

L^AT_EX Grundlagen

Computerlabor

Thomas

12.04.2010

Outline

- ▶ Ziel
- ▶ Geschichte
- ▶ Was macht \LaTeX anders?
- ▶ \LaTeX installieren
- ▶ Ein einfaches \LaTeX Dokument
- ▶ Verbesserungen / Anpassungen an Deutsch
- ▶ Crash-Kurs Formatierungen, Tabellen, Bilder
- ▶ Wie weiter?

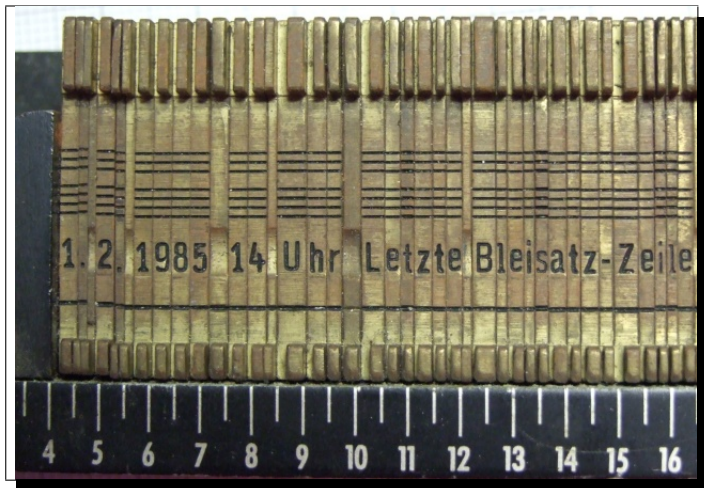
Ziel

- ▶ Grundlegende Einführung in \LaTeX
(wir werden keine fortgeschrittenen Themen besprechen)
- ▶ Kurzer Einblick, was \LaTeX anders macht als bspw. OpenOffice
- ▶ Ein Beispieldokument erstellen, welches auf deutschsprachige Verhältnisse ausgerichtet ist
- ▶ Kennenlernen, welche Befehle zur Formatierung es gibt
- ▶ Vorbereitung auf den nächsten Workshop zum Präsentieren

Geschichte

- ▶ Die Grundlage für \LaTeX hat Donald E. Knuth gelegt
- ▶ Er war Professor an für Informatik an der Stanford University
- ▶ Eines seiner Projekte war die Verfassung von *The Art of Computer Programming*
- ▶ Dieses Projekt begann in 1962 und war sehr ambitioniert - er wollte in sechs Bänden als erster Autor überhaupt die Programmierung behandeln
- ▶ Allerdings war er mit der Qualität des Buchsatzes nicht zufrieden
- ▶ Damals wurde noch der Bleisatz benutzt, der die Vorlage für die Druckerpresse war

Geschichte



Geschichte

- ▶ Ab 1977 begann er sein eigenes Textsatzsystem zu entwickeln und wollte es möglich machen, dass ein Computer die Aufgabe vom Schriftsetzer übernehmen kann
- ▶ Sein Textsatzsystem heisst T_EX und wurde 1986 fertig gestellt
- ▶ In den 70er und 80er Jahren waren grafische Oberflächen noch nicht entwickelt oder steckten noch in den Kinderschuhen
- ▶ Deshalb verarbeitete dieses Textsatzsystem reinen Text, der durch spezielle Steueranweisungen ergänzt war
- ▶ T_EX ist die Abkürzung des griechischen Wortes für „Kunst“ und das X wird als „Ach-Laut“ ausgesprochen
- ▶ Sozusagen war T_EX in der damaligen Zeit das erste Programm, welches es ermöglichte einen Computer für Publishing/Layouting zu nutzen

Geschichte

- ▶ Von der Technik her handelt es sich bei T_EX um einen Interpreter - er interpretiert eine Textdatei und erkennt darin vorkommende Steuerbefehle
- ▶ T_EX selber verfügt über rund 300 eingebaute Befehle
- ▶ Es können *Makros* (und somit eigene Funktionen) definiert werden - häufig gebrauchte Abfolgen von Befehlen lassen sich damit einfach zusammenfassen und wiederverwenden
- ▶ Der Textsatz erfolgt dabei nach verschiedenen Algorithmen
- ▶ Für T_EX entwickelten sich schnell eine Vielzahl an Makros, die über die damaligen Datennetze ge- und verteilt wurden

