

Einführung in das Textsatzsystem \LaTeX

Allgemeine Formatierung und Pakete

Moritz Brinkmann
`moritz.brinkmann@iwr.uni-heidelberg.de`

28. Oktober 2016

- ① Engines und Formate
- ② Makrotypografie
 - Der Satzspiegel
 - Kopf- und Fußzeilen
 - Umgebungen
 - Schriften (und Kodierungen)
- ③ Mikrotypografie
- ④ Sprachen
- ⑤ Dokumentation
- ⑥ Fehlermeldungen

engine Das Programm, das die eigentliche Satz-Arbeit macht:
T_EX, pdfT_EX, LuaT_EX

format Große Sammlung von Makros, die die Arbeit erleichtern sollen:
plainT_EX, L^AT_EX, ConT_EXt

distribution Bundle von Engines, Formaten, Erweiterungen (Paketen, Modulen) und Hilfsprogrammen:
T_EX Live, MacT_EX, MiK_T_EX

T_EX-Engines und -Formate

Wichtige Engines

- T_EX** Das ursprüngliche, von Donald E. Knuth geschriebene Programm.
- pdfT_EX** Engine, die direkt PDF-Dateien schreiben kann
Ermöglicht viele PDF-spezifische Features wie z. B. Mikrotypografie.
- X_YT_EX** Verarbeitet standardmäßig utf8-Encoding, bietet die Möglichkeit, Systemschriften zu benutzen und die Textrichtung einfach zu ändern.
- LuaT_EX** Bietet quasi alles was X_YT_EX kann und enthält die Skriptsprache Lua, die man aus dem T_EX-Dokument heraus aufrufen kann.

Ausgeführtes Programm bestimmt Engine und Format:

`pdftex` pdfT_EX-Engine, plain-Format

`pdflatex` pdfT_EX-Engine, L^AT_EX 2_ε-Format

`latex` pdfT_EX-Engine, L^AT_EX 2_ε-Format, DVI-Output

`xelatex` X_ƎT_EX-Engine, L^AT_EX 2_ε-Format

`lualatex` LuaT_EX-Engine, L^AT_EX 2_ε-Format, PDF-Output

Teil II

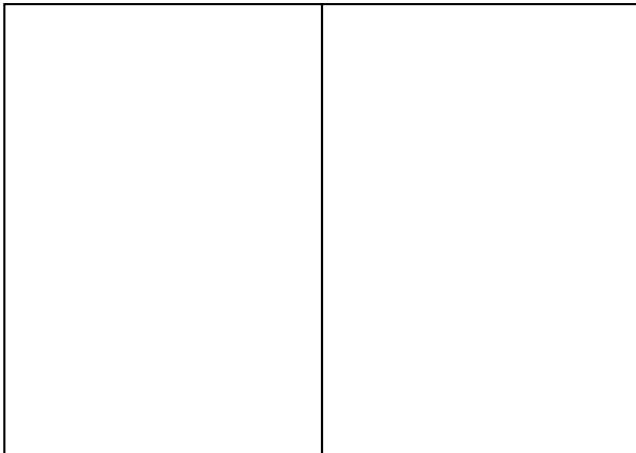
Makrotypografie

- Satzspiegel
- Kopf und Fußzeilen
- Wahl der Schriften
- Formatierung von Abständen
- Aussehen von Verzeichnissen, Fußnoten, ...

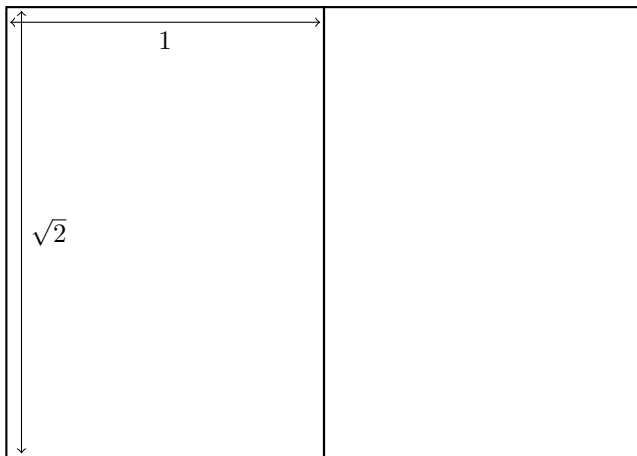
Mit Satzspiegel bezeichnet man die vom Text bedeckte Fläche (im Gegensatz zu den Rändern)

