Blockkurs LaTeX 2017

(nach einer Vorlage von Ralf Meyer, Göttingen)

Thomas Markwig

September 2017

Inhaltsverzeichnis

Webseite der Vorlesung

`http://www.math.uni-tuebingen.de/~keilen/Lehre/SS16/atss16de.html`

Teil I
Elemente der globalen Textgestaltung

1 Einführung

Word versus Latex

Frage 1. Wie unterscheiden sich Word und LaTeX?

Word
- "What you see is what you get!"
- Der Autor muß sich um den Inhalt *und* das Erscheinungsbild kümmern!

LaTeX
- Der Autor kümmert sich im wesentlichen *nur* um den Inhalt.
- LaTeX *setzt* den Text.
- *Der Preis*: mehrere Arbeitsschritte sind notwendig, um das Ergebnis zu sehen!

Die Arbeitsschritte

LaTeX ist eine Programmiersprache!

- *Schreiben* des Quellcodes in einem beliebigen Texteditor.
  - Z.B. `XEmacs`, `Kile`, `TeXnicCenter`, `TeXMaker`, `TeXStudio`
  - Ergebnis speichern als: `datei.tex`
- *Kompilieren* des Quellcodes mit LaTeX.
  - `latex` erzeugt `datei.dvi`
  - `dvips` erzeugt daraus `datei.ps`
  - `pdflatex` erzeugt direkt `datei.pdf`
- *Anschauen* des Ergebnisses mit geeigneten Programm.
  - Z.B. `xdvi` zum Betrachten von `dvi`-Dateien
  - Z.B. `ghostview` zum Betrachten von `ps`-Dateien
  - Z.B. `acroread` zum Betrachten von `pdf`-Dateien

2 Installation

Installation unter Linux

- Jede aktuelle Linux Installation sollte die LaTeX Installation *TeX Live* standardmäßig mitbringen.
- Als Editoren empfehlen sich

  - *Kile* (besonders wenn man wenig Programmiererfahrung besitzt) oder
  - *XEmacs* mit AUCTEX.

- Auf dem System sollten *xdvi*, eine Variante von *ghostview* und ein PDF-Viewer wie *acroread* installiert sein. Alternativen sind *evince* oder *okular*, die mit jedem der Formate umgehen können.

Sofern die Programme noch nicht installiert sind, sollte man sie mit dem bevorzugten Paketmanager nachinstallieren.

Installation unter Windows

- Unter Windows kann man zwischen *TEX Live* und *MiKTEX* (http://miktex.org) wählen.
- Als Editor empfiehlt sich hier *TEXnicCenter* (http://www.texniccenter.org).
- *Acrobat* sollte unter Windows standardmäßig da sein. Es empfiehlt sich aber, zudem *Ghostscript* (http://www.ghostscript.com/download) und *Ghostview* (http://pages.cs.wisc.edu/~ghost/gsview/get50.htm) zu installieren.

Achtung:
Man kann TEX Live und MiKTEX nicht parallel verwenden!

3 Grundaufbau einer LATEX–Datei

Eine Latexdatei besteht immer aus zwei Teilen, dem

Programmkopf
\documentclass{dokumentklasse}
⋮
⋮

und dem

Programmkörper
\begin{document}
⋮
⋮
\end{document}

Der Programmkopf ...

- beginnt immer mit dem Befehl *\documentclass{dokumentklasse}*, wobei dokumentklasse der Name einer LATEX bekannten Dokumentklasse ist. Z.B.:

  article Standard für kürzere Artikel

  beamer für Präsentationen wie diese

  dinbrief für Briefe gemäß DIN-Regeln

report Standard für längere Artikel

book Standard für Bücher

amsart Artikel in den Zeitschriften der American Mathematical Society (AMS)

- enthält *nie* Text oder Befehle, die Text erzeugen.
- enthält Befehle, die die globale Struktur des Textes festlegen.

Der Programmkörper ...

- beginnt immer mit *\begin{document}*.
- endet immer mit *\end{document}*.
- Enthält den eigentlichen Text des Dokumentes.

Ein erstes Beispiel

```
\documentclass{article}
    \begin{document}
        Mein erstes Beispiel!
    \end{document}
```

Wie erzeuge ich ein Dokument mit LaTeX?

1. Schritt Speichere den Text aus obigem Beispiel mit Hilfe eines beliebigen Texteditors in der Datei *beispiel.tex*.

2. Schritt Kompiliere die Datei entweder mit dem Befehl

```
latex beispiel.tex
```

oder mit dem Befehl

```
pdflatex beispiel.tex.
```

3. Schritt Zeige die Datei entweder mit dem Befehl

```
xdvi beispiel.dvi
```

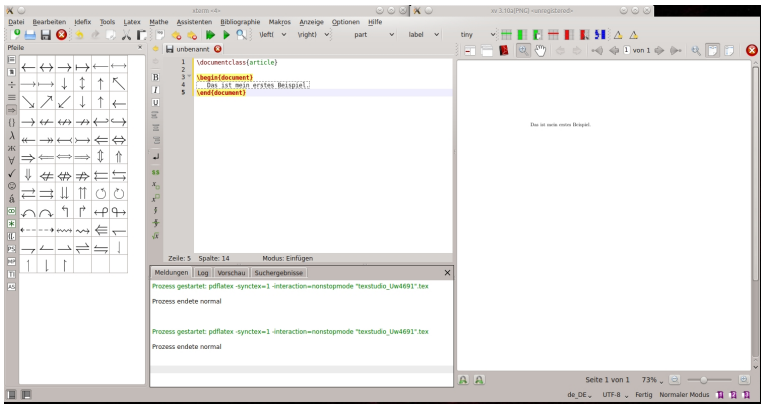
oder mit dem Befehl

```
acroread beispiel.pdf
```

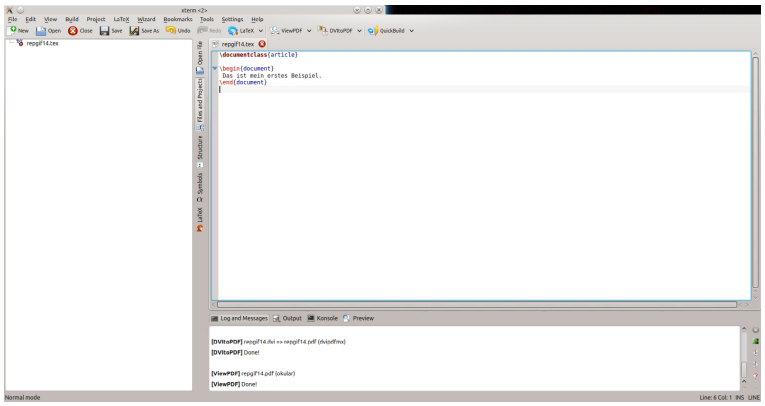
an.

Achtung:
Wer *Kile* oder *TeXnicCenter* verwendet, kann die Schritte 2 und 3 auch aus dem Editor heraus aufrufen.

TexStudio



Kile



4 LaTeX–Sonderzeichen

Die folgenden Zeichen haben in LaTeX eine Sonderfunktion:

\ # $ & ~ _ % { }

- \ leitet einen Befehle ein, z.B. \documentclass.
- { öffnet eine Gruppe
- } schließt eine Gruppe.
- *Gruppen* begrenzen die Wirkung von Befehlen und dienen dazu, Befehlen Argumente zu übergeben.
- Will man die obigen Sonderzeichen, im Text verwenden, muß man ihnen das Befehlzeichen \ voranstellen.

  Z.B. der Befehl \$ erzeugt das $-Zeichen.
- Das \-Zeichen bildet eine Ausnahme:

- Es wird durch den Befehl *\textbackslash* erzeugt,
- im Mathematikmodus durch den Befehl *\backslash*.
- \\ erzwingt einen Zeilenumbruch!

Die Bedeutung der anderen Sonderzeichen

`%` leitet Kommentare ein und veranlasst TEX, den gesamten Rest der Zeile zu ignorieren – einschließlich des abschließenden Newline-Charakters.

Dadurch kann man eine neue Zeile anfangen, ohne ein Leerzeichen zu erzeugen.

`$` schaltet den *Mathematikmodus* ein und aus

`#` wird *nur* bei der Definition von *Makros* benutzt

`^` wird *nur* im Mathematikmodus für *Exponenten* benutzt

`_` wird *nur* im Mathematikmodus für *Indices* benutzt

`&` markiert Tabulatorpositionen und Tabellenspalten

`~` geschütztes Leerzeichen (kein Zeilenumbruch)

5 Syntax eines Befehls

- Das Zeichen \ leitet *Befehle* ein.
- Der *Name* des Befehls besteht aus *Buchstaben*.
- Der erste Nichtbuchstabe (etwa eine Zahl, ein \ oder ein Leerzeichen) nach dem \ beendet den Namen des Befehls.
- Ein Befehl kann mehrere *optionale* und *verpflichtende* Argumente haben.
- Allgemeine Syntax:

  \befehlsname[optional]{argument1}{argument2}...

- *Optionale* Argumente stehen unmittelbar hinter dem Befehlsnamen in eckigen Klammern und sind durch Kommata voneinander getrennt.
- Es muß keine optionalen Argumente geben.
- *Verpflichtende* Argumente kommen nach den optionalen Argumenten in geschweiften Klammern.
- Jedes verpflichtende Argument hat ein eigenes Klammernpaar.
- Beispiele:
  - \documentclass[a4paper,reqno]{article} hat zwei optionale Argumente, ein verpflichtendes
  - \LaTeX hat kein Argument erzeugt den Text LATEX,

– \setlength{\textwidth}{12.5cm} hat zwei verpflichtende Argumente; legt die Textbreite auf 12.5cm fest

Vorsicht
Beendet ein Leerzeichen einen Befehlsnamen, so wird es im Dokument ignoriert. Dies wird durch eine *leere Gruppe* `{}` verhindert.

```
\LaTeX AA \LaTeX{} AA   ↦   LaTeXAA LaTeX AA
```

6 Umgebungen

- Eine Umgebung beginnt mit dem Befehl *\begin{umgebung}*.
- Sie endet mit dem Befehl *\end{umgebung}*.
- Umgebungen gruppieren Text und behandeln ihn gemäß den Umgebungsparametern.
- Beispiele für Umgebungen:

  document gruppiert den Textkörper des Dokumentes.

  center zentriert den gruppierten Text.

  quote rückt den Text beidseitig ein.

  math stellt den Mathematikmodus bereit.
- Wir werden im Laufe des Kurses sehr viele vordefinierte Umgebungen kennen lernen.

Die `verbatim`-Umgebung

- Um Quellcode *wörtlich* einzugeben, steht uns die *`verbatim`*-Umgebung zur Verfügung.
- Wörtlich heißt, LaTeX interpretiert Sonderzeichen und Befehle nicht.
- Die `verbatim`-Umgebung erzeugt einen eigenen Absatz.
- Innerhalb eines Absatzes kann auch der Befehl *`\verb`* benutzt werden.
- Direkt auf `\verb` folgt ein Markierzeichen, das das Ende des Codes bezeichnet.
- Zum Beispiel erzeugt `\verb+\item+` die Ausgabe `\item`.

Warnung
Wenn man diese Befehle mit der Klasse `beamer` in einer `frame`-Umgebung einsetzen will, dann muß die Umgebung das optionale Argument *`fragile`* haben!

7 Pakete

Pakete laden

- Man kann die Funktionalität von LaTeX erheblich erweitern, indem man zusätzliche Pakete einlädt.
- Pakete werden immer im Programmkopf eingeladen.
- Dazu dient der Befehl *\usepackage*.
- Einige Pakete:

  babel Unterstützung für verschiedene Sprachen.

  inputenc Unterstützung verschiedener Zeichenkodierungen.

  graphicx Unterstützung der Graphikeinbindung.

  amsthm Theoremumgebung der AMS.
- Manchen Paketen werden beim Laden zusätzliche optionale Argumente übergeben, z.B.

  \usepackage[ngerman]{babel}.

Wie finde ich Hilfe zu einem Paket?

- *Google!*
- Mit *texdoc* auf die interne Dokumentation zugreifen (was leider nicht immer funktioniert):

  texdoc graphicx

  Den Befehl unter Linux in einer Konsole eingeben, unter Windows im Command-Fenster (funktioniert nicht unbedingt).
- Eine weitere Alternative unter Linux, nach einer Datei zu suchen, ist der Befehl *locate*:

  locate babel

  Dies liefert die Pfadnamen zu allen im System bekannten Dateien, die `babel` enthalten. Vorteil: man muß den Dateinamen nicht genau kennen.

8 Sprache und Umlaute

Wahl der Sprache

- Verschiedene *Sprachen* verwenden ganz andere Regeln zur Worttrennung.
- Das *babel*-Paket wählt die Trennregeln der richtigen Sprache aus.

- Bei mehrsprachigen Texten sollte `babel` mit allen benötigten Sprachen aufgerufen werden, etwa durch

  \usepackage[british,ngerman]{babel}

- Dann ist zunächst `ngerman` als Sprache voreingestellt.
- Durch *\selectlanguage{british}* wird auf `british` umgestellt, durch *\selectlanguge{ngerman}* wird auf `ngerman` gewechselt.
- Für kurze Passagen in einer Fremdsprache:

  \foreignlanguage{british}{text in british}

Umlaute

Es gibt in LaTeX verschiedene Möglichkeiten, Umlaute zu erzeugen.

- Umlaute können durch Voranstellen von `\"` erzeugt werden, z.B.

  `\"a \"O`, aber `\ss` ↦ ä Ö ß

- Das Paket *babel* mit deutscher Sprachunterstützung macht *Umlaute* durch Voranstellen von `"` verfügbar, z.B.

  `"a "O "s` ↦ ä Ö ß

- Das Paket *inputenc* erlaubt es, im Editor gleich Umlaute einzugeben.
  - `inputenc` muß die verwendete Zeichenkodierung als optionales Argument übergeben werden.
  - Z.B.: *\usepackage[latin1]{inputenc}*
  - Z.B.: *\usepackage[utf8]{inputenc}*

9 Leerraum

Leerraum und Zeilenumbruch

- Die Bemessung des Leerraums zwischen Zeichen, Worten und Zeilen ist die eigentliche Aufgabe des Setzers.
- Entsprechend behandelt LaTeX Leerraum ganz anders als Textverarbeitungsprogramme.
- Eine *beliebige Anzahl* von aufeinanderfolgenden Space-, Tab-, und Newline-Zeichen innerhalb eines Absatzes fügt *einen* Leerraum ein.
- Dieser Leerraum hat eine natürliche Breite und kann in gewissen Grenzen schrumpfen oder wachsen.
- Beim Zeilenumbruch streckt und staucht LaTeX die Leerräume so, daß ein möglichst gleichmäßiger Blocksatz entsteht. Außerdem werden dabei noch verschiedene traditionelle Regeln des Buchdrucks beachtet.

