation „`interface3`“ zu entnehmen.

Die Grundidee der LaTeX3-Programmierung

Schnittstelle

Die Entwickler von LaTeX3 sehen die Nutzung von LaTeX auf drei Ebenen, die nachstehend aufgelistet sind. Die Erläuterung ist der Dokumentation `expl3` detailliert zu entnehmen.

1. **Document mark-up**: Anpassung in der „.tex“-Datei
2. **Designer interface**: Anpassung durch den tyographischen Gestalter
3. **Programmer interface**: Implementierung der Basisfunktionen

Die nachstehenden Ausführungen beziehen sich auf das **Programmer interface**

Das Namensschema

Dieser Abschnitt soll kurz die Besonderheiten der in erster Linie merkwürdig erscheinenden Befehlsnamen erläutern. Häufig sieht man etwas wie `\l_tl_new:N`. Bisher sind Macros durch `@` und evtl. kryptischen Erweiterungen wie `@ii` als internes Macro gekennzeichnet, wobei es hier keine Konventionen gibt. LaTeX3 hat sich als Ziel gesetzt, eine klare und eindeutige Befehlsnamenstruktur einzuführen. Hierzu werden interne Makros wie bisher mittels `\` eingeleitet, aber nicht das `@` sondern `_` und `:` sind die Bestandteile des Befehlsnamen. Zur Vereinheitlichungen gibt es zudem einen *vorgeschriebenen, Befehlsnamenaufbau.

14.2 Weblinks

Webseiten zu LaTeX3 sind fast ausschließlich in englischer Sprache verfügbar.

- Die LaTeX3 News erscheinen in der Regel halbjährlich:
☞ <http://www.latex-project.org/l3news/>.

15 *TeX und LaTeX Glossar*

Biber	Programm, um Literatureinträge zu sortieren. Wird mit BibLaTeX verwendet
BibLaTeX	Makropaket für LaTeX, das Literatureinträge darstellt
BibTeX	Programm, um Literatureinträge zu sortieren und nach LaTeX zu formatieren
ConTeXt	Alternatives Format zu LaTeX. Im Gegensatz zu LaTeX besteht es aus einem relativ umfangreichen Kern und wenig Zusatzpaketen. ConTeXt ist deutlich stärker darauf ausgerichtet, komplexe Layouts umzusetzen. Einfaches Beispiel in: ☞Mein erstes ConTeXt-Dokument.
Engine	Programm, das TeX ausführen kann. Beispiele sind PDFTeX, XeTeX, LuaTeX und das „original Knuth TeX“, das nur DVI erzeugen kann. Man benutzt eine Engine nicht alleine, sondern immer in Kombination mit einem Format
Format	Befehlssatz für eine Engine. Beispiele sind plain, LaTeX und ConTeXt
LaTeX	Befehlssatz für TeX (Format), das am weitesten verbreitet ist
LuaTeX	Nachfolger von PDFTeX mit eingebettetem ☞Lua-Interpreter. Kann mit Unicode und OpenType-Fonts umgehen
LyX	Editor für LaTeX, dessen Besonderheit es ist, dass man sofort eine graphische Ausgabe sieht.
MkII, MkIV	Versionsangaben von ConTeXt (Mark II = mit pdfTeX oder XeTeX, Mark IV = mit LuaTeX)
plain TeX	Einfaches Befehlssatz für TeX von Knuth, entri-

16 Was ist das Dateiformat DVI und wie kann ich es anzeigen?

DVI (das steht für »device independent«) ist das ursprüngliche Ausgabeformat von TeX und wurde ausschließlich in der TeX-Welt benutzt, da andere Formate wie PDF oder PostScript noch gar nicht existierten oder noch nicht genügend Verbreitung fanden.

Eine DVI-Datei enthält die fertigen Satzinformationen für die Ausgabe (wo werden welche Elemente platziert), aber keinerlei Ressourcen wie Schriftarten, Bilder oder ähnliches. Daher kann eine DVI-Datei nicht für sich alleine stehen, es werden mindestens immer die zugehörigen Schriftdateien benötigt. Daher hat sich dieses Format nie als Austauschformat durchgesetzt.

Auf Unix-Rechnern bietet sich das Programm `xdvi` bzw. `kdvi` und Okular (KDE) an, für Windows werden Vorschau-Programme mit den TeX-Distributionen mitgeliefert: `Yap` in MiKTeX, `dviout` in TeX Live. Auf Mac OS X kann Skim durch automatische Konvertierung nach PDF DVI-Dateien anzeigen.

Üblicherweise wurden die DVI-Dateien nach PostScript gewandelt, das alle Ressourcen enthält. Ein Nachteil dieses Formats ist aber die geringe Verbreitung von Anzeigeprogrammen. Mittels PostScript-nach-PDF-Konvertierungsprogrammen, bzw. PDFTeX seit 1997/98, hat sich die Situation deutlich geändert. Ab dem Zeitpunkt konnte man dank PDFTeX (und dadurch LaTeX und ConTeXt) Dokumente einem großen Publikum zur Verfügung stellen, da die Verbreitung von PDF-Anzeigeprogrammen sehr groß ist. DVI ist daher heute nicht mehr das bevorzugte Ausgabeformat, grundsätzlich empfiehlt sich die Ausgabe nach PDF.

17 Wozu dienen Dateien mit der Namensendung .xyz?

17.1 Dateien, die vom Benutzer erstellt werden

- .tex Datei, die den eigentlichen Text enthält
- .bib Quelle für das Literaturverzeichnis im BibTeX Format
- .ist Indexstil für Makeindex. Enthält Anweisungen, wie aus den Einträgen in der .idx/.glo'-Datei die .ind/.gls-Datei generiert werden soll.

17.2 Dateien, die LaTeX erzeugt

- .toc Enthält das Inhaltsverzeichnis.
- .log Protokolldatei. Enthalten sind alle Ausgaben von LaTeX, inklusive Warnungen und Fehlermeldungen.
- .aux Automatisch erzeugte Datei mit Querverweisen.
- .dvi Alternatives Ausgabeformat zu PDF. Der ursprüngliche Standard von TeX, wird aber heutzutage fast nicht mehr verwendet, da es kaum Software gibt, um DVI Dateien zu betrachten.
- .pdf Gängiges Ausgabeformat von PDFTeX, XeTeX und LuaTeX.
- .lot Liste der Tabellen (list of tables).
- .lof Liste der Abbildungen (list of figures).

17.3 Sonstige Dateien

- `.dtx` docstrip-Archivdatei, kann mit dem in TeX geschriebenen Programm `docstrip` entpackt werden. (docstrip: ↗CTAN:macros/latex/base/docstrip.dtx und ↗CTAN:macros/latex/base/docstrip.ins)
- `.cls` Klassendateien, z.B. `article.cls` für die Standardklasse „article“.
- `.sty` LaTeX Pakete, die mit `\usepackage{⟨Paketname⟩}` eingebunden werden.
- `.enc` Encoding Datei, die die 256 Zeichen einer Fontdatei mit Postscript Namen verbindet.
- `.map` Map-Datei mit Anweisungen, welcher TeX-Name welcher physikalischen Schriftdatei entspricht.

17.4 Schriftdateien

- . t t f TrueType oder OpenType Fonts im TrueType Format (vektorbasiert)
- . o t f OpenType Font (vektorbasiert)
- . p f b PostScript Type1 Font (vektorbasiert)
- . a f m Zu . p f b gehörige Metrikdatei
- . m f Metafont Quelldateien
- . g f Enthält die Pixeldaten eines Meta-fonts in einer bestimmten Auflösung
- . p k Wie . g f, nur komprimiert
- . t f m TeX Fontmetrik Datei. Das ist die Datei, die das klassische TeX als einzige Fontdatei einliest. Erst im zweiten Schritt werden die eigentlichen Umrissdateien eingelesen (DVI-Treiber oder PDFTeX).
- . v f Virtueller Font. Enthält Metriken einer Schriftdatei, die aus mehreren echten Fonts zusammengebaut werden können.
- . v p l Property list eines virtuellen Fonts.
- . p l Property list, Ascii Repräsentation der . t f m-Datei. Kann mit t f t o p l erzeugt bzw. mit p l t o t f wieder in eine . t f m Datei konvertiert werden.

18 Was ist CTAN?

CTAN steht für »Comprehensive TeX Archive Network«. Dies ist ein Verbund von z. Z. zwei Rechnern im Internet. Hierauf ist fast alles zu finden, was mit TeX zu tun hat. Die Server gleichen ihre Dateien untereinander ab, so daß Sie auf jedem der CTAN-Server

dieselben Dateien finden können. Dadurch ist es gleichgültig, wo Sie die Dateien holen. Sie sollten aber bedenken, daß es am günstigsten ist, die Dateien bei dem Server zu holen, der Ihnen am nächsten ist. Dies ist für den deutschsprachigen Raum der von DANTE e.V. finanzierte ftp-Server oder einer der Spiegel-Server (mirror).

CTAN ist über <http://ctan.org/> zu erreichen, die Spiegel unter <http://mirror.ctan.org/>. Es ist hilfreich immer den zweiten Link zu verwenden, da man automatisch auf einen passenden Server weiter geleitet wird. Die Verweise in diesem Wiki werden alle über mirror.ctan.org erstellt.

Der Hauptknoten ist wie folgt zu erreichen:

Name	<code>ftp.dante.de</code>
Rechnerstandort	Köln
ftp	<code>ftp://ftp.dante.de/tex-archive</code>
rsync	<code>rsync.dante.ctan.org/CTAN/</code>
http	<code>http://ctan.org/</code>

19 Was ist TDS, die »TeX Directory Structure«?

Hauptfunktion von TDS ist die Festlegung einer Grundstruktur für ein TeX-System. In dieser kann TeX samt Hilfswerkzeugen, Makropaketen, Fonts etc. systematisch geordnet und damit für alle Installationen und Systeme gleich und einheitlich untergebracht werden. Somit können implementierungsunabhängige Teile beispielsweise auch über NFS-Mounts oder von CD-ROM für alle Systeme zur Verfügung gestellt werden. Die von TDS festgelegte

Grundstruktur ist offen, kann also jederzeit für weitere Applikationen erweitert werden.

Der TDS-Standard richtet sich primär an TeX-Systemadministratoren und Personen, die TeX-Distributionen zusammenstellen. Wenn Entwickler eines TeX-Pakets von einer generell gültigen Verzeichnisstruktur ausgehen können, kann auch die Installation eines Pakets leichter automatisiert oder zumindest mit sehr präzisen Anweisungen versehen werden. Für Endbenutzer ist eine solche wohldefinierte Systemstruktur ebenfalls von Interesse, wenn sie sich mit der Ausstattung des Systems, das sie benutzen, näher vertraut machen wollen.

Das TDS-Dokument kann über `texdoc` abgerufen werden (`texdoc tds`).

20 Wie installiert man am besten ein neues Paket, eine neue Klasse oder eine neue Schrift in einem bestehenden TeX-System?

Aktuelle TeX-Systeme legen alle Dateien in einer hierarchischen Struktur gemäß dem  TeX-Directory-Structure-Standard (TDS) an. Das Wurzelverzeichnis eines solchen Baumes lautet meist `texmf`. Unter `texlive` und anderen TeX-Systemen werden die Suchpfade in der Datei `texmf.cnf` angegeben, deren Ort man mit `kpsewhich texmf.cnf` herausfinden kann. Für nachträglich installierte Pakete empfiehlt es sich, diese in einer eigenen lokalen Verzeichnishierarchie zu installieren (beispielsweise unter dem Verzeichnis `texmf.local`, siehe Variable `TEXMFLOCAL` in `texmf.cnf`). Besteht ein LaTeX-Paket nach dem eventuell

notwendigen Entpacken der dtx-Dateien aus einer Einzeldatei, so wird diese in der TDS-Hierarchie in `texmf.local/tex/latex/misc/` abgelegt. Gehören zu dem Paket mehrere sty-, cls-, fd- oder ähnliche Dateien, so werden diese in ein neu erzeugtes Verzeichnis unter `texmf.local/tex/latex/Paketname` verschoben. Schriften, Dokumentation, BibTeX-Dateien, dvips-Konfigurationsdateien u. ä. werden analog gemäß TDS in entsprechende Unterverzeichnisse unter `texmf.local` verschoben. Näheres sollte in der README-Datei des Pakets erläutert sein. Zum Abschluss darf man nicht vergessen, die sogenannte Filename-Datenbank auf den aktuellen Stand zu bringen, da sonst die neu installierten Dateien nicht gefunden werden.

Einige TeX-Systeme bieten ein Konfigurations-Werkzeug, um zusätzliche Pakete nachträglich zu installieren oder vorhandene Pakete zu aktualisieren. Unter MikTeX verwendet man das Options-Tool. Das sorgt nach der Änderung automatisch für die Aktualisierung der Dateinamens-Datenbank.

21 Soll ich lieber KOMA-Script oder Memoir nehmen?

(Auszug aus einer Frage von tex.stackexchange.com.)

KOMA-Script und Memoir sind beides Dokumentklassen, die alternativ zu den Standardklassen `article`, `book` bzw. `report` benutzt werden können. Aber warum sollte man eine andere Klasse benutzen und dann lieber KOMA-Script oder memoir?

21.1 *Pro memoir*

- memoir hat deutlich mehr Funktionalität eingebaut als die Standardklassen, bzw. KOMA-Script
- Die Dokumentation von memoir geht auf deutlich mehr Themen ein als KOMA-Script
- Die Dokumentation von memoir enthält mehr Beispiele
- Weiter verbreitet im englischsprachigem Raum, mehr Hilfe in englischsprachigen Foren

21.2 *Pro KOMA-Script*

- Im deutschsprachigem Raum mehr verbreitet
- Gute Unterstützung durch den Autor
- Weniger ist mehr: weniger Grundfunktionen als Memoir, dafür höhere Flexibilität durch die gewohnten LaTeX Zusatzpakete
- Aufgeteilt in mehrere Pakete, man kann die KOMA-Script Funktionalität teilweise auch mit anderen Klassen nutzen
- Enthält eine Klasse für Briefe

22 *Wie ändere ich die Ränder meines Dokuments?*

Die Seitenränder in LaTeX und in TeX lassen sich mit Bordmitteln nicht ganz so leicht ändern wie man es erwarten würde. Abhilfe

gibt es in dem Paket `geometry` und mit der Dokumentklasse KOMA Script.

Bei der Einstellung der Ränder sollte man beachten, dass der Text nicht zu breit läuft. Die Empfehlung ((FIXME) Zitieren!) liegt bei ca. 60-70 Zeichen je Zeile, was im Widerspruch zu den vermeintlich großen Rändern steht, die bei einem schmalen Textblock naturgemäß auftreten.

22.1 *Geometry-Paket*

Ohne viele Worte zu verlieren, hier ein Beispiel für die Benutzung des Pakets `geometry`:

```
\documentclass[a4paper,ngerman]{article}
\usepackage{layout} % um die Seitenränder als Bild
    auszugeben
\usepackage{geometry}

\geometry{
  left=2cm,
  textwidth=8cm,
  marginpar=3cm}

\begin{document}
  \layout
\end{document}
```

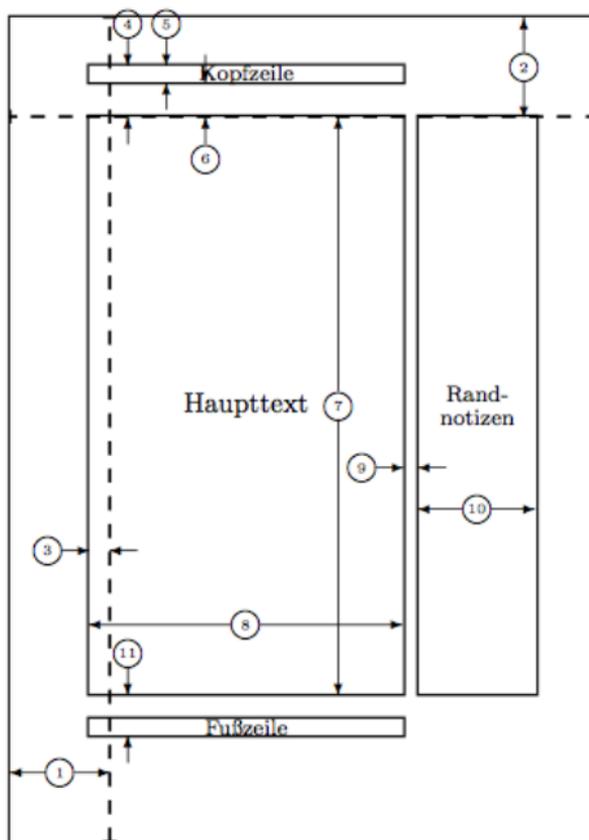
Wichtig ist die Angabe des Papierformats beim Laden der Dokumentklasse (hier `a4paper`). Die Voreinstellung ist `letterpaper`, mit den Seitenrändern für eine Seite mit der Größe 8,5×11 Zoll, auch wenn die PDF-Seitengröße 210×297 mm (A4) ist.

Der Befehl `\layout` aus dem gleichnamigen Paket zeigt eine graphische Darstellung der Seitenränder. Das ist sehr praktisch,

um die Angaben für die Seitenränder zu überprüfen.

Das Paket `geometry` ist recht komplex und bietet viele Einstellungsmöglichkeiten, daher werden hier nur ein paar grundlegende Einstellungen gezeigt. Eine ausführliche Anleitung gibt es mit `texdoc`.

Die Ausgabe aus obigem Quelltext:



1	ein Zoll + \hoffset	2	ein Zoll + \voffset
3	\oddsidemargin = -15pt	4	\topmargin = -37pt
5	\headheight = 12pt	6	\headsep = 25pt
7	\textheight = 418pt	8	\textwidth = 227pt
9	\marginparsep = 11pt	10	\marginparwidth = 85pt
11	\footskip = 30pt		\marginparpush = 5pt (ohne Abbildung)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 421pt		\paperheight = 597pt

Die Anweisungen für die Seitenränder kann an auch beim Laden des Pakets mit angeben. Äquivalent zu dem Beispiel ist:

```
\usepackage[left=2cm,textwidth=8cm,marginpar=3cm]{
  geometry}
```

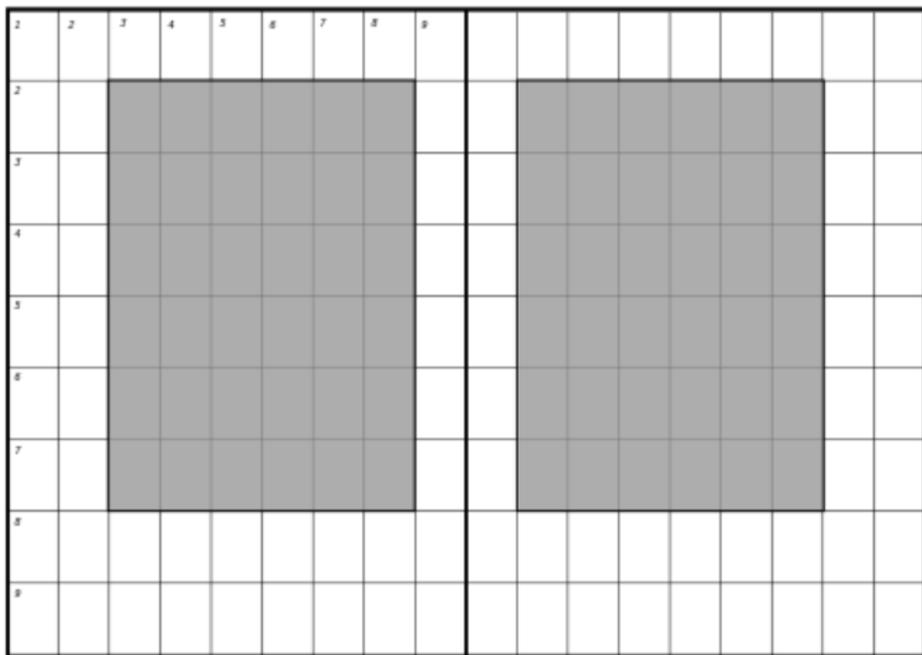
Die Seitenränder lassen sich mit den Parametern `left`, `right`, `top` und `bottom` verändern. Mit den Optionen `includehead` bzw. `includefoot` beziehen sich die Angaben für `top` und `bottom` auf den Abstand zwischen dem oberen Rand der Kopfzeile und der Oberkante des Papiers bzw. dem unteren Rand der Fußzeile und der Unterkante des Papiers.

Die Textbreite und Texthöhe lassen sich mit den Parametern `textwidth` und `textheight` verändern. Die Texthöhe kann man ebenfalls in *Anzahl der Zeilen* angeben mit `lines=32` z.B. Diese und viele weitere Einstellungsmöglichkeiten finden sich im Kapitel 5 des Pakets `geometry`. Gibt man zu viele (widersprüchliche) Parameter an, meckert das Paket:

```
Package geometry Warning: Over-specification in `h'-  
direction.  
`width' (227.62204pt) is ignored.
```

22.2 KOMA-Script

Die Klassen aus dem KOMA-Script-Familie bieten zwei verschiedene Möglichkeiten an, die Seitenränder zu verändern. Die erste Methode besteht in der *Satzspiegelkonstruktion* über eine Flächeneinteilung, im Beispiel ist es eine 9er Teilung:



Im Prinzip wird die Seite horizontal und vertikal in n gleiche Teile eingeteilt und in einem doppelseitigem Layout werden innen und oben ein Teil und unten und außen zwei Teile als Rand genommen während der Rest den Textblock ergibt. Das Beispiel aus dem Bild würde so erzeugt werden:

```
\documentclass[twoside,DIV=9]{scrartcl}
```

Wenn man das fertige Dokument ausdruckt und binden lässt, »verschwindet« meist ein Teil der Fläche mit der Bindung. Diesen Wert kann man bei KOMA-Script zusätzlich angeben:

```
\documentclass[twoside,DIV=9,BCOR=5mm]{scrartcl}
```

Hier wird vor der Satzspiegelberechnung 5mm des Papiers für Bindung reserviert.

Alternativ zur Einteilung des Satzspiegels wie oben lässt sich der Befehl `\areaset` nutzen:

```
\documentclass{scrartcl}
\areaset{10cm}{15cm}
```

erzeugt einen Textblock mit den Maßen 10×15 cm. Die Bindekorrektur kann in eckigen Klammern angegeben werden: `\areaset[5mm]{10cm}{15cm}`.

Die Maße der Ränder werden dann automatisch so gut wie möglich eingestellt. Ein Satzspiegel ähnlich dem oben gezeigten Beispiel mit `geometry` könnte z.B. so verwirklicht werden:

```
\usepackage{typearea}
\areaset{8cm}{14.7cm}
\setlength{\marginparwidth}{3cm}
```

Im `scrguide` wird dies in den Kapiteln 2 und 15 sehr umfassend behandelt; ebenso im Buch »KOMA-Script« Markus Kohm und Jens-Uwe Morawski, hier sind es in der 4. Auflage die Kapitel 2 und 16 (bzw. 2 und 13 in der 3. Auflage).

Hinweise:

1. Will oder muss man konkrete Vorgaben zum Satzspiegel umsetzen, kann man die Standardklassen von KOMA-Script auch in Verbindung mit dem Geometry-Paket benutzen.
2. Das Paket `typearea` kann man auch ohne KOMA-Script einbinden.

23 Auf welche Weisen kann man eine Seite beenden bzw. umbrechen?

Sowohl `\newpage`, `\pagebreak` als auch `\clearpage` leiten eine neue Seite ein. Was ist der Unterschied zwischen diesen Befehlen? Wann soll ich welchen verwenden?

23.1 `\newpage`

Beendet eine Seite oder in `twocolumn`-Dokumenten die aktuelle Spalte *sofort* und füllt sie jeweils mit Weißraum auf. Der auf `\newpage` folgende Text landet auf der Folgeseite.

Bemerkung: Wartet noch eine Gleitumgebung auf die Gelegenheit, ausgegeben zu werden, kann diese am Beginn der Seite vor dem Folgetext landen.

23.2 `\clearpage`

Sorgt dafür, dass eventuelle Gleitumgebungen (`figure` und `table`), die noch nicht ausgegeben wurden, allesamt ausgegeben werden. Dafür werden so viele Seiten eingefügt wie nötig. Dann wird die Seite umgebrochen.