Geschichte

- ▶ \LaTeX ist eine Sammlung von Makropaketen, die eine sehr einfache Nutzung von \TeX ermöglichen
- ▶ Die Komplexität wird sozusagen durch die Makros abstrahiert
- ▶ \LaTeX wurde von Leslie **L**amport entwickelt, deshalb der Name des Makropaketes
- ▶ \LaTeX selber wiederum kann durch Packages erweitert werden
- ▶ Deshalb wird normalerweise nicht \TeX oder \LaTeX alleine installiert, sondern eine Distribution die aus \TeX + \LaTeX + Packages + weiterem besteht
- ▶ Die bekanntesten zwei sind *teTeX* und *TeX Live*
- ▶ TeX Live ist heutzutage beinahe überall anzutreffen, teTeX wird nicht mehr aktiv weiterentwickelt

Geschichte

- ▶ Durch die Entwicklung von \LaTeX wurde das gesamte Textsatzsystem rasch populär und wird vor allem an Universitäten und Fachhochschulen benutzt - vor allem im mathematisch-naturwissenschaftlichen Bereich
- ▶ Es existieren aber auch spezielle Packages bspw. für den Ausdruck von Lautschriften für Linguisten, zum Bibliografieren für Geisteswissenschaftler oder für den Notensatz für Musiker
- ▶ In vielen Firmen wird \LaTeX zudem dafür eingesetzt, Dokumente automatisch zu generieren

Was macht LaTeX anders?

- ▶ Office-Software wie OpenOffice sind WYSIWYG-Programme - *What you see is what you get*
- ▶ \LaTeX verfolgt den Ansatz WYGIWYM - *What you get is what you mean*
- ▶ In OpenOffice markieren wir ein Wort und drücken auf den „Kursiv“ Button
- ▶ In \LaTeX schreiben wir folgendes und erhalten ein kursives Wort:

```
1 % Was mit % beginnt wird vom Interpreter ignoriert
2 % textit heisst "text italic" (kursiver Text)
3 \textit{Wort}
```

L^AT_EX installieren

- ▶ Für Ubuntu/Debian
Installation der Pakete *latex-beamer* und *texlive-lang-german*
(*latex-beamer* benötigen wir erst nächstes Mal, aber es zieht uns alle Pakete als Abhängigkeit rein, die wir benötigen)
- ▶ Für Mac OS X
Installation von MacT_EX
<http://www.tug.org/mactex/>
- ▶ Für Windows
Installation von TeX Live über das Internet
<http://www.tug.org/texlive/acquire-netinstall.html>
Alternative: Download des TeX Live DVD ISO Images und brennen auf DVD
<http://www.tug.org/texlive/acquire-iso.html>

Ein einfaches LaTeX Dokument

```
1 \documentclass{article}
2
3 \begin{document}
4
5 \section{Abschnitt 1}
6
7 Dies ist der Abschnitt 1.
8
9 \section{Abschnitt 2}
10
11 \subsection{Abschnitt 2.1}
12
13 Hier haben wir nun einen Unterabschnitt.
14
15 \end{document}
```

Das einfache LaTeX Dokument verarbeiten

- ▶ \LaTeX Dokumente sind reine Textdokumente
- ▶ Dabei können wir jeden Texteditor nutzen
- ▶ Alternativen sind *kile*, *winefish*, *LyX* und *TeXShop*, welches spezielle \LaTeX Editoren sind
- ▶ Wir speichern das Dokument dann mit der Endung `.tex` ab und lassen `pdflatex meindokument.tex` laufen - wir erhalten dann `meindokument.pdf`
- ▶ Daneben erzeugt \LaTeX noch diverse andere Dateien - wir ignorieren diese, sie werden von \LaTeX intern benutzt
- ▶ Die Datei mit der Endung `.toc` ist bspw. die *Table of Contents*, die \LaTeX intern führt

Wieso der Aufwand?