Leerraum nach Satzzeichen

- Standardmäßig setzt LaTeX nach Satzzeichen einen größeren Leerraum als nach Buchstaben, damit die Textstruktur leichter zu erkennen ist.
- Dieses Verhalten wird durch den Befehl `\frenchspacing` abgeschaltet – dies ist der Standard, wenn die deutsche Sprache gewählt ist – und durch `\nonfrenchspacing` eingeschaltet – dies ist Standard, wenn Englisch als Sprache gewählt wird.
- Nach Abkürzungen wie z. B. oder bzw. soll ein normaler Leerraum folgen.
- Um LaTeX mitzuteilen, daß der folgende Leerraum ein gewöhnlicher ist und nicht ein Satzende markiert, wird dem Leerzeichen ein `\` vorangestellt, etwa `bzw.\␣`.
- Das Zeichen `~` erzeugt ein Leerzeichen der gleichen Art wie `\␣`, bei dem zusätzlich der *Zeilenumbruch verboten* ist.

Leerraum im Text von Hand einfügen

- Folgende Befehle erzeugen verschieden große Leerräume:

`\,`	3/18 eines Quad
`\:`	4/18 eines Quad
`\;`	5/18 eines Quad
`\!`	-3/18 eines Quad
`\␣`	ein Leerzeichen
`\quad`	ein Quad (M-Breite)
`\qquad`	zwei Quad

- `\hspace{1cm}` erzeugt Leerraum der Breite $1cm$; eine Vielzahl von Maßeinheiten ist erlaubt.
- Um Leerraum am Zeilenanfang zu erzwingen, verwendet man den Befehl `\hspace*` statt `\hspace`.
- Feste Maßeinheiten wie `cm` werden bei Änderungen der Schriftart und -größe nicht skaliert.
- Einheiten wie `ex` (x-Höhe) ändern sich mit dem Zeichensatz.

10 Worttrennung

Worttrennung

- Kann ein Absatz durch Anpassen des Leerraums nicht gut gesetzt werden, versucht LaTeX die *Worttrennung*.
- Wenn der Blocksatz einmal nicht gelingt, erzeugt LaTeX eine Warnung (Overfull box). Dann muss von Hand eingegriffen werden (Worte umstellen, Worttrennungen manuell einfügen, ...).

- Bei Verwendung von *pdfLaTeX* zusammen mit dem Paket *microtype* verschwinden fast alle overfull boxes.
- In modernen Zeichensätzen in der T1-Kodierung sind Umlaute Buchstaben, so daß LaTeX damit auch Worte mit Umlauten trennen kann.
- Durch *\usepackage[T1]{fontenc}* in der Präambel wird ein entsprechend kodierter Zeichensatz verwendet.
- Man lade dazu mit `\usepackage{lmodern}` die Schriftfamilie `lmodern`.

Manuelle Trennhilfe

- Mit dem Befehl `\-` werden die Trennstellen eines Wortes von Hand festgelegt.

  Beispiel 2. Schreiben wir "Kör\-per\-au\-to\-mor\-phis\-mus" statt "Körperautomorphismus", so findet LaTeX alle korrekten Trennstellen und keine falschen mehr.
- Der Befehl *\hyphenation* in der Präambel legt Trennausnahmen fest.

  Beispiel 3. `\hyphenation{Hil-bert-raum Hil-bert-raums}`

  Da LaTeX nichts von Grammatik versteht, müssen wir bei Bedarf alle deklinierten Formen eines Wortes angeben.

Bindestriche und Gedankenstriche

- Die Typographie unterscheidet vier Arten von Strichen:

  - *Bindestrich* `-`

  – Bis-Strich (10–12 Uhr), *deutscher Gedankenstrich* `--`

  — *englischer Gedankenstrich* `---`, wird in deutschen Texten in der Regel nicht verwendet

  − *Minuszeichen* (`-` im Mathematikmodus)
- Im Englischen werden sowohl – als auch — als Gedankenstriche verwendet.

11 Die Wahl der Schrift

Auswahl der Schriftfamilie

- Als Standard benutzt LaTeX die *Computer Modern*-Schriften von Donald Knuth. Dies ist eine ganze Familie von Dutzenden eng verwandter Schriften, die für TeX entwickelt wurden. Neben allen erdenklichen Varianten für gewöhnliche Zeichen

  $$\mathrm{fff\,F}\mathbf{\mathit{fff}}\ \mathrm{ff\,F}\mathbf{\mathit{fff}}\ \mathtt{f\,f\,fff}$$

  in verschiedenen Schriftgrößen gehören dazu auch noch Hunderte von Zeichen für den Mathematiksatz.

- *Times* und *Palatino* sind andere Schriftfamilien.

  Um im ganzen Dokument diese Schriften zusammen mit dazu passenden mathematischen Symbolen zu verwenden, fügt man im Programmkopf `\usepackage{mathptmx}` für Times ein und `\usepackage{mathpazo}` für Palatino.

Auswahl der Schriftgröße

- Die *optionalen Parameter* `10pt`, `11pt` und `12pt` des `documentclass`-Befehls legen die Schriftgröße fest.
- `\documentclass[11pt]{article}` setzt das gesamte Dokument in etwas größerer Schrift.
- In Fußnoten, Überschriften und mathematischen Formeln wird die Schriftgröße von LaTeX *automatisch* angepaßt.
- Wegen des Prinzips der Trennung von Form und Inhalt sollte man die Schriftgröße im Text *nicht* explizit ändern.
- Ausnahmen sind die Gestaltung von Titelseiten oder Kopf- und Fußzeilen, oder die Beschriftung von Bildern.
- Die entsprechenden Befehle lauten:

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

- Sie haben kein Argument: `{\large großer Text}`

Auswahl der Schriftart

Für jeden Schriftsatz gibt es unterschiedliche Formen, die sich nach *Familie*, *Variante* und *Stärke* unterscheiden. Aus standardmäßig drei Familien, fünf Varianten und zwei Stärken ergeben sich 30 Formen derselben Schrift.

Familien	Befehle		
Schrift mit Serifen	\rm	\rmfamily	\textrm
Schrift ohne Serifen	\sf	\sffamily	\textsf
`Maschinenschrift`	\tt	\ttfamily	\texttt
Varianten	**Befehle**		
gerade	\up	\upshape	\textup
kursiv	\it	\itshape	\textit
oblique	\sl	\slshape	\textsl
Kapitälchen	\sc	\scshape	\textsc
Hervorgehoben	\em	\emshape	\emph
Stärken	**Befehle**		
normal	\md	\mdseries	\textmd
fett	\bf	\bfseries	\textbf

- Die Befehle der ersten und zweiten Spalte werden wie die Befehle zur Schriftgröße ohne Argument angewendet und können mit diesen kombiniert werden: `\large\slshape Hallo` $\mapsto$ *Hallo*

- Vorsicht, die Befehle der ersten Spalte sind veraltet und legen z.T. mehr fest, als sie sollen!
- Die Befehle der dritten Spalte haben den Text, den sie beeinflussen, als verpflichtendes Argument: `\textbf{fett}` ↦ **fett**
- Die Befehle zur Kontrolle der Familie, der Variante und der Stärke können miteinander kombiniert werden, wenn sie in der Form der zweiten oder dritten Spalte verwendet werden: `\texttt{\textit{Hallo}}` ↦ *Hallo*

  Achtung

  Die TEX-Live Version von Ubuntu hat Probleme mit einigen Fontkombinationen und kann sie nicht darstellen!

12 Texthervorhebung

- Zur Texthervorhebung dient der Befehl `\emph{hervorgehoben}`, der sein Argument hervorhebt.
- In vielen Dokumentklassen macht `\emph` folgendes:
  - Meist wird der *hervorzuhebende Text* in *kursiver Schrift* gesetzt.
  - *Ist aber die aktuelle Schrift schon kursiv, so wird der* hervorzuhebende Text *in gerader Schrift gesetzt.*
- *Farbige* Hervorhebung ist ideal für am Bildschirm zu lesende Dokumente.
- **Fette** Schrift wird in LATEX standardmäßig für *Struktur*elemente wie Überschriften eingesetzt, nicht aber für die Hervorhebung innerhalb des Textes.

  Denn dies erzeugt ein **unregelmäßiges Schriftbild**, das ästhetisch wenig ansprechend ist und den Leser irritiert.

Teil II
Textgliederung und einige wichtige Umgebungen

13 Titelseite und Zusammenfassung

Titelseite

- Mit der *titlepage*-Umgebung kann man eine Titelseite frei gestalten.
- Für Zeitschriftenartikel gibt es Standardbefehle, die die üblichen Titeldaten erzeugen.
- In den Standardklassen wie `amsart` gibt es dafür die Befehle *\author*, *\title* und *\date*, sowie die Befehle *\thanks* für Fußnoten im Titel und den Befehl *\and* zum Trennen verschiedener Autoren.
- Nachdem die Titeldaten festgelegt sind, setzt *\maketitle* sie.

Warnung
Gerade bei den Titelangaben unterscheiden sich die verschiedenen Dokumentklassen. Jede Zeitschrift hat hier ihren eigenen Standard, und die entsprechenden LaTeX-Klassen benutzen jeweils andere Befehle.

Zusammenfassung

- Nach dem Titel folgt in der Regel eine *Zusammenfassung*, die mit der *abstract*-Umgebung erzeugt wird.
- In der Zusammenfassung sollten möglichst wenige Formeln verwendet werden, damit sie leicht in andere Formate (HTML, ...) zu konvertieren ist, und sie sollte auch für Nichtexperten möglichst verständlich sein.

Warnung
In vielen Klassen ist die Zusammenfassung Teil des Titelmaterials und kommt daher vor `\maketitle`.

In der Standardklasse `article` ist es genau umgekehrt.

14 Gliederung eines Textes

Gliederung eines Textes

- LaTeX stellt folgende Gliederungsebenen zur Verfügung:

  part Teil

  chapter Kapitel (nicht alle Dokumentklassen)

  section Abschnitt

subsection Unterabschnitt

subsubsection Unterunterabschnitt

paragraph Absatz

subparagraph Unterabsatz

- Meist reichen drei Gliederungsebenen aus.

Gliederungsbefehle

- Ein neuer Abschnitt wird eingeleitet mit `\section[Kurzüberschrift]{Überschrift}` oder `\section{Überschrift}`.
- Dies bewirkt unter anderem folgendes:
  - Der Abschnitt erhält eine fortlaufende *Nummer*, und diese erscheint zusammen mit der Überschrift in angemessener *Schriftart* und *-größe* in sinnvollem *Abstand* zum Rest des Textes.
  - Der Abschnitt wird ins Inhaltsverzeichnis eingetragen, wobei, wenn vorhanden, die Kurzüberschrift benutzt wird.
  - Falls dies der Dokumentstil vorsieht, wird die (Kurz)überschrift auf allen Seiten des Abschnitts im *Seitenkopf* angezeigt.
  - Falls Gleichungen, Theoreme, oder Ähnliches abschnittsweise numeriert werden, werden die entsprechende Zähler zurückgesetzt.
  - Mit gewissen Zusatzpaketen werden bei der Erzeugung von Dokumenten im PDF-Format passende *Hyperlinks* und *Bookmarks* erzeugt.
- Die anderen Gliederungsebenen werden ähnlich erzeugt, zum Beispiel erzeugt `chapter` Kapitel.
- Alle diese Befehle haben dieselbe Syntax.
- Die *-Variante der Gliederungsbefehle:
  - Statt `\section` kann man auch `\section*` verwenden.
  - Der Abschnitt wird dann *nicht* numeriert.
  - Der Abschnitt kommt *nicht* ins Inhaltsverzeichnis.
  - Dies ist insbesondere sinnvoll für Vorwort, Literaturverzeichnis oder Index.
- Will man bei der *-Variante, daß zwar die Numerierung unterdrückt wird, die Überschrift aber im Inhaltsverzeichnis erscheint, so muß man den Eintrag per Hand hinzufügen. Dies geschieht durch den Befehl

  `\addcontentsline{toc}{section}{Überschrift}`

15 Inhaltsverzeichnis

- LaTeX kann aus den Gliederungsbefehlen selbständig ein *Inhaltsverzeichnis* erstellen.
- Der Befehl *\tableofcontents* erzeugt an der Stelle seines Auftretens ein Inhaltsverzeichnis.
- LaTeX schreibt beim Kompilieren die Titel der Gliederungsebenen und ihre Nummern in eine Datei mit Endung `.toc`.
- Erst bei *erneutem* Kompilieren werden die aktuellen Daten aus der `toc`-Datei ins Dokument übernommen.
- Mit dem Zähler *tocdepth* wird reguliert, wie viele Gliederungsebenen im Verzeichnis aufgenommen werden.
- Z.B. *\setcounter{tocdepth}{2}* legt fest, daß die beiden obersten Ebenen aufgenommen werden.
- Welche Gliederungsebenen vorkommen, hängt von der gewählten Dokumentklasse ab. Bei `article` wären die beiden obersten Ebenen `section` und `subsection`.

16 Listen und Aufzählungen

Listen mit der `itemize`-Umgebung

- In LaTeX gibt es spezielle Umgebungen für Listen und Aufzählungen.
- Wir sehen hier gerade eine Liste, die mit folgenden Befehlen erzeugt wurde:

```
\begin{itemize}
\item In \LaTeX{} gibt es spezielle Umgebungen für Listen und
Aufzählungen.
\item Wir sehen hier gerade eine Liste, die mit folgenden
Befehlen ...
\end{itemize}
```

- Die `itemize`-Umgebung enthält die Aufzählung.
- Für jeden neuen Punkt verwenden wir den *\item*-Befehl.
- LaTeX kümmert sich um alles andere.

Aufzählungen mit der `enumerate`-Umgebung

1. Die *`enumerate`-Umgebung* erzeugt numerierte *Aufzählungen*.
2. Wir ersetzen einfach `itemize` durch `enumerate` und beginnen weiterhin jeden Punkt mit `\item`.
3. Geschachtelte Aufzählungen sehen in der Beamerklasse so aus:

   3.1 Erster Unterpunkt

   3.2 Zweiter Unterpunkt

   3.2.1 Erster Unterunterpunkt

   3.2.2 Zweiter Unterunterpunkt
4. In den meisten Dokumentklassen wird die Numerierung durch die Befehle `\labelenumi`, `\labelenumii`, usw. verändern. Z.B.

   $$\renewcommand{\labelenumi}{\alph{enumi}.}$$

   legt fest, daß die erste Verschachtelungsstufe mit Buchstaben (`\alph`) gefolgt von einem Punkt numeriert werden sollen (also a., b., ...).

Freie Listen mit der `description`-Umgebung

- Eine dritte Art von Liste wird durch die *`description`-Umgebung* erzeugt:

  Beispiel 4.
  ```
  \begin{description}
   \item[Autor] Tick
   \item[Gestalter] Trick
   \item[Setzer] Track
  \end{description}
  ```

  Autor Tick

  Gestalter Trick

  Setzer Track
- Die eckigen Klammern begrenzen ein optionales Argument für den `item`-Befehl.

* Dieses optionale Argument kann auch in gewöhnlichen Listen benutzt werden, wie hier `\item[*]`.

Das ist nur bei sehr kurzen Markierern sinnvoll.

