



# L<sup>A</sup>T<sub>E</sub>X Einführung

Vincent

Labor Bochum e.V.

29. Juni 2009



# Motivation

- Einstieg erleichtern
- Euch etwas Gutes tun
- Dem Labor etwas Gutes tun
- Selber etwas dabei lernen
- Versuch: Häufige Probleme erkennen und lösen

# Vorgehen

- Teil 1: Einführung
- Teil 2: Basiswerkzeug
- Teil 3: Makroprogrammierung
- Teil 4: Präsentationen mit  $\text{\LaTeX}$

# Ablauf

- Vortrag
- Pause (inkl. Pizzabestellung)
- Praxis (*freiwillig* inkl. Aufgabenstellung)

## 1 Teil 1 – Einführung

- Geschichte und Hello World
- Zeichensätze und Dateikodierungen
- Minimale Standardvorlage
- Warum soll ich mir das antun?
- Einsteigerliteratur und Probleme, Ausblick

# Geschichte zu $\LaTeX$

- $\tau$  („tau“)  $\epsilon$  („epsilon“)  $\chi$  („chi“) =  $\TeX$
- Textsatzsystem mit eingebauter Makrosprache, basierend auf *primitives*
- Ab 1977 von Donald E. Knuth entwickelt; 1986 fertig gestellt
- Befehle werden durch einen Interpreten übersetzt
- Makro = Folge von Befehlen, die automatisch bei Aufruf ablaufen
- $\LaTeX$  = Makro-Sammlung zur einfacheren Benutzung von  $\TeX$
- Entwickelt von Leslie **L**amport.  $\text{La} + \text{\TeX} = \text{\LaTeX}$  (ab 1980)
- Aktuelle Versionen:  $\TeX$ : 3.1415926,  $\LaTeX 2_{\epsilon}$
- **Ziel:** „Schöne“ Bücher, Artikel, etc.

# „Hello World!“ in T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X

```
1 Hello World!  
2 \bye
```

Listing 1: T<sub>E</sub>X

```
1 \documentclass{article}  
2 \begin{document}  
3 Hello World!  
4 \end{document}
```

Listing 2: L<sup>A</sup>T<sub>E</sub>X

- Quellcode wird in Textdatei geschrieben → HelloWorld.tex
- Erklärung zu documentclass kommt noch!
- Logische Kennzeichnung von Textstellen nennt man „Markup“
- Eventuell bekannt aus anderen Sprachen, zum Beispiel: HTML

# Theorie: Arbeitsschritte bei $\text{\LaTeX}$

- Es gibt 2 Möglichkeiten ein  $\text{\LaTeX}$ -Dokument zu erstellen:
  - ▶ Früher: `latex + dvips/(dvipdf)` (`.tex`→`.dvi`→`.ps/(.pdf)`)
  - ▶ Seit 2000: `pdflatex` (`.tex`→`.pdf`) ★ Empfehlung ★
- Vereinfacht:
  - ▶ PS (1984)/PDF (1993) sind Seitenbeschreibungssprachen von Adobe
  - ▶ Ziel: Originalgetreue Wiedergabe von Dokumenten unabhängig von Anwendung, Betriebssystem, Hardwareplattform, etc.
  - ▶ DVI (1979) ähnlich, jedoch werden Zeichensätze nicht in Datei abgelegt!
- PostScript wird immer mehr von PDF verdrängt
  - ▶ PDF 1.7 seit 2008 ISO-Standard und damit ein offener Standard!
- DVI außerhalb von  $\text{\LaTeX}$  eher unüblich ...
  - ▶ `dvips/dvipdf` sind „Druckausgabetreiber“
  - ▶ `xdvi` für Bildschirmausgabe



# Vorführung

Hello World mit: T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X (beide Varianten)

