Erstellung wissenschaftlicher Texte mit LaTeX

Stephan Graf, Beate Herrmann
Zentralinstitut für Angewandte Mathematik

5.-7. Mai 1998

Gliederung

Benutzung und Grundlagen von LaTeX

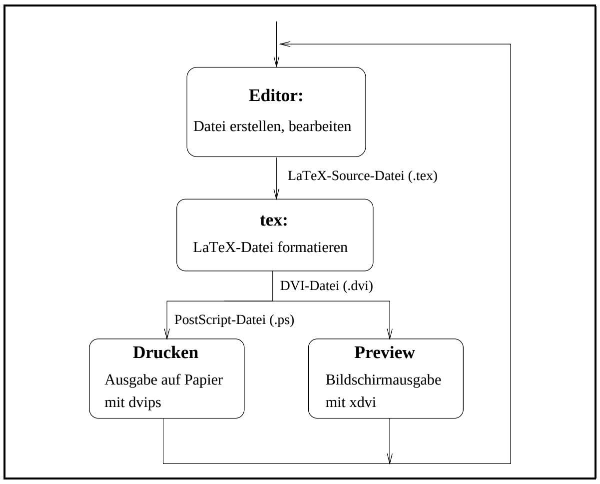
Allgemeines

- TeX ist eine Art Programmiersprache
- LaTeX ein darin geschriebenes Makropaket
- statt eines *ausführbaren Programms* wird ein *druckbares Dokument* erzeugt
- Erstellung des LaTeX-Codes mit Hilfe eines Editors
- Übersetzung des Codes mit dem Formatierer `tex` in ein druckerunabhängiges Beschreibungsformat (*dvi*-Datei)
- Betrachten des formatierten Dokuments mit einem Previewer (`xdvi`)
- Drucken mit einem Druckertreiber (`dvips`)
- Oberfläche `xtem` zum Steuern der einzelnen TeX-Programme

Historie

- **TeX**
  - Formatierer für die Programmiersprache TeX
  - von Donald E. Knuth 1977 an der Stanford University entwickelt
  - sollte rechnerunabhängig sein
  - sollte als Public Domain Software frei benutzbar sein
  - ist besonders im akademischen Bereich weit verbreitet
- METAFONT
  - dient zum Entwurf von Schriften und Schriftfamilien
  - wird für die von TeX benötigten Computer Modern Schriften gebraucht
- **LaTeX**
  - ein von Leslie Lamport in TeX geschriebenes Makropaket
  - gibt das Layout des Dokuments vor, Eingabe bezieht sich eher auf die Dokumentstruktur
  - mehrere Dokumentklassen: *article*, *report*, *book*, ...
  - letzte Version: LaTeX 2.09
- **LaTeX 2ε**
  - seit Juni 1994 Standard für LaTeX
  - bessere Behandlung von Erweiterungspaketen
  - neues Verfahren für die Zeichensatzauswahl
  - jedes halbe Jahr ein Update
- **LaTeX3**
  - seit 1989 in Planung, aber noch Zukunft
  - Merkmale: effizienter Kern, keine Deklarationen mehr, Entwicklung in Richtung SGML

Verfahren



- kein WYSIWYG
- iterativer Prozeß: Editieren, Formatieren, Kontrolle
- Eingabedatei sollte immer die Endung *.tex* haben
- Ausgabedatei hat die Endung *.dvi* und wird vom Druckertreiber und vom Previewer benutzt
- Drucken mit Hilfe von `dvips` über die Seitenbeschreibungssprache PostScript

Editor

- Editor GNU Emacs wird empfohlen
- Eigenschaften:
  - Einfügen von Templates (Schablonen) für logische Strukturen wie Umgebungen und Abschnitte
  - Tastenkombination für die oft genutzten LaTeX-Konstrukte, wie Fontänderungen, Akzente und mathematische Symbole
  - Starten aller TeX-Programme über eine Tastenkombination `C-c C-c`; dabei selbständige Auswahl des nächsten Bearbeitungsschritts
  - Unterstützung von Dokumenten, die auf mehrere Dateien aufgeteilt sind
  - Online-Hilfe bei Fehlermeldungen von LaTeX
  - Bearbeitung des Dokuments
  - Formatierung des Source-Codes und farbliche Darstellung von TeX-Befehlen
- Aufruf: `emacs file.tex`
- AUC TeX ist hier schon eingestellt
- Referenzkarten und Dokumentation zu AUC TeX:
  (unter **/usr/local/emacs/doc**)
  - `auc-tex.dvi`, „User's Guide“
  - `tex-ref.dvi`, „AUC TeX Reference Card“
  - `math-ref.dvi`, „Math Mode Reference Card“

Formatieren

- Aufruf: `latex file.tex`
- im Editor mit `C-c C-c` (`LaTeX`)
- Bildschirm-Ausgabe bei erstem Beispiel:

```
zdv087@zam100: latex bsp1.o.tex
This is TeX, Version 3.1415 (C version 6.1)
(bsp1.o.tex
LaTeX2e <1995/06/01>
Hyphenation patterns for english, german, loaded.
(/usr/local/texmf/tex/latex2e/base/article.cls
Document Class: article 1995/06/26 v1.3g
Standard LaTeX document class
(/usr/local/texmf/tex/latex2e/base/size10.clo))
(/usr/local/texmf/tex/latex2e/packages/german/german.sty
Package `german', Version 2.5b of 95/01/20.)
(bsp1.o.aux) [1] (bsp1.o.aux) )
Output written on bsp1.o.dvi (1 page, 1424 bytes).
Transcript written on bsp1.o.log.
```

- Anzeige im Editor mit `C-c C-l`
- LaTeX geht bei Fehlern in den interaktiven Modus über:
  - `x`: LaTeX verlassen; `q`: im Batch weiter formatieren
  - `C-d`: falls nach einer nicht vorhandenen Datei gefragt wird
- beim Start vom Editor aus:
  - kein interaktiver Modus
  - Fehlerstelle kann mit `` C-c ` `` direkt angesprungen werden

Preview

- Aufruf: `xdvi file.dvi`
- im Editor mit `C-c C-c` (`View`)
- PostScript-Graphiken und -Schriften werden auch angezeigt

Drucken

- Aufruf: `dvips -P`*queue* `file.dvi`
- im Editor mit `C-c C-c` (`Print`)
- Fonts werden im Bitmap (*.pk*) Format benötigt und ggf. mit METAFONT erzeugt.
- Ausgabe auf eine PostScript-Datei:
  `dvips [-P`*queue*`] file.dvi -o`
  erzeugt `file.ps`
- PostScript-Datei ist je nach angewähltem Drucker unterschiedlich, da andere Bitmap-Fonts verwendet werden

Oberfläche `xtem`

- Aufruf: `xtem file.tex`
- steuert alle benötigten Funtionen über Tastendruck
- viele Einstellungsmöglichkeiten

Erstes Beispiel

```
\documentclass{article}
\usepackage{german}
\begin{document}
F"ur eine Publikation "ubergibt der Autor dem Verleger
"ublicherweise ein maschinengeschriebenes Manuskript.
Der Buch-Designer des Verlages entscheidet
dann "uber das Layout des Schrift"-st"ucks (L"ange
einer Zeile,
Schriftart, Abst"ande vor und nach Kapiteln usw.) und
schreibt dem Setzer die daf"ur notwendigen
Steuerdaten dazu.

\LaTeX\ ist sozusagen der Buch-Designer, \TeX\ ist sein
Setzer. Die eingegebenen \LaTeX-Befehle werden in um
Stufen niedrigere \TeX-Setzbefehle "ubersetzt.

Ein menschlicher Buch-Designer erkennt die Absichten
des Autors meistens auf Grund seines Fachwissens aus dem
Inhalt des Manuskripts. \LaTeX\ dagegen
ist "`nur"' ein Programm und
ben"otigt daher zus"atzliche Informationen vom Autor,
die die logische Struktur des Textes angeben.
\end{document}
```

Für eine Publikation übergibt der Autor dem Verleger üblicherweise ein maschinengeschriebenes Manuskript. Der Buch-Designer des Verlages entscheidet dann über das Layout des Schriftstücks (Länge einer Zeile, Schriftart, Abstände vor und nach Kapiteln usw.) und schreibt dem Setzer die dafür notwendigen Steuerdaten dazu.

LaTeX ist sozusagen der Buch-Designer, TeX ist sein Setzer. Die eingegebenen LaTeX-Befehle werden in um Stufen niedrigere TeX-Setzbefehle übersetzt.

Ein menschlicher Buch-Designer erkennt die Absichten des Autors meistens auf Grund seines Fachwissens aus dem Inhalt des Manuskripts. LaTeX dagegen ist „nur“ ein Programm und benötigt daher zusätzliche Informationen vom Autor, die die logische Struktur des Textes angeben.

Syntax der Befehle

- spezielle Befehle aus einem Zeichen:
  % # $ & ~ _ ^ { }
  können mit einem Backslash \ maskiert werden
- Befehle aus zwei Zeichen,
  z.B. \&, meistens Maskieren der oben genannten Befehle, \\ (Zeilenumbruch)
- allgemeine Befehle:
  - beginnen mit \
  - gefolgt von einem oder mehreren Buchstaben
  - wird vom ersten Zeichen beendet, das kein Buchstabe ist
- Argumente:
  - zwingende Argumente mit { und }
  - optionale Argumente mit `[` und `]`
  - mehrere zwingende Argumente in einzelne Klammerpaare
  - mehrere optionale Argumente mit Komma trennen

- Befehlsende
  - bei optionalen und zwingenden Argumenten durch Klammer
  - Leerzeichen wird als Befehlsende erkannt und erscheint **nicht** im Text
  - Buchstaben direkt hinter dem Befehl:
    `\TeX niker` → TeXniker,
    `\TeX{}niker` → TeXniker,
    `{\TeX}niker` → TeXniker
  - Leerzeichen erwünscht:
    `\LaTeX\ Symbol` → LaTeX Symbol
- Beispiele:
  - `\usepackage[dvips]{epsfig}`
  - `\rule[-3pt]{4cm}{5mm}` →

Umgebungen

- verändertes Verhalten für einen ganzen Bereich
- Syntax:

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

- Beispiele:
  - **quote**-Umgebung
    → rückt den Text beidseitig ein
  - **center**-Umgebung
    → zentriert den Text

Namenlose Umgebungen

- werden mit { und } markiert
- haben keine direkte Wirkung auf den Text innerhalb der Klammern
- sinnvoll im Zusammenhang mit Deklarationen (→ nächste Folie)

Deklarationen

- in der Kopka-Reihe auch mit *Erklärungen* benannt
- Veränderungen von Parametern (z.B. Schriftarten), die
  - mit der Deklaration selbst beginnt und
  - bis zum Ende der aktuellen Umgebung reicht (`\end{...}` oder } einer namenlosen Umgebung)
- Beispiele:
  - Wechsel der Schriftart
    `{ text \bfseries text }`
    → der Befehl `\bfseries` setzt die aktuelle Schriftart auf Fettdruck (*bold font*), die Wirkung endet mit }
  - Wechsel der Schriftgröße
    `{ text \small text }`
    → der Befehl `\small` setzt den Text hinter dem Befehl in einer kleineren Schriftgröße
- spezielle Deklarationen:
  - Deklarationen für Längen
    `\setlength{\parindent}{0cm}` → setzt die Einrücktiefe der ersten Zeile eines Paragraphen (Absatz)
  - Deklarationen für Zähler `\setcounter{page}{4}` → setzt die Seitennummer
- globale Parameter
  - unabhängig von Umgebungen

Maßeinheiten

- **feste Maße**
  - Dezimalzahl, ggf. mit Vorzeichen,
    mit Dezimalpunkt oder Komma,
    gefolgt von einer Maßeinheit:
    `cm` : Zentimeter,
    `mm` : Millimeter,
    `in` : Inch, Zoll, 2.54 cm,
    `pt` : Punkt, 1 in = 72.27 pt,
    `pc` : Pica, 1 pc = 12 pt,
    `em` : Breite des großen ≫M≪ im aktuellen Zeichensatz
    `ex` : Höhe des Buchstaben ≫x≪ im aktuellen Zeichensatz
- Beispiele:
  `3mm`, `1.5em`, `-3.5ex`, `1,5ex`
- Zuweisung durch Befehl:
  `\setlength{\befehl}{Wert}`
- **elastische Maße**
  - Maße, die um einen bestimmten Betrag von TEX erhöht oder erniedrigt werden dürfen
  - z.B. wichtig bei dem Abstand zweier Absätze, bessere Aufteilung der Seite
  - Syntax:
    sollwert **plus***dehnwert* **minus***schrumpfwert*
  - Beispiel: (Abstand zweier Absätze)
    `\setlength{\parskip}{1ex plus0.5ex minus0.4ex}`

Sonderzeichen

- Zeichen, die nicht über die Tastatur erreicht werden können
- Umlaute und ß
  - mit **german**-Style:
    `"a`→ä, `"o`→ö, `"u`→ü,
    `"A`→Ä, `"O`→Ö, `"U`→Ü, `"s`→ß
  - ohne **german**-Style:
    `\"a`→ä, `\"o`→ö, `\"u`→ü,
    `\"A`→Ä, `\"O`→Ö, `\"U`→Ü, `\ss`→ß
  - deutsche Tastatur: Erweiterungspaket **inputenc**
- Anführungstriche:
  - doppelte Hochkomma: ``` ``Wort'' ``` → “Wort”
  - mit **german**-Style: ``"`Wort"'`` → „Wort“
  - originale ”durch `\dq`
  - französische Anführungszeichen:
    `\frqq Wort \flqq` → ≫Wort ≪

- Striche
  - Trennstrich: `Trenn-strich` → Trenn-strich
  - Streckenstrich:
    `Hamburg--J"ulich` → Hamburg–Jülich
  - Gedankenstrich:
    `denken --- woran` → denken — woran
    - -- ist bei Gedankenstrich auch üblich
- Akzente:
  `\`{o}`→ ò, `\'{o}`→ ó, `\^{o}`→ ô,
  `\.{o}`→ ȯ, `\u{o}`→ ŏ, `\~{o}`→ õ,
  `\v{o}`→ ǒ, `\H{o}`→ ő, `\b{o}`→ o̲,
  `\c{o}`→ o̧, `\d{o}`→ ọ, `\t{oo}`→ o͡o
- Kommentar
  - Kommentarzeichen: %
  - Text vom Zeichen % bis zum Zeilenende wird von TEX nicht beachtet

Dokumentklassen und Packages

Klassen

- Dokumentklasse legt Layout und Struktur des Dokuments fest
- Syntax: `\documentclass[`*options*`]{`*klasse*`}`
- Dokumentklassen (Standard):

book	Erstellung von kompletten Büchern
report	für längere Berichte, die aus mehreren Kapiteln bestehen, Diplomarbeiten, Dissertationen u.ä.
proc	für Konferenzbände (proceedings)
article	für Artikel in wiss. Zeitschriften, Praktikums- und Seminararbeiten, kürzere Berichte u.ä.
letter	für Briefe

- weitere Klassen:

slides	Erstellung von Folien (**slitex**)
foils	Erstellung von Folien (**foiltex**)
dinbrief	für deutsche Briefe
scrbook, scrart, ...	Klassen aus dem Paket **koma-script**, Anpassung an deutsches Layout

- Source-Dateien mit den Definitionen für die Klasse haben die Endung *.cls*
- **Optionen**

11pt, 12pt	Größenangabe für die voreingestellte Schrift
twoside	Erstellen zweiseitiger Dokumente
twocolumn	zweispaltige Ausgabe
a4paper	Größenanpassung an DIN A4-Format

- Beispiel:
  `\documentclass[11pt,twoside,a4paper]{book}`

Dokumentaufbau

- Dokument besteht aus Vorspann und Hauptteil

```
\documentclass[options]{class}
...
(Vorspann)
...
\begin{document}
...
(Hauptteil, Body)
...
\end{document}
```

- Vorspann
  - enthält Definitionen, die für das ganze Dokument gelten
  - enthält Befehle, die weitere Styles oder Pakete laden
- Haupteil, Body
  - enthält den Text des Dokuments
  - Aufbau hängt von der Dokumentklasse ab
  - bei **article**, **report** und **book** bis auf wenige Ausnahmen gleiche Dokumentstruktur

Dokumentstruktur

logische Struktur eines kleinen Dokuments

```
\documentclass{article}
\begin{document}
  ...
  \maketitle
  \begin{abstract}...\end{abstract}
  \section{...} ...
  \section{...} ...
    \subsection{...} ...
      \subsubsection{...} ...
    \subsection{...} ...
  \section{...} ...
  \begin{thebibliography}...
        \end{thebibliography}
\end{document}
```

besteht aus

- Dokumentklasse **article**
- Titelseite
  - wird durch `\maketitle` gedruckt
  - muß mit anderen Befehlen vorher definiert werden
- Abstract
  - Zusammenfassung des Artikels
- Kapitel und Unterkapitel
  - bei Dokumentklasse **article** kein **chapter**
- Literaturverzeichnis

Logische Struktur eines größeren Dokuments

```
\documentclass{book}
\begin{document}
   ...
   \maketitle
   \section*{Vorwort}
   \tableofcontents
   \listoffigures
   \listoftables
   \part{...} ...
   \chapter{...} ...
      \section{...} ...
   \part{...} ...
   \appendix
   \chapter{...} ...
   \chapter{...} ...
   \begin{thebibliography}...
\end{thebibliography}
   \begin{theindex}...\end{theindex}
\end{document}
```

besteht zusätzlich aus

- Vorwort, ohne Eintrag ins Inhaltsverzeichnis
- Inhaltsverzeichnis, Liste der Abbildungen und Liste der Tabellen
- Abschnitte, die durch **\part** gekennzeichnet sind
- einem Anhang, der bei **\appendix** beginnt
- einem Index (Stichwortverzeichnis)