- ▶ Wir müssen spezielle Kommandos eingeben und von Hand einen Befehl aufrufen
- ▶ Viele denken, dass sei zu aufwendig
- ▶ Vorteil bei \LaTeX ist aber der schönere Textsatz
- ▶ Viele Benutzer halten es zudem auch für einen Vorteil, beim Schreiben nicht abgelenkt zu werden durch grafische Elemente
- ▶ Zudem werden Programme wie OpenOffice bei Dokumenten ab ca. 80 bis 100 Seiten immer langsamer und grosse Dokumente sind schwierig zu handhaben

Was haben wir im einfachen Dokument?

- ▶ Wir definieren eine `documentclass` - in diesem Falle `article`
- ▶ Wir drücken den `begin` und das `end` des Dokumentes aus
- ▶ Wichtig: Was einmal mit `begin` „geöffnet“ wurde, muss auch mit `end` wieder geschlossen werden
- ▶ Wir haben zwei Kapitel und ein Unterkapitel
- ▶ Wir haben den eigentlichen Text ganz normal eingegeben
- ▶ \LaTeX versteht den eigentlichen Text als einen Abschnitt (*Paragraph*)

Schönerer Titel

- ▶ Das Dokument ist sehr knapp und hat keinen Titel
- ▶ Dies können wir ändern
- ▶ \LaTeX kann mit Titeln, Abstracts, Autorenangaben und mehr umgehen

Titel, Autor und Datum angeben

```
1 % Wir fügen nach der Dokumentenklasse etwas ein  
2 \documentclass{article}  
3  
4 % Wir geben einen Titel, einen Autor und ein Datum an  
5 \title{Mein LaTeX Dokument}  
6 \author{Vorname Nachname}  
7 % \today setzt immer das aktuelle Datum ein  
8 % Alternative: \date{5. Januar 2010}  
9 \date{\today}  
10  
11 % Den Rest des Dokumentes lassen wir gleich  
12 [...]
```

Titel und Inhaltsverzeichnis einfügen

```
1 % Wir fügen nach dem Beginn des Dokuments ein  
2 \begin{document}  
3  
4 % Den Titel generieren  
5 \maketitle  
6  
7 % Ein Abstract des Dokuments  
8 % Wenn ich den nicht will, lasse ich ihn weg  
9 \begin{abstract}  
10 Meine ersten Versuche mit LaTeX.  
11 \end{abstract}  
12  
13 % Ein Inhaltsverzeichnis generieren  
14 \tableofcontents  
15  
16 % Einen Seitenumbruch einfügen  
17 \newpage  
18  
19 % Ab hier lassen wir alles gleich...  
20 [...]
```

Dokument neu übersetzen

- ▶ Wir lassen nochmals `pdflatex meindokument.tex` laufen
- ▶ Wir sehen im PDF gleich den Unterschied
- ▶ Das Inhaltsverzeichnis wird noch leer sein - erst beim zweiten Lauf von `pdflatex` wird dieses richtig aktualisiert
- ▶ Uns stören aber die englischen Bezeichnungen...

LaTeX auf Deutsch umschalten

- ▶ \LaTeX ist standardmässig auf Englisch eingestellt
- ▶ Wir können dies einfach ändern - \LaTeX kann durch Packages ergänzt und verändert werden
- ▶ Wir benötigen dazu das Package `ngerman` und binden dies **nach** der Dokumentenklasse ein. Danach nochmals `pdflatex` laufen lassen.

```
1 \documentclass{article}
2
3 \usepackage{ngerman}
4
5 % Den Rest lassen wir gleich ...
6 [...]
```

Kurze Info zu Packages

- ▶ Solche Packages werden mit der Distribution (teTeX oder TeX Live) ausgeliefert
- ▶ Auf Ubuntu/Debian sehen wir, dass TeX Live in viele kleine Pakete verteilt ist
- ▶ Je nachdem kann es also auf Ubuntu/Debian vorkommen, dass wir eines der Pakete nachinstallieren müssen
- ▶ `ngerman` kommt auf Ubuntu bspw. durch das Paket *texlive-lang-german* aufs eigene System
- ▶ Es ist möglich, von CTAN (<http://www.ctan.org/>) zusätzliche Packages downzuloaden und zu installieren - dies lassen wir in diesem Workshop aber aus

Versteht LaTeX Umlaute?

