

LaTeX-Praxis

# Einführung

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

# Übersicht

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

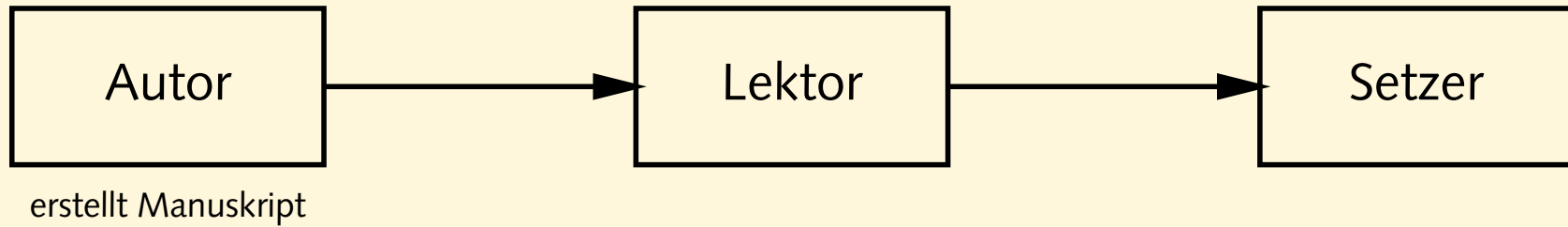
# Was ist $\text{\LaTeX}$ ?

- $\text{\LaTeX}$  ...
  - ist keine Textverarbeitung (wie StarOffice, Word, ...)
  - bietet kein WYSIWYG (what you see is what you get)
  - ist 20 Jahre 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

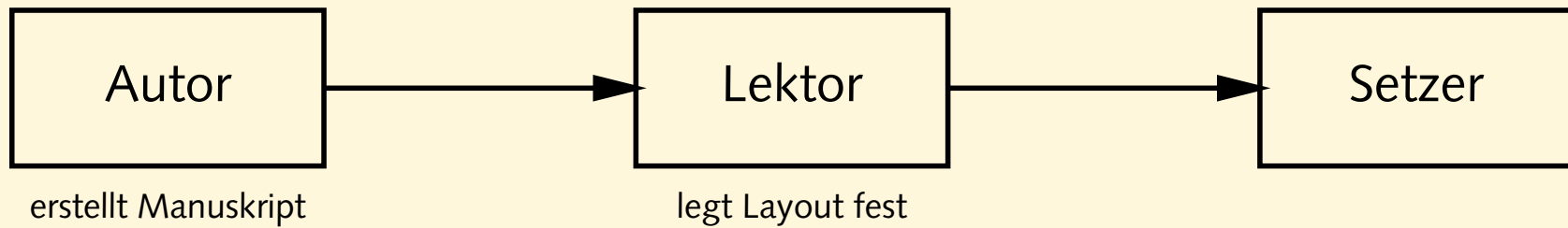
# Textverarbeitung – damals ...



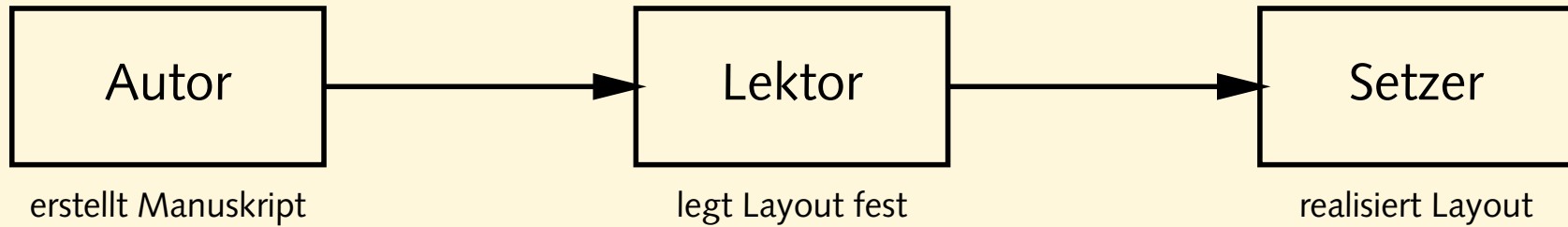
# Textverarbeitung – damals ...