Titelseite

- kann entweder mit
  \begin{titlepage}...\end{titlepage}
  und weiteren Formatierungsbefehle (z.B. `\centerline`) selbst gestaltet,
- oder in einem festen Format mit folgenden Befehlen definiert werden:
  `\title`{*Titelüberschrift*}
  - zentrierte Ausgabe
  - Umbruch mit Hilfe des \\-Befehls möglich

  `\author`{*Autorenname und ggf. deren Anschriften*}
  - mehrere Autoren können mit `\and` getrennt werden und erscheinen dann nebeneinander

  `\date`{*Datumstext*}
  - ist nicht zwingend erforderlich, wird auch automatisch eingefügt
  - Angabe eines Textes ersetzt aktuelles Datum
- mit `\maketitle` wird die Titelseite ausgegeben
- Beispiel:

```
\title{Erstellung wissenschaftlicher Texte
\\mit \LaTeX}
\author{Stephan Graf, Beate Herrmann\\
Zentralinstitut für Angewandte Mathematik}
\date{5.-7. Mai 1998}
\maketitle
```


Abstract

- wird mit folgenden Befehlen erzeugt:
  \begin{abstract}...**\end**{abstract}
- in Dokumentklasse **article**
  - beidseitig eingerückt
  - kleinere Schriftart
  - mit Überschrift **Abstract** oder **Zusammenfassung**
  - nicht auf einer eigenen Seite
- in Dokumentklasse **report**
  - normale Schrift
  - mit Überschrift **Abstract** oder **Zusammenfassung**
  - auf einer eigenen Seite ohne Seitennummer
- in Dokumentklasse **book**
  - nicht vorhanden, da unüblich

Inhaltsverzeichnis

- wird mit dem Befehl `\tableofcontents` eingefügt
- ggf. mehrere TEX-Durchläufe nötig
- Einträge werden in einer Datei *fn*`.toc` gesammelt und beim nächsten Durchlauf im Inhaltsverzeichnis gedruckt
- zusätzliche Eintragungen über die Befehle `\addcontentsline` und `\addtocontents`

Weitere Verzeichnisse

- werden mit den Befehlen `\listoffigures` und `\listoftables` eingefügt
- Einträge werden in den Dateien *fn*`.lof` und *fn*`.lot` gesammelt

Einteilung in Kapitel

- Befehle
  `\part (-1) \chapter (0) \subsection (2) \paragraph (4)`
  `\section (1) \subsubsection (3) \subparagraph (5)`
  (i) = Hierachiestufe
- Kapitel müssen korrekt geschachtelt sein
- Überschrift erhält fortlaufende Numerierung
- Überschrift von Unterkapiteln wird die Nummer der übergeordnete Kapitel vorangestellt
- bei `\chapter` zusätzliches Voranstellen von **Chapter** bzw. **Kapitel**
- `\part` ist nicht Teil der Hierarchie, verändert nicht die Numerierung der Kapitel
- `\chapter` ist in der Dokumentklasse **article** nicht vorhanden
- Stern-Form (z.B. `\chapter*`):
  - kein Eintrag ins Inhaltsverzeichnis
  - keine Numerierung der Überschrift
- Kurzform der Überschrift
  gliederungsbefehl[*kurzform*]{*Überschrift*}
  - bei langen Überschriften sinnvoll
  - *kurzform* wird für das Inhaltsverzeichnis und die Kopf- bzw. Fußzeile benutzt

Anhang

- wird durch den Befehl `\appendix` eingeleitet
  - Numerierung für Kapitel werden zurückgesetzt
  - Numerierung mit Großbuchstaben (A, B, C, ...)
  - bei dem Befehl `\chapter` wird nun **Appendix** bzw. **Anhang** vorangestellt

Einbinden von weiteren Paketen oder Styles

- über den Befehl
  \usepackage [*options*] {*package*}
  im Vorspann des Dokuments
- definieren z.B.
  - neue Befehle
  - Layoutänderungen
  - Änderungen bestehender Befehle
- *package* steht für den Paketnamen
- mehrere Pakete können mit Komma getrennt angegeben werden
- *options* sind Optionen, die für das zu ladende Paket gelten
- Optionen, die über den `\documentclass`-Befehl angegeben werden, sind *global* und werden zuerst interpretiert
- wichtige Pakete:

german	Erstellung deutscher Texte
alltt	Erweiterung für die Einbindung von Listings
float	bessere Behandlung von gleitenden Objekten
multicol	mehrspaltiger Text
longtable	mehrseitige Tabellen
graphicx	EPS-Graphiken
rotating	gedrehte Objekten

- Liste aller unter Unix verfügbaren Pakete:
  Kapitel 5 „LaTeX 2$_\varepsilon$ und die Ergänzungspakete" im BHB-0112
- Pakete sind auch in der emTeX-Installation enthalten

Schriftbild

Allgemeines

- in LaTeX 2$_\varepsilon$ neues Zeichensatzauswahlverfahren:
  NFSS: New Font Selection Scheme
- Attribute einer Schrift können einzeln und unabhängig voneinander gesetzt werden
- Auswahl der Schrift an Hand des zu formatierenden Elements
- Standardschrift: Roman
- TeX verwendet als Standard die Computer Modern Schriften
- Computer Modern Schriften sind nicht in allen Schriftarten vorhanden
- NFSS ist für die Zukunft entworfen und zur Einbindung von
  - anderen Schriften, die mit METAFONT entworfen sind,
  - PostScript-Schriften, die in der Größe flexibel sind,

  vorbereitet

Übersicht Auswahlverfahren

- Definition der Schriften über ihre Attribute:
  - Kodierung (encoding)
  - Schriftfamilie (family)
  - Form (shape)
  - Stärke (series)
  - Schriftgröße (fontsize)
- spezielle Befehle für vereinfachte Auswahl von Schriften
  - Schriftfamilie, Schriftstärke, Schriftform und Schriftgröße
- Font-Befehle aus dem alten LaTeX 2.09
  - Befehle definieren feste Schriftarten

Schriftattribute

- **Kodierung (encoding)**
  - Kodierungstabelle des Zeichensatzes
  - Festlegung, welches Zeichen bei einem bestimmten Hex-Code gedruckt wird
  - bei Computer Modern Schriften ist Kodierung immer gleich (128 Zeichen pro Zeichensatz)
  - z.B. PostScript-Schriften haben andere Kodierungen
  - DC-Schriften besitzen 256 Zeichen pro Zeichensatz, europäische Erweiterung (später EC-Schriften), bessere Trennung bei Umlauten
  - in LaTeX 2$_\varepsilon$ bekannte Kodierungen [L2b]

T1	erweiterte TeX-Zeichensätze (Cork, *dc*)
OT1	Standard TeX-Zeichensätze für Text (Computer Modern, *cm*)
OML	mathematische TeX-Zeichensätze für Text (*cmmi*)
OMS	mathematische TeX-Zeichensätze für Symbole (*cmsy*)
OMX	Erweiterung des math. TeX-Zeichensatzes (*cmex*)
U	unbekannte Kodierung

  - in Ergänzungspaketen benutzte Kodierungen [L2b]

OT2	kyrillische Zeichensätze
OT3	internationale phonetische Zeichensätze

  - eigene Kodierungen sollten mit **L** beginnen
  - LaTeX-Befehl: `\fontencoding`{*code*}

- **Schriftfamilie (family)**
  - Familie besteht aus verschiedenen Schriften, die aber das gleiche Design besitzen
  - einzelne Schriften unterscheiden sich innerhalb der Familie durch Fettschrift, Neigungsschrift usw.
  - Proportionalschrift:
    Zeichen erhalten nur den benötigten Platz $\boxed{\text{i}}$
    i braucht weniger Platz als m $\boxed{\text{m}}$
  - Nicht-Proportionalschrift:
    Zeichen erhalten alle den gleichen Platz $\boxed{\texttt{i}}$
    i ist genauso breit wie ein m $\boxed{\texttt{m}}$
  - Schrift mit Serifen
    Times Roman (Computer Modern Roman) $\boxed{\mathrm{R}}$
    Haken und Füßchen an den Kanten
    geeignet für längere Texte, da für das Auge besser lesbar
  - serifenlose Schrift
    Helvetica (Computer Modern Sans Serif) $\boxed{\mathsf{R}}$
    keine Haken und Füßchen
    besser geeignet für Plakate und Folien

- LaTeX-Befehl: `\fontfamily`{*family*}
- Familien der Computer Modern Schriften:

cmr	Computer Modern Roman, → Computer Modern Roman
cmss	Computer Modern Sans Serif, → Computer Modern Sans Serif
cmtt	Computer Modern Typewriter, → `Computer Modern Typewriter`
cmfib	Computer Modern Fibonacci, → Computer Modern Fibonacci
cmfr	Computer Modern Funny Roman, → Computer Modern Funny Roman
cmdh	Computer Modern Dunhill, → Computer Modern Dunhill
cmm	Computer Modern Math. Text
cmsy	Computer Modern Math. Symbol
cmex	Computer Modern Math. Extension
lasy	LaTeX-Zusatzsymbole

- **Stärke (series)**
  - besteht aus der Strichdicke (Stärke) und der Zeichenbreite (Weite)

Strichdicke:		Zeichenbreite:	
ul	Ultralight	**uc**	Ultracondensed (50 %)
el	Extralight	**ec**	Extracondensed (62.5 %)
l	Light	**c**	Condensed (75 %)
sl	Semilight	**sc**	Semicondensed (87.5 %)
m	Medium	**m**	Medium (100 %)
sb	Semibold	**sx**	Semiexpanded (112.5 %)
b	Bold	**x**	Expanded (125 %)
eb	Extrabold	**ex**	Extraexpanded (150 %)
ub	Ultrabold	**ux**	Ultraexpanded (200 %)

  - LaTeX-Befehl: `\fontseries`{*st_br*}
  - Beispiele:
    `\fontseries{lsc}` → Light und Semicondensed,
    `\fontseries{bx}` → **Bold und Expanded**
  - bei Computer Modern sind nicht alle Stärken und Weiten vorhanden, siehe Tabelle auf Folie 34
  - Bei der Paarung von **m** (Medium) mit einem anderen Attribut muß **m** entfallen:
    `\fontseries{b}` → Bold und Medium
    `\fontseries{m}` → Medium (Stärke) + Medium (Weite)

- **Form (shape)**
  - Merkmale:

n	(*normal*) aufrechte Form → aufrechte Schrift
it	(*italic*) kursive Form, hauptsächlich bei kursiven Schriften → *kursive Schrift*
sl	(*slanted*) schräge Form, hauptsächlich bei serifenlosen Schriften → *schräge Schrift*
sc	(*small caps*) Kapitälchen, Kleinbuchstaben werden als Großbuchstaben mit kleinerer Größe gedruckt → KAPITÄLCHEN-SCHRIFT
ui	(*upright italic*) ungeneigte kursive Schrift → ungeneigte kursive Schrift

  - kursive Schrift ist eingeständig entworfen und nicht nur eine schräge Normalschrift
  - LaTeX-Befehl: `\fontshape`{*shape*}
  - bei Computer Modern sind nicht alle Formen bei allen Schriftarten vorhanden, siehe Tabelle auf Folie 34

- **Schriftgröße (fontsize)**
  - LaTeX-Befehl: \fontsize{*größe*}{*z_abstand*}
  - Angabe kann als Zahlenangabe oder als Größenangabe erfolgen
  - Zahlenangaben erhalten die Einheit **pt** (Punkte)
  - erster Parameter *größe* → Schriftgröße
  - zweiter Parameter *z_abstand* → Zeilenabstand
  - Computer Modern Schriften sind nicht allen Größen vorhanden:
    5, 6, 7, 8, 9, 10, 10.85, 12, 14.4, 17.28, 20.48 und 24.88 (Vergrößerungfaktoren: 1.2^n)
  - vereinfachte Auswahlbefehle: angepaßte Größenbefehle
  - Beispiele:
    \fontsize{12pt}{14pt}
    \fontsize{10pt}{3ex}

Schriftauswahl über Attribute

- Einstellung der Schriftart mit den Befehlen:
  \fontencoding{*code*}
  \fontfamily{*family*}
  \fontseries{*st_br*}
  \fontshape{*shape*}
  \fontsize{*größe*}{*z_abstand*}
- Aktivierung der Schriftart mit: \selectfont
- Befehl ist eine Deklaration, gilt also bis zum Ende der Umgebung
- Attributveränderungen sind unabhängig
- Beispiel: {\fontshape{it}\selectfont}
  kursive Form der aktuellen Schrift
- Befehl für die gemeinsame Einstellung aller Werte:
  \usefont{*code*}{*family*}{*st_br*}{*shape*}
  - stellt Attribute ein und aktiviert die Schriftart
- falls Schriftart nicht vorhanden:
  - Suche einer Ersatz-Schriftart:
    → zuerst Suche nach Font in der gleichen Entwurfsgröße,
    → dann die in der Größe benachbarte Schriftart
  - Ausgabe einer Warnung:

```
LaTeX Font Warning: Font shape 'OT1/fcmss/m/n'
in size <18> not available
(Font) size <17.28> substituted on input line 264.
```


- tatsächlich vorhandene Schriftarten für die Computer Modern Schriften [L2b]:

Familie	Strichstärke	Form
cmr	m	n, it, sl, sc, ui
	bx	n, it, sl
	b	n
cmss	m	n, sl
	bx	n
	b	n
cmtt	m	n, it, sl, sc
cmfib	m	n
cmfr	m	n, it
cmdh	m	n, it

Vereinfachte Schriftauswahl

- Befehle der Form (Parameterform):
  \text*nn*{*...text...*}
  ändern Schriftfamilie, Schriftstärke und Schriftform
- Schriftfamilie:
  \textrm Roman-Schriftart (cmr)
  \textsf Sans Serif-Schriftart (cmss)
  \texttt Typewriter-Schriftart (cmtt)
- Schriftstärke:
  \textbf Bold Face, fette Schriftart
  \textmd Medium, normale Schriftstärke
- Schriftform:
  \textit kursive Schrift
  \textsl schräge Schrift
  \textsc Small Caps Schrift
  \textup aufrechte kursive Schrift
- Namen können auch für Umgebungen genutzt werden:
  \begin{textbf}
  ...
  \end{textbf}
- Deklarationen:
  - Schriftfamilie: \rmfamily, \sffamily, \ttfamily
  - Schriftstärke: \bfseries, \mdseries
  - Schriftform: \itshape, \slshape, \scshape, \upshape
- Hervorhebungen mit \emph{*...text...*}
  → schaltet je nach umgebenden Text auf kursive oder normale Schrift um (Deklaration: \em)

- Befehle für die Schriftgröße:

`\tiny`	winzige Schrift
`\scriptsize`	sehr kleine Schrift
`\footnotesize`	ziemlich kleine Schrift
`\small`	kleine Schrift
`\normalsize`	normale Schrift
`\large`	etwas größere Schrift
`\Large`	ziemlich große Schrift
`\LARGE`	große Schrift
`\huge`	sehr große Schrift
`\Huge`	riesige Schrift

  - Befehle sind Deklarationen und wirken bis zum Ende der aktuellen Umgebung
  - Änderung relativ zur Grundschrift
    → Optionen beim Befehl `\documentclass`:
    `10pt, 11pt, 12pt`
- Italic-Korrektur
  - beim Wechsel von kursiver nach normaler Schrift hängt letztes Zeichen über
  - z.B. *P*ause bzw. *P* ause
  - wird bei den `text`nn-Befehlen automatisch durchgeführt
  - ansonsten Befehl: `\/`

Fontbefehle aus LATEX 2.09

- definieren feste Schriftart
- Befehle bestehen aus zwei Buchstaben (`\bf`, `\it`, `\rm`, `\sc`, `\sl`, `\tt` und `\em`)
- sollten nicht mehr benutzt werden
- sind in den Dokumentklassen **article**, **report**, **book**, **letter** und **slides** noch definiert (gleiches Verhalten wie LATEX 2.09)
- können mit dem Erweiterungspaket **newlfonts** neu definiert werden:
  - verhalten sich dann wie die Deklarationen `\ttfamily`, ...