- ▶ Natürlich versteht \LaTeX Umlaute - nur können diese in der Standardeinstellung nicht direkt eingegeben werden
- ▶ Wir fügen mal folgendes direkt **innerhalb** des Dokuments ein:

```
1 \subsection{Abschnitt 2.1}
2
3 Hier haben wir nun einen Unterabschnitt.
4
5 Erste Umlaute: ö ä ü
6 Zweite Umlaute: "o "a "u
7
8 \end{document}
```

Versteht LaTeX Umlaute?

- ▶ Das Ergebnis ist nicht befriedigend
- ▶ Die zweiten Umlaute werden angezeigt - durch "o wird ein ö erzeugt
- ▶ Die direkt auf dem Keyboard eingegebenen Umlaute werden aber nicht beachtet
- ▶ Dies liegt daran, dass \LaTeX standardmässig nur den amerikanischen Zeichensatz versteht
- ▶ Wer bei Umlauten also nicht immer "o schreiben will, tut gut daran, dies umzustellen

Das inputenc Package

```
1 \documentclass{article}
2
3 \usepackage{ngerman}
4
5 % Wir binden das inputenc Package ein
6 \usepackage[latin1]{inputenc}
7
8 % Den Rest lassen wir gleich
9 [...]
```


Das inputenc Package

- ▶ Wir haben etwas neues gelernt: Das `usepackage`-Kommando kann mehrere Optionen haben - in den geschweiften Klammern steht immer der Name des Packages, in den eckigen die Parameter, die dem Package gegeben werden
- ▶ Wir haben `inputenc` eingebunden und diesem die Option `latin1` mitgegeben
- ▶ Nun weiss \LaTeX , dass es die Eingabedatei mit `latin1` bzw. ISO-8859-1 Zeichensatz lesen soll
- ▶ Wenn euer Editor die Datei in diesem Zeichensatz abgespeichert hat, solltet ihr nun im PDF bereits beide Arten von Umlauten lesen können

Das inputenc Package

- ▶ Wenn euer Editor nicht in diesem Zeichensatz abgespeichert hat, werdet ihr eine Fehlermeldung bekommen beim Lauf von `pdflatex`
- ▶ Viele Linux-Distributionen haben UTF-8 voreingestellt
- ▶ Manchmal kann es im Editor auch konfiguriert werden
- ▶ Wenn euer Editor UTF-8 ausgibt, dann machen wir als Alternative zu vorher:

```
1 % Wir binden das inputenc Package ein  
2 % Dieses mal mit UTF-8  
3 \usepackage[utf8]{inputenc}  
4  
5 % Den Rest lassen wir gleich  
6 [...]
```

Umbrüche

- ▶ Im PDF bleibt nun nur noch ein Schönheitsfehler: Die beiden Umlaute-Zeilen sind auf derselben Zeile gesetzt.
- ▶ \LaTeX geht davon aus, dass sie zum selben Absatz gehören, da keine Leerzeile dazwischen steht.
- ▶ Machen wir also ein Leerzeile rein:

```
1 \subsection{Abschnitt 2.1}
2
3 Hier haben wir nun einen Unterabschnitt.
4
5 Erste Umlaute: ö ä ü
6
7 Zweite Umlaute: "o "a "u
8
9 \end{document}
```

Umbrüche

Wir haben auch die Möglichkeit, andere Umbrüche einzufügen:

```
1 % Ein doppelter Schrägstrich erzeugt eine  
2 % Leerzeile in der Ausgabe  
3 Irgendein Text \\  
4  
5 % newpage beendet die Seite und startet eine neue  
6 \newpage
```

Optionen der documentclass

- ▶ Wie auch die Packages, versteht auch die Dokumentenklasse gewisse Optionen
- ▶ Wir können die Standardschriftgröße sowie das Papier ändern
- ▶ Wir werden mit folgendem sehen, dass nun die Schriftgröße im gesamten Dokument grösser wird, zudem setzen wir noch explizit A4 als Seitengröße (falls unsere L^AT_EX Installation dies nicht schon selber macht):