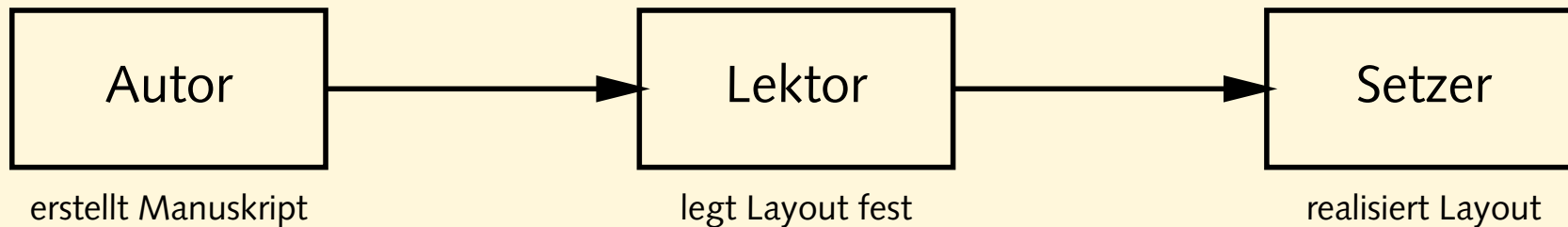
# Textverarbeitung – damals ...



# Textverarbeitung – damals ...



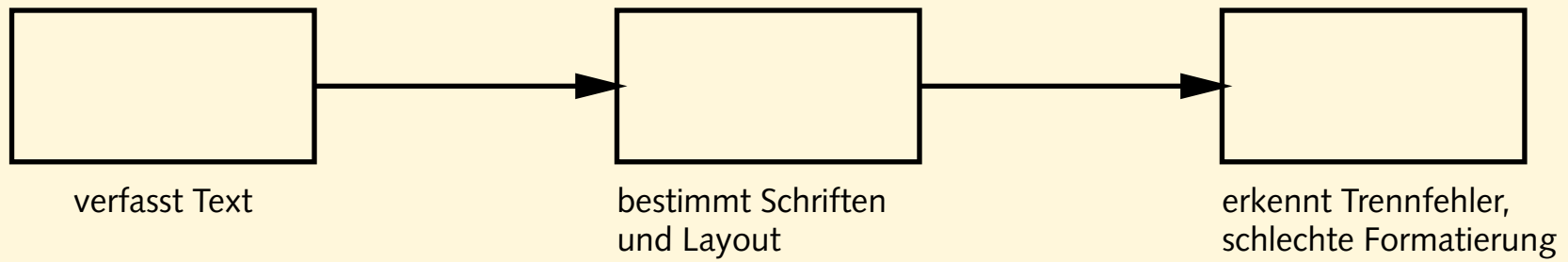
# Textverarbeitung – damals ...



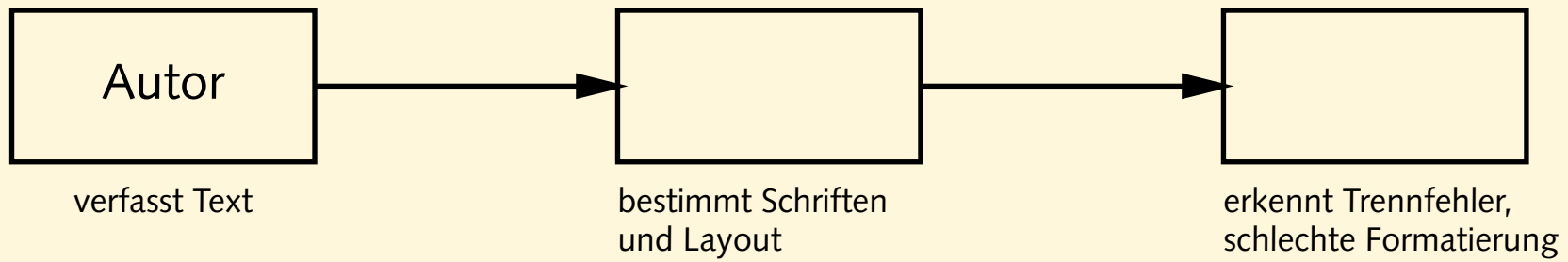
- Tätigkeit von Lektor/Layouter und Setzer
  - nicht trivial
  - Handwerk
  - Kunst
  - braucht Erfahrung