17 Tabellen mit der `tabular`-Umgebung

- Die *`tabular`*-Umgebung erzeugt eine Tabelle.
- Sie hat ein optionales Argument (`m`, `t` oder `b`) zur *Ausrichtung* der Tabelle:

  ***m*iddle** mittig

  ***t*op** nach oberster Zeile

***b*ottom** nach unterster Zeile

- Sie hat ein verpflichtendes Argument zum *Format*.
- Durch das Format *ccrlp{4cm}* werden fünf Spalten erzeugt:
  - die ersten beiden werden zentriert,
  - die dritte ist rechtsbündig,
  - die vierte ist linksbündig,
  - die fünfte wird im Blocksatz zu einer Breite von 4 cm gesetzt.
- Beim Eingeben der Tabelle trennen wir die Spalten jeweils durch `&` und die Zeilen durch `\\`.

Ein Beispiel für eine Tabelle

Beispiel 5.
```
\begin{tabular}[m]{ccl}
  1 & Dreieck & hat drei Ecken\\
  2 & Viereck & hat vier Ecken\\
  3 & Fünfeck & hat fünf Ecken
\end{tabular}
```

- Im *verpflichtenden* Argument der `tabular`-Umgebung sind noch weitere Formatierungsanweisungen erlaubt.
- Durch | erhalten wir einen vertikalen Strich in unseren Tabellen zum Trennen der Spalten.
- Normalerweise werden die Spalten durch einen gewissen Leerraum getrennt. Diesen kann man durch einen beliebigen Text ... ersetzen durch *@{...}*.

Beispiel 6.
```
\begin{tabular}[m]{@{ Ein }c@{ hat }l@{ Ecken.}}
  Dreieck & drei \\
  Viereck & vier
\end{tabular}
```

Ergebnis
Ein Dreieck hat drei Ecken.
Ein Viereck hat vier Ecken.

- Durch *\hline* erzeugen wir einen horizontalen Strich.
- Durch *\cline{2-3}* erzeugen wir einen Strich nur unter den Spalten 2–3.
- Durch *\multicolumn{3}{c}{Text}* werden drei Spalten der Tabelle zusammengefaßt und der Text wird zentriert.
- Zusatzpakete definieren auch einen *multirow*-Befehl.
- Die Umgebung *array* im Mathematikmodus verhält sich wie `tabular`, nur daß ihre Einträge im Mathematikmodus gesetzt werden.

Beispiel 7.
```
\begin{tabular}[m]{|c|c|l|}
  \hline
   Nr.& \multicolumn{2}{c|}{Ecken}\\\hline\hline
   1 & Dreieck & hat drei Ecken\\\hline
   2 & Viereck & hat vier Ecken\\\hline
\end{tabular}
```

Ergebnis

Nr.	Ecken	
1	Dreieck	hat drei Ecken
2	Viereck	hat vier Ecken

18 Mathematische Sätze

- In manchen Dokumentklassen sind schon Umgebungen für mathematische Sätze vordefiniert.
- Zum Beispiel erzeugt (mit der Klasse `beamer`):

  ```
  \begin{Satz}[Bolzano-Weierstraß]
    Jede beschränkte unendliche Menge reeller  Zahlen besitzt
  einen Häufungspunkt.
  \end{Satz}
  ```

  Satz 8 (Bolzano-Weierstraß)**.** *Jede beschränkte unendliche Menge reeller Zahlen besitzt einen Häufungspunkt.*

- Ohne das optionale Argument `[Bolzano-Weierstraß]` erhalten wir die Standardform

  Satz 9. *Jede beschränkte unendliche Menge reeller Zahlen besitzt einen Häufungspunkt.*

Beweisumgebung

- Die `proof`-Umgebung benutzen wir für Beweise:
  ```
  \begin{proof}
    Dies ist trivial.
  \end{proof}
  ```

  Beweis. Dies ist trivial. □

- Diese Umgebung ist in vielen Klassen vordefiniert, ansonsten stellt sie das Paket `amsthm` zur Verfügung.
- Ein optionales Argument ist möglich:

  ```
  \begin{proof}[Beweis des Satzes von BW]
    Dies ist trivial.
  \end{proof}
  ```

  Beweis des Satzes von BW. Dies ist trivial. □

- Das Zeichen □ für das Befehlsende kann mit *\qedhere* an einer anderen Stelle plaziert werden.
- Das Wort *Beweis* hängt von der gewählten Sprache ab.

Der Befehl newtheorem

- In den Standardklassen von LaTeX sind noch keine mathematischen Sätze vordefiniert.
- Sie werden in der Präambel durch `newtheorem` definiert.
- *\newtheorem{meintheorem}{Hauptsatz}* definiert eine neue Umgebung mit Namen `meintheorem`.

  Sätze dieser Art werden im ganzen Dokument fortlaufend numeriert in der Form *Hauptsatz 1*.
- `\newtheorem{meintheorem}{Hauptsatz}`*[section]*

  Diese Art von Satz wird abschnittsweise numeriert, der erste in Abschnitt 3 heißt also *Hauptsatz 3.1*.
- `\newtheorem{meintheorem}`*[theorem]*`{Hauptsatz}`

  Diese Art von Satz wird mit demselben Zähler numeriert wie die `theorem`-Umgebung.

  Dies ist sinnvoll, weil es den Leser verwirrt, wenn Theorem 1 zwischen Lemma 25 und Satz 7 steht.

Gestaltung von Sätzen

- Es gibt mindestens zwei Arten von mathematischen Sätzen, die typographisch anders zu behandeln sind:

  Sätze Name fett, Text kursiv

  Definitionen Name fett, Text normal

  Bemerkungen Name kursiv, Text normal.
- Genauer gesagt hängen die Details der Schriftwahl von der Dokumentklasse ab.
- Ich persönlich verwende für Bemerkungen den gleichen Stil wie für Definitionen.
- Ist das Paket *amsthm* geladen, so wählen die *theoremstyle*-Befehle aus, von welcher Art jeweils die nächsten deklarierten Theoreme sind.

Theoremdeklarationen im Programmkopf

```
\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{lemma}[theorem]{Lemma}
\newtheorem{corollary}[theorem]{Corollary}
\theoremstyle{definition}
\newtheorem{definition}[theorem]{Definition}
\newtheorem{remark}[theorem]{Remark}
\newtheorem{example}[theorem]{Example}
```

- *Alle* Sätze benutzen den gleichen Zähler `theorem` wie mathematische Gleichungen: Auf Satz 1 folgt Lemma 2, Beispiel 3, Definition 4, Lemma 5, ...

Anpassen der Satznumerierung

- Im Programmkopf legen wir fest, wie Theoreme *gestaltet* und *gezählt* werden sollen.

  Im Haupttext beschreiben wir dann Sätze rein *funktional*. Gestaltung und Numerierung geschehen automatisch.

- Durch Ändern der `newtheorem`-Befehle im Programmkopf wird das Erscheinungsbild von Sätzen im ganzen Dokument *einheitlich* geändert.

- Dies ist gut so, weil verschiedene Zeitschriften ihren eigenen Stil haben und von ihren Autoren erwarten, sich daran zu halten.

19 Textbezüge

Textbezüge

- Der LaTeX-Befehl *`\label{Marke}`* erzeugt eine (unsichtbare) Markierung im Text.

- Durch *`\pageref{Marke}`* erhält man die Nummer der *Seite* auf der diese Markierung steht, hier 23.

- Durch *`\ref{Marke}`* erhält man den Wert eines *Zählers*, hier zum Beispiel 57; je nach Zusammenhang ist das die Nummer des Kapitels oder aktuellen Unterabschnitts, eines Satzes, eines Punktes in einer Aufzählung, einer Tabelle oder Abbildung.

- Es empfiehlt sich, *immer* gleich zu Beginn jedes Abschnitts und jedes mathematischen Satzes einen `label`-Befehl mit einem deskriptiven Namen unterzubringen.

20 Literaturverzeichnis

- Die Umgebung *thebibliography* enthält das Literaturverzeichnis.
- Sie hat den Nachteil, daß die Einträge von Hand formatiert werden müssen. Dies ist lästig, wenn wir das Format ändern wollen.
- Zwei Ergänzungen zu LaTeX beheben dieses Problem:

  bibtex benutzt eine externe Datei und eine eigene Sprache, um die notwendigen Daten für die Literaturliste zu sammeln.

  amsrefs erweitert die LaTeX-Syntax, so daß diese Daten direkt in der LaTeX-Datei *funktional* beschrieben werden können.
- Aus dem *Zentralblatt*, *MathSciNet* und dem Katalog der UB erhalten Sie leicht vollständige Einträge im BibTeX- oder amsrefs-Format für mathematische Veröffentlichungen.

BibTeX

- Man sollte *BibTeX* verwenden, wenn man auf gewisse Literatur immer wieder in seinen Arbeiten verweisen muß.
- Dazu speichert man sein Literaturverzeichniseinträge in einer Datei mit der Endung `.bib`, z.B. `lit.bib`.
- Das Literaturverzeichnis wird dann mit dem Befehl `\bibliography{lit}` erzeugt, der auf `lit.bib` zugreift.
- Man *muß* den Stil, in dem die Literatur im Verzeichnis angezeigt wird, durch den Befehl `\bibliographystyle` steuern, z. B. `\bibliographystyle{amsalpha}`.
- Anschließend muß man einmal LaTeX, dann einmal *bibtex*, dann noch zweimal LaTeX auf die LaTeX-Datei anwenden, um das Literaturverzeichnis zu erhalten.
- Manche Programme, z. B. TeXnicCenter, führen `bibtex` automatisch aus.
- BibTeX legt dabei eine Datei mit der Endung `.bbl` an, die die `thebibiography`-Umgebung enthält.

Ein typischer BibTeX-Eintrag – Artikel

```
@Article{Kei04,
  author =    {Keilen, Thomas},
  title =     {Smoothness of Equisingular Families},
  journal =   {Trans. Amer. Math. Soc.},
  year =      {2005},                                  }
  volume =    357,
  number =    6,
  pages =     {2467-2481}
```

Das Kürzel `Kei04` ist dann eine Marke. Auf diese kann mit dem Befehl *\cite{Kei04}* verwiesen werden.

Ein typischer BibTEX-Eintrag – Buch

```
                author =      {Beutelspacher, Albrecht},
                title =       {{Lineare Algebra}},
@Book{Beu00,    publisher =   {Vieweg},                    }
                year =        2000,
                edition =     4
```

Das Kürzel `Beu00` ist dann eine Marke. Auf diese kann mit dem Befehl *\cite{Beu00}* verwiesen werden.

Varianten von `\cite`

- Der `cite`-Befehl verträgt auch einen optionalen Parameter. Ein Verweis auf Theorem 2.1 in der obigen Arbeit geschieht durch `\cite[`*`Theorem 2.1`*`]{Kei04}`.
- Wegen der Unmöglichkeit, optionale Argumente zu verschachteln, führt dies manchmal zu Problemen.
- In `amsrefs` werden deshalb optionale Argumente mit einer anderen Syntax eingegenben: `\cite{Kei04}*{`*`Theorem 2.1`*`}`.
- Es ist auch möglich, mehrere Werke auf einmal zu zitieren: übergeben Sie dafür dem `cite`-Befehl eine durch Kommata getrennte Liste von Einträgen im Literaturverzeichnis.
- Bei Verwendung von `amsrefs` wird dafür der Befehl `\cites` statt `\cite` empfohlen, also etwa `\cites{Beu00,Kei04}`.

21 Buchmarken und Verweise in PDF-Dokumenten

- Das Paket *hyperref* reichert die LaTeX-Ausgabedatei mit *Buchmarken* und *Verweisen* an.
- Die `pageref`-, `ref`- und `cite`-Befehle erzeugen im PDF-Dokument jeweils Verweise zum Bezugsobjekt.
- Abschnitte erzeugen Buchmarken mit der Kurzüberschrift als Name.
- Dabei sollte der Kurzname keine LaTeX-Befehle enthalten.
- Falls das nicht zu erreichen ist, hilft der Befehl

  \texorpdfstring{tex}{pdf}:

  Beispiel 10. `\section{Bewegungen \texorpdfstring{des $\mathbb{R}^2$}{der reellen Ebene}}`

  erzeugt einen Abschnitt mit Namen “Bewegungen des $\mathbb{R}^2$” und eine zugehörige Buchmarke mit Namen “Bewegungen der reellen Ebene”.

22 Index

- Füge im Text dort, wohin ein Stichworte verweisen soll, den Befehl *\index{Stichwort}* ein.
- Die Daten der *index*-Befehle werden von LaTeX in eine Datei geschrieben. Das Programm *makeindex* verarbeitet diese Daten (sortieren, zusammenfassen).
- Um tatsächlich einen Index zu erstellen, muss das LaTeX-Paket *makeidx* geladen werden, gefolgt vom Befehl *\makeindex* in der Präambel.
- Der Index wird an der Stelle erzeugt, wo der Befehl *\printindex* steht.
- *\index{Gruppe!abelsche}* liefert unter dem Indexeintrag für *Gruppe* einen Eintrag *abelsche*.
- *\index{abelsch|see{Gruppe}}* liefert einen Indexeintrag *abelsch* der auf den Eintrag *Gruppe* verweist.
- *\index{Gruppe|uu}* unterstreicht die Seitenzahl im Indexeintrag für *Gruppe*. Das sagt, daß der Begriff dort definiert wurde.

Hilfsdateien
LaTeX erzeugt eine Reihe von Hilfsdateien, darunter:

`Name.log` eine Log-Datei, die hilft, Syntax-Fehler in der Eingabedatei zu finden

`Name.aux` eine Hilfsdatei unter anderem für Textbezüge

`Name.toc` eine Hilfsdatei fürs Inhaltsverzeichnis

Frage
Warum?

- Das Inhaltsverzeichnis steht meist am Beginn eines Dokuments, bevor der Inhalt bekannt ist.
- Das Literaturverzeichnis steht am Ende, so daß seine Daten im Text noch nicht bekannt sind.
- Auch Textbezüge verweisen oft nach vorne.

Warnung
Damit Textbezüge und Inhaltsverzeichnis stimmen, muss LaTeX *mehrmals aufgerufen* werden.

23 Fußnoten

- Fußnoten werden mit dem Befehl `\footnote{Fußnotentext}` erzeugt. Dabei wird im Text eine Markierung eingefügt – meist eine hochgestellte Zahl – und am unteren Rand der Seite der Fußnotentext zusammen mit der Markierung gesetzt.
- In mathematischen Texten haben Fußnoten keine Tradition, so daß sie Leser leicht irritieren.
- In anderen Fächern werden Fußnoten regelmäßig eingesetzt, etwa für *Quellenangaben* in geschichtswissenschaftlichen Texten.

24 Bilddateien einbinden

Bilddateien einbinden

- Mit Zusatzpaketen können wir in LaTeX-Dokumenten Bilddateien geeigneter Formate einbinden.
- Welche Formate erlaubt sind, hängt davon ab, ob pdfLaTeX oder LaTeX aufgerufen wird.
- *LaTeX* kann nur ps und eps-Dateien einbinden.
- *pdfLaTeX* kann nur pdf, png, jpg, und gewisse eps-Dateien einbinden, die von MetaPost erzeugt wurden.
- Mit Programmen wie *gimp*, *pdf2ps*, *ps2pdf* können Sie verschiedene Grafikformate ineinander konvertieren.
- Wenn eine Bilddatei groß ist, sollte sie in eine *figure*-Umgebung eingepackt werden, um ihre Platzierung zu erleichtern.
- Zusätzlich ermöglicht das auch, eine Liste der Abbildungen zu erstellen.