1

```
\documentclass[12pt , a4paper]{ article }
```

Ausgegebene Schriftart verbessert

- ▶ \LaTeX benutzt standardmässig auf amerikanische Verhältnisse ausgerichtete Schriftarten
- ▶ Ein Umlaut wird dabei aus zwei Zeichen zusammengesetzt: Dem Buchstaben und dem doppelten Punkt
- ▶ Wenn ihr aus dem PDF die Umlaute rauskopiert, seht ihr, dass ihr pro Umlaut zwei Zeichen erhält
- ▶ Für unseren Sprachraum gibt es die sog. T1 Fontkodierung, die im speziellen bei Umlauten schönere Resultate erzeugt:

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

Zusammenfassung

- ▶ Wir haben nun von einem simplen Dokument ausgehend wichtige Anpassungen vorgenommen
- ▶ Wir haben dafür gesorgt, dass \LaTeX richtig für deutsche Verhältnisse angepasst ist
- ▶ Wir haben dafür gesorgt, dass wir eine Titelseite und ein Inhaltsverzeichnis erhalten
- ▶ Wir haben nun sozusagen einen minimalen Draft, den wir als Startpunkt für das Verfassen von Dokumenten nutzen können
- ▶ Wenn wir das nächste Mal das Präsentieren mit \LaTeX angehen, werden wir eine andere Dokumentenklasse nutzen, benötigen die Anpassungen an die deutschen Verhältnisse aber auch dann wieder

Packages

- ▶ Wir haben auch gesehen, dass wir Packages einbinden können
- ▶ Es existiert eine grosse Menge an Packages für verschiedene Arbeiten
- ▶ So existieren bspw. spezielle Packages um schöne Kopf- und Fusszeilen zu machen
- ▶ Es existieren auch Packages um einfacher mit Grafiken umzugehen
- ▶ Wer \LaTeX nutzt wird sehr schnell auf sehr viele nützliche Packages stossen

Wie bekomme ich Hilfe?

- ▶ Eine gute Hilfe ist das Buch auf Wikibooks:
<http://en.wikibooks.org/wiki/LaTeX>
- ▶ Eine deutsche Variante gibt es auf:
<http://de.wikibooks.org/wiki/LaTeX>
- ▶ Es gibt eine Deutsche T_EX-FAQ:
<http://projekte.dante.de/DanteFAQ/WebHome>
- ▶ *the Comprehensive TeX Archive Network* (CTAN):
<http://www.ctan.org/>
- ▶ Hilfreich ist oft auch eine Suche im Web im Sinne von:
latex documentclass oder *latex fontenc package*

Weiter im Crash-Kurs

- ▶ Über die Verwendung von \LaTeX könnte stundenlang referiert werden. . .
- ▶ Nachdem wir die erste Einrichtung zusammen gemacht haben, gehen wir nun ein bisschen schneller vorwärts
- ▶ Formatierungen, Tabellen und Bilder werden gezeigt
- ▶ Probiert anschliessend alles selber aus, mit Hilfe der Slides
- ▶ Formatierungen, Tabellen und Bilder werden nächstes Mal im Workshop zum Präsentieren genau gleich wieder verwendet werden

Weiter im Crash-Kurs

- ▶ Wichtig ist, dass ihr euch bewusst seid, dass es in \LaTeX einen Unterschied zwischen dem einfachen Befehl und einer Umgebung (Environment) gibt
- ▶ Mit `begin` wird eine Umgebung geöffnet, mit `end` muss sie wieder geschlossen werden
- ▶ Mindestens eine Umgebung hat jedes \LaTeX Dokument, nämlich `document` (der eigentliche Inhalt des Dokumentes)
- ▶ Was vor der `document` Umgebung passiert, sind Einrichtungsarbeiten, wie eben das Einbinden von Packages
- ▶ Umgebungen dürfen ineinander verschachtelt werden, wie wir noch sehen werden
- ▶ Andere Befehle, wie `section`, öffnen **keine** Umgebung, sondern sind ein einfacher Befehl bzw. Makro

Text-Formatierungen

Dieses Wort ist *kursiv* (italic)

Dieses Wort ist *slanted* (slanted)

Dieses Wort ist **fett** (bold face)

Dieses Wort ist teletype (teletype)

Dieses Wort ist unterstrichen

Dieses Wort ist *ausgezeichnet* (emphasis)

Natürlich kann jeder dieser Befehle auch mit mehr als einem Wort umgehen.

