Einführung in das Textsatzsystem

LaTeX

10 – (Mikro-)Typographie und Kontrollstrukturen

Separates the Lions from the Boys.

24. Juni 2011

Inhalt

Teil I
Mikrotypographie

Typographie
Makrotypographie

- Gestaltung des Dokuments im Großen: Aufteilung in Kapitel, Absätze, Abschnitte
- Anordnung von Text auf einer Seite, Fuß- und Kopfzeilen
- Textumbruch und Absatzausrichtung
- Anordnung von Bildern, konsistente Abstände
- passende Wahl von Schriften (auf Inhalt und Zielgruppe sowie untereinander abgestimmt)

Typographie

„korrekte“ Typographische Zeichen

- Echte Anführungszeichen: „ “ statt
- Mit TeX: `\glqq` `\grqq`
- Mit LaTeX und `babel`: `` `" '" `` oder direkte Eingabe von „ " (mit LuaLaTeX)
- Im Englischen “ ” !
- Korrekte Striche: – statt - etc.
- Korrekte Ellipsen: … statt ...

Typographie
Kuriositätenkabinett

- Im Satz alter Schriftstücke oder bei Verwendung von Fraktur ist das ſ zu verwenden.
- s wird nur am Silbenende verwendet: Fluſs, ſtand, ſt, einſilbig, eiskalt. (In anderen Sprachen u. U. deutlich komplexere Regeln)
- Aufeinandertreffen von ſ und s kann als ß gelöst werden: Fluſs ⟹ Fluß.
- Im Versalsatz kann das zu ß werden.

Typographie
Kuriositätenkabinett

- Im Satz alter Schriftstücke oder bei Verwendung von Fraktur ist das ſ zu verwenden.
- s wird nur am Silbenende verwendet: Fluſs, ſtand, ſt, einſilbig, eiskalt. (In anderen Sprachen u. U. deutlich komplexere Regeln)
- Aufeinandertreffen von ſ und s kann als ß gelöst werden: Fluſs ⟹ Fluß.
- Im Versalsatz kann das zu ß werden.
- Für erstaunte Ausrufe und Verwunderung: Interrobang und Gnaborretni ‽ ⸘

Typographie
Mikrotypographie

typographische Feinheiten auf Buchstaben- oder „Subbuchstabenniveau“:

- character protrusion
- font expansion
- the adjustment of interword spacing
- additional kerning
- hyphenatable letterspacing (tracking)
- possibility to disable all or selected ligatures.

(Aus der `microtype`-Dokumentation)

Typographie
Mikrotypographie

typographische Feinheiten auf Buchstaben- oder „Subbuchstabenniveau“:

- character protrusion
- font expansion
- the adjustment of interword spacing
- additional kerning
- hyphenatable letterspacing (tracking)
- possibility to disable all or selected ligatures.

(Aus der `microtype`-Dokumentation)
Das typolexikon bietet eine ausführlichere Definition; auch:
„Die mikrotypographische Qualität optimiert deutlich die Lesbarkeit einer Schriftsatzarbeit. Sie beeinflusst auch wesentlich deren Glaubhaftigkeit, Anmutung, Inszenierung und Funktion.“

Mikrotypographie

Dies ist ein dummer, kleiner Text, der ohne seltsame Zeichen auskommen soll und einfach nur ein wenig die Effekte von Mikrotypographie, vor allem Font Expansion, allerdings etwas übertrieben, zeigen soll.

Dies ist ein dummer, kleiner Text, der ohne seltsame Zeichen auskommen soll und einfach nur ein wenig die Effekte von Mikrotypographie, vor allem Font Expansion, allerdings etwas übertrieben, zeigen soll.

pdfTeX

- großartige Neuerungen bei Einführung von pdfTeX:
- Erzeugen von pdfs direkt aus dem TeX-Code
  (vorher nur über Umweg dvi $\Longrightarrow$ ps $\Longrightarrow$ pdf möglich)
- spezielle pdf-Features verfügbar (slide transitions ...)
- Schnittstelle zu typographischen Feinheiten

microtype

- Paket microtype bietet einfachen Zugriff auf alle mikrotypographischen Effekte
- benötigt pdfTeX als Engine
- für alle Features: aktuelle Engine (pdfTeX > 1.40, aktuell 1.40.12)
- LuaTeX > 0.25 ermöglicht die meisten Features (aktuell 0.70)
- für LuaTeX mit OpenType Schriften: `microtype v 2.5` nötig!
- darüber hinaus: `fontspec` für XeLaTeX und LuaLaTeX bietet Features auf Schriftebene

microtype

- protrusion, expansion funktionieren „immer“ und sind standardmäßig aktiviert

  deaktivieren: Paketoption oder `\microtypesetup`:
  `[protrusion=false,expansion=false]`