Umbruch

Konzept der „Boxen“

- Formatierer TEX betrachtet Zeichen als Boxen, die eine Breite, eine Höhe und eine Tiefe (Versatz zur Grundlinie) besitzen
- Formatierung:
  - Boxen der einzelnen Zeichen eines Wortes werden zu einer neuen Box zusammengefügt
  - Boxen der Worte werden zu einer Zeilenbox (LR-Box) zusammengefügt
    → Abstand zwischen den Worten ist in gewissen Grenzen variabel (Klebstoff, *glue*)
  - Zeilenboxen werden zu einer Absatzbox zusammengefügt
  - Absatzboxen werden zu Rumpfbox zusammengefügt
  - Seitenbox wird aus Kopf-, Fuß- und Rumpfbox erstellt
- Zeilenboxen werden für Blocksatz gedehnt oder gestaucht
  - variable Abstände zwischen den Worten nur in gewissen Grenzen veränderbar, ansonsten:
    - Warnung: `Overfull hbox` falls Wortabstände zu gering
    - Warnung: `Underfull hbox` falls Wortabstände zu groß
  - Wörter werden ggf. getrennt
  - Warnungen kommen relativ häufig vor
    - Befehl `\sloppy` veranlaßt TEX zu einer weniger genauen Behandlung
    - Markierungen am Seitenrand durch den Befehl `\setlength{\overfullrule}{3pt}`

Absätze

- sind zusammenhängende Textblöcke
- werden in der Eingabedatei
  - mit Leerzeilen oder
  - mit dem Befehl `\par`

  voneinander getrennt
- mehrere Leerzeilen werden in der Eingabe zu einer zusammengefaßt
- Absatzabstand über `\parskip`:
  `\setlength{\parskip}{1.5ex plus0.5ex minus0.5ex}`
- erste Zeile eine Absatzes wird eingerückt (Default)
- Einrücken der ersten Zeile eines Absatzes kann mit
  `\setlength{\parindent}{0pt}`
  verändert werden

Trennung

- deutsches Trennwörterbuch bei der Verwendung der Erweiterungspaketes **german**
- Trennung bei Silben mit Umlauten nicht bei **OT1**-Kodierung
  - Kodierung **T1**: 256 Zeichen pro Zeichensatz, Umlaute als eigenständige Zeichen
  - Aktivierung mit `\usepackage{t1enc}`
- Trennvorschläge
  - im Text mit dem Befehl `\-`
    - z.B. `Druck\-erzeugnis`, `Drucker\-zeugnis`
    - nur die markierten Stellen sind als Trennstellen zugelassen
  - Befehl `"-` aus dem Erweiterungspaket **german**
    - Stelle wird zusätzlich zu den bisher bekannten Stellen definiert
    - z.B. `Ur"-instinkt`, `be"-inhalten`
      → Ur-in-stinkt, be-in-hal-ten
- Wörter, die mehrmals auftreten
  - Befehl `\hyphenation`{*Trennliste*} im Vorspann des Dokuments
  - z.B. `\hyphenation{Ur-in-stinkt be-in-hal-ten}`
  - keine Angabe von Sonderzeichen, daher keine Umlaute
  - kann in einer Datei als eigenes Trennlexikon abgespeichert und mit `\input` wieder eingeladen werden

- Erweiterungspaket **german**:
  `"ff, "ll, "mm, "nn, "pp, "tt, "ck`
  `Dru"cker` → Druk-ker, `Be"ttuch` → Bett-tuch
- Anzeigen der Trennstellen (nur in der Bildschirmausgabe)
  - Befehl `\showhyphens`{*Wortliste*}
- Vermeidung von Trennungen
  - festes Leerzeichen: `~`
    z.B. `Rio~de~Janeiro`
    keine Trennung am Leerzeichen
  - `\mbox`{*Wort*} verhindert Trennung des Wortes
  - trotzdem Trennung kontrollieren

Zeilenumbruch

- explizites Beenden einer Zeile mit den Befehlen
  `\\` und `\newline`
  - Zeile wird beendet und Worte werden linksbündig gesetzt
  - Angabe eines zusätzlichen Abstands zum Zeilenwechsel:
    `\\[`*abstand*`]`
    `\\*[`*abstand*`]`
  - Abstand wird am Seitenende unterdrückt, bei der Stern-Variante nicht
  - Beispiel:

```
erste Zeile\\
zweite Zeile\\[1cm]
dritte Zeile\\
```

- Befehl `\linebreak[`num`]`
  - erschwert oder erleichtert den Zeilenumbruch
  - Befehl ohne Parameter erzwingt den Umbruch
  - num: ganze Zahl zwischen 0 und 4, Empfehlung an den Formatierer:
    0 → nicht dringend, 4 → Umbruch wird erzwungen
  - Zeile wird beendet und in Blocksatz gesetzt
- Befehl `\nolinebreak[`num`]`
  - entgegengesetzte Wirkung
  - Befehl ohne Parameter verhindert den Umbruch
  - num: ganze Zahl zwischen 0 und 4, Empfehlung an den Formatierer:
    0 → nicht dringend, 4 → Umbruch wird verhindert

Seitenumbruch

- explizites Beenden der Seite mit den Befehl `\newpage`
  - Seite wird umgebrochen, Rest der Seite bleibt leer
- Befehle `\pagebreak` und `\nopagebreak` entsprechen den Befehlen `\linebreak` und `\nolinebreak`
- Befehl `\pagebreak[`num`]`
  - erschwert oder erleichtert den Seitenwechsel
  - Befehl ohne Parameter erzwingt den Seitenwechsel
  - num: ganze Zahl zwischen 0 und 4, Empfehlung an den Formatierer:
    0 → nicht dringend, 4 → Seitenwechsel wird erzwungen
  - Seite wird beendet und der Abstand der Paragraphen so weit erhöht, bis die Seite von oben bis unten gefüllt ist
- Befehl `\nopagebreak[`num`]`
  - entgegengesetzte Wirkung
  - Befehl ohne Parameter verhindert den Seitenwechsel
  - num: ganze Zahl zwischen 0 und 4, Empfehlung an den Formatierer:
    0 → nicht dringend, 4 → Seitenwechsel wird verhindert
- `\clearpage`
  - Seite wird beendet, Rest der Seite bleibt frei
  - alle noch ausstehenden gleitenden Objekte werden ausgegeben

- `\cleardoublepage`
  - Dokumentklassen-Option `twoside`
  - `\cleardoublepage` bewirkt einen Seitenvorschub auf die nächste Seite mit ungerader Seitennummer
- zweispaltiger Druck
  - Dokumentklassen-Option `twocolumn`
  - `\pagebreak` und `\newpage` wirken auf die aktuelle Spalte
  - `\clearpage` und `\cleardoublepage` wirken auf die aktuelle Seite
- Seitenwechsel unterdrücken
  - Deklaration `\samepage`:
    Wirkung bis zum Ende der Umgebung
  - kann auch als Umgebung benutzt werden:
    ```
    \begin{samepage}
        ...
    \end{samepage}
    ```
  - Wirkung:
    - Seitenwechsel nur zwischen Absätzen
    - kein Seitenwechsel vor oder hinter Formeln und Einrückungen

Zeichen- und Wortabstände

- Zwischenraum nach Satzende
  - Punkt, Ausrufezeichen, Fragezeichen und Doppelpunkt kennzeichnen ein Satzende
  - TEX fügt größeren Zwischenraum ein
  - unerwünschter Zwischenraum: z.B. bei Dr. Schmidt
  - Unterdrückung mit den Befehlen `\␣` und `~`
  - `~` verhindert zusätzlich die Trennung
  - Satzendezeichen hinter Großbuchstaben werden nicht erkannt, kann aber durch Voranstellen von `\@` erzwungen werden: `NASA\@.`
- Frenchspacing
  - Unterdrücken des Zwischenraums am Satzende mit: `\frenchspacing`
  - Deklaration, wirkt also bis zum Ende der aktuellen Umgebung
  - kann auch im Vorspann genutzt werden
  - `\nonfrenchspacing` hebt Wirkung auf
  - bei Erweiterungspaket **german** ist `\frenchspacing` gesetzt

- Ligaturen
  - TEX kennt die Ligaturen: ff, fi, fl, ffi und ffl
  - z.B. Auflage, Auflage
  - können mit Befehl `\/` verhindert werden:
    z.B. `Auf\/lage`

Horizontale und vertikale Abstände

- horizontale Abstände in einer Zeile
  - Befehle
    `\hspace{`*abstand*`}`
    `\hspace*{`*abstand*`}`
  - *abstand* ist eine Längenangabe (z.B. 1.3 cm)
  - fügt an der aktuellen Stelle in der Zeile entsprechenden Zwischenraum ein
  - Stern-Form fügt den Zwischenraum auch am Zeilenende oder Zeilenanfang ein
  - Längenangabe darf negativ sein, Überdrucken von Zeichen möglich
  - Leerzeichen vor und hinter dem Befehl werden zusätzlich zum Abstand eingefügt
  - Befehl `\hfill` entspricht `\hspace{\fill}`
    - fügt Zwischenraum ein, so daß Zeile aufgefüllt wird
    - bei mehreren `\hfill` wird der Zwischenraum gleich aufgeteilt
  - feste Abstände:
    - `\quad`, Abstand, der der Schrifthöhe entspricht
    - `\qquad`, Abstand, der der doppelten Schrifthöhe entspricht

- vertikale Abstände
  - Befehle
    `\vspace{`*abstand*`}`
    `\vspace*{`*abstand*`}`
  - *abstand* ist eine Längenangabe (z.B. 1.3 cm)
  - fügt an der aktuellen Stelle in der Zeile entsprechenden vertikalen Zwischenraum ein
  - Stern-Form fügt den Zwischenraum auch bei einem Seitenwechsel ein
  - innerhalb eines Absatzes wird die aktuelle Zeile aufgefüllt und der Absatz beendet
  - Längenangabe darf negativ sein, Höherrücken von Text möglich
  - Befehl `\vfill` entspricht `\vspace{\fill}`
    - fügt Zwischenraum ein, so daß die Seite aufgefüllt wird
    - bei mehreren `\vfill` wird der Zwischenraum gleich aufgeteilt
  - elastische vertikale Abstände möglich:
    z.B. `\vspace{1cm minus5mm plus 4mm}`
  - feste Abstände:
    - `\bigskip`, `\medskip`, `\smallskip`
    - fügen vertikalen Abstand ein

Zentrierter Text

- Umgebung
  `\begin{center}`
  Zeile1`\\`*Zeile2*`\\`*Zeile3*
  `\end{center}`
- Text wird zeilenweise zentriert
- ohne `\\`:
  - Zeilen werden mit festen Wortabständen aufgefüllt und dann zentriert
  - keine Worttrennung
- vor und hinter der Umgebung wird vertikaler Zwischenraum eingefügt
- Deklaration `\centering` wirkt bis zum Ende der aktuellen Umgebung
- Befehl `\centerline{`*text*`}`
  - zentriert eine Textzeile
  - kein zusätzlicher vertikaler Abstand

Einseitig bündiger Text

- Umgebungen
  `\begin{flushleft}` *Zeile1*`\\`*Zeile2*`\\`*Zeile3*
  `\end{flushleft}`
  `\begin{flushright}` *Zeile1*`\\`*Zeile2*`\\`*Zeile3*
  `\end{flushright}`
- formatieren Text links- bzw. rechtsbündig
- ohne `\\`:
  - Zeilen werden mit festen Wortabständen aufgefüllt und dann links- bzw. rechtsbündig formatiert
  - keine Worttrennung
- Deklarationen
  - `\raggedright` (linksbündiger Text)
  - `\raggedleft` (rechtsbündiger Text)

  wirken bis zum Ende der aktuellen Umgebung

Einrückungen

- beidseitig eingerückter Text
  - Umgebungen
    `\begin{quote}` *Text* `\end{quote}`
    `\begin{quotation}` *Text* `\end{quotation}`
  - Text wird im Blocksatz formatiert
  - `quotation`: erste Zeile des Absatzes wird eingerückt
  - `quote`: Absatz erhält zusätzlich vertikalen Abstand
- Verseinrückungen
  - Umgebung: `\begin{verse}` *Gedicht* `\end{verse}`
- Beispiel:

> Text, der nicht in einer quote-Umgebung steht, wird auf die normale Spaltenbreite formatiert.
>
> > Dies ist ein langer relativ sinnloser Text, der zeigt, wie eine quote-Umgebung aussieht. Damit er nicht zu lang wird ist dieser Satz der letzte.
>
> Und zum Schluß wieder ein normal breiter Satz. Die quote-Umgebung schließt an den normalen Text mit einem gewissen vertikalen Abstand an.

Regelsätze

- Textstrukturen wie:

  Satz 1. [Bolzano-Weierstraß] *Jede beschränkte unendliche Punktmenge besitzt mindestens einen Häufungspunkt*

  Axiom 4.1. *Die natürlichen Zahlen bilden eine Menge Z von unterschiedlichsten Elementen . . .*
- Struktur: Schlüsselbegriff, fortlaufende Nummer in Fettdruck, Text in *Kursiv*
- Befehl:
  `\newtheorem`{*strukt_name*}{*struktur_begriff*}[*zusatz_zaehler*]
  definiert neue Struktur
- *strukt_name*: beliebiger Name, unter dem die Umgebung für die Struktur aufgerufen wird
- *strukt_begriff*: Wort in Fettdruck
- *zusatz_zaehler*: meistens Gliederungsname (`chapter`), der für das Zurücksetzen und die vorangestellte Nummer dient
- Beispiele (obige Strukturen):

```
\newtheorem{satz}{Satz}
\newtheorem{axiom}{Axiom}[chapter]
\begin{satz}[Bolzano-Weierstra"s]
  Jede beschr"ankte unendliche Punktmenge besitzt
  mindestens einen  H"aufungspunkt
\end{satz}
\begin{axiom}
  Die nat"urlichen Zahlen bilden eine Menge Z von
  unterschiedlichsten Elementen \ldots
\end{axiom}
```


Nicht formatierter Text

- Ausdruck von Originaltext oder Computerlistings
- Umgebungen:
  `\begin{verbatim}` *Originaltext* `\end{verbatim}`
  `\begin{verbatim*}` *Originaltext* `\end{verbatim*}`
- Stern-Form ersetzt Leerzeichen durch ␣
- Beispiel:

```
\begin{verbatim}
if($level==$prev_level)
{ ... }
\end{verbatim}
```

```
if ($level == $prev_level)
{ &write($level,$y,$text);
  $pos[$level] .= " $y";
  $bottom[$level] = $y;
  $y -= FONT_SIZE; }
```

- Befehl `\verb` für kurze Textstücke
  - Syntax:
    `\verb`*zeichen Originaltext zeichen*
    `\verb*`*zeichen Originaltext zeichen*
- erstes Zeichen hinter dem Befehl dient als Abgrenzung zum übrigen Text
- Beispiel:
  `\verb+\begin{verbatim}+` → `\begin{verbatim}`
- Umgebung: `\begin{alltt}` *Originaltext* `\end{alltt}`
  - läßt LaTeX-Befehle innerhalb des Originaltextes zu
  - Erweiterungspaket muß mit `\usepackage{alltt}` geladen werden

Textanmerkungen

Fußnoten

- werden im Text mit einem Verweis gekennzeichnet
- Text der Fußnote wird am Seitenende abgesetzt und in kleinerer Schrift formatiert
- Befehl `\footnote`{*fußnotentext*}
- Marker werden fortlaufend numeriert
  - bei **article** für das ganze Dokument
  - bei **report** und **book** beginnt Numerierung bei jedem Kapitel neu
- Beispiele:

```
Dies ist ein langer
Text \footnote{und das
eine Fu"snote}.
```

Dies ist ein langer Text [1].

```
Dies ist ein langer
Text \footnote{und das
auch eine Fu"snote}.
```

Dies ist ein langer Text [2].

- Fußnoten in unerlaubten Modi
  - z.B. in LR-Boxen, Tabellen und mathematischen Formeln
  - mit `\footnotemark`[*num*] wird nur der Marker gesetzt
  - mit `\footnotetext`[*num*]{*fußnotentext*} wird dann der Text definiert
  - *num*: optionale Angabe der Fußnotennummer
- Modifizieren der Fußnotennummer:
  `\footnotesymbol` o.ä.

[1] und das eine Fußnote
[2] und das auch eine Fußnote

Randnotizen (Marginalien)

- werden mit
  `\marginpar`{*randnotiz*}
  erzeugt
- Text erscheint am rechten Rand der Seite
- bei der Dokumentklassenoption `twoside` am äußeren Rand
- beginnt in der Höhe der Zeile, in der sie definiert wird   Dies ist eine Randnotiz
- Beispiel:
  `beginnt in der Höhe \marginpar{Dies ist eine Randnotiz} der Zeile ...`
- kann mit dem Befehl `\rule[-10mm]{1mm}{15mm}` auch für Markierungen neuer Textpassagen benutzt werden
- wichtige Punkte können auch mit einem Pfeil markiert werden: ⟸
  - Problem bei doppelseitigem Druck (Richtung des Pfeils)
  - optionaler Parameter:
    `\marginpar`[*l_randnotiz*]{*r_randnotiz*}
    getrennte Definition für eine linke und eine rechte Seite

Querverweise

- beliebige Stellen im Dokument können mit einen **Label** versehen werden
- Label: Name, mit dem ein Bezug zu der Seite oder dem Kapitel erstellt werden kann
- Befehl `\label`{*markierung*} setzt Label
- Befehl `\pageref`{*markierung*} erzeugt die Seitennummer der markierten Textstelle
- Beispiel:

```
Erweiterungspaket varioref \label{erwref} ...
Eine Erweiterung der Funktionalität ist mit
dem Paket \textbf{varioref} gegeben (siehe Folie
\pageref{erwref})
```

- falls `\label`-Befehl hinter der Referenzierung auftritt:
  - zwei TEX-Durchläufe benötigt
  - nach dem ersten Durchlauf ist Referenz unbekannt und wird durch `??` ersetzt
- Befehl `\ref`{*markierung*} erzeugt die Kapitelnummer (o.ä.) der markierten Textstelle
  - `\label`-Befehl muß hinter einem Gliederungsbefehl oder in einer `equation-`, `eqnarray-`, `figure-`, `table-` oder `enumerate-`Umgebung stehen
- Eine Erweiterung der Funktionalität ist mit dem Paket **varioref** gegeben (siehe Folie 57)

- Erweiterungspaket **varioref**
  - erzeugt Verweise der Art: Siehe Formel 2 auf Seite 97
  - faßt in der Referenz Nummer des Elementes und Seitennummer zusammen
  - Seitennummer wird nur bei entfernten Seiten angegeben
  - Verweise auf vorhergehende oder nachfolgende Seite erfolgen direkt
  - keine Seitennummer, falls Label auf gleicher Seite
  - Befehl: `\vref`{*markierung*}
  - Beispiel:
    `Siehe Formel \vref{eqnarray} ...`
  - Option `german` beim Laden angeben:
    `\usepackage[german]{varioref}`