Text-Formatierungen

```
1 Dieses Wort ist \textit{kursiv} (italic) \\
2 Dieses Wort ist \textsl{slanted} (slanted) \\
3 Dieses Wort ist \textbf{fett} (bold face) \\
4 Dieses Wort ist \texttt{teletype} (teletype) \\
5 Dieses Wort ist \underline{unterstrichen} \\
6 Dieses Wort ist \emph{ausgezeichnet} (emphasis) \\
7 \textsl{\textbf{Natürlich kann jeder dieser Befehle auch mit}} \\
8 \texttt{mehr als einem Wort umgehen.}
```

Inhalt strukturieren

- ▶ Der Inhalt kann strukturiert werden in verschiedene Kapitel
- ▶ Diese Kapitel werden automatisch nummeriert
- ▶ Sie erscheinen automatisch im Inhaltsverzeichnis
- ▶ Wenn wir ein PDF erzeugen, wird dabei automatisch ein PDF-Inhaltsverzeichnis erstellt
- ▶ Viele PDF-Viewer zeigen das Inhaltsverzeichnis dann an und erlauben das schnelle navigieren

Inhalt strukturieren

```
1 \section{Kapitel}  
2 \subsection{Unterkapitel}  
3 \subsubsection{Unterunterkapitel}  
4 \paragraph{Abschnitt}  
5 \subparagraph{Unterabschnitt}
```

Listen – itemize

- ▶ Punkt 1
- ▶ Punkt 2
- ▶ Punkt 3

Listen – itemize

```
1 % Denkt dran: was beginnt...
2 \begin{itemize}
3   \item Punkt 1
4   \item Punkt 2
5   \item Punkt 3
6 \end{itemize}
7 % ...muss auch enden
```

Listen – enumerate

1. Erster Punkt
2. Zweiter Punkt
3. Dritter Punkt

Listen – enumerate

```
1 \begin{enumerate}
2   \item Erster Punkt
3   \item Zweiter Punkt
4   \item Dritter Punkt
5 \end{enumerate}
```

Listen – description

Erstens Mein erster Punkt

Zweitens Mein zweiter Punkt

Drittens Mein dritter Punkt

Erstens

Mein erster Punkt

Zweitens

Mein zweiter Punkt

Drittens

Mein dritter Punkt

Listen – description

```
1 \begin{description}
2   \item[Erstens] Mein erster Punkt
3   \item[Zweitens] Mein zweiter Punkt
4   \item[Drittens] Mein dritter Punkt
5 \end{description}
6
7 % hfill füllt horizontal, wir sagen LaTeX dann
8 % das ein Zeilenumbruch gemacht werden soll
9 \begin{description}
10  \item[Erstens] \hfill \\
11    Mein erster Punkt
12  \item[Zweitens] \hfill \\
13    Mein zweiter Punkt
14  \item[Drittens] \hfill \\
15    Mein dritter Punkt
16 \end{description}
```

Fussnoten

1 Ich möchte in meinem Text eine Fussnote
2 anbringen. `\footnote{Als Beispiel diese.}`

Zitate

Ein kurzes Zitat, max. ein Absatz.

Ein längeres Zitat, auch mehrere Absätze lang.

Dabei wird jeder Absatz neu eingerückt.

Beim „quote“ oben wird nicht eingerückt.

Zitate

```
1 \begin{quote}
2 Ein kurzes Zitat , max. ein Absatz.
3 \end{quote}
4
5 \begin{quotation}
6 Ein längeres Zitat , auch mehrere Absätze lang.
7
8 Dabei wird jeder Absatz neu eingerückt.
9
10 Beim "'quote'" oben wird nicht eingerückt.
11 \end{quotation}
```


Spezielle Zeichen

Ein Text „in Anführungs und Schlusszeichen“

Französische »Anführungszeichen«

Englische “Anführungs und Schlusszeichen”

Ein Zeichen kann eine Tildë haben

Ein Zeichen kann ein Çirconflex haben

Ein Zeichen mit Akzènt oder Akzént

Striche: X-Ray, Seite 28–34, ja — oder nein?