# Zusammenfassung 1: Hello World

- Quellcode ist eine „einfache“ Textdatei mit der Endung `.tex`
- Beim Übersetzen entstehen weitere Dateien, hier (vorerst nur):
  - ▶ `.log`: Logdatei gibt Auskunft über den Übersetzungsprozess
  - ▶ `.aux`: Dient zur Zwischenspeicherung von Informationen
- Struktur eines Befehls: `\name[optional]{Pflicht}` bzw. `\befehl`
- Spezielle Zeichen: `# $ & ~ _ ^ % { }`, Darstellung z. B. durch `\$`
- Neue Zeilen: `\\` bzw. `\newline`
- Neue Seiten (erzwingen): `\newpage`
- Backslash durch `\textbackslash`

# Zusammenfassung 1: Hello World

```
1 \documentclass[options]{class}
2 %Further global commands and specifications, called preamble
3 %(Präambel)
4 \begin{document}
5 %Text mixed with commands of local effect, called body
6 %(Dokumentkörper)
7 \end{document}
```

Listing 3: Dokumentstruktur von L<sup>A</sup>T<sub>E</sub>X

# Vorführung

Hello World mit Umlauten, Einführung von Paketen

## Zusammenfassung 2: Hello World

- Probleme bei Eingabe und Speicherung von Sonderzeichen und Umlauten im Quelltext
- **Keine** Probleme damit beim fertigen (gedruckten) Dokument!
- $\LaTeX$  kann durch Pakete erweitert werden
- Zusammenfassung/ Verteilung von Paketen nennt man Distribution
  - ▶ TeXLive (Unix/Mac/Windows)
  - ▶ MikTeX (Windows)
- Freie Editorwahl, jedoch bieten  $\LaTeX$ -Editoren sinnvolle Funktionen!
  - ▶ Windows: WinShell, TeXnicCenter.
  - ▶ Linux/Unix: Kile, Plugin für: Vim, gedit, Emacs.
  - ▶ Mac: TeXShop, iTeXMac.
  - ▶ Multi-Plattform: Eclipse + TeXlipse, TeXmaker, (LyX).
- Einzelne Pakete: CTAN.org (*Comprehensive T<sub>E</sub>X Archive Network*)

# Vergleich von latex mit pdflatex

- Annahme: latex  $\rightarrow$  .ps, pdflatex  $\rightarrow$  .pdf!
  - ▶ Unterschiede (auch) durch unterschiedliches Dateiformat begründet!
- Vorteile latex:
  - ▶ PostScript basierte Pakete funktionieren
- Nachteile latex:
  - ▶ Nur .eps Graphiken möglich (**kein** .jpg oder .png!)
- PS: Keine dynamischen Inhalte, (keine Transparenzobjekte)

# Vergleich von latex mit pdflatex

- Vorteile pdf<sub>l</sub>atex:
  - ▶ Direkte Einbindung von Graphikformaten: .jpg, .png, .pdf ...
  - ▶ Direkter Zugriff auf PDF-Features: Hyperlinks, Dokumenteninformationen, Formulare
- Nachteile pdf<sub>l</sub>atex:
  - ▶ Einbinden von .eps nicht direkt möglich (epstopdf)
  - ▶ Spezielle Pakete funktionieren nicht (bspw. pstricks)
- PDF: Dynamische Inhalte, Transparenzobjekte

## 1 Teil 1 – Einführung

- Geschichte und Hello World
- **Zeichensätze und Dateikodierungen**
- Minimale Standardvorlage
- Warum soll ich mir das antun?
- Einsteigerliteratur und Probleme, Ausblick



# Zeichensätze Teil 1

- Irgendwo muss festgelegt sein, wie Zeichen abgespeichert werden.
- Problem: Es existieren unterschiedliche Zeichensätze:
  - ▶ ANSI X3.4-1986 (US-ASCII) – 128 Zeichen
  - ▶ ISO 8859-1 (Latin-1) und ISO 8859-9 (Latin-9) – 191 Zeichen
  - ▶ Windows-1252 Schnittmenge aus Latin-1 und -9
  - ▶ Codepage 437 (DOS-US) - 256 Zeichen
  - ▶ MacRoman – 223 Zeichen
  - ▶ Unicode 2.0 – 65536 Zeichen
- Zusätzlich: Zeilenenden werden unterschiedlich kodiert – ist  $\text{\LaTeX}$  egal!
  - ▶ Windows: CR LF
  - ▶ Commodore, Apple Mac OS (bis Version 9): CR
  - ▶ Unix-Derivate: LF
  - ▶ *CarriageReturn* – Sprung der Schreibmarke an Zeilenanfang
  - ▶ *LineFeed* – beschreibt das Drehen der Walze
- Beispiel ASCII: Keine Umlaute, daher auch keine direkte Umlauteingabe möglich!  $\text{\LaTeX}$ -Lösung: `\" {a}=ä, \ss{}=ß`