Literaturverweise

- Erstellung eines Literaturverzeichnis:
  `\begin`{`thebibliography`}{*muster_marke*}
  `\bibitem`[*marke_1*]{*bezug_1*}*eintrag1*
  `\bibitem`[*marke_2*]{*bezug_2*}*eintrag2*
  ...
  `\end`{`thebibliography`}
- Referenz auf einzelne Einträge mit:
  `\cite`{*bezug*}
- Beispiel: `\cite{kopkab}` $\rightarrow$[L2b]

```
\begin{thebibliography}{999}
\bibitem[L1]{lamport} Leslie Lamport. \LaTeX\ --
    A Document Preparation System, Addison--Wesley Co.,
    Inc., Reading, MA, 1985
\bibitem[L2]{kopka} Helmut Kopka. \LaTeX, Band 1--3,
    Addison Wesley (Deutschland) GmbH, Bonn, 1993
\bibitem[L2a]{kopkaa}Band 1:
  \LaTeX-\textit{Einf"uhrung}, 1993
\bibitem[L2b]{kopkab}Band 2:
  \LaTeX-\textit{Erg"anzungen}, 1994
\bibitem[L2c]{kopkac}Band 3:
  \LaTeX-\textit{Erweiterungen ---  mit einer
    Einf"uhrung in \MF}, 1994
\bibitem[L3]{comp}M. Goosens, F. Mittelbach, A. Samarin,
    The \LaTeX\ Companion, Addison--Wesley Co., Inc.,
    Reading, MA, 1994
\end{thebibliography}
```


Listen

Aufzählungen

- Umgebungen
  `\begin`{`itemize`} *Text* `\end`{`itemize`}
  `\begin`{`enumerate`} *Text* `\end`{`enumerate`}
  `\begin`{`description`} *Text* `\end`{`description`}
- Text wird nach rechts eingerückt
- Befehl `\item` leitet Eintrag ein
- Eintrag erhält eine Markierung
- Beispiel:

```
...
\begin{itemize}
 \item Text wird nach links einger"uckt
 \item Befehl \texttt{$\backslash$item}
   leitet ...
 \item Eintrag erh"alt eine Markierung
\end{itemize}
...
```

- `itemize`: Einträge werden durch Symbole markiert
- `enumerate`: Einträge werden durch laufende Nummer markiert
- `description`: Einträge werden durch Schlüsselwörter markiert

Schachtelung

- Listen können bis zu einer Tiefe von 4 beliebig (wechselseitig) geschachtelt werden
- Tiefe wird für jeden Listentyp einzeln gezählt
- Einrücktiefe und Markierung ist von der Tiefe abhängig
- Beispiel:

> - Erstes Element der obersten Liste
>   1. Erstes Element einer numerierten Liste
>   - Erstes Element der dritten Liste
>   - Zweites Element der dritten Liste
>   (a) Erstes Element der untersten Liste
>   (b) Zweites Element der untersten Liste
>   - Drittes Element der dritten Liste
>   2. Zweites Element einer numerierten Liste
> - Zweites Element der obersten Liste
> - Drittes Element der obersten Liste

- Elemente einer numerierten Liste können mit dem `\label`-Befehl referenziert werden

Defintionslisten

- Schlüsselwörter werden als optionaler Parameter beim `\item`-Befehl eingegeben
  `\item`[*schlüsselwort*] *text ...*
- falls Schlüsselwort länger als Einrücktiefe:
  - erste Zeile der Beschreibung wird entsprechend eingerückt
- Beispiel:

> **Begriff** Beschreibung des ersten Elements der Definitionsliste
> **Langer *Begriff*** Zweites Element der obersten Liste, Begriff ist teilweise kursiv geschrieben. Die erste Zeile dieser Beschreibung wird eingerückt.
> **xyz** Drittes Element der obersten Liste
>   **Schachtelung** ist möglich
>   **zweites** Element der dieser Liste
> **xyz** Drittes Element der obersten Liste

```
\begin{description}
\item[Begriff] Beschreibung des ...
\item[Langer \textit{Begriff}] Zweites ..
\item[xyz] Drittes Element der obersten Liste
  \begin{description}
  \item [Schachtelung] ist m"oglich
  \item [zweites]  Element der obersten Liste
  \end{description}
\item[xyz] Drittes Element der obersten Liste
\end{description}
```


Erweiterte Definitionsliste

- Erweiterungspaket **expdlist**
  - optionaler Parameter
    `\begin{description}`[*dekl*] *Text* `\end{description}`
  - Deklarationen:
    - `\setleftmargin{`*länge*`}` verändert die Einrücktiefe der Liste
    - `\setlabelphantom{`*text*`}` berechnet die Einrücktiefe aus der Länge von *text*
    - `\breaklabel` Text beginnt erst in der nächsten Zeile, falls Schlüsselwort zu lang
    - `\compact` Einträge werden nicht durch Leerzeilen getrennt
    - `\setlabelstyle{`*schriftart*`}` definiert eine Schriftart, in der die Schlüsselwörter gedruckt werden (z.B. `\bfseries`)
  - Beispiel:

> *test1* erste Zeile
> *langer Testbegriff* zweite Zeile, sehr lang
> *Begriff* dritter Zeile

```
\begin{description}[\setlabelstyle{\itshape}
  \compact\setleftmargin{2cm}]
\item[test1] erste Zeile
\item[langer Testbegriff] zweite Zeile,
  sehr lang
\item[Begriff] dritter Zeile
\end{description}
```


Layoutänderungen

- Marker
  - können beim einzelnen Eintrag über den optionalen Parameter eingegeben werden
  - können über folgenden Befehl für jede Tiefe neu definiert werden:
    `\renewcommand{\label`*typtiefe*`}{`*wert*`}`
  - `itemize`-Liste:
    `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`
    Beispiel: `\renewcommand{\labelitemiii}{+}`
  - `enumerate`-Liste:
    `\labelenumi`, `\labelenumii`, `\labelenumiii`, `\labelenumiv`
    Beispiel:
    `\renewcommand{\labelenumiii}{\arabic{enumii}}`
- Abstände (*Auswahl*)
  - zu setzen mit: `\setlength{\`*befehlwert*`}{`*länge*`}`
  - `\leftmargin`: Einrücktiefe
  - `\topsep`: vertikaler Abstand zum vorhergenden und nachfolgenden Text
  - `\itemsep`: Abstand zweier Listeneinträgen
- LaTeX verfügt über einen allgemeinen Listentyp
  - kann als Basis für eigene Listentypen genutzt werden
  - siehe z.B. Kopka-Buchreihe [L2a]

Tabulatoren und Tabellen

- Tabulatoren
  - Definition fester horizontaler Positionen
  - von der Schreibmaschine her bekannt
- Tabellen
  - enthalten Rahmen
  - horizontale und vertikale Linien
  - Zusammenfassung mehrerer Spalten

Verbrauch	1/94	2/94	3/94	4/94
Farbe	10 l	12 l	3 l	9 l
Pinsel	2	4	1	4
Eimer	1	2	2	1

Zahlendarstellung		
Zahl 1984	hexadezimal	7C0
	octal	3700
	binär	11111000000

Tabulatoren

- Umgebung:
  \begin{tabbing}
  Zeilen
  \end{tabbing}
- Tabulatoren sind nur in dieser Umgebung vorhanden
- Tabulatorstops werden[3]
  - von links nach rechts numeriert
  - mit \= definiert
  - mit \> angewählt
- Zeilen müssen mit \\ beendet werden
- Beispiel:

```
\begin{tabbing}
Verbrauch \=1/94 \=2/94 \=3/94 \=4/94 \\[5mm]
Farbe     \>10 l \>12 l \>3 l  \>9 l\\
Pinsel    \>2    \>4    \>1    \>4\\
Eimer     \>1    \>2    \>2    \>1\\
\end{tabbing}
```

Verbrauch	1/94	2/94	3/94	4/94
Farbe	10 l	12 l	3 l	9 l
Pinsel	2	4	1	4
Eimer	1	2	2	1

[3] eigentlich Tabulatorstopps

- Musterzeile
  - dient zum Setzen der Tabulatoren
  - wird nicht im Dokument gedruckt
  - anstatt \\ Befehl \kill
  - sollte die jeweils breiteste Eintragung einer Spalte enthalten
  - Tabstops nach bestimmten festen Längen mit dem Befehl \hspace
- Beispiel:

```
\begin{tabbing}
\hspace*{2cm}\=\hspace{3cm}\=\hspace{2cm}\=
                   \hspace{2cm}\=\hspace{2cm}\=\kill
\>Verbrauch \>1/94 \>2/94 \>3/94 \>4/94 \\[5mm]
\>Farbe     \>10 l \>12 l \>3 l  \>9 l\\
\>Pinsel    \>2    \>4    \>1    \>4\\
\>Eimer     \>1    \>2    \>2    \>1
\end{tabbing}
```

Verbrauch	1/94	2/94	3/94	4/94
Farbe	10 l	12 l	3 l	9 l
Pinsel	2	4	1	4
Eimer	1	2	2	1

- linker Rand
  - Rand der Umgebung, nullter Tabstop
  - wird für den Text *vor* dem ersten Tabstop benutzt
  - kann mit \+ dauerhaft auf den nächsten Tabstop gesetzt
  - und mit \- zurückgesetzt werden
  - \+ und \- können mehrfach benutzt werden
  - \< setzt den linken Rand nur für eine einzelne Zeile zurück
- Beispiel:

```
\begin{tabbing}
\hspace*{2.5cm}\=\hspace{2.5cm}\=\hspace{2.5cm}\=
                   \hspace{2.5cm}\=\hspace{2.5cm}\=\kill
linker Rand \>1 \>b   \\
\+linker Rand \>2 \>b \\
linker Rand \>3 \>b   \\
\+linker Rand \>4 \>b \\
\-linker Rand \>5 \>b \\
linker Rand \>6 \>b   \\
\-linker Rand \>7 \>b \\
linker Rand \>8 \>b
\end{tabbing}
```

```
linker Rand 1          b
linker Rand 2          b
            linker Rand 3          b
            linker Rand 4          b
                        linker Rand 5          b
            linker Rand 6          b
            linker Rand 7          b
linker Rand 8          b
```

- Bemerkungen
  - \> geht zum nächsten logischen Tabstop
  - ggf. werden Texte überschrieben
- Beispiel:

```
\begin{tabbing}
 erste Zelle           \= zweite Zelle\\
 Wort                  \> Wort\\
 Zelle mit viel Inhalt \> kurzes Wort
\end{tabbing}
```

```
erste Zelle zweite Zelle
Wort        Wort
Zelle mit vikurzes Wort
```

- neue Definition von Tabulatoren
  - ist in jeder Zeile möglich
  - \= definiert den nächsten logische Tabulator neu
  - \= hinter letztem Tabulator fügt neuen Tabulator hinzu
- Befehle `\pushtabs` und `\poptabs`
  - speichern aktuelle Tabulatordefinitionen auf einem Stack und rufen diese wieder ab
  - sinnvoll bei zwischenzeitlicher Umdefinition
  - können mehrfach auftreten
  - Anzahl von `\pushtabs` und `\poptabs` müssen am Ende der Umgebung ausgeglichen sein
- Befehle für Seitenwechsel sind in der Umgebung nicht erlaubt
  - Trick: Verwendung eines großen vertikalen Vorschubs: `\\[10cm]`

- Weitere Tabulatorbefehle
  - Befehl \'
    - Syntax: *ltext*\'*rtext*
    - *ltext* wird mit einem kleinen Abstand vor der aktuellen Tabulator gesetzt
    - *rtext* wird an die Position des aktuellen Tabstops gesetzt
    - setzen des Abstand mit Befehl: `\setlength{\tabbingsep}{5mm}`
  - Befehl \`
    - Syntax: \`*text*
    - *text* wird rechtsbündig zum rechten Rand der `tabbing`-Umgebung gesetzt
    - danach darf in dieser Zeile kein weiterer Tabulator benutzt werden

- größeres Beispiel [L2a]

```
\begin{tabbing}
G"urteltier \=    \kill
M"ucken:\>erschlagen von: \=Menschen \+\+ \\
  K"uhen\\
  und\' Pferden \- \\
  wenig s"attigend \- \\
G"urteltier:\> nicht genie"sbar\\
\pushtabs
(siehe auch: \= Ameisenb"ar\\
  \> Albatros \` gesch"utzt)\\
\poptabs
Pferde \> gefressen von \> M"ucken
\end{tabbing}
```

```
Mücken:  erschlagen von: Menschen
                         Kühen
                     und Pferden
         wenig sättigend
Gürteltier:nicht genießbar
(siehe auch: Ameisenbär
             Albatros                  geschützt)
Pferde   gefressen von   Mücken
```


Tabellen

- drei Umgebungen:
  - `tabular`-Umgebung
    `\begin{tabular}`[*pos*]{*sp_form*}
    Zeilen
    `\end{tabular}`
    Breite der Tabelle richtet sich nach deren Inhalt
  - `tabular*`-Umgebung
    `\begin{tabular*}`{*breite*}[*pos*]{*sp_form*}
    Zeilen
    `\end{tabular*}`
    Breite der Tabelle wird als Parameter angegeben
  - `array`-Umgebung
    `\begin{array}`[*pos*]{*sp_form*}
    Zeilen
    `\end{array}`
    Verwendung nur in mathematischer Umgebung
- Umgebung wird von TEX wie eine Minipage behandelt, kann also auch wieder Tabellen enthalten
- erstes Beispiel

```
\begin{tabular}[t]{|c|c|}
  \hline
  a&b\\
  \hline
  c&d\\
  \hline
\end{tabular}
```

a	b
c	d

- Parameter *pos*
  - vertikale Positionierung
    - **t** Ausrichtung der obersten Tabellenzeile auf die laufende Umgebung
    - **b** Ausrichtung der untersten Tabellenzeile auf die laufende Umgebung
    - **c** Ausrichtung der Tabellenmitte auf die laufende Umgebung (Default bei keiner Angabe von *pos*)
- Parameter *breite*
  - nur bei der `tabular*`-Umgebung definiert
  - bestimmt die Gesamtbreite der Tabelle
  - *sp_form* sollte am Anfang `@{\extracolsep{\fill}}` enthalten (Spalten werden entsprechend aufgeweitet)
- Parameter *sp_form*
  - bestimmt die Aufteilung einer Tabellenzeile
  - für jede Spalte ein Formatierungseintrag
  - zusätzlich Einträge für die Ränder und den Spaltenzwischenraum
  - Einträge für Spalteninformationen:
    - **l** Inhalt der Spalte wird linksbündig formatiert
    - **r** Inhalt der Spalte wird rechtsbündig formatiert
    - **c** Inhalt der Spalte wird zentriert
    - **p{*br*}** Spalte erhält die Breite *br*, Zeilen werden als Paragraph formatiert, oberste Zeile wird auf die anderen Spalten ausgerichtet
  - bei **l,r,c**: Inhalt der Zelle wird **nicht** umgebrochen
    mehrzeiliger Text → **p{*br*}**

  - Formatierungszeichen für Ränder und Spaltenzwischenraum
    - | erzeugt eine vertikale Linie
    - || erzeugt zwei dicht nebeneinanderliegende vertikale Linien
    - **@{*text*}** *text* wird in jeder Zeile zwischen den davor und danach definierten Spalten eingefügt
      - ersetzt den normalen Zwischenraum
      - Befehl `\extracolsep` definiert zusätzlichen Zwischenraum für alle nachfolgenden Spalten
  - Wiederholung von Definitionen
    - Syntax: `*{`*num*`}{`*sp_form*`}`
    - *num*: Anzahl der Wiederholungen
    - *sp_form*: Spaltenformatierung
    - Beispiel:
      `*{5}{|c}|` enspricht `|c|c|c|c|`
- Struktur der *Zeilen*
  - werden durch \\ voneinander getrennt
  - Spalteneinträge werden durch & voneinander getrennt
  - Anzahl der Spalten muß mit der Anzahl der Spaltendefinitionen übereinstimmen
  - Text eines Spalteneintrags wird so behandelt als wäre er geklammert { }, Deklarationen gelten nur für die Zelle

- Befehle innerhalb der Zeilen
  - `\hline` erzeugt eine horizontale Linie über die Breite der Tabelle
    - darf nur am Anfang der Tabelle oder direkt hinter \\ stehen
    - zwei aufeinanderfolgende `hline`-Befehle erzeugen Doppellinie
  - `\cline{`***n - m***`}` erzeugt horizontale Linie vom linken Rand der Spalte *n* bis zum rechten Rand der Spalte *m*
    - darf nur am Anfang der Tabelle oder direkt hinter \\ stehen
    - mehrere `cline`-Befehle dürfen aufeinanderfolgen
  - `\multicolumn{`***num***`}{`***sp***`}{`***text***`}` faßt die nächsten *num* Spalten zu einer Zelle zusammen
    - *sp* enthält eine Spaltendefinition, die die neue Zelle beschreibt
    - mit *num*=1 kann das Layout einer bestimmten Zelle neu definiert werden
    - Befehl darf nur am Anfang der Zeile oder direkt hinter einem & stehen
  - `\vline` erzeugt eine vertikalen Linie in der Höhe der Zelle

- Bemerkungen
  - Tabellen können nicht über mehrere Seiten formatiert werden
  - Layoutänderungen
    - `\tablecolsep` bestimmt die halbe Breite des Spaltenzwischenraums
    - `\arraycolsep` bestimmt die halbe Breite des Spaltenzwischenraums bei der `array`-Umgebung
    - `\arrayrulewidth` bestimmt die Dicke der Linien in einer Tabelle
    - `\doublerulesep` Abstand der Linien bei einer Doppellinie
    - `\arraystretch` Faktor, um den der Zeilenabstand in der Tabelle verändert wird

    Werte können mit dem `\setlength`-Befehl verändert werden
  - Zentrieren der Tabelle
    - mit der `\center`-Umgebung möglich, da Tabelle als Minipage behandelt wird

Beispiele

- Überschrift in einer Tabelle

```
\begin{tabular}[t]{|r|l|r|}
  \hline \multicolumn{3}{|c|}{Zahlendarstellung}\\
  \hline
  Zahl 1984 & hexadezimal & 7C0 \\
            & octal       & 3700 \\
            & bin"ar      & 11111000000 \\
  \hline
\end{tabular}
```

