

LaTeX-Praxis

# Einführung

Jörn Clausen  
joernc@gmail.com

# Übersicht

- Was ist  $\LaTeX$ , was ist  $\LaTeX$  nicht?
- Wie funktioniert  $\LaTeX$ ? Wieso gerade so?
- erste Schritte mit  $\LaTeX$
- Dokumente mit  $\LaTeX$  strukturieren

## Was ist $\text{\LaTeX}$ ?

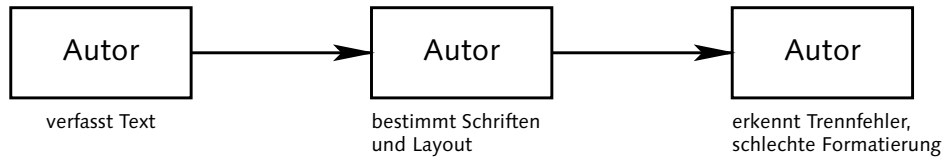
- $\text{\LaTeX}$  ...
  - ist keine Textverarbeitung, sondern ein Satzsystem
  - bietet kein WYSIWYG (what you see is what you get)
  - ist ziemlich alt
- Sind das Nachteile? Nicht wirklich:  $\text{\LaTeX}$  ...
  - liefert hochwertige Ergebnisse
  - wird von vielen Autoren und Verlagen verwendet
  - bietet Lösungen zu vielen Satz-Problemen
  - läuft auf fast jedem Computer

## Textverarbeitung – damals ...



- Tätigkeit von Lektor/Layouter und Setzer
  - nicht trivial
  - Handwerk
  - Kunst
  - braucht Erfahrung

## ... und heute



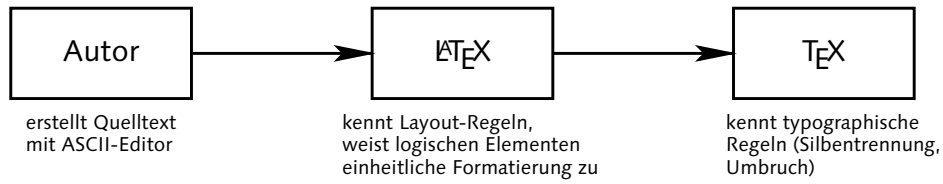
- klassische Textverarbeitungen überfordern den Benutzer
- noch schlimmer: DTP-Programme
- Beweis: *Glückwunschkarten* mit 20 **verschiedenen** SCHRIFTARTEN

## Revival eines Paradigmas

- Autor bestimmt Inhalt und logische Struktur des Textes
- „Experte“ legt Layout und Formatierung fest
- Satzsystem realisiert Ausgabe
- Dokumentklassen (Buch, Report, Brief, Kochrezept, . . .)
- Autor beschreibt Bedeutung, nicht Aussehen:

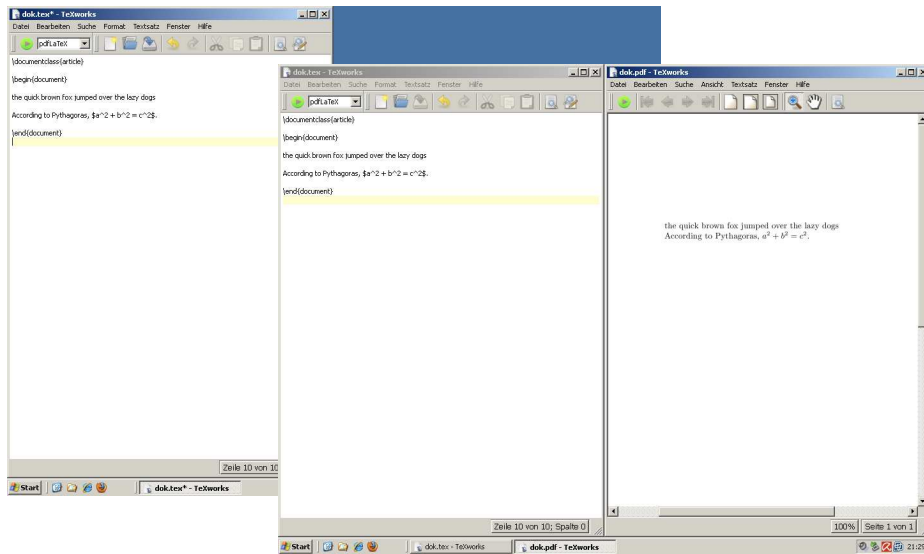
hervorheben	statt	kursiv
Überschrift	statt	15pt, fett, zentriert

## Schriftsatz mit $\LaTeX$



- $\TeX$  1977 von Donald Knuth entwickelt
- mathematischer Schriftsatz, gut lesbare Fonts
- $\LaTeX$  1985 von Leslie Lamport entwickelt
- „frontend“ für  $\TeX$  mit mächtigen Makro-Befehlen

# LaTeX im Fenster: TeXworks





# Aufgaben

- Eine minimale  $\LaTeX$ -Datei könnte so aussehen:

```
\documentclass{article}
\begin{document}
Hello world!
\end{document}
```

Speichere sie als „hello.tex“, übersetze sie und sieh Dir das Ergebnis an. Was fällt Dir auf?