## Zeichensätze Teil 2

- Vorteil der  $\LaTeX$ -Lösung: Portabilität!
- Nachteil der  $\LaTeX$ -Lösung: Umständliche Eingabe, keine Rechtschreibüberprüfung – was ist mit noch spezielleren Zeichen?!
- Lösung:  $\LaTeX$ -Interpret anderen Zeichensatz mitteilen:
  - ▶ Windows: `\usepackage[ansinew]{inputenc}`
  - ▶ Unix/Linux: `\usepackage[latin1]{inputenc}` oder `[latin9]`
  - ▶ Mac OS X: `\usepackage[applemac]{inputenc}`
  - ▶ Multi-Plattform: `\usepackage[utf8]{inputenc}`
- UTF-8 ist der Unicode Zeichensatz der auf allen (modernen) Plattformen verfügbar sein sollte
- Sonst: Eingeschränkte Portabilität des Quellcodes!
- Empfehlung: Vorlagen mit `\"{}` , Text mit direkter Eingabe.

- $\text{\LaTeX}$  interpretiert nur Text, dieser muss von einem Editor entsprechend abgespeichert werden!
- Editoren verwenden zum Abspeichern einen bestimmten Zeichensatz, dieser kann nicht bei jedem Editor eingestellt werden!
- Abhilfe: Gute Editoren verwenden, die das können! ;-)
- Vom Editor verwendeter Zeichensatz muss mit dem im Quellcode angegebenen Zeichensatz übereinstimmen!
- Editoren verwenden, die alle Zeilenenden interpretieren können!

## 1 Teil 1 – Einführung

- Geschichte und Hello World
- Zeichensätze und Dateikodierungen
- **Minimale Standardvorlage**
- Warum soll ich mir das antun?
- Einsteigerliteratur und Probleme, Ausblick

# Minimale Standardvorlage

- Fehlt noch: Trennregeln für Wörter und unterschiedliche Sprachen!
  - ▶ `\usepackage[ngerman]{babel}`
- Reihenfolge bei Einbindung *kann* wichtig sein!

```
1 \documentclass[a4paper]{article}%Papiergröße=A4
2 \usepackage[utf8]{inputenc}%Eingabekodierung, zuerst laden!
3 \usepackage[T1]{fontenc}%Ausgabekodierung
4 \usepackage[ngerman]{babel}%Trennregeln
5 \usepackage{lmodern}%Verbesserter Standardfont
6 \usepackage{microtype}%Verbesserter optischer Randausgleich
7 \usepackage{hyperref}%Sprungmarken, immer zuletzt laden!
8 \begin{document}
9 Jetzt auch: Hällo Wörld ;- )
10 \end{document}
```

Listing 4: Standardvorlage

## Anmerkung zu Fonts bzw. `lmodern`

- $\text{\LaTeX}$  verwendet normalerweise die *Computer Modern*-Schriften (`cm`)
- `cm`-Fonts sind Bitmaps! („verpixelt“)
- *Latin Modern* (`lmodern`) ist besser! (PostScript Level 1 Font)
- Alternativ: `cm-super` Fontpaket nutzen (PostScript Level 1 Font)
  - ▶ Starker Unterschied beim  $\beta$  im Vergleich zu `lmodern`

# Weitere Fehlerursachen für (optisch) schlechte Ergebnisse

- Standard: `cm` in Old-TeX-Kodierung (OT1)
- Keine Berücksichtigung von Umlauten!
- Daher unbedingt `[T1]{fontenc}` für korrekte Ausgabekodierung und korrekte Trennung dieser Wörter verwenden!
- $\LaTeX$  schaltet dazu zur „europäischen“ CM-Variante EC (*Extended Computer*)
- Qualität jedoch nicht so gut wie bei der CM-Variante
- Keine Postscript Level 3 Fonts benutzen!