- Ein- oder zweiseitiger Satz?
- Schriftgröße, Laufweite,
- Kopf- und Fußzeilen
- Textspalten

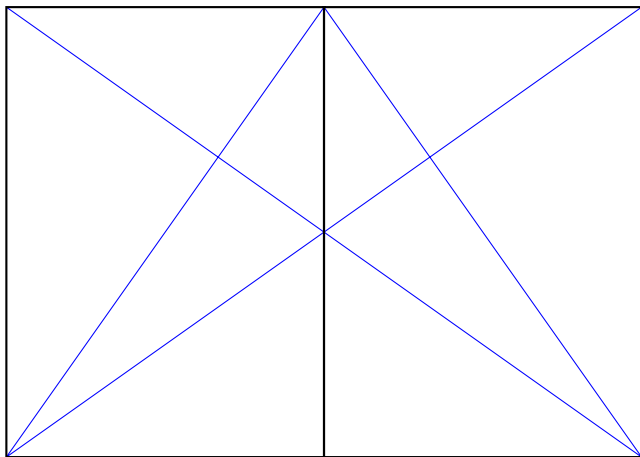
Klassische Satzspiegelkonstruktion



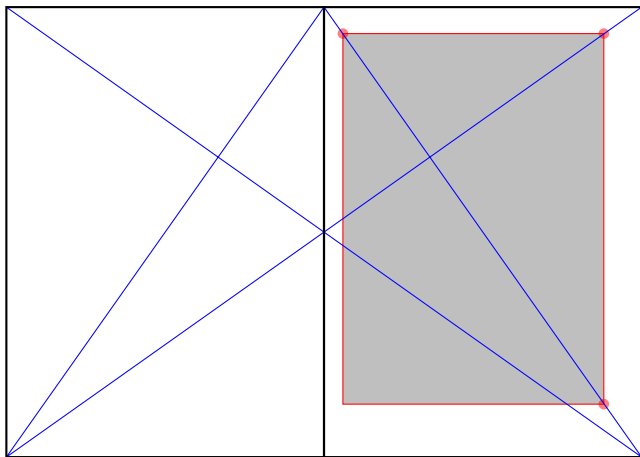
Klassische Satzspiegelkonstruktion



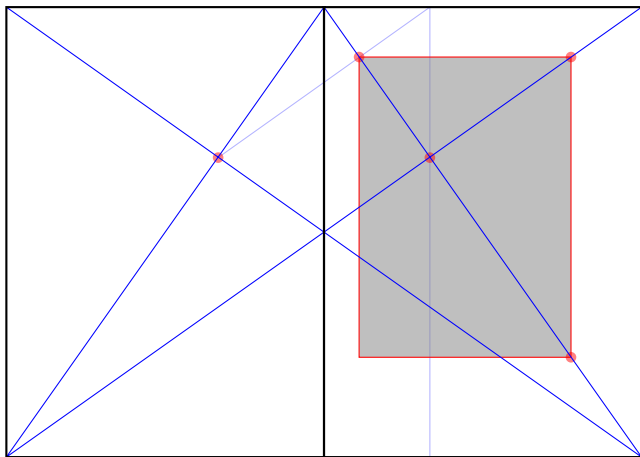
Klassische Satzspiegelkonstruktion



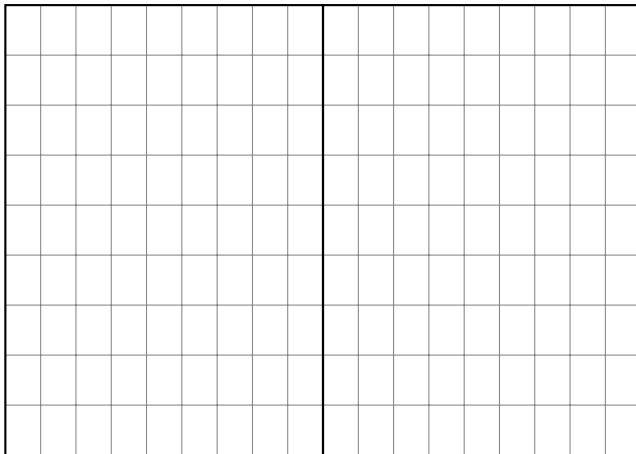
Klassische Satzspiegelkonstruktion



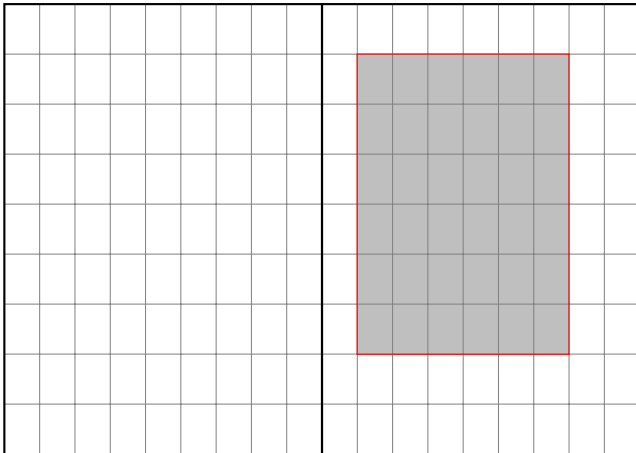
Klassische Satzspiegelkonstruktion



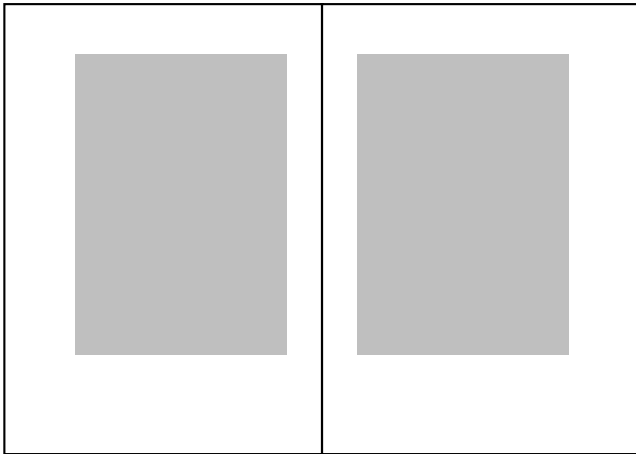
Satzspiegelkonstruktion mit Neunerteilung



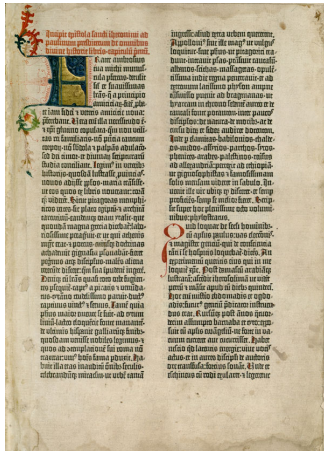
Satzspiegelkonstruktion mit Neunerteilung



Satzspiegelkonstruktion mit Neunerteilung



Satzspiegel bei Gutenberg



Satzspiegel mit KOMA-Skript

- KOMA-Skript bietet optimale Satzspiegelkonstruktion mittels eigenem Paket typearea
- Anpassung eigentlich nur bei besonders breiten oder engen Schriften nötig: Option DIV=*⟨Faktor⟩*
Autom. Berechnung anhand der Seitengröße: DIV=calc
Berechnung nach mittelalterl. Buchseitenkanon: DIV=classic
- Bindekorrektur mittels Option BCOR=*⟨Länge⟩*

```
\documentclass[DIV=9, BCOR=12mm]{scrbook}
```

Bei Nicht-KOMA-Klassen muss typearea direkt geladen werden:

```
\usepackage[DIV=13, BCOR=2cm]{typearea}
```

In Overleaf ausprobieren:



<http://polr.me/tex0101>

Satzspiegel mit geometry

Paket geometry erlaubt manuelle Einstellung des Satzspiegels:

```
\usepackage[top=2cm, bottom=5cm]{geometry}
```

oder:

```
\usepackage{geometry}  
\geometry{top=2cm, bottom=5cm}
```

In Overleaf ausprobieren:



<http://polr.me/tex0101>

mögliche Optionen

```
paper  
left, right, inner, outer, hmargin  
top, bottom, vmargin  
margin  
bindingoffset, textwidth, textheight  
twocolumn, columnsep, marginparsep, footnotesep  
headsep, footsep, nofoot, nohead  
hoffset, voffset, offset
```

In Overleaf ausprobieren:



<http://polr.me/tex0101>

Kopf- und Fußzeilen

- Kopf- und Fußzeilen enthalten wichtige Informationen über das Dokument
 - lebende Kolumnentitel
 - Seitenzahlen
- Anpassung mittels verschiedener Pakete
- Auswahl über `\pagestyle{Seitenstil}` oder `\thispagestyle{Seitenstil}`
- Voreinstellungen: `empty`, `plain`, `headings`

In Overleaf ausprobieren:



<http://polr.me/tex0101>

Kopf- und Fußzeilen mit fancyhdr

```
\usepackage{fancyhdr}  
\pagestyle{fancy}
```

Einseitiger Satz:

```
\lhead{}    \lfoot{}  
\chead{}    \cfoot{}  
\rhead{}    \rfoot{}
```

Zweiseitiger Satz:

```
\fancyhead[LO]{}  
\fancyhead[RO,LE]{}  
\fancyhead[CE]{}  
\fancyfoot[LO]{}  
\fancyfoot[RO,LE]{}  
\fancyfoot[CE]{}  

```

In Overleaf ausprobieren:



<http://polr.me/tex0101>

Kopf- und Fußzeilen mit scrlayer-scrpage

Paket definiert zwei Seitenstile: scrheadings und screadings.plain
Anpassung mittels z. B.

`\lehead[<Inhalt plain.scrheadings>]{<Inhalt scrheadings>}`

<code>\lehead</code> <code>\cehead</code> <code>\rehead</code>	<code>\lohead</code> <code>\cohead</code> <code>\rohead</code>
<code>\lefoot</code> <code>\cefoot</code> <code>\refoot</code>	<code>\lofoot</code> <code>\cofoot</code> <code>\rofoot</code>

```
\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead*{Peter Musterheinzl}
\rohead*{Seitenstile mit KOMA-Script}
\pagestyle{scrheadings}
```

In Overleaf ausprobieren:

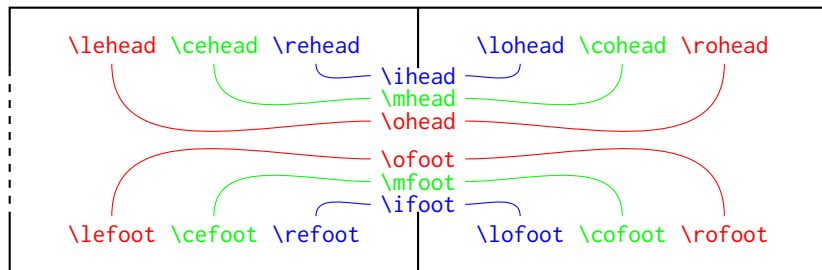


<http://polr.me/tex0101>

Kopf- und Fußzeilen mit scrlayer-scrpage

Paket definiert zwei Seitenstile: scrheadings und screadings.plain
Anpassung mittels z. B.

`\lehead[<Inhalt plain.scrheadings>]{<Inhalt scrheadings>}`



```
\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead*{Peter Musterheinzl}
\rohead*{Seitenstile mit KOMA-Script}
\pagestyle{scrheadings}
```

In Overleaf ausprobieren:



<http://polr.me/tex0101>

- \LaTeX -Dokumente werden oft von Umgebungen strukturiert:

```
\begin{⟨Umgebung⟩}[⟨ggf. opt. Argumente⟩]{⟨ggf. Argumente⟩}
```

...

```
\end{⟨Umgebung⟩}
```

- Am Anfang und Ende werden Befehle ausgeführt um bestimmtes Verhalten innerhalb der Umgebung zu erreichen.
- Jede Umgebung ist eine Gruppierung (wie `{}`)
⇒ Alle Einstellungen innerhalb einer Umgebung sind lokal.