Bilddateien einbinden mit `graphicx`

- Grafiken kann man mit dem Paket *graphicx* einbinden.
- Durch `\includegraphics{Name}` wird die Datei `Name` eingebunden, mit dem optionalen Parameter `width=4cm` kann die Breite auf vier Zentimeter festgelegt werden.
- Falls *Name* nicht existiert, wird auch `Name.pdf`, `Name.jpg`, `Name.png` oder `Name.eps` eingebunden.
- Um mit pdfLaTeX beliebige eps-Dateien mit automatischer Konvertierung einbinden zu können, kann man das Paket *epstopdf* verwenden.
  - Es muß *nach* dem Paket `graphicx` eingebunden werden.

    ```
    \usepackage[pdftex]{graphicx} \usepackage{epstopdf}
    ```
  - Dann muß man pdfLaTeX mit der Option *-shell-escape* aufrufen:

    ```
    pdflatex -shell-escape datei.tex
    ```

25 Tabellen und Abbildungen

- Große Tabellen und Abbildungen erzeugen oft Probleme mit dem Seitenumbruch.
- Die traditionelle Lösung besteht darin, diese Objekte getrennt vom eigentlichen Text dorthin zu setzen, wo gerade Platz ist.
- Dafür stellt LaTeX die zwei Umgebungen *table* und *figure* zur Verfügung, die jeweils für Tabellen und Abbildungen gedacht sind.
- Innerhalb dieser Umgebungen kann sich beliebiges Material befinden.
- Der Befehl *\centering* zentriert die Umgebung.
- Der Befehl *\caption* erzeugt eine Beschriftung
- *Nach dem* Befehl `\caption` liefert ein *label*-Befehl die Nummer der Tabelle bzw. Abbildung.

Platzierung von Fließmaterial

- Die Umgebungen `figure` und `table` vertragen jeweils ein optionales Argument, welches die Möglichkeiten zur Platzierung des Inhalts beschreibt.
- Zum Beispiel versucht `\begin{figure}[htbp]` folgende Positionen:

  ***h*ere** dort im Text, wo der Befehl auftritt

  ***t*op** oben auf einer Seite

  ***b*ottom** unten auf einer Seite

  ***p*age** auf einer speziellen Seite voller Abbildungen
- Dies ist dann relevant, wenn in kurzem Abstand mehrere Abbildungen auftreten. Diese werden von LaTeX nach und nach abgearbeitet und gemäß der erlaubten Plazierungen im Dokument abgelegt.
- Die Befehle *\listoftables* und *\listoffigures* erzeugen Verzeichnisse der Tabellen und Abbildungen analog zum Inhaltsverzeichnis.

26 Dateien einbinden

Dateien einbinden mit `\input`

- Ein langes LaTeX-Dokument kann man in kleinere Einheiten zerlegen, indem man etwa jedes Kapitel oder jeden Abschnitt in eine eigene Datei auslagert.
- Mit dem Befehl *\input* kann man die Teile dann wieder in die Hauptdatei einfügen.

Beispiel 11.
```
\documentclass{article}
\begin{document}
  \input{teil1.tex}
  \input{teil2.tex}
  \input{teil3.tex}
\end{document}
```

Dateien einbinden mit `\include`

- Man kann stattdessen auch den Befehl `\include` verwenden.
- Anders als `\input` beginnt `\include` eine neue Seite.
- `\include` darf nur im Programmkörper stehen. Im Programmkopf wird es durch den Befehl `\includeonly` komplementiert.
- Der folgende Code bewirkt, daß nur `teil2` eingelesen und kompiliert wird, daß aber für die Numerierung der Seiten, usw. `teil1` berücksichtigt wird, wenn zuvor einmal das ganze Dokument kompiliert wurde:

```
\documentclass{article}
\includeonly{teil2}
\begin{document}
  \include{teil1}
  \include{teil2}
\end{document}
```

Teil III
Der Mathematikmodus

27 Mathematikmodus

- Mathematische Formeln gibt es im *laufenden Text* – $x^2 - 3x + 1 = 0$ – oder *vom Text abgesetzt*:
$$x^2 - 3x + 1 = 0.$$
- Für beide Arten von Formeln schaltet LaTeX in einen eigenen *Mathematikmodus*.
- Viele Befehle sind nur im Mathematikmodus erlaubt und erzeugen außerhalb Fehlermeldungen.
- Trifft LaTeX außerhalb des Mathematikmodus auf solche Befehle, so schaltet es von selbst in den Mathematikmodus, was oft *Folge*fehler liefert.
- Die Zusatzpakete *amsmath*, *amsfonts*, *amssymb*, *mathtools* definieren weitere Befehle für den Mathematikmodus.

Mathematikmodus ein- und ausschalten

Wir können den Mathematikmodus auf drei verschiedene Weisen ein- und ausschalten:

	Umgebung	LaTeX-Notation	TeX-Notation
im Text	`math`	`\(...\)`	*`$...$`*
abgesetzt	*`displaymath`*	`\[...\]`	`$$...$$`

- Die drei Notationen für Mathematik im Text sind völlig äquivalent, außer daß die ersten beiden "*zerbrechlich*" sind und daher in bestimmten Situationen – insbesondere in Überschriften – zu *mysteriösen Fehlern* führen.
- Die drei Notationen für herausgehobene Formeln sind im wesentlichen äquivalent. Ich verwende stets die *displaymath*-Umgebung.

28 Brüche

Brüche, Binomialkoeffizienten, Wurzeln

- Brüche werden im Mathematikmodus mit *\frac{Zähler}{Nenner}* erzeugt.

  Beispiel 12. `\frac{\frac{1}{a}+b}{c+d}` erzeugt im Absatz $\frac{\frac{1}{a}+b}{c+d}$ und als herausgehobene Formel
$$\frac{\frac{1}{a} + b}{c + d}.$$

- Zähler und Nenner in Brüchen können beliebige mathematische Ausdrücke sein.
- *Binomialkoeffizienten* $\binom{n}{k}$ werden erzeugt durch `\binom{oben}{unten}` – das funktioniert wie `\frac`.
- *Wurzeln* wie $\sqrt{2}$ oder $\sqrt[2n]{x}$ werden mit `\sqrt` gesetzt, hier: `\sqrt{2}` und `\sqrt[2n]{x}`.

Kleinere Brüche im Text

- *Vermeiden* Sie möglichst komplizierte *Brüche im laufenden Text*, denn sie erzwingen einen höheren Zeilenabstand, was zu einem unruhigen Schriftbild führt.
- Komplizierte Brüche kommen daher besser in eine abgesetzte Formel.
- Als Ersatz für kleine Brüche taugt oft die platzsparendere Notation a/b.
- Das Paket `nicefrac` stellt mit `\nicefrac{Zähler}{Nenner}` schönere platzsparende Brüche zur Verfügung: $\nicefrac{a}{b}$, $\nicefrac{1}{2}$.

29 Indizes

Hoch- und Tiefstellung von Zeichen

Beispiel 13.
- `x^2` erzeugt x^2, `x_2` erzeugt x_2, `x^2_n` erzeugt x^2_n,
- `x^{2n}` erzeugt x^{2n}, `x_{i,i+j}` erzeugt $x_{i,i+j}$
- `x^{2^n}` erzeugt x^{2^n}
- `x^2^n` erzeugt eine Fehlermeldung
- `x^{\frac{1}{n}}` erzeugt $x^{\frac{1}{n}}$
- Höher- und tiefergestellten Formeln erscheinen in einer kleineren Schrift: $1^{2^{3^{4^5}}}$
- Ab der dritten Stufe wird die Schrift nicht mehr kleiner.

30 Operatoren und Funktionen

Große Operatoren wie Summen und Integrale

Beispiel 14.
- `\int_0^1` erzeugt $\int_0^1$ im laufenden Text und $\displaystyle\int_0^1$ in abgesetzten Formeln.
- `\sum_{n=0}^k` erzeugt $\sum_{n=0}^k$ im laufenden Text und $\displaystyle\sum_{n=0}^k$ in abgesetzten Formeln.
- $\sum$ und $\int$ sind im Text und in abgesetzten Formeln unterschiedlich groß.

- Es gibt noch Dutzende anderer *großer Operatoren*, die sich genau wie $\sum$ und $\int$ verhalten.

  Einige davon sind $\oint \prod \coprod \bigcap \bigcup \bigvee \bigwedge \bigodot \bigotimes \bigoplus \biguplus$

- Eine vollständige Liste mit Befehlsnamen finden Sie in der Datei

  `symbols-a4.pdf`.

Grenzwerte und Ähnliches

- `\lim_{n\to\infty}` erzeugt $\lim_{n\to\infty}$ im laufenden Text und $\lim\limits_{n\to\infty}$ in abgesetzten Formeln.
- Es gibt Dutzende von anderen Befehlen, die sich wie lim verhalten, darunter $\liminf$, $\limsup$, inf, sup, max, min, $\varprojlim$, $\varinjlim$.
- Um selbst einen neuen Befehl dieser Art zu definieren, verwenden wir im *Programmkopf* den Befehl `\DeclareMathOperator*{\Name}{Symbol}`
- Dies erzeugt einen Befehl `\Name`, der einen Operator namens `Symbol` einfügt.

Beispiel 15. Die Standarddefinition von lim ist äquivalent zu

```
\DeclareMathOperator*{\lim}{lim}
```

- Funktionen wie Sinus oder Logarithmus werden in Formeln *nicht kursiv* gesetzt: $\sin(x)$ statt $sin(x)$.
- Für viele bekannte Funktionen – von arccos bis sinh – gibt es schon LaTeX-Befehle wie `\arccos` und `\sinh`.
- Anders als bei lim und ähnlichen Befehle stehen bei ihnen Exponenten und Indizes niemals darunter: $\sin^2(x)$ im Text und auch $\sin^2(x)$ in abgesetzten Formeln.
- Neue Befehle dieser Art werden definiert durch

  `\DeclareMathOperator{\Name}{Symbol}`

- Dies erzeugt einen Befehl `\Name`, der eine Funktion namens `Symbol` einfügt.

  Beispiel 16. Die Standarddefinition von sin ist äquivalent zu

  `\DeclareMathOperator{\sin}{sin}`.

31 Schriften

Griechische Buchstaben

- LaTeX stellt alle griechischen Klein- und Großbuchstaben für den Mathematiksatz zur Verfügung.

  Sie werden jeweils über ihren Namen angesprochen:

- `\alpha\beta\gamma\delta` erzeugt $\alpha\beta\gamma\delta$
- `\Gamma\Delta` erzeugt $\Gamma\Delta$
- Bei einigen Kleinbuchstaben gibt es zwei Varianten:

`\epsilon-\varepsilon`	$\epsilon - \varepsilon$
`\theta-\vartheta`	$\theta - \vartheta$
`\rho-\varrho`	$\rho - \varrho$
`\pi-\varpi`	$\pi - \varpi$
`\phi-\varphi`	$\phi - \varphi$

Mathematische Schriften

- `\mathbb{C\supset R\supset Q}` erzeugt $\mathbb{C} \supset \mathbb{R} \supset \mathbb{Q}$ (das Paket `amssymb` muß geladen sein)
- `\mathfrak{g=p+k}` erzeugt $\mathfrak{g} = \mathfrak{p} + \mathfrak{k}$
- `\mathcal{ABC}` erzeugt $\mathcal{ABC}$ (keine Kleinbuchstaben)
- `\mathit{fein}` und `fein` erzeugen jeweils *fein* und $fein$
- Dies ist in einer serifenlosen Schrift wie in beamer kaum zu unterscheiden, aber in echten kursiven Schriften sind die Zwischenräume zwischen den Buchstaben (Kerning) deutlich anders.
- Die Standard-Mathematikschrift in LaTeX ist nicht zum Setzen von Worten gemacht.
- `\mathrm{fein}` erzeugt fein.
- `\mathsf{fein}` erzeugt fein.
- Auch die Befehle `\textup`, und so weiter, sind erlaubt. Allerdings gibt `\textup{A\subset B}` Fehler.

Kursiv oder nicht?

Regel
Bekannte mathematische Ausdrücke wie die Eulersche Zahl e $\approx 2{,}7$, die Funktion sin und die imaginäre Einheit $\mathrm{i} = \sqrt{-1}$ werden *nicht kursiv* gesetzt.

- Damit wird klar, daß es sich bei 2i+1 um eine komplexe Zahl handelt, während $\sum_{i=1}^{n} 2i + 1$ ein anderes i meint.
- Man sollte trotzdem Ausdrücke wie $a_{ij} = 2\pi\mathrm{i} \cdot (i + j)$ vermeiden – so deutlich ist der Unterschied zwischen i und i nicht.
- Wortbruchstücke in Formeln, etwa $\mathrm{C}^*_{\mathrm{red}}(G)$ für die reduzierte C*-Algebra werden ebenfalls *gerade* gesetzt.
- Je nach Dokumentklasse wird dies mal durch `\mathrm`, mal durch `\mathsf` erreicht. Immer funktioniert `\textup`.

32 Sonderzeichen

Sonderzeichen

- Die Anzahl der zusätzlichen Sonderzeichen im Mathematikmodus ist riesig:

$$\exists, \forall, \Longleftrightarrow, \rightarrow, \mapsto, \oplus, \prod, \bigcup, \subseteq, \ldots$$

- Für einen Überblick schlagen Sie am Besten in *symbols-a4.pdf* nach.
- Man kann viele Operatoren durch das Voranstellen des Befehls `\not` *negieren*, z.B.

\not=	$\neq$
\not\subseteq	$\nsubseteq$
\not\Rightarrow	$\not\Rightarrow$

33 Mathematische Akzente

- Die üblichen Akzentbefehle für normalen Text sind im Mathematikmodus *ungültig* und erzeugen Fehlermeldungen.
- Stattdessen hat der Mathematikmodus *eigene* Akzente:

`\tilde{a} \hat{a} \vec{a} \dot{a} \mathring{a} \bar{a}` erzeugt $\tilde{a}\hat{a}\vec{a}\dot{a}\mathring{a}\bar{a}$

- Auch hier ist das Angebot noch größer, siehe *symbols-a4.pdf*.
- Das Apostroph `'` ist im Mathematikmodus meistens synonym zu `^\prime` und erzeugt ein *Ableitungssymbol* wie in f'.
- `f''` liefert f'' wie erwartet, nicht aber `f^\prime^\prime`.