Was passieren kann ...

...wenn man solche Dinge nicht beachtet:

Netzsich  
Netzsich



# Dokumentenklassen

- $\text{\LaTeX}$  für amerikanische Layoutvorstellungen ausgelegt.
  - ▶ Anpassung zum Beispiel an das Papierformat Letter (statt DIN A4).
  - ▶ Grundklassen sind: `book`, `report`, `article`, `letter`.
  - ▶ Unterschiede: Layout (Textanordnung, einseitig, zweiseitig), Organisation und Unterteilungsmöglichkeiten, etc.
- Europa: KOMA-Script für (deutsches) Textlayout:
  - ▶ Grundklassen sind: `scrbook`, `scrreprt`, `scrartcl`, `scrllttr2`
- Mögliche Optionen zum Beispiel: `[a4paper]`, `[a5paper, twoside]`
  - ▶ `\documentclass{article}` – Papierangabe fehlt!
  - ▶ `\documentclass[a4paper]{article}` – OK!
  - ▶ `\documentclass[a4paper]{scrartcl}` – Noch besser!
- $\text{\LaTeX}$ -Präsentationen nutzen als Dokumentenklasse: `beamer`

## 1 Teil 1 – Einführung

- Geschichte und Hello World
- Zeichensätze und Dateikodierungen
- Minimale Standardvorlage
- **Warum soll ich mir das antun?**
- Einsteigerliteratur und Probleme, Ausblick

# Was bietet L<sup>A</sup>T<sub>E</sub>X?

- Viele gute Sachen:
  - ▶ Einmalig guter Formelsatz!
  - ▶ Kostenlos, lange Entwicklungszyklen, ausgereift
  - ▶ Gute Quellenverwaltung
  - ▶ Fast beliebig erweiterbar – „lernt“
  - ▶ Im Code schreiben = schneller, effizienter
  - ▶ Autor bleibt Autor, Layout-Designer ist L<sup>A</sup>T<sub>E</sub>X, Setzer ist T<sub>E</sub>X
- Natürlich auch:
  - ▶ Kerning, Ligaturen, optischer Randausgleich
  - ▶ Regeln um Schusterjungen und Hurenkinder zu vermeiden
- Vorsicht: `\{a}=ä` ist nicht `{\a}=ä` – kein Kerning!

• Vergleiche: **VÄTER** und **VÄTER**

• Vergleiche: **häufig** und **häufig**

# Vergleich L<sup>A</sup>T<sub>E</sub>X vs MS Formeleditor

- L<sup>A</sup>T<sub>E</sub>X:

$$\sum_{i \in V} \frac{\alpha}{\beta} \stackrel{?}{\equiv} \int \sqrt[3]{4} \stackrel{!}{\approx} \underbrace{3^3 + 4 \cdot 7}_{109} \stackrel{*}{\neq} \binom{3}{4} \propto \vec{\zeta} \pmod{n} \quad (1)$$

$$\lim_{x \rightarrow \infty} \left( \sum_{\substack{0 \leq i \in V \\ 0 < j < n}}^{\infty+} \right) \stackrel{(1.12)}{=} \int_E P(x, A) \pi(x) \mu(dx) \quad (2)$$

$$\stackrel{(1.14)}{=} \int_A \left[ \int_E p(y, x) \pi(y) \mu(dx) \right] \mu(dy) \quad (3)$$

- Office: Ich lasse mir gerne zeigen, wie es besser geht ...