Bemerkung: Nach der letzten Gleitumgebung ist definitiv ein Seitenumbruch. Der Folgetext steht *immer* am Beginn einer neuen Seite.

23.3 `\cleardoublepage`

Wie `\clearpage`, doch bei doppelseitigen Dokumenten beginnt der Folgetext auf einer ungeraden Seite, also auf einer rechts liegenden Seite. Wenn nötig, wird eine leere linke (geradzahlige) Seite eingefügt.

23.4 `\pagebreak`

Gibt einen möglichen Seitenumbruch an. Dabei wird, wenn die Seite umbrochen wird, der Inhalt der Seite so verteilt, dass sie am unteren Rand bündig ist. Dafür wird, falls nötig, vorhandener vertikaler Weißraum gestreckt, wenn das Dokument mit `\flushbottom` gesetzt wurde, wie mit der `book`-Klasse.

In jedem Fall erfolgt der Seitenumbruch erst *nach der aktuellen Zeile*.

Die Dringlichkeit des Seitenumbruchs kann mit einem optionalen Argument angegeben werden, das einer von fünf Werten sein kann:

1. 0: Hier ist gut, muss aber nicht umgebrochen werden (`penalty: 0`),
2. 1: Hier wäre es ganz schön, aber Du findest bestimmt was Besseres (`penalty: -51`, eigentlich `-\@lowpenalty`)
3. 2: Hier wäre es schon ganz passend, die Seite umzubrechen (`penalty: -151`, eigentlich `-\@medpenalty`)
4. 3: Hier würde ein Seitenumbruch richtig gut passen, aber

wenn Du was *noch* besseres weißt... (penalty: -301, eigentlich - \@highpenalty)

5. 4: Mache auf jeden Fall hier einen Seitenumbruch (penalty: -10000, Voreinstellung)

23.5 Zusammenfassender Praxis-Tipp

`\clearpage` ist häufig das sinnvollste:

- Aufgelaufene Grafiken werden ausgegeben im Gegensatz zu den anderen.
- Es wird nicht mit größeren vertikalen Abständen gedehnt (kann eher störend als angenehm sein).

In doppelseitigen Dokumenten trifft das dann natürlich auf `\clearpage` und `\clearpage` zu.

23.6 Quelle

Clemens und Herbert auf www.texwelt.de: Was ist der Unterschied zwischen `\newpage`, `\pagebreak` und `\clearpage`?

24 ***Wieso ist mit der Klasse »book« oder mit der Klassenoption »twoside« der innere Rand kleiner als der äußere?***

Bei doppelseitiger Formatierung werden die Seitenränder so gewählt, dass sie nicht für eine einzelne Seite sondern für eine auf-

geschlagene Doppelseite passen. Bei dieser bilden die Ränder der beiden Seiten drei vertikale weiße Streifen (links außen, mittig, rechts außen), die gleich breit sein sollen. Da der mittlere Streifen aus den Rändern zweier Seiten gebildet wird, sind diese auch nur halb so breit. Aus diesem Grund legt man unter LaTeX die Werte für `\evensidemargin` und `\oddsidemargin`, also für den linken Rand der geraden (linken) und den der ungeraden (rechten) Seite, entsprechend fest. Näheres findet sich u. a. in der Dokumentation der KOMA-Script-Klassen.

25 Wie ändere ich die Schriftart in meinem Dokument?

Es gibt mehrere Möglichkeiten, die Schriftart in einem Dokument zu ändern. Für manche Schriftarten werden Pakete gepflegt, die über `\usepackage` eingebunden werden können. Eine Übersicht über diese Pakete findet sich im [LaTeX Font Catalogue](#). Dort steht auch jeweils, wie die Schrift eingebunden werden kann.

Wenn statt einer Serifenschrift eine serifenlose Schrift (Serifen sind Schnörkel an Buchstaben) als Standard verwendet werden soll, kann der Befehl

```
\renewcommand{\familydefault}{\sfdefault}
```

verwendet werden.

25.1 XeLaTeX und LuaLaTeX

Bei der Verwendung von XeLaTeX und LuaLaTeX kann jede im System installierte Schriftart verwendet werden. Hierfür dienen die Befehle `\setmainfont`, `\setsansfont` und `\setmonofont`:

```
\setmainfont[Mapping=tex-text]{Linux Libertine O}  
\setsansfont[Mapping=tex-text]{Source Sans Pro}  
\setmonofont[Mapping=tex-text]{Source Code Pro}
```

Hierfür muss man das Paket `fontspec` geladen haben.

25.2 Mathematikschriften

Nicht für jede Schrift stehen passende Mathematik-Schriften zur Verfügung. Einen Überblick, zu welcher Schriftart welcher Mathefont wie genutzt werden kann, siehe die Übersicht unter http://tug.ctan.org/pub/tex-archive/info/Free_Math_Font_Survey/survey.html

26 *Wie hebt man Wörter im Text hervor?*

Wenn man im Fließtext bestimmte Sachen wie Schlüsselwörter oder Namen hervorheben möchte, bieten sich z.B. Kapitälchen (Großbuchstaben), Kursiv, oder Fettdruck an.

Sollte man im Fließtext dafür Schrift-Befehle wie `\textsc`, `\textit` oder `\textbf` verwenden? **Nein.**

LaTeX fördert in starkem Maße die Trennung von Form und Inhalt. Das bedeutet, dass innerhalb des Dokuments, also zwischen `\begin{document}` und `\end{document}` nicht mehr festgelegt wird, wie etwas aussehen soll, sondern was für eine Art von Text etwas ist. Ein gutes Beispiel dafür sind Überschriften. In LaTeX sollte man eine Überschrift niemals als

```
{\Large\textbf{Überschrift}}\par
```

schreiben, sondern immer mit einer Anweisung wie `\chapter`, `\section` etc.

Dieses Prinzip sollte man auch auf eigene Textarten, also auch auf unterschiedliche Arten von Begriffen ausweiten. Für die in der Frage genannten Beispiele sollte man also Definition wie:

```
\newcommand*{\Name}[1]{\textsc{#1}}  
\newcommand*{\Befehl}[1]{\texttt{#1}}  
\newcommand*{\Fachbegriff}[1]{\emph{#1}}
```

in der Dokumentpräambel einfügen und dann diese neuen LaTeX-Befehle im Text verwenden.

Nun mag der Einwand kommen, dass `\emph` doch bereits eine semantische Auszeichnung sei, man also Fachbegriffe auch im Dokument selbst als `\emph{Fachausdruck}` hervorheben könne, ohne erst eine LaTeX-Anweisung `\Fachbegriff` zu definieren. Dem ist entgegen zu halten, dass `\emph` lediglich sagt: »Hebe hervor«. Es gibt hingegen keinen Grund für die Hervorhebung an. So könnten beispielsweise neben Fachbegriffen auch die Namen lediglich mit `\emph` hervorgehoben werden. Dann müsste man nach derselben Logik, nach der die Fachbegriffe nur direkt per `\emph{...}` statt per `\Fachbegriff{...}` hervorgehoben würden, auch die Namen direkt per `\emph{...}` markiert

werden. Am Ende hat man dann wieder unterschiedliche Arten von Begriffen auf dieselbe Weise ausgezeichnet. Die Anweisung `\emph{...}` würde ich deshalb als *schwache semantische Auszeichnung* bezeichnen. Daher sollte man in allen Fällen, in denen sich der Grund für die Auszeichnung benennen lässt, wie am Beispiel erklärt verfahren. Der Befehl zur Auszeichnung sollte also den Grund für die Auszeichnung widerspiegeln und möglichst nicht nur, dass eine Auszeichnung erfolgen soll.

Ein großer Vorteil dieser Trennung von Form und Inhalt ist, dass zur Änderung der Form nicht der gesamte Text nach entsprechend markierten Texten durchsucht werden muss, um dort die Markierung zu ändern. Stattdessen muss nur die Definition der Form geändert werden. Sollen also Name nun doch nur in der Standardhervorhebung markiert werden und beispielsweise Fachausdrücke auch in kursivem Text kursiv gesetzt werden, so genügt eine Änderung dieser beiden Definitionen:

```
\newcommand* {\Name} [1] {\emph{#1}}  
\newcommand* {\Befehl} [1] {\texttt{#1}}  
\newcommand* {\Fachbegriff} [1] {\textit{#1}}
```

Es lohnt sich daher, die Trennung von Form und Inhalt konsequent fortzuführen und einfache Auszeichnungsbefehle wie die Befehle zur Umschaltung der Schriftattribute innerhalb des eigentlichen Inhalts nicht zu verwenden.

Zusammenfassen könnte man dies mit dem Satz: *Befehle zur Textauszeichnung innerhalb des Dokuments sollen nicht nach dem Ergebnis der Auszeichnung, sondern nach den Grund der Auszeichnung unterscheiden.*

Verfasst von saputello auf [TeXwelt.de](https://www.texwelt.de): [Wie sollte ich vorgehen,](#)

wenn ich bestimmte Arten von Begriffen, beispielsweise Namen, hervorheben will?.

27 Was ist der Unterschied zwischen `\emph{}` und `\textit{}`?

In der PDF-Ausgabe sehen zunächst die Ergebnisse beider Befehle gleich aus.

`\textit{<text>}` erzeugt kursiven Text, ist also ein Schriftformatierungsbefehl. `\emph{<text>}` hebt Text hervor (engl. *emphasize* – hervorheben), ist also ein semantischer Markup-Befehl. Obwohl die Hervorhebung in der Voreinstellung ebenfalls kursiver Text ist, lässt sich das prinzipiell ändern. Z.B. definiert das Paket `ulem` den Befehl um, so dass er seinen Inhalt unterstreicht (bitte vermeiden!).

Während man zur Hervorhebung von Textteilen also unbedingt `\emph` verwenden kann, sollte man die Verwendung von `\textit` auf Definitionen eigener Befehle beschränken, wenn man nicht explizit das semantische Markup „kursiver Text“ möchte.

Ein wesentlicher Unterschied von `\emph` im Gegensatz zu `\textit` ist außerdem, dass er auch in Verschachtelung funktioniert, und Hervorhebungen in hervorgehobenem Text beispielsweise aufrecht setzt:

```
\textit{kursiver Text mit einem \textit{kursiven} Teil}
\emph{hervorgehobener Text mit einem \emph{
    hervorgehobenen} Teil}
```

kursiver Text mit einem kursiven Teil

hervorgehobener Text mit einem hervorgehobenen Teil

Man könnte natürlich sagen, dass auch `\textit` in Verschachtelung funktioniert: es setzt kursiven Text in kursivem Text!

Verfasst von Clemens auf www.TeXwelt.de: Was ist der Unterschied zwischen `\emph{}` und `\textit{}`?

28 Wie ändere ich die Schriftgröße in meinem Dokument?

Die Schriftgröße kann man entweder für den gesamten Text im Dokument verändern (Abschnitt »Schriftgröße dauerhaft ändern«) oder nur kurzzeitig für bestimmte Wörter (siehe unten »Schriftgröße vorübergehend ändern«).

Will man die Schriftgröße aller Fußnoten, Bildunterschriften oder Überschriften um eine Stufe kleiner oder größer wählen, so ist es besser die Makros für Fußnoten, Unterschriften bzw. Überschriften entsprechend abzuändern. Dies kann sehr einfach mit Paketen wie »footmisc«, »caption2«, »titlesec« oder »sectsty« erfolgen.

28.1 Schriftgröße dauerhaft ändern

Standardklassen

Die Standardklassen kennen die Schriftgrößen 10pt, 11pt und 12pt. D.h. wenn man beispielsweise die Klasse `article` lädt mit

folgender Option:

```
\documentclass[9pt]{article}
```

gibt es folgende Warnung:

```
LaTeX Warning: Unused global option(s):  
[9pt].
```

Die Dokumentklassen aus der `extsizes`-Familie bieten die Optionen `8pt`, `9pt`, `10pt`, `11pt`, `12pt`, `14pt`, `17pt` und `20pt`. Die Benutzung ist ganz einfach, anstelle der Klasse `article` einfach `extarticle` (bzw. `extreport`, `extbook`, `extletter` oder `extproc` für die anderen Klassen) laden:

```
\documentclass[17pt]{extarticle}  
\usepackage[T1]{fontenc}  
\usepackage[utf8]{inputenc}  
\begin{document}
```

```
\section{Einleitung}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52.

```
\end{document}
```

KOMA-Script

Bei der Verwendung von KOMA-Script-Klassen kann die Schriftgröße global über das Argument `fontsize` geändert werden, zum Beispiel:

```
\documentclass[fontsize=12pt]{scrartcl}
```

als Größe (anstelle von 12pt) kann man jede beliebige TeX Einheit nehmen. Wenn die Einheit weggelassen wird, wird pt angenommen.

28.2 Schriftgröße vorübergehend ändern

Befehl *fontsize*

Eine bestimmte Schriftgröße stellt man mit dem Befehl `\font-size` ein. Das erste Argument gibt die Schriftgröße an, das zweite den Zeilenabstand. Wichtig ist aber, dass eine Schrift geladen wird, die auch die entsprechenden Größen verarbeiten kann. Ansonsten gibt es eine Warnung:

```
\documentclass[a4paper,ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
```

```
\fontsize{15pt}{17pt}\selectfont
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf.

```
\end{document}
```

```
LaTeX Font Warning: Font shape `T1/cmr/m/n' in size <15>
not available
```

```
(Font) size <14.4> substituted on input line 8.
```

```
[1{/opt/texlive2012/texmf-var/fonts/map/pdftex/updmap/
pdftex.map}] (./test.aux)
```

LaTeX Font Warning: Size substitutions with differences (Font) up to 0.6pt have occurred.

Mit einer passenden Schriftart (und einer passenden FD-Datei) geht das problemlos. Im Falle der Computer Modern reicht es auf die Type1-Variante »Latin Modern« zurückzugreifen: `\usepackage{lmodern}`.

Befehle `\large` & Co.

Alternativ kann man die Befehle `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge` und `\Huge` verwenden, mit denen einzelne Worte, Sätze, Bereiche größer oder kleiner gesetzt werden.

Es ist wichtig, die Befehle mit `{` und `}` zu Gruppieren (= deren Wirkung am Ende der Gruppe wieder aufzuheben):

```
\documentclass[a4paper,ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und `{\large` dazwischen} ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon `{\Huge` vielversprechend} lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis `{\tiny` Oberhausen}. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum

gleichmäßigen Rollen zwingen. Es wird links überholt,
es wird rechts überholt.

Es wird gar nichts mehr, alles steht und macht lange
Gesichter.

```
\end{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen
in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus
Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch
kurz auf die A 52. Schon **vielversprechend** lebhaft. Hinter
dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen.
Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Ge-
schwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links
überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht
lange Gesichter.

Eine kleine Stolperfalle gibt es noch: Es macht einen Unterschied,
ob das Absatzende innerhalb der Schriftumschaltung erfolgt oder
nicht:

```
\documentclass[a4paper,ngerman]{article}  
\usepackage{babel}  
\usepackage[T1]{fontenc}  
\usepackage[utf8]{inputenc}  
\begin{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine
Familie lebt dagegen in
Hamburg. Und dazwischen ich, aber ganz cool.
Vollbremskombination aus Ampel
oder Einfädeln oder beides auf einmal. Geht nur mit
Tricks. Eben noch kurz auf
die A 52. Schon `\huge` vielversprechend lebhaft. Hinter
dem Breitscheider Kreuz geht
es richtig los. Stau auf der A 3 bis Oberhausen.)

```
\end{document}
```

ergibt folgende Ausgabe:

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon **vielversprechend lebhaft**. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen.

während ein scheinbar harmloses `\par` (das ist dasselbe wie eine Leerzeile) am Ende eine große Auswirkung hat:

```
\documentclass[a4paper,ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
```

```
Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine
  Familie lebt dagegen in
Hamburg. Und dazwischen ich, aber ganz cool.
  Vollbremskombination aus Ampel
oder Einfädeln oder beides auf einmal. Geht nur mit
  Tricks. Eben noch kurz auf
die A 52. Schon {\huge vielversprechend lebhaft. Hinter
  dem Breitscheider Kreuz geht
es richtig los. Stau auf der A 3 bis Oberhausen.\par}

\end{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen.

Das Paket relsize

Mit dem Paket `relsize` erreicht man ähnliches wie oben mit den Befehlen `\large` & `co`. So bewirkt der Befehl `\relsize{-2}` aus dem Paket, dass die aktuelle Schriftgröße um zwei Stufen verringert wird. Dasselbe macht `\smaller[2]` bzw. in der anderen Richtung `\larger[2]`. Auch hier müssen die Befehle durch Klammern (`{` und `}`) gruppiert werden. Alternativ bietet das Paket die Befehle `\textlarger[<i>]{<Text>}` bzw. `\textsmall[<i>]{<Text>}`, mit der der Text im Argument um `i` Stufen größer oder kleiner gesetzt wird.

29 Ich benötige ein spezielles Symbol in meinem Dokument. Wie kann ich es finden?

Meist kann man nur in einem Handbuch nach dem Befehl suchen, der ein spezielles Symbol erzeugt. Hilfreich hat sich das Dokument

»Comprehensive LaTeX symbols list« erwiesen, das unter <http://texdoc.net/pkg/symbols-a4.pdf> oder mithilfe der Kommandozeile `texdoc symbols-a4` zu lesen ist.

Daneben gibt es noch einen prima Webdienst, [Detextify](#). Hier zeichnet man mit der Maus auf einer Webseite ein Symbol, das man sucht und bekommt den Befehl zurück.

Detextify² - LaTeX symbol classifier

[classify](#) [symbols](#) [blog](#)

Draw here!

Did this help? Helping Detextify make money allows us to pay our servers and make better software for you. [DONATE](#) Amount: 11.111.546€

\varpropto Search ID: 11689628773080774 `\varpropto` method: `mathmode`

\propto Search ID: 14774281404811937 `\propto` method: `mathmode`

α Search ID: 16180808081081988 `\alpha` method: `mathmode`

\mathscr{B} Search ID: 101822775037740 `\mathscr{B}` method: `mathmode`

\prec Search ID: 12349658675582636 `\prec` method: `mathmode`

[What is this?](#)

[How do I use it?](#)

up ↑

30 Ich bekomme die Meldung `Overfull \hbox` – was kann ich tun?

TeX erzeugt manchmal Warnungen der Art:

Overfull `\hbox` (0.21693pt too wide) in paragraph at lines 14–22
`\OT1/cmr/m/n/10` geht es richtig los. Stau auf

Was ist das und was kann ich dagegen tun?

Normalerweise deutet das darauf hin, dass TeX eine Zeile nicht richtig umbrechen kann, ohne, dass der Absatz (aus TeX's Sicht) furchtbar schlimm aussieht und zu große Lücken entstehen. Meist passiert das bei schmalen Spalten oder bei Wörtern, die TeX nicht richtig trennen kann. (Siehe auch [☞ Wie funktioniert der Absatzumbruch in TeX?](#))

Folgender Code verdeutlicht das Problem:

```
\documentclass[ngerman, draft]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage[textwidth=4cm]{geometry}
\begin{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht lange Gesichter.

```
\end{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht lange Gesichter.

```
Overfull \hbox (4.44154
pt too wide) in
paragraph at lines
8--16
```

```
[ ]\T1/cmr/m/n/10 Seit
zwei Jah-ren ist
mei-|
```

```
Overfull \hbox (10.0754
pt too wide) in
paragraph at lines
8--16
```

```
\T1/cmr/m/n/10 ne Ar-
beit in D?s-sel-
dorf. Mei-|
```

```
Overfull \hbox (1.24422
pt too wide) in
```

Die schwarzen Balken zeigen, wo die übervolle Box aufgetreten ist (Option `draft` bei `\documentclass`). Es gibt hier verschiedene Lösungsansätze:

- Worttrennungen überprüfen
- Paket `microtype` benutzen
- Andere Schriftart auswählen
- Andere Parameter beim Umbruch benutzen
- Spaltenbreite vergrößern (in manchen Fällen auch verringern)
- Text umformulieren

30.1 Worttrennungen überprüfen

Manchmal besteht das Problem einfach darin, dass TeX gewisse Wörter nicht richtig trennen kann und diese dann über den rechten Rand hinausragen oder einen guten Umbruch verhindern. Leider lässt sich das mit TeXs Bordmitteln relativ schwer überprüfen. Die Meldungen oben zeigen nur die Worttrennungen falls an dieser Stelle die übervolle Box aufgetreten ist. Oft ist aber eine fehlerhafte Trennung in der Nähe verantwortlich für einen schlechten Umbruch. Mit LuaTeX kann das Paket `showhyphens` benutzt werden, um alle möglichen Trennungen anzuzeigen, im Beispiel sind Worttrennungen aber kein Problem.

30.2 Das Paket *microtype* einbinden

Das Paket `microtype` aktiviert das sogenannte *HZ-Programm*, mit dem bessere Umbruchpunkte gefunden werden können. Dazu verändert es die Buchstabenbreite in einem gewissen Maß, dass es dem Leser nicht auffällt. Durch diese Veränderung (bis ca. 2 Prozent) kann der Absatz meist so umbrochen werden, dass deutlich weniger schlechte Stellen auftreten. Hier wieder dasselbe Dokument von oben, nur mit `\usepackage{microtype}` und ohne die `draft`-Option, die das `microtype` Paket wieder ausschaltet’:

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage[textwidth=4cm]{geometry}
\usepackage{microtype}
\begin{document}
...
\end{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr alles steht und macht lange Gesichter.

```
Overfull \hbox (0.87909
pt too wide) in
paragraph at lines
8--16
```

```
[ ]\T1/cmr/m/n/10 (-20)
Seit zwei Jah-ren
ist mei-
```

```
Overfull \hbox (6.11137
pt too wide) in
paragraph at lines
8--16
```

```
\T1/cmr/m/n/10 (-20) ne
Ar-beit in D?s-sel
-dorf. Mei-
```

```
Overfull \hbox (0.67159
```

Mit sieben statt zwölf Warnungen schon deutlich besser, aber noch nicht gut genug.

30.3 *Andere Schriftart auswählen*

Meist lässt sich durch eine andere (schmäler laufende) Schriftart das Problem vollständig beheben. Hier ein beispielhafter Vorspann (Präambel) mit der Schriftart Linux Libertine:

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage[textwidth=4cm]{geometry}
\usepackage{libertine}
\usepackage{microtype}
\begin{document}
...
```

Das Ergebnis ist zwar immer noch nicht gut, aber es geht schon in die richtige Richtung:

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht lange Gesichter.

```
Overfull \hbox (3.97018
pt too wide) in
paragraph at lines
9--17
```

```
\T1/LinuxLibertineT-OsF/
m/n/10 (-20) ich,
aber ganz cool. Voll-
brems-
```

```
Overfull \hbox (0.155pt
too wide) in
paragraph at lines
9--17
```

```
\T1/LinuxLibertineT-OsF/
m/n/10 (-20) kom-bi-
na-ti-on aus Am-
pel oder
```

```
Overfull \hbox (0.85176
pt too wide) in
```

30.4 Andere Parameter beim Umbruch benutzen

Das ist die letzte Stellschraube, an der man drehen kann, ohne den Text oder die Spaltenbreite zu verändern. TeX hat eine Vielzahl an Parametern für den Absatzumbruch. Beispielsweise, wie weit Leerräume zwischen den Wörtern gedehnt werden dürfen. Die Voreinstellung für diese Parameter ist meist *sehr streng*. LaTeX bietet über den Befehl `\sloppy` bzw. der Umgebung `sloppy` etwas lockere Parameter für den Umbruch. In unserem Beispiel angewendet ergibt sich dann ein brauchbares Ergebnis:

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage[textwidth=4cm]{geometry}
\usepackage{libertine}
\usepackage{microtype}
\begin{document}
\sloppy
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht lange Gesichter.