34 Ausdehnbare Pfeile und ähnliches

- `\xrightarrow[unten]{oben}` erzeugt $\xrightarrow[unten]{oben}$.
- Die Länge dieses Pfeils paßt sich der Länge der Dekorationen an.
- `\underbracket{Formel}_{Index}` erzeugt $\underbracket{Formel}_{Index}$.
- `\underbrace{Formel}_{Index}` erzeugt $\underbrace{Formel}_{Index}$.
- *amsmath* und *mathtools* stellen noch mehr derartige Pfeile und Klammern zur Verfügung.
- `\overset{!}{=}` erzeugt $\overset{!}{=}$

35 Große Klammern und Relationen

Große Klammern

- Vergleiche

$$(\frac{a^2}{b^2}) \quad (\frac{a^2}{b^2}) \quad \left(\frac{a^2}{b^2}\right) \quad \left(\frac{a^2}{b^2}\right) \quad \left(\frac{a^2}{b^2}\right) \quad \left(\frac{a^2}{b^2}\right)$$

- die größeren Klammern werden jeweils erzeugt durch `\bigl`, `\Bigl`, `\biggl`, `\Biggl` für linke und `\bigr`, `\Bigr`, `\biggr`, `\Biggr` für rechte Klammern.
- das letzte Klammerpaar wird mit `\left(...\right)` erzeugt. Dies liefert Klammern, deren Größe an die dazwischen liegende Formel angepaßt ist und die beliebig groß werden können.

  Beispiel 17. $f \circ g(x) = f\big(g(x)\big)$ wird erzeugt durch

```
f\circ g(x)=f\bigl(g(x)\bigr).
```

  Hier hilft `left–right` nicht, weil auch die kleinsten Klammern groß genug sind.

Große Relationen

Der Befehl `\bigm`, `\Bigm` liefert große Relationen:

$$\frac{a}{b} \,\Big|\, \frac{c}{d} \qquad (a/b \text{ teilt } c/d).$$

Frage

Was ist der Unterschied zwischen bigl–bigr–bigm?

- LaTeX unterscheidet beim Formelsatz zwischen verschiedenen Arten von Symbolen, etwa öffnende und schließende Klammern und Relationen.
- Die Befehle `\left` und `\right` müssen immer als *Paar* auftreten, sonst produziert LaTeXeine Fehlermeldung.
- Will man nur eine öffnende geschweifte Klammer mit beliebiger Größe, so sollte dem `\left\{` ein `\right.` folgen. Damit treten `\left` und `\right` als Paar auf, der Punkt erzeugt aber keine Klammer.

36 Arrays

Matrizen

Der flexibelste und leistungsfähigste Befehl für Matrizen ist die *`array`-Umgebung*. Ihre Syntax ist die gleiche wie die der *`tabular`-Umgebung* für Tabellen – außer, daß alle Einträge im Mathematikmodus bearbeitet werden.

Beispiel 18.
```
\left(
  \begin{array}{cc}
    0&1\\
    2&3
  \end{array}
\right)
```
erzeugt die Matrix

$$\left(\begin{array}{cc} 0 & 1 \\ 2 & 3 \end{array} \right).$$

Das Argument `cc` zur `array`-Umgebung bedeutet, daß die Matrix zwei zentrierte Spalten hat.

Einfachere Matrizen

- *amsmath* definiert mehrere Abkürzungen für spezielle einfache Matrizen.
- *mathtools* liefert noch mehr solcher Abkürzungen.

Beispiel 19. `\begin{pmatrix}0&-1\\2&3\end{pmatrix}`

$$\begin{pmatrix} 0 & -1 \\ 2 & 3 \end{pmatrix}$$

Beispiel 20. `\begin{pmatrix*}[r]0&-111\\2&3\end{pmatrix*}`

$$\begin{pmatrix*}[r] 0 & -111 \\ 2 & 3 \end{pmatrix*}$$

Das Multiplikationszeichen und Punkte

- `\cdot` erzeugt das Multiplikationszeichen "$\cdot$".
- `\cdots` erzeugt drei zentrierte Punkte "$\cdots$".
- `\ldots` erzeugt die Ellipse "$\ldots$".
- `\vdots` erzeugt die vertikale Punkte "$\vdots$".
- `\ddots` erzeugt die diagonale Punkte "$\ddots$".

Beispiel 21.

$$\begin{aligned} r_0 &= r_1 q_1 + r_2, & \nu(r_2) &< \nu(r_1), \\ r_1 &= r_2 q_2 + r_3, & \nu(r_3) &< \nu(r_2), \\ &\vdots \\ r_{k-2} &= r_{k-1} q_{k-1} + r_k, & \nu(r_k) &< \nu(r_{k-1}), \end{aligned}$$

Ein Beispiel mit der array-Umgebung

Beispiel 22.
```
\setlength{\arraycolsep}{0.5mm}
\begin{array}{rcrcrcrcrcrcrcccc}
  (t^2&&&-&1&)&:(t-1)&=&t+1\\
   t^2&-&t\\\cline{1-3}
  && t&-&1\\
  && t&-&1\\\cline{3-5}
  &&&&0
\end{array}
```

$$\begin{array}{rcrcrcrcrcrcrcccc}
  (t^2&&&-&1&)&:(t-1)&=&t+1\\
   t^2&-&t\\\cline{1-3}
  && t&-&1\\
  && t&-&1\\\cline{3-5}
  &&&&0
\end{array}$$

Fallunterscheidungen und Text in Formeln

- Für Fallunterscheidungen gibt es die *cases*-Umgebung.
  Beispiel 23. `\begin{cases} 0, & x<0\\x, & x\geq 0 \end{cases}`
  Dies erzeugt $\begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$
- Meist ist es sinnvoll, in der zweiten Spalte der cases-Umgebung Text einzufügen.
- Dafür gibt es viele äquivalente Befehle: *\hbox{Text}*, *\mbox{Text}* und *\text{Text}*.
- Der Vorteil von *\text* ist, daß dieser Befehl auch in Indizes und Exponenten die richtige Schriftgröße wählt: `x_{\text{minimal}}` erzeugt x_{minimal}, `x_{\mbox{minimal}}` erzeugt $x_{\mbox{minimal}}$.

37 Leerraum im Mathematikmodus

- Im Mathematiksatz hängt der Leerraum zwischen zwei Zeichen von ihrer *syntaktischen Funktion* ab. Vergleiche $a = b\ aRb\ a + b\ a(b\ a)b\ a, b$
- Entsprechend unterscheidet TEX im Mathematiksatz sieben Arten von Objekten:

  \mathord gewöhnliche Zeichen wie $12ab\alpha$
  \mathopen öffnende Klammern wie $([\{\langle$
  \mathclose schließende Klammern $)]\}\rangle$
  \mathbin binäre Operatoren wie $+- : \oplus$
  \mathrel Relationen wie $=<>\leq\geq\approx$
  \mathpunct Satzzeichen wie , ; :
  \mathop große Operatoren wie $\int \sum \sin \bigoplus$
- In jeder Zeile steht, mit welchem Befehl wir die Funktion eines Ausdrucks manuell setzen können.

Leerraum richtig erzeugen

Beispiel 24. `[0,1\mathclose[\cup\mathopen]0,1]` gibt $[0,1[\cup]0,1]$
`[0,1[\cup ]0,1]` erzeugt $[0,1[\cup]0,1]$

Beispiel 25. Eine Relation R auf einer Menge X heißt *reflexiv*, wenn $a\,R\,a$ für alle $a \in X$ gilt.

Benutze `a\mathrel{R}a`.

Beispiel 26. Der Doppelpunkt `:` ist für TEX ein binärer Operator (Division). Für das entsprechende Satzzeichen gibt es den Befehl `\colon`. Vergleiche $f\colon X \to Y$ mit $f : X \to Y$.

Leerraum richtig erzeugen II

Beispiel 27. `/` ist für LATEX ein gewöhnliches Zeichen. Vergleiche $(x+y)/(1+x^2+y^2)$ und $(x+y)\,/\,(1+x^2+y^2)$.

- *Gruppen* werden von LATEX standardmäßig wie gewöhnliche Zeichen behandelt. Vergleiche: `a{=}b` erzeugt $a{=}b$, `3{,}1415` erzeugt 3,1415 `a=b` erzeugt $a = b$, `3,1415` erzeugt $3,1415$
- Im zweiten Beispiel ist dies wünschenswert.

  Übrigens ist `\mathord{,}` synonym zu `{,}`.
- Auch Gruppen können wir eine andere syntaktische Funktion zuordnen: `a\mathrel{\dot R}b` erzeugt $a\,\dot R\,b$, `V\mathbin{\hat\otimes_\pi}W` erzeugt $V\,\hat\otimes_\pi\,W$.

Integrale, Leerraum von Hand einfügen

- Die Gleichung

$$f\colon \mathbb{R} \to \mathbb{R}, \qquad x \mapsto \int_0^x t^2 - 3t + 1 \,\mathrm{d}t$$

  wird gesetzt durch

```
f\colon \mathbb{R}\to\mathbb{R}, \qquad x\mapsto \int_0^x
                t^2-3t+1 \,\textup{d}t
```

- `\colon` liefert einen Doppelpunkt als Satzzeichen.
- Durch `\qquad` wird ein größerer Leerraum eingefügt, durch `\,` ein kleiner.
- Durch `\textup{d}` wird ein nicht-kursives d gesetzt, wie es in Integralen gebräuchlich ist.

38 Indizes an Operatoren

- Bei großen Operatoren wie $\sum$ oder lim können wir die Plazierung von Indizes durch die Befehle `\nolimits` und `\limits` steuern.

- Der erste sorgt dafür, daß auch in abgesetzten Formeln Indizes und Exponenten wie im Text gesetzt werden, der zweite sorgt dafür, daß auch im Text Indizes und Exponenten wie in abgesetzten Formeln gesetzt werden.

Beispiel 28. Der abgeleitete projektive Limes-Funktor $\varprojlim^1$ wird durch `\varprojlim\nolimits^1` erzeugt. Ohne `\nolimits` erhielten wir in abgesetzten Formeln $\varprojlim^1$.

Ein komplexes Beispiel

- In der analytischen Zahlentheorie ist folgende Notation gebräuchlich:

$$\sum_{a,b\in\mathbb{Z}}{}' \frac{1}{a^s+b^s}$$

  (Der Strich zeigt an, daß in der Summe $(a,b)=(0,0)$ auszulassen ist.)

- Dies wird dadurch kompliziert, daß wir Indizes in zwei inkompatiblen Stellungen mischen. Eingegeben wird diese Summe durch

```
\mathop{\sum\nolimits'}_{a,b\in\mathbb{Z}}
```

Ein komplexes Beispiel II

- Benutzen wir diese gestrichene Summe öfter, so deklarieren wir im Programmkopf einen entsprechenden Operator mit

```
\DeclareMathOperator*{\sumprime}
                     {\sum\nolimits'}
```

- Dann erzeugt `\sumprime_{a,b\in\mathbb{Z}}` die obige Summe.

- Übrigens folgende Version funktioniert nicht:

```
\DeclareMathOperator*{\sumprime}{\sum\nolimits'}
```

39 Gleichungen

Numerierte Gleichungen

- Die *equation*-Umgebung erzeugt eine numerierte Gleichung

```
\begin{equation}
  \label{eq:idem}
  x^2=x
\end{equation}
```

  liefert die Gleichung

$$x^2 = x \tag{1}$$

- `\ref{eq:idem}` liefert die Nummer 1.
- `\eqref{eq:idem}` liefert die Nummer (1) in Klammern.
- Die Platzierung der Nummer läßt sich durch Optionen des `documentclass`-Befehls steuern: die Option *leqno* erzeugt Gleichungsnummern links.

Bündige und nicht bündige Gleichungen

- Die *gather*-Umgebung erzeugt mehrere numerierte Gleichungen, die jeweils für sich zentriert werden.
- Die *align*-Umgebung erzeugt mehrere numerierte Gleichungen, die an der Position des Tabulatorzeichens `&` ausgerichtet werden:

  ```
  \begin{align}x^2&=x\\y^2&=y+1\end{align}
  ```

  erzeugt

$$x^2 = x \tag{2}$$
$$y^2 = y + 1 \tag{3}$$

- Die *multline*-Umgebung erzeugt eine lange Gleichung, die sich über mehrere Zeilen erstreckt:

$$\begin{aligned} x = x = x = x = x = x = x = x \\ = x = x = x = x = x = x = x \\ = x = x = x = x = x = x = x \end{aligned} \tag{4}$$

Mehrere bündige Gleichungen

- Die *alignat*-Umgebung erzeugt mehrere bündige Blöcke pro Zeile:

  ```
  \begin{alignat}{2}
    f&\colon X\to Y, &\quad x&\mapsto f(x)\\
    g&\colon Y\to Z, &\quad y&\mapsto g(y).
  \end{alignat}
  ```

  erzeugt

$$f\colon X \to Y, \quad x \mapsto f(x), \tag{5}$$
$$g\colon Y \to Z, \quad y \mapsto g(y). \tag{6}$$

Weitere Varianten

- Bei den verwandten Umgebungen *gather**, *align**, *multline**, *alignat** fällt die Numerierung weg.
- Die Umgebungen *gathered*, *aligned*, *multlined*, *alignedat* können innerhalb einer Gleichung benutzt werden und beliebig kombiniert werden.

  Beispiel 29. Die Kombination von *equation* und *aligned* erzeugt eine Reihe von bündigen Gleichungen mit einer einzigen Gleichungsnummer:

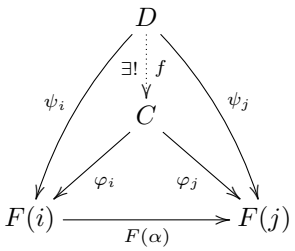
$$\begin{aligned} x^2 &= x \\ y^2 &= y + 1 \end{aligned} \tag{7}$$

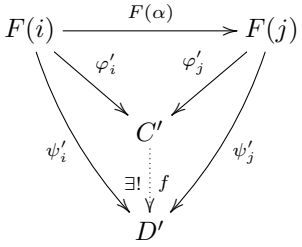
40 Das Paket xy

Kommutative Diagramme mit dem Paket xy

- Das Macropaket ***xy*** ist das leistungsfähigste Paket zum Setzen von kommutativen Diagrammen.

Example 30.

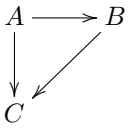




- Der Aufruf dieses Pakets erfolgt in der Regel mit `\usepackage[all]{xy}`.
- Da ***xy*** noch für TEX geschrieben wurde, ist seine Syntax etwas anders als in LATEX üblich.

Ein einfaches Beispiel

```
\xymatrix{ A \ar[d] \ar[r] & B \ar[dl] \\ C }
```

erzeugt 

- Der Befehl ***xymatrix*** erzeugt das Diagramm.
- Dieses wird formatiert wie eine Matrix, wobei es zusätzlich zu den Einträgen noch Pfeile gibt, die mit dem Befehl `\ar` eingegeben werden können.
- Im einfachsten Fall hat `\ar` ein Argument in eckigen Klammern, das angibt, wohin der Pfeil zeigt.
  - `\ar[llu]` erzeugt einen Pfeil, der zwei Schritte nach links (left) und einen nach oben (up) zeigt.
  - `\ar[rrd]` erzeugt einen Pfeil, der zwei Schritte nach rechts (right) und einen nach unten (down) zeigt.