...wird also nicht passieren! :-)

$$\sum_{i \in V} \frac{\alpha}{\beta} \stackrel{?}{\equiv} \int \sqrt[3]{4} \stackrel{!}{\approx} \underbrace{3^3 + 4 \cdot 7}_{109} \stackrel{*}{\neq} \binom{3}{4} \propto \vec{\zeta}$$

# Vorführung

# L<sup>A</sup>T<sub>E</sub>X hat aber auch Nachteile

- Kein „What you see is what you get“ (WYSIWYG)
- Extrem hoher Lernaufwand am Anfang
- „Einfache“ Wünsche oft nur mit viel Wissen umsetzbar
- Etwas Inkonsistenz durch unterschiedliche Paketversionen
- Kein Support durch einen Hersteller
- Definitiv keine Tabellenkalkulation

## 1 Teil 1 – Einführung

- Geschichte und Hello World
- Zeichensätze und Dateikodierungen
- Minimale Standardvorlage
- Warum soll ich mir das antun?
- Einsteigerliteratur und Probleme, Ausblick

# Literaturhinweise für Einsteiger

- Problem sehr vieler Bücher: Veraltete Befehle und/oder Pakete!
  - ▶ Spart euch das Geld!
- Gute Einsteigerliteratur im Internet:
  - ▶ Manuela Jürgens: „ $\LaTeX$  – eine Einführung und ein bisschen mehr ...“
  - ▶ Mark Trettin: „ $\LaTeX$ -Sündenregister“
- Ansonsten:
  - ▶ Forum: [www.mrunix.de](http://www.mrunix.de) (deutsch)
  - ▶ Newsgroup: [comp.text.tex](mailto:comp.text.tex) (englisch) und [de.comp.text.tex](mailto:de.comp.text.tex) (deutsch)
  - ▶  $\TeX$ -Usergroup: DANTE e.V.
- Falls doch ein Buch gekauft wird:
  - ▶ Frank Mittelbach: „Der  $\LaTeX$ -Begleiter“
  - ▶ Mit Vorsicht: Helmut Kopka: „LaTeX Band 1“
- Google: Hilft auch!



# Vorführung

Beispiele zeigen, TeXShop vorstellen

## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse
- Kopf-/Fußzeile, Bilder, Tabellen
- Formelsatz
- Fehlerbehandlung
- Fragen vom letzten Termin
- Aufgaben

# Vorab: Trennung zwischen T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X-Befehlen

- Der L<sup>A</sup>T<sub>E</sub>X-Interpreter versteht auch T<sub>E</sub>X-Befehle
  - ▶ Eine Trennung zwischen den Befehlen ist daher schwierig!
- *Meistens* funktioniert es trotzdem!

# Vorab: Trennung zwischen T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X-Befehlen

- Der L<sup>A</sup>T<sub>E</sub>X-Interpret versteht auch T<sub>E</sub>X-Befehle
  - ▶ Eine Trennung zwischen den Befehlen ist daher schwierig!
- *Meistens* funktioniert es trotzdem!
  - ▶ **Aber nicht immer!**

# Schriftstil ändern

Schriftfamilie: roman, sans serif, typewriter

Schriftstärke: **bold font**, medium

Schriftform: aufrecht, *italic*, *slanted*, SMALL CAPITALS

Mehr Beispiele:

*betont*, ***verschachtelt***

```
1 Schriftfamilie: \textrm{roman}, \textsf{sans serif}, \texttt{
   typewriter}\\
2 Schriftstärke: \textbf{bold font}, \textmd{medium}\\
3 Schriftform: \textup{aufrecht}, \textit{italic}, \textsl{
   slanted}, \textsc{small capitals}\\
4 Mehr Beispiele: \\
5 \emph{betont}, \textbf{\textit{verschachtelt}}
```

# Schriftstil ändern, so jedoch nicht!

so sollte man es **nicht** machen  
so sollte man es eigentlich auch **nicht** machen

```
1 so sollte man es {\bf nicht} machen\\  
2 so sollte man es eigentlich auch {\bfseries nicht} machen
```

Warum nicht? → „ $\text{\LaTeX}$ -Sündenregister“

# Schriftgröße

winzig, sehr klein

Fußnote, klein

normal, groß

größer, noch größer

riesig, gigantisch

Als Umgebung: andere Möglichkeit

```
1 {\tiny winzig}, {\scriptsize sehr klein}\  
2 {\footnotesize Fußnote}, {\small klein}\  
3 {\normalsize normal}, {\large groß}\  
4 {\Large größer}, {\LARGE noch größer}\  
5 {\huge riesig}, {\Huge gigantisch}\  
6 Als Umgebung: \begin{small} andere Möglichkeit \end{small}
```

Zusätzlich Basisgröße als *class option* angeben: 10pt, 11pt, 12pt

# Aufzählungen mit Punkten

Normaler Text ohne Inhalt.