- Was könnten die einzelnen Zeilen in der Datei bedeuten?

- Der Text "Hello world!" wird auf eine Seite gesetzt, mit etwas Rand nach oben und links. Ausserdem erscheint unten auf der Seite eine Seitenzahl.
- `\documentclass` bestimmt die Art des Dokuments. Dadurch wird die Formattierung des nachfolgenden Textes festgelegt. Neben "article" kennt  $\LaTeX$  noch eine Reihe weiterer Klassen, die wir später kennenlernen werden.
- `\begin{document}` und `\end{document}` grenzen den eigentlichen Inhalt des Dokuments ab. Zwischen diesen Anweisungen kann z.B. einfacher Text stehen, so wie in diesem Beispiel.

## Befehle

- Befehle werden mit einem backslash (\) eingeleitet

- einfacher Befehl:

```
\befehl1
```

- Befehl mit einem Pflicht-Argument:

```
\befehl2{argument}
```

- Befehl mit zwei Pflicht-Argumenten:

```
\befehl3{argument1}{argument2}
```

- Befehl mit optionalem Argument:

```
\befehl2[optargument]{argument}
```

# Aufgaben

- Ersetze die erste Zeile durch

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

und übersetze die Datei erneut. Was hat sich geändert?

- Ergänze die Hello-world-Zeile um

```
Hello world! It's \emph{so} great to use \LaTeX.
```

und übersetze das Dokument erneut.

- Welche Dateien entstehen bei der Übersetzung?

- Das Dokument wird auf eine Seite im DIN-A4-Format gedruckt. Gibt man die Papiergröße nicht explizit an, wird das amerikanische Format "letter" verwendet. Neben der Option `a4paper` kann man noch sinnvoll die Größe `a5paper` verwenden. Andere Papiergrößen der DIN-A-Serie kann man bei der Dokumentklasse "article" nicht angeben.
- Ausserdem wird das Dokument in einer etwas größeren Schrift gesetzt. Voreingestellt sind 10 Punkt, alternativ kann man noch 11 Punkt große Schrift auswählen. Die Angabe bezieht sich nur auf die Schrift für den Fließtext. Schriften in anderen Größen (z.B. Überschriften) werden entsprechend skaliert.
- Es entsteht eine Datei `hello.aux`, die allerdings keine interessanten Informationen enthält. Bei komplexeren Dokumenten können mit Hilfe dieser Datei Querverweise aufgelöst werden.
- Zusätzlich entsteht die Datei `hello.log`. Sie enthält eine Kopie der Ausgabe von `TeX`, die beim Übersetzen von `hello.tex` angezeigt wurde. Ausserdem enthält sie einige statistische Informationen, die bei der Suche nach Problemen hilfreich sein können.
- Bei der Verwendung von `TeXworks` entsteht außerdem die Datei `hello.synctex.gz`. Diese dient dazu, den Editor und den PDF-Previewer zu koordinieren.

## Fehler passieren

- mehr oder weniger hilfreiche Fehlermeldungen
- nicht immer leicht zu verstehen, aber ...
- **LESEN HILFT OFT WEITER!!!**

```
! Undefined control sequence.  
1.6 Hello world! It's \emph  
      {so} great to use \LaTeX.
```

- Übersetzung mit „x“ beenden
- Zeilennummer beachten
- Fehler direkt vor „Zeilenwechsel“

# Aufgaben

- Füge einen längeren Text in die Datei ein.  
Falls Du nicht weißt, was Du schreiben sollst, kannst Du mit dem „Lipsum generator“ beliebig viel lateinisch anmutenden Text erzeugen:  
<http://www.lipsum.com/>
- Welche Bedeutung haben Zeilenumbrüche im Quelltext?
- Welche Bedeutung haben Leerzeilen im Quelltext?