Pfeile beschriften

- Durch die Ergänzungen `^{...}` und `_{...}` wird der Pfeil oben oder unten mit dem Text ... dekoriert: `\xymatrix{A\ar[r]^{f}_{\cong}&B}`

  $A \xrightarrow[\cong]{f} B$

- Oben heißt hier: im Gegenuhrzeigersinn von der Pfeilrichtung:

  `\xymatrix{A&B\ar[l]^{f}_{\cong}}` $A \xleftarrow[f]{\cong} B$.

- Standardmäßig erscheint die Beschriftung in der *Mitte* zwischen den benachbarten *Einträgen*. Sind diese verschieden groß, so ist dies verschieden von der *Mitte* des *Pfeils*.
- Durch `^-{...}` wird die Beschriftung in die Mitte des Pfeils gesetzt. Vergleiche LangerEintrag $\overset{f}{\longrightarrow}$ kurz und LangerEintrag $\xrightarrow{f}$ kurz

Varianten

- Der *Abstand* zwischen Spalten (columns) und Zeilen (rows) der xymatrix kann angepaßt werden. Durch `\xymatrix@C+2em@R-1em` wird der Spaltenabstand um 2em erhöht und der Zeilenabstand um 1em erniedrigt, jeweils gegenüber dem Standardwert.
- Die *Form der Pfeile* wird wie folgt geändert: `\ar@{.>>}[r]` erzeugt einen punktierten Pfeil mit Doppelspitze: $A \cdots\!\!\gg B$. Für verschiedene andere Varianten vergleiche die Dokumentation des Pakets `xy`.
- Pfeile können nach unten oder oben *gebogen* werden durch `\ar@/^/[r]` und `\ar@/_/[r]`.

  Example 31. `\xymatrix{A\ar@/^/@{.>>}[r]&B\ar@/^/@{<->}[l]}` [0.2cm] erzeugt $A \overset{\longrightarrow}{\underset{\longleftarrow}{\cdots\!\!\gg}} B$
- Man kann festlegen, wie stark Pfeile gebogen sein sollen.

Teil IV
Präsentationen mit `beamer`

41 Vorbemerkungen

- Die Dokumentation *`beameruserguide.pdf`* zur Klasse `beamer` enthält neben einer *Anleitung* auch *Beispiele* und *kommentierte Vorlagen* für verschiedene Arten von Vorträgen.
- Letztere sind am Anfang sehr sinnvoll, weil man weitgehend ohne Kenntnis von *beamer* durch Anpassen und Einfügen sinnvoller Daten das Grundgerüst einer Präsentation erhält.
- Sie enthalten auch einige sinnvolle Mahnungen zur Gestaltung von Präsentationen.

Warum eine spezielle Klasse?
Eine Präsentation sollte anders aussehen als ein Buch:

- Damit sie überhaupt lesbar ist, sollten wir eine angemessene Schriftart und -größe wählen.
- Außerdem sollten die einzelnen Seiten *nicht zu viel* Information enthalten, damit die Zuhörer folgen können.
- Textbezüge, Literaturverweise, *numerierte* Gleichungen und Sätze sind in Präsentationen meist fehl am Platz.
- Hervorhebungen geschehen am besten *farbig*, nicht durch Wechsel der Schriftart.
- *Im Unterschied zu* Folien *bietet die Präsentation die Möglichkeit, Seiten schrittweise aufzubauen oder zu verändern.*
- Hier liegt eine der Stärken der `beamer`-Klasse.

42 Seiten schrittweise aufbauen

- Eine Präsentation gliedert sich in *Rahmen* und *Seiten*.
- Die *`frame`*-Umgebung erzeugt jeweils einen *Rahmen*.
- Ein Rahmen wird meist aus *mehreren* Seiten schrittweise aufgebaut, jedenfalls sind die verschiedenen Seiten eines Rahmens miteinander verwandt.
- Der Inhalt einer `frame`-Umgebung ist normaler LaTeX-Code mit zusätzlichen Markierungen und Befehlen, die etwas nur auf *bestimmten Seiten* des Rahmens erscheinen läßt.
- Daran sieht LaTeX auch, wie viele Seiten der aktuelle Rahmen haben sollte.

Standard-Overlay

Einfachste, aber noch unflexible Methode
Schreibe in den Programmkopf der Eingabedatei den Befehl

`\beamerdefaultoverlayspecification{<+->}`.

Meist liefert das gute aber nicht optimale Resultate.
Dann muss von Hand nachjustiert werden.

***Lokale* Variante**
Gebe einer Umgebung das optionale Argument `[<+->]`, etwa

`\begin{frame}[<+->]`

Dann wird die erste Methode innerhalb dieser Umgebung benutzt.

Schrittweises Aufbauen der Seite mit `pause`

- Ändert man den Standardoverlay wie beschrieben, so ist *jeder* Punkt einer Liste oder Aufzählung und *jeder* Satz ein eigener Schritt beim Aufbau des Rahmens.
- Möchte man einige dieser Schritte aussparen oder weitere einfügen, so muss man von Hand *markieren*, wo der Seitenaufbau jeweils anhalten soll.
- Dies leistet der Befehl `\pause`.
- Benutzen Sie eine default overlay specification, können Sie diese im aktuellen Rahmen mit `\begin{frame}[<*>]` ausschalten.

42.1 Overlay-Angaben

Seitenaufbau mit detaillierten Overlay-Angaben

- Mit `\pause` werden Seiten immer *linear* aufgebaut. `\item<1-3,5->`
- Wir können bestimmten LaTeX-Befehlen und -Umgebungen, darunter `\item` und die verschiedenen `newtheorem`-artigen Umgebungen, jeweils mit einer *Overlay-Angabe* versehen. `\item<3->`
- Den ersten Punkt dieser Aufzählung haben wir durch `\item<1-3,5->` eingegeben. Dadurch wird er nur auf Seiten 1–3 und ab Seite 5 angezeigt, auf der aktuellen vierten Seite verschwindet er. `\item<4->`
- Diesen Punkt der Aufzählung haben wir durch `\item<5->` eingegeben. `\item<5->`
- Manchmal benötigt man einen nicht-linearen Seitenaufbau. `\item<2->`

Verschiedene Overlayangaben

- Die Angabe von Overlays durch <1->, <2->, ..., hat den Nachteil, daß sich alles ändert, wenn wir die Reihenfolge ändern oder etwas einfügen.
- Statt Ziffern können wir auch `+` und `.` benutzen. Sie beziehen sich auf den gleichen Zähler – `beamerpauses` – der auch vom `pause`-Befehl benutzt wird.
- Durch `+` wird der Wert dieses Zählers eingefügt und der Zähler um 1 erhöht. Durch `.` wird der Wert dieses Zählers *minus 1* eingefügt und der Zähler wird nicht erhöht.
- Dies erklärt auch die Funktion der *default overlay specification* <+->.

Beispiele 32.
```
\begin{itemize}
  \item<+-> Apfel
  \item<+-> Birne
  \item<+-> Pflaume
  \item<+-> Orange
\end{itemize}
```

```
\begin{itemize}
  \item<1-> Apfel
  \item<2-> Birne
  \item<3-> Pflaume
  \item<4-> Orange
\end{itemize}
```

```
\begin{itemize}
  \item<+-> Apfel
  \item<.-> Birne
  \item<+-> Pflaume
  \item<.-> Orange
\end{itemize}
```

```
\begin{itemize}
  \item<1-> Apfel
  \item<1-> Birne
  \item<2-> Pflaume
  \item<2-> Orange
\end{itemize}
```

sind äquivalent.

Mehr overlay-Angaben

- Tritt `+` mehrmals in *einer* overlay-Angabe auf, so wird der Zähler trotzdem *nur einmal* erhöht.
- Das Symbol `+` in Overlays kann noch durch einen *offset* verschoben werden.

  Beispiel 33. Hat `beamerpauses` den Wert 3, so ist <+(-1)-+(2)> äquivalent zu <2-5>.

42.2 only und uncover

[t]

Der Befehl only

- Durch *\only<...>{Text}* wird Text nur auf den angegebenen Seiten gezeigt und belegt auf anderen auch keinen Platz.
- Wollen Sie zum Beispiel eine externe Grafik schrittweise aufbauen, so haben Sie vielleicht zwei Grafikdateien `Grafik1.pdf` und `Grafik2.pdf`.
- Durch

  ```
  \only<-3>{\includegraphics{Grafik1.pdf}}
  \only<4->{\includegraphics{Grafik2.pdf}}
  ```

  erscheint bis Seite 3 die erste und ab Seite 4 die zweite Grafik.
- Sind die beiden Grafiken nicht exakt gleich groß, schalten Sie durch die Option *[t]* an der *frame*-Umgebung die automatische Zentrierung der Seite ab: *\begin{frame}[t]*

Der Befehl uncover

- Durch *\uncover<...>{Text}* wird Text nur auf den angegebenen Seiten gezeigt, belegt aber auf anderen Seiten weiterhin Platz.

43 Gestaltung der Seiten

43.1 Hervorhebung

- Der Befehl zum (farbigen) *Hervorheben* in `beamer`-Dokumenten ist *\alert* statt `\emph`.
- Der Befehl `\alert` kann mit einer Overlayangabe versehen werden, um Text *nur auf einigen Seiten* hervorzuheben.
- Gerade haben wir *\alert<2>{nur ...}* benutzt.
- Wir können auch die Overlay-Angaben von Befehlen durch `alert`-Angaben anreichern.
- Den letzten Punkt haben wir mit *\item<4-| alert@4>* eingegeben. Dadurch wird er ab Seite 4 angezeigt und nur auf Seite 4 hervorgehoben.
- In *älteren* Versionen von `beamer` *muss* hinter | ein *Leerzeichen* folgen.

43.2 Umgebungen

Die `itemize`-Umgebung

- Die Umgebung *`itemize`* wird in Präsentationen öfter eingesetzt als in normalen Texten und funktioniert wie gewohnt.
- Die Umgebung hat eine default overlay specification als optionales Argument.

  Beispiel 34. *<.->*

```
\begin{itemize}[<+->]
  \item Ab der ersten Seite
  \item Ab der zweiten Seite
  \item<1-> Ab der ersten Seite
  \item Ab der dritten Seite
\end{itemize}
```

Hervorhebung in `itemize`-Umgebungen

```
\begin{itemize}[<+-| alert@+>]
  \item Dies erscheint ab der ersten Seite und
    ist nur auf der ersten Seite hervorgehoben.
   \item Dies erscheint ab der zweiten Seite und
    ist nur auf der zweiten Seite hervorgehoben.
\end{itemize}
```

```
\begin{itemize}[<+->]
  \item Dies erscheint ab der \alert<.>{ersten}
   Seite, und nur dort gilt die Hervorhebung.
  \item Dies erscheint ab der \alert<.>{zweiten}
   Seite, und nur dort gilt die Hervorhebung.
\end{itemize}
```

Aufzählungen und freie Listen in `beamer`

- Aufzählungen werden wie gewohnt mit der *`enumerate`*-Umgebung gesetzt. Sie verträgt zwei getrennte optionale Argumente:
  1. Eine default overlay specification
  2. Ein Format für den Zähler wie im Paket `enumerate`.
- Auch die Umgebung *`description`* funktioniert wie gewohnt. Sie verträgt zwei optionale Argumente:
  1. Eine default overlay specification
  2. Die *längste Markierung* – wird für korrekte Einrückung der Liste benutzt und kann bei kurzen Markierungen entfallen.

Beispiele `description`

```
\begin{description}
  \item[Löwe] König der Savanne
  \item[Tiger] König des Dschungels
\end{description}
```

Löwe König der Savanne

Tiger König des Dschungels

```
\begin{description}[lange Marke]
  \item[kurz] Text
  \item[lange Marke] Text
\end{description}
```

kurz Text

lange Marke Text

Blöcke

Mit verschiedenen *`block`*-Umgebungen werden Textblöcke mit einer Überschrift gesetzt:

```
\begin{block}{Titel}
   Text
\end{block}
```

Titel
Text

```
\begin{alertblock}{Titel}
   Text
\end{alertblock}
```

Titel
Text

```
\begin{exampleblock}{Titel}
   Text
\end{exampleblock}
```

Titel
Text

Daneben sind auch verschiedene Umgebungen für Sätze, Definitionen und Beispiele vordefiniert – sowohl mit deutschen als auch mit englischen Namen.

verbatim in beamer

Will man in einem Rahmen die *verbatim*-Umgebung oder den Befehl *\verb* verwenden, dann muß man der *frame*-Umgebung das optionale Argument *fragile* übergeben.

Übergebt die beiden Umgebungen *niemals* als Teile eines Parameters an LaTeX-Befehle (wie z. B. \alert).

Beispiel 35.
```
\begin{frame}[fragile]
   \begin{verbatim}
      Text
   \end{verbatim}
   So setzt man Text in \verb+verbatim+!
\end{frame}
```

verbatim und \alert

- In dem Beispiel auf der vorherigen Seite haben wir in der `verbatim`-Umgebung scheinbar den Befehl `\alert` verwendet. Das geht nicht!
- Stattdessen haben wir die Umgebung *semiverbatim* benutzt, die nur in der Klasse *beamer* existiert.
- Die Option *fragile* muß wieder gesetzt sein.
- In ihr behalten die Zeichen \, *{* und *}* ihren Befehlscharakter, so daß man einfache Befehle wie `\alert` verwenden kann.
- Der Preis dafür ist, daß man die Zeichen \, *{* und *}* durch voranstellen eines \ maskieren muß, wenn man sie als Text in der Umgebung haben will.

Beispiel 36.
```
\\begin\{semiverbatim\}
   Dieser \\alert\{Text\} wird hervorgehoben!
\\end\{semiverbatim\}
```

43.3 Mehrere Spalten

- Die *columns*-Umgebung erlaubt es, Teile eines Rahmens mehrspaltig zu setzen.
- Innerhalb der `columns`-Umgebung erzeugt

  \begin{column}{5cm}...\end{column}

  eine Spalte der Breite 5 cm.

Beispiel 37.
```
\begin{columns}
  \begin{column}{5cm}
    Erste Spalte
  \end{column}
  \begin{column}{5cm}
    Zweite Spalte
  \end{column}
\end{columns}
```

Rahmenbestandteile

- Jeder Rahmen kann neben dem eigentlichen Inhalt folgende Bestandteile haben:
  - *Kopf- und Fußzeile*
  - *Linker und rechter Seitenstreifen*
  - *Hintergrund*
  - *Navigationsleisten*
  - *Navigationssymbole*
  - *Titel und Untertitel*
  - *Logo*
- Die ersten fünf werden in der Regel automatisch erzeugt und können durch Wahl von *themes* angepaßt werden.
- Titel und Untertitel werden mit `\frametitle` und `\framesubtitle` festgelegt.
- Das Logo wird mit `\logo` definiert, meist ist dies ein Kommando zum Einfügen einer Grafik.