  Feineinstellungen:

  `stretch=50,shrink=30` (default: 20, shrink = stretch)

  `factor=2000` (protrusion; default: 1000)

⟹ Finetuning kann nötig sein bei sehr schmalen Spalten

- andere Features zum Teil experimentell, daher deaktiviert

  aktivieren mittels Optionen
  `[tracking=true,spacing=true,kerning=true]`

microtype

Anpassung

- alle Features sind schriftabhängig!
- Voreinstellungen für wenige Schriften verfügbar
- Standardeinstellungen für unbekannte Schriften
- alle Einstellungen frei und beliebig aufwendig einstellbar
  (Standardeinstellungen sind meist aber völlig ausreichend)

LuaLaTeX mit fontspec

- LuaLaTeX bietet Zugriff auf moderne Schrifttechnologien
- `fontspec` bietet einfache Schnittstelle
- intuitive Schriftbefehle statt low-level-Befehle

LuaLaTeX mit fontspec

- LuaLaTeX bietet Zugriff auf moderne Schrifttechnologien
- `fontspec` bietet einfache Schnittstelle
- intuitive Schriftbefehle statt low-level-Befehle
- möglich:
- schriftunabhängige Features
- Aktivieren von OpenType / AAT-Features
- verschiedene Features für verschiedene Schnitte

fontspec

schriftunabhängige Features

- Skalierung: Option `Scale`
  skaliert die Schrift „ohne Rücksicht auf Verluste“
  speziell: `MatchLowercase` und `MatchUppercase`
- Farbe:
  `\addfontfeature{Color=FF000099}W}` (Farbe FF0000 plus Transparenz 99)
- Interword space
  Skalierung des normalen Wortabstandes:
  `Wordspace=0.8`, `Wordspace={0.8,0.8,0.8}`

fontspec

schriftunabhängige Features

- Platz nach Interpunktion:
  `PunktuationSpace=0.5` (nur bei `\nonfrenchspacing`!)
- Letter spacing
  `LetterSpace=0.1`; `0.5`; `2.0` etc.
- hyphenation character
  `[HyphenChar=None/{-}/"F6BA]`
- Schrifttransformationen, falls keine richtigen Schnitte verfügbar sind (handle with care!)
  `[FakeSlant=0.2,FakeStretch=1.2,FakeBold=1.5]`

fortgeschrittene OpenType-Features (schriftabhängig)

- OpenType-Standard spezifiziert viele typographische Features
- nur wenige Programme können alle korrekt umsetzen
- Typographie wird zu großen Teilen der Schrift überlassen (sowohl Vorteil als auch Nachteil …)
- Setzen beim Laden / Umstellen der Schrift
- nur implementierte Features können verwendet werden

OpenType/AAT-Features

- optische Größe:
  `[OpticalSize=12]`
  (verschiedene Schnitte für Überschriften, Abschnitte etc.)
  `SizeFeatures={{Size=-10,OpticalSize=8},`
  `{Size=10-14,OpticalSize=10}}`
- Ligaturen (otf bieten sehr vielfältige Ligaturen):
  `[Ligatures=Rare,NoCommon,Logos,Rebus,`
  `Diphtong,Squared,AbbrevSquared,Icelandic]`
- Letters, Numbers, Contextuals, Vertical position
- Fractions, Variants, Alternates, Style, Diacritics
- Kerning, CJK shape, character width, Annotation

Und nun?
typokurz

- Tips, wie man all diese Features verwenden sollte:
- Dokument `typokurz.pdf` gibt kurze, übersichtliche Anleitung zu typographischen Tips
- im Internet unter dem Dokumentnamen zu finden
- erläutert Textauszeichnungen, Striche, Abkürzen etc.