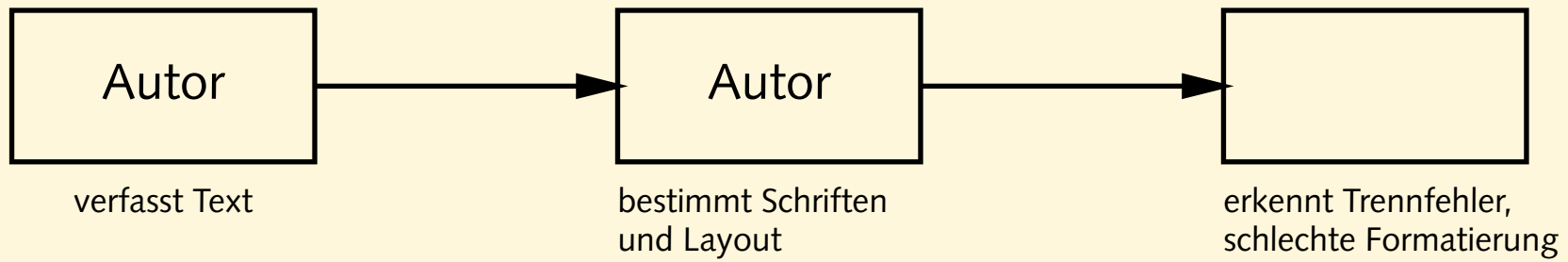
## ... und heute



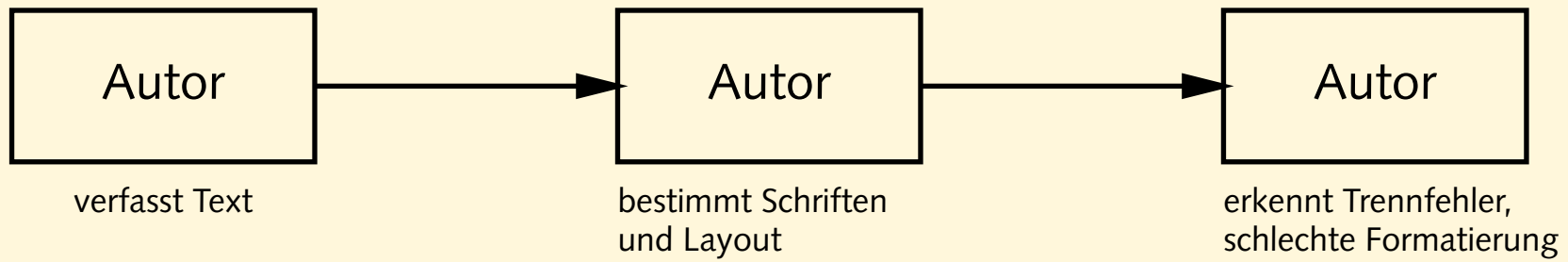
## ... und heute



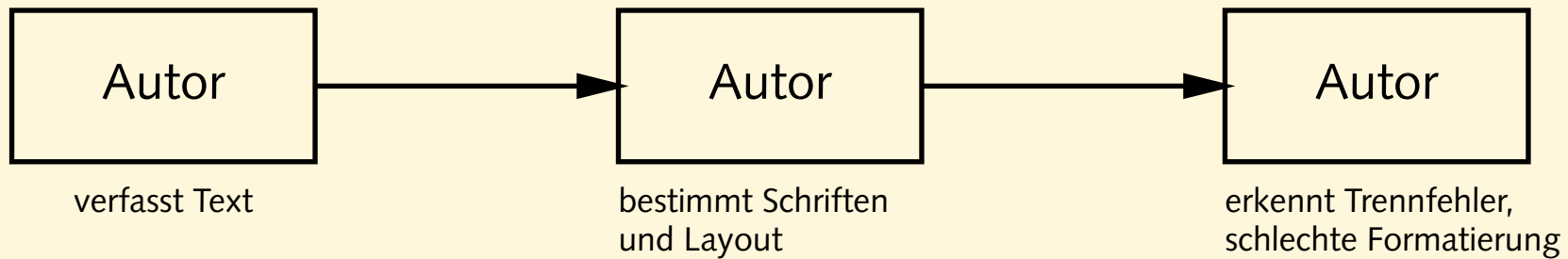
## ... und heute



## ... und heute



## ... 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

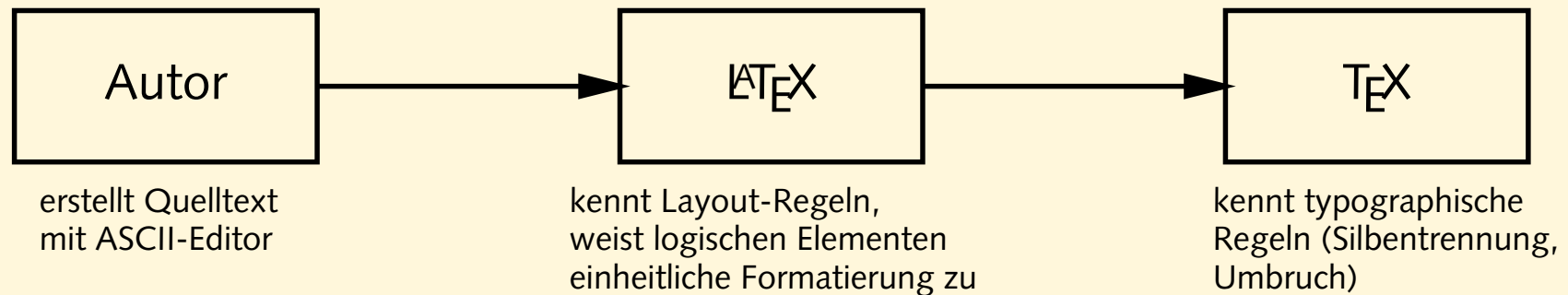
kursiv

Überschrift

statt

15pt, fett, zentriert

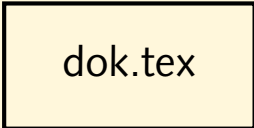
# Schriftsatz mit $\text{\LaTeX}$



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

# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*