44 Globale Struktur der Präsentation

44.1 Titelseite

- Die *Titelseite* wird erzeugt durch

```
\begin{frame}<presentation>
  \titlepage
\end{frame}
```

- Sie benutzt die folgenden Daten

  `\title` Titel

  `\subtitle` Untertitel

  `\author` Autor

  `\date` Datum

  `\institute` Institut

  `\titlegraphic` Titelgrafik

Kurz- und Langformen

- Titel, Untertitel, Autor, Datum und Institut können als optionales Argument eine *Kurzform* des Eintrags bekommen. Diese wird in Kopf- und Fußzeilen benutzt.
- Mehrere Autoren werden durch `\and` getrennt, und durch den Befehl `\inst` wird gegebenenfalls erklärt, welcher Autor zu welchem Institut gehört:

\author[Keilen and Tyomkin] {Thomas Keilen\inst{1} *and* Ilya Tyomkin\inst{2}}
\institute{\inst{1}TU Kaiserslautern *and* \inst{2}University of Tel Aviv}

44.2 Abschnitte und Inhaltsverzeichnis

- Abschnitte und Unterabschnitte werden wie gewohnt durch `\section`, `\subsection`, `\subsubsection` erzeugt.
- Sie erscheinen in Navigationsleisten und im Inhaltsverzeichnis, das mit dem üblichen Befehl `\tableofcontents` erzeugt wird.
- Die **-Varianten* davon erscheinen in der Navigationsleiste, aber nicht im Inhaltsverzeichnis.
- Der Befehl `\tableofcontents` kann verschiedene Optionen erhalten, unter anderem:

  currentsection hebt aktuellen Abschnitt hervor

  currentsubsection hebt aktuellen Unterabschnitt hervor

  pausesection fügt für jeden Abschnitt einen `\pause`-Befehl ein

  pausesubsection fügt für jeden Unterabschnitt einen `\pause`-Befehl ein

Vorträge mit mehreren Teilen

- Durch den Befehl `\part` wird die Präsentation in Teile gegliedert, die *nichts* miteinander zu tun haben.
- Dies ist *nur für sehr lange* Vorträge sinnvoll.
- Abschnitte eines anderen Teils erscheinen nicht in der Navigationsleiste, und das Inhaltsverzeichnis kann auf einen Teil beschränkt werden durch das optionale Argument `[part=...]` für `\tableofcontents`.
- Durch `\partpage` wird eine Titelseite für den gerade aktuellen Teil des Vortrags eingefügt.

44.3 Literaturverzeichnis und Anhang

Literaturverzeichnis

- Die `thebibliography`-Umgebung fügt ein Literaturverzeichnis ein.
- Die Einträge werden mit `\bibitem[Name]{Marke} ...` eingefügt, wobei auf diesen Eintrag durch `\cite{Marke}` verwiesen wird und dann `[Name]` erscheint
- Zwischen Autor, Titel, Journal, und einer eventuellen Notiz zum Eintrag sollte jeweils der Befehl `\newblock` benutzt werden, damit `beamer` die Einträge gut formatieren kann.

- Im Literaturverzeichnis erscheint statt des Namens jeweils ein *Bild* als Marke.
- Die Befehle

  `\beamertemplate`*`book`*`bibitems` und
  `\beamertemplate`*`article`*`bibitems`

  wählen aus, ob ein Buch- bzw. Artikelsymbol benutzt wird.

Beispiel eines Literaturverzeichnisses

Literatur

[LaTeX-Introduction] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. The Not So Short Introduction to LaTeX 2_ε. Part of most LaTeX installations.

[xydoc] Kristoffer H. Rose. XY-pic User's Guide. `.../generic/xypic/xyguide.pdf`

[Beamerdoc] Till Tantau. User's Guide to the Beamer Class. `.../latex/beamer/doc/beameruserguide.pdf`

Quellcode zum Literaturverzeichnis

```
\begin{thebibliography}{99}
  \beamertemplatebookbibitems
   \bibitem[\LaTeX-Introduction]{lshort}
     Tobias Oetiker, Hubert Partl, ....
     \newblock The Not So Short Introduction ....
     \newblock Part of most \LaTeX{} ....


   \beamertemplatearticlebibitems
   \bibitem[xydoc]{xy}
     Kristoffer H. Rose.
     \newblock XY-pic User's Guide.
     \newblock \texttt{\dots/generic/...}
\end{thebibliography}
```

Anhang

- Im Anhang stehen Rahmen, die im eigentlichen Vortrag eigentlich nicht gezeigt werden sollen, die aber vielleicht nützlich sein könnten, um Fragen zu beantworten.
- Der Anhang beginnt (wie üblich) mit `\appendix`.
- Rahmen und Abschnitte im Anhang erscheinen nicht in den Navigationsleisen oder im Inhaltsverzeichnis.

44.4 Interne Links

Interne Links

- Man kann einzelne Rahmen überspringen – etwa einen Beweis, für den am Schluß doch keine Zeit bleibt.
- Man kann sich die Möglichkeit offen halten, an bestimmte Stellen im Anhang zu springen und wieder zurück.
- Dafür brauchen wir *Hyperlinks* und *-targets*.
- Am einfachsten erzeugen wir die Zielmarkierungen für Sprünge durch die `label`-Option der `frame`-Umgebung.
- Durch `\begin{frame}[label=Name]` werden für jede Seite des Rahmens eine Marke mit Namen `Name<1>`, `Name<2>`, und so weiter angelegt.
- Durch `\hyperlink{Marke}{Text}` wird Text eingefügt und dafür gesorgt, daß das Anklicken von Text an die Marke `Marke` springt. Hier folgt ein Sprung zur ersten Seite dieses Rahmens: `\hyperlink{Hyper<1>}`

Schönere Sprungknöpfe

- Damit der Nutzer leicht erkennen kann, wo er klicken kann und was das bewirkt, benutzen sie folgende Knöpfe zum Springen:

`\beamerbutton` Text

`\beamergotobutton` Zum Beweis

`\beamerskipbutton` Beweis überspringen

`\beamerreturnbutton` Zurück

- Diese Befehle haben jeweils ein Argument: den Text, und vertragen die üblichen Overlay-Angaben.
- Die obigen Knöpfe sind blind, weil die *\hyperlink*-Befehle fehlen.
- Richtig ist die Kombination

  `\hyperlink{Hyper<3>}{\beamerreturnbutton{Text}}`

Weitere Sprungbefehle

- Es gibt spezielle Makros für oft benötigte Sprünge:

  `\hyperlinkslideprev` eine Seite zurück

  `\hyperlinkslidenext` eine Seite vor

  `\hyperlinkframestart` Anfang des Rahmens

  `\hyperlinkframeend` Ende des Rahmens

  `\hyperlinkframestartnext` Anfang des nächsten Rahmens

  `\hyperlinkframeendprev` Ende des letzten Rahmens

45 Themen der beamer-Klasse

- *Themen* und *Muster* steuern das Aussehen von Präsentationen.
- Verschiedene Arten von Themen regeln verschiedene Aspekte der Gestaltung:

  Umfassende Themen regeln *alles*

  Farbthemen regeln Farben

  Font-Themen regeln Zeichensätze

  Innere Themen regeln das *Innere* der Rahmen (Umgebungen wie `block`, `enumerate`)

  Äußere Themen steuern den *Rand* der Rahmen wie Kopf- und Fußzeile, Seitenstreifen, Titel
- `beamer` kommt mit einer Vielzahl solcher Themen.

Themen laden
Ein Thema wird geladen durch einen der folgenden Befehle:

- *\usetheme*
- *\usecolortheme*
- *\usefonttheme*
- *\useinnertheme*
- *\useoutertheme*

Syntax
`\use...theme[options]{name list}`

Beispiel 38 (Themen dieser Folien). `\usetheme{Goettingen}`
`\useinnertheme[shadow]{rounded}`
`\usecolortheme{orchid}`

45.1 Umfassende Themen

- Umfassende Themen kombinieren meist je ein vollständiges Farb-, Font-, inneres und äußeres Thema.
- Ihr Name ist jeweils ein Ort (wo dieses Thema zuerst eingesetzt wurde).

  Beispiele 39. Bergen, Boadilla, Madrid, AnnArbor, CambridgeUS, Pittsburgh, Rochester, Antibes, JuanLesPins, Montpellier, Berkeley, PaloAlto, Goettingen, Marburg, Hannover, Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt, Singapore, Szeged, Copenhagen, Luebeck, Malmoe, Warsay, *default*
- In der *Anleitung* von `beamer` werden diese Themen mit Beispielrahmen illustriert.
- Experimentiere mit den verschiedenen Themen.

Grobunterteilung der Themen

Die verschiedenen umfassenden Themen gliedern sich grob nach der Art von *Navigationshilfe*:

- Gar keine Navigationshilfen (wie default)
- Baumartige Navigationshilfe im Seitenkopf
- Inhaltsverzeichnis im Seitenstreifen
- Inhaltsverzeichnis in Mini-Frame im Kopf
- Abschnitt, Unterabschnitt, Autor, Titel in Kopf- und Fußzeile

Welches Thema soll ich wählen?

Frage
Woran muss ich die Zuhörer auf jeder Seite erinnern?

- Mein Name?
- Titel des Vortrags?
- Aktueller Abschnitt?

Frage
Wie wichtig sind die Strukturelemente?

- Ein dominantes Thema betont die Struktur, ein schlichtes wie default nicht.

45.2 Farb- und Zeichensatzthemen

Zoologie der Farbthemen

- legen die Farben für verschiedene Rahmenbausteine fest
- erzeugen dadurch auch Hintergrundeffekte, zum Beispiel eine Hintergrundeinfärbung von Blöcken
- *vollständige Farbthemen* legen alle Farben fest und heißen nach *Tieren* (albatross, beetle, crane, dove, fly, seagull, wolverine, beaver)
- *innere Farbthemen* legen Farben für innere Elemente fest und heißen nach *Blumen* (lily, orchid, rose).
- *äußere Farbthemen* legen Farben für äußere Elemente fest und heißen nach *Wassertieren* (whale, seahorse, dolphin).

Farbthemen als Modifizierung

- Wir können umfassende Themen durch Laden eines Farbthemas oder eines inneren Themas ändern.
- Wir können auch vollständige Farbthemen durch zusätzliches Laden eines inneren *oder* äußeren Farbthemas ändern.
- Da Farbthemen die Positionierung der Seitenbausteine nicht beeinflussen, können wir sie ganz zum Schluß anpassen.
- legen die Zeichensätze fest
- wichtigste Wahlen: *default*, *serif*, *structurebold*
- Zeichensatzfamilie: lade Pakete wie `mathptmx`, `helvet`
- `beamer` erlaubt im `documentclass`-Befehl viele default font sizes: 8pt, 9pt, 10pt, *11pt*, 12pt, 14pt, 17pt, 20pt

45.3 Innere und äußere Themen

Innere Themen
legen die Marken in `itemize`- und `enumerate`-Umgebungen fest

default dreieckige Markierung

circles kreisförmige Markierung

rectangles quadratische Markierung

rounded abgerundete Markierungen (und abgerundeter Hintergrund in Blöcken)

inmargin Titel von Blöcken in einem Seitenstreifen

Äußere Themen
steuern die Kopf- und Fußzeile, den Titel der Rahmen, die Seitenstreifen und die Platzierung des Logos

default minimalistisch

infolines Aktueller Abschnitt und Unterabschnitt in der Kopfzeile, Autor, Institution und Titel in der Fußzeile

split ähnlich infolines, aber weniger Daten

shadow ähnlich split, mit Schatteneffekten

miniframes Informative Fußzeile, Umfangreiche Navigationsangaben in der Kopfzeile

smoothbars Wie miniframes, aber mit Farbübergang in der Kopfzeile

sidebar Inhaltsverzeichnis im Seitenstreifen

tree Titel, Abschnitt, Unterabschnitt in der Kopfzeile

smoothtree Wie tree mit Farbübergängen

46 Folien- und Artikelversion

Folien- oder Artikelversion erstellen

- Aus der Vorlage für eine Präsentation kann eine *Folien*fassung und eine *Artikel*fassung erstellt werden.
- Die *documentclass*-Option *trans* unterdrückt Overlays und liefert eine *Folienversion* des Vortrags.
- Eine *Artikelversion* wird erzeugt durch Ändern des Programmkopfes:

  `\documentclass{article} \usepackage{beamerarticle}`

  statt

  `\documentclass{beamer}`

- Das Paket *beamerarticle* stellt die die notwendigen `beamer`-Makros bereit.

Handout-Version erstellen

- Man kann auch ein *Handout-Version* der Präsentation erstellen.
- Dazu dient die *documentclass*-Option *handout*.
- Es empfiehlt sich, zusätzlich das Paket *pgfpages* zu laden und den *\pgfpagesuselayout* zu verwenden. Damit kann man mehrere Rahmen auf eine Seite bringen.
- *\pgfpagesuselayout{2 on 1}[a4paper]* bringt zwei Rahmen auf eine Seite.
- *\pgfpagesuselayout{4 on 1}[a4paper,landscape]* bringt vier Rahmen auf eine Seite.

Modusspezifische Befehle

- Die beamer-, Folien-, Handout- und Artikelversion der Präsentation unterscheiden sich nur durch die ersten zwei Zeilen der Eingabedatei.
- Wir können Teile des LaTeX-Codes nur in bestimmten Versionen ausführen lassen.
- Dazu dient der Befehl *\mode<...>* mit Argument:

  beamer beamer-Präsentation

  trans Folien

  handout Handout

  article Artikel

  presentation beamer oder trans

- *\only<beamer>{...}* und Ähnliches funktioniert auch.

Beispiel I

- Oft wird der Titel eines neuen Abschnitts im ersten Rahmen als `frametitle` wiederholt.
- In der Präsentation sieht man die Abschnittsüberschrift nicht.
- Im Artikel werden Abschnitts- *und* Rahmenüberschrift gezeigt.
- `\frametitle<presentation>{Titel}` vermeidet eine Doppelung.

Mehr Beispiele

Beispiel 40. Möchten wir im Artikelmodus anders als in der Präsentation eine *Schrift mit Serifen* wählen oder bestimmte Pakete wie `hyperref` laden, so verpacken wir die entsprechenden Befehle in `\mode<article>{...}`.

- Verwenden wir den `\only`-Befehl, um etwa eine Grafik schrittweise aufzubauen, so werden in den `trans`- und `article`-Versionen *alle* Alternativen nacheinander angezeigt.
- Dies verhindern wir, indem wir die Alternativen, die nicht gezeigt werden sollen, in `\mode<beamer>{...}` oder `\only<beamer>{...}` verpacken.

Unterschiede Artikelversion zu Präsentation

- In der *Artikelversion* werden viele Gestaltungselemente *ignoriert*:
  - `\\`
  - `column(s)`-Umgebungen
  - `\alert` wird `\emph`
  - `frame`-Umgebungen
- Dies ist *in der Regel* wünschenswert.
- `\newline` erzeugt in allen Versionen einen manuellen Zeilenumbruch. Einen manuellen Zeilenumbruch nur in der Artikelversion liefert

  `\only<article>{\\}`.

Teil V
Wie bereite ich einen Vortrag vor?

Worum geht es?

Fragen

- Wie bereite ich einen guten Vortrag vor?
- Was ist in Vorträgen und Artikeln anders?
- Wie sollte ein guter Vortrag aufgebaut sein?
- Worauf sollte ich besonders achten?