Zahlendarstellung		
Zahl 1984	hexadezimal	7C0
	octal	3700
	binär	11111000000

- Tabelle fester Breite

```
\begin{tabular*}{12cm}[t]{@{\extracolsep{\fill}}|r|l|r|}
  \hline \multicolumn{3}{|c|}{Zahlendarstellung}\\
  \hline
  Zahl 1984 & hexadezimal & 7C0 \\
            & octal      & 3700 \\
            & bin"ar     & 11111000000 \\
  \hline
\end{tabular*}
```

Zahlendarstellung		
Zahl 1984	hexadezimal	7C0
	octal	3700
	binär	11111000000

- Tabelle in einer Tabelle

```
\begin{tabular}[t]{|c|c|}
  \hline
   \begin{minipage}[t]{4cm}
     \begin{center}
       \begin{tabular*}{3cm}[t]{c|c|c}
         x& & \\ \hline  & &o\\ \hline  & & \\
       \end{tabular*}
     \end{center}
     \vspace{5mm}
   \end{minipage}
        & ... \\
  \hline
   ... & ... \\
  \hline
\end{tabular}
```

x		
		o

x		
	x	o
	o	

x		x
	x	o
	o	o

x		x
	x	o
o	o	o

Erweiterungspakete für Tabellen

- Tabelle über mehrere Seiten

  longtable Formatierung von Tabellen über mehrere Seiten
  - Tabellenüberschrift und Fußzeile werden auf jeder Seite wiederholt
  - separate Definition der Tabellenüberschrift für die erste Seite
  - separate Definition der Fußzeile für die letzte Seite
  - Beschreibung: Kapitel 27 in BHB-0134 und **/usr/local/texmf/doc/latex2e/contrib/packages/tools/longtable.dvi**

  supertabular Formatierung von Tabellen über mehrere Seiten
  - prüft nach jedem \\, ob noch genug Platz vorhanden ist
  - beginnt auf der neuen Seite einen neue Tabelle
  - Breite der Tabelle kann auf den einzelnen Seiten unterschiedlich sein

- Sonstige Erweiterungspakete

  tabularx erweitert die `tabular*`-Umgebung
  - Aufweitung der Spalte und nicht des Spaltenzwischenraums
  - definiert für die Spaltendefinition den Buchstaben `X`: Spalte wird aufgeweitet

  array erweitert die Spaltendefinition, so daß z.B. Spalten in einer anderen Schriftart gedruckt werden können

Beispiel einer Tabelle mit dem Erweiterungspaket tabularx

- Setzt Paket **array** voraus
- Fußnoten sind jetzt auch direkt mit \`footnote` in der Tabelle möglich
- Keine Ausrichtung der Tabelle an Grundlinie möglich
- Beispiel:

```
 \begin{tabularx}{0.7\textwidth}{|c|X|c|X|}
\hline
 1
 & Dies ist ein langer Text in der zweiten Spalte
 & 2
 & Auch in der vierten Spalte steht ein langer Text \\
 \hline
 3 & Neue Zeile: kaum Text & 4 & Neue Zeile: \newline kurzer Text \\
 \hline
\end{tabularx}
```

1	Dies ist ein langer Text in der zweiten Spalte	2	Auch in der vierten Spalte steht ein langer Text
3	Neue Zeile: kaum Text	4	Neue Zeile: kurzer Text

Boxen, Rahmen und Striche

Boxen

- Box ist ein Stück Text und für TEX eine Einheit
- Inhalt einer Box kann nicht mehr umgebrochen werden
- drei Boxen-Typen: *LR-Boxen*, *Par-Boxen* und *Rule-Boxen*
- LR-Box:
  - Einzelbestandteile werden von links nach rechts angeordnet
  - Befehle:
    `\mbox`{*text*}, `\fbox`{*text*},
    `\makebox`[*br*][`pos`]{*text*}, `\framebox`[*br*][`pos`]{*text*}
  - `\mbox` und `\fbox` erzeugen Box, die der Breite des enthaltenen Texts entspricht
  - `\fbox` rahmt den $\boxed{\textit{Text}}$ zusätzlich ein
  - `\makebox` und `\framebox` erzeugen eine (gerahmte) Box festgelegter Breite (*br*)
  - Parameter pos gibt an, wie der Text formatiert werden soll:
    - linksbündig, pos=**l**
    - zentriert, keine Angabe
    - rechtsbündig, pos=**r**
  - Beispiel: $\boxed{\hspace{2.5cm}\text{rechtsbündiger Text}}$ wurde erzeugt mit `\framebox[8cm][r]{rechtsb. Text}`
  - LATEX bietet Befehle um Boxen zu speichern und mehrmals zu verwenden

- vertikale Verschiebungen
  - Befehl `\raisebox`{*lift*}[*oberlänge*][*unterlänge*]{*text*}
  - ezeugt eine Box vom Typ `\mbox`
  - Box wird um *lift* nach oben verschoben
  - *oberlänge* und *unterlänge* beschreiben, um wieviel die Box über und unter die Grundlinie ragt
  - Beispiel:

    ```
    Grundlinie \raisebox{1ex}{hoch} und
    \raisebox{-1ex}{tief} und zurück
    ```

    liefert:

    Grundlinie $^{\text{hoch}}$ und $_{\text{tief}}$ und zurück

Absatzboxen

- Par-Boxen: Inhalt wird als Absatz formatiert
- Befehl:
  `\parbox`[*pos*]{*breite*}{*text*}
- und Umgebung:
  `\begin{minipage}`[*pos*]{*breite*}
  text
  `\end{minipage}`
- optionale Parameter *pos*:
  - **b**, die unterste Zeile der Box
  - (ohne Angabe), die Mitte der Box
  - **t**, die oberste Zeile der Box

  wird auf die aktuelle Umgebung ausgerichtet
- Beispiel:

  ```
  \parbox{3.5cm}{Text der ersten Box,..}\hfill
  Laufende Zeile\hfill\parbox{5cm}{Text der
  zweiten Box, ...}
  ```

Text der ersten Box, Text der ersten Box, Text der ersten Box, Text der ersten Box	Laufende Zeile	Text der zweiten Box, Text der zweiten Box, Text der zweiten Box, Text der zweiten Box, Text der zweiten Box

- Unterschied Par-Box und Minipage:
  - Minipage darf fast alle Befehle enthalten,
  - Par-Box darf keine Umgebungen wie `flushleft`, … enthalten
  - Fußnoten werden innerhalb der Minipage gedruckt

Rule-Boxen

- Rule-Box ist ein gefülltes Rechteck
- Befehl:
  `\rule`[*lift*]{*breite*}{*höhe*}
- erzeugt gefülltes Rechteck der Breite *breite* und der Höhe *höhe*, das um *lift* über der Grundlinie liegt
- Beispiel:
  `\rule[4mm]{4cm}{4mm}` erzeugt

Verschachtelte Boxen

- Box-Befehle können beliebig verschachtelt werden
- doppelte Umrahmung mit geschachtelten `\fbox`, `\fbox` und `\parbox`:

  Eine Parbox, umrahmt von einer fbox, und dann noch eine fbox.

Box-Stilparameter

- für Befehle `\fbox` und `\framebox` gelten folgende Größen:
  - `\fboxrule`, Linienstärke der Rahmenbox, Default: 0.4 pt
  - `\fboxsep`, Abstand zwischen Rahmen und Text, Default: 4 pt
- Beispiel:
  `\setlength{\fboxrule}{1pt}`

Erweiterungpaket für Boxen

- Erweiterungpaket **fancybox** definiert folgende Befehle zusätzlich:
  - `\shadowbox`, Box mit $\boxed{\text{Schatten}}$
  - `\doublebox`, Box mit $\boxed{\boxed{\text{Doppelrahmen}}}$
  - `\ovalbox`, Box mit $\boxed{\text{Ovalrahmen}}$
  - `\Ovalbox`, Box mit $\boxed{\text{Ovalrahmen}}$, dickere Linien
- wird geladen mit:
  `\usepackage{fancybox}`

Mathematik-Modus

- Formeln werden in TEX und LATEX durch beschreibenden Text erzeugt
- z.B. $(a + b)^2$ durch `$(a+b)^2$`
- spezieller $mathematischer\ Bearbeitungmodus$
- zwei Arten:
  - Formeln im Fließtext
  - abgesetzte Formeln
- Numerierung möglich bei abgesetzten Formeln
- beschreibender Text ist für alle Arten gleich

Mathematische Umgebungen

- Formeln im Fließtext
  - Umgebung: `\begin{math}` $formeltext$ `\end{math}`
  - oder mit `\(`$formeltext$`\)`
  - oder mit `$`$formeltext$`$` (übliche Form, da kurz)
  - werden in kleinen Schriften gesetzt
- abgesetzte Formeln
  - Umgebungen:
    - abgesetzte Formel
      `\begin{displaymath}` $formel$ `\end{displaymath}`
      Abkürzung: `\[`$formeltext$`\]`
    - abgesetzte Formel mit Numerierung (rechtsbündig)
      `\begin{equation}` $formeltext$ `\end{equation}`
  - Formeln werden horizontal zentriert
  - Dokumentklassenoption `fleqn`
    Formeln werden linksbündig mit wählbarer Einrücktiefe gesetzt (`\setlength{\mathindent}{`$tiefe$`}`)
- Formelgruppen:
  - Formelgruppe mit Numerierung
    `\begin{eqnarray}` $formeltext$ `\end{eqnarray}`
  - Formelgruppe ohne Numerierung
    `\begin{eqnarray*}` $formeltext$ `\end{eqnarray*}`

Konstruktionselemente

- Konstante, Variable und ihre Verknüpfungen
  - Zahlen sind Konstanten, Buchstaben sind Variablen
  - mathematischer Standard: Konstanten in Schrift Roman, Variablen in Schrift $Kursiv$
  - Leerzeichen werden von TEX nicht beachtet
  - Abstände werden automatisch je nach Kontext generiert
  - mathematische Symbole auf der Tastatur:
    `+ - = < > / : ! ' | [ ] ( )`
  - logische Klammerung mit { }, als Zeichen: `\{ \}`
- Hoch- und Tiefstellungen
  - Exponenten: Befehlszeichen ^
    z.B. `$x^i$` $\rightarrow x^i$
  - Index: Befehlszeichen _
    z.B. `$q_n$` $\rightarrow q_n$
  - gleichzeitige Hoch- und Tiefstellung: `$q^i_n$` $\rightarrow q_n^i$
    Reihenfolge gleichgültig
  - mehrere Zeichen werden mit { } geklammert:
    z.B. `$A_{i,j,k}^{-n+2}$` $\rightarrow A_{i,j,k}^{-n+2}$
  - Schachtelung durch Klammern { }:
    z.B. `$A^{x_i^2}_{j^{2n}_{n,m}}$` $\rightarrow A^{x_i^2}_{j^{2n}_{n,m}}$
  - Index kann auch auch vorangestellt werden:
    `$_nA$` $\rightarrow {}_nA$
- Befehle sind nur im mathematischen Modus erlaubt

- Brüche
  - kurze Brüche: Bruchzeichen /
    z.B. `$(n+m)/2$` → $(n+m)/2$
  - größere Brüche: Befehl `\frac`{*Zähler*}{*Nenner*}
    z.B. `$\frac{1}{x+y}$` → $\frac{1}{x+y}$
  - Bruchstrich mit der Breite des längeren Teils
  - kürzerer Teil wird zentriert
  - Brüche können geschachtelt werden, z.B.
    `$\frac{\frac{1}{x+y}+\frac{1}{x-y}}{y-3}$`
    $\frac{\frac{1}{x+y}+\frac{1}{x-y}}{y-3}$
- Wurzeln
  - Befehl `\sqrt`[*n*]{*arg*}
    z.B. `$\sqrt[3]{8}=2$` → $\sqrt[3]{8}=2$
  - ohne Parameter n: quadratische Wurzel $\sqrt{a}$
- Fortsetzungspunkte und Ellipsen
  - hintereinanderstehende Punkte: . . .
  - `\ldots` (*low dots*) z.B. $x_1 \ldots x_n$
  - `\cdots` (*center dots*) z.B. $x_1 + \cdots + x_n$
  - `\vdots` (*vertical dots*) $\vdots$
  - `\ddots` (*diagonal dots*) $\ddots$

- Summen und Integrale
  - Summe: Befehl `\sum` → $\sum$
  - Integral: Befehl `\int` → $\int$
  - untere und obere Grenze mit _ und ^
    z.B. `$\int_a^b \sum_{i=1}^n$` → $\int_a^b \sum_{i=1}^n$
  - Behandlung in abgesetzten Formeln anders: $$\int_a^b \sum_{i=1}^n$$
  - Befehl `\limits` setzt Grenzen unter bzw. über das Zeichen:

    `$\int\limits_{x=0}^{x=1}$` $\int\limits_{x=0}^{x=1}$
  - Abrücken des Differentialoperators:

```
\[\int^b_a f_i(x)
  g_i(x) \, dx \]
```

$$\int^b_a f_i(x) g_i(x) \, dx$$

Mathematische Symbole

- griechische Buchstaben
  - Kleinbuchstaben: α `\alpha` . . . ξ `\xi`
  - Großbuchstaben: Γ `\Gamma` . . . ξ `\Xi`
- kalligraphische Buchstaben
  - 26 kalligraphische Buchstaben:
    `${\cal A, ...,Z}$`
    $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L}, \mathcal{M}, \mathcal{N}, \mathcal{O}, \mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$
- binäre Operatoren
  - Verknüpfung mathematischer Größen:
    z.B. $\pm$ `\pm`, $\div$ `\div`, $\vee$ `\vee` ,$\wedge$ `\wedge`
- Vergleichssymbole und deren Negation
  - Beziehungsoperationen:
    z.B. $\le$ `\le` oder `\leq`, $\ge$ `\ge` oder `\geq`, $\in$ `\in` ,$\sim$ `\sim`, $\neq$ `\neq`
  - Negation durch Voransetzen von `\not`
    z.B. $\not\in$ `\not\in`, $\not\ge$ `\not\ge`
- Pfeile und Zeiger
  z.B. $\leftarrow$ `\leftarrow` oder `\gets`, $\rightarrow$ `\rightarrow` oder `\to`, $\Leftarrow$ `\Leftarrow`, $\Rightarrow$ `\Rightarrow`, $\Leftrightarrow$ `\Leftrightarrow`

- Funktionsnamen
  - Standard: Funktionsnamen werden in der Schriftart Roman gesetzt
  - Funktionen sind als Befehle definiert:
    z.B. `\cos`, `\arccos`, `\lim`, `\sinh`
  - Grenzen: z.B. `$\lim_{a\to\infty}$` → $\lim_{a\to\infty}$
- mathematische Akzente
  - für einzelne Symbole:
    $\hat{a}$ `\hat{a}`, $\breve{a}$ `\breve{a}`, $\grave{a}$ `\grave{a}`, $\bar{a}$ `\bar{a}`, $\dot{a}$ `\dot{a}`, $\check{a}$ `\check{a}`, $\acute{a}$ `\acute{a}`, $\tilde{a}$ `\tilde{a}`, $\vec{a}$ `\vec{a}`, $\ddot{a}$ `\ddot{a}`
  - Buchstaben i und j verlieren Punkt mit:
    $\vec{\imath}$ `\vec{\imath}`, $\vec{\jmath}$ `\vec{\jmath}`,
  - für Formelteile:
    $\widehat{x+1}$ `\widehat{x+1}`, $\widetilde{x+1}$ `\widetilde{x+1}`

Weitere Konstruktionselemente

- automatische Größenanpassung von Klammern
  - Klammern sollten in der Größe der eingeschlossenen Teilformel angepaßt sein
  - Befehlspaar `\left`*lsymbol Teilformel* `\right`*rsymbol*
  - *lsymbol* und *rsymbol* stehen für öffnende und schließende Klammersymbole und können verschieden sein
  - Klammersymbol . steht für eine unsichtbare Klammer, wird bei Fallunterscheidungen gebraucht
  - Beispiele:

```
\[ \left[ \int +\int
   \right]_{x=0}^{x=1} \]
```

$$\left[\int + \int \right]_{x=0}^{x=1}$$

```
\[ y= \left\{
\begin{array}{r@{\quad:\quad}l}
-1 & x<0 \\
 0 & x=0 \\
+1 & x>0
\end{array} \right. \]
```

$$y= \left\{ \begin{array}{r@{\quad:\quad}l} -1 & x<0 \\ 0 & x=0 \\ +1 & x>0 \end{array} \right.$$