Teil II
Kontrollstrukturen

Strukturen

- LaTeX als Nutzerebene: normaler Nutzer muss nicht programmieren
- oft für spezielle Anwendungen dennoch nötig:
- Fallunterscheidungen, Schleifen, Abfragen u. ä.
- beim Paketschreiben sind Abfragen fast unumgänglich

Kontrollstrukturen in TeX

- Zähler
- Schleifen (for, while)
- if-Abfragen
- case-Abfragen

Kontrollstrukturen

- Programmieren auf verschiedenen Ebenen möglich:
- TeX selbst bietet rudimentäres Programmierinterface
- LaTeX 2_ε-Kernel erweitert und vereinfacht das Interface
- spezialisierte LaTeX 2_ε-Pakete ermöglichen sehr einfache Bedienung
- LaTeX3-Kernel bietet sehr konsequentes und strukturiertes Interface (basiert aber immer noch auf εTeX)
- LuaTeX: „ordentliche“ Programmiersprache verfügbar, normales Programmieren möglich (*plus* alle Fähigkeiten von TeX, LaTeX 2_ε/LaTeX3)

low level

Zähler

- `\count1 = 15`, `\advance\count1 by 3`, `\count1` $\Longrightarrow$ 18
  in plainTeX: `\newcount\mycounter` möglich
- Ausgabe mit `\the\mycounter` (vgl. `\thepage`)

low level
if

- TeX bietet verschiedenste `if`-Konstrukte an:
- `\if<token1><token2>` prüft, ob character codes übereinstimmen
- `\ifodd<number>` prüft, ob die Zahl ungerade ist
- `\ifnum<number1><relation><number2>` überprüft, ob die Relation erfüllt ist
- `\ifdim`, `\ifv/h/mmode`, `\ifcat`, `\iftrue`, `\iffalse` ...
- Verwendung:
  `\if...<Argumente> then-Teil \else else-Teil \fi`

low level
Schleifen

- TeX bietet einen Schleifenmechanismus:
- `\loop A \if... B \repeat`

$\Longrightarrow$ führt `A` aus, prüft dann `\if...`

- falls `\if` erfüllt, `B`, dann wieder von vorne

LaTeX 2ε-Kernel
Zähler

- vereinfachtes Konstrukt für Zähler:
- `\newcounter{count}`
- `\setcounter{count}{wert}`
- `\addtocounter{count}{wert}`
- `\stepcounter{count}`, `\refstepcounter{referenzcount}`
- `\value{count}`
- formatierte Ausgabe: `\Arabic`, `\arabic`, `\Roman`, `\alph`, `\fnsymbol` (footnote-Symbol)
- siehe Auszüge aus der Dokumentation (`texdoc source2e`)

Pakete

- `forloop` definiert einfache Nutzer-Schnittstelle für Schleifen:

```
\forloop[step]{counter}{initial value}{condition}{code}
\newcounter{ct}\forloop{ct}{1}
                {\value{ct} < 10}{\arabic{ct} }
```

Pakete

- `forloop` definiert einfache Nutzer-Schnittstelle für Schleifen:

```
\forloop[step]{counter}{initial value}{condition}{code}
\newcounter{ct}\forloop{ct}{1}
                {\value{ct} < 10}{\arabic{ct} }
```

- `pgffor` (automatisch bei TikZ dabei) bietet Kontrollstrukturen:

```
\foreach \x in {1,2,3} {$x =\x$, }
```

Pakete

- ifthen bietet sehr einfaches und praktisches Interface für sämtliche \if-Abfragen:
- \ifthenelse{Abfrage}{then}{else}

```
<number> =,<,> <number>
\isodd{ number }
\isundefined{ command name }
\equal{ string }{ string }
\lengthtest{ dimen < dimen }
\lengthtest{ dimen = dimen }
\lengthtest{ dimen > dimen }
\boolean{ name }
```

- \whiledo{}{} bietet einfache while-Schleife

LaTeX3

- LaTeX3 bietet völlig neu überarbeiteten Kernel
- systematischer Aufbau
- strikte Trennung von TeX und Makropaket
- eigene Programmiersprache „in TeX geschrieben“
- viele Definitionen von Kontrollsequenzen
- in LaTeX 2_ε verfügbar über Paket `expl3`
- Boolsche Variablen: `\bool_new:N`, `\bool_set_true:N`, `\boo_if:NTF`

luaTeX: Lua!

- Idee: alles, was „Programmieren“ ist, wird einer richtigen Programmiersprache überlassen

$\Longrightarrow$ Integration von Lua und TeX

$\Longrightarrow$ sämtliche Kontrollstrukturen von Lua direkt verfügbar:

```
\directlua(if a then tex.print("hallo")
                else tex.print("byebye"))
```

lua vs. 3

- LuaTeX und LaTeX3: sinnvolles und benutzbares Programmierinterface
- LuaTeX auf Engine-Ebene, LaTeX3 auf Macroebene

$\Longrightarrow$ Entwicklung von LuaTeX und LaTeX3 wird das LaTeX-Umfeld in den nächsten Jahren stark ändern (vereinfachen!) können – wenn die Nutzer es annehmen …