dok.tex



# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

```
juser@vino latex dok.tex
```

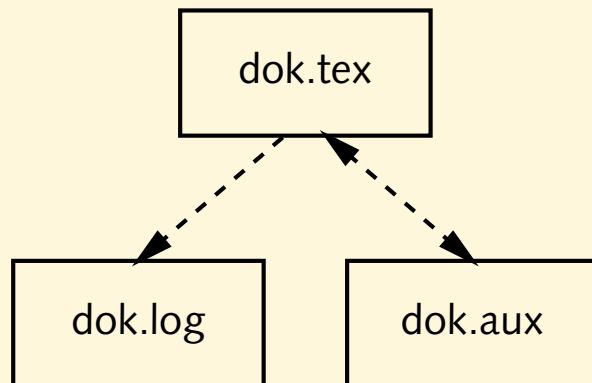


dok.tex

# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

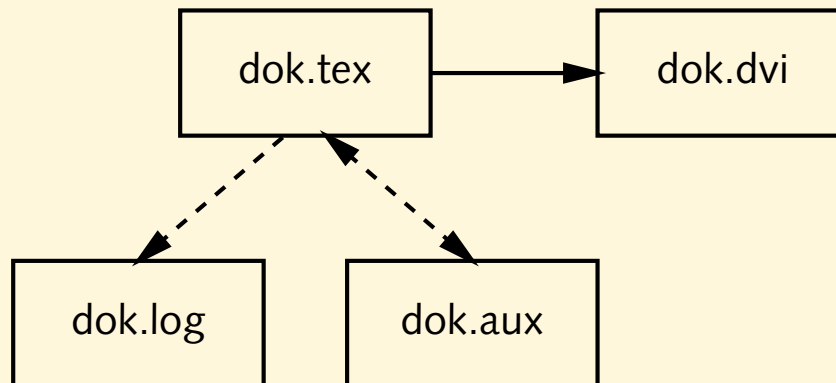
```
juser@vino latex dok.tex
```



# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

```
juser@vino latex dok.tex
```

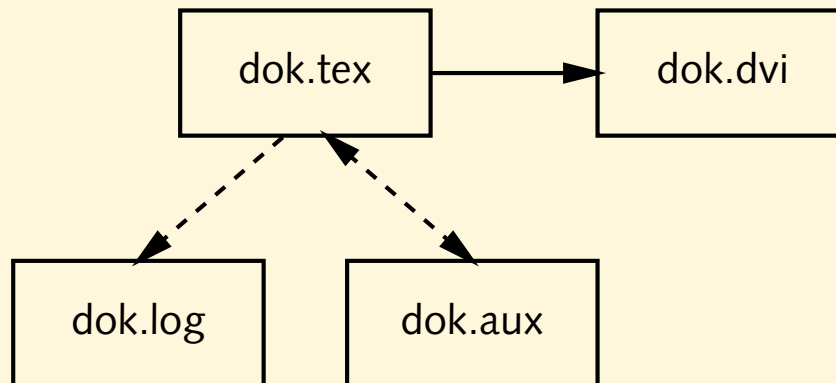


# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

```
juser@vino latex dok.tex
```

```
juser@vino xdvi dok.dvi
```



