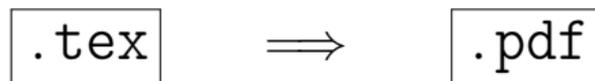


Dokumente erstellen mit T_EX/L^AT_EX

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n$$

- \LaTeX -Begleiter; Frank Mittelbach, Michel Goossens
- \LaTeX Der typographische Einstieg; Tobias Berndt
- DANTE e.V.: <http://www.dante.de/> und Bücher dort, insbesondere von Herbert Voß, z.B.
 \LaTeX Referenz; Herbert Voß
Tabellen mit \LaTeX ; Herbert Voß
- \LaTeX GE-PACKT; Karsten Günther
- \LaTeX kurz & gut; Matthias Kalle Dalheimer

T_EX ist ein Textsatzsystem, das ähnlich einem Compiler eine **Quelldatei** (mit der Endung `.tex`) einliest und daraus eine formatierte Dokumentdatei erstellt (i.d.R. eine `.pdf`-Datei → `pdflatex`).



```
\documentclass{scrartcl}
\usepackage[german,english]{babel}
\usepackage{graphicx}
\begin{document}
\section{Einleitung}
\TeX\ ist ein von Donald Knuth entwickeltes
Textsatzsystem. █
\end{document}
```

1 Einleitung

T_EX ist ein von Donald Knuth entwickeltes Textsatzsystem.

Zur Begriffsklärung

$\text{T}_{\text{E}}\text{X}$

das ursprüngliche, von Donald Knuth entwickelte Textsatzsystem, Seitenbeschreibungssprache (vgl. HTML)

\LaTeX

Makroerweiterung für $\text{T}_{\text{E}}\text{X}$ von Leslie Lamport, Dokumentklassen, Pakete

pdf \LaTeX

Programm für \LaTeX zur direkten Ausgabe von pdf, unterstützt erweiterte Mikrotypografie und jpeg/png/pdf-Grafikdateien

Anwendungen von L^AT_EX:

- Publikationen: Paper, Bücher, Masterarbeiten usw.
- Übungsblätter, Formulare u.v.m.
- Präsentationen: diese (Dokumentklasse `beamer`), Folien
- Briefe (auch Serienbriefe)
- Vektorgrafik mittels PSTricks
- Poster

L^AT_EX-Beispiel

```
\documentclass{scrartcl}

\begin{document}

Hallo, Welt! % Dokumenttext

\end{document}
```

Übersetzen mit Hilfe des **Shellbefehls** (Achtung: kein GUI-Programm!)

```
pdflatex hallowelt.tex
```

Ergebnis ist die Datei `hallowelt.pdf`.

`\` – Backslash leitet $\text{T}_{\text{E}}\text{X}$ -Befehle ein

`{ }` – geschweifte Klammern umschließen Pflichtargumente von $\text{T}_{\text{E}}\text{X}$ -Befehlen. Im Text können `{ }` mittels `\{` und `\}` erzeugt werden.

`\documentclass` – definiert den Dokumentstil, **stets als 1. Zeile!**, grundlegende Einstellungen für Seitenformat, Schriftgröße, Fonts, etc. (nur $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$), Beginn der Präambel

`scrartcl` – KOMA-Script-Klasse für kurze Dokumente in DIN A4

Präambel – der Bereich zw. `\documentclass{}` und `\begin{document}`, Setzen von dokumentweiten Einstellungen

`\begin{document}` – Ende der Präambel, Beginn des eigentlichen Dokuments

`\end{document}` – Ende des Dokuments, alles danach wird ignoriert

`%` – Prozentzeichen leitet Kommentar ein, falls `%` im Text verwendet werden soll: `\%` (Backslash direkt vor Prozentzeichen)

Falls `pdflatex` einen \LaTeX -Befehl nicht erkennt, etwa weil er falsch geschrieben wurde, stoppt die Übersetzung. Wurde z.B. `\nd{document}` statt `\end{document}` geschrieben:

```
\! Undefined control sequence.  
1.6 \nd  
      {document}  
?
```

In diesem Fall bewirkt `x`, Eingabetaste Abbruch der Übersetzung.

Tipp: `h`

Statt `x` kann man zunächst auch mit `h` eine kurze Hilfe zum Problem anfordern. Das ist leider nur manchmal wirklich hilfreich.

Hinweise zum Bearbeiten von `.tex`-Dateien mit dem Editor

- Fast jeder Editor erkennt $\text{T}_\text{E}\text{X}$ -Dateien anhand der Dateinamensendung `.tex` (z.B. `hallowelt.tex`) und hebt dann $\text{T}_\text{E}\text{X}$ -Befehle in der Ansicht optisch (z.B. durch andere Farbe/Font) hervor.
- Dafür sollte schon beim Starten des Editors der Dateiname **immer** als Argument angegeben werden:

```
emacs hallowelt.tex &
```

- Der Dateiname sollte **keine Leerzeichen** enthalten! (Warum?)

Aufgabe 3.1 Einfaches \LaTeX -Dokument

- 1 Legen Sie ein neues Verzeichnis für Ihre \LaTeX -Dokumente an.
- 2 Erstellen Sie darin mithilfe eines Editors, z.B. emacs eine Datei `hallowelt.tex` mit dem im Beispiel gezeigten Inhalt und übersetzen Sie diese mittels `pdflatex`. Betrachten Sie die so erzeugte PDF-Datei mittels `okular`:

```
okular hallowelt.pdf &
```

Hinweis: PDF-Dateien anzeigen

Falls man ein PDF-Dokument selbst erstellen möchte, sodass dieses sich während des Anzeigens u.U. ändert, ist es sinnvoll, `okular` statt `acroread` zu verwenden. Der PDF-Viewer `okular` lädt die PDF-Datei automatisch neu und aktualisiert die Anzeige, sobald sich die Datei ändert.

Pakete einbinden I

Die Funktionalität von \LaTeX wird durch das Einbinden von Paketen (*package*) wesentlich erweitert. Der \LaTeX -Befehl

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

bindet das Paket `babel` mit der Option `ngerman` ein: Sprachanpassung für deutschsprachige Dokumente, z.B. *Inhaltsverzeichnis* statt *table of contents* usw.