- gewöhnlicher Text
  - geht über eine LR-Box
  - Befehl: `\mbox`{*normaler Text*}
  - Zwischenraum z.B. mit `\quad`
  - Beispiel:

    z.B. `$y=3/2 \mbox{\quad f"ur \quad} x=0$`

    $\rightarrow y = 3/2 \mbox{\quad für \quad} x = 0$

- Matrizen und Felder
  - werden mit der `array`-Umgebungen erstellt
  - `array`-Umgebungen können geschachtelt werden
  - stellt als Struktur eine vertikale Box dar $\rightarrow$ einzelnes Zeichen
  - Verknüpfung mit anderen Elementen möglich
  - Beispiele:

```
\[ \left(
\begin{array}{cc}
  a_{11} & a_{12} \\
  a_{21} & a_{22} \\
\end{array}
\right) \]
```

$$\rightarrow \left(\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right)$$

```
\[ \Delta_{
\begin{array}{l}
   p_1p_2\cdots p_{n-k}\\
   p_1p_2\cdots p_{n-k}
\end{array}
} \]
```

$$\Delta_{ \begin{array}{l} p_1p_2\cdots p_{n-k}\\ p_1p_2\cdots p_{n-k} \end{array} }$$

```
\[ \begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{array} \]
```

$$\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array}$$

- Über- und Unterstreichen von Teilformeln
  - Überstreichen: Befehl `\overline`{*teilformel*}
  - Unterstreichen: Befehl `\underline`{*teilformel*}
  - horizontale geschweifte Klammer:
    `\overbrace`{*teilformel*} und `\underbrace`{*teilformel*}
  - Hoch- und Tiefstellung setzt Text über bzw. unter die Klammer
  - Beispiele:

```
\[ \overline{\overline{a}^2
   + \underline{xy}
   + \overline{\overline{z}}} \]
```

$$\overline{\overline{a}^2 + \underline{xy} + \overline{\overline{z}}}$$

```
\[ \underbrace{
    a+\overbrace{b+c}^{x}+d
    }_{y}
\]
```

$$\underbrace{ a+\overbrace{b+c}^{x}+d }_{y}$$

- gestockte Symbole
  - Befehl `\stackrel`{*oberes Symbol*}{*unteres Symbol*}
  - Symbole werden zentriert übereinandergesetzt
  - Beispiel:

```
\[ \vec{x}
   \stackrel{\textrm{def}}{=}
   (x_1,\ldots,x_n) \]
```

$$\vec{x} \stackrel{\textrm{def}}{=} (x_1,\ldots,x_n)$$

- zusätzliche mathematische TEX-Befehle
  - nützliche Befehle aus Plain-TEX : `\atop` und `\choose`
  - `\atop`: Struktur wie ein Bruch, aber ohne Bruchstrich
  - `\choose`: zusätzliche runde Klammer
  - Syntax:
    {*oben* `\atop` *unten*}, {*oben* `\choose` *unten*}
  - Beispiele:

```
\[ \Delta_{
     p_1p_2\cdots p_{n-k}
     \atop
     p_1p_2\cdots p_{n-k}}
\]
```

$$\Delta_{p_1p_2\cdots p_{n-k} \atop p_1p_2\cdots p_{n-k}}$$

```
\[ {n+1 \choose k}
     = {n \choose k}
       + {n \choose k-1}
\]
```

$${n+1 \choose k} = {n \choose k} + {n \choose k-1}$$

  - `\atop` benutzt kleinere Schriftarten, ist daher bei Indexfeld gegebenüber `array` vorzuziehen
  - Formelbefehle in Plain-TEX anders, aber in LATEX vorhanden

- mehrzeilige Formeln
  - Formelgruppe mit Numerierung
    `\begin{eqnarray}` *zeile*`\\`...`\\`*zeile* `\end{eqnarray}`
  - Formelgruppe ohne Numerierung
    `\begin{eqnarray*}` *zeile*`\\`...`\\`*zeile* `\end{eqnarray*}`
  - Aufbau einer Zeile:
    linker Formelteil & *mittlerer Formelteil* & *rechter Formelteil* `\\`
  - der linke Teil wird rechtsbündig zum ersten & gesetzt
  - der mittlere Teil wird zu den beiden Positionierungen zentriert
  - der rechte Teil wird linksbündig zum zweiten & gesetzt
  - mittlere Teil sollte nur ein einzelnes Zeichen sein: =, >
  - Beispiel:

```
(x+y)(x-y) & = & x^2 - xy +xy -y^2 \\
           & = & x^2 - y^2 \\
   (x+y)^2 & = & x^2 +2xy +y^2
```

$$\begin{aligned}(x+y)(x-y) &= x^2 - xy + xy - y^2 \quad (1)\\ &= x^2 - y^2 \quad (2)\\ (x+y)^2 &= x^2 + 2xy + y^2 \quad (3)\end{aligned}$$

Mathematische Formatierungshilfen

- horizontale Abstände
  - kleiner Zwischenraum: `\,` (3/18 eines `\quad`)
  - mittlerer Zwischenraum: `\:` (4/18 eines `\quad`)
  - großer Zwischenraum: `\;` (5/18 eines `\quad`)
  - negativer Zwischenraum: `\!` (-3/18 eines `\quad`)
- manuelle Größenwahl der Klammersymbole
  - Klammersymbolen können folgende Befehle vorangestellt werden:
    `\big`, `\Big`, `\bigg`, `\Bigg`
  - Klammern müssen nicht gepaart sein
  - Beispiel:

```
\[
   \Bigg(\bigg(\Big(\big(
 a \big]\Big]\bigg]\Bigg]
\]
```

$$\Bigg(\bigg(\Big(\big(a \big]\Big]\bigg]\Bigg]$$

Numerierte und gleitende Objekte

- Abbildungen und Tabellen passen häufig nicht mehr auf die aktuelle Seite
- Abbildung soll auf die nächste Seite gleiten
- Seite soll mit nachfolgendem Text aufgefüllt werden
- Unterschrift, Tabellentitel usw. sollen mit dem Objekt auf die nächste Seite gleiten
- Umgebungen
  `\begin{figure}`[*wohin*] *bild* `\end{figure}`
  `\begin{figure*}`[*wohin*] *bild* `\end{figure*}`
  `\begin{table}`[*wohin*] *tabelle* `\end{table}`
  `\begin{table*}`[*wohin*] *tabelle* `\end{table*}`
- Stern-Form für zweispaltige Formatierung: Abbildung geht über beide Spalten
- *bild* steht für beliebiges Material, z.B. Graphik
- *tabelle* steht für eine `tabular`-Umgebung
- bei beiden kann ein `\caption`-Befehl eingefügt werden
- *wohin* ist eine Angabe über die Plazierung der Abbildung

Plazierung von Gleitobjekten

- optionaler Parameter *wohin* kann einen oder mehrere der folgenden Buchstaben enthalten:
  - **h** *Here*: Objekt wird an der aktuellen Stelle gedruckt, falls möglich (nicht bei den Stern-Formen erlaubt)
  - **t** *Top*: Objekt wird an den Anfang der aktuellen Seite gesetzt, falls die Seite nicht schon zu voll ist
    Ansonsten Positionierung am Anfang der nächsten Seite
  - **b** *Bottom*: Objekt wird am Ende der aktuellen Seite gedruckt, falls die Seite nicht schon zu voll ist
    Ansonsten Positionierung am Ende der nächsten Seite (nicht bei den Stern-Formen erlaubt)
  - **p** *Page of floats*: Objekt wird auf einer Seite gedruckt, die nur gleitende Objekte enthält
- Buchstaben sind kombinierbar und geben eine Reihenfolge der Positionierungsvorschläge an
- Default: `tbp` (*Top, Bottom, Page of floats*)
- Buchstaben kann ein Ausrufezeichen `!` vorangestellt werden:
  - Restriktionen der Stilparameter werden nicht beachtet
  - Anzahl der Objekte pro Seite ist nicht beschränkt
  - Beispiel: `!t` druckt Objekt in (fast) jedem Fall an den Anfang der Seite
  - neue Option in LaTeX 2_ε

- Regeln:
  - kein Objekt erscheint auf einer früheren Seite als der, auf der es definiert wurde
  - Ausdruck der Objekte erfolgt in der Reihenfolge ihrer Definition
  - Stilparameter werden bei der Positionierung berücksichtigt
  - bei `ht` hat der Parameter `h` Vorrang, auch wenn am Anfang der Seite Platz ist
- `\clearpage` und `\cleardoublepage` bewirken die Ausgabe aller noch offenen Gleitobjekte
- Unterdrückung der Ausgabe von Objekten
  - Befehl `\suppressfloats`[*wohin*] unterdrückt die Ausgabe von Objekten am Anfang (`t`) oder am Ende (`b`) der Seite
  - sinnvoll bei Kapiteln, die in der Mitte der Seite beginnen und die Objekte aber am Anfang der Seite positioniert werden
  - neuer Befehl in LaTeX 2_ε

Stilparameter

- Zähler:
  - *topnumber*
    maximale Anzahl von Objekten am Anfang einer Seite (2)
  - *bottomnumber*
    maximale Anzahl von Objekten am Ende einer Seite (1)
  - *totalnumber*
    maximale Anzahl von Objekten insgesamt auf einer Seite (3)

  werden mit `\setcounter`{*zählername*}{*wert*} gesetzt
- andere Stilbefehle:
  - `\topfraction`
    gibt den Bruchteil der Seite an, bis zu dem die Seite oben mit Objekten gefüllt werden darf (0.7)
  - `\bottomfraction`
    gibt den Bruchteil der Seite an, bis zu dem die Seite unten mit Objekten gefüllt werden darf (0.3)
  - `\textfraction`
    gibt den Bruchteil der Seite an, der mindestens für Text zur Verfügung stehen muß (0.2)
  - `\floatpagefraction`
    gibt den Bruchteil einer eigenen Seite für Gleitobjekte an, der mindestens gefüllt sein muß, bevor eine neue Seite begonnen wird (0.5)

  werden mit `\renewcommand`{*cmd*}{*wert*} gesetzt

- Längenangaben:
  - `\floatsep`
    Abstand zwischen zwei gleitenden Objekten
    12pt plus2pt minus2pt (bei Schriftgröße 10pt, 11pt),
    14pt plus2pt minus2pt (bei Schriftgröße 12pt)
  - `\textfloatsep`
    Abstand zwischen einem gleitenden Objekt und dem vorhergenden (*Bottom*) oder dem nachfolgenden (*Top*) Text (20pt plus2pt minus4pt)
  - `\intextsep`
    Abstand zwischen dem gleitenden Objekt bei dem Parameter `h` (Default entspricht `\floatsep`)

  werden mit `\setlength`{*cmd*}{*wert*} gesetzt
- Parameter für zweispaltigen Ausdruck sind nicht aufgeführt

Über- und Unterschriften

- Befehl \caption[*kurzform*]{*überschrift*}
- erstellt einen Titel, der die laufende Nummer enthält
- getrennte Numerierung für Tabellen und Abbildungen
- beginnt bei Abbildungen mit `Abbildung` n: *(german-Style)*
- beginnt bei Tabellen mit `Tabelle` n:
- *kurzform* wird in das Abbildungs- oder Tabellenverzeichnis eingetragen, falls vorhanden, ansonsten *überschrift*
- Titel wird an der Stelle der Definition ausgegeben:
  über dem Objekt, falls vorher definiert
  unter dem Objekt, falls nachher definiert
- Titel wird zentriert, falls er kürzer als die Zeilenlänge ist
- ansonsten wird er wie ein Absatz formatiert
- Titel kann mit einer Par-Box an die Breite der Abbildung angepaßt werden
- Objekte können mit einem \label-Befehl versehen werden und somit referenziert werden
- bessere Layoutgestaltung durch das Erweiterungspaket **caption**

Verzeichnisse

- Tabellenverzeichnis:
  - Befehl: \listoftables, Datei: *fn*.lot
- Abbildungsverzeichnis:
  - Befehl: \listoffigures, Datei: *fn*.lof
- ggf. mehrere TEX-Durchläufe

Erweiterungspakete

- **caption**
  - Layoutänderungen am Abbildungstitel
  - Befehl: \usepackage[*options*]{caption}
  - Option *options*:

normal	normale LATEX-Layout des Abbildungstitels
hang oder **isu**	Beschreibungstext wird eingerückt
center	Titel wird zentriert
centerlast	Letzte Zeile des Titels wird zentriert
nooneline	kurze Titel werden nicht zentriert
scriptsize, . . . , **Large**	setzen Schriftgröße
up, it, sl, sc, md, bf, rm, sf, oder **tt**	setzen Schriftattribute

- **float**
  - unterstützt die Definition neuer gleitender Objekte, z.B.: Algorithmen, Listings, ...
  - bietet neue Styles für das Layout von gleitenden Objekten: `boxed`, `ruled`
  - neuer Positionierungsparameter:
    H Objekt wird immer an der aktuellen Stelle gedruckt
  - bei den Objekten `figure` und `table` Befehl \restylefloat{*object*} notwendig
  - Beispiel:

```
\floatstyle{boxed}
\restylefloat{figure}
```


- **floatflt, wrapfig**
  - ermöglichen, daß schmale Abbildungen neben dem Text stehen können
- **subfigure**
  - unterstützt die Ausgabe von Abbildungen, die aus mehreren Einzelabbildungen bestehen
  - Abbildungstitel sowohl für die gesamte als auch für die einzelnen Abbildungen möglich
- **endfloat**
  - trennt die Abbildungen vom eigentlichen Text
  - Platz für Abbildungen im Text bleibt frei
  - Abbildungen werden am Ende des Dokuments auf einzelnen Seiten gedruckt
  - Struktur wird von manchen Verlagen erwartet

Graphiken

PostScript-Graphiken

- **PostScript**
  - Sprachstandard von *Adobe*
  - definiert das Aussehen einer Seite (Seitenbeschreibungssprache)
  - PostScript-Dateien sind im ASCII-Format abgespeichert
  - Interpretation der Sprache durch *Ghostview/Ghostscript* oder durch *Interpreter* im Drucker

```
%!PS-Adobe-2.0 EPSF-2.0
%%Creator: gnuplot
%%DocumentFonts: Times-Roman
%%BoundingBox: 50 50 410 302
%%EndComments
```

- **EPS** (Encapsulated PostScript)
  - **Bounding Box**-Kommentar gibt Größe des umgebenden Rahmens in Points ($\frac{1}{72}$ inch) an (TEX-Points: $\frac{1}{72.27}$ inch)
  - PostScript-Befehl **showpage** darf nicht vorkommen

- Einbinden von Graphiken
  - stützt sich auf das Box-Konzept von TEX ab
  - mögliche Formate:
    - PostScript (EPS, PS)
    - Bitmap (PCX, BMP, MSP)
  - notwendige Pakete:
    - **graphicx**-Paket
    - **graphics**-Paket
  - aus Kompatibilitätsgründen:
    - **epsfig**-Paket
      ersetzt **epsf**-, **psfig**- und **epsfig**-Umgebungen aus LATEX 2.09
  - Syntax:

```
\usepackage[dvips]{...}    % Name des
                           % verwendeten Pakets
\begin{figure}[...]
{\centering        % oder center-Umgebung
\...               % hier erscheint der Befehl
                   % zur Graphikeinbindung
\caption{...}      % hier wird die Bild-
                   % unterschrift eingetragen
\label{...}        % ein Label für spätere
                   % Referenzierungen
}\end{figure}
```


- Das **graphicx**-Paket
  - Syntax wie bei \psfig-Kommando von LATEX 2.09
  - unterstützt Skalierung und Rotation von Bildern
  - erlaubt die Positionierung von Bildern innerhalb einer Zeile
  - vorhandene Befehle:

```
\includegraphics[options]{filename}
\rotatebox{angle}{argument}
\scalebox{h-scale}[v-scale]{argument}
\resizebox{width}{height}{argument}
\resizebox*{width}{totalheight}{argument}
```

- Das **graphics**-Paket
  - Syntax wie bei \epsf-Kommando von LATEX 2.09
  - Skalierung und Rotation von Bildern ähnlich **graphicx**-Befehlen (\rotatebox, \resizebox, . . .)
  - vorhandene Befehle:

```
\includegraphics[llx,lly][urx,ury]
    {filename}
\includegraphics*[llx,lly][urx,ury]
    {filename}
```


- Das **epsfig**-Paket
  - stellt Kompatibilität zu LATEX 2.09 her
  - Aufruf mit:

```
\epsfig{file=...,height=...,width=...,
       angle=...}
```

- Draft-Modus
  - für Probeausdrucke wird der Bildinhalt durch den Dateinamen ersetzt
  - die Größe des Bildes wird durch einen Rahmen markiert
  - Syntax:

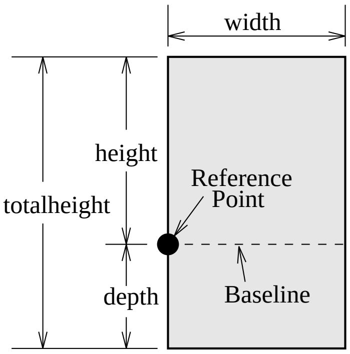
```
\usepackage[draft]{graphicx} oder
\usepackage[draft]{graphics}
```

- Einbindung von komprimierten EPS-Dateien
  - durch den Befehl \DeclareGraphicsRule wird festgelegt, wie komprimierte Dateien (*.gz, . . .) behandelt werden
  - Syntax:

```
\DeclareGraphicsRule{ext}{type}{sizefile}
{command}
```

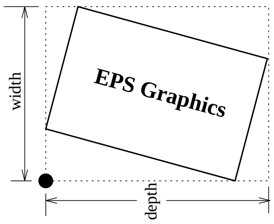

  - Vorgaben:
    ***.eps.Z** ***.ps.Z**
    ***.eps.gz** ***.ps.gz**
    ***.pz**
    $\Longrightarrow$ dekomprimieren durch **gunzip**
  - eigene Vorgaben sind möglich
- Vorgabe von Dateitypen
  - ermöglicht das Weglassen des Dateityps
  - Vorteil: bei Veränderungen des Graphikformats muß die LATEX-Datei nicht abgeändert werden
  - Dateitypen, die automatisch erkannt werden:
    ***.eps, *.ps, *.eps.gz, *.ps.gz, *.eps.Z**
  - durch \DeclareGraphicsExtensions-Befehl Erweiterung dieser Liste für eigene Bedürfnisse möglich

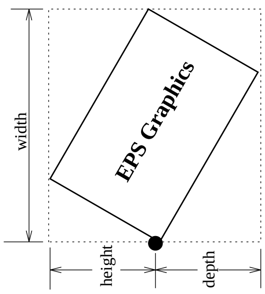
- Bilder und LaTeX-Boxen
  - Bezugspunkt an linker Box-Seite
  - Grundlinie geht durch Bezugspunkt
  - Box wird charakterisiert durch ihre *Höhe*, *Tiefe* (Unterlänge) und *Breite*

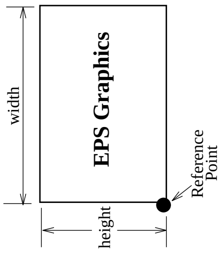


  - bei nicht-rotierten Boxen ist Bezugspunkt mit unterer linker Ecke identisch
  - durch Rotation „wandert“ der Bezugspunkt

- man beachte die Maße an der Box:

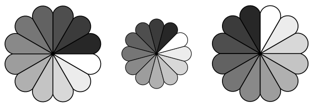




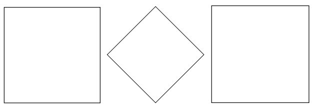


Rotation

- **graphicx**- und **epsfig**-Paket durch Angabe eines Winkels
- **graphics**-Paket durch Verwendung des Kommandos `\rotatebox` (auch allgemein zu verwenden)
- es wird entgegen dem Uhrzeigersinn gedreht (positiver Winkel ergibt Drehung nach links)
- es wird immer die Box (mit Inhalt) gedreht $\Longrightarrow$ Größenveränderungen des Bildes sind möglich:



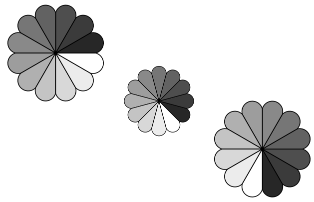
- dazugehörige Boxen:

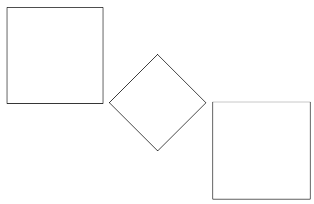


- Bilder wurden durch die folgenden LaTeX-Befehle erzeugt:

```
\begin{center}
\includegraphics[totalheight=1in]{fig5.epsi}
\includegraphics[angle=45,totalheight=1in]
{fig5.epsi}
\includegraphics[angle=90,totalheight=1in]
{fig5.epsi}
\end{center}
```


- Ausrichtung von rotierten Bildern (Boxen)

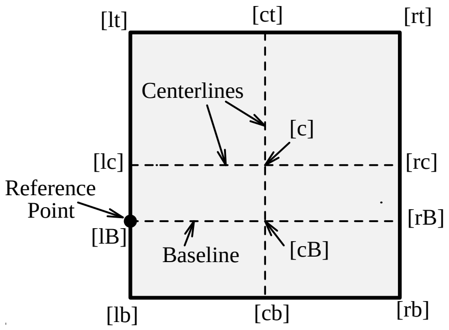




- LaTeX-Befehle ohne Berücksichtigung des Bezugspunkts

```
\begin{center}
\includegraphics[totalheight=1in]{fig5.epsi}
\includegraphics[angle=-45,totalheight=1in]
{fig5.epsi}
\includegraphics[angle=-90,totalheight=1in]
{fig5.epsi}
\end{center}
```

- Bezugspunkte für das \rotatebox-Kommando:



- Abhilfe: Berücksichtigung des Bezugspunkts

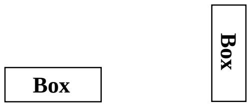
```
\begin{center}
\includegraphics[totalheight=1in]{fig5.epsi}
\rotatebox[origin=c]{-45}{
\includegraphics[totalheight=1in]
{fig5.epsi}}
\rotatebox[origin=c]{-90}{
\includegraphics[totalheight=1in]
{fig5.epsi}}
\end{center}
```


- weiteres Beispiel:

```
\begin{center}
\includegraphics[width=1in]{fig4.epsi}
\hspace{1in}
\includegraphics[width=1in,angle=-90]{fig4.epsi}
\end{center}
```

- Abhilfe: Bezugspunkt richtig wählen

```
\begin{center}
\includegraphics[width=1in]{fig4.epsi}
\hspace{1in}
\rotatebox[origin=br]{-90}{
\includegraphics[width=1in]{fig4.epsi}}
\end{center}
```



Skalierung

- Größe einer Graphik wird festgelegt durch:
  - **Bounding Box**-Größenangabe im EPS-File
  - expliziter Angabe einer Höhe im LaTeX-File – Breite wird dazu proportional von LaTeX ermittelt
  - expliziter Angabe einer Breite im LaTeX-File – Höhe wird dazu proportional von LaTeX ermittelt
  - expliziter Angabe von Höhe und Breite im LaTeX-File – Stauchung oder Streckung der Graphik ist möglich
- Beispiel:

```
\includegraphics[width=\textwidth]{box.eps}
```

- weitere Befehle:
  - \scalebox ermöglicht die Angabe von Faktoren für Stauchung (Faktor <0) bzw. Streckung (Faktor >0) der Seitenlängen
  - Syntax:

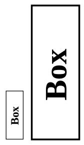
```
\scalebox{h-scale}[v-scale]{argument}
```

  - \resizebox ermöglicht die Angabe einer Maßzahl für die Seitenlängen
  - Syntax:

```
\resizebox{width}{height}{argument}
\resizebox*{width}{totalheight}{argument}
```


- bei Kombination von Drehung mit Skalierung Reihenfolge der Befehle beachten:

```
\begin{center}
\includegraphics[angle=90,totalheight=0.5in]
   {fig4.epsi}
\includegraphics[totalheight=0.5in,angle=90]
   {fig4.epsi}
\end{center}
```



Paket epsfig

- Vorspann: \usepackage[dvips]{epsfig}
- Befehl:

```
\epsfig{file=fn,
        height=ht,width=wd, angle=deg,
        bbllx=blx, bblly=bly,
        bburx=brx, bbury=bry
        silent=, clip= }
```

- Parameter:

file	Dateiname
height	gewünschte Höhe der Graphik
width	gewünschte Breite der Graphik
angle	Drehwinkel (gegen Uhrzeigersinn)
bbllx, ...	Koordinaten der Bounding-Box
silent	keine Bildschirmmeldungen
clip	PostScript-Elemente außerhalb der Bounding-Box werden nicht gedruckt

- Beispiel:

```
\begin{center}
\epsfig{file=fig4.eps, width=7cm, clip=}
\end{center}
```


Spezielle Möglichkeiten
Teildokumente

- Source-Code des Dokuments kann in verschiedene Dateien aufgeteilt werden
- besonders bei längeren Dokumenten hilfreich (Übersichtlichkeit, Größe der Dateien usw.)
- zwei Befehle `\input` und `\include`
- Befehl `\input`
  - Syntax:
    `\input`{*file_n*} oder `\input` *file_n* (TEX)
  - Endung *.tex* muß nicht angegeben werden
  - fügt bei der Formatierung des Dokuments den Inhalt der angegebenen Datei ein
  - eingebundene Datei darf selber wieder `\input`-Befehle enthalten, Schachtelung möglich
  - bei großen Dokumenten geschieht die Aufteilung meistens auf der Kapitel-Ebene
  - Beispiel (LATEX 2_ε-Kurzbeschreibung):

```
\input{l2k1}
\clearpage

\input{l2k2}
\clearpage

\input{l2k3} \input{l2ksym}
\clearpage

\input{l2k4}
```


- Befehl `\include`
  - mit `\input` wird immer das ganze Dokument formatiert
  - mit `\include` ist eine partielle Formatierung des Dokuments möglich
    - nur das Kapitel, das gerade verändert wird, wird auch formatiert
  - Syntax:
    `\include`{*file_n*}
  - Endung *.tex* muß nicht angegeben werden
  - Dateien dürfen nicht geschachtelt werden
  - Befehl im Vorspann des Dokuments:
    `\includeonly`{*dateiliste*}
  - definiert diejenigen Dateien, die eingebunden werden sollen
  - fehlt der Befehl `\includeonly`, werden alle Dateien eingebunden
  - Befehl `\include`{*file_n*} wird ersetzt durch:
    - `\clearpage`, falls Datei nicht eingebunden werden soll fehlt
    - `\clearpage\input`{*file_n*}`\clearpage`, falls Datei im `\includeonly`-Befehl vermerkt oder `\includeonly`-Befehl fehlt
  - bei der Formatierung nur eines Kapitels werden Referenzen und Seitennummer wie bei der Formatierung des ganzen Dokuments erstellt

Index-Erstellung

- Stichwortverzeichnis
- Befehle im Vorspann:
  `\usepackage{makeidx}`
  `\makeindex`
- Markierung der Einträge:
  `\index`{*key*}
- *key* besteht aus Schlüsselwörtern, die mit ! getrennt sind:
  - vordere Wörter sind Überschriften, unter denen der Begriff (letztes Wort) einsortiert wird
  - Beispiel: `\index{Kurse!\LaTeX-Kurs}`
- Formatierung: Index-Einträge mit der entsprechenden Seitennummer werden in eine Datei (*.idx*) geschrieben
- Datei (*.idx*) muß mit dem Programm ***makeindex*** sortiert werden, →Datei *.ind*
- sortierte Datei *.ind* wird mit dem LATEX-Befehl `\printindex` z.B. im Anhang eingebunden
- Erweiterungspaket **showidx** druckt im Dokument alle Indexeinträge am linken Rand

Mehrspaltiger Druck

- Dokumentklassenoption `twocolumn`
  - zweispaltiger Druck des gesamten Dokuments
  - Befehl `\onecolumn` beginnt eine neue Seite und einspaltigen Ausdruck
  - Befehl `\twocolumn{`*text*`}` beginnt eine neue Seite und zweispaltigen Ausdruck, *text* wird zentriert über beide Spalten formatiert
  - Abstand der beiden Spalten:
    `\setlength{\columnsep}{`*abstand*`}`
  - Dicke der Trennungslinie zwischen beiden Spalten:
    `\setlength{\columnseprule}{`*dicke*`}`
    Default: 0pt, unsichtbar
  - Erweiterungspaket **multicol**
    - Erstellung von mehrspaltigem Ausdruck
    - Spaltenanzahl kann innerhalb der Seite wechseln
    - Umgebung:
      `\begin{multicols}{`*columns*`}[`*preface*`][`*skip*`]` ...
      `\end{multicols}`
    - Spalten werden automatisch ausgeglichen

Sprachumschaltung

- bei der Verwendung der Erweiterungspakets **german**
  - Befehl: `\selectlanguage{`*sprache*`}`
  - *sprache* steht für:
    `austrian`, `english`, `french`, `german` oder `USenglish`
  - Sprache kann auch innerhalb des Dokuments umgeschaltet werden
  - Trennung und Literale werden auf die entsprechende Sprache eingestellt
  - sinnvoll bei kürzeren Zitaten in anderer Sprache

Schriften

Weitere METAFONT-Schriften

- stehen als METAFONT-Source zur Verfügung
- können über Style-Dateien oder über die 5 Font-Befehle `\fontencoding`, `\fontfamily`, `\fontseries`, `\fontshape`, `\fontsize` und `\selectfont` aktiviert werden
- **Eulersche Schriften**
  - werden in naturwissenschaftlichen und mathematischen Büchern benötigt
  - Erweiterungspaket **euler**:
    - Euler Roman Schrift: Standard für mathematische Schrift
    - Befehle
      `\mathversion{normal}` und `\mathversion{bold}`
      schalten zwischen normaler und fetter Schrift um
  - vorhandene Schriftarten:
    Kodierung `U`

Familie	Stärke	Form	Beispiel
eur	m	n	Euler Roman medium
eur	b	n	**Euler Roman bold**
eus	m	n	EULER SCRIPT
euf	m	n	Euler Fraktur

  - keine Umlaute vorhanden

- **Altdeutsche Schriften**
  - Erweiterungspaket **oldgerm**:
    - Befehle `\textgoth`, `\textfrak` und `\textswab` schalten entsprechende Schriften ein
  - vorhandene Schriftarten: Kodierung `U`

Familie	Stärke	Form	Beispiel
ygoth	m	n	Gothische Schrift
yfrak	m	n	Fraktur Schrift
yswab	m	n	Schwabacher Schrift

- **Sütterlin-Schriften**
  - Erweiterungspaket **suetterl**:
    - Befehl `\textsuetterlin` schaltet entsprechende Schrift ein
  - vorhandene Schriftarten: Kodierung `T1`

Familie	Stärke	Form	Beispiel
suetterl	m	n,it	Sütterlin Schrift

- **Initialen**
  - Kod. `U`,

Familie	Stärke	Form	Beispiel
yinit	m	n	Initialen

PostScript-Schriften

- PostScript: Seitenbeschreibungssprache
- 35 PostScript-Standardschriften sind meistens in den Druckern eingebaut
- PostScript-Schriften sind frei skalierbar
- METAFONT muß nicht benutzt werden
- unter LaTeX 2_ε stehen Erweiterungspakete für den Gebrauch der PostScript-Schriften bereit
- Erweiterungspakete ändern die Default-Schriften für Roman, Sans Serif und Typewriter
- Zuordnung [L2b]:

ps_stil	\rmfamily	\sffamily	\ttfamily
avant		Avant Garde	
bookman	Bookman	Avant Garde	Courier
helvet		Helvetica	
newcent	New Century Schoolbook	Avant Garde	Courier
palatino	Palatino	Helvetica	Courier
times	Times	Helvetica	Courier

- Pakete überschreiben vorherige Einstellung nur an den nicht leeren Feldern der Tabelle
- Beispiel: `\usepackage{times,avant}`
- weitere Pakete vorhanden, aber PostScript-Schriften nur kommerziell

- PostScript-Schriften und Formeln
  - Formeln werden weiterhin in Computer Modern Schriften gesetzt
  - PostScript-Schriften enthalten nicht alle benötigten Symbole
  - Erweiterungspaket **mathptm**
    - entnimmt den PostScript Schriften Symbol, Zapf-Chancery und Times die verfügbaren Symbole
    - restliche Symbole aus den CM-Schriften
    - sollte im Zusammenhang mit `times` benutzt werden
  - CM-Schriften werden für verschiedene Größen entworfen, PostScript-Schriften nicht
  - Skalierung von Symbolen problematisch
    - große Symbole werden zu fett

Benutzereigene Strukturen

- Definition eigener Makros und Umgebungen
- benutzen häufig Zähler und Längenmaße

Zähler

- Zähler sind
  - Variablen, die mit bestimmten Befehlen definiert, gesetzt, erhöht und abgerufen werden können
  - Wert ist eine ganzzahlige und im allgemeinen nicht negative Zahl
- LaTeX definiert selber Zähler:

part	chapter	paragraph	figure	enumi
	section	subparagraph	table	enumii
	subsection	page	footnote	enumiii
	subsubsection	equation	mpfootnote	enumiv

- Benutzereigene Zähler
  - Befehl `\newcounter`{*zähler_name*}[*rücksetzer*]
  - *rücksetzer* ist ein anderer Zähler, bei dessen Erhöhung der neue Zähler auf Null zurückgesetzt wird
  - sollte nur im Vorspann auftreten, nicht in `\include`-Dateien

- Zähler verändern:
  - Befehl `\setcounter`{*zähler*}{*wert*}
    setzt den Zähler auf einen neuen Wert
  - Befehl `\addtocounter`{*zähler*}{*wert*}
    erhöht den Zähler und *wert*, *wert* darf auch negativ sein
  - Befehl `\stepcounter`{*zähler*}
    erhöht den Zähler um 1, alle davon abhängigen Zähler werden zurückgesetzt
  - Befehl `\refstepcounter`{*zähler*}
    entspricht vorhergehenden Befehl, zusätzlich kann mit `\ref` der Zähler referenziert werden
  - Befehl `\value`{*zähler*}
    ruft den Wert des Zählers ab, üblich in Verbindung mit `\setcounter`:
    `\setcounter{mypage}{\value{page}}`
- Ausgabe eines Zählers
  - Befehle:
    `\arabic`{*zähler*}, arabische Ziffer
    `\Roman`{*zähler*}, große römische Ziffer
    `\roman`{*zähler*}, kleine römische Ziffer
    `\alph`{*zähler*}, kleiner Buchstabe
    `\Alph`{*zähler*}, großer Buchstabe
    `\fnsymbol`{*zähler*}, Fußnotensymbole
  - Befehl `\the`*zähler* ist für viele Zähler definiert, entspricht im wesentlichen `\arabic`{*zähler*}

Längen

- Befehl `\setlength`{*längen_befehl*}{*maß*}
  weist *längen_befehl* eine neue Länge zu (elastisch oder fest)
- als Maß darf wiederum eine Längenbefehl angegeben werden
- Dezimalzahl vor diesem Längenbefehl gilt als Faktor
  Beispiel:
  `\setlength{\rightmargin}{0.5\leftmargin}`
- Befehl `\addtolength`{*längen_befehl*}{*maß*}
  erhöht den Wert des Längenbefehls, *maß* darf auch negativ sein
- Befehl `\settowidth`{*längen_befehl*}{*text*}
  setzt Wert des Längenbefehls auf Breite des Textes *text*
- Befehl `\newlength`{*neuer_längen_befehl*}
  definiert einen neuen Längenbefehl
- Befehl `\stretch`{*dezimalzahl*}
  erzeugt elastische Länge, die einem durch *dezimalzahl* bestimmten Vielfachen von `\vfill` entspricht

Eigene Befehle

- werden definiert durch:
  `\newcommand`{*befehl*}[*arg*]{*definition*}
  `\renewcommand`{*befehl*}[*arg*]{*definition*}
- `\renewcommand` definiert einen bereits bestehenden Befehl neu
- *arg* gibt die Anzahl der Übergabeparameter an (0, ..., 9)
- Befehle ohne Parameter
  - für wiederkehrende Folge von LaTeX-Befehlen oder Texten
  - Beispiel:
    ```
    \newcommand{\xvec}%
               {\mbox{$x_1,\ldots,x_n$}}
    ```
    erzeugt beim Aufruf `\xvec` → $x_1, \ldots, x_n$
  - `\mbox` erlaubt, daß Befehl im Mathematikmodus und im normalen Text benutzt werden darf
  - Befehle mit Parameter
    - Anzahl der Parameter wird über *arg* angegeben
    - in der Definition ist `#`*num* Platzhalter für Parameter *num*
    - nur zwingende Parameter möglich[4]
    - Beispiel:
      ```
      \newcommand{\xvec}[1]%
                {\mbox{$#1_1,\ldots,#1_n$}}
      ```
      erzeugt beim Aufruf `\xvec{y}` → $y_1, \ldots, y_n$

[4] in LaTeX 2_ε ist jetzt auch die Definition eines optionalen Parameters möglich, ...