- Eine
  - kleine
- ! Aufzählung :-]

Text wird hier fortgesetzt.

```
1 Normaler Text ohne Inhalt.  
2 \begin{itemize}  
3 \item Eine  
4 \item kleine  
5 \item[!] Aufzählung :-]  
6 \end{itemize}  
7 Text wird hier fortgesetzt.
```

Unterpunkte mit `\subitem` oder `\subsubitem`



# Aufzählungen mit Nummern

Normaler Text ohne Inhalt.

- 1 Eine
- 2 kleine
- 3 Aufzählung :-]

Text wird hier fortgesetzt.

```
1 Normaler Text ohne Inhalt.  
2 \begin{enumerate}  
3 \item Eine  
4 \item kleine  
5 \item Aufzählung :-]  
6 \end{enumerate}  
7 Text wird hier fortgesetzt.
```

Unterpunkte mit `\subitem` oder `\subsubitem`

## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- **Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse**
- Kopf-/Fußzeile, Bilder, Tabellen
- Formelsatz
- Fehlerbehandlung
- Fragen vom letzten Termin
- Aufgaben

# Anführungszeichen

“Englisch” „Deutsch“ „anderes Deutsch“

```
1 ‘ ‘Englisch’ ’ ’ “Deutsch” ’ \glqq{}anderes Deutsch\grqq
```

Geht aber viel schöner:

„Hier ein ganzer Satz in Anführungszeichen“

```
1 %\usepackage[ngerman]{babel}  
2 %\usepackage[babel, german=quotes]{csquotes}  
3 \enquote{Hier ein ganzer Satz in Anführungszeichen}
```

# Anführungszeichen

„Deutsch“ “Englisch” « Französisch »

```
1 %\usepackage[english, french, ngerman]{babel}
2 %\usepackage[babel,english=american, french=guillemets, german=
   quotes]{csquotes}
3 {\selectlanguage{ngerman} \enquote{Deutsch}}
4 {\selectlanguage{english} \enquote{Englisch}}
5 {\selectlanguage{french} \enquote{Französisch}}
```

# Was jetzt?

- Feedback? → LaborWiki!
- Pizzabestellung?
- Mit Pizza:  $\text{\LaTeX}$  installieren und HelloWorld erstellen

# Strukturieren im Text

```
1 \chapter{Kapitel} % nur mit der Klasse book
2 \part{Teilstück} % selten benötigt
3 \section{Abschnitt}
4 Beispieltext \dots
5 \subsection{Unterabschnitt}
6 Beispieltext \dots
7 \subsubsection{Unter-Unterabschnitt}
8 Beispieltext \dots
```

Listing 5: Gliederungsmöglichkeiten

# Strukturieren im Quellcode

```
1 \include{dateiname} % erzeugt \newpage, bindet Quellcode ein
2 \input{dateiname} % ohne \newpage
```

Listing 6: Einbinden von Quellcode

Besonderheiten:

- Teildokumente ohne Präambel
- Ohne `\begin{document}` bzw. `\end{document}`
- `\include` erzeugt pro Teildokument eine `.aux`-Datei
- `\includeonly` erlaubt nur bestimmte Teile für Übersetzung
- → Bearbeitung einzelner Kapitel und trotzdem passende Seitenzahlen

Aufpassen bei Pfadangaben (z. B. zu Bildern) in den Teildokumenten!

# Titelseite und Inhaltsverzeichnis

```
1 \documentclass{article}
2 \author{Labor e.V.}
3 \title{LaTeX-Workshop}
4 \begin{document}
5 \maketitle
6 \tableofcontents
7 \section{Einleitung}
8 \section{Hauptteil}
9 \section{Schluss}
```

Mehrfaches Übersetzen des Quellcodes nötig (2-3×!)



## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse
- **Kopf-/Fußzeile, Bilder, Tabellen**
- Formelsatz
- Fehlerbehandlung
- Fragen vom letzten Termin
- Aufgaben

# Kopf-/Fußzeile

- Dafür gibt es 2 Pakete:
  - ▶ fancyhdr
  - ▶ scrpage2
- Welches Paket ist besser? – Falsche Frage!
  - ▶ fancyhdr an article angepasst
  - ▶ scrpage2 an scrartcl angepasst

# Kopf-/Fußzeile

```
1 \documentclass[a4paper]{scrartcl}
2 \usepackage{scrpage2}
3 \usepackage{lastpage}
4 \ihead{oben links}
5 \chead{oben mittig}
6 \rohead{oben rechts}
7 \ifoot{unten links}
8 \cfoot{\thepage/\pageref{LastPage}} % Seite / Seitenanzahl
9 \rofoot{unten rechts}
10
11 \setheadsepline{0.5pt} % Linie zur Begrenzung
12 \setfootsepline{0.5pt} % Linie zur Begrenzung
13
14 \pagestyle{scrheadings} % gemachte Einstellungen anwenden
15 \begin{document}
16 Hier geht es los!
17 \end{document}
```

Listing 7: Beispiel scrpage2

## Bilder einbinden



Abbildung: Mit passender Beschreibung

# Bilder einbinden

```
1 %\usepackage{graphicx}
2 %\usepackage{epstopdf}
3 %Reihenfolge ist wichtig!
4 \begin{figure}[htbp]
5 \centering
6 \includegraphics[scale=0.3]{panda2}
7 \caption{Mit passender Beschreibung}
8 \end{figure}
```

Listing 8: Bilder einbinden

# Bilderpositionierung

Oder warum erscheint mein Bild nicht wo es soll?

- Bild befindet sich in einer „gleitenden“ Umgebung
  - ▶  $\LaTeX$  findet den optimalen Platz ...
  - ▶ ...meistens zumindestens!
- Abhilfe:

```
1 %\usepackage{placeins} % Möglichkeit 2
2 \begin{figure}[!htb] % Möglichkeit 1
3 \centering
4 \includegraphics[scale=0.3]{panda2}
5 \caption{Mit passender Beschreibung}
6 \end{figure}
7 \FloatBarrier % Möglichkeit 2
```

Listing 9: Bilder fest positionieren

# Bilderpositionierung

Oder warum erscheint mein Bild nicht wo es soll?

```
1 %\usepackage{float} % Möglichkeit 3
2 \begin{figure}[H] % Möglichkeit 3
3 \centering
4 \includegraphics[scale=0.3]{panda2}
5 \caption{Mit passender Beschreibung}
6 \end{figure}
```

Listing 10: Bilder fest positionieren

# Tabellensatz

- t – Positionierung
- c – center text, | – senkrechte Linien
- & Spaltenbegrenzer, \\ Zeilenbegrenzer
- \hline horizontale Linie

1	2
3	4

```
1 \begin{tabular}[t]{|c|c|}\hline
2 1 & 2 \\ \hline
3 3 & 4 \\ \hline
4 \end{tabular}
```



## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse
- Kopf-/Fußzeile, Bilder, Tabellen
- **Formelsatz**
- Fehlerbehandlung
- Fragen vom letzten Termin
- Aufgaben

# Formelsatz

Es gibt inline Formeln  $1^2 + 1^{1-1} = 2$  und auch abgesetzte Formeln

$$\alpha_{i\dots n}^2 + \beta_1^2 = \psi$$

ebenso wie Brüche:  $\frac{1}{2}$  und Wurzeln:  $\sqrt[3]{27}$  und Summen:  $\sum_{\infty^-}^{\infty^+}$

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4 Es gibt inline Formeln  $1^2 + 1^{1-1} = 2$  und auch abgesetzte
  Formeln  $\alpha_{i\dots n}^2 + \beta_1^2 = \psi$  ebenso
  wie Brüche:  $\frac{1}{2}$  und Wurzeln:  $\sqrt[3]{27}$  und
  Summen:  $\sum_{\infty^-}^{\infty^+}$ 
5 \end{document}
```

nicht mit  $\$ \$ \dots \$ \$$  für abgesetzte Formeln arbeiten!

# Weitere Umgebungen

- `gather`: zentrierte Formeln
- `align`: Ausrichtung wie bei Tabellen mit `&`
- `multiline` und `split`: Aufteilung
- `cases`: Fallunterscheidung
- ...und noch mehr!

## komplexeres Beispiel

$$\mathcal{F}_\omega\{x(t)\} = \int_{-\infty}^{\infty} \exp^{-j\omega t} dt \quad (\text{Fourier-Transformation})$$