Pakete werden in der **Präambel**, also **nach** `\documentclass{ }` und vor `\begin{document}` eingebunden:

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

Eckige Klammern []

In eckigen Klammern [] stehen i.d.R. optionale Argumente von \LaTeX -Befehlen, d.h. solche, die nicht unbedingt angegeben werden müssen.

Der eigentliche Text des Dokuments

- wird als Fließtext eingegeben, Zeilenumbrüche und mehrfache Leerzeichen und Leerzeilen werden von \LaTeX i.d.R. ignoriert
- eine **Leerzeile** erzwingt einen neuen **Absatz**, d.h. inklusive Einzug (*indent*) der nächsten Zeile
 - sonstige Formatierungen werden von \LaTeX automatisch oder mittels \LaTeX -Befehlen vorgenommen

Hinweis: Eingabe von Umlauten und anderen Nicht-ASCII-Zeichen

Aktuelle \LaTeX -Installationen können mit UTF8 umgehen und verstehen daher auch Umlaute usw. Für ältere \LaTeX -Compiler (vor 2018) wird u.U. noch das `inputenc`-Paket benötigt:

```
\usepackage[utf8]{inputenc}
```

- ein doppelter Backslash `\\` bewirkt einen Zeilenumbruch (sofern von \LaTeX an dieser Stelle erlaubt)

Probleme mit Leerzeilen

Unter gewissen Umständen gestattet \LaTeX keinen Zeilenumbruch mittels `\\`, z.B. direkt nach bestimmten *Umgebungen*, wenn diese bereits einen Zeilenumbruch implizieren. Dann erscheint die Fehlermeldung:

```
! LaTeX Error: There's no line here to end.
```

Soll dann trotzdem ein vertikaler Abstand eingefügt werden, so kann dies z.B. mittels:

```
\vspace{0.2cm}
```

erfolgen.

- standardmäßig wird in \LaTeX der Text im **Blockatz** gesetzt, \LaTeX nimmt mithilfe eines ausgeklügelten Algorithmus automatisch die Worttrennung (“hyphenation”) vor

Aufgabe 3.2 Text im Blockatz

Überzeugen Sie sich von den o.g. Eigenschaften, indem Sie das Pakete `babel` in Ihr Dokument einbinden.

Fügen Sie außerdem (z.B. mittels *Copy & Paste*) einen hinreichend langen Text in Ihr Dokument ein, um die Worttrennung im Ausgabedokument zu sehen.

Hinweis:

- Um den Effekt der automatischen Worttrennung zu provozieren, können Sie auch den *Satzspiegel* verkleinern:

```
\documentclass [DIV=4] {scrartcl}
```

Eventuell ist es dann auch von Vorteil die Option `\sloppy` in der **Präambel** zu setzen.

Exkurs: Zeichenkodierung

- Unicode: int. Standard, um *jedem* Schriftzeichen einen Code zuzuordnen, 17×2^{16} sog. Codepoints, z.B. U+00DF für ß
- ASCII (American Standard Code for Information Interchange): 7 Bit (=128 Zeichen) kodieren das englische Alphabet und ein paar Sonderzeichen (beginnend mit dem Leerzeichen) → universeller Minimalstandard
- ISO-8859-1 (Latin-1): 8 bit (=256 Zeichen) erlauben neben ASCII (= erste 128 Zeichen) die Kodierung weiterer Zeichen, z.B. Umlaute → Familie von Kodierungen
- UTF-8 (Unicode Transformation Format): Kodierung für Unicode, basierend auf 1 bis 4 Bytes (je 8 Bit), erste 128 Zeichen entsprechen ASCII → Quasi-Standard des Internets
- UTF-32: Kodierung aller Unicode-Zeichen mit fester Länge von 4 Bytes → geringer Programmieraufwand vs. Platzverschwendung ($4 \times$ größere Dokumente als mit Latin-1)

Tipp: Aus Gründen der Kompatibilität sollten \LaTeX -Dokumente besser nur ASCII-Zeichen enthalten. Man kann Sonderzeichen mittels \LaTeX -Befehlen erzeugen:

<code>\"a</code>	→	ä	<code>\'{\i}</code>	→	í	<code>\~{n}</code>	→	ñ
<code>\"o</code>	→	ö	<code>\' {e}</code>	→	é	<code>\c{c}</code>	→	ç
<code>\"u</code>	→	ü	<code>\`{a}</code>	→	à	<code>\v{c}</code>	→	č
<code>\ss{}</code>	→	ß	<code>\^{o}</code>	→	ô	<code>\AA</code>	→	Å
<code>\flq</code>	→	‹	<code>\frq</code>	→	›			
<code>\flqq</code>	→	«	<code>\frqq</code>	→	»	(Guillemets [†])		

u.v.m.

[†] Verwendung in Frankreich/Schweiz: « L'État, c'est moi. »

Verwendung in Deutschland: »Machd doch eiern Drägg alleene!«

Anführungszeichen

\LaTeX beherrscht alle Arten von Anführungszeichen, allerdings müssen diese explizit angegeben werden, z.B.

typografische	<code>``</code> (2 schräge)	“H
	<code>"</code> (2 gerade)	H”
deutsche (96)	<code>"`</code> (doppelt + schräg)	„H
	<code>"'</code> (doppelt + gerade)	H“

\LaTeX -Sonderzeichen

Einige Zeichen haben im \LaTeX -Quellcode eine besondere Bedeutung und müssen daher ggf. mit `\` „escaped“ werden.

<code>\textbackslash</code>	→	<code>\</code>	<code>\&</code>	→	<code>&</code>
<code>\#</code>	→	<code>#</code>	<code>\\$</code>	→	<code>\$</code>
<code>\</code>	→	(Leerzeichen)	<code>\%</code>	→	<code>%</code>

L^AT_EX ermöglicht als Textsatzsystem eine automatische Verwaltung von speziellen Textelementen, wie Überschriften, Bildunterschriften, Formeln, Tabellen und Zitaten. Die wesentliche Gliederung eines Dokument erfolgt durch Befehle wie z.B.