Eigene Umgebungen

- werden definiert durch:
  `\newenvironment`{*umgebung*}[*arg*]{*begdef*}{*enddef*}
  `\renewenvironment`{*umgebung*}[*arg*]{*begdef*}{*enddef*}
- *umgebung* steht für einen beliebigen Namen, unter dem die Umgebung aufgerufen werden kann
- *arg* ist die Anzahl der Parameter
- *begdef* sind Definitionen, die am Anfang der Umgebung durchlaufen werden
- *endef* sind Definitionen, die am Ende der Umgebung durchlaufen werden
- Beispiel:
  ```
  \newenvironment{sitquote}%
  {\begin{quote}\small\itshape}{\end{quote}}
  ```
  erzeugt eine modifizierte `quote`-Umgebung, deren Text in kleiner und kursiver Schrift gesetzt wird
- Definitionsbefehl darf zwischen den Parametern keinen weiteren Text enthalten, daher ist der Zeilenumbruch als Kommentar maskiert

Layout-Gestaltung

Einstellungen über

- Aufteilung der Seite
- laufende Kopf- und Fußzeile der Seite
- Parameter für Abstände und Größen

Aufteilung der Seite

- zweiseitiger Druck → Aufteilung für gerade und ungerade Seite unterschiedlich
- Elemente:

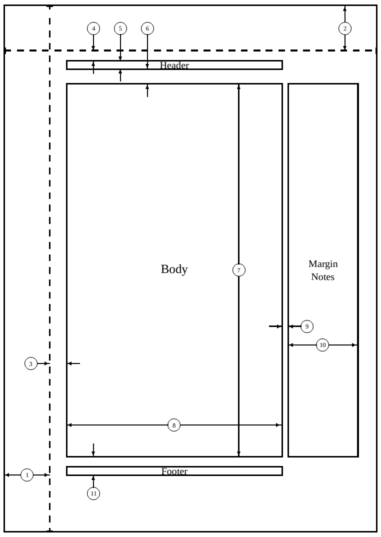
Body	enthält den Text der Seite
Header	laufende Seitenüberschrift der Seite
Footer	laufende Seitenunterschrift der Seite
Margin	äußerer Seitenrand

- Optionen bei `\documentclass`:

twoside	Formatierung für zweiseitigen Druck
a4paper	Layout für Seitengröße DIN A4 (210 x 297 mm)
a5paper	Layout für Seitengröße DIN A5 (148 x 210 mm)
b5paper	Layout für Seitengröße DIN B5 (176 x 250 mm)

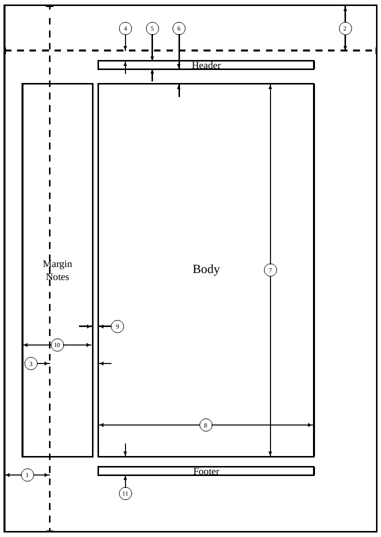
- Paket **layout**
  - definiert Befehl `\layout`
  - fügt Darstellung des Layouts in das Dokument ein
  - siehe Abbildungen auf den beiden nachfolgenden Folien
  - Maßangaben nur in Punkte `pt`
  - Umrechnung:
    maß_cm $= \frac{2.54}{72.27} \bullet$ maß_pt $\sim 0.0351 \bullet$ maß_pt

Aufteilung der ungeraden Seite (a4paper)



1	`one inch + \hoffset`	2	`one inch + \voffset`
3	`\oddsidemargin = 28.0pt`	4	`\topmargin = 17.0pt`
5	`\headheight = 12.0pt`	6	`\headsep = 25.0pt`
7	`\textheight = 598.0pt`	8	`\textwidth = 345.0pt`
9	`\marginparsep = 11.0pt`	10	`\marginparwidth = 111.0pt`
11	`\footskip = 30.0pt`		`\marginparpush = 5.0pt (not shown)`
	`\hoffset = 0.0pt`		`\voffset = 0.0pt`
	`\paperwidth = 597.50787pt`		`\paperheight = 845.04684pt`

Aufteilung der geraden Seite (a4paper)



1	`one inch + \hoffset`	2	`one inch + \voffset`
3	`\evensidemargin = 79.0pt`	4	`\topmargin = 17.0pt`
5	`\headheight = 12.0pt`	6	`\headsep = 25.0pt`
7	`\textheight = 598.0pt`	8	`\textwidth = 345.0pt`
9	`\marginparsep = 11.0pt`	10	`\marginparwidth = 111.0pt`
11	`\footskip = 30.0pt`		`\marginparpush = 5.0pt (not shown)`
	`\hoffset = 0.0pt`		`\voffset = 0.0pt`
	`\paperwidth = 597.50787pt`		`\paperheight = 845.04684pt`

- Dimensionsparameter, (*Werte für a4paper*)

`\textheight`	Höhe des Textkörpers (Body), ***21cm***
`\textwidth`	Breite des Textkörpers, ***12.1cm***
`\comlumnsep`	Spaltenabstand bei Mehrspaltensatz,
`\comlumnseprule`	Breite der Spaltentrennlinie, ***0pt***
`\evensidemargin`	linker Rand auf der geraden Seite, ***1cm***
`\oddsidemargin`	rechter Rand auf der ungeraden Seite, ***2.78cm***
`\footskip`	Abstand Grundlinie letzte Zeile und erste Zeile Fußzeile, ***1.05cm***
`\headheight`	Höhe der Kopfzeile, ***0.42cm***
`\headsep`	Abstand zwischen Kopfzeile und Textkörper, ***0.88cm***
`\topmargin`	vertikaler Abstand Blattrand zu Kopfzeile, ***0.6cm***
`\marginparpush`	vertikaler Abstand zweier Randbemerkungen, ***5pt***
`\marginparsep`	horizontaler Abstand zwischen Text und Randbemerkungen, ***0.38cm***
`\marginparwidth`	Breite der Randbemerkungen, ***3.9cm***

- Druckertreiber
  - setzen Bezugspunkt für TEX ein Inch (***2.54cm***) unterhalb und rechts neben der oberen Ecke
  - Parameter zum Verschieben des Bezugspunktes

`\voffset`	vertikaler Abstand Blattrand zu Bezugspunkt, ***0cm***
`\hoffset`	horizontaler Abstand Blattrand zu Bezugspunkt,***0cm***

- Beispiel für eigene Einstellungen (im Vorspann):

```
\setlength{\voffset}{-1in}
\setlength{\hoffset}{-1in}
\setlength{\textwidth}{15cm}
\setlength{\oddsidemargin}{3.5cm}
\setlength{\evensidemargin}{2.5cm}
\setlength{\topmargin}{1.6cm}
\setlength{\headheight}{0.8cm}
\setlength{\headsep}{0.6cm}
\setlength{\textheight}{23cm}
\setlength{\footskip}{2.3cm}
```


laufende Über- oder Untertitel der Seite

- Seitenstil
  - verändert laufende Seitenüber- und unterschrift
  - Befehle: `\pagestyle`{*stil*}, `\thispagestyle`{*stil*}
  - *stil*:

empty	Kopf- und Fußzeile sind leer
plain	Kopfzeile leer, Fußzeile enthält zentrierte Seitennummer
headings	Kopfzeile enthält Informationen über Kapitel und Seitennumer, Fußzeile ist leer
myheadings	selbstdefinierte Kopf und Fußzeile

- Seitennummer
  - `\pagenumbering`{*stil*} definiert Stil neu und setzt Seitennummer auf 1 zurück
  - *stil*: *Alph* (A,B,C,...), *alph* (a,b,c,...), *Roman* (I,II,III,...), *roman* (i,ii,iii,...), *arabic* (1,2,3,...)
- Beispiel für Frontteil, Hauptteil

```
\documentclass[...]{book}
...
\begin{document}
\title{...} ...
\pagenumbering{roman}
\maketitle
... (Copyright, Vorwort, ...)
\cleardoublepage
\tableofcontents
\cleardoublepage
\listoffigures
\cleardoublepage
\pagenumbering{arabic}
... (erstes Kapitel)
\cleardoublepage
... (zweites Kapitel)
\end{document}
```

Paket **fancyheadings**

- dreiteilige Kopf- und Fußzeile
- Linien für Kopf- und Fußzeile
- Kopf- und Fußzeile können breiter als die Textbreite sein
- mehrzeilige Kopf- und Fußzeile
- verschiedene Kopf- und Fußzeilen für gerade und ungerade Seite
- andere Kopf- und Fußzeile für erste Seite des Kapitels
- Befehl: `\pagestyle{fancy}`
- Definition der Kopf- und Fußzeile

LK-gerade	MK-gerade	RK-gerade
	gerade Seite	
LF-gerade	MF-gerade	RF-gerade

LK-ungerade	MK-ungerade	RK-ungerade
	ungerade Seite	
LF-ungerade	MF-ungerade	RF-ungerade

```
\lhead[LK-gerade]{LK-ungerade}
\chead[MK-gerade]{MK-ungerade}
\rhead[RK-gerade]{RK-ungerade}

\lfoot[LF-gerade]{LF-ungerade}
\cfoot[MF-gerade]{MF-ungerade}
\rfoot[RF-gerade]{RF-ungerade}
```


- vordefinierte Befehle
  - `\thepage`
    fügt Seitennumer ein
  - `\leftmark`
    vom Dokumentstyle vorgesehener Text für linke Seite (i.A. höchste Kapitelüberschrift (chapter bei book)
  - `\rightmark`
    vom Dokumentstyle vorgesehener Text für rechte Seite (i.A. zweithöchste Kapitelüberschrift (section bei book)
- Begrenzungslinien
  - `\headrulewidth`
    Dicke der Linie unter der Kopfzeile (Default: 0.4pt)
  - `\footrulewidth`
    Dicke der Linie über der Fußzeile (Default: 0pt)
    `\setlength{\footrulewidth}{1pt}`
- Breite der Kopf- und Fußzeile
  `\addtolength{\headwidth}{\marginparsep}`
  `\addtolength{\headwidth}{\marginparwidth}`
  → Kopf- und Fußzeile gehen bis über die Randnotizen
- Kopf- und Fußzeile auf der ersten Seite eines Kapitels
  - müssen bei `\ihead` und `\ifoot` einzeln definiert werden:
    `\pagestyle{fancyplain}`
    `\fancyplain{`*plain_wert*`}{`*normal_wert*`}`
    plain_wert → erste Seite
    normal_wert → restliche Seiten `\plainheadrulewidth` und `\plainfootrulewidth` für Linien
- mehrzeiliger Text
  Text wird in einer `\parbox` gesetzt
  → Zeilenumbruch mit `\\`

- Beispiel (Layout des LaTeX-Buches von Leslie Lamport)

```
\documentclass{book}
\usepackage{fancyheadings}
\pagestyle{fancyplain}
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
% Kapitel"uberschrift merken
\renewcommand{\chaptermark[1]%
            {\markboth{#1}{}}
\renewcommand{\sectionmark[1]%
            {\markrigth{\thesection #1}}

\lhead[\fancyplain{}{\bfseries\thepage}}]%
      {\fancyplain{}{\bfseries\rigthmark}}%
\rhead[\fancyplain{}{\bfseries\leftmark}}]%
      {\fancyplain{}{\bfseries\thepage}}%
\cfoot{}
```

(aus „Der LaTeX-Begleiter“ [L3a])

Parameter für Abstände und Größen

- Dimensionsparameter im Vorspann oder Body
  - `\baselineskip`
    Abstand zwischen zwei Zeilen, Grundlinie zu Grundlinie, abhängig von der Schriftgröße, sollte nicht geändert werden
  - `\baselinestretch`
    Faktor, mit dem `\baselineskip` intern multipliziert wird
    `\renewcommand{\baselineskip}{2.0}` → doppelter Zeilenabstand
  - `\parskip`
    Abstand zwischen zwei Absätzen,
    `\setlength{\parskip}{1.5ex plus0.5ex minus0.5ex}`
    sollte von der Schriftgröße abhängig sein
  - `\parindent`
    Einzug der ersten Zeile eines Absatzes
    `\setlength{\parindent}{5mm}`
  - `\noindent`
    erste Zeile des nachfolgender Absatzes wird nicht eingerückt
- Numerierung von Kapitelüberschriften
  - secnumdepth
    Tiefe, bis zu der Überschriften numeriert werden sollen
    z.B.: `\setcounter{secnumdepth}{1}`
  - tocdepth
    Tiefe, bis zu der Überschriften im Inhaltsverzeichnis erscheinen werden sollen
    z.B.: `\setcounter{tocdepth}{2}`

Drucken

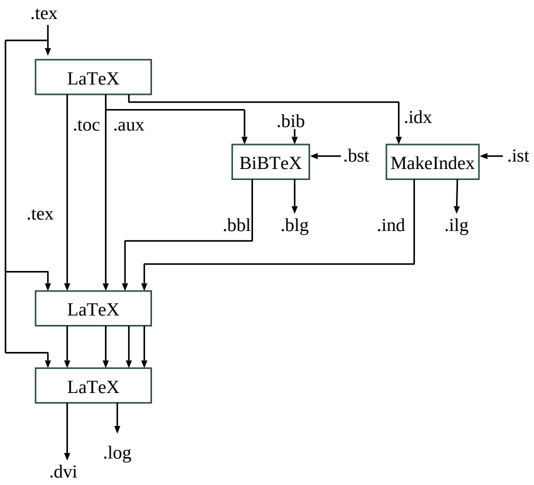
Generieren der PostScript-Dateien für

- zentrale Ausgabe im TIA
  `dvips -Ptia00pd test.dvi -o`
  → `test.ps`
  direktes Drucken:
  `dvips -Ptia00pd test.dvi`
  → `lpr -Ptia00pd test.ps`
  Aufteilen in mehrere Ausgabedateien (20 Seiten pro Datei)
  `dvips -Ptia00pd -o test.ps -S 20 -i test.dvi`
  → `test.001, test.002, ...`
  benutzt spezielle Fonts für den ausgewählten Drucker (`tia00pd`)
  Fremde mit `dvips` generierte PostScript-Dateien auf Drucker `zam13pd` ausgeben:
  `lpr -Pzam13pd fremd.ps`
- zentrale Ausgabe auf der Docutech im TIA-G
  `dvips -mode docutech -D 600 test.dvi -o`
  → `test.ps`
  `CLASS="Kostenstelle, Auflage, Einband, Termin, .."`
  `lpr -Ptia00px -C "$CLASS" test.ps`

Dokumentation:

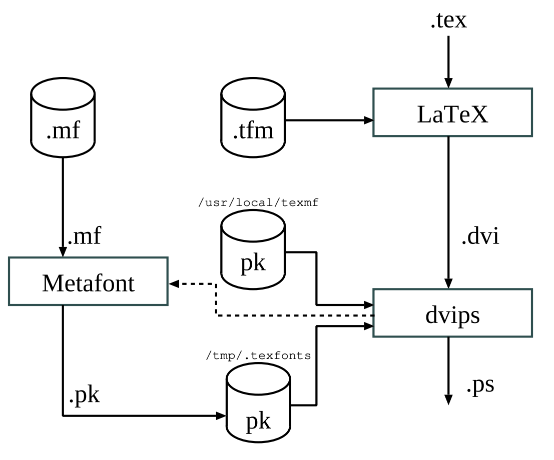
- **dvips**: `/usr/local/texmf/doc/dvips/dvips.dvi`
- TKI-0197: „Zentral unterstützte Ausgabe von Graphik- und Textdateien“

Flußdiagramm: LaTeX , BiBTeX und Index



- `.toc`: Informationen für Inhaltsverzeichnis
- `.aux`: Informationen über Verweise und Abbildungen
- `.bib`: Literatur-Datenbank
- `.bbl`: Literaturverzeichnis für dieses Dokument
- `.blg`: Log-Datei von BiBTeX
- `.bst`: Stil-Datei für BiBTeX
- `.idx`: unsortierte Indexeinträge (+ Seitenzahl)
- `.ind`: sortierte Indexeinträge (+ Seitenzahl)
- `.bst`: Stil-Datei für MakeIndex
- `.log`: Log-Datei von LaTeX

Flußdiagramm: Drucken



- `.mf` : Fontinformationen in Vektorformat (Linien, Splines)
- `.tfm`: Informationen über Größen der einzelnen Buchstaben
- `.pk` : Pixel-Informationen, abhängig von Größe und Drucker
- Metafont generiert für jeden Drucker spezielle Fonts
- `dvips` ruft Metafont (`mf`) für nicht vorhandene PK-Fonts auf (zeitaufwendig) und legt diese Fonts in `/tmp/.texfonts/` ab

Literatur

[L1] Leslie Lamport. LaTeX – A Document Preparation System, Addison–Wesley Co., Inc., Reading, MA, 1985

[L2] Helmut Kopka. LaTeX, Band 1–3, Addison Wesley (Deutschland) GmbH, Bonn, 1993

[L2a] Band 1: LaTeX-*Einführung*, 1993

[L2b] Band 2: LaTeX-*Ergänzungen*, 1994

[L2c] Band 3: LaTeX-*Erweiterungen — mit einer Einführung in METAFONT*, 1994

[L3] M. Goosens, F. Mittelbach, A. Samarin, The LaTeX Companion, Addison–Wesley Co., Inc., Reading, MA, 1994

[L3a] M. Goosens, F. Mittelbach, A. Samarin, Der LaTeX Begleiter, Addison–Wesley Co., Inc., Reading, MA, 1995

[L4] R. Wonneberger. Kompaktführer LaTeX, 3. durchgesehene Auflage. Addison–Wesley (Deutschland) GmbH, Bonn, 1988