- Zeilenumbrüche im Quelltext werden als einfache Leerzeichen interpretiert. Auch mehrere aufeinanderfolgende Leerzeichen oder Tabulatoren werden als ein Leerzeichen gesetzt. Vor allem sorgt `\TeX` (bzw. `TeX`) automatisch für eine Silbentrennung (die allerdings auf englischen Text eingestellt ist und daher bei dem pseudo-lateinischen Text eventuell zu ungewöhnlichen Ergebnissen führen kann).
- Eine oder mehrere aufeinanderfolgende Leerzeilen werden üblicherweise zur Trennung von Absätzen verwendet. In der Dokumentklasse „article“ werden Absätze ohne Abstand gesetzt, dafür aber durch einen Einzug der ersten Zeile gekennzeichnet.

## Klassen und Stile

- die Dokumentklasse bestimmt Art und damit Aussehen (Layout) des Dokuments:

- kurzer Artikel

- ```
\documentclass{article}
```

- Vortragsfolien

- ```
\documentclass{slides}
```

- Stile bestimmen Art und Aussehen einzelner Bestandteile oder Aspekte des Dokuments
- Stile werden in der Präambel der  $\LaTeX$ -Datei eingebunden

```
\documentclass[a4paper]{article}
\usepackage{color}
\usepackage{chess}
```

- Befehle werden in Dokumentklassen und Stilen definiert

## Aufgaben

- Füge die folgende Zeile in der Präambel von `hello.tex` ein:

```
\usepackage{palatino}
```

Was ändert sich?

- Binde das Paket „`pifont`“ ein und füge diese beiden Zeilen zum Dokument hinzu:

```
\ding{41} joernc (at) gmail (dot) com
```

```
\ding{37} 0521/106-12601
```

- Binde das Paket „`createsudoku`“ ein und verwende den Befehl `\generategrid`

- Es wird die Schriftart „Palatino“ (oder eine sehr ähnlich aussehende Kopie) verwendet. Diese Änderung betrifft auch alle Varianten, z.B. das kursiv gesetzte Wort „so“.
- Mit dem Befehl `\ding` können Symbole aus der Schriftart „Zapf Dingbats“ eingebunden werden. Das Symbol mit der Nummer 41 ist ein Briefumschlag, das an der Stelle 37 ein Telefon.
- Das Paket „`createsudoku`“ berechnet ein (lösbares) Sudoku-Rätsel.

# Umlaute und Anführungszeichen

```
\documentclass[12pt,a4paper]{article}
\usepackage[german]{babel}
\usepackage[utf8]{inputenc}

\begin{document}

Umlaute: \"a \"o \"u \"A \"O \"U \ss{}
Akzente: \'a \'a ^a \c{C} \o{} \~n
Anf\"uhrungszeichen: ``Ha!'' `Hu?'

mit babel: "a "o "u "A "O "U "s "`He!""
mit inputenc: ä ö ü Ä Ö Ü ß á à â ø ñ

\end{document}
```



## Aufgaben

- Füge die Anweisung

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

in die  $\LaTeX$ -Datei ein. Wie ändert sich die Formatierung?

- Füge Umlaute, Akzente und Anführungszeichen in den Text ein.

- Eventuell ändert sich die Silbentrennung einzelner Wörter, da jetzt deutsche Trennregeln verwendet werden.

# Dokumentenstruktur

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

\begin{document}
\section{Einleitung}
  \subsection{Motivation}
    aaa bbb ccc ddd eee
  \subsection{Danksagungen}
    aaa bbb ccc ddd eee
\section{Hauptteil}
  \subsection{Andere Verfahren}
    aaa bbb ccc ddd eee
  \subsection{Unsere L\osung}
    aaa bbb ccc ddd eee
    \subsubsection{Ansatz}
    \subsubsection{Probleme}
\end{document}
```

## Aufgaben

- Strukturiere das Dokument mit Hilfe der Befehle `\section`, `\subsection` und `\subsubsection`. Was passiert?
- Füge am Anfang des Dokuments die Anweisung

