Einführung in das Textsatzsystem

LaTeX

04 – Umgebungen, Pakete, Fehlersuche

Extend the Lion ~ and Help Him After Tripping

15. November 2013

Inhalt

TeXen beschleunigen

- TeX an sich kann sehr schnell kompilieren
- Schriftenladen mit `fontspec` ist recht langsam
- Setzen mit `microtype` unter LuaLaTeX ist sehr langsam …

TeXen beschleunigen

- TeX an sich kann sehr schnell kompilieren
- Schriftenladen mit `fontspec` ist recht langsam
- Setzen mit `microtype` unter LuaLaTeX ist sehr langsam …

$\Rightarrow$ Wenn man nur „kurz kompilieren“ will, ohne das Endlayout zu sehen:

- `fontspec` und `microtype` nicht laden

TeXen beschleunigen

- TeX an sich kann sehr schnell kompilieren
- Schriftenladen mit `fontspec` ist recht langsam
- Setzen mit `microtype` unter LuaLaTeX ist sehr langsam …

⇒ Wenn man nur „kurz kompilieren“ will, ohne das Endlayout zu sehen:

- `fontspec` und `microtype` nicht laden
- Da TeX nicht parallel laufen kann, bringen mehrere Rechenkerne leider keinen Vorteil

TeXen beschleunigen – ein einfacher Vorschlag

Wechsel zwischen pdfLaTeX und LuaLaTeX, der mit `ifluatex` abgefragt wird:

```
\usepackage{ifluatex}
..
\ifluatex
\usepackage{fontspec}
\setmainfont{...}
\usepackage{microtype}
\else
\usepackage[T1]{fontenc}
\usepackage[utf8x]{inputenc}
\fi
```

$\Longrightarrow$ Umstellen im Editor ermöglichst schnelles Überprüfen des Dokuments