47 Regeln für gute Präsentationen

Zeitvorgaben beachten

- Meist hat man am Ende *weniger Zeit* als gedacht.
- Es kommt fast nie vor, dass ein Vortragender zu wenig vorbereitet hat.
- Presse nicht zu viel in eine Präsentation hinein.
- Nicht mehr als ein Rahmen pro Minute, eher deutlich weniger.
- Es ist wahrscheinlich nötig, viele Details wegzulassen.
- Überlege *vorher*, was wirklich wichtig ist und *was weggelassen werden kann.*

Globale Struktur und Gliederung

- Teile das Material in *Abschnitte* und *Unterabschnitte* ein – oder auch Teile bei sehr langen Vorträgen.
- Verwende nicht zu viele Abschnitte.
- *Titel* von Abschnitten und Unterabschnitten sollen *leicht verständlich* sein, damit das Inhaltsverzeichnis schon vor dem eigentlichen Vortrag sinnvolle Information enthält.
- Zum *Abschluß* soll die Hauptbotschaft des Vortrags noch einmal kurz und einfach *zusammengefaßt* werden.

48 Unterschiede zwischen Artikeln und Vorträgen

Zusammenfassung und Einleitung

Artikel

- Die *Zusammenfassung* dient dazu, dem Leser sofort mitzuteilen, ob es sich lohnt, den Artikel zu lesen.
- Die *Einleitung* sollte *alle* wesentlichen Ergebnisse ansprechen, weil erfahrungsgemäß viele Leser nicht mehr als die Einleitung lesen.

Vortrag

- Die *Zusammenfassung* ist am *Schluß* besser untergebracht, weil Zuhörer selten weglaufen.
- Die *Einleitung* soll für möglichst viele Zuhörer *verständlich* sein und ihnen die *Hauptbotschaft* des Vortrags vermitteln.

Satz–Definition–Beweis

Artikel

- *Vollständige Beweise* mit allen *Details* sind Pflicht.
- *Alle* wichtigen Sätze und Definitionen gehören in eine entsprechende *Umgebung*.
- Sätze werden *numeriert*, damit sie leichter zitiert werden können.

Vortrag

- Die Zuhörer können *Details* und *Beweise* im Artikel *nachlesen*, im Vortrag können sie meist nur grobe Ideen davon mitnehmen.
- Satz- und Definitionsumgebungen nur für exakte und vollständige Aussagen
- Sätze und Definitionen werden *nicht numeriert*. Sätze, auf die wir uns beziehen wollen, bekommen einen Namen (Hauptsatz, Gaußlemma, ...)

Literatur

Artikel

- Im *Literaturverzeichnis* muss *alle* relevante Literatur vorkommen, sonst fühlen sich unsere Kollegen *übergangen*.
- präzise zitieren

Vortrag

- Das *Literaturverzeichnis* enthält, wenn es überhaupt vorkommt, eher Empfehlungen für die *weitere Lektüre*.
- Da der Zuhörer das Literaturverzeichnis ohnehin nicht sieht, reichen *grobe Angaben* wie: nach einem Satz von Grauert ...

Zeichensätze, Typographie, Formulierung

Artikel

- *Serifen* führen das Auge und erleichtern das Lesen.
- *kursive Hervorhebung*
- *Blocksatz* wird sogar durch Worttrennung erzwungen.
- Formuliere immer in *ganzen Sätzen*.

Vortrag

- Bei schlechter Auflösung verschwimmen Serifen und verringern den Kontrast
- *Farbige Hervorhebung*
- Zeilenumbrüche orientieren sich an Phrasen im Text. Man muß ggf. von Hand eingreifen.
- Verwende eher *knappe Phrasen*.

Was haben Artikel und Vorträge gemeinsam?

- Formuliere einfach und verständlich.
- Verwende viel Zeit darauf, Formulierungen zu straffen.
- Verben sind oft prägnanter und farbiger als Nomen und Adjektive.

Beispiel 41. <+-> Vergleiche: Die Konvergenz der Folge (a_n) impliziert die Existenz eines Häufungspunkts. Die Folge (a_n) hat einen Häufungspunkt, weil sie konvergent ist. Die Folge (a_n) hat einen Häufungspunkt, weil sie konvergiert.

49 Ratschläge zur Seitengestaltung

Einleitung und Zusammenfassung

- In der *Einleitung* erklärt man,
  - was das *Problem* des Vortrags ist,
  - warum es interessant ist (*Motivation*),
  - was die *Hauptergebnisse* des Vortrags sind.
- In der *Zusammenfassung* werden die *Hauptergebnisse* und eventuell wichtige Methoden *kurz* und *allgemeinverständlich* zusammengefaßt.
- Die Zusammenfassung umfaßt höchstens *einen* Rahmen und sollte möglichst nicht mehr als fünf Sätze enthalten.
- Zuhörer sind am *Beginn* und *Ende* des Vortrags besonders *aufmerksam*.

  Dort sollte man die wichtigsten Botschaften des Vortrags unterbringen.

Wie gestalten wir eine Seite?

Regel
Jede Seite braucht einen *verständlichen und erklärenden Titel*.

Frage
Wie viel darf auf eine Seite?

- besser zu wenig als zu viel
- Richtwert: *20 bis 40 Worte* pro Rahmen, jedenfalls nicht mehr als 80.
- Benutze nicht kleinere Schriften oder die `shrink`-Option, um mehr auf eine Seite zu quetschen.

  Regel

  Schreibe nur Dinge auf, die im Vortrag erklärt werden.

Text

- Zuhörer sehen einen Rahmen nur eine Minute, da bleibt ihnen kaum Zeit, komplizierte Sätze zu verfolgen.

  Regel

  Verwende kurze Sätze oder Phrasen statt Sätzen.

Wie gliedere ich eine Seite?

- Verwende möglichst *Grafiken*, `block`-Umgebungen oder `description`-Umgebungen.
- Mehrere *Spalten* sind gut, Fußnoten und verschachtelte Listen sind schlecht.

Regel
Hebe wichtige Worte durch `\alert` hervor.

Was ist bei Grafiken zu beachten?

- nicht mehr Details, als auch erklärt werden
- *Vektorgrafiken* sollen ähnliche *Farbregeln* verwenden wie der Text.
- Vermeide überflüssige Effekte: statt die Aufmerksamkeit der Zuhörer zu gewinnen, lenken sie sie eher ab.

Farben sinnvoll einsetzen

Regel
Setze Farben *sparsam* ein.

- ausreichend *Kontrast*

  Beispiel 42. Rote Schrift auf blauem Grund ist kaum zu lesen, ebenso wenig leuchtende Farben auf weißem Grund.

- Hintergrundschattierungen verringern den Kontrast

Warnung
Bildschirme haben besseren Kontrast als Projektoren.

Soll ich Formeln vermeiden?

Regel
Man beschränke sich auf die wichtigsten Formeln.

- Formeln haben oft eine so hohe *Informationsdichte*, dass die Zuhörer viel Zeit brauchen, sie zu verarbeiten.
- Viele mathematische *Formeln* lassen sich auch einfach in deutsche *Sätze* verwandeln.
- Diese sind in aller Regel leichter zu verarbeiten.

50 Vorgehensweise beim Erstellen einer Präsentation

Wie erstelle ich eine Präsentation?

1. Inhalt planen, inklusive *nicht*-Inhalt
2. Zusammenfassung schreiben
3. Titelseite, Inhaltsverzeichnis, Gliederung anlegen
4. Rahmen erzeugen: zunächst nur Titel, Inhalt, Struktur, Hervorhebung
5. Präsentation testen, dabei zu lange Rahmen kürzen und Zeilenumbruch gestalten
6. Zusammenfassung überprüfen
7. Overlay-Angaben einfügen
8. Präsentation inklusive Overlay-Angaben testen

   Achte dabei auch auf die Zeit!

Teil VI
Befehle und Umgebungen in LaTeX definieren

Worum geht es?

Antwort
Wir lernen, in LaTeX eigene *Befehle* und *Umgebungen* zu definieren.

Frage 43. *Wofür* braucht man das?

- *Vereinfachung* der Eingabe
- *Einheitlichkeit* der Gestaltung
- Vereinfachung von *globalen Änderungen*

51 LaTeX-Befehle und -Umgebungen ohne Parameter

LaTeX-Befehle ohne Parameter

- `\newcommand` definiert einen neuen Befehl.
  Beispiel 44. `\newcommand{\e}{\textup{e}}` definiert den Befehl `\e`.

  Beim Lesen der Datei wird `\e` jeweils durch `\textup{e}` ersetzt.

Beispiele – Befehle als Eingabeerleichterung I
Beispiel 45. `\newcommand{\N}{\mathbb{N}}`
`\newcommand{\7}{\textbackslash}`

- `\N` erzeugt $\mathbb{N}$ (im Mathematikmodus).
- `\7` erzeugt \ (im Textmodus).
- Dies vereinfacht die Eingabe dieser Vorlesung ungemein.
- Noch besser wäre vielleicht `\newcommand{\7}{\ttfamily\textbackslash}`
- Dann setzt `{\7Befehl}` gleich `\Befehl` in Schreibmaschinenschrift.

Frage 46. Wozu dienen die Klammern?

Beispiele – Befehle als Eingabeerleichterung II
Beispiel 47. `\newcommand{\defeq}{\mathrel{\vcentcolon=}}`

- `\defeq` erzeugt im Mathematikmodus das Zeichen := und sorgt dafür, dass es als Relation gesetzt wird.
- Das Zeichen : ist ein vertikal zentrierter Doppelpunkt aus dem Paket `mathtools`.

Beispiele – Befehle als Eingabeerleichterung III

Beispiel 48. `\newcommand{\nbd}{\nobreakdash-\hspace{0pt}}`

- `\nbd` erzeugt einen Bindestrich, bei dem keine Worttrennung erfolgt, und erlaubt im Rest des Wortes die Worttrennung.
- Zum Beispiel schreibe ich `$K$\nbd{}Vektorraum` für K-Vektorraum.
- Durch `\hspace{0pt}` wird ein Leerraum ohne Breite eingefügt – dadurch beginnt für LaTeXs Trennalgorithums ein neues Wort.

Beispiele – globale Änderungen erleichtern

Beispiel 49. `\newcommand{\Base}{Z} \newcommand{\base}{z}`

- Angenommen, in einer Arbeit treten mehrere Räume auf.
- Zunächst nennen wir sie X, Y, Z, und ihre Elemente x, y, z.
- Später bemerken wir, dass Sie z auch für komplexe Zahlen benutzen, und fürchten, dass dies den Leser verwirrt.
- Aber jetzt ist es schwer, die Doppelung der Notation aufzuheben, weil wir jedes z anschauen müssen.
- Durch das Makro können wir bei konsequenter Anwendung mit einem Federstrich ihre Notation beliebig ändern und solche Konflikte wieder beheben.

Beispiele – Einheitliche Gestaltung

Beispiel 50. `\newcommand{\Endframe}{{\color{yellow}\ensuremath{\bullet}}}`

- erzeugt am Ende eines Rahmens •.
- `\color` setzt die Farbe und versteht neben numerischen Parametern auch viele Standardnamen – lade Paket `xcolor`.
- `\ensuremath` sorgt dafür, dass das Argument immer im Mathematikmodus gesetzt wird.

  Dadurch kann `\Endframe` sowohl im Text als auch in Formeln auftreten.•

Existierende Befehle ändern

- `\renewcommand` ändert die Definition eines schon existierenden Befehls.

Beispiel 51 (*Nicht empfehlenswert*).
`\renewcommand{\epsilon}{\varepsilon}`
Jetzt erzeugen sowohl `\epsilon` als auch `\varepsilon` das Symbol ε.

Vorsicht
Ändern bestehender LaTeX-Befehle kann überraschende Wirkungen haben.

52 Befehle mit Parametern

Befehle mit Parametern

- Durch eine Option für `\newcommand` erzeugen Sie Befehle mit Parametern.

Syntax
`\newcommand{\Name}[Anzahl]{Definition}`

- Dies definiert einen Befehl `\Name` mit `Anzahl` Parametern.
- In der Definition steht `#1`, `#2`, ... für den ersten, zweiten, ... Parameter.
- Die Variante `\newcommand*` erzeugt einen Befehl, der nur *kurze Argumente* annimmt (keine Leerzeilen erlaubt).

Beispiel – Konjugation

Beispiel 52. `\newcommand*{\conj}[1]{\overline{#1}}`

- Also erzeugt `\conj{a+b} = \conj{a}+\conj{b}` die Gleichung $\overline{a+b} = \overline{a}+\overline{b}$.
- Wegen der *-Form sind nur kurze Argumente erlaubt.
- Wenn ich mich vertippe und Klammern weglasse, führt das zu sinnvolleren Fehlermeldungen.

Beispiel – Betragsstriche

Beispiel 53. `\newcommand*{\abs}[1]{\lvert#1\rvert}`

- `\abs{x}` erzeugt im Mathematikmodus $|x|$.
- Dabei werden die linken und rechten Betragsstriche verwendet.

Beispiel – Ideal

Beispiel 54. `\newcommand*{\ideal}[2]{\langle#1\mid #2\rangle}`

- Der Befehl `\ideal` hat zwei Parameter.

Frage 55. Was erzeugt `\ideal{x}{y}`?

Befehle mit optionalen Parametern

Syntax
`\newcommand{\Name}[Anzahl][default]{Definition}`

- Falls `default` angegeben wird, ist der Parameter optional und es wird, falls beim Aufruf des Befehls der optionale Parameter fehlt, `default` verwendet.

Beispiel 56. Sie wollen für einen Hilbertraum kalligraphische Buchstaben benutzen. Fast immer heißt ihr Hilbertraum $\mathcal{H}$, manchmal benutzen Sie aber auch andere Buchstaben.

- `\newcommand{\Hils}[1][H]{\mathcal{#1}}`
- jetzt liefert `\Hils` $\mathcal{H}$ und `\Hils[L]` $\mathcal{L}$.

53 Umgebungen definieren

- Durch `\newenvironment` wird eine neue Umgebung definiert.

Syntax
`\newenvironment{Name}[Anzahl]{Anfang}{Ende}`

- erzeugt die Umgebung `Name`, die `Anzahl` Parameter benötigt.
- Am Anfang der Umgebung wird `Anfang` eingefügt, am Ende `Ende`.

Frage 57. Welche Umgebungen brauchen Parameter?

Beispiel – Simulieren der `block`-Umgebung

- Eine Version der `block`-Umgebung für Artikel soll als Parameter den Titel des Blocks verwenden und diesen in einer eigenen Zeile in Fettschrift setzen.
- Vor und nach der Umgebung soll sie Leerraum einfügen.

```
\newenvironment{block}[1]{\medskip%
\begin{flushleft}#1\end{flushleft}\smallskip}%
{\par\medskip}
```

`\par` beendet den Absatz

`\medskip` mittelgroßer vertikaler Leerraum

`\smallskip` kleiner vertikaler Leerraum

Frage 58. Wozu sind die Prozentzeichen % notwendig?