```
\end{document}
```

erzeugt folgende Ausgabe:

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht lange Gesichter.

Genau das, was erreicht werden sollte! Nicht ganz perfekt, die fünfte Zeile stört ein wenig, aber zumindest keine Zeilen mehr, die über den Rand gehen.

31 Worin bestehen die Unterschiede zwischen `\textwidth`, `\linewidth` und `\columnwidth`?

Bei Abbildungen, Tabellen, `minipages`, `parboxen` etc. erfolgt die Breitenangabe oft als Bruchteil von `\textwidth` oder `\linewidth`, also zum Beispiel `0.8\linewidth`. Was ist der Unterschied? Und wie ist das beim zweispaltigen Satz?

- `\textwidth` ist die komplette Breite des Textbereichs auf einer Seite. In einem einspaltigen Dokument ist das gleich groß wie `\columnwidth`. In einer `minipage` wird dieser Parameter auf die aktuelle Breite angepasst.
- `\columnwidth` ist die Textbreite einer Spalte. In einem mehrspaltigen Dokument ist das dann natürlich weniger als `\textwidth`. In einer `minipage` wird dieser Parameter auf die aktuelle Breite angepasst.
- `\linewidth` ist die Länge einer Textzeile abhängig von der aktuellen Umgebung. In normalem Text ist sie gleich lang wie `\columnwidth`. In einer Liste, einer `\parbox` (und deshalb auch in p-Spalten von Tabellen), einer `minipage`, u.ä. gibt `\linewidth` dann den Wert der aktuell verfügbaren Breite an.

Das folgende Dokument verdeutlicht das für normalen Text, eine `minipage`, eine `\parbox`, eine `itemize` und eine `mdframed` Umgebung:

```
\documentclass[parskip=half-,  
%twocolumn  
{scrartcl}  
\usepackage[T1]{fontenc}  
\usepackage[vmargin=2.5cm]{geometry}
```

```

\setlength\fbboxsep{0pt}
\usepackage{xcolor}
\usepackage{mdframed}

\newcommand*\showwidth[1]{%
  \textcolor{blue}{\rule{\csname#1\endcsname}{1pt}}\
  newline
  \texttt{\textbackslash#1}: \expandafter\the\csname#1\
  endcsname
  \par
}

\begin{document}

% Rahmen für Spalte und Textbereich zeichnen
\smash{%
  \makebox[0pt][l]{\fbbox{\parbox[t][\textheight]{\
    textwidth}{\strut}}}%
  \makebox[0pt][l]{\fbbox{\parbox[t][\textheight]{\
    columnwidth}{\strut}}}%
}%

\minisec{Normaler Text}
\showwidth{textwidth}
\showwidth{columnwidth}
\showwidth{linewidth}

\medskip
\fbbox{%
  \begin{minipage}{6cm}
    \minisec{minipage (Breite 6\,cm)}
    \showwidth{textwidth}
    \showwidth{columnwidth}
    \showwidth{linewidth}
    \bigskip
  \end{minipage}%
}

\medskip
\fbbox{%
  \parbox{6cm}{%
    \minisec{parbox (Breite 6\,cm)}
  }
}

```

```
\showwidth{textwidth}
\showwidth{columnwidth}
\showwidth{linewidth}
\bigskip
}%
}

\medskip
\minisec{itemize}
\begin{itemize}
  \item \showwidth{textwidth}
        \showwidth{columnwidth}
        \showwidth{linewidth}
\end{itemize}

\medskip
\begin{mdframed}[linecolor=red,userdefinedwidth=6cm]
  \minisec{mdframed (Breite 6\,cm)}
  \showwidth{textwidth}
  \showwidth{columnwidth}
  \showwidth{linewidth}
  \bigskip
\end{mdframed}

\end{document}
```

Normaler Text

\textwidth: 418.25372pt

\columnwidth: 418.25372pt

\linewidth: 418.25372pt

minipage (Breite 6 cm)

\textwidth: 170.71652pt

\columnwidth: 170.71652pt

\linewidth: 170.71652pt

parbox (Breite 6 cm)

\textwidth: 418.25372pt

\columnwidth: 418.25372pt

\linewidth: 170.71652pt

itemize

- \textwidth: 418.25372pt

\columnwidth: 418.25372pt

\linewidth: 390.87866pt

mdframed (Breite 6 cm)

\textwidth: 149.91653pt

\columnwidth: 149.91653pt

\linewidth: 149.91653pt

Das gleiche mit `twocolumn` als Klassenoption:

Normaler Text

`\textwidth: 418.25372pt`

`\columnwidth: 204.12686pt`

`\linewidth: 204.12686pt`

minipage (Breite 6 cm)

`\textwidth: 170.71652pt`

`\columnwidth: 170.71652pt`

`\linewidth: 170.71652pt`

parbox (Breite 6 cm)

`\textwidth: 418.25372pt`

`\columnwidth: 204.12686pt`

`\linewidth: 170.71652pt`

itemize

-

`\textwidth: 418.25372pt`

`\columnwidth: 204.12686pt`

`\linewidth: 182.2268pt`

mdframed (Breite 6 cm)

`\textwidth: 149.91653pt`

`\columnwidth: 149.91653pt`

`\linewidth: 149.91653pt`

Von Clemens auf [TeXwelt.de](https://www.texwelt.de): [Worin besteht der Unterschied zwischen `\textwidth` und `\linewidth`?](#)

32 *Wie kann ich »Hurenkinder« (engl. widows), also einzelne Zeilen eines Absatzes am Kopf von Buchseiten, bzw. »Schusterjungen/Waisenkinder« (engl. clubs), einzelne Absatzzeilen auf der Seite unten, verhindern?*

TeX kennt die Parameter `\widowpenalty` und `\clubpenalty`, mit denen man die »Strafpunkte« festlegen kann, falls ein Hurenkind bzw. ein Schusterjunge beim Seitenumbruch entstehen sollte. Außerdem gibt es noch den Parameter `\displaywidowpenalty` für Hurenkinder vor einer abgesetzten mathematischen Formel. Da TeX den Wert 10000 als unendlich große Strafe interpretiert, kann man mit den Zeilen

```
\clubpenalty = 10000
\widowpenalty = 10000
\displaywidowpenalty = 10000
```

Hurenkinder und Schusterjungen fast komplett unterdrücken.

32.1 *Paket nowidow*

Sehr einfach geht das mit dem Paket `nowidow`:

```
\documentclass[paper=a4,ngerman]{scrartcl}
\usepackage[
  lines=10,
  showframe,
]{geometry}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{babel}

\usepackage[all]{nowidow}
```

```
\begin{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool.

Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt.

Es wird gar nichts mehr, alles steht und macht lange Gesichter. Ich dagegen wechsle die Cassette. Es geht weiter. Vor und hinter mir hektische

Spurenwechsel. So zieht sich das Stück A 2 bis zum Recklinghauser Kreuz. Der erhofften Entspannung folgt zuverlässig der Vollfrust.

Die A 43 bis Münster ist genauso voll wie das Kamener Kreuz zur Rush-hour.

```
\end{document}
```

Seit zwei Jahren ist meine Arbeit in Düsseldorf. Meine Familie lebt dagegen in Hamburg. Und dazwischen ich, aber ganz cool. Vollbremskombination aus Ampel oder Einfädeln oder beides auf einmal. Geht nur mit Tricks. Eben noch kurz auf die A 52. Schon vielversprechend lebhaft. Hinter dem Breitscheider Kreuz geht es richtig los. Stau auf der A 3 bis Oberhausen. Danach entspannt es sich auch nur deshalb, weil enge Baustellen mit rüden Geschwindigkeitsbegrenzungen zum gleichmäßigen Rollen zwingen. Es wird links überholt, es wird rechts überholt. Es wird gar nichts mehr, alles steht und macht lange Gesichter. Ich dagegen wechsle die Cassette. Es geht weiter. Vor und hinter mir hektische

Spurenwechsel. So zieht sich das Stück A 2 bis zum Recklinghauser Kreuz. Der erhofften Entspannung folgt zuverlässig der Vollfrust. Die A 43 bis Münster ist genauso voll wie das Kamener Kreuz zur Rush-hour.

32.2 Bemerkungen

Man sollte jedoch bedenken, dass man bei einem Satzspiegel mit fester Höhe (`\flushbottom`) und registerhaltigem Satz Hurunkinder und Schusterjungen nur schwerlich verhindern kann – insbesondere wenn man dies bei zweiseitigem Druck für jede Doppelseite beachten muss. Hier ist es oft besser, die einzelne Zeile zuzulassen oder zu versuchen, einen Absatz durch geringfügige Umformulierung eine Zeile länger oder kürzer zu machen.

Weiterhin gibt es die Möglichkeit, Absätze, die schon sehr voll sind, mit `\looseness=1` um eine Zeile zu dehnen oder Absätze, die recht leer sind, mit `\looseness=-1` um eine Zeile zu kürzen. Manchmal führt dies jedoch zu merklichen Änderungen des Grauwerts des Absatzes. (Achtung: Der Wert von `\looseness` gilt außerhalb von Listen-Umgebungen nur für den aktuellen Absatz. Innerhalb einer Listen-Umgebung muss die Zuweisung zusammen mit `\par` durch eine Gruppe lokal gehalten werden `{\looseness=1 \par}`, wenn die Änderung nur für einen Absatz gelten soll.)

33 *Wie kann ich den Abstand zwischen den Aufzählungspunkten verringern?*

Der doch recht große Abstand zwischen den einzelnen Aufzählungspunkten bei den `enumerate`- und `itemize`-Umgebungen wird oft kritisiert, gerade bei Aufzählungen mit wenig Text.

Die Pakete `enumitem`, `paralist` und `mdwlist` lassen sich für diese Änderungen gut nutzen.

```
\documentclass[paper=a4,ngerman]{scrartcl}
\usepackage{babel}
\usepackage{enumitem}
\begin{document}
```

Mit viel Abstand:

```
\begin{itemize}
  \item Hallo
  \item Welt
\end{itemize}
```

Mit weniger Abstand:

```
\setlist{noitemsep}
\begin{itemize}
  \item Hallo
  \item Welt
\end{itemize}
\end{document}
```

Mit viel Abstand:

- Hallo
- Welt

Mit weniger Abstand:

- Hallo
- Welt

Das funktioniert genau so mit `enumerate` anstelle von `enumerate`.

34 *Wie kann ich die Aufzählungspunkte farbig markieren?*

Mit dem Paket `enumitem` kann man das Aussehen der Labels einfach ändern:

```
\documentclass[paper=a4,ngerman]{scrartcl}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{enumitem,xcolor}
\begin{document}

% gilt für alle folgenden Listen:
\setlist[itemize]{font=\color{blue}}

\begin{itemize}
\item Das hier hat einen blauen Aufzählungspunkt
\end{itemize}
\end{document}
```

Möchte man die Änderung nur für eine Liste haben, kann man die Option auch an eine Liste anhängen:

```
\documentclass[paper=a4,ngerman]{scrartcl}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{enumitem,xcolor}
\begin{document}

\begin{itemize}
\item Das hier hat einen schwarzen Aufzählungspunkt
\end{itemize}

\begin{itemize}[font=\color{blue}]
\item Das hier hat einen blauen Aufzählungspunkt
\end{itemize}
\end{document}
```

- Das hier hat einen schwarzen Aufzählungspunkt
- Das hier hat einen blauen Aufzählungspunkt

35 Wie binde ich Bilder in ein LaTeX-Dokumente ein?

Um Bilder einzubinden muss man das Package `graphicx` mit `\usepackage{graphicx}` in der Präambel laden. Das Bild bindet man dann mit dem Befehl `\includegraphics[Optionen]{Dateiname}` ein. Zulässige Bildformate für `pdflatex` sind PNG, JP(E)G und PDF. Als Optionen kann man beispielsweise mit `width` die Breite des anzuzeigenden Bildes angeben und mit `height` die Höhe. Gibt man nur einen dieser Werte an, dann wird der andere entsprechend skaliert, so dass keine Verzerrung entsteht.

Beispiele:

- `\includegraphics{ordner/datei.pdf}` – das Bild wird in seiner Originalgröße gesetzt
- `\includegraphics[width=5cm]{ordner/datei.pdf}` – das Bild wird 5cm breit, die Höhe wird automatisch skaliert
- `\includegraphics[width=5cm,height=3cm]{ordner/datei.pdf}` – das Bild wird 5cm breit und 3cm hoch

Zusätzlich sollte man Bilder in eine Gleitumgebung setzen und mit einer Bildunterschrift (`\caption{...}`) versehen:

```
\begin{figure}
\includegraphics{ordner/datei.pdf}
\caption{Stadansicht von Minas Tirith}
\label{fig:minastirith}
\end{figure}
```

36 Wie kann ich Programmlistings in mein Dokument einbinden?

Das Paket `listings` bietet eine sehr umfassende Möglichkeit, Programmlistings in das Dokument einzubinden. Mit der `verbatim`-Umgebung, die LaTeX bietet, ist es aber auch möglich, wenn auch ohne die ganzen Schmankerl.

36.1 Verbatim-Umgebung

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\usepackage{libertine}
\begin{document}

\verbatim Das ist das Shell-Skript, mit dem ich meine G
  äste begrüße:

\begin{verbatim}
#!/bin/bash
x=1
while [ $x -le 5 ]
do
  echo "Willkommen zum $x., ich begrüße dich!"
  x=$(( $x + 1 ))
done
\end{verbatim}
```

```
\end{document}
```

Das ist das Shell-Skript, mit dem ich meine Gäste begrüße:

```
#!/bin/bash
x=1
while [ $x -le 5 ]
do
    echo "Willkommen zum $x., ich begrüße dich!"
    x=$(( $x + 1 ))
done
```

36.2 Paket listings

Das Paket listings bietet sehr viele Möglichkeiten der Formatierung an. Im Normalfall wirkt es etwas ungewöhnlich:

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\usepackage{libertine}
\usepackage{listings}
\begin{document}

\noindent Das ist das Shell-Skript, mit dem ich meine G
    äste begrüße:

\begin{lstlisting}[language=sh]
#!/bin/bash
x=1
while [ $x -le 5 ]
do
    echo "Willkommen zum $x."
\end{lstlisting}
```

```
x=$(( $x + 1 ))
done
\end{lstlisting}
\end{document}
```

Das ist das Shell-Skript, mit dem ich meine Gäste begrüße:

```
#!/bin/bash
x=1
while [ $x -le 5 ]
do
    echo "Willkommen_zum_$x."
    x=$(( $x + 1 ))
done
```

Ein Problem, über das viele Anwender stolpern, kann man leicht umgehen: Normalerweise kommt `listings` nicht mit UTF-8 kodierten Umlauten zurecht. Folgender Quelltext zeigt den Trick für die Umlaute (`literate=`) und ändert die Formatierung ein wenig. Für die Farben wird das Paket `xcolor` benutzt.

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\usepackage{libertine}
\usepackage{listings}
\usepackage{xcolor}
\begin{document}

\noindent Das ist das Shell-Skript, mit dem ich meine G
äste begrüße:

\lstset
{
```

```

morekeywords={draw,clip,path},
frame=single,
framesep=\fboxsep,
framerule=\fboxrule,
rulecolor=\color{red},
xleftmargin=\dimexpr\fboxsep+\fboxrule\relax,
xrightmargin=\dimexpr\fboxsep+\fboxrule\relax,
breaklines=true,
basicstyle=\small\tt,
keywordstyle=\color{blue},
commentstyle=\color[rgb]{0.13,0.54,0.13},
backgroundcolor=\color{yellow!10},
tabsize=2,
columns=flexible,
morekeywords={maketitle},
literate=%
{Ö}{{\ "O}}1
{Ä}{{\ "A}}1
{Û}{{\ "U}}1
{ß}{{\ss}}1
{ü}{{\ "u}}1
{ä}{{\ "a}}1
{ö}{{\ "o}}1
{~}{{\textasciitilde}}1
}

```

```

\begin{lstlisting}[language=sh]
#!/bin/bash
x=1
while [ $x -le 5 ]
do
  echo "Willkommen zum $x., ich begrüße dich!"
  x=$(( $x + 1 ))
done
\end{lstlisting}
\end{document}

```

Das ist das Shell-Skript, mit dem ich meine Gäste begrüße:

```
#!/bin/bash
x=1
while [ $x -le 5 ]
do
  echo "Willkommen_zum_$x.,_ich_begrüße_dich!"
  x=$(( $x + 1 ))
done
```

37 Wie kann ich die Positionierung der Gleitumgebungen (Tabellen und Abbildungen) beeinflussen?

Der Gleitumgebung kann ein optionaler Parameter angehängen werden, der die Optionen h, t, b, p und ! enthalten kann.

```
\begin{figure}[tb]
...
\end{figure}
```

Dabei bedeuten:

- t dass die Gleitumgebung am oberen Rand einer Seite positioniert werden darf
- b dass die Gleitumgebung am unteren Rand einer Seite positioniert werden darf
- h dass die Gleitumgebung an der Stelle, an der sie im Quelltext steht, positioniert werden darf
- p dass die Gleitumgebung auf einer eigenen Seite positioniert werden darf

Bei diesen Optionen versucht LaTeX trotzdem bestimmte Regeln

zu beachten. (z. B. mind. 30% Text auf einer Seite usw.) Gelingt es nicht, die Gleitumgebung an den in den Optionen vorgeschlagenen Positionen unterzubringen, ignoriert LaTeX die Vorgaben und versucht auch die nicht angegeben. Mit ! kann man LaTeX dazu zwingen, die angesprochenen zusätzlichen Regeln zu ignorieren. Dies kann aber unerwünschte Nebeneffekte haben, dass z. B. alle Gleitumgebungen (auch die folgenden) an das Ende des Kapitels verschoben werden, da es LaTeX nicht anders setzen kann.

Die Reihenfolge, in der die Parameter angegeben werden, ist dabei egal. Es macht also keinen Unterschied, ob man [htb] oder [bth] angibt.

(FIXME) Angaben wie H aus dem Package float erwähnen?

38 Wie funktionieren Gleitobjekte und wie kann man ihre Positionierung beeinflussen?

Achtung lang!

38.1 Terminologie

Klassen von Gleitumgebungen

Jede von LaTeX's Gleitumgebungen gehört einer **Klasse** an. Der LaTeX-Kernel kennt von sich aus erst einmal zwei: `figure` und `table`. Dokumentenklassen oder Pakete können weitere definieren. Die Klasse einer Gleitumgebung beeinflusst Eigenschaften wie zum Beispiel die voreingestellte Platzierungsoption.

LaTeX wird nie die Reihenfolge von Gleitumgebungen einer Klasse ändern. Hat man etwa `figure1`, `table1`, `figure2` in einem Dokument, dann wird `figure2` auf jeden Fall *nach* `figure1` platziert werden. `table1` gehört einer anderen Klasse an und wird daher unabhängig davon platziert.

Gleitumgebungsbereiche

In einer Kolumne kennt LaTeX zwei **Bereiche**, in denen es Gleitumgebungen platzieren kann: den *top*-Bereich und den *bottom*-Bereich. In einem zweispaltigen Layout gibt es außerdem einen *top*-Bereich, der beide Spalten umfasst. Einen entsprechenden *bottom*-Bereich gibt es nicht.

Außerdem kann LaTeX Gleitkolumnen oder -seiten machen, also Kolumnen oder Seiten, die nur Gleitumgebungen enthalten.

Zu guter letzt kann LaTeX Gleitumgebungen „*inline*“ in den Text platzieren.

Platzierungsparameter

Um eine Gleitumgebung in einen der genannten Bereiche zu platzieren, benötigt eine Gleitumgebung Platzierungsparameter als optionales Argument. Ohne Argument werden die Voreinstellungen verwendet. Das optionale Argument kann die folgenden Zeichen in beliebiger Reihenfolge enthalten:

- ! manche Einschränkungen sollen ignoriert werden (siehe später)

- h die Gleitumgebung darf *inline* (here) platziert werden
- t die Gleitumgebung darf in den *top*-Bereich platziert werden
- b die Gleitumgebung darf in den *bottom*-Bereich platziert werden
- p die Gleitumgebung darf auf einer Gleitkolumne oder Gleitseite platziert werden

Die Reihenfolge, in der diese Zeichen verwendet werden, beeinflussen nicht, wie der Algorithmus bestimmt, wo die Gleitumgebung platziert wird. Beispielsweise machen [ht] und [th] keinerlei Unterschied! Das ist eines der häufigsten Missverständnisse über Gleitumgebungen.

Wenn jedoch ein Zeichen nicht verwendet wird, dann wird die entsprechende Option überhaupt nicht beachtet!

Parameter des Algorithmus

Es gibt ca. zwanzig Parameter, die die Platzierung einer Gleitumgebung beeinflussen. Sie bestimmen grundsätzlich

- wie viele Gleitumgebungen in einen bestimmten Bereich platziert werden dürfen.
- wie groß ein Bereich werden darf.
- wie viel Text auf einer Seite sein muss oder anders gesagt wieviel Platz *top*- und *bottom*-Bereich einnehmen dürfen.
- wie viel Weißraum eingefügt wird

- zwischen aufeinander folgenden Gleitumgebungen
- zwischen Gleitbereich und dem Text darüber oder darunter

Referenzpunkt einer Gleitumgebung

Die Stelle, an die eine Gleitumgebung im Quelltext geschrieben wird, beeinflusst die Platzierung der Gleitumgebung im Output, da dadurch festgelegt ist, wann LaTeX die Gleitumgebung zum ersten Mal sieht. Wenn eine Gleitumgebung mitten in einem Absatz platziert wurde, dann ist der Referenzpunkt der nächste Zeilenumbruch oder Seitenumbruch in dem Absatz, in dem die Gleitumgebung geschrieben wurde.

38.2 Grundlegende Regeln des Gleitmechanismus

Mit dem jetzigen Wissen können wir in das Verhalten des Algorithmus etwas eintauchen.

Zunächst muss man verstehen, dass alle von LaTeXs Algorithmen dergestalt konzipiert wurden, dass Rückverfolgung vermieden wird. Das bedeutet, LaTeX liest den Quelltext, formatiert, was es findet, und setzt es mehr oder weniger sofort. Diese Designentscheidung wurde getroffen, um einerseits die (dennoch recht hohe) Komplexität niedrig zu halten und andererseits eine vernünftige Geschwindigkeit zu erhalten. (Man denke daran, dass das aus den frühen Achtzigern stammt).

Der Algorithmus für Gleitumgebungen ist *gierig*, d.h., wenn er eine

Gleitumgebung findet, wird er versuchen, sie sofort zu platzieren. Wenn er erfolgreich war, wird er diese Entscheidung niemals rückgängig machen. Das bedeutet, er könnte eine Lösung wählen, die eigentlich unterlegen ist, wenn man später erhaltene Daten berücksichtigt.

Wenn eine `figure` beispielsweise in den *top*-Bereich platziert werden darf, entscheidet LaTeX vielleicht, sie dort zu platzieren. Wenn diese Abbildung von zwei `tables` gefolgt wird, die *nur* in den *top*-Bereich platziert werden dürfen, passen sie vielleicht nicht mehr hinein. Eine Lösung, die möglicherweise funktioniert hätte aber nicht ausprobiert wurde, wäre gewesen, die Abbildung in den *bottom*-Bereich zu platzieren, und beide Tabellen in den *top*-Bereich.

Die grundlegende Reihenfolge

Der Algorithmus durchläuft folgende Schritte:

- Wenn eine Gleitumgebung gefunden wird, versucht LaTeX, sie nach seinen Regeln (die noch besprochen werden) sofort zu platzieren.
 - Wenn das klappt, wird die Gleitumgebung platziert und die Entscheidung nie mehr rückgängig gemacht.
 - Wenn es nicht klappt, wird die Gleitumgebung in eine Warteschlange gestellt, damit sie beim nächsten Seitenumbruch (nicht früher) wieder berücksichtigt werden kann.
- Wenn eine Seite vollständig ist, begutachtet LaTeX seine

Warteschlange und versucht, sie so gut wie möglich zu leeren. Dafür wird es zunächst so viele Gleitseiten wie möglich erzeugen, in der Hoffnung, dass sich die Warteschlange dadurch leert. Wenn diese Möglichkeit erschöpft ist, wird als nächstes versucht, die übrigen Gleitumgebungen auf die *top*- und *bottom*-Bereiche zu verteilen. Alle verbleibenden Gleitumgebungen werden begutachtet und entweder platziert oder für die nächste Seite wieder in die Warteschlange gestellt.

- Danach wird das Material für die aktuelle Seite verarbeitet. Dabei können weitere Gleitumgebungen entdeckt werden.
- Wenn das Ende des Dokuments oder ein `\clearpage` erreicht wird, startet LaTeX eine neue Seite, lockert LaTeX alle einschränkenden Regeln und gibt alle Gleitumgebungen der Warteschlange aus, indem es sie auf Gleitseiten setzt.

Regeln, nach denen eine gefundene Gleitumgebung platziert wird

Immer, wenn LaTeX eine Gleitumgebung im Quelltext findet, schaut es zunächst in die Warteschlange, um zu sehen, ob schon eine weitere Umgebung derselben Klasse in der Schlange steht. Wenn das der Fall ist, wird die Gleitumgebung der Warteschlange sofort hinzugefügt.

Wenn nicht, dann schaut LaTeX nach den Platzierungsparametern für diese Gleitumgebung, entweder die des optionalen Arguments oder die der Voreinstellung der entsprechenden Klasse.

- Wenn die Parameter ein `!` enthalten, wird der Algorithmus

alle Einschränkungen bezüglich der Anzahl der Umgebungen, die in einen Bereich platziert werden dürfen, oder der maximalen Größe, die ein Bereich einnehmen darf. Andere Einschränkungen werden angewandt.

- Als nächstes wird geschaut, ob h angegeben wurde.
 - Wenn ja, dann wird versucht, die Umgebung genau da auszugeben, wo sie im Quelltext eingegeben wurde. Wenn das klappt, wenn also genügend Platz da ist, wird die Umgebung platziert, und der Prozess ist beendet.
 - Wenn nicht, dann wird als nächstes nach einem t geschaut. Ist es da, wird versucht, die Umgebung in den *top*-Bereich zu platzieren. Wenn es keine Einschränkungen gibt, die das verhindern, wird die Gleitumgebung ausgegeben und der Algorithmus stoppt.
 - Ist es auch nicht da, wird schließlich nach einem b geschaut und, wenn es gefunden wird, versucht, die Umgebung in den *bottom*-Bereich zu platzieren, natürlich unter Beachtung eventueller Einschränkungen.
- Wenn das nicht funktioniert hat oder wegen fehlender Parameter nicht erlaubt war, wird die Gleitumgebung der Warteschlange hinzugefügt.
- Ein eventuell vorhandenes p wird in diesem Prozess nicht berücksichtigt. Es wird beim nächsten Seitenumbruch beachtet.

Damit ist der Prozess beendet, der durchlaufen wird, wenn eine

Gleitumgebung im Dokument gefunden wird.

Leeren der Warteschlange bei einem Seitenubruch

Nachdem eine Seite fertiggestellt ist, begutachtet LaTeX die Warteschlange, und versucht, sie so gut wie möglich zu leeren. Dafür wird es zuerst versuchen, Gleitseiten zu erzeugen.

Alle Gleitumgebungen, die an einer Gleitseite (oder -kolumne) teilnehmen, müssen ein `p` als Parameter angegeben haben. Fehlt es, darf die Umgebung nicht auf eine Gleitseite gesetzt werden. *Obendrein verhindert es, dass irgendeine andere Umgebung der gleichen Klasse auf die Gleitseite gesetzt wird!*

Darf eine Gleitumgebung auf eine Gleitseite platziert werden, wird sie für die Seite vorgemerkt. Der Algorithmus kann aber immer noch abrechnen, wenn die Gleitseite nicht „voll genug“ würde. (Das hängt von den Parametereinstellungen für Gleitseiten ab.) Erst ganz am Ende des Dokuments oder wenn ein `\clearpage` gefunden wird, werden diese Beschränkungen aufgehoben, und eine Gleitumgebungen wird auf eine Gleitseite gesetzt, auch wenn sie kein `p` hat und die einzige Gleitumgebung der Seite ist.

Das Erstellen von Gleitseiten geht so lange weiter, bis der Lagorithmus entweder keine weiteren Gleitumgebungen zum platzieren übrig hat, oder er scheitert, eine Gleitseite zu erstellen. Im letzten Fall werden alle Gleitumgebungen, die bis dahin noch nicht platziert wurden, als Kandidaten für die *top*- und *bottom*-Bereiche der nächsten Seite (oder Kolumne) behandelt.

Das Vorgehen dann ist dasselbe, wie oben beschrieben mit der

Ausnahme, dass

- das Zeichen `h` keine Bedeutung mehr hat (an dieser Stelle sind wir weit vom ursprünglichen „here“ entfernt) und daher ignoriert wird,
- die Gleitumgebungen an dieser Stelle nicht mehr vom Quelltext stammen, sondern eine nach der anderen von der Warteschlange.

Jede Gleitumgebung, die bis hierhin noch nicht platziert werden konnte, wird wieder auf die Warteschlange gesetzt. Wenn LaTeX also wieder soweit ist, weiteres Textmaterial aus dem Quelltext aufzunehmen, kann die Warteschlange also bereits Gleitumgebungen enthalten. Eine Konsequenz daraus ist, dass dann eine Gleitumgebung, die im Dokument gefunden wird, zurückgehalten wird, nur weil eine frühere Gleitumgebung schon auf der Wartebank sitzt.

Details zu den Parameter, die die Platzierung beeinflussen

Es gibt vier Zähler, die kontrollieren, wie viele Gleitumgebungen in de verschiedenen Bereiche gesetzt werden dürfen:

- `totalnumber` (Voreinstellung 3), die maximale Anzahl von Gleitumgebungen auf einer Textseite (*nicht* Gleitseite!)
- `topnumber` (Voreinstellung 2), die maximale Anzahl von Gleitumgebungen, die in einen *top*-Bereich platziert werden dürfen

- `bottomnumber` (Voreinstellung 1), die maximale Anzahl von Gleitumbegungen, die in einen *bottom*-Bereich platziert werden dürfen
- `dbltopnumber` (Voreinstellung 2), die maximale Anzahl von Gleitumbegungen, die in einen ganzseitigen *top*-Bereich in einem zweispaltigen Dokument platziert werden dürfen

Die Größe der einzelnen Bereiche, wird durch Parameter kontrolliert, die den maximalen oder minimalen Anteil eines Bereichs als Bruchteil der Seitenhöhe angeben. Sie können mit `\renewcommand` geändert werden.

- `\topfraction` (Voreinstellung 0.7), maximale Größe des *top*-Bereichs.
- `\bottomfraction` (Voreinstellung 0.3), maximale Größe des *bottom*-Bereichs.
- `\dbltopfraction` (Voreinstellung 0.7), maximale Größe des ganzseitigen *top*-Bereichs in einem zweispaltigen Dokument.
- `\textfraction` (Voreinstellung 0.2), *minimale* Größe des Textbereichs, also des Teils, der nicht von Gleitumbegungen belegt werden darf.

Der Weißraum, der die Gleitumbegungen innerhalb eines Bereichs voneinander trennt, wird durch die folgenden Parameter bestimmt. Sie alle sind Gummilängen, dürfen also stauch- oder dehnbare Anteile enthalten. Ihre Voreinstellung hängt von der Schriftgröße des Dokuments ab und ändert sich, wenn Klassenoptionen wie `11pt` oder `12pt` verwendet werden. Hier sind die Voreinstellungen für

10pt:

- `\floatsep` (Voreinstellung 12pt plus 2pt minus 2pt), der Abstand zwischen Gleitumgebungen in den *top*- oder *bottom*-Bereichen.
- `\dblfloatsep` (Voreinstellung 12pt plus 2pt minus 2pt) der Abstand zwischen Gleitumgebungen im ganzseitigen *top*-Bereich eines zweispaltigen Dokuments.
- `\textfloatsep` (Voreinstellung 20pt plus 2pt minus 4pt) der Abstand zwischen *top*- oder *bottom*-Bereich und Textbereich
- `\dbltextfloatsep` (Voreinstellung 20pt plus 2pt minus 4pt) der Abstand zwischen ganzseitigem *top*-Bereich eines zweispaltigen Dokuments und Textbereich.

Gleitumgebungen, die „here“ platziert werden, wird der Abstand zum umgebenden Text durch folgende Länge vorgegeben:

- `\intextsep` (Voreinstellung 12pt plus 2pt minus 2pt)

Im Fall von Gleitseiten oder Gleitkolumnen (also einer Seite oder Spalte, die nur Gleitumgebungen enthält), gelten die Parameter nicht. Sie werden durch diesen Parameter kontrolliert:

- `\floatpagefraction` (Voreinstellung 0.5), minimaler Anteil einer Seite oder Kolumne, der durch Gleitumgebungen besetzt werden muss, bevor eine Gleitseite oder Gleitkolumne gebildet werden darf.

38.3 Konsequenzen des Algorithmus

Eine Gleitumgebung kann früher im Dokument erscheinen als im Quelltext

Die Platzierung einer Gleitumgebung im Quelltext bestimmt den frühesten Punkt, an dem eine Gleitumgebung erscheinen darf. Sie kann im Dokument zu einem bestimmten Grad rückwärts wandern, da sie in den *top*-Bereich der aktuellen Seite platziert werden könnte. Sie kann aber nicht auf einer früheren Seite als der umgebende Text erscheinen, da LaTeX keine Rückverfolgung macht und die früheren Seiten bereits gesetzt sind.

Daher wird eine Gleitumgebung normalerweise in der Nähe seiner ersten Erwähnung (Text wie „siehe Tabelle~3“) platziert, weil man dadurch sicherstellt, dass die Gleitumgebung entweder auf der gleichen Seite wie der Text oder eine Seite später erscheint. In manchen Fällen möchte man die Gleitumgebung vielleicht auf der vorhergehenden Seite platzieren (wenn die Seite von der Erwähnung aus immer noch sichtbar ist). Das kann man nur erreichen, indem man den Quelltext an eine frühere Stelle schiebt.

Ganzseitige Gleitumgebungen in zweispaltigen Dokumenten werden immer erst zurückgehalten

Wenn LaTeX eine ganzseitige Gleitumgebung (also `figure*` oder `table*`) im zweispaltigen Satz findet, wird sie sofort auf die Warteschlange geschoben. Das liegt wieder am „gierigen“ Verhalten des Algorithmus: wenn LaTeX gerade die zweite Spalte einer Seite sammelt, ist die erste Spalte bereits eingesammelt und

für den Satz gespeichert. Da LaTeX nicht zurückschaut, gibt es keine Möglichkeit, die Umgebung auf die aktuelle Seite zu setzen. Um den Algorithmus einfach zu halten, geschieht das gleiche sogar, wenn die erste Spalte bearbeitet wird.

Um also eine Gleitumgebung auf die aktuelle Seite zu setzen, muss man sie manuell an eine andere Stelle in der Quelle verschieben – vor den Anfang der aktuellen Seite. Offensichtlich kann jede weitere Veränderung am Dokument diese Anpassung obsolet machen. Wenn man solche Anpassungen also überhaupt vornimmt, dann am besten in der allerletzten Phase der Dokumenterstellung – wenn alles Material geschrieben ist und man sich auf Feineinstellungen des optischen Erscheinungsbilds konzentrieren kann.

Es gibt keinen bottom-Bereich für ganzseitige Gleitumgebungen

Eigentlich ist das keine Konsequenz aus dem Algorithmus sondern eine Tatsache über ihn. Die einzigen Bereiche für zweispaltige Gleitumgebungen sind der *top*-Bereich und eine Gleitseite. Wenn man also ein `h` oder `b` als Parameter hinzufügt, werden sie einfach ignoriert. `{figure*}[b]` impliziert, dass die Gleitumgebung bis zum Ende des Dokuments oder dem nächsten `\clearpage` zurückgehalten wird.

Alle Parameter schränken (normalerweise) die Platzierungsmöglichkeiten ein

Das mag offensichtlich erscheinen, ist es aber Wert, wiederholt zu werden: jeder Parameter beschränkt LaTeXs Möglichkeiten,

Gleitumgebungen zu platzieren. es gibt immer eine Möglichkeit, einen Parameter so einzustellen, dass er die Platzierung nicht mehr beeinflusst. Leider lädt man damit zu eher schlecht aussehenden Platzierungen ein.

LaTeXs Voreinstellungen sind ziemlich großzügig. Damit beispielsweise eine Gleitseite akzeptiert wird, muss/müssen die Gleitumgebung(en) mindestens eine halbe Seite einnehmen. Anders gesagt ist es erlaubt, dass eine solche Seite zur Hälfte leer ist, was in den meisten Fällen sicherlich nicht die bestmögliche Lösung ist.

Oft geschieht es, dass Anwender versuchen, solche Einstellungen zu verbessern, und dann überrascht sind, wenn sich plötzlich alle Gleitumgebungen am Ende des Dokuments aufstapeln. Um bei dem Beispiel zu bleiben: wenn man `\floatpagefraction` auf, sagen wir, `0.8` einstellt, heißt das, dass eine Gleitumgebung, die `0.75` einer Seite einnehmen würde, keine Gleitseite bilden darf. Wenn es also keine zweite Gleitumgebung gibt, die hinzugefügt werden kann, wird die Umgebung zurückgehalten, zusammen mit allen weiteren Umgebungen der gleichen Klasse. Es ist noch schlimmer: diese spezielle Umgebung ist zu groß, um auf die nächste Seite im *top*-Bereich platziert zu werden, da die maximal erlaubte Größe in der Voreinstellung `0.7` beträgt. Als Konsequenz wird die Umgebung bis zum nächsten `\clearpage` zurückgehalten.

Aus diesem Grund ist es am besten, nicht an den Parametern zu fummeln, während man ein Dokument schreibt. Oder wenigstens nicht so, dass man es dem Algorithmus schwerer macht, die Gleitumgebung in der Nähe der Erwähnung zu platzieren. Zum Korrekturlesen ist es wesentlich wichtiger, dass die Gleitumgebung in der Nähe ihrer Erwähnung ist, als halbleere Seiten zu

vermeiden.

Einen weiteren Schluss, den man hier ziehen kann, ist, dass es Abhängigkeiten zwischen einigen der Parameter gibt. Es ist wichtig, dass man sie berücksichtigt, wenn man die Werte ändert.

„Hier“ bedeutet eigentlich „hier, wenn es passt“

... und oft passt es nicht. Für viele Leute ist es etwas überraschend, aber durch die Art und Weise, wie der Algorithmus konzipiert wurde, ist `h` kein unbedingter Befehl. Wenn ein unbedingter Befehl benötigt wird, bieten Erweiterungen zum Beispiel den `H` Parameter, der wirklich bedeutet „hier und starte eine neue Seite, wenn nötig“. (In einem solchen Fall kann man sich auch überlegen, ob man wirklich eine *Gleitumgebung* braucht.)

Gleit-Optionen legen keine bevorzugte Reihenfolge fest

Wie bereits erwähnt, versucht der Algorithmus, die Gleitumgebungen in einer vorgegebenen Reihenfolge in den verfügbaren Bereichen zu platzieren. Die Reihenfolge ist festprogrammiert: „here“, „top“, „bottom“ und – bei Seitenumbrüchen – zuerst „page“ und, nur falls das nicht möglich ist, „top“ gefolgt von „bottom“ auf der folgenden Seite.

Wenn man also `[bt]` angibt, heißt das nicht, dass zuerst „bottom“ und erst danach „top“ ausprobiert wird. Es bedeutet lediglich, dass diese Gleitumgebung nur in den *top*- und *bottom*-Bereich gehen darf, genau wie bei `[tb]`.

Zusammenhang zwischen Gleitumgebungen und Fußnoten

Dieser Punkt ist nicht wirklich eine Konsequenz des Algorithmus' sondern eher eine seiner Implementierung: jedes mal, wenn LaTeX sich zu entscheiden versucht, wo eine Gleitumgebung (oder ein `\marginpar`) platziert werden soll, löst es die *Output routine* (OR) aus. Als Teil dieses Prozesses werden alle Fußnoten von ihrem aktuellen Platz auf der „Druckfahne“ entfernt und zusammen in die `\footins`-Box gesammelt. Nachdem die Gleitumgebung platziert oder auf die Warteschlange gesetzt wurde, kehrt LaTeX zur Druckfahne zurück. Wegen der OR wurde diese aber *geändert*: LaTeX muss die Fußnoten nun irgendwo aber *alle gemeinsam* platzieren. Also fügt es die Fußnoten (genauer: den Fußnotentext) am Ende der Fahne ein.

Wenn jetzt aber die Seite schlußendlich an einer anderen Stelle gebildet wird, kann es passieren, dass die Fußnoten auf der falschen Seite oder in der falschen Spalte erscheinen. Das ist zwar ein ziemlich unwahrscheinliches Szenario, aber falls es passiert, sollte man überprüfen, ob in der Nähe des Seitenumbruchs eine Gleitumgebung ist. Dann kann man entweder die Gleitumgebung an eine andere Stelle schieben oder einen expliziten Seitenumbruch einfügen.

Ein spezieller Fall ist es Wert, hervorgehoben zu werden: platziere keine Gleitumgebung direkt nach einer Überschrift, außer es ist eine Überschrift, die immer am Anfang einer Seite steht. Überschriften sind normalerweise recht große Objekte, verhindern aber einen Seitenumbruch nach sich. Eine Gleitumgebung hier mittenrein zu setzen, bedeutet, die OR auszulösen, bevor sich LaTeX entscheiden kann, wo es umbrechen soll, wodurch Fußnoten

an die falsche Stelle geschoben werden.

39 Wie kann ich Tabellen und Abbildungen in der Nähe des Aufrufs behalten?

- Durch Vermeidung einer Gleitumgebung
- Durch geeignete Positionierung der Gleitumgebung mit der Option `!ht`: z.B. `\begin{figure}[!ht] ... %` kann auch eine Tabelle sein, die in der `figure`-Umgebung aufgerufen wird. `\end{figure}`

40 Wie kann ich ein Gleitobjekt das alleine auf einer Seite steht oben an der Seite ausrichten und nicht vertikal zentriert?

Hat man ein Gleitobjekt das alleine auf einer Seite steht, dann wird dieses automatisch auf die Mitte der Seite gesetzt. Dies ist häufig unerwünscht.

Es gibt drei Längen die die Platzierung beeinflussen:

- `\@fptop` ist der Abstand vom oberen Seitenrand zum ersten Float-Objekt
- `\@fpsep` ist der Abstand zwischen zwei Float-Objekten
- `\@fpbot` ist der Abstand des unteren Randes der untersten Abbildung zum unteren Seitenrand

Die Standards von LaTeX sind wie folgt definiert:

```
\@fptop = 0pt + 1fil  
\@fpsep = 8pt + 2fil  
\@fpbot = 0pt + 1fil
```

Diese Definitionen sorgen dafür, dass der Freiraum zwischen einem oder mehreren Bildern gleichmäßig verteilt wird.

Um ein Gleitobjekt nun oben auf der Seite zu platzieren kann man den oberen Abstand redefinieren:

```
\makeatletter  
\setlength{\@fptop}{0pt}  
\makeatother
```

Es kann nun sein, dass die Gleitumgebung im Vergleich zu einer Textseite als zu hoch erscheint. Dazu kann man nun `\@fptop` auf den Wert `{4pt}` setzen was ungefähr dem Wert `\topskip` - der Höhe des normalen Texts entspricht. `\topskip` ist der Abstand vom oberen Seitenrand zur Grundlinie der ersten Textzeile.

Es ist zu beachten, dass dies eine globale Einstellung ist die sich auf das ganze Dokument auswirkt. Deshalb sollte diese Änderung in der Klassendatei (`*.cls`) oder in der Präambel des Dokumentes vorgenommen werden.

41 Wieso werden die meisten meiner Abbildungen an das Ende des Kapitels oder Dokuments verschoben?

Das optionale Argument der »figure«- oder »table«-Umgebung schränkt die Platzierungsmöglichkeiten auf die angegebenen Orte

ein. Mit dem optionalen Argument gibt man also nicht, wie oft fälschlich geglaubt und weiterverbreitet wird, die Platzierung für weitere Orte an. Deshalb nie Platzierungsangaben nur auf [h] einschränken, sondern mindestens einen weiteren Ort mit angeben! Grund: Kann die Abbildung nicht hier plaziert werden, darf sie nirgendwo anders gesetzt werden und wird nach hinten an das Ende des Kapitels bzw. des Dokuments geschoben. Da LaTeX die Reihenfolge der Abbildungen bzw. der Tafeln nicht verändert, werden alle nachfolgenden Abbildungen bzw. Tafeln mit dieser nicht plazierbaren Abbildung oder Tafel nach hinten verschoben. Genau dasselbe kann passieren, wenn die Platzierungsbeschränkung mit [p] angegeben wird, das Bild kleiner als `\floatpagefraction` ist und das nächste Bild nicht zusammen mit diesem auf eine Seite passt. Ebenso bei einer Beschränkung auf [t] und einem Bild, das größer als `\topfraction` ist, dto. für [b] und `\bottomfraction`.

Tipp: Einfach am Anfang das optionale Argument der »figure«- oder »table«-Umgebung weg lassen – es ist nicht ohne Grund ein optionales Argument dieser Umgebungen! Und erst später in der Endfassung des Dokuments nur an den Stellen ein optionales Argument mit Platzierungseinschränkungen einfügen, wenn z. B. eine bestimmte Abbildung oder Tafel auf einer eigenen Seite oder auf einer Seite oben erscheinen sollte.

Weitere Hinweise gibt es in

- Axel Reichert: »☞Gleitobjekte – die richtige Schmierung«
- Axel Reichert: »☞Satz von Tabellen«
 - Piet van Oostrum: »☞Floats in LaTeX«

42 Wie kann ich in einer Abbildung zwei Bilder nebeneinander setzen?

Man kann zum Beispiel zwei »minipage«-Umgebungen verwenden, wobei man dann auch mit deren optionalem Argument die Ausrichtung der beiden Bilder angeben kann:

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage{graphicx}
\begin{document}

\begin{figure}
  \begin{minipage}[b]{.4\linewidth} % [b] =>
    Ausrichtung an \caption
    \includegraphics[width=\linewidth]{ozean}
    \caption{Wasser}
  \end{minipage}
  \hspace{.1\linewidth}% Abstand zwischen Bilder
  \begin{minipage}[b]{.4\linewidth} % [b] =>
    Ausrichtung an \caption
    \includegraphics[width=\linewidth]{cow}
    \caption{Land}
  \end{minipage}
\end{figure}
\end{document}
```

Ein sehr viel bequemerer Weg, um mehrere zusammengehörende Abbildungen oder Tafeln in einer Float-Umgebung zu setzen, bietet das Paket `subfig` (das ist der Nachfolger des Pakets `subfigure`). Hierzu stellt das Paket unter anderem die Anweisung `\subfloat` zur Verfügung. Die Formatierung der Einzelabbildungen läßt sich sehr leicht anpassen.

```
\documentclass[ngerman]{article}
\usepackage{babel}
```

```

\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage{graphicx}
\usepackage{subfig}
\begin{document}

\begin{figure}%
  \centering
  \subfloat[][]{\includegraphics[width=0.4\linewidth]{
    ozean}}}%
  \qqquad
  \subfloat[][]{\includegraphics[width=0.4\linewidth]{
    cow}}}%
  \caption{Zu Wasser und zu Land}%
\end{figure}
\end{document}

```

43 Absolute Positionierung von Bildern

Manchmal muss man Bilder an einer bestimmten (festen) Position auf der Seite platzieren. Das passt nicht direkt zur Philosophie von LaTeX, alles nacheinander von oben nach unten zu platzieren. Für eine absolute Positionierung der Elemente bietet sich das Paket `eso-pic` an. Damit ist eine feste Positionierung möglich.

```

\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[utf8]{inputenc}
\usepackage{eso-pic,picture,graphicx}
\begin{document}
\AddToShipoutPictureFG*{%
  \AtPageUpperLeft{%
    \put(22mm,-20cm){%
      \makebox(0,0)[lt]{\includegraphics[width=3cm]{
        ocean}}}%
    }%
  }%
}

```

```
}%  
Hello world  
\clearpage  
Hello world  
\end{document}
```

Hier wird das Bild nur auf der ersten Seite (das ist der *bei* `\AddToShiptoutPictureFG`) fest platziert. Soll es auf allen Seiten fest erscheinen, so muss das Sternchen weggelassen werden.

44 *Wie kann ich mir mit Makros das Schreiben erleichtern?*

LaTeX selbst ist eine Makrosprache, das bedeutet sie bietet von Haus aus die Möglichkeiten für Automatismen. Diese Makros werden im Deutschen oft als Befehle bezeichnet, im Englischen als *commands*. Die ursprüngliche Bezeichnung ist übrigens *control sequence*.

LaTeX als solches stellt den Befehl `\newcommand` zur Verfügung. Betrachten wir als Beispiel die Definition eines Befehls `\fragen`:

```
\newcommand{\fragen}{Wieso? Weshalb? Warum?}  
%Neuer Befehl, der heißt fragen und ersetzt sich selbst  
%mit dem, was in Klammern steht
```

Im Text können wir nun den Befehl `\fragen` benutzen. Es ist übrigens hilfreich die Beispiele mit dem eigenen System zu reproduzieren.

```
\documentclass{article}  
\newcommand{\fragen}{Wieso? Weshalb? Warum?}
```

```
\begin{document}
Es blieben einige Fragen: \fragen
\end{document}
```

LaTeX ersetzt `\fragen` mit all dem, was wir in Klammern geschrieben haben.

Solche einfachen Befehle kennen wir aus unseren Dokumenten (`\TeX`, `\tableofcontents` oder `\listoffigures`).

```
Er selbst bezeichnete sich als \emph{\TeX niker aus
Leidenschaft}, den er liebte \TeX sehr.
```

Was passiert an dieser Stelle? *TeXniker* wird zusammen geschrieben, leider aber auch *TeXsehr*. Warum aber verhält sich das Computerprogramm so? Wir haben uns das Wort *TeXniker* aus `\TeX` und `niker` gebildet. Würden wir beide Teile direkt hier zusammenfügen, würde LaTeX bemerken, dass der Befehl `\TeXniker` niemals definiert wurde (`undefined control sequence`). Das Leerzeichen dient uns also dazu beide Teile voneinander zu trennen, geschweifte Klammern sind dazu aber auch möglich.

```
Er selbst bezeichnete sich als \emph{\TeX niker aus
Leidenschaft}, denn er liebte \TeX{} sehr.
```

Jeder kennt aber auch Befehle, welche ein Argument benötigen, also irgendetwas, was LaTeX verarbeiten muss. Denken wir beispielsweise an Überschriften in unseren Dokumenten.

```
\section{Ich bin ein neuer Abschnitt}
```

Alles, was hier in Klammern steht, wird von LaTeX zu einer Überschrift verarbeitet, was im Allgemeinen bedeutet den Text größer und fetter zu darzustellen.

Auch solche Befehle können wir uns selbst definieren. Beispielsweise wollen wir in unserem Dokument alle Namen von Programmiersprachen in einer Schreibmaschienschrift setzen.

```
\newcommand{\programmiersprache}[1]{\texttt{#1}}
```

Im Text können wir das nun Nutzen:

```
Er programmierte in \programmiersprache{python}, obwohl  
er
```

Angst vor Schlangen hatte.

Genau wie bei einer Überschrift verarbeitet LaTeX unser Argument (der Teil in geschweiften Klammern). LaTeX muss allerdings vorher wissen, wie viele Argumente (Klammerpaare) vorhanden sein werden ([1]). In diesem fälle haben wir nur ein Klammerpaar, auf das wir uns mit # 1 beziehen. Das selbst wiederum ist das Argument für \texttt, unsere Schreibmaschinenschrift. @sputello liefert ein paar hilfreiche Tipps zu semantischem Markup in [Wie kann ich bestimmte Begriff hervorheben?](#)

Natürlich können wir uns auch Befehle definieren, welche mehr als nur ein Argument haben:

```
\newcommand{\deadpoet}[3]{#1 (#2 -- #3)}  
%\deadpoet{<name>}{<born>}{<died>}
```

Im Dokument können wir uns nun auf den toten Dichter beziehen:

```
\deadpoet{Goethe}{1759}{1832} trug wesentlich zur  
Entwicklung  
der deutschen Sprache bei.
```

Ziel eines Dokumentes ist, vom Inhalt abgesehen, ein sauberes und einheitliches Erscheinungsbild. Semantisches Markup bietet

uns die besten Chancen dieses Erscheinungsbild zu erreichen.

```
\renewcommand{\deadpoet}[3]{#1 (geboren #2, gestorben #3)}  
\deadpoet{Goethe}{1759}{1832} trug wesentlich zur  
Entwicklung  
der deutschen Sprache bei.
```

Hier haben wir unseren Befehl `\deadpoet` nochmal definiert. Das Ergebnis in der pdf sieht für alle nachfolgenden Ereignisse eines *toten Dichters* gleich aus.

Was bedeutet das jetzt für unser Dokument?

Ich muss mir nur einmal festlegen, wie etwas aussehen soll, LaTeX kümmert sich dann darum, dass es nachfolgend einheitlich aussieht.

Wo sollte ich meine Befehle definieren, wenn ich möchte, dass *alle* Ereignisse im Dokument einheitlich sind?

Im Allgemeinen kann man sagen, dass die Definition eigener Befehle oder Makros immer in den Vorspann, also die Präambel, gehört.

Es gibt natürlich noch mehr zu entdecken, man denke nur an das optionale Argument von `\section`, welches seinen Inhalt in das Inhaltsverzeichnis schreibt.

Einen ausführlicheren Überblick gibt [Wie definiere ich eigene Kommandos?](#) von Marco Daniel. Darin werden auch Vorteile und Möglichkeiten des Paketes `xparse` vorgestellt.

Unser Minimalbeispiel ist mittlerweile Angewachsen auf:

```
\documentclass{article}  
\newcommand{\fragen}{Wieso? Weshalb? Warum?}
```

```

\newcommand{\programmiersprache}[1]{\texttt{#1}}
\newcommand{\deadpoet}[3]{#1 (#2 -- #3)}
\newcommand{\fragenB}[1]{Wieso #1\par Weshalb #1?\par
  Warum #1?}
\begin{document}
Es blieben einige Fragen: \fragen

Er selbst bezeichnete sich als \emph{\TeX niker aus
Leidenschaft}, den er liebte \TeX sehr.

Er selbst bezeichnete sich als \emph{\TeX niker aus
Leidenschaft}, den er liebte \TeX{} sehr.

Er programmierte in \programmiersprache{python}, obwohl
er
Angst vor Schlangen hatte.

\deadpoet{Goethe}{1759}{1832} trug wesentlich zur
  Entwicklung
der deutschen Sprache bei.

\renewcommand{\deadpoet}[3]{#1 (geborenn #2, gestorben
  #3)}
\deadpoet{Goethe}{1759}{1832} trug wesentlich zur
  Entwicklung
der deutschen Sprache bei.

\fragenB{sah er traurig auf die verregnete Gasse}
\end{document}

```

Von Johannes auf [TeXwelt.de](https://www.texwelt.de) geschrieben: ☞ Wie kann ich oft wiederkehrende Strukturen automatisieren?

45 Wie kann ich innerhalb des optionalen Arguments einer Anweisung zusätzliche eckige Klammern, beispielsweise die eines optionalen Arguments einer anderen Anweisung, verwenden?

Im Unterschied zu den geschwungenen Klammern $\{ \dots \}$, die immer paarweise und korrekt geschachtelt werden müssen, sucht LaTeX bei eckigen Klammern einfach nach der nächsten schließenden eckigen Klammer `]`, ohne auf korrekte Schachtelung dieser Klammern zu achten. Um dies zu verhindern, muss man einfach innerhalb der eckigen Klammern `[...]` ein geschweiftes Klammerpaar einfügen, wie dies hier als Beispiel für `\twocolumn` gezeigt wird:

```
\twocolumn[ { ... \sqrt[3]{x^2} ... } ]
```

46 Gibt es einen einfachen Weg, die Seitennummer als »Seite 1 von 16« anzugeben?

Es gibt verschiedene Pakete, die die letzte Seitennummer ausgeben. Es folgt ein Beispiel für das Paket `zref`.

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage{zref-lastpage,zref-user}
\begin{document}
```

Letzte Seite ist: `\zpageref{LastPage}`.

```
\clearpage
(Leer)
\end{document}
```

47 Was machen `\makeatletter` und `\makeatother`?

Alle Zeichen in LaTeX besitzen einen Kategoriecode (category code, catcode). Insgesamt gibt es 16 catcodes, wobei z.B. normalen Buchstaben (a-z,A-Z) den catcode 11 erhalten. Sonderzeichen wie `\` (catcode 0) oder `{` (catcode 1) und eben auch `@` (catcode 12) werden daher nicht als normale Buchstaben behandelt und dürfen somit vom normalen Benutzer nicht in Makros verwendet werden. Deshalb sind z.B. Makronamen wie `\foo123` oder `\foo!` normalerweise ungültig (allerdings gibt es Wege, solche Makronamen zu erzeugen, z.B. mit `\csname ... \endcsname`).

In Klassen- (.cls) und Stildateien (.sty) hat `@` jedoch den catcode 11 und somit sind Makros wie z.B. `\@foo` zulässig. Dies erlaubt, dass solche Definitionen vom normalen Benutzer geschützt/versetzt werden um z.B. ein versehentliches Überschreiben eines Makros zu verhindern.

Manchmal ergibt sich jedoch dennoch der Bedarf, interne Klassen- oder Stilmakros in der eigenen Datei zu modifizieren. Dazu möchte man nicht immer gleich eine eigene Klassen- oder Stildatei anlegen, sondern die Änderung direkt im eigenen Dokument vornehmen (meist in der Präambel). Dafür können dann folgende Befehle verwendet werden:

```
\makeatletter % ändert @ catcode von 12 zu 11  
---Definition---  
\makeatother % ändert @ catcode zurück von 11 zu 12
```

`makeatletter` bedeutet dabei „mache das `@` zu einem normalen Buchstaben“, `makeatother` „gebe dem `@` wieder die spezielle Funktion“.

Ein Beispiel ist das Ändern der Position von Gleitobjekten auf sonst leeren Seiten:

```
\makeatletter
\setlength{\@fptop}{0pt} % Gleitobjekt oben statt
    mittig auf sonst leerer Seite
\makeatother
```

Vollständigkeitshalber hier noch eine Liste aller catcodes und ihrer Bedeutungen:

0	Escapesymbol, Befehlsanfang	\
1	Beginn einer Gruppe	{
2	Ende einer Gruppe	}
3	Anfang und Ende des Inline-Mathemodus	\$
4	Tabulatorzeichen (in Tabellen)	&
5	Zeilenende (in der Eingabe)	
6	Parametersymbol in Makros	#
7	mathematische Hochstellung	^
8	mathematische Tiefstellung	_
9	zu ignorierendes Zeichen	
10	Leerzeichen	
11	Buchstaben	A-Za-z
12	sonstiges Zeichen (alles, was nicht in 1–11 und 13–15 genannt ist)	~, !
13	aktives Zeichen	~
14	Kommentarbeginn	%
15	ungültiges bzw. nicht erlaubtes Zeichen	

48 *Ich erhalte die Fehlermeldung »! You can't use \spacefactor in vertical mode.« oder »Command \@ already defined.«. Mache ich etwas falsch?*

In den meisten Fällen liegt die Schuld an der Verwendung oder Neudefinition eines Makros beginnend mit \@, also einem At-Zeichen @ am Anfang des Namens, in einem Dokument. Das At-Zeichen @ ist aber nicht als Bestandteil eines Makronamens innerhalb des Dokuments erlaubt.

Beispielsweise tritt der Fehler bei der Definition

```
\renewcommand{section}{%
  \@startsection {section}{1}{\z@}{-3.5ex plus-1ex
    minus-.2ex}%
    {2.3ex plus.2ex}{\normalfont\Large\
      bfseries}}
```

in der Dokumentpräambel auf.

Vermeiden kann man diesen Fehler, indem man die Definition in eine .sty-Datei verschiebt oder in der Präambel mit \makeatletter . . . \makeatother klammert, wie beispielsweise

```
\makeatletter
\@removefromreset{footnote}{chapter}
\makeatother
```

Siehe auch den Abschnitt über  *Was machen \makeatletter und \makeatother?*.

49 Warum verschluckt LaTeX ein Leerzeichen nach manchen Befehlen?

Nach einigen Befehlen verschluckt TeX bzw. LaTeX die Leerzeichen, nach manchen anderen aber nicht. Das liegt daran, dass in TeX zwischen sogenannten *control words* und *control symbols* unterschieden wird. *control words* sind Befehle, die aus einem Schrägstrich und nachfolgenden Buchstaben bestehen, wie z.B. `\section` oder `\LaTeX`. Nach diesen Befehlen werden alle Leerzeichen ignoriert. Im ersten Fall steht meistens `\section{...}` im Dokument, so dass diese Regel hier nicht greift. Im zweiten Fall schreibt man aber Sätze wie „Mit `\LaTeX` kann ich tolle Dokumente erstellen“. Da hier das Leerzeichen als Befehlsende gelesen wird, verschwindet es aus dem Text. Eine Lösung besteht darin, das Ende des Befehls durch das Befehlsmakro „`\`“ zu markieren: „Mit `\LaTeX\` kann ich tolle Dokumente erstellen“. Vgl. TeXBook Seite 8ff.

49.1 Alternative: Das Paket *xspace*

Das Paket `xspace` löst das *Problem*, in dem es an passender Stelle ein Leerzeichen einfügt, was besonders bei der Definition von Befehlen hilfreich ist.

```
\usepackage{xspace}
...
\newcommand\ltx{\LaTeX\xspace}
```

Mit `\ltx` lassen sich tolle Dokumente erstellen. Ich benutze `\ltx`.

Hier wird in dem Satz ein Leerzeichen eingefügt, vor dem Punkt ganz am Ende aber nicht:

Mit L^AT_EX lassen sich tolle Dokumente erstellen. Ich benutze L^AT_EX.

50 *Wie erhalte ich automatisch Großbuchstaben?*

Wenn man einen Text hat, der in Großbuchstaben gewandelt werden soll, so gibt es bei LaTeX den Befehl `\MakeUppercase`

```
\documentclass{article}
\usepackage[ngerman]{babel}
\begin{document}
\MakeUppercase{Hallo Welt}
\end{document}
```

Den TeX-Befehl `\uppercase` sollte man nicht benutzen, da er nicht alle Fälle erschlägt:

```
\MakeUppercase{\oe} und \uppercase{\oe}
```

ergibt die Zeichen Œ und œ . Das zweite ist nicht das gewünschte Ergebnis.

Natürlich gilt das geschriebene analog zu den Kleinbuchstaben, nur halt mit `\MakeLowercase` und dem Befehl `\lowercase`, der in LaTeX nicht benutzt werden soll.

51 Wie kann ich Wörter wie »Inhaltsverzeichnis«, »Abbildung« etc. umbenennen?

51.1 KOMA-Script

Die Klassen aus »KOMA-Script« stellen dafür einen speziellen Befehl bereit.

```
\renewcaptionname<{>Sprache<} {>Begriff<} {>Inhalt}
```

also z.B.

```
\renewcaptionname{ngerman}{\contentsname}{Inhalt}
\renewcaptionname{ngerman}{\listfigurename}{Abbildungen}
\renewcaptionname{ngerman}{\listtablename}{Tabellen}
\renewcaptionname{ngerman}{\figurename}{Abb.}
\renewcaptionname{ngerman}{\tablename}{Tab.}
```

51.2 Basisklassen

Für die »normalen« Klassen wie `article.cls` bei Benutzung von Babel gibt es diese Möglichkeit:

```
\addto{\captionsngerman}{%
  \renewcommand*{\contentsname}{Inhalt}
  \renewcommand*{\listfigurename}{Abbildungen}
  \renewcommand*{\listtablename}{Tabellen}
  \renewcommand*{\figurename}{Abb.}
  \renewcommand*{\tablename}{Tab.}
}
```

(Angelehnt an eine / übernommen aus einer [Antwort](#) aus dem deutschsprachigen Forum)

52 *Wie kann ich festlegen, welche Gliederungsebenen im Inhaltsverzeichnis erscheinen sollen?*

Es gibt einen Zähler (counter), der die Tiefe des Inhaltsverzeichnisses bestimmt. Das heißt also, welche Gliederungsebenen im Inhaltsverzeichnis erscheinen.

Die Voreinstellung bei der Klasse Buch (`book.cls`) ist z.B. 2. Damit werden die »subsections« im Inhaltsverzeichnis aufgelistet. Möchte man nur z.B. die Ebene `section` und darüber im Verzeichnis haben, so hilft ein

```
\setcounter{tocdepth}{1}
```

- 3 subsection und höher
- 2 subsection und höher
- 1 section und höher
- 0 chapter und höher
- 1 part
- 2 Keine Einträge im Inhaltsverzeichnis

53 *Was ist BibTeX?*

BibTeX ist ein Programm, mit dem man Literaturverweise und Bibliographien erstellen kann. Es benutzt dazu »Anker« (`\cite`-Befehle) in einem LaTeX-Dokument, die es zunächst sammelt und in einem zweiten Schritt die dazugehörigen Daten aus einer separaten Datei mit der Endung `.bib` entnimmt. Die Darstellung der Verweise und des Literaturverzeichnisses lässt sich durch

`bst`-Dateien beeinflussen (`bst` steht für »bibliography style«). Es gibt eine Reihe von Bibliographiestilen, die unterschiedliche Vorgaben erfüllen. Eine Auswahl davon kann man sich anhand von Beispielen auf der Seite [BibTeX style examples](#) anschauen. Die Anpassung von Bibliographiestilen ist nicht gerade trivial, da sie in einer Programmiersprache abgefasst sind, die in Form einer Stackmaschine mit nur wenigen Befehlen funktioniert (↺Umgekehrte Polnische Notation).

BibTeX wurde 1988 geschrieben (bis Version 0.99c) und 2010 aktualisiert (Version 0.99d).

(FIXME)

- kurz etwas zu den Beschränkungen
- `bibtex8`
- `bibtexu`

54 Was ist *biber*?

Biber ist ein Ersatz für [BibTeX](#). Beide Programme dienen dazu, Fundstellen bzw. Belege für Zitate und Bibliographien in LaTeX zu erstellen. Anders als BibTeX, dient Biber vorwiegend zum Sortieren der Bibliographie, während die Formatierung der Zitatnachweise und der bibliographischen Angaben (Zitier- und Bibliographiestile) von dem LaTeX-Paket `biblatex` und den auf `biblatex` aufbauenden weiteren LaTeX-Paketen (weiteren Zitier- und Bibliographiestilen) übernommen werden.

Biber ist in der Programmiersprache Perl geschrieben. Die regulären Ausdrücke von Perl können somit bei der Formatierung und gegebenenfalls bei der weiteren Bearbeitung der Literaturverzeichniseinträge genutzt werden.

Einer der Vorteile von `biber` gegenüber BibTeX ist der Unicode-Support.

Biber ist in aktuellen Versionen von TeX Live und MiKTeX verfügbar. Es sollte immer in der zur verwendeten `biblatex`-Version passenden Version installiert werden. Bei MiKTeX muss man dazu ggf. im Paketmanager (bei MultiUser-Installation der Admin-Version) das Paket `miktex-biber-bin` installiert werden. Das `biblatex` auch über den Benutzerpaketmanager installiert sein kann, müssen in der Regel Updates für `biber` im Admin-Update-Manager zeitgleich zu Updates von `biblatex` im normalen Update-Manager vorgenommen werden. Alternativ kann man auch `biblatex` (und davon abhängige Pakete) im Admin-Paketmanager installieren und im normalen Paketmanager entfernen. Dann muss nur Admin die Updates durchführen.

In früheren 64-bit MiKTeX-Installation musste `biber` von Hand installiert werden. Dies geschah durch Download der Binärdatei von [sourceforge](https://sourceforge.net/projects/biber/) und kopieren der `exe` in das MiKTeX binaries Verzeichnis. Anstatt dies für ältere Systeme zu tun, sollte ein Update in Erwägung gezogen werden.

Mit Erläuterungen von Johannes von [TeXwelt.de](https://www.texwelt.de/): [Wie verwende ich biber in meinem Editor?](https://www.texwelt.de/faq/biber/)

55 *Wie verwende ich biber in meinem Editor?*

In einer `⌘`Eingabeaufforderung/Terminal wird `biber` auf den Namen des Hauptdokumentes ohne Dateierweiterung (`basename`) angewendet. Also beispielsweise:

```
biber MeinDokument
```

Wo muss ich in den geläufigen Editoren *was* einstellen, um `biber` auf Knopfdruck ausführen zu können?

Für diese LaTeX-Editoren und Tools (in alphabetischer Reihenfolge) gibt es detaillierte bebilderte Antworten hier:

- `⌘`arara (kein Editor aber trotzdem interessant)
- `⌘`Emacs mit AucTeX
- `⌘`Emacs mit TeX.el (grundlegender TeX-mode statt AucTeX)
- `⌘`Kile
- `⌘`Texmaker
- `⌘`TeXnicCenter
- `⌘`TeXshop
- `⌘`TeXstudio
- `⌘`TeXworks

Diese Infos wurden auf Johannes' Archiv-Frage auf [TeXwelt.de](https://www.texwelt.de) `⌘`dort zusammengetragen.

56 Was sind .bib-Dateien?

.bib-Dateien enthalten die bibliographischen Daten für die Erzeugung von Zitatnachweisen und Bibliographien mit LaTeX und BibTeX bzw. bei Verwendung von biblatex: vorzugsweise mit Biber. Die einzelnen Einträge bestehen aus einer Datensatzkennung (z.B. Meyer : 1999) und mehreren Schlüssel-Wert-Paaren für die verschiedenen bibliographischen Kategorien, z.B. `author = {Meier, Wilhelm}` und `journal = {Zeitschrift für Kleintierkunde}`.

.bib-Dateien sind reine Textdateien und können deshalb mit einem beliebigem Texteditor bearbeitet werden. Alternativ gibt es Frontends zur Literaturverwaltung, die eine Bearbeitung der Datensätze mit Eingabemasken unterstützen. Eines der besten  Literaturverwaltungsprogramme ist  JabRef, bei dem man sowohl mit Eingabemasken als auch im BibTeX-Quelltext arbeiten kann. Auf Apple Macintosh steht  BibDesk zur Verfügung.

Ob eine .bib-Datei BibTeX- oder biblatex-Daten erhält, ist allein aufgrund des Dateinamens nicht zu sagen, sondern nur aufgrund der in der Datei verwendeten Kategorien. BibTeX unterstützt von Haus aus nur wenige bibliographische Kategorien, während biblatex hier mehr Auswahl bietet. Teilweise weicht der Gebrauch auch vom ursprünglichen BibTeX ab; das sollte man in der Praxis beachten. Beispielsweise unterstützt Biblatex verschiedene auch von BibTeX definierte Kategoriebezeichnungen als Auffanglösung (Fallback). So wird etwa in BibTeX das Feld `year` verwendet, in biblatex dagegen `date`. Wenn nur das Jahr benötigt wird, 'versteht' biblatex auch 'year'. Wenn aber ein Datum angegeben werden soll, muss man 'date' verwenden. Ist eine .bib-Datei erst einmal

für biblatex optimiert, wird man bei Benutzung derselben Datei mit Zitierstilen, die auf BibTeX beruhen, wahrscheinlich keine brauchbaren Ergebnisse mehr erhalten.

57 Was ist biblatex?

▫ Biblatex ist ein Makropaket zum Erstellen von Literaturverzeichnissen mit LaTeX und BibTeX, das bereits mehrere Zitierstile enthält und eine hohe Flexibilität aufweist. Das Paket wurde im August 2006 von Philipp Lehman erstmals vorgestellt. Seit der Version 0.9 ist es auch in den gängigen TeX-Distributionen wie MiKTeX oder TeX Live enthalten. Seit April 2012 wird Biblatex von Philip Kime, Audrey Boruvka und Joseph Wright weiterentwickelt.

Anders als »herkömmliche« Bibliographie-Pakete, benutzt biblatex das Programm BibTeX nur zum Sortieren der Einträge im Literaturverzeichnis und zum Erzeugen von Labels. Anstelle von BibTeX wird mittlerweile aber empfohlen, biblatex in Verbindung mit dem BibTeX-Ersatz  Biber zu verwenden. Die Formatierung von Fundstellen und Bibliographie wird vollständig mit LaTeX-Befehlen vorgenommen, was es – im Unterschied zu BibTeX – einfacher macht, die vorhandenen Bibliographiestile anzupassen oder eigene Stile zu schreiben. Biblatex unterstützt auch gegliederte Bibliographien, mehrere Bibliographien innerhalb eines Dokuments und separate Listen bibliographischer Abkürzungen (Sigelverzeichnisse).

Es gibt auch eine  deutsche Übersetzung der Dokumentation zu Biblatex.

58 *Wie kann ich alle Einträge aus meinem Literaturverzeichnis ausgeben lassen?*

Ganz einfach, mit `\nocite{*}`. Nachfolgend ein Beispiel für BibLaTeX mit Biber, das funktioniert aber genau so mit dem klassischen Weg über BibTeX:

```
\documentclass[ngerman]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{babel, csquotes}
\usepackage[backend=biber]{biblatex}
\addbibresource{l2kurz.bib}
\begin{document}
\nocite{*}
\printbibliography
\end{document}
```

59 *Gibt es bequeme Editoren für BibTeX-Literaturdatenbankdateien?*

Die folgende Liste mit einigen Editoren und Frontend-Programme, zur Eingabe und Manipulation von BibTeX-Literaturdatenbanken, ist noch sehr unvollständig:

59.1 *Freie Software*

-  Emacs mit dem BibTeX-Mode: Der Editor Emacs ist für alle gängigen Betriebssysteme verfügbar.
-  JabRef: Java Programm für alle Java-Plattformen.

- *z*BibDesk für Mac OS X.
- Eine freie Lösung, die Mikroformate aus Bibliothekskatalogen und anderen Websites direkt beim Surfen aus dem Webbrowser in bibliographische Daten übernehmen kann, ist *z*Zotero.

Mit Emacs können BibTeX-Dateien nur bearbeitet werden. Literaturverwaltungen wie JabRef, BibDesk oder Zotero können dagegen Daten über spezielle Schnittstellen aus Bibliothekskatalogen abrufen.

Eine webbasierte Cloud-Lösung, mit der man BibTeX-Daten importieren und exportieren kann, ist der Social-Bookmarking-Dienst *z*BibSonomy, der gemeinsam von den Universitäten Kassel, Würzburg und Hannover betrieben wird. Mit BibSonomy können sowohl Literaturangaben als auch Bookmarks gesammelt werden. Auch Gruppenarbeit ist damit möglich. Bibliographische Daten kann BibSonomy direkt aus dem Worldcat abrufen.

Außerdem können viele Bibliothekskataloge Daten im BibTeX-Format exportieren, so dass man sie leicht lokal weiter verarbeiten kann.

59.2 *Proprietäre Lösungen*

- *z*Endnote: Windows und Mac OS X
- *z*Citavi: Windows (für den Bildungssektor ist eine EDU-Lizenz erhältlich)

Ein Vergleich gängiger kostenpflichtiger Software für die Litera-

turverwaltung findet sich unter www.slub-dresden.de/.../Literaturverwaltungsprogramme_im_Überblick.pdf

60 Wie schreibe ich einen Brief mit LaTeX?

Um Briefe zu schreiben, bietet sich die Klasse `scr1ttr2` aus dem LaTeX-Bundle KOMA-Script an (ein „Bundle“ enthält mehrere LaTeX-Pakete). Ein kleines Beispiel:

```
\documentclass[ngerman]{scr1ttr2}

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

% Angaben zum Versender
\setkomavar{fromname}{Lieschen Müller}
\setkomavar{fromaddress}{Musterstraße 456\\67890
  Musterstadt}
\setkomavar{fromphone}{0123 456 7890}
\setkomavar{signature}{L. Müller}

\begin{document}

% Betreff des Briefes
\setkomavar{subject}{Bewerbung für eine Stelle}
\begin{letter}{% Adressat
Firma Mustermann\\%
Musterstraße 123\\%
12345 Musterstadt}
  \opening{Sehr geehrte Damen und Herren,}

  ich möchte mich gerne auf eine Stelle bewerben.

  \closing{Mit freundlichem Gruß}
  \ps PS: Ich freue mich auf ein erneutes Treffen
  \encl{Lebenslauf, Zeugnisse}
\end{letter}
```

Das Ergebnis dieses Beispiels sieht so aus:

Lieschen Müller
Musterstraße 456
67890 Musterstadt

Lieschen Müller, Musterstraße 456, 67890 Musterstadt

Firma Mustermann
Musterstraße 123
12345 Musterstadt

11. März 2013

Bewerbung für eine Stelle

Sehr geehrte Damen und Herren,
ich möchte mich gerne auf eine Stelle bewerben.

Mit freundlichem Gruß

L. Müller

PS: Ich freue mich auf ein erneutes Treffen

Anlage(n): Lebenslauf, Zeugnisse

61 Wie erzeuge ich Überschriften bzw. Gliederungen in Briefen?

Briefe enthalten üblicherweise keine Überschriften oder Gliederungen. Diese können aber bei längeren Briefen sinnvoll sein, beispielsweise bei längeren juristischen Schriftsätzen. Hierfür gibt es die Letter-Class-Options-Datei `sections.lco`.

Für den Satz von Verträgen und Gesetzestexten stellt Koma-Script das `scrjura` Paket bereit, das zahlreiche spezialisierte Funktionen wie beispielsweise das Referenzieren auf einzelne Sätze eines Paragraphen bietet.

62 Wie schreibe ich Zahlen mit Einheiten richtig?

In naturwissenschaftlichen und technischen Texten verwendet man physikalische Einheiten wie Gramm, Meter, Kilojoule u.a. Es gibt wissenschaftlich-technischen Normen und typographischen Gepflogenheiten, denen man möglichst folgt.

Dazu gehören:

- Abstand zwischen Zahl und Einheit
- aufrechte Schriftart der Einheiten (sonst sehen g für Gramm und m für Meter genauso aus wie die Variablen g und m)
- Erleichterung und Konsistenz beim Schreiben wie durch Abkürzungen mit Makros

62.1 Der beste Weg

Verwende das Paket `siunitx`.

Wieso? Das Paket stellt grundsätzlich das Makro `\SI{<Zahl>}{<heit>}` bereit, mit dem schon mal zwei der Punkte geklärt werden: der Abstand zwischen Zahl und Einheit und die die richtige Schriftwahl für die Einheiten. Und das Makro selbst erfüllt dann Punkt drei Deiner Frage.

```
\documentclass{article}
\usepackage{siunitx}
\begin{document}
\[v=\SI{2.3}{\metre\per\second}\]
\end{document}
```

$$v = 2.3 \text{ m s}^{-1}$$

Das Paket stellt zahlreiche Möglichkeiten zur Anpassung bereit:

```
\documentclass{article}
\usepackage{siunitx}
\sisetup{
  locale = DE ,
  per-mode = symbol
}
\begin{document}
\[v=\SI{2.3}{\metre\per\second}\]
\end{document}
```

$$v = 2,3 \text{ m/s}$$

Daneben hat das Paket noch eine ganze Reihe weiterer nützlicher Funktionen: Makros (`\num`) zum ebenso leichten Angeben von Zahlenwerten, die (auch bei `\SI`) z.B. gerundet werden können, in wissenschaftlicher Schreibweise ausgegeben werden können, ... Auch Gradzahlen kann man mit `siunitx` leicht als solche markieren (`\ang`).

Nicht zuletzt stellt es Möglichkeiten bereit, um Zahlen in Tabellen formatiert und ausgerichtet zu setzen.

Obwohl die Makros in den Beispielen im Mathemodus eingesetzt wurden, ist das keine Voraussetzung. *Alle* von `siunitx`' Makros funktionieren sowohl im Text- als auch im Mathe-Modus.

Das Paket ist der offizielle Nachfolger der Pakete `SIstyle`, `si` und `siunits` (sie werden alle vom gleichen Autor maintained). Andere Pakete in diesem Zusammenhang laufen IMHO unter ferner liefen. `siunitx` hat auch von `dcolumn` viele Anregungen übernommen, übertrifft es aber in einigen Bereichen.

62.2 Fortgeschrittenes

Man kann außerdem für einzelne physikalische Größen Makros definieren, um die Verwendung zu vereinfachen (und Tipparbeit zu sparen):

```
\DeclareSIUnit{\dicte}{\gram\per\cubic\centi\meter}
\DeclareDocumentCommand\Dichte{0{m}}{\SI[#1]{#2}{\dicte}}
```

Macht man das für viele verschiedene Größen, lohnt es sich, das etwas zu automatisieren. Der hier verwendete Befehl `\NewDocumentCommand` stammt vom Paket `expparse`, das von `siunitx` bereits geladen wird:

```
\documentclass{article}
\usepackage{siunitx}

\NewDocumentCommand\DeclareNewQuantity{mmm}{%
  \DeclareSIUnit{#2}{#3}%
  \DeclareDocumentCommand{#1}{0{m}}{\SI[##1]{##2}{#2}}%
}

\DeclareNewQuantity
  \Dicte
  \dicte
  {\gram\per\cubic\centi\meter}
\DeclareNewQuantity
  \Stoffmenge
  \stoffmenge
  {\mole}
\DeclareNewQuantity
  \Molalitaet
  \molalitaet
  {\mole\per\kilo\gram}

\begin{document}

\[ \rho = \Dicte{1.02} \]

\end{document}
```

62.3 Quelle

Verfasst von Clemens auf [TeXwelt.de](https://www.texwelt.de): [Wie schreibe ich Zahlen mit Einheiten richtig?](#).

63 *Wie schreibe ich griechische Buchstaben aufrecht?*

Im Fließtext kann man griechische Buchstaben haben, die nicht kursiv wie mathematische Variablen sein sollen, sondern aufrecht. Beispiele sind γ -Strahlung, α -Strahlung und β -Strahlung. Aufrechte griechische Buchstaben werden häufig in der Chemie benötigt.

63.1 *LaTeX-Pakete und Schriften*

Die einfachste Möglichkeit ist `\upgreek`:

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage{upgreek}
\begin{document}

 $\upalpha$ -Strahlung,
 $\upbeta$ -Strahlung,
 $\upgamma$ -Strahlung

\end{document}
```

α -Strahlung
 β -Strahlung
 γ -Strahlung

Liste von geeigneten Paketen und Schriften:

- `\upgreek`
- `\textgreek`
- `\newtx`
- `\kpfonts`
- `\mathdesign`
- `\fourier`
- `\chemmacros`

Mit dem `textgreek`- oder dem `textalpha`-Paket, mit den Macros `\textalpha`, `\textbeta`, `\textgamma` ... oder – mithilfe der richtigen Kodierung – auch mit den richtigen griechischen Buchstaben α , β und γ :

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{textalpha}
\begin{document}
```

```
\textalpha-Strahlung, \textbeta-Strahlung, \textgamma-Strahlung
-Strahlung, -Strahlung, -Strahlung
\end{document}
```

Ein Beispiel mit dem Paket chemgreek aus dem chemmacros-Bundle:

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}

% Paket der Wahl, 'chemmacros' wird das registrieren;
% bei konkurrierenden
% Paketen wie hier wählt 'chemmacros' nach Priorität
% tenliste;
\usepackage{kpfonts}
\usepackage[artemisia]{textgreek}
\usepackage{upgreek}

% Überschreiben der 'kpfonts' für den normalen Text:
\usepackage{lmodern}

% zur Sicherheit die Priorität überschreiben:
\usepackage[greek=upgreek]{chemmacros}[2014/01/24]

% Verwenden der generischen Makros in semantischen:
\DeclareChemParticle\alphaparticle{\chemalpha}
\DeclareChemParticle\betaparticle{\chembeta}
\DeclareChemParticle\gammaparticle{\chemgamma}

\begin{document}

\begin{tabular}{p{.3\textwidth}p{.3\textwidth}p{.3\textwidth}}
\texttt{upgreek} & \texttt{textgreek} & \texttt{kpfonts} \\
\alphaparticle-Strahlung & \betaparticle-Strahlung & \gammaparticle-Strahlung
\end{tabular}
\selectchemgreekmapping{textgreek}
```

```

\alphaparticle-Strahlung\par
\betaparticle-Strahlung\par
\gammaparticle-Strahlung &
\selectchemgreekmapping{kpfonts}
\alphaparticle-Strahlung\par
\betaparticle-Strahlung\par
\gammaparticle-Strahlung
\end{tabular}
\end{document}

```

upgreek	textgreek	kpfonts
α -Strahlung	α -Strahlung	α -Strahlung
β -Strahlung	β -Strahlung	β -Strahlung
γ -Strahlung	γ -Strahlung	γ -Strahlung

63.2 Mit babel

Mit der greek-Option des babel-Pakets (polutonikogreek erlaubt auch akzentuierte griechische Buchstaben):

```

\documentclass{article}
\usepackage{lmodern}
\usepackage[greek, ngerman]{babel}
\newcommand*\textGreek[1]{\greektext#1}
\begin{document}
\textGreek{a}-Strahlung, \textGreek{b}-Strahlung, \
\textGreek{g}-Strahlung
\end{document}

```

Es bietet sich natürlich an, gleich `\ta`, `\tb` und `\tg` zu definieren, wenn das das einzige Griechisch ist, das man verwenden wird.

63.3 Mit XeLaTeX oder LuaLaTeX

Außer `fontspec` ist hier kein weiteres Paket nötig. Nur die gewählte Schriftart muss Glyphen dafür bereitstellen.

```
% !TEX TS-program=xelatex
% !TEX encoding = UTF-8 Unicode
\documentclass{article}
\usepackage[ngerman]{babel}

\usepackage{fontspec}
% Latin Modern Roman unterstützt leider keine
% altgriechischen Zeichen,
% aber Schriftarten wie TeX Gyre Pagella, TeX Gyre
% Termes oder
% die "neuen" Microsoft C-Schriftarten und vermutlich
% viele andere
\setmainfont{CMU Serif}

\begin{document}
-Strahlung, -Strahlung und -Strahlung oder besser "=
  Strahlung, "=Strahlung und "=Strahlung

Oder wenn man es lieber mag: \textalpha"=Strahlung, \
  textbeta"=Strahlung und
\textgamma"=Strahlung.
\end{document}
```

`fontspec` stellt freundlicherweise auch die `\textalpha`, `\textbeta`, etc. Befehle bereit.

α -Strahlung, β -Strahlung und γ -Strahlung oder besser α -Strahlung, β -Strahlung und γ -Strahlung.
Oder wenn man es lieber mag: α -Strahlung, β -Strahlung und γ -Strahlung.

Von Clemens, Qrrrbirlbel und quinmars auf [TeXwelt.de](http://www.texwelt.de): [Wie schreibe ich griechische Buchstaben aufrecht?](#)

64 *Formelzeichenverzeichnis*

Für die Erstellung von Formelzeichen-Verzeichnissen bietet sich das Paket `nomencl` von Boris Veytsman und anderen Autoren an. Das Paket benötigt `makeindex`.

Eine weitere Option ist das sehr moderne Paket `glossaries` von Nicola Talbot.

65 *Wie kann ich meinen TeX-Lauf automatisieren?*

Moderne Editoren vereinfachen das Arbeiten und somit das Kompilieren von LaTeX-Dokumenten. Diese Editoren bedienen sich im eigentlichen Sinne nur Compilern, die im Suchpfad des Systems hinterlegt sind. Das Setzen eines einfachen Briefes oder kleineres Essays ist daher mit einem Klick getan. Häufig führt der Klick PDFLaTeX, LuaLaTeX oder XeLaTeX aus.

Die Anforderungen an ein Textsatzsystem steigen kontinuierlich. Beispielsweise kommt kaum ein Buch ohne einen Index oder ein Literaturverzeichnis aus. Wie diese erstellt werden, ist bereits in anderen Fragen besprochen worden. Zusammenfassend ist festzuhalten, dass ein Kompilierungslauf nicht ausreicht. Bspw. ist folgende Reihenfolge vorstellbar:

```
pdflatex
makeindex
makeglossaries
bibtex
pdflatex
pdflatex
```

Das bedeutet, dass der Editor an jedes Projekt angepasst werden muss, um die gewünschte Kompilierung mittels eines Buttons zu gewährleisten. Alternativ muss ich mindestens 4 Buttons haben, die verschiedene Compiler aufrufen. Genau an dieser Stelle setzen die unten aufgeführten Tools an. Es muss nur ein Button konfiguriert werden und die restlichen Schritte werden separat angepasst.

65.1 *arara*

`arara` ist nicht nur ein Vogel, sondern der Name eines „cool TeX automation tool“, wie es der Author Paulo R. M. Cereda bezeichnet. Dieses Werkzeug ist für TeX-Live-Nutzer direkt verfügbar. MiKTeX-Nutzer können es sehr einfach mittels des Installers installieren. Das Paket `arara` inklusive der Dokumentation ist

1. auf CTAN verfügbar: [↗"arara" auf CTAN](#)
2. auf der Entwicklerseite: [↗"arara" auf github](#)

Wie funktioniert arara?

In der Einleitung wurde bereits angesprochen, dass der Editor nur eine einzige Kompilierung durchführen muss, um alle benötigten Compiler aufzurufen. `arara` weiß selbst nicht, was wie kompiliert werden soll, man kann es ihm aber in der `.tex`-Datei mitteilen. Ein einfaches Beispieldokument soll dies verdeutlichen:

```
% arara: pdflatex
% arara: biber
% arara: pdflatex
```

```
% arara: pdflatex
\documentclass[a4paper, 12pt]{article}
\usepackage[style=verbose]{biblatex}
\addbibresource{biblatex-examples.bib}
\begin{document}
\cite{knuth:ct}
\printbibliography
\end{document}
```

Angenommen, die Datei heißt `beispieldokument.tex`, so muss dieses Dokument nur mittels

```
arara beispieldokument
```

kompiliert werden und alle benötigten Kompilierungen werden ausgeführt. Die hieraus erwachsenen Möglichkeiten sind hervorragend dokumentiert. Der Entwickler selbst ist auf [TeX.SX](#) sehr aktiv, so dass Fragen am ehesten dort beantwortet werden.

65.2 *latexmk*

`latexmk` ist ein Programm mit dem man ebenfalls einen TeX-Lauf bzw. den Aufruf zugehöriger Tools automatisieren kann, wobei hier der große Vorteil darin besteht, dass nur die Dateien neu generiert werden müssen, wenn sie sich (bzw. eine der Abhängigkeiten) geändert haben.

Angenommen die Datei `test.tex` enthält die folgenden Anweisungen:

```
\documentclass[ngerman]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{babel, csquotes}
\usepackage[backend=biber]{biblatex}
```

```
\addbibresource{l2kurz.bib}
\begin{document}
\nocite{*} % gibt alle Literaturstellen aus
\printbibliography
\end{document}
```

reicht ein einfaches

```
latexmk test -pdf
```

aus, um pdf_latex und  biber in der richtigen Reihenfolge und Anzahl durchlaufen zu lassen. Ruft man jetzt noch einmal `latexmk test -pdf` auf, erkennt latexmk automatisch, dass die PDF Datei noch *gültig* ist und führt keine Programme aus. Wenn sich beispielsweise nur die Literaturdatei geändert hat, wird diese erneut mit biber übersetzt und anschließend noch einmal pdf_latex. biber führt somit immer nur die benötigten Programme aus, was eine Menge Zeit spart.

latexmk lässt sich per Konfigurationsdatei dazu anweisen, immer eine bestimmte Datei bzw. Ausgabeformat zu wählen, so dass man das nicht mehr angeben muss. Für die LaTeX Kurzbeschreibung l2kurz wird folgende Datei eingesetzt²:

```
$pdf_mode = 1;
@default_files = ('l2kurz');
$pdflatex = 'pdflatex -synctex=1 %O %S';
```

Das gibt den Dateinamen vor und aktiviert den PDF-Modus und  synctex.

² <https://github.com/texdoc/l2kurz>

66 *Wie kann ich die Kodierung meines Dokuments bestimmen? Oder: warum funktionieren meine Umlaute nicht?*

Ein Dokument auf der Festplatte besteht aus einer langen Reihe von Zahlen, die erst beim Einlesen (von TeX) in Buchstaben interpretiert werden. So würde TeX die Zahlenfolge 72 97 108 108 111 als Wort »Hallo« lesen. Letztendlich ist das aber nur eine Übereinkunft, von der es aber leider viele unterschiedliche gibt.³ So ist nicht festgelegt, als welcher Buchstabe die Zahl 228 interpretiert wird. Es könnte ein »ä« sein, ein »%« oder ein ungültiges Zeichen. Das kommt auf die *Kodierung* an, mit der das Dokument gespeichert wurde. Üblich ist heutzutage UTF-8, aber es gibt noch genügend »Altlasten«, die einem das Leben schwer machen. Üblich waren früher Latin1 (=ISO 8859-1) und Windows-1252 und vereinzelt auch Mac Roman, wobei die ersten beiden Kodierungen sich stark überschneiden, nur in den Bereichen 128-159 unterschiedlich sind. Mit Windows-1252 kann man beispielsweise das Eurozeichen (128) darstellen.

Wenn man in einem Texteditor das Dokument speichert, passiert das immer in einer bestimmten Kodierung. Manchmal kann (oder muss) man sie angeben, oftmals ist das aber sehr verdeckt oder nicht sichtbar. Wichtig ist jedoch im Dokument anzugeben, mit welcher Kodierung das Dokument gespeichert ist. Da man das aber nicht immer weiß, hilft das Paket `selinput`. Das funktioniert mit allen Engines (Knuth, PDFTeX, LuaTeX, XeTeX). Hier ein vollständiges Beispiel:

³ Die Zahlen bis 127 sind im ASCII Standard festgelegt, die von allen Kodierungen eingehalten wird.

```

\documentclass[11pt]{article}
\usepackage[T1]{fontenc}
\usepackage{selinput}
\SelectInputMappings{
  adieresis={ä},
  germandbls={ß}
}

\begin{document}
Schöne Grüße vom Mädchen aus Österreich.
\end{document}

```

Damit erkennt TeX vollautomatisch die korrekte Kodierung und man ist von allen Sorgen diesbezüglich befreit.

66.1 *LuaTeX und XeTeX*

Bei diesen beiden Engines wird als Eingabe immer UTF-8 angenommen, so dass keine Zeile wie oben notwendig ist, solange das Dokument mit UTF-8 gespeichert wurde.

66.2 *TeXnicCenter*

TeXnicCenter speichert bis zur Version 2 (Stand 2013) die Dateien im Format »Windows-1252« ab und kann nicht mit UTF-8 umgehen. Version 2 und höher speichert die Dateien im Format UTF-8 ab.

66.3 Alternativen

Anstelle des Pakets `selinput` kann man die Kodierung auch direkt angeben, sofern man sie kennt:

```
\usepackage[latin1]{inputenc} % ISO-8859-1, Latin-1
\usepackage[utf8]{inputenc} % UTF-8
\usepackage[ansinew]{inputenc} % Windows CP 1252
```

man muss sich dann halt die passende Zeile herausuchen.

Tipp: Wenn man auf einen Unix/Linux System arbeitet, kann man manchmal mit dem Kommando `file` die Kodierung erfahren. Dazu gibt man den Dateinamen des Dokuments an.

```
$ file test.tex
test.tex: LaTeX 2e document, UTF-8 Unicode text
```

67 Wie funktioniert der Absatzumbruch in TeX?

TeX berechnet beim Zeilenumbruch zunächst alle möglichen Trennstellen eines Absatzes und ordnet jeder einzelnen bestimmte Strafpunkte zu, je nachdem, wie weit sie vom (imaginierten) idealen Layout abweicht. Um den Zeilenumbruch mit dem Rand des Satzspiegels in Deckung zu bringen, können die Leerstellen im Text in bestimmten Grenzen gedehnt oder gestaucht werden. Je größer die Abweichung von der Standardbreite der Leerstelle ist, desto höher die *badness* der entsprechenden Trennstelle. TeX berücksichtigt auch die benachbarten Zeilen, so dass nicht eine besonders lose (gedehnte) Zeile neben einer besonders dichten (gestauchten) zu liegen kommt, Unterlängen nicht direkt auf Oberlängen in der Folgezeile treffen usw. usf. Wenn auch bei voller Ausnutzung

der erlaubten Dehnung bzw. Stauchung der Zeilenumbruch mit dem Rand des Satzspiegels nicht in Deckung kommt und alle unerwünschten Effekte vermieden werden können, gibt TeX eine Warnung aus.

Die in TeX vorgegebenen Regeln für die zulässige Dehnung und Stauchung sind kompliziert und streng. Jede Regel kann durch Parameter einzeln geändert werden. Was für TeX schlampig gesetzt ist, sieht für unsere geschundenen Augen oftmals noch sehr sehr gut aus.

68 *SyncTeX*

SyncTeX ist eine Erweiterung zu den bekannten  engines PDFTeX, XeTeX und LuaTeX, die ein *Springen* zwischen Texteditor, PDF-Betrachter und zurück ermöglicht. Dazu müssen die beteiligten Programme das SyncTeX-Protokoll verarbeiten können. Beispielsweise arbeiten der PDF-Betrachter Skim (Mac OS X) und der Editor Sublime Text zusammen: Klickt man in der PDF-Datei auf das Wort »korrigieren« mit Cmd-Shift-Mausklick, öffnet sich der eingestellte Texteditor mit der entsprechenden Datei und springt zur richtigen Zeile.

- \LaTeX -Dokumente sind zwischen verschiedenen Installationen und Rechnerplattformen austauschbar.
- Im Gegensatz zu vielen WYSIWYG-Programmen bearbeitet \LaTeX auch lange oder komplizierte Dokumente zuverlässig, und sein Ressourcenverbrauch (Speicher, Rechenleistung) ist vergleichsweise mäßig.

Ein Nachteil soll freilich auch nicht verschwiegen werden:

- Dadurch, dass der Text erst von \LaTeX nach PDF gewandelt wird, unterscheidet sich der Arbeitsablauf von \LaTeX stark von den üblichen Textverarbeitungen bzw. DTP-Programmen. Das erfordert ein Umdenken und eine gewisse Einarbeitung.

1.2.4 Der Arbeitsablauf

Der typische Ablauf beim Arbeiten mit \LaTeX ist:

1. Ein Eingabefile schreiben, das den Text und die \LaTeX -Befehle enthält.
2. Dieses File mit \LaTeX bearbeiten; dabei wird eine Datei erzeugt, die den gesetzten Text in einem geräteunabhängigen Format (DVI, PDF oder auch PostScript) enthält.
3. Einen „Probeausdruck“ davon auf dem Bildschirm anzeigen (Preview).
4. Wenn nötig, die Eingabe korrigieren und zurück zu Schritt 2.
5. Die Ausgabedatei drucken.

Zeitgemäße Betriebssysteme machen es möglich, dass der Texteditor und das Preview-Programm gleichzeitig in verschiedenen Fenstern „geöffnet“ sind; beim Durchlaufen des obigen Zyklus brauchen sie also nicht immer wieder von neuem gestartet werden. Nur die wiederholte \LaTeX -Bearbeitung des Textes muss noch von Hand angestoßen werden und läuft ebenfalls in einem eigenen Fenster ab.

Wenn der Texteditor keine Schnittstelle anbietet, um \LaTeX direkt aus einem Menüpunkt heraus aufzurufen, dann ist der übliche Weg über die Kommandozeile bzw. Eingabeaufforderung. Dort wird dann das Kommando `pdflatex` aufgerufen und als Parameter wird der Name der Datei angegeben, unter der das Dokument auf der Festplatte gespeichert ist.

```
pdflatex masterarbeit.tex
```

Das Ergebnis des Aufrufs ist eine PDF-Datei, die wie die Eingabedatei heißt, nur mit der Endung `.pdf`. \LaTeX gibt einige Meldungen auf der Konsole aus, die beispielsweise Auskunft über die Anzahl der Seiten des Dokuments geben.

```
Arbeitsablauf von \LaTeX stark von den üblichen Textverarbeitungen bzw. UIP-Programmen. Da  
erfordert ein Umdenken und eine gewisse Einarbeitung.  
136 \end{itemize}  
137  
138 \subsubsection{Der Arbeitsablauf}  
139 Der typische Ablauf beim Arbeiten mit \LaTeX ist:  
140 \begin{enumerate}  
141 \item Ein Eingabefile schreiben, das den Text und die \LaTeX-Befehle  
142 enthält.  
143 \item Dieses File mit \LaTeX bearbeiten; dabei wird eine Datei  
144 erzeugt, die den gesetzten Text in einem geräteunabhängigen Format  
145 (\textsc{dvi}, \textsc{pdf} oder auch PostScript) enthält.  
146 \item Einen "Probeausdruck" davon auf dem Bildschirm anzeigen (Preview).  
147 \item Wenn nötig, die Eingabe korrigieren und zurück zu Schritt-2.  
148 \item Die Ausgabedatei drucken.  
149 \end{enumerate}  
150 Zeitgemäße Betriebssysteme machen es möglich, dass der Texteditor  
151 und das Preview-Programm gleichzeitig in verschiedenen Fenstern  
152 "geöffnet" sind; beim Durchlaufen des obigen Zyklus brauchen sie  
153 also nicht immer wieder von neuem gestartet werden. Nur die  
154 wiederholte \LaTeX-Bearbeitung des Textes muss noch von Hand  
155 angestoßen werden und läuft ebenfalls in einem eigenen Fenster ab.  
156
```

SyncTeX ist derzeit (Stand: 2013) in allen modernen  Engines eingebaut. Weitere (technische) Informationen finden sich unter den folgenden Links:

- <http://itexmac.sourceforge.net/SyncTeX.html>
- <http://river-valley.tv/direct-and-reverse-synchronization-with-synctex/>

69 Was ist `-shell-escape` und wie aktiviere ich das im Editor?

Manchmal möchte man aus dem TeX-Dokument heraus externe Programme ausführen. Beispiele:

- Automatische Umwandlung von eps-Grafiken zu pdf bei Verwendung von pdfLaTeX, unter Verwendung des Pakets

epstopdf

- Auslagerung der Erzeugung von Bildern (Externalisieren), z.B. von TikZ-Bildern
- Verwendung des Pakets `minted` für Syntax-Hervorhebung mittels Pygments

So etwas erfordert den Aufruf von LaTeX mit Option `--shell-escape`. Dadurch wird erlaubt, während des Übersetzend Aufrufe im Terminal (egal welcher Art) zu starten. Oft sind das weitere Aufrufe von LaTeX, ghostscript oder imagemagick. Da grundsätzlich auch gefährliche Aufrufe gemacht werden können, muss der Schalter `-shell-escape` explizit gesetzt werden. Am sichersten geht dies beim Aufruf [im Terminal](#).

Will man diesen Schalter trotzdem in seinem Editor dauerhaft setzen, wie kann man das tun?

Hier ist es für mehrere Editoren beschrieben:

- [TeXmaker](#) und [TeXstudio](#)
- [TeXstudio](#)
- [TeXnicCenter](#)

- `'arara'` (kein Editor, trotzdem gut)

Von Johannes auf [TeXwelt.de](#) geschrieben: [Wie aktiviere ich -shell-escape in meinem Editor?](#)

70 *Wie kann ich einen Text klickbar machen und ihn zu einem Link auf einen anderen Text machen?*

In beliebigen Dokumenten, von einfachen `article`-Dokumenten bis zu Präsentationen mit `beamer`, kann man Text „aktiv“ machen, anklickbar. Wenn man darauf klickt, dann soll zu einer anderen Stelle im PDF-Dokument gesprungen werden, die man selbst bestimmen kann.

Das kann man mit dem Paket `hyperref` tun.

Schon das Laden des `hyperref`-Pakets macht bereits automatisch Querverweise und Verzeichnis-Einträge zu anklickbaren Hyperlinks.

Befehle für den Fall, selbst Hyperlinks auf und zu Texten zu setzen:

- `\hypertarget{Name}{Text}` erzeugt ein mit benanntes Ziel, für den ein Text ins Dokument geschrieben wird
- `\hyperlink{Name}{Text}` schreibt einen Text und verlinkt ihn mit dem Ziel dieses Namens
- `\phantomsection` setzt einen Anker mit automatisch gewähltem Namen, ähnlich `\hypertarget{}{}`
- `\hyperref[name]{text}` macht den Text zu einem Link auf die Stelle, wo `\ref{name}` hinzeigen würde

Hier ist ein Komplettbeispiel zum Ausprobieren der Wirkung der Befehle:

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage{blindtext}
```

```
\usepackage{hyperref}
\begin{document}
\blindtext
\phantomsection% Setzt einen Anker
\label{link1}
```

Erstes Ziel

```
\blindtext[4]
\hyperref[link1]{Klick mich}
\blindtext[2]

\hypertarget{link2}{Zweites Ziel}
```

```
\blindtext[2]
\hyperlink{link2}{Klich mich auch}
\end{document}
```

Mehr kann man im [hyperref-Handbuch](#) und im [README](#) lesen.

Quelle: Student auf [TeXwelt.de](#): [Wie kann ich einen Text klickbar machen und ihn zu einem Link auf einen anderen Text machen?](#).

71 Was sollen die Kommentar-Zeichen am Ende von Zeilen?

In vielen Code-Beispielen und Paketen sieht man das Kommentar-Zeichen am Ende von Zeilen. Wieso? Welchen Zweck hat das?

Kurz und knapp: so verhindert man, dass ein Zeilenende wie ein Leerzeichen wirkt und ungewollt Zwischenraum eingefügt wird. Häufig in Makro-Definitionen sinnvoll.

Wenn TeX das Ende einer Zeile sieht, wandelt es das in ein Leer-

zeichen um, exakt äquivalent, als wenn man ein Leerzeichen eingegeben hätte:

```
Text
Text
```

ist genau gleich wie

```
Text Text
```

Genaugenommen wird das Zeichen eingefügt, das der Nummer des Registers `\endlinechar` entspricht (solange der Wert zwischen einschließlich 0 und 255 liegt). In der Voreinstellung ist das 13 und entspricht der Eingabe der Enter-Taste (*carriage return*, $\text{\^{\^}M}$ in TeX-Schreibweise mit Kategorie-Code 5=Enter-Taste). Folgt dem Ende einer Zeile ein weiteres Ende einer Zeile, wird stattdessen `\par` eingefügt.

In Makro-Definitionen gilt das genauso wie im laufenden Text. Die Definition

```
\newcommand*\foo{
foo
bar
}
```

ist gleich der Definition

```
\newcommand*\foo{ foo bar }
```

wie folgendes Beispiel demonstriert:

```
\documentclass{article}
\newcommand*\foo{
foo
bar
}
\begin{document}
```

```
bla\foo{}bar  
\end{document}
```

bla foo bar bar

Da Leerzeichen am Anfang einer Zeile verschluckt werden, nicht aber an anderen Stellen, sollte man Leerzeichen, die durch das Ende von Zeilen verursacht werden, verstecken, damit sie nicht unbeabsichtigt an Stellen auftauchen, an denen man sie nicht möchte:

```
\newcommand* \foo{%  
  foo  
  bar%  
}
```

Diese Definition ist jetzt gleichwertig zu

```
\newcommand* \foo{foo bar}
```

wie dieses Beispiel demonstriert:

```
\documentclass{article}  
\newcommand* \foo{%  
  foo  
  bar%  
}  
\begin{document}  
bla\foo{}bar  
\end{document}
```

blafoo barbar

Jetzt sollte man aber auch nicht den Schluss ziehen, dass man einfach immer an das Ende einer Zeile ein % setzen sollte. Aufpassen

muss man zum Beispiel nach Zuweisungen oder Primitiven, die Zahlen nach sich erwarten, wie `\ifnum` oder `\count\foo=...`. Diese schauen nämlich so lange weiter nach Zahlen, bis sie entweder ein nichtexpandierbares Token finden, das keine Zahl ist, oder bis sie auf ein Leerzeichen finden, das sie dann entfernen. Mit folgender Definition

```
\newcommand*\test[1]{%
  \ifnum1=#1%
    1%
  \else
    keine 1%
  \fi
}
```

liefert

```
\test{1}
```

vielleicht unerwarteterweise »keine 1«! Das liegt daran, dass die Definition gleichbedeutend ist mit

```
\newcommand*\test[1]{\ifnum1=#1\else keine 1\fi}
```

und tatsächlich 1 mit 1 1 verglichen wurde. Übrigens: nach Anweisung wie im Beispiel `\else` und `\fi` muss man kein Leerzeichen verstecken, da TeX Leerzeichen nach Anweisungen ignoriert. Eine bessere Definition des obigen wäre z.B.:

```
\newcommand*\test[1]{%
  \ifnum1=#1 % Leerzeichen nach `#1'
    1%
  \else
    keine 1%
  \fi
}
```

Ähnlich vorsichtig sollte man nach Zuweisungen sein, die Skips (Längen, die dehn- und stauchbare Anteile enthalten können) zuweisen und nach plus und minus suchen.

Von Clemens auf [TeXwelt.de](http://www.TeXwelt.de): [Was sollen die Kommentar-Zeichen am Ende der Zeilen?](#)

72 *Wie kann man Ligaturen verhindern?*

Eine Ligatur ist die Verschmelzung zweier oder mehrerer Buchstaben zu einem Symbol, um aus ästhetischen Gesichtspunkten optische Lücken zu vermeiden, insbesondere bei Buchstabenfolgen mit Oberlängen, wie fi, ff, fl, und ffl. [Wikipedia: Typografische Ligatur](#))

Bei zusammengesetzten Wörtern möchte man diese Verschmelzung mitunter vermeiden, damit in diesem Falle das Auge die Wortteile besser als separat erfasst.

Beispiel:

```
\documentclass{article}
\begin{document}
Auflaufform
\end{document}
```

Auflaufform

Es gibt mehrere Möglichkeiten, Ligaturen aufzubrechen.

72.1 Lokal

Man fügt einfach `\kern0pt` zwischen den zu trennenden Buchstaben ein. Im Beispiel

```
\documentclass{article}
\begin{document}
Auf\kern0pt lauf\kern0pt form
\end{document}
```

Es ist ebenfalls möglich die Ligatur durch eine Kursivkorrektur `\/` aufzubrechen. Da diese jedoch zusätzliches „explizites“ Kerning einfügt ist diese Methode nicht zu empfehlen. Siehe dazu auch TeXbook S. 306, Antwort zu Aufgabe 5.1:

„In fact, the latter idea—to insert an italic correction—*is preferable because TEX will reinsert the ff ligature by itself after hyphenating she lf {} fu l.* (Appendix H points out that ligatures are put into a hyphenated word that contains no „explicit kerns“, and an italic correction is an explicit kern.) But the italic correction may be too much (especially in an italic font); *she lf {\kern0pt} fu l* is often best.“

72.2 Global

Unter Zuhilfenahme des `microtype` Pakets kann man Ligaturen auch im kompletten Dokument deaktivieren.

```
\documentclass{article}
\usepackage{microtype}
\DisableLigatures{}
\begin{document}
Auf laufform
\end{document}
```

72.3 Automatisch mit LuaTeX und selnolig

Wenn man LuaLaTeX verwendet, hat man das ziemlich neue (und ziemlich coole) `selnolig`-Paket von Mico Loretan zur Verfügung. Für das gegebene Beispiel reicht es schon, es mit der Option `ngerman` zu laden:

```
\documentclass{article}
\usepackage{fontspec}
\usepackage[ngerman]{selnolig}

\begin{document}

Auflaufform beflügelt Koffer

\end{document}
```

Auflaufform beflügelt Koffer

Von Henri und Clemens auf [TeXwelt.de](http://www.texwelt.de): [Wie kann ich Ligaturen verhindern?](#)

73 *Wie kann ich LaTeX an die deutsche Rechtschreibung anpassen?*

Die Regeln für die deutsche Rechtschreibung (Silbentrennung) aktiviert man, in dem man das Paket `babel` mit der Option `german` bzw. `ngerman` (für die neue Rechtschreibung) lädt:

```
\documentclass{article}
\usepackage[ngerman]{babel}
...
```

oder äquivalent dazu:

```
\documentclass[ngerman]{article}
\usepackage{babel}
```

Im zweiten Fall gilt die Option »ngerman« für alle Pakete, die geladen werden.

74 Warum trennt LaTeX Worte mit Umlauten nicht oder nicht korrekt?

Vermutlich fehlt das Paket fontenc mit der Option T1:

```
\usepackage[T1]{fontenc}
```

in der Präambel.

Das Paket fontenc schaltet auf eine andere Fontkodierung um. Die Schriftart, die TeX (und somit LaTeX) normalerweise nutzt, liegt in einer Kodierung vor, die Umlaute nicht enthält. Ein Umlaut wird durch den zugehörigen Vokal und zwei Punkten erzeugt, die darüber gesetzt werden. Damit wird aber verhindert, dass TeX hinter dieser Stelle noch weitere Stellen für Silbentrennung findet.

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage[utf8]{inputenc}
\begin{document}
\showhyphens{Augenhöhe}
\showhyphens{Mädchen}
\end{document}
```

ergibt

```
[ ] \OT1/cmr/m/n/10 Au-genh[ ]ohe
[ ] \OT1/cmr/m/n/10 M[ ]adchen
```

Hier findet TeX wenigstens noch eine Trennung im Wort »Augenhöhe«. Aktiviert man das Paket fontenc,

```
\documentclass[ngerman]{article}
\usepackage{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\begin{document}
\showhyphens{Augenhöhe}
\showhyphens{Mädchen}
\end{document}
```

so ist alles in Ordnung:

```
[ ] \T1/cmr/m/n/10 Au-gen-h?-he
[ ] \T1/cmr/m/n/10 M?d-chen
```

(die ? brauchen uns nicht zu irritieren).

75 Wie kann ich Trennungsausnahmen mit Umlauten in \hyphenation verwenden?

Auch hier gilt, daß man in `\hyphenation` in den Trennungsausnahmen nur Zeichen verwenden kann, für die auch Lettern in der momentan benutzten Schrift existieren. Da mit CM-Schriften die Umlaute aus zwei Lettern zusammengesetzt werden müssen, kann man für Wörter mit Umlauten keine Trennungsausnahmen angeben. Erst bei Verwendung von T1-kodierten Schriften, wie der EC-Schriftfamilie, oder auch von Schriften in einer entsprechenden anderen Kodierung, die eigene Lettern für Umlaute enthalten, kann man Trennungsausnahmen angeben.

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}
```

```
% \hyphenation{Ver-stär-ker-aus-gang}
\begin{document}
\showhyphens{Verstärkerausgang}
\end{document}
```

ergibt (ohne Zeile mit `\hyphenation{...}`)

```
[ ] \T1/cmr/m/n/10 Ver-st?r-ke-r-aus-gang
```

und mit `\hyphenation{...}`

```
[ ] \T1/cmr/m/n/10 Ver-st?r-ker-aus-gang
```

(auch hier wird das ä durch ein ? ersetzt bei der Logausgabe)

76 Wieso werden Texte in »typewriter«-Schriften nicht getrennt? Gibt es eine Möglichkeit, dass doch getrennt werden kann?

TeX trennt nur, wenn die aktuell verwendete Schrift einen Divis (Trennungsstrich) besitzt. Da zu ist jeder Schrift ein Divis durch den Parameter `\hyphenchar` zugeordnet, der bei den meisten Schriften auf den Wert `-`, also auf das Zeichen `-`, gesetzt ist. Bei den »typewriter«-Schriften ist dieser Wert aber auf `-1` gesetzt, d. h. auf ein nicht vorhandenes Zeichen. Damit ist die Trennung für diese Schriften unterdrückt, was in den meisten Fällen auch sinnvoll ist.

Will man dennoch, dass TeX einen in »typewriter« gesetzten Textteil automatisch trennt, so muss man den `\hyphenchar` für diese Schriften auf `-` setzen. Dies geht in LaTeX am einfachsten, wenn man die Anweisung `\ttfamily` undefiniert:

```
\newcommand{\origttfamily}{}% sollte noch nicht
  definiert sein!
\let\origttfamily=\ttfamily % alte Definition von \
  ttfamily sichern
\renewcommand{\ttfamily}{\origttfamily \hyphenchar\font
  =`\-}
```

Wichtiger Hinweis: TeX kann eine Schrift unter unterschiedlichen Namen laden, wobei TeX alle diese Schriften nur ein einziges Mal lädt und für die anderen Schriften nur »Aliases« anlegt. Daher gibt es `\hyphenchar` nur einmal pro »echten« Font und `\hyphenchar` ist – wie auch `\fontdimen` – ein besonderer Parameter: Alle Zuweisungen an `\hyphenchar` werden immer global ausgeführt. Außerdem sieht sich TeX diesen Parameter erst am Absatzende beim Umbruch dieses Absatzes an, so dass es mit obiger Änderung nicht möglich ist, die Trennung für einen in »typewriter« gesetzten Textteil zu ermöglichen, und gleichzeitig für einen anderen im selben Absatz zu verbieten.

Eine weitere Methode, Trennungen für »typewriter«-Schriften zu erlauben, bietet das Paket `hyphenat`:

```
\usepackage[htt]{hyphenat}
```

Darüber hinaus kann es auch benutzt werden, um Trennung selektiv zu erlauben oder zu verbieten.

77 Warum trennt LaTeX Wörter mit Bindestrich nicht bzw. nur am Bindestrich, auch wenn die Trennausnahmeliste diese enthält?

TeX fügt nach dem aktuellen Trennzeichen implizit (fast) immer einen sogenannten `\discretionary` ein, der wie - eine explizite Trennstelle angibt.

Da sowohl - als auch `\discretionary` dazu führen, dass die Trennung in den übrigen Wortteilen unterdrückt wird, werden solche Wörter außer am Trennstrich nicht mehr getrennt. Dieses eigenwillige Verhalten von TeX ist für Sprachen, in denen wenige Bindestriche verwendet werden, günstig und sinnvoll, da zusätzliche Trennungen in solchen Wörtern meist die Lesbarkeit erschweren. Um Wörter mit Bindestrichen dennoch zu trennen, existiert folgende Möglichkeit:

Man gibt im Dokumenttext bei diesen Wörtern, insbesondere wo schlecht getrennt werden kann, mit - oder " - weitere mögliche Trennstellen vor oder man verwendet " = anstelle des Bindestrichs. Näheres ist hierzu der Dokumentation des Paketes »babel« zu entnehmen.

78 Warum soll ich `[...]` anstelle von `$$...$$` benutzen?

`$$...$$` ist ein reiner plain-TeX-Befehl. Bei der Verwendung in LaTeX kann es passieren, dass Abstände vor und nach der abgesetzten Formel nicht richtig gesetzt werden. Daher ist in LaTeX immer die Variante `[...]` für abgesetzte Formeln zu verwenden.

79 Gibt es einen Unterschied zwischen $\$...\$$ und $\{(...)\}$?

Im Gegensatz zu dem abgesetzten Mathematikmodus ist die Empfehlung hier, weiterhin die $\$$ -Zeichen zu benutzen.

- $\$...\$$ benötigt kein `\protect`, kann also in Überschriften & Co eingesetzt werden
- $\$...\$$ hat nur einen theoretischen Nachteil gegenüber $(...)$. Ein Texteditor könnte leichter *Anfang* und *Ende* der Mathematikumgebung erkennen.
- $(...)$ kann nicht ohne Weiteres in Überschriften eingesetzt werden.

80 Wie kann ich einen aufrechten Differentialoperator erzeugen?

```
\newcommand*\diff{\mathop{\!}\mathrm{d}}
```

Quelle: H. Voß: Mathematiksatz. 2. Aufl. S. 76

81 Warum fügt TeX in einer mathematischen Formel hinter einem Dezimalkomma einen kleinen Zwischenraum ein?

Die meisten TeX-Makropakete und -Formate gehen davon aus, dass man einen Dezimalpunkt statt eines -kommata verwendet. Für

den Dezimalpunkt, der in einer mathematischen Formel die Bedeutung eines »Ordinary«-Symbols hat, wird daher zwischen Ziffern kein zusätzlicher Zwischenraum eingefügt. Das Komma hat in Formeln die Bedeutung eines »Punctuation«-Symbols, man kann es daher als Trennzeichen in einer Aufzählung verwenden, da TeX mit einem kleinen Zwischenraum dafür sorgt, dass Formeln wie »f(x, y)« gut aussehen.

Das Problem des falschen Abstandes für ein Dezimalkomma kann man auf verschiedene Arten beheben. Für einzelne Fälle mag es reichen, das Komma einfach mit {...} zu klammern, da geklammerte Formelteile dieselben Abstände wie ein »Ordinary«-Symbol bekommen:

```
$\pi = 3{,}1415\ldots$
```

Eine andere Methode erklärt das Komma zu einem »Ordinary«-Symbol statt eines Trenzeichens:

```
%%% alte Definition:
%% \mathcode\,="613B
%%% neue Definition:
\mathcode\,="013B
```

und wenn man statt der \mathcode-Zuweisung die Anweisung \DeclareMathSymbol in LaTeX verwendet:

```
%%% alte Definition:
%% \DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
%%% neue Definition:
\DeclareMathSymbol{,}{\mathord}{letters}{"3B}
```

Der Preis hierfür ist natürlich, dass in »f(x, y)« der zusätzliche Zwischenraum fehlt. Aber auch das lässt sich in den Griff bekommen: Das Paket `icomma` ändert die Definition des Kommazeichen

so, dass es im mathematischen Modus zwischen »3,1415« und »f(x, y)« unterscheiden kann: Steht hinter dem Komma ein Leerzeichen, ist das Trennzeichen mit etwas Abstand gemeint, ansonsten das Dezimalkomma. Im Unterschied zu ähnlichen Lösungen funktioniert diese Paket mit beliebigen Mathematik-Fonts.

Mit dem Paket `ziffer` ist das Leerzeichen nach dem Komma, um Mengen oder Argumente zu kennzeichnen, bis auf einige Ausnahmen nicht notwendig, so dass bestehende Texte nicht verändert werden müssen.

Alternativ kann man den Befehl `\num` des Pakets `siunitx` nutzen:

```
\documentclass[paper=a4,ngerman]{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{babel}
\usepackage{siunitx}
\sisetup{
  output-decimal-marker = {,}
}

\begin{document}
\[ 1.23 \times \num{1.23} \times 1,23 \]
\end{document}
```

$$1.23 \times 1,23 \times 1,23$$

82 *Wie kann man Symbole in mathematischen Formeln fett drucken?*

Es gibt prinzipiell drei Arten, fette Symbole zu erzeugen.

1. `\boldsymbol` mit dem `amsmath` Paket
2. `\bm` mit dem `bm` Paket
3. `\pmb` auch mit dem `amsmath` Paket

Während `\boldsymbol` und `\bm` tatsächlich die fetten Symbole aus der aktuellen Mathematik-Schriftart auswählen erzeugt `\pmb` das fette Symbol durch Überlagerung mehrerer normalgewichtiger Symbol, was nicht besonders schön aussieht, aber die einzige Lösung ist, wenn kein fettes Symbol verfügbar ist.

```
\documentclass{article}
\pagestyle{empty}% zum freistellen
\usepackage{amsmath}
\let\amsboldsymbol\boldsymbol
\usepackage{bm}% ändert \boldsymbol
\let\boldsymbol\amsboldsymbol
\begin{document}
\obeylines
\verb|\boldsymbol{a}| :  $\boldsymbol{a}$ 
\verb|\pmb{a}| :  $\pmb{a}$ 
\verb|\bm{a}| :  $\bm{a}$ 
\end{document}
```

`\boldsymbol{a}`: ***a***
`\pmb{a}`: ***a***
`\bm{a}`: ***a***

Vergleich von `\boldsymbol` und `\bm`

- `\bm` erhält die Kursivkorrektur für folgende hoch- oder tiefgestellte Zeichen
- `\bm` erhält korrekten Zwischenraum im Formelkontext, `\boldsymbol` manchmal nicht - letzteres ist ok bei Relationen und Operatoren, jedoch nicht bei Klammer-Abständen
- `\bm` benötigt manchmal zusätzliche geschweifte Klammern zum Funktionieren, z.B. `\bm{\{\dot{s}\}}` sowie bei doppelten Akzenten: `\bm{\{\hat{\dot{\phi}}\}}`
- Schlussendlich funktionieren verschachtelte Mathematik-Alphabete besser mit `\boldsymbol` als mit `\bm`.

Diese Erklärung von Henri finden man ausführlicher mit Beispielen auf TeXwelt.de: [☞ Wie druckt man am besten Symbole fett in mathematischen Formeln?](#)

Ein Hinweis von Felix auf derselben Seite:

Fette Symbole (und allgemein spezifische Formatierungen) sollten *stets in der Präambel definiert werden*, und nie im Text solche Befehle wie `\mathbf` zu verwendet werden. Wenn man später

entscheidet, dass die im Text verstreuten fetten Symbole doch nicht optimal aussehen, kann man es so ganz einfach in der Präambel ändern.

Dafür empfiehlt sich das `bm`-Paket und der Befehl

```
\DeclareBoldMathCommand[<Mathe-Version>]{<Kommando>}{<mathematischer Ausdruck>}
```

`<Version>` ist optional und per default `bold`. Man könnte auch `heavy` angeben, wenn man eine besonders fette Schrift zur Verfügung hat.

Ein Beispiel:

```
\documentclass{article}
\usepackage{bm}
\DeclareBoldMathCommand\balpha{\alpha}
\begin{document}
 $\sin(\balpha) = 1$ 
\end{document}
```

83 Was ist LuaTeX und kann ich es anstelle von LaTeX benutzen?

LuaTeX ist eine relativ neue  TeX-Implementierung (engine) – quasi der Nachfolger von pdfTeX, das heutzutage meist genutzte TeX-Programm. Wenn man sagte, „Ich schreibe meine Texte mit LaTeX“, meinte man meist, dass man seinen Text mit einem Editor schrieb, dabei die Makrosprache LaTeX verwendete, um Gliederung, Zitate, Fußnoten etc. auszuzeichnen, und anschließend mit pdfTeX als Engine aus diesem LaTeX-Quelltext das Dokument zu

einer PDF-Datei kompilierte: als letztem Schritt des Satzvorgangs. Anstelle von pdfTeX kann man dafür auch LuaTeX verwenden.

Die zweite Frage lässt sich nicht beantworten, weil sie ungenau gestellt ist. Die Ungenauigkeit ist darauf zurückzuführen, dass aktuelle TeX-Distributionen oder Editoren den Aufruf von pdfTeX zum Setzen eines LaTeX-Quelldokuments mit „pdflatex“ und mit „latex“ erlauben. Will man als Satzengine LuaTeX verwenden, wäre der richtige Befehl „lualatex“. Den Editor könnte man so konfigurieren, dass beim Aufruf von „latex“ auch „lualatex“ verwendet wird, d.h. die Engine LuaTeX für mit LaTeX geschriebene Quelldokumente.

84 Was muss ich beim Umstieg auf LuaTeX beachten?

Wenn man ein klassisches LaTeX-Dokument hat, das mit PDFTeX erzeugt wird, kann man meist ohne großartige Änderungen einfach LuaLaTeX benutzen.

Als erstes merkt man meist den Unterschied in der Fähigkeit von LuaTeX, UTF-8-kodierte Dokumente direkt zu verarbeiten, ohne das Paket `inputenc` zu benutzen. Das muss man also herausnehmen. Aus dem einfachen Dokument

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
Schönen Gruß aus Österreich!
\end{document}
```

muss man für LuaLaTeX den Aufruf des Pakets `inputenc` entfernen, sonst gibt es folgende Fehlermeldung:

```
! Package inputenc Error: Keyboard character used is
  undefined
(inputenc)          in inputencoding `utf8'.
```

```
See the inputenc package documentation for explanation.
Type H <return> for immediate help.
...
```

```
1.6 Schö
      nen Gruß aus Österreich!
```

Möchte man weiterhin seine Dokumente mit PDFLaTeX als auch mit LuaLaTeX verarbeiten können, empfiehlt sich das Paket `luainputenc`, das auch mit PDFLaTeX funktioniert:

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{luainputenc}
\begin{document}
Schönen Gruß aus Österreich!
\end{document}
```

Dieses Dokument funktioniert mit PDFLaTeX und LuaTeX.

85 Erste Schritte mit LuaTeX

Das folgende Beispiel zeigt die einfache Möglichkeit, einen Lua-Befehl innerhalb von LaTeX auszuführen. Dasselbe funktioniert auch mit Plain TeX.

```
\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}
```

```
\begin{document}
Die Kreiszahl  $\pi$  hat den Wert \directlua{tex.print(
  math.pi)}.
\end{document}
```

Die Kreiszahl π hat den Wert 3.1415926535898.

Hier ergeben sich aber direkt einige Probleme. Man kann beispielsweise die üblichen Lua-Kommentarzeichen (`--`) nicht mehr benutzen, weil alles, was innerhalb von `\directlua{...}` steht, in einer Zeile interpretiert wird. So hat folgender Code einen Syntaxfehler:

```
\directlua{
  for x=1,10 do -- Schleifenanfang
    tex.print(x)
  end
}
```

weil der Lua-Interpreter folgendes liest:

```
\directlua{ for x=1,10 do -- Schleifenanfang tex.print(
  x) end }
```

und den Rest hinter dem Schlüsselwort `do` ignoriert. Daher empfiehlt es sich, Lua-Code soweit wie möglich in eine eigene Datei zu schreiben und in dem `\directlua`-Befehl nur den Befehl zum Laden der Datei anzugeben.

Sollte das nicht möglich sein, kann man auf das Paket `luacode` mit der gleichnamigen Umgebung ausweichen:

```

\documentclass{article}
\usepackage[ngerman]{babel}
\usepackage[T1]{fontenc}
\usepackage{luacode}
\begin{document}
\begin{luacode*}
  for x=1,10 do -- Schleifenanfang
    tex.print(x)
  end
\end{luacode*}
\end{document}

```

Trotzdem gilt hier die dringende Empfehlung, möglichst wenig Code in den LaTeX-Quellen zu verwenden und für den Rest eine eigene Datei zu nutzen. Damit wird man auch aussagekräftigere Fehlermeldungen mit Zeilennummer erhalten.

86 Wie binde ich OpenType-Schriften ein?

86.1 XeLaTeX und LuaLaTeX

Für LaTeX in Verbindung mit XeTeX und LuaTeX ist die Sache ganz einfach. Hier gibt es das Paket `fontspec`, mit dem man ganz einfach beliebige OpenType-Schriftarten in das Dokument einbinden kann:

```

\documentclass{article}
\usepackage{fontspec}
\setmainfont{Linux Libertine O}
\begin{document}
Dieser Text ist mit Linux Libertine gesetzt.
\end{document}

```

In der [Dokumentation](#) zu fontspec wird das ausführlich erläutert. Interessant ist, dass hier keine FD-Dateien für LaTeX benötigt werden. fontspec sucht selbständig nach den passenden Fonts aus der Schriftfamilie (fett, kursiv oder fettkursiv) und erzeugt auch die benötigten Fonttabellen für LaTeX (NFSS).

87 ! LaTeX Error: There's no line here to end.

Wenn dieser Fehler auftritt, sollte man sich genau anschauen, wo und warum man `\` verwendet. Die Problemzeile ist in der Fehlermeldung angegeben, jedoch kann sie sich auch in einem Makro verbergen.

87.1 Problembeispiel:

Hier wird der Fehler für Zeile 5 gemeldet:

```
\documentclass{article}
\begin{document}
\raggedright
Zentriert\
\
\end{document}
```

87.2 Warum?

Die im Titel genannte Fehlermeldung tritt immer dann auf, wenn man im linksbündigen, rechtsbündigen oder zentrierten Kontext `\` missbraucht, um eine leere Zeile einzufügen. Im Wir-

kungsbereich von `\centering`, `\raggedright` und `\raggedleft` ist `\\` nämlich so definiert, dass es zunächst prüft, ob sich TeX im horizontalen Modus, also innerhalb eines Absatzes und damit innerhalb einer Zeile befindet. Ist das nicht der Fall, gibt es quasi keine aktuelle Zeile und somit kann die aktuelle Zeile auch nicht beendet werden. Deshalb gibt LaTeX dann die Fehlermeldung aus. Das ist besonders dann wichtig, wenn man eher versehentlich noch keine Zeile begonnen hat

87.3 Was nun?

- `\\` im Fließtext nach Möglichkeit weglassen - oft meint man sowieso Absatzumbrüche, die man mit Leerzeile oder `\par` macht
- Nicht `\\` am Beginn von Textmakros verwenden
- Schon gar nicht mehrere `\\` nacheinander - wenn man Abstände erzeugen will, gibt es dafür Befehle oder das optionale Argument: z.B. überspringt `\\[\baselineskip]` genau eine Zeile

87.4 Weiterführendes

Saputello hat auf [TeXwelt.de](https://www.texwelt.de) weitere Phänomene und Hintergründe erklärt: [LaTeX Error: There's no line here to end.](#)

88 ! LaTeX Error: Option clash for package <name>.

Ein *option clash* geschieht, wenn ein Paket zweimal mit unterschiedlicher Optionenliste geladen wird. Oft geschieht das, weil ein Paket von einem anderen geladen wird, ohne dass man das weiß, so wie in diesem Beispiel - hier wird TikZ verwendet und zusätzlich xcolor mit der table-Option geladen:

```
\documentclass{article}
\usepackage{tikz}
\usepackage[table]{xcolor}
\begin{document}
text
\end{document}
```

Man erhält:

```
! LaTeX Error: Option clash for package xcolor.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...
1.4 \begin
      {document}
```

Zwei Fälle kann man hier unterscheiden:

Einmal wird das Paket *ohne* Optionen und einmal *mit* Optionen geladen. Hier genügt es, die Ladereihenfolge zu vertauschen, so dass der Aufruf *mit* Optionen zuerst erfolgt. Also statt

```
\documentclass{article}
\usepackage{xcolor}% vielleicht versteckt in einem
                    anderen Paket
\usepackage[table]{xcolor}
\begin{document}
```

Fehler

```
\end{document}
```

Die andere Reihenfolge:

```
\documentclass{article}
\usepackage[table]{xcolor}
\usepackage{xcolor}% vielleicht versteckt in einem
    anderen Paket
\begin{document}
Kein Fehler
```

```
\end{document}
```

Das ist daher auch in der Regel die erste Maßnahme, die man ausprobiert.

Zweite Möglichkeit: das Paket wird zweimal *mit* Optionen geladen, aber mit unterschiedlichen:

```
\documentclass{article}
\usepackage[svgnames]{xcolor}
\usepackage[table]{xcolor}% vielleicht versteckt in
    einem anderen Paket
\begin{document}
Fehler
```

```
\end{document}
```

In einem solchen Fall kann man anstatt das Paket ein zweites Mal zu Laden, die Optionen an das Paket weiterreichen. Dafür gibt es den Befehl `\PassOptionsToPackage{<Optionen>}{<Paket>}`:

```
\documentclass{article}
\PassOptionsToPackage{svgnames}{xcolor}
\usepackage[table]{xcolor}% vielleicht versteckt in
    einem anderen Paket
\begin{document}
```

```
Kein Fehler  
\end{document}
```

Diese Maßnahme hätte auch im ersten Fall funktioniert. `\PassOptionsToPackage` muss übrigens *vor dem ersten Laden* des betreffenden Pakets eingesetzt werden. Setzt man es daher vor `\documentclass`, dann ist man auf der sicheren Seite.

Oft hilft es auch, die zusätzliche Option *global* einzusetzen, das heißt, sie der Klasse mitzugeben:

```
\documentclass[svgnames]{article}  
\usepackage[table]{xcolor}% vielleicht versteckt in  
    einem anderen Paket  
\begin{document}  
Kein Fehler  
\end{document}
```

Diese letzte Maßnahme funktioniert in der Regel auch, wenn das betreffende Paket nicht von anderen Paket sondern von der Dokumentenklasse geladen wird. (So lädt die `beamer`-Klasse z.B. ebenfalls das `xcolor` Paket):

```
\documentclass[svgnames,table]{beamer}  
\begin{document}  
\begin{frame}  
\textcolor{Goldenrod}{Kein Fehler}  
  
\begin{tabular}{|l|c|}  
\rowcolor[gray]{.9}  
eins & zwei\\  
\rowcolor[gray]{.5}  
drei & vier  
\end{tabular}  
\end{frame}  
\end{document}
```

Wenn alle Stricke reißen, kann man das betreffende Paket auch *vor* der Klasse zu laden (mit `\RequirePackage` statt `\usepackage`), oder (wie oben schon vorgeschlagen) eben den Aufruf `\PassOptionsToPackage` vor `\documentclass` zu setzen. Nötig kann das z.B. sein, da der Einsatz als globale Option an *alle* Pakete weitergereicht wird, die die Option kennen, was manchmal unerwünscht sein kann. **Vorsicht: nicht jedes Paket kann erfolgreich vor der Klasse geladen werden! Diese Maßnahme sollte man nur in Ausnahmefällen verwenden, wenn man weiß, was man tut.**

Clemens hat das auf www.texwelt.de hier geschrieben: `LaTeX Error: Option clash for package` - wie beheben?

89 ! No room for a new \dimen

Ursprünglich hatte TeX „nur“ 256 Register-Plätze für `\dimens`. Heutzutage stehen mit e-TeX stehen zwar grundsätzlich 32768 zur Verfügung, es sind aber von vorherein nur die 256 aktiviert. Manchmal werden die verbraucht durch eine bestimmte Paket-Kombination, und wenn dann versucht wird, einen weiteren Platz zu belegen, dann kommt die besagte Fehlermeldung.

Man kann die Tatsache ausnutzen, dass e-TeX die Anzahl der verfügbaren Register erhöht. Dafür reicht es im Beispiel, das Paket `etex` zu laden, das die zusätzlichen Plätze aktiviert.

Dies funktioniert zum Beispiel:

```
\documentclass{beamer}
\usepackage{etex}
\usepackage{multicol}
```

```
\usepackage{pstricks}
\usepackage{pst-circ}
\usepackage{pstricks-add}
\begin{document}
\begin{frame}
  Text
\end{frame}
\end{document}
```

Ohne `etex` tritt jedoch bereits in diesem kurzen Beispiel der genannte Fehler auf.

Clemens erklärte das auf www.texwelt.de hier: [☞](#)Wie behebe ich den Fehler: ! No room for a new \dimen?

90 Was ist DANTE?

Die *Deutschsprachige Anwendervereinigung TeX e. V.*, kurz DANTE, hat es sich zum Ziel gesetzt, die Nutzung von TeX durch Betreuung, Beratung und Förderung von Entwicklungsprojekten zu unterstützen. Nähere Informationen finden sich auf der Homepage [☞www.dante.de](http://www.dante.de).

Der Verein wurde am 14. April 1989 in Heidelberg gegründet, ist in das Vereinsregister eingetragen (e.V.) und vom Finanzamt als gemeinnützig anerkannt worden. Zur Zeit (2013) hat der Verein etwas mehr als 2000 Mitglieder.

Zu erreichen ist DANTE e.V. wie folgt:

Post DANTE, Deutschsprachige Anwendervereinigung
TEX e.V.
Postfach 10 18 40
69008 Heidelberg

Telefon (0 62 21) 2 97 66

Fax (0 62 21) 16 79 06

E-Mail [✉dante@dante.de](mailto:dante@dante.de)

Web [✉www.dante.de](http://www.dante.de)

90.1 Vereinszweck

Zweck des Vereins ist es, alle Interessenten in geeigneter Weise bei der Anwendung von TeX- Software zu unterstützen, insbesondere durch Erfahrungsaustausch mit den Anwendern. Ferner soll die weitere Entwicklung von TeX unterstützt werden. Hierzu gehören

- die Beratung und Zusammenarbeit mit Anwendern der TeX-Software,
- die Unterbreitung von Mitgliedervorschlägen für zukünftige Versionen und Komponenten der TeX-Software,
- die Zusammenarbeit mit anderen nationalen und internationalen Benutzerorganisationen,
- die Förderung von Literatur, die TeX betrifft,
- die Vertretung nationaler Belange in der TeX Users Group,
- die Organisation von Schulungen.

91 Warum soll ich Mitglied werden?

- DANTE fördert eine Reihe von Projekten im TeX-Umfeld. Beispielsweise wird die Weiterentwicklung freier Schriftarten (auch zusammen mit anderen TeX-Benutzergruppen) finanziert. DANTE übernimmt die Kosten für den deutschen CTAN Knoten, der TeX Software sammelt und so erst Distributionen wie TeXlive ermöglicht.
- Vereinszeitung Die TeXnische Komödie erscheint viermal im Jahr und enthält viele exzellente Artikel rund um TeX.
- Als Mitglied kann man günstiger an Tagungen teilnehmen und erhält immer aktuelle Informationen zu den Veranstaltungen und Stammtischen.
- Es gibt eine Reihe Vergünstigungen, beispielsweise bei den Büchern der DANTE-Edition und für die WinEdt Lizenz.
- Die aktuellen TeX Distributionen und CTAN Abzüge werden meist jährlich auf DVD an die Mitglieder verschickt.

92 Wie werde ich Mitglied?

Der Mitgliedsantrag kann einfach online auf dante.de oder schriftlich mit dem entsprechenden Formular gestellt werden. Dort sind auch die aktuellen Preise aufgeführt.

Interessant: es gibt eine Schnuppermitgliedschaft zu einem sehr günstigen Einstiegspreis, die alle Vorteile einer Vereinsmitgliedschaft für das laufende Jahr umfaßt (wie DTK, TeXlive-DV, Ermä-

ßigungen, ...) und automatisch am Ende desselben ausläuft (ohne Kündigung). Näheres hierzu auf der [Webseite](#) von DANTE.

93 Die TeXnische Komödie

- Vereinszeitschrift
- erscheint 4x/Jahr
- Nach einem Jahr als PDF frei verfügbar
- Weblinks

94 Tagungen und Veranstaltungen

Die TeX-Anwendervereinigungen führen regelmäßig Tagungen, Tutorien, Mitgliederversammlungen etc. durch:

94.1 DANTE e.V.

Jährliche Tagung, Mitgliederversammlung und Tutorien (3 bis 3 1/2 Tage) im Frühjahr; meist im Februar oder März. Jährliche Mitgliederversammlung mit Tutorien (1 bis 1 1/2 Tage) im Herbst; meist im September oder Oktober.

Link: <http://www.dante.de>

94.2 Europäische Anwendervereinigungen

Jährliche europäische Tagung (EuroTeX), meist im September. Entfällt im allgemeinen in den Jahren, in denen die TUG in Europa eine Tagung ausrichtet.

94.3 TUG

Jährliche weltweite Tagung; meist Ende Juli, Anfang August; im Wechsel an einem Ort der Westküste der USA, der Ostküste und im »Rest der Welt«.

Auch Personen, die noch nicht Mitglied bei einer Anwendervereinigung sind, können an den Tagungen teilnehmen. Tutorien, die DANTE e.V. vor oder nach einer Mitgliederversammlung durchführt, sind frei und können kostenlos besucht werden.

95 Buchliste

Die Buchliste befindet sich derzeit noch im »alten Wiki«:

☞ <http://projekte.dante.de/DanteFAQ/Literatur>

96 TeXwelt.de

Auf ☞ <http://texwelt.de> kann man Fragen zu TeX und LaTeX stellen, Antworten schreiben und kommentieren.

Für Fragen und Antworten kann man Stimmen abgeben. So kristallisieren sich die nützlichsten und schönsten Beiträge heraus. Sowohl Fragen als auch Antworten können editiert und verbessert werden.

Fragen können Themen zugeordnet werden, auch bekannt als „Tags“, das vereinfacht das spätere Browsen und Suchen.

Auf diese Weise entsteht quasi nebenbei beim Nutzer-Support ein Wissensarchiv, das nach Themen und Qualität sortiert ist.

- Was sich über TeXwelt zu wissen lohnt: ↗Antworten auf häufige Fragen
- Diese Information etwas ausführlicher: ↗Über TeXwelt

Einige sehr wertvolle Antworten auf häufige Fragen wurden von TeXwelt ausgewählt und hier eingearbeitet. Diese Seite wurde angelegt, damit man diese Auswahl gesammelt sieht, nämlich: ↗Links auf TeXwelt-Beiträge.