Spezielle Zeichen wie ©, ®, ¶, <, >

Weitere spezielle Zeichen wie #, &, {, }, §

Das % nicht als Kommentar interpretiert

Ein Absatz kann von Punkten gefolgt sein. . .

Das Wort L^AT_EX oder T_EX

Spezielle Zeichen

```
1 Ein Text "in Anführungs und Schlusszeichen" \\  
2 Französische \frqq Anführungszeichen\flqq \\  
3 Englische "Anführungs und Schlusszeichen" \\  
4 Ein Zeichen kann eine Tild\~{e} haben \\  
5 Ein Zeichen kann ein \c{C}irconflex haben \\  
6 Ein Zeichen mit Akz\'{e}nt oder Akz\'{e}nt \\  
7 Striche: X-Ray, Seite 28--34, ja --- oder nein? \\  
8 Spezielle Zeichen wie \copyright, \textregistered, \P↵  
9 , \textless, \textgreater \\  
10 Weitere spezielle Zeichen wie \#, \&, \{, \}, \S \\  
11 Das \% nicht als Kommentar interpretiert \\  
12 Ein Absatz kann von Punkten gefolgt sein\ldots \\  
13 Das Wort \LaTeX\ oder \TeX
```

Verschlucken von Leerzeichen

- ▶ Achtung: Es gibt gewisse Kommandos, die das darauf folgende Leerzeichen „schlucken“
- ▶ Im Beispiel vorhin waren dies die französischen Anführungszeichen und die \LaTeX oder \TeX Symbole
- ▶ Wenn man das verschlucken verhindern will, dann wird ein Schrägstrich nach dem Kommando platziert:

1 Nach `\LaTeX` wird das Leerzeichen verschluckt .
2 Mit `\LaTeX\` wird es nicht verschluckt .

Crash-Kurs: Tabellen

- ▶ \LaTeX kann auch mit Tabellen umgehen
- ▶ Nicht erschrecken, es sieht ein bisschen kompliziert aus 😊

Eine simple Tabelle

Spalte 1	Spalte 2	Spalte 3
eins vier	zwei fünf	drei sechs

Eine simple Tabelle

```
1 \begin{tabular}{ | l | c || r | }  
2 \hline  
3 \textbf{Spalte 1} & \textbf{Spalte 2} & \textbf{↔  
   Spalte 3} \\ \hline  
4 \hline  
5 eins & zwei & drei \\ \hline  
6 vier & fünf & sechs \\ \hline  
7 \hline  
8 \end{tabular}
```

Die Spezifikation der Tabelle

- ▶ `| l | c | | r |` nach dem Beginn der `tabular` Umgebung ist die sog. *Table-Spec*
- ▶ Sie gibt an, wieviele Spalten vorhanden sind und ob der Inhalt einer Spalte links (`l`), zentriert (`c`) oder rechts (`r`) angeordnet wird
- ▶ Zudem wird spezifiziert wo vertikale Linien vorkommen - entweder eine einfache Linie (`|`) oder eine doppelte (`||`)
- ▶ Nachher können innerhalb der `tabular` Umgebung durch `&` die einzelnen Spalten getrennt werden und durch zwei Schrägstriche die Zeile umgebrochen werden
- ▶ Horizontale Linien muss der Autor durch den `hline`-Befehl selber machen

Eine komplexere Tabelle

Die Definition von Tabellen sieht komplex aus, lässt aber sehr flexible Tabellen zu:

eins und eins	gibt zwei
zwei und zwei	gibt vier
drei und drei	gibt sechs
vier und vier	gibt acht

Eine komplexere Tabelle

```
1 \begin{tabular}{|r|||}
2 \hline
3 eins und eins & gibt zwei \\
4 % \cline zieht wie \hline eine horizontale Linie
5 % Unterschied: Bei \cline kann angegeben werden, von
6 % welcher Spalte bis zu welcher Spalte die Linie
7 % laeuft. Hier also nur von Spalte 2 bis Spalte 2
8 zwei und zwei & gibt vier \\ \cline{2-2}
9 drei und drei & gibt sechs \\
10 \hline \hline
11 vier und vier & gibt acht \\
12 \hline
13 \end{tabular}
```

Beschriftung von Tabellen

Tabellen können auch beschriftet werden, wenn die `tabular` in eine `table` Umgebung verschachtelt wird:

eins und eins	gibt zwei
zwei und zwei	gibt vier
drei und drei	gibt sechs
vier und vier	gibt acht