In der .tex-Datei

```
\section{Einleitung}
```

Der Befehl `\section{ }` bewirkt folgendes:

- Hervorhebung/Formatierung als Überschrift
- Setzen eines Zählers für die entsprechende Gliederungsebene
- Anlegen eines Eintrags im Inhaltsverzeichnis

Ausgabe in der .pdf-Datei

1 Einleitung

Es gibt folgende vordefinierte Gliederungsebenen (absteigend):

- 1 part
- 2 chapter (nicht in scrartcl)
- 3 section
- 4 subsection
- 5 subsubsection
- 6 paragraph
- 7 subparagraph

Das Inhaltsverzeichnis

Sobald ein Dokument Gliederungsebenen enthält, kann mithilfe des Befehls

```
\tableofcontents
```

an der entsprechenden Stelle im Dokument, i.d.R. vorn, aber **nach** `\begin{document}`, ein automatisch erzeugtes Inhaltsverzeichnis angelegt werden. Darin werden standardmäßig nur die obersten Gliederungsebenen erfasst.

Möchte man danach eine neue Seite beginnen, so hilft

```
\newpage
```

Dieser Befehl funktioniert auch an anderen Stellen im Dokument.

Aufgabe 3.3 Gliederung und Inhaltsverzeichnis

Gliedern Sie Ihr \LaTeX Dokument (längerer Text) mittels der o.g. Gliederungsebenen, denken Sie sich sinnvolle Überschriften aus.

Erzeugen Sie auch ein [Inhaltsverzeichnis](#). Dafür ist es erforderlich mindestens **zweimal** `pdflatex` auszuführen (Warum?).

Falls man in der Präambel mittels

```
\author{Karl-Theodor zu Guttenberg}
```

einen Verfasser und mit

```
\title{Verfassung und Verfassungsvertrag}
```

einen Titel definiert hat, so kann man mittels

```
\maketitle  
\newpage
```

im Dokument (nach `\begin{document}`) eine [Titelseite](#) erzeugen. Das Datum lässt sich mittels

```
\date{30.02.2005}
```

manipulieren.

Beim Übersetzen der `.tex`-Datei werden außer der Ausgabe-PDF-Datei i.d.R. noch weitere Dateien erzeugt:

- `.toc`
Inhaltsverzeichnis
- `.aux`
im Dokument verwendete Gliederungsbefehle,
(Literatur-)Referenzen, etc.
- `.log`
Informationen vom \LaTeX -Compiler (Versionen, Pakete, etc.)

\LaTeX aktualisiert diese Dateien bei jedem Compilerdurchlauf und benutzt dafür Informationen aus dem Dokument. Daher muss z.B. für ein konsistentes Inhaltsverzeichnis zweimal kompiliert werden.

Stichpunkte und nummerierte Listen werden in \LaTeX mithilfe von **Umgebungen** realisiert:

```
\begin{umgebung}
```

```
\end{umgebung}
```

Einfache Stichpunktlisten ermöglicht die Umgebung `itemize`:

```
\begin{itemize}
```

```
  \item ein Stichpunkt
```

```
  \item weiterer Stichpunkt
```

```
\end{itemize}
```

Ergebnis:

- ein Stichpunkt
- weiterer Stichpunkt

Möchte man eine Aufzählung erzeugen, so empfiehlt sich die Verwendung der Umgebung `enumerate`:

```
\begin{enumerate}
  \item erster Punkt
  \item zweiter Punkt
\end{enumerate}
```

Ergebnis:

- 1 ein Stichpunkt
- 2 weiterer Stichpunkt

Man kann bis zu vier Ebenen ineinander verschachteln, wobei für die Nummerierung je nach Ebene folgende Zählungen verwendet werden:

- Arabische Ziffern: 1.
- Kleinbuchstaben: a)
- römische Ziffern (klein): i.
- Großbuchstaben: A.

Aufgabe 3.4 Listen

Erstellen Sie ein \LaTeX -Dokument mit einer **Stichpunktliste** (z.B. Einkaufsliste) und einer **nummerierten Liste** (z.B. bisherige Kursinhalte). Versuchen Sie auch, beide Listen zu schachteln:

```
\begin{enumerate}
  \item erster Punkt
    \begin{enumerate}
      \item erster Unterpunkt
    \end{enumerate}
\end{enumerate}
```

Zwei weitere wichtige Umgebungen sind `center` und `quote`.

Text zentrieren

```
\begin{center}
Auf halbem Weg des Menschenlebens

fand\\
ich mich in einen finstern Wald verschlagen,\\
Weil ich vom rechten Weg mich abgewandt.
\end{center}
```

Ergebnis:

Auf halbem Weg des Menschenlebens
fand
ich mich in einen finstern Wald verschlagen,
Weil ich vom rechten Weg mich abgewandt.

(Dante, Die göttliche Komödie)

Die Umgebung `verbatim` für wortwörtliche Textdarstellung, inklusive der \LaTeX -Sonderzeichen wie `\`, `{}` usw., z.B. für die Darstellung von Programmcode:

```
#include <iostream>
using namespace ::std ;
int main (){ return 0 ; }
```

wird erzeugt durch

```
\begin{verbatim}
#include <iostream>
using namespace ::std ;
int main (){ return 0 ; }
\end{verbatim}
```

Zitate werden mittels quote vom Blocksatz abgesetzt:

Zitate

```
\begin{quote}  
  Gerettet ist das edle Glied\\  
  der Geisterwelt vom Bösen,\\  
  wer immer strebend sich bemüht,\\  
  den können wir erlösen.  
\end{quote}
```

Ergebnis:

*Gerettet ist das edle Glied
der Geisterwelt vom Bösen,
wer immer strebend sich bemüht,
den können wir erlösen.*

(Goethe, Faust II)

Tabellen lassen sich mit Hilfe der Umgebung `tabular` setzen. Diese Umgebung erfordert die Angabe von *Pflichtargumenten* in geschweiften Klammern, nämlich der Anzahl und Ausrichtung der Tabellenspalten (ein Buchstabe pro Spalte):

```
\begin{tabular}{lcr}
Stadt      & Land  & Fluss      \\
Yaoundé   & Yemen & Yukon River
\end{tabular}
```

Ergebnis:

Stadt	Land	Fluss
Yaoundé	Yemen	Yukon River

Dabei bedeuten die Spaltenargumente:

- `l` linksbündig
- `c` centered (zentriert)
- `r` rechtsbündig

Häufig möchte man die Tabellen eines Dokuments durchnummerieren und mit einer *Überschrift* versehen. Dafür gibt es die Umgebung `table`.

```
\begin{table}[h] % versucht, die Tabelle [h]ier zu setzen
\begin{center} % empfehlenswert
\caption{Taxonomie der Biologie}
\begin{tabular}{lcr}
Regnum & Divisio & Subphylum \\
Reich & Abteilung & Unterstamm
\end{tabular}
\end{center}
\end{table}
```

Ergebnis:

Table: Taxonomie der Biologie

Regnum	Divisio	Subphylum
Reich	Abteilung	Unterstamm

Die Optionen zur Positionierung von Tabellen

Tabellen zählen in \LaTeX zu den Gleitobjekten (*float*), d.h. sie werden nicht unbedingt dort im Ausgabe-Dokument erscheinen, wo sie im Eingabetext stehen, sondern dort, wo es den Regeln der Typografie gemäß korrekt ist, i.d.R. am *Anfang einer Seite*.