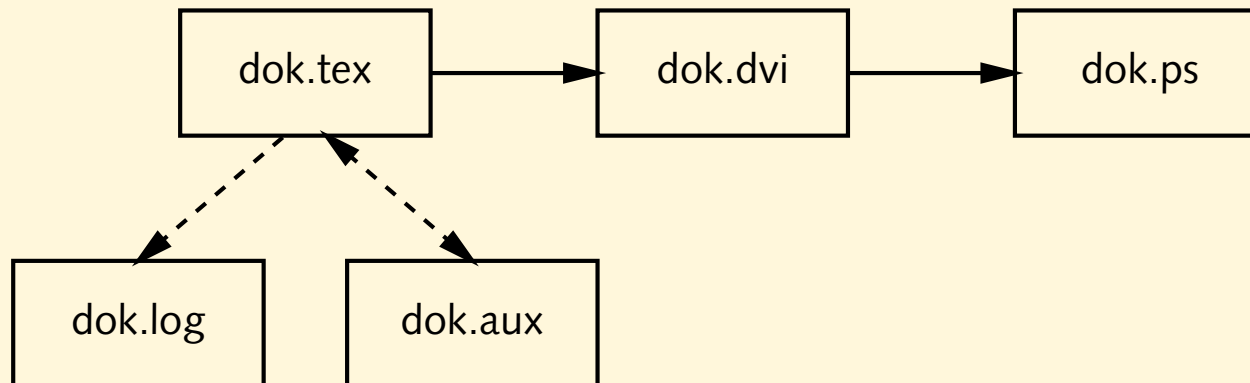
# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

```
juser@vino latex dok.tex
```

```
juser@vino xdvi dok.dvi
```