```
1 \documentclass[a4paper]{article}
2 \usepackage{amsmath}
3 \begin{document}
4   \begin{align}
5     \mathcal{F}_{\omega}\{x(t)\} = \int_{-\infty}^{\infty} \exp
      ^{-\mathrm{j} \omega t} \mathrm{d}t
6   \tag{Fourier-Transformation}
7   \end{align}
8 \end{document}
```

Für „wichtige“ Dokumente ISO-31 Norm beachten!

## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse
- Kopf-/Fußzeile, Bilder, Tabellen
- Formelsatz
- **Fehlerbehandlung**
- Fragen vom letzten Termin
- Aufgaben

# Fehlerbehandlung

- 2 Fehlerarten: *errors* und *warnings*
- *errors*:
  - ▶ Gravierender Fehler! Übersetzung stoppt!
  - ▶ Hinweistext oft unverständlich . . .
- *warnings*:
  - ▶ L<sup>A</sup>T<sub>E</sub>X benutzt Strafpunktesystem für „unschönen“ Text
  - ▶ *badness* gibt Schweregrad an (max. 10000)
- „GoToError“-Funktion nutzen (falls verfügbar)
- Manchmal: `.aux`-Dateien löschen!
- Wissenschaft für sich! Deshalb erst gar keine Fehler machen :-)

## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse
- Kopf-/Fußzeile, Bilder, Tabellen
- Formelsatz
- Fehlerbehandlung
- **Fragen vom letzten Termin**
- Aufgaben

# Vorgaben einhalten

Ränder, Zeilenabstand, ...?

Ränder:

```
1 \documentclass[a4paper]{article}
2 \usepackage[left=3cm, right=3cm, top=2.5in]{geometry}
3 \begin{document}
4 Ränder korrekt einstellen
5 \end{document}
```



# Vorgaben einhalten

Ränder, Zeilenabstand, ...?

Zeilenabstand: (*nicht* empfehlenswert!)

```
1 \documentclass[a4paper]{article}
2 \usepackage{setspace}
3 \onehalfspacing %1.5-facher Zeilenabstand
4 % \doublespacing % 2-facher Zeilenabstand
5 \begin{document}
6 Zeilenabstand manipulieren.
7 \begin{singlespace}1-fach\end{singlespace}
8 \begin{onehalfspace}1.5-fach\end{onehalfspace}
9 \begin{doublespace}2-fach\end{doublespace}
10 \end{document}
```

## PDF/A-1a bzw. PDF/A-1b mit pdf<sub>l</sub>atex?

- PDF/A-1a: garantiert „textuelle“ Reproduzierbarkeit
  - ▶ Nur mit sehr viel Aufwand mit pdf<sub>l</sub>atex möglich – „Handarbeit“
- PDF/A-1b: garantiert „visuelle“ Reproduzierbarkeit
  - ▶ Passendes Paket: [a-1b]pdf<sub>x</sub>
  - ▶ Konformität muss dennoch überprüft werden!

## 2 Teil 2 – Basiswerkzeug

- Wichtige Hinweise, Textformatierung, Aufzählungen
- Anführungszeichen, Strukturieren, Titelseiten, Inhaltsverzeichnisse
- Kopf-/Fußzeile, Bilder, Tabellen
- Formelsatz
- Fehlerbehandlung
- Fragen vom letzten Termin
- **Aufgaben**

# Was jetzt?

- Feedback? → LaborWiki!
- Pizzabestellung?
- Aufgaben bearbeiten
- Material und Aufgaben im Wiki

## 3 Teil 3 – Makroprogrammierung

- Diverse Pakete, Makroprogrammierung und/oder Formulare