Dennoch kann man die Positionierung von Tabellen beeinflussen:

```
\begin{table}[!htbp]
```

- [h] hier, falls möglich
- [t] top (oben auf der Seite)
- [b] bottom (unten auf der Seite)
- [p] page, auf extra Seite Tabellen sammeln
- [!] überschreiben bestimmter Float-Beschränkungen
- [H] Hier! (unbedingt) mittels Paket float

Der Standard ist [tbp], empfehlenswert ist [htbp]. Der Parameter H (`\usepackage{float}`) erzwingt die Positionierung an der Stelle des Auftretens um den Preis eines zerhackten Layouts.

Linien in Tabellen

Gelegentlich möchte man ein Tabelle durch Linien lesbarer machen.

Vertikale Linien zwischen den Spalten

```
\begin{tabular}{1|1|1|1}
```

→ gerader Strich | (*Pipe*) zw. den entsprechenden Spalten

Horizontale Linien zwischen Zeilen

```
\hline
```

→ nach einem Zeilenumbruch

Exkurs: Typografische Tabellen

Schöne Tabellen haben nur horizontale Linien verschiedener Dicke. Dafür kann man das Paket

```
\usepackage{booktabs}
```

einbinden. Statt `\hline` verwendet man nun

Linientyp	Ort
<code>\toprule</code>	oben, vor Tabellenkopf
<code>\midrule</code>	nach Tabellenkopf, im Tabellenkörper
<code>\bottomrule</code>	unten, nach letzter Zeile

Aufgabe 3.5 Tabelle erstellen

Setzen und vervollständigen Sie mittels \LaTeX folgende Tabelle:

Sprache	Typ	Kommentarzeichen	Semikolon
Shell	Skript		mehrere Befehle pro Zeile
gnuplot			
\LaTeX			
C++			

Referenzen auf Tabellen

Die Nummerierung, die der `table`-Umgebung von \LaTeX zugeordnet wird, kann mittels `\label{name}` (direkt nach `\caption{ }`) einem festen **internen** Namen zugeordnet werden und mittels `\ref{name}` referenziert werden:

```
...  
\caption{Computersprachen im Kurs}  
\label{tab:programmiersprachen}  
...  
\end{table}
```

Die in Tab.~`\ref{tab:programmiersprachen}`
aufgef\"uhrten Beispiele sind ...

Aufgabe 3.6 Referenzen in \LaTeX

Erzeugen Sie mittels `label` einen internen Namen für Ihre Tabelle und referenzieren Sie diesen Namen im Text.

Hierfür muss `pdflatex` wieder **zweimal** ausgeführt werden.

Was ist der Vorteil einer solchen Referenzierung gegenüber literalen Nummernangaben?

Weitere Anwendungen von Referenzen

Die Verwendung von `\label{name}` und `\ref{name}` ist nicht auf `table`-Umgebungen beschränkt, sondern funktioniert mit allen **Umgebungen**, die eine Nummer erhalten, also auch **Abbildungen**.

Darüberhinaus funktioniert das Referenzieren auch mit **Gliederungsebenen** (z.B. `section`) und **Formeln** (s.u.).

Durch Einbinden des Pakets

```
\usepackage{graphicx}
```

ist es möglich auch Abbildungen aus externen Dateien einzubinden. Dabei werden von [pdflatex](#) folgende Formate unterstützt:

pdflatex: Unterstützte Grafikformate

pdf

png

jpeg

Das Einbinden erfolgt mittels des Befehls

```
\includegraphics[width=x.x,height=y.y,angle=nnn]{datei.pdf}
```

Dabei *kann* die Breite, Höhe und ein Rotationswinkel angegeben werden. Mögliche Einheiten für die Breite und Höhe sind z.B. cm, mm, pt (≈ 0.34 mm), `\textwidth`, `\textheight` usw. Der Winkel wird in Grad angegeben (ohne Einheit dahinter), z.B.:

```
\includegraphics[width=\textwidth,%  
                height=4.5cm,angle=270]{datei.pdf}
```

Aufgabe 3.7 Einbinden von pdf-Grafik

Binden Sie mittels des Pakets `graphicx` und des Befehls `includegraphics` die Grafik-Datei `wr-pz.pdf` in Ihre `.tex`-Datei ein. Versuchen Sie, die Grafik so breit wie möglich zu machen.

Ähnlich wie bei Tabellen, gibt es auch für Abbildungen eine Umgebung, die diese einbettet, eine Nummerierung und eine Bildunterschrift erlaubt:

```
\begin{figure}[htpb!]  
\begin{center}  
\includegraphics{datei.pdf}  
\caption{Impaktparameter-Koordinaten}  
\label{fig:pz-coord}  
\end{center}  
\end{figure}
```

Aufgabe 3.8 Die figure-Umgebung

- 1 Umschließen Sie die eingebundene Abbildung mit einer `figure`-Umgebung.
- 2 Referenzieren Sie die Abbildung.

Anmerkung

Mittels des Befehls

```
\listoffigures
```

kann im \LaTeX -Dokument ein Abbildungsverzeichnis (ähnlich dem Inhaltsverzeichnis) automatisch erstellt werden.

Mit dem Befehl

```
\listoftables
```

kann analog auch ein Verzeichnis aller Tabellen im Dokument erzeugt werden.