wichtige Umgebungen

Aufzählung	itemize
Nummerierung	enumerate
wörtliche Wiedergabe	verbatim
zweispaltiger Satz	twocolumn
Zitat	quotation
zentriert	center
abgeschlossene Einheit	minipage
Tabelle	tabular, tabularx, tabulary, supertabular etc.
Abbildung	figure
Gleitumgebung	table
Beamerfolie	frame
Gleichung	align (Mathe)
Matrix	matrix (Mathe)

```
\begin{itemize}  
  \item Erster Punkt  
  \item Zweiter Punkt  
  \item[3] Dritter Punkt  
\end{itemize}
```

- Erster Punkt
- Zweiter Punkt
- 3 Dritter Punkt

```
\begin{enumerate}  
  \item Erster Punkt  
  \item Zweiter Punkt  
  \item[3] Dritter Punkt  
\end{enumerate}
```

- ① Erster Punkt
- ② Zweiter Punkt
- 3 Dritter Punkt

Aussehen von `itemize` und `enumerate` wird von Dokumentenklasse bestimmt.

- Früher™ hat man Buchstaben mit 7 bit gespeichert
z. B. ASCII-Zeichensatz:

`_!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~`





- FrüherTM hat man Buchstaben mit 7 bit gespeichert
z. B. ASCII-Zeichensatz:

```
_!"#$%&'()*+,-./0123456789:;<=>?  
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_  
`abcdefghijklmnopqrstuvwxyz{|}~
```

- pdf_{La}T_EX geht von ASCII-Kodierung aus und versteht normalerweise keine Umlaute.
Kodierung kann mittels `\usepackage[utf8]{inputenc}` auf Unicode umgestellt werden.



- Früher™ hat man Buchstaben mit 7 bit gespeichert
z. B. ASCII-Zeichensatz:

```
_! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _  
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
```

- pdf \LaTeX geht von ASCII-Kodierung aus und versteht normalerweise keine Umlaute.
Kodierung kann mittels `\usepackage[utf8]{inputenc}` auf Unicode umgestellt werden.
- X \LaTeX und Lua \LaTeX gehen von UTF8-Kodierung aus.

- Auch wenn pdf \LaTeX Unicode-Eingabe versteht, erscheinen in der Ausgabe nicht unbedingt Umlaute. z. B. $\ddot{u} \rightarrow \text{"u}$
- Ausgabekodierung kann festgelegt werden mittels `\usepackage[Kodierung]{fontenc}`
- Es verschiedene Kodierungen zur Verfügung:
OT1 (original \TeX -Encoding, 7 bit), T1 (Latein, Mitteleuropa, 8 bit), T2A – T2C (Kyrillisch), T3 (Phonetisches Alphabet), T4 (Latein, Afrika), T5 (Vietnamesisch), ...

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

- Auch wenn pdf \LaTeX Unicode-Eingabe versteht, erscheinen in der Ausgabe nicht unbedingt Umlaute. z. B. $\ddot{u} \rightarrow \text{"u}$
- Ausgabekodierung kann festgelegt werden mittels $\text{\usepackage[*Kodierung*]{fontenc}}$
- Es verschiedene Kodierungen zur Verfügung:
OT1 (original $\text{T}_{\text{E}}\text{X}$ -Encoding, 7 bit), T1 (Latein, Mitteleuropa, 8 bit), T2A – T2C (Kyrillisch), T3 (Phonetisches Alphabet), T4 (Latein, Afrika), T5 (Vietnamesisch), ...

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

- $\text{X}_{\text{E}}\text{\LaTeX}$ und Lua \LaTeX nutzen intern automatisch EU1- bzw. EU2-Kodierung (Unicode). T1 muss nur bei Verwendung von pdf \LaTeX -Schriften explizit angegeben werden.

Schriften in pdf \LaTeX

- pdf \LaTeX benötigt bestimmtes Schriftformat (\TeX font metrics)
- Schriften werden mittels Paketen geladen.

```
\usepackage{kpfonts}
```

- In CTAN verfügbare Schriften findet man z. B. im „LaTeX Font Catalogue“
<http://www.tug.dk/FontCatalogue/>



In Overleaf ausprobieren:



<http://polr.me/tex0102>

Schriften in Xe_ΛT_EX und Lua_ΛT_EX

- Paket fontspec erlaubt es auf Systemschriften (OTF, AAT, TTF) zuzugreifen.