Tabelle: Die Beschriftung der Tabelle

Beschriftung von Tabellen

```
1 \begin{table}
2
3   \begin{tabular}{|r||l|}
4     \hline
5     eins und eins & gibt zwei \\
6     zwei und zwei & gibt vier \\
7     drei und drei & gibt sechs \\
8     \hline \hline
9     vier und vier & gibt acht \\
10    \hline
11  \end{tabular}
12
13  \caption{Die Beschriftung der Tabelle}
14
15 \end{table}
```

Crash-Kurs: Einfügen von Bildern

- ▶ Wir benötigen das Package `graphicx`, um Bilder einfügen zu können

1

```
\usepackage{graphicx}
```

Bild einfügen



Bild einfügen

1

```
\includegraphics [ scale = 0.5 ] { src / kuzeb _ picture . jpg }
```


Der `includegraphics`-Befehl

- ▶ Der `includegraphics`-Befehl kommt aus dem `graphicx` Package, welches wir geladen haben
- ▶ Dem Befehl kann ein Pfad zu einer Bild-, PDF- oder Postscript-Datei angegeben werden, dieses Bild wird dann eingefügt
- ▶ Optional kann, wie im Beispiel, durch die Option `scale=0.5` erreicht werden, dass das Bild um 0.5 skaliert (d.H. mit halber Grösse) dargestellt wird

Wie bekomme ich noch mehr Hilfe?

- ▶ Generell hilft CTAN bei Fragen zu bestimmten Packages
<http://www.ctan.org/>
- ▶ Wenn wir dort nach `graphicx` suchen, finden wir das Package und seine Beschreibung, inklusive einem *Guide to graphics in LaTeX*
- ▶ Wenn CTAN nicht weiterhilft, kann eine Suche mit einer Suchmaschine im Internet weiterhelfen
- ▶ So könnte einfach nach *LaTeX itemize* gesucht werden, wenn man Hilfe mit der `itemize` Umgebung benötigt

Wie weiter?

- ▶ Auf der Computerlabor-Seite findet ihr die Slides sowie ein *Simple Document* und ein *Simple Document - mit Beispielen*
- ▶ Ihr seht in der Variante mit Beispielen die Formatierungen alle angewendet
- ▶ Wer aufmerksam ist, wird darin noch zwei Dinge finden, die in den Slides nicht vorkommen ☺
- ▶ Probiert aus, was euch interessiert
- ▶ Ihr könnt euch auch Links auf Slide 33 anschauen - bspw. um im Buch zu L^AT_EX nachzusehen, ob es neben `article` noch andere Document-Classes gibt
- ▶ Ihr könnt auch versuchen, ob ihr herausfindet, wie der Smiley ☺ in die Präsentation kommt . . .
... oder die Waschanweisung: 

Wie weiter?

- ▶ Wir haben nun grundlegende Einblicke in \LaTeX erhalten
- ▶ Das nächste Mal werden wir uns mit einer anderen Dokumentenklasse befassen: *beamer*
- ▶ Die Klasse ist eine Alternative zur *article* Klasse, die wir nun benutzt haben
- ▶ Formatierungen, Tabellen und Bilder können wir bei Beamer genau so benutzen wie wir es heute getan haben
- ▶ Deshalb können wir uns nächstes Mal hauptsächlich auf die Features und Besonderheiten von Beamer kümmern

Vielen Dank...

... für die Aufmerksamkeit.

Auflösung zu den Symbolen. . .

- ▶ Sucht mit einer Suchmaschine nach *latex symbol list*
- ▶ Ihr werdet *The Comprehensive LATEX Symbol List* finden
- ▶ Es listet die Packages, die spezielle Symbole bereitstellen

Auflösung zu den Symbolen...

```
1 % Der Smiley – das Package und das Kommando im
2 % Document drin
3 \usepackage{wasysym}
4 ...
5 \smiley
6
7 % Die Waschanleitung – das Package und das Kommando
8 % im Document drin
9 \usepackage{marvosym}
10 ...
11 \Handwash\ \NoBleech\ \NoIroning\ \NoTumbler\
```