Mithilfe des Programms `pdflatex` ist es möglich, PDF-Ausgabe direkt zu erzeugen. Ursprünglich erzeugt das Programm `latex` lediglich Dateien im DVI-Format (`device independent`). DVI-Dateien lassen sich mit DVI-Viewern anzeigen und in PostScript-Dateien umwandeln.

Aufgabe 3.9 DVI-Ausgabe

Übersetzen Sie eine `.tex`-Datei, die keine Abb. enthält (z.B. `meinedatei.tex`) mittels `latex minedatei.tex`

Versuchen Sie die entstandene `.dvi`-Datei zu öffnen.

Hinweis: `latex` kann nur Grafiken im PostScript-Format einbinden (`eps`, `ps`). DVI-Dateien enthalten keine Grafiken und sind deshalb auch relativ klein.

Im nächsten Schritt kann man die `.dvi`-Datei in eine PostScript-Datei mithilfe des Programms `dvips` (`dvi`→`ps`) umwandeln:

Aufgabe 3.10 Das Programm `dvips`

Rufen Sie das Programm `dvips` auf, als Argument übergeben sie den Namen der DVI-Datei aus der letzten Aufgabe, z.B.

```
dvips meinedatei.dvi
```

Öffnen Sie die so erzeugte PS-Datei.

Die so erzeugte PS-Datei enthält nun schon alle wesentlichen Elemente, z.B. auch Grafiken, und lässt sich mittels `lpr` direkt drucken.

Aufgabe 3.11 PDF-Ausgabe

Mithilfe des Programms `ps2pdf` lassen sich PostScript-Dateien in PDF-Dateien umwandeln
`ps2pdf meinedatei.ps`

Vergleichen Sie das so erzeugte PDF mit dem direkt aus `pdflatex` erzeugten PDF.

<code>latex</code>	vs.	<code>pdflatex</code>
<code>dvi</code>	Ausgabe	<code>pdf</code>
<code>ja!</code>	PS-Funktionen	nein
<code>eps, ps</code>	Grafikinput	<code>jpeg, pdf, png</code>
beschränkt	Microtyping	Paket <code>microtype</code>

Neben der bereits behandelten Dokumentklasse `scrartcl` gibt es unzählige weitere. Speziell aus der KOMA-Script-Sammlung (für DIN-Formate):

- `scrartcl` → article
- `scrbook` → book (mit `\chapter`, `\frontmatter` & `\mainmatter`)
- `scrreprt` → report
- `scrlettr` → letter (Briefe)

Dahinter ist jeweils die Entsprechung der Standardklasse (Letter-Papierformat) angegeben.

Für die Klasse `scrbook` ist ein zweiseitiges Layout die Voreinstellung, für die anderen einseitig. Dies kann geändert werden:

```
\documentclass[twoside]{scrartcl}
```

Die wichtigste Option dieser Klassen ist der DIV-Wert zur Erstellung des Satzspiegels:

```
\documentclass [DIV=15] {scrartcl}  
% sehr schmaler Rand; es werden immer 3 Streifen für den Rand verwendet
```

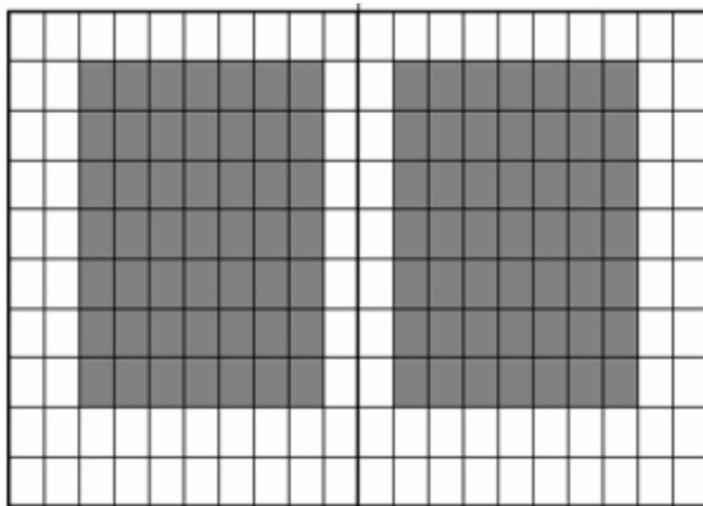


Abbildung: Satzspiegelkonstruktion mittels klassischer Neunerteilung

Exkurs: Satzspiegelkonstruktion mit den KOMA-Script-Klassen

Mit der Option `[DIV=calc]` wird der Satzspiegel automatisch für andere Formate als DIN A4 berechnet, z.B. beim Verwenden der Option `[a5paper]`. Durch die Option `[DIV=classic]` wird eine mittelalterliche Satzspiegelkonstruktion verwendet. *Nach* dem Laden bestimmter Pakete (z.B. für alternative Fonts) ist u.U. eine erneute Satzspiegelberechnung mittels `\typearea[current]{calc}` nötig.

Die sogenannte Bindekorrektur wird mittels der Option `[BCOR=Länge]` angegeben.

Beispiel:

```
\documentclass[a5paper,DIV=classic,BCOR=5mm]{scrreprt}
\usepackage{bookman}
\typearea[current]{calc}
```

Weitere Dokumentklassen:

- beamer: Bildschirmpräsentationen (diese)
- aa: Publikationsstil für Astronomy & Astrophysics
- sciposter: Poster, z.B. im A0-Format

u.v.m.

Die Standardeinstellungen von L^AT_EX bzw. den verwendeten Dokumentklassen, wie etwa `scrartcl` sind i.d.R. sinnvoll und sollten nicht willkürlich geändert werden. Dennoch sind Änderungen möglich.

Fontgröße

kann am Anfang des Dokuments absolut in Punkten festgelegt werden:

```
\documentclass[11pt,DIV=12]{scrbook}
```

Darüberhinaus kann die Schriftgröße jederzeit relativ geändert werden:

```
{\Large}Eine \"Uberschrift}\\
```

Die \"Anderung gilt nur innerhalb der geschweiften Klammern.

Eine Überschrift

Die Änderung gilt nur innerhalb der geschweiften Klammern.