- Fonts werden über spezielle Befehle geladen

`\setmainfont[Optionen]{Name der Schrift}`

```
\setromanfont{Linux Libertine 0}  
\setsansfont{Linux Biolinum 0}  
\setmonofont[Scale=.95]{DejaVu Sans Mono}
```

- Laden bestimmter Schriften oder Features im Dokument mit

`\fontspec{Name der Schrift}[Features]`

Die Größe der Browserschrift kann durch Klassenoption geändert werden:

```
\documentclass[12pt]{scrartcl}
```

Größe von `\large`, `\small`, etc. passt sich automatisch an.
Standardklassen unterstützen 10pt, 11pt und 12pt.

Die Größe der Browserschrift kann durch Klassenoption geändert werden:

```
\documentclass[12pt]{scrartcl}
```

Größe von `\large`, `\small`, etc. passt sich automatisch an.
Standardklassen unterstützen 10pt, 11pt und 12pt.

Wer *genau weiß*, was er will:

```
\fontsize{⟨Größe⟩}{⟨Durchschuss⟩}\selectfont
```

```
\fontsize{10}{12}\selectfont
```

Teil III

Mikrotypografie

Mikrotypografie

Mikrotypografie bezeichnet die Gestaltung von Feinheiten auf Buchstabenebene:

tracking Anpassung des
Glyphenabstands innerhalb
der Wörter ($\leq 3\%$)

VA Fo
VA Fo

Mikrotypografie

Mikrotypografie bezeichnet die Gestaltung von Feinheiten auf Buchstabenebene:

tracking Anpassung des
Glyphenabstands innerhalb
der Wörter ($\leq 3\%$)

expansion Anpassung der Glyphenbreite
($\leq 2\%$)

Text
Text

Mikrotypografie bezeichnet die Gestaltung von Feinheiten auf Buchstabenebene:

tracking Anpassung des
Glyphenabstands innerhalb
der Wörter ($\leq 3\%$)

expansion Anpassung der Glyphenbreite
($\leq 2\%$)

protrusion Optischer Randausgleich

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Mikrotypografie

Mikrotypografie bezeichnet die Gestaltung von Feinheiten auf Buchstabenebene:

tracking	Anpassung des Glyphenabstands innerhalb der Wörter ($\leq 3\%$)	fi fi fl fl
expansion	Anpassung der Glyphenbreite ($\leq 2\%$)	ff ff ffl ffl
protrusion	Optischer Randausgleich	Qu Qu
ligatures	Verbindung mehrerer Buchstaben zu einer Glyphe	

Das Paket `microtype` kümmert sich um diese typografischen Feinheiten.

In der Regel reicht die Voreinstellung:

```
\usepackage{microtype}
```

- Aktiviert automatisch `protrusion` (in `pdfTeX`, `XYTeX` und `LuaTeX`) und `expansion` (in `pdfTeX` und `LuaTeX`)
- Für weitere Optionen: Dokumentation

Leerräume und Striche

Gute Typografie unterscheidet zwischen verschieden breiten Leerzeichen und horizontalen Strichen

- normales Leerzeichen
- schmales Leerzeichen (Spatium): \, z. B. z. B. z.B.
- kleiner Abstand (Halbgeviert): \enskip a b
- weißes Quadrat (Geviert): \quad a b
- negativer Abstand: \! ab

Leerräume und Striche

Gute Typografie unterscheidet zwischen verschieden breiten Leerzeichen und horizontalen Strichen

- normales Leerzeichen
- schmales Leerzeichen (Spatium): `\,` z. B. z. B. z.B.
- kleiner Abstand (Halbgeviert): `\enskip` a b
- weißes Quadrat (Geviert): `\quad` a b
- negativer Abstand: `\!` ab
- explizites Ändern des Abstands (Kerning): `a\kern-.1em b` ab

Leerräume und Striche

Gute Typografie unterscheidet zwischen verschieden breiten Leerzeichen und horizontalen Strichen

- normales Leerzeichen
 - schmales Leerzeichen (Spatium): \, z. B. z. B. z.B.
 - kleiner Abstand (Halbgeviert): \enskip a b
 - weißes Quadrat (Geviert): \quad a b
 - negativer Abstand: \! ab
 - explizites Ändern des Abstands (Kerning): a\kern-.1em b ab
-
- Viertelgeviertstrich, Bindestrich: - a-b
 - Halbgeviertstrich, Gedankenstrich: -- a—b
 - Geviertstrich, engl. Gedankenstrich: --- a—b
 - Minuszeichen: \$-\$ $a - b$
 $a + b$

Teil IV

Sprachen, Dokumentation & Fehlermeldungen

Dokument muss je nach Eingabesprache lokalisiert werden.