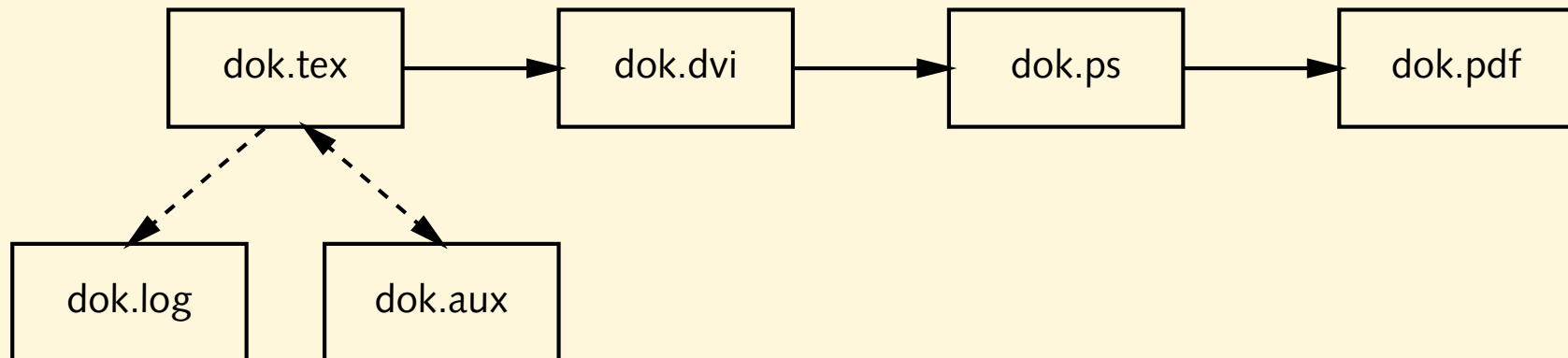
```
juser@vino dvips dok.dvi -o
```



# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

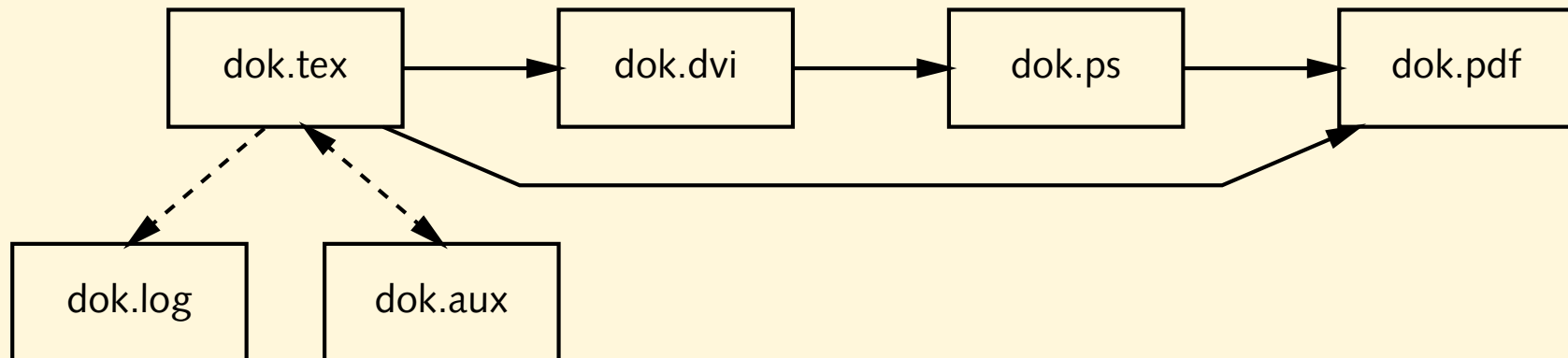
```
juser@vino latex dok.tex  
juser@vino xdvi dok.dvi  
juser@vino dvips dok.dvi -o  
juser@vino ps2pdf dok.ps
```



# Datenfluss in L<sup>A</sup>T<sub>E</sub>X

- batch-System: Dokumente werden *compiliert*

```
juser@vino latex dok.tex  
juser@vino xdvi dok.dvi  
juser@vino dvips dok.dvi -o  
juser@vino ps2pdf dok.ps  
juser@vino pdflatex dok.tex
```



# Aufgaben

- Eine minimale  $\text{\LaTeX}$ -Datei könnte so aussehen:

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

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

- Welche Dateien entstehen zusätzlich zur dvi-Datei?



# Aufgaben

- Was könnten die einzelnen Zeilen in der Datei bedeuten?

- Ersetze die erste Zeile durch

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

und übersetze die Datei erneut. Was hat sich geändert?

- Konvertiere die dvi-Datei in eine PostScript-Datei

```
juser@vino dvips hello.dvi -o
```

und sieh Dir das Ergebnis mit Ghostview (gv) an.

- Konvertiere die PostScript-Datei in eine PDF-Datei

```
juser@vino ps2pdf '-sPAPERSIZE=a4' hello.ps
```

und sieh sie Dir mit dem Acrobat Reader (acroread) an.

# 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?

# Umlaute und Anführungszeichen

```
\documentclass[12pt,a4paper]{article}
\usepackage[german]{babel}
\usepackage[latin1]{inputenc}
```

```
\begin{document}
```

```
Umlaute: \"a \"o \"u \"A \"O \"U \ss{}
```

```
Akzente: \'a \'a ^a \c{C} \O{} \~n
```

```
Anführungszeichen: ``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.
- Kopiere einen deutschen Text aus dem Web in die  $\LaTeX$ -Datei. Falls er „echte“ Umlaute enthält, binde das Paket

```
\usepackage[latin1]{inputenc}
```

ein.

# 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 ganz am Anfang des Textes die Anweisung `\tableofcontents` ein. Was passiert beim Übersetzen? Welche Dateien entstehen?
- Übersetze die  $\text{\LaTeX}$ -Datei ein zweites Mal. Was hat sich verändert? Erkläre Deine Beobachtungen.

# 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?

# Dokumentklassen

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

- Gliederungsstufen:

<code>article</code>		<code>\section</code>	<code>\subsection</code>	<code>...</code>
<code>report</code>	<code>\chapter</code>	<code>\section</code>	<code>\subsection</code>	<code>...</code>
<code>book</code>	<code>\chapter</code>	<code>\section</code>	<code>\subsection</code>	<code>...</code>

- `...`, `\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.

# Aufgaben

- Erweitere die  $\text{\LaTeX}$ -Datei folgendermaßen:

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

Welchen Effekt haben die zusätzlichen Anweisungen?