Es gibt folgende Größenbefehle (i.d.R. jeweils 1pt Unterschied):

<code>\tiny</code>	aA
<code>\scriptsize</code>	aA
<code>\footnotesize</code>	aA
<code>\small</code>	aA
<code>\normalsize</code>	aA
<code>\large</code>	aA
<code>\Large</code>	aA
<code>\LARGE</code>	aA
<code>\Huge</code>	aA

Fontform ändern

Je nach Dokumenttyp gibt es eine voreingestellte Schriftart, z.B. Computer Modern mit Serifen (CMS) für scrartcl.

Die Erscheinung kann u.a. wie folgt geändert werden:

<code>\sffamily</code>	Sans Serif – ohne Serifen
<code>\rmfamily</code>	Roman – normale Schrift mit Serifen
<code>\ttfamily</code>	TypeWriter – Monospace-Schrift mit breiten Serifen
<code>\slshape</code>	<i>slanted</i> – geneigte Schrift
<code>\scshape</code>	SMALL CAPITALS – Kapitälchen
<code>\bfseries</code>	Bold Face – Fettdruck
<code>\emph{}</code>	<i>emphasize</i> – meist durch kursive, geneigte Schrift

Bestimmte Kombinationen sind auch möglich:

```
{\bfseries\sffamily{ }Fett ohne Serifen}
```

Fett ohne Serifen

Die größte Stärke von \LaTeX ist der Formelsatz. Formeln können auf zwei Arten gesetzt werden:

- im laufenden Text, also etwa $\sum \frac{1}{x^2}$ usw. mittels:

... etwa `\sum \frac{1}{x^2}` usw.

- oder abgesetzt mit fortlaufender Nummerierung:

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad (1)$$

Die Formel für die Riemannsche Zetafunktion von 2 wurde so erzeugt:

```
\begin{align} % erfordert \usepackage{amsmath}
\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2}
         = \frac{\pi^2}{6}
\end{align}
```

Aufgabe 3.12 Formeln setzen

Schreiben Sie in ihrem \LaTeX -Dokument die Reihendarstellung für die Eulersche Zahl sowohl im laufenden Text als auch als abgesetzte Formel (Package `amsmath` erforderlich).

Hinweis: Die Pakete `amsmath`, `amsfonts`, `amssymb` erweitern die Möglichkeiten zur Darstellung von (mathematischen) Symbolen enorm, z.B. `\checkmark` = ✓, `\lesssim` = ≲

Desweiteren ermöglicht das Paket `amsmath` die Verwendung des Befehls `\text{ }` in der Matheumgebung, um **normalen Text**, z.B. **für physikalische Einheiten**, zu setzen:

$$U = 230 \text{ V}$$

mittels: `U = 230\,\text{V}`

Außerdem `\text{ }` auch für tiefgestellten Text, der keine Variablen bezeichnet, z.B. M_{ini} , v_{rad}

Die `align`-Umgebung erlaubt darüberhinaus mehrzeiligen Formelsatz mit Formatierung (intern ähnlich einer Tabelle mit `&`):

```
\begin{align}
\zeta(2) &= \sum_{n=1}^{\infty} \frac{1}{n^2} \\
&= \frac{\pi^2}{6} \label{eq:pi}
\end{align}
```

Achtung: Keine Leerzeilen in `align`!

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} \quad (2)$$

$$= \frac{\pi^2}{6} \quad (3)$$

Die einzelnen Zeilen tragen fortlaufende Nummern, die sich wieder mit `label/ref` referenzieren lassen, übliche Form: Gl. (3).

Möchte man Formeln ohne Nummerierung setzen, so kann die Nummerierung mittels * unterdrückt werden, also

```
\begin{align*}
  \sin ^2 + \cos ^2 = 1
\end{align*}
```

$$\sin^2 + \cos^2 = 1$$

Hinweis: Genauso wie Einheiten (V, kg, cm, usw.) werden auch Funktionsnamen (sin, cos, tan, usw.) in Formeln gerade als normaler Text gesetzt. Dafür gibt es vordefinierte Makros, nämlich `\sin`, `\cos`, usw.

Aufgabe 3.13 Mehrzeilige Formeln

Setzen Sie eine mehrzeilige Formel und referenzieren Sie diese im Text. (Bitte daran denken, `pdflatex` zweimal auszuführen.)

Exkurs: Formelumgebungen

Die für das Setzen von **Inline-Formeln** verwendeten Dollarzeichen $\$ \dots \$$ sind eine Abkürzung für `\begin{math} \dots \end{math}`. Alternativ kann man auch die Abkürzung `\(\dots \)` verwenden.

Einzeilige abgesetzte Formeln können in L^AT_EX mit `\begin{displaymath} \dots \end{displaymath}` bzw. mit der entsprechenden Abkürzung `\[\dots \]` gesetzt werden, erhalten dann aber **keine** fortlaufende **Nummerierung**. Außerdem gib es noch eine abgesetzte Formelumgebung, die bereits in T_EX existiert und mit doppelten Dollarzeichen umschlossen wird: $\$ \$ \dots \$ \$$ (nicht empfohlen).

Einzeilige abgesetzte Formeln **mit fortlaufender Nummerierung** erhält man auch mittels `\begin{equation} \dots \end{equation}`.

Neben der `align`-Umgebung aus dem `amsmath`-Package gibt es für mehrzeilige abgesetzte Formeln auch noch die `eqnarray`-Umgebung, die allerdings die Abstände falsch setzt und daher nicht empfohlen wird:

$$x = 3 \quad \text{equation} \quad (4)$$

$$x = 3 \quad \text{align} \quad (5)$$

$$x \quad = \quad 3 \quad \text{eqnarray} \quad (6)$$

(Quelltext für `eqnarray`: `x \& = \& 3`, also intern drei Spalten, statt zwei wie bei `align`.)

Es gibt noch weitere Möglichkeiten Formeln zu formatieren, z.B.

```
e = \lim _{n\rightarrow \infty}
\left( 1 + \frac{1}{n} \right)^n
\nonumber
```

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n$$

Der Befehl `\nonumber` schaltet die Nummerierung für die aktuelle Formelzeile ab.

Die Befehle `\left` und `\right` gefolgt von einer entsprechenden Klammer `()` `[]` `\{ \}` sorgen dafür, dass die Höhe der Klammer an den Inhalt angepasst wird.