- Umbruchregeln
- Bezeichnungen von Verzeichnissen, Kapiteln, ...
- typografische Besonderheiten

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

Dokument muss je nach Eingabesprache lokalisiert werden.

- Umbruchregeln
- Bezeichnungen von Verzeichnissen, Kapiteln, ...
- typografische Besonderheiten

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

Modernere Alternative für Lua_ΛT_EX und X_ΛT_EX:

```
\usepackage{polyglossia}  
\setmainlanguage{german}
```


Standardpakete

pdf \LaTeX

```
\usepackage[ngerman]{babel}  
\usepackage[T1]{fontenc}  
\usepackage[utf8]{inputenc}
```

X \LaTeX

```
\usepackage{polyglossia}  
\setmainlanguage{german}  
\usepackage{xltextra}
```

Lua \LaTeX

```
\usepackage{fontspec}  
\usepackage{polyglossia}  
\setmainlanguage{german}
```



- \LaTeX ist hervorragend dokumentiert
- Jede Klasse und jedes Paket bringt normalerweise eine eigene Anleitung mit.
- Dokumentation kann mittels des `texdoc`-Befehls aufgerufen werden

Auf der Kommandozeile:

- `texdoc` durchsucht die \LaTeX -Ordner nach Dokumentationen
- `texdoc amsmath` öffnet `amsmath.pdf`
- `texdoc -l amsmath` listet alle Ergebnisse auf
- `texdoc -s amsmath` liefert Ergebnisse aus erweiterter Suche
- `texdoc --help` zeigt eine Hilfe an

Mit `texdoctk` existiert eine grafische Oberfläche

Was tun, wenn L^AT_EX anhält?

- Ruhe bewahren! (tex-Dateien können nicht beschädigt werden)
- Mit der Fehlersuche beim den letzten Änderungen anfangen.
- Ggf. Schreibfehler korrigieren.
- log-Datei Lesen!
- Viele Editoren helfen bei der Fehlersuche, indem sie zur Zeile springen, in der der Fehler aufgetreten ist. (Das muss nicht die fehlerhafte Zeile sein.)

Typische Fehlermeldung:

```
! Undefined control sequence.  
l.3 Ein \Latex-Dokument  
  
?  
! Emergency stop.  
l.3 Ein \Latex-Dokument.  
  
No pages of output.  
Transcript written on document.log.
```

⇒ Befehl in Zeile 3 falsch geschrieben

Typische Fehlermeldung:

```
Runaway argument?  
{itemize \item Erstes Item  
! Paragraph ended before \begin was complete.  
<to be read again>  
                                \par  
1.60  
  
?
```

⇒ Irgendwo nach itemize ein } oder ein \end{} vergessen.

Vollständiges Minimalbeispiel

Bei Hilfestellung in Webforen/Usenet wird in der Regel ein *vollständiges Minimalbeispiel* (MWE) verlangt.

- ① solange Code aus dem Dokument löschen, bis der Fehler gerade noch auftritt
- ② alle überflüssigen Pakete entfernen
- ③ falls Dokumentenklasse keine Rolle spielt, `minimal` verwenden
- ④ wenn Fehler nur bei viel Text auftritt, `blindtext` verwenden

Oft findet man den Fehler beim erstellen des MWE schon ganz alleine.



Markus Kohm.

„Satzspiegelkonstruktionen im Vergleich“

<http://www.dante.de/tex/Dokumente/KohmSatzspiegel.pdf>



Robert Schlicht.

„The microtype package. Subliminal refinements towards typographical perfection“

[texdoc microtype](#)



Friedrich Forssman und Ralf de Jong.

„Detailtypografie“

Mainz: Verlag Hermann Schmidt, 2002.



Robert Bringhurst.

„The Elements of Typographic Style“

Vancouver: Hartley & Marks, 1992.