```
\tableofcontents
```

ein. Was passiert beim Übersetzen? Welche Dateien entstehen?

- Übersetze die  $\LaTeX$ -Datei ein zweites Mal. Was hat sich verändert? Erkläre Deine Beobachtungen.

- Durch die Befehle werden Überschriften erzeugt. Diese Überschriften werden entsprechend ihrer Hierarchie-Stufe nummeriert. Eine neue section setzt den Zähler für die subsections zurück.
- Nachdem das Dokument einmal übersetzt wurde, ist in der gesetzten Datei das Wort "Inhaltsverzeichnis" zu sehen, allerdings fehlt das eigentliche Inhaltsverzeichnis selbst. Es ist eine Datei `h110.toc` entstanden, die die Überschriften der einzelnen Abschnitte enthält.
- Nachdem man das Dokument nochmal übersetzt hat, enthält das Inhaltsverzeichnis tatsächlich eine Übersicht über die sections, subsections und subsections.  $\LaTeX$  bindet beim Übersetzen die Datei `h110.toc` aus dem vorhergehenden Durchlauf ein. Daher "hinkt" das Inhaltsverzeichnis immer einen Übersetzungs-lauf hinter dem Dokument hinterher. Wenn man also einen neuen Abschnitt eingefügt hat, muss man  $\LaTeX$  zweimal aufrufen, um ein korrektes Dokument zu erhalten.

# Aufgaben

- Füge vor dem Inhaltsverzeichnis folgende Zeilen ein:

```
\title{Hello World}  
\author{Joe User}  
\maketitle
```

Welche Informationen erscheinen jetzt im gesetzten Dokument?

- Lösche die aux- und die toc-Datei, und kommentiere die Zeile mit dem Babel-Paket aus:

```
%\usepackage[german]{babel}
```

Was ändert sich?

- Der Titel und der Autor werden als Überschrift gesetzt. Zusätzlich wird das aktuelle Datum ausgegeben. Mehrere Autoren können durch die Anweisung `\and` getrennt werden:  

```
\author{Jane Doe \and Joe User}
```

Das Datum kann auch durch den `\date`-Befehl angegeben werden:  

```
\date{24.\ Dezember 2003}
```
  - Der backslash vor dem Leerzeichen verhindert, dass der Punkt als Satzende interpretiert wird und der Abstand zum nachfolgenden "D" zu groß gesetzt wird.
  - Statt "Inhaltsverzeichnis" erscheint der Ausdruck "Contents", und das Datum des Titels wird in einem anderen Format gesetzt. Dies funktioniert allerdings nur, wenn das Datum von `\LaTeX` erzeugt wird, nicht wenn es durch den `\date`-Befehl vorgegeben ist.
- Die Sprache bestimmt also nicht nur die Trennregeln, sondern vor allem vorgefertige Textstücke, wie "Abbildung" bzw. "Figure", "Literaturverzeichnis" bzw. "Bibliography", usw.

## Dokumentklassen

- original  $\LaTeX$ : `article`, `report`, `book`
- KOMA-Skript: `scrartcl`, `scrreprt`, `scrbook`
- Gliederungsstufen:

```
article          \section \subsection ...
report  \chapter \section \subsection ...
book     \chapter \section \subsection ...
```

- `...`, `\subsubsection`, `\paragraph`, `\subparagraph`
- zusätzlich `\part`
- mehrere `article` können ein `book` bilden

## Aufgaben

- Ersetze die Dokumentklasse `article` durch `scrartcl`. Wie ändert sich die Formatierung?
- Definiere mit der Anweisung `\chapter` ein oder mehrere Kapitel und probiere die Dokumentklassen `report`, `scrreprt`, `book` und `scrbook` aus.

- Die KOMA-Skript-Klassen definieren die Ränder anders (vor allem angenehmer) als die Standard-Klassen von  $\LaTeX$ . Ausserdem werden die Überschriften in einer anderen Schrift gesetzt.
- In den Klassen `report` und `book` wird der Titel auf eine eigene Seite gesetzt. Die Buch-Klassen formatieren den Text für doppelseitigen Druck, d.h. die Seitenzahlen werden am linken oder rechten Rand gesetzt. Ausserdem werden bei Bedarf leere Seiten eingefügt.

# Aufgaben

- Erweitere die  $\LaTeX$ -Datei folgendermaßen:

```
\maketitle
\frontmatter
\tableofcontents
\mainmatter
\chapter{...}
\section{...}
\chapter{...}
\section{...}
\appendix
\chapter{...}
\chapter{...}
```

Welchen Effekt haben die zusätzlichen Anweisungen?

- Der Bereich zwischen `\frontmatter` und `\mainmatter` wird, mit Seite eins beginnend, mit römischen Seitenzahlen versehen. Durch `\mainmatter` wird der Seitenzähler zurückgesetzt, und die Seiten werden arabisch nummeriert. Nach `\appendix` werden die Kapitel nicht mit Nummern sondern mit Buchstaben versehen. Zusätzlich gibt es noch den Befehl `\backmatter`, mit dem der Bereich nach den Anhängen abgetrennt werden kann. Kapitel erhalten dann überhaupt keine Kennzeichnung mehr. Dieser Bereich kann für Nachworte, ein Kolophonium oder das Literaturverzeichnis verwendet werden.