Ein weiteres Beispiel:

```
H(x) = \left\{
  \begin{array}{ll}
    0: & x < 0 \\
    1: & x \geq 0
  \end{array}
\right.
```

erzeugt die stückweise definierte Heaviside-Funktion:

$$H(x) = \begin{cases} 0 & : x < 0 \\ 1 & : x \geq 0 \end{cases} \quad (7)$$

Dabei fungiert `\right.` als Blindmarkierung für die Klammernanpassung. Die Umgebung `array` funktioniert wie die `tabular`-Umgebung und kann auch zur Darstellung von $n \times m$ -Matrizen verwendet werden.

Einige weitere Formelzeichen bzw. Formatierungen:

$$\dot{L} \stackrel{!}{=} 0 \quad \text{\dot{L} \stackrel{!}{=} 0}$$

$$\sqrt{\sqrt{16}} = \sqrt[4]{16} \quad \text{\sqrt{\sqrt{16}} = \sqrt[4]{16}}$$

$$\int_M d\omega = \oint_{\partial M} \omega \quad \text{\int_M d\omega = \oint_{\partial M} \omega}$$

$$e = \underbrace{\frac{1}{0!}}_{0!=1} + \dots \quad \text{e=\underbrace{\frac{1}{0!}}_{0!=1}+\ldots}$$

- hyperref

```
\usepackage{hyperref}
```

→ ermöglicht anklickbare Links in PDF-Dokumenten, z.B. Inhaltsverzeichnis, Referenzen, URLs

```
\url{https://www.astro.physik.uni-potsdam.de}
```

<https://www.astro.physik.uni-potsdam.de>

(Obiger Link ist anklickbar!)

Achtung!

Das Paket `hyperref` muss stets als **letztes** Paket in der Präambel eingebunden werden.

- PSTricks

```
\usepackage{pstricks}
```

→ PostScript-Grafik mittels \LaTeX -Befehlen (erfordert latex + dvips)

```
\begin{pspicture}(-1,-1)(3,1)  
  \psline{<->,linecolor=blue}(0,0)(2,0.5)  
\end{pspicture}
```



Weitere Beispiele, Pakete und Anleitung unter:

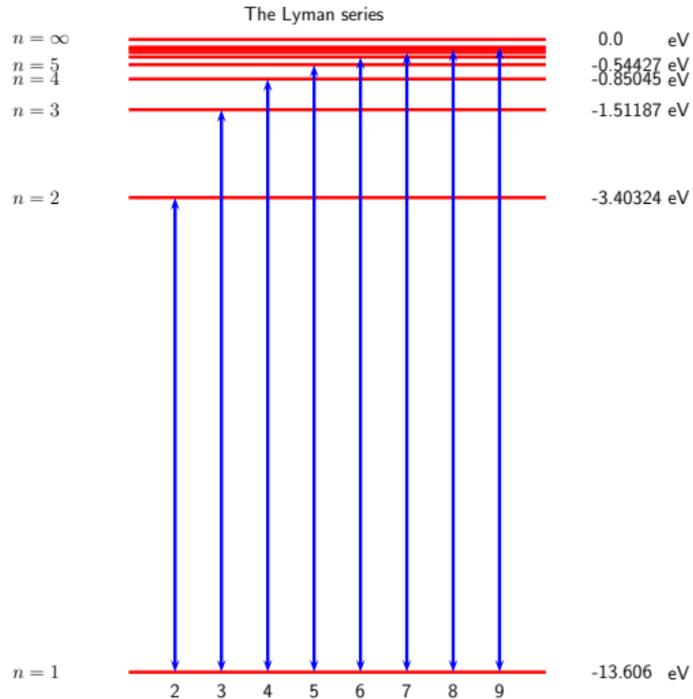
```
http://tug.org/PSTricks/main.cgi/
```

- `multido`

```
\usepackage{multido}
```

→ einfache Schleifen zu Befehlswiederholung

```
\multido{\i= 2 + 1 }{8}  
{  
  \setlength{\xdist}{1.0cm * \real{\i}}  
}
```



Beispiel: `pstricks + multido (lymanseries.tex)`

Für \LaTeX gibt es ein komfortables System zum Erstellen von Literaturreferenzen namens `bibtex`.

Dazu wird die Referenz in einer Datei mit der Endung `.bib` geschrieben:

Format von bib-Einträgen

```
@ARTICLE{acknein2003,  
  author = {{Acker}, A. and {Neiner}, C.},  
  title = "{Quantitative classification of WR nuclei  
    of planetary nebulae}",  
  journal = {A \& A},  
  year = 2003,  
  volume = 403,  
  pages = {659-673}  
}
```

In der ersten Zeile wird der *interne Name* für das Referenzieren angegeben (hier: `acknein2003`). Dieser kann dann im Dokumenttext mittels `\cite{acknein2003}` referenziert.

Das Erstellen einer Bibliografie im Dokument veranlasst der Befehl

```
\bibliography{meinebibdatei} % ohne Endung .bib
```

am Ende des Dokuments.

Damit das Referenzieren funktioniert, muss das Programm `bibtex` ausgeführt werden, dieses benötigt als Argument den Namen der `.aux`-Datei des `TEX`-Dokuments:

```
bibtex texdocument.aux
```

Danach muss noch **zweimal** `pdflatex` ausgeführt werden.

Hinweis: Dafür empfiehlt sich der Einsatz einer \LaTeX -Entwicklungsumgebung wie TeXnic-Center, TeX-Shop oder der Einsatz eines Shell-Skripts.

Das Referenzieren erfolgt mithilfe des Befehls

Beispiel für das Zitieren

```
\cite{acknein2003}
```

An der entsprechenden Stelle im Dokument wird eine Referenz angezeigt und diese im Bibliografie-Abschnitt dargestellt. Wie die Referenz dargestellt wird, muss mittels

Bibliografiestil definieren

```
\bibliographystyle{plain}
```

in der Präambel vereinbart werden. Es stehen viele Stile zur Auswahl, Zeitschriften (z.B. A & A) definieren häufig ihre eigenen Stile.

Beispiel

In [1] werden Wolf-Rayet-Zentralsternen von Planetarische Nebel klassifiziert.

Literatur

[1] A. Acker and C. Neiner.

Quantitative classification of WR nuclei of planetary nebulae.

A & A, 403:659–673, 2003.

Aufgabe 3.14 BibTeX verwenden

Legen Sie selbst eine `.bib`-Datei für Ihr Dokument an, fügen Sie einen Eintrag ein und zitieren Sie diesen in Ihrem Dokument. Denken Sie an die darüber hinaus notwendigen Befehle `\bibliographystyle{ }` und `\bibliography{ }` sowie das Ausführen von `bibtex` und `pdflatex`.

Zusammenfassung:

Das .bib file

```
@ARTICLE{NameYear, ... }
```

Im eigentlichen .tex-Dokument

```
...  
\bibliographystyle{plain} % oder: alpha, apalike, ...  
\begin{document}  
...  
\cite{NameYear} % Label aus .bib-Datei  
...  
\bibliography{...} % Name der .bib-Datei ohne Endung .bib
```

- 1 pdflatex document.tex
- 2 bibtex document.aux
- 3 pdflatex document.tex
- 4 pdflatex document.tex

Während der Bibliographie-Stil `plain` sehr kompakt und verbreitet in der Physik ist, wird in der Astrophysik meist ein anderer, besser lesbarer Stil verwendet:

Exkurs: Bibliographie-Stil in der Astrophysik

In der Präambel:

```
\usepackage{natbib}
\bibliographystyle{mnras}
```

Das `natbib`-Paket stellt `\citet{}` und `\citep[[] [] {}]` zur Verfügung, die entsprechenden Referenzen im Text für `\citet{cak1975}` und für `\citep[e.g.,][and references therein]{cak1975}` sehe dann so aus:

... wie in `Castor et al. (1975)` beschrieben ...

Viele Autoren (z.B. `Castor et al., 1975, und Referenzen darin`) haben ...

Und der Bibliographie-Eintrag (Einträge in alphabetischer Reihenfolge):

Castor J. I., Abbott D. C., Klein R. I., 1975, ApJ, 195, 157

Zur Dokumentstrukturierung stehen noch viele weitere Möglichkeiten zur Verfügung, z.B.

- Fußnoten¹ mittels `\footnote{ }`
- Appendix mittels `\appendix`
- Index mit dem Programm `makeindex` und den Befehlen: `\makeindex` (Präambel, erfordert `\usepackage{makeidx}`), `\index{ }` (im Text) und `\printindex :`
`pdflatex datei.tex ; makeindex datei.idx ; pdflatex datei.tex`
- Rechtschreibprüfung mittels `ispell`

```
ispell -t -d deutsch hallowelt.tex
```

¹Fußnoten sind z.B. bei Tabellen nützlich.

Mikrotypografie

- die mikrotypografischen Fähigkeiten von pdf \TeX sind mithilfe des Pakets `microtype` nutzbar und werden z.T. schon durch Einbinden des Pakets aktiviert:

```
\usepackage{microtype}
```

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstabhäusern an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen sie in Buchstabhäusern an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht.

links: ohne Paket `microtype`
rechts: mit `microtype`, insbesondere mit optischem Randausgleich für *optisch* glatte Satzkante (rechts)

- ggf. ist es nötig, manuell einzugreifen:

Befehl	Bedeutung	Beispiel
<code>\kernx</code>	fügt einen Abstand x (auch negativ!) ein	<code>\kern1mm</code>
<code>\,</code>	fügt einen kleinen Abstand ein	<code>\$U=230\$\,V</code>
<code>\!</code>	fügt einen kleinen negativen Abstand ein	<code>W\!er</code>

Wer? *geometrisch* gleichmäßiger Abstand ohne Kerning

Wer? *optisch* gleichmäßiger Abstand mit Kerning

Nachteile von \LaTeX :

- Standardausgabeformat ist das ansonsten ungebrauchliche DVI (Device independent file format):
→ aber: `pdflatex` erzeugt direkt PDF-Dateien
- hohe Hürden für Einsteiger durch Abstraktion des Textsatzes (kein WYSIWYG)
- einige Automatismen des Textsatzes beschränken freie Gestaltungsmöglichkeiten (kein *echtes* Layouting)
- einige Pakete werden nur von `latex`, andere nur von `pdflatex` unterstützt (z.B. `pstricks` und `microtype`)
- obwohl Ausgabe als PDF mittlerweile üblich, keine volle PDF-Unterstützung (z.B. Medien-Einbettung vom Betriebssystem abhängig)

Stärken von \LaTeX :

- **kostenlos und auf allen Systemen verfügbar** → \TeX Live
- Trennung von Form und Inhalt (leider nicht vollständig)
- dadurch hohe Wiederverwertbarkeit & Flexibilität
- perfekt gesetzte Formeln und Dokumente (z.B. Briefe)

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

- Automatismen für Inhaltsverzeichnis, Literaturverzeichnis, Hyperlinks
- beliebig lange Dokumente, Änderungen an beliebigen Stellen möglich, stabil, geringer Speicherbedarf
- weit verbreitet - in vielen Bereichen der De-facto-Standard

Anhang: Weitere Pakete

Fußnoten

- `\usepackage{tablefootnote}`: Fußnoten auch in Tabellen mittels `\tablefootnote`, so wie normale Fußnoten, Fußnotentext unten, abgesetzt, auf der Seite; funktioniert nur in `table`-Umgebung
- `\usepackage{threeparttable}`: die Umgebung `threeparttable` in der `table`-Umgebung, um `tabular`-Umgebung, ermöglicht mittels `\tnote{zeichen}` Fußnotenmarkierung in `tabular`, die mittels `\item[zeichen]` in nachfolgender `tablenotes`-Umgebung referenziert werden kann:

threeparttable

```
\begin{table} \begin{center}
\begin{threeparttable}
\begin{tabular}{llll}
...
Rom & Athen & London\tnote{*} & Berlin \\
...
\end{tabular}
\begin{tablenotes}
\footnotesize
\item[*] nicht-EU
\end{tablenotes}
\end{threeparttable}
\end{center}\end{table}
```

Bunte Tabellen

- `\usepackage{color}` % Farben verwenden
`\usepackage{colortbl}` % Tabellenzellen färben

in der Präambel Farben definieren, z.B.

```
\definecolor{hg}{rgb}{0.95,0.95,0.95}
```

und in der Tabellenzelle dann: `\cellcolor{hg}`, z.B.

Sprache `\cellcolor{hg}` & Typ & Kommentarzeichen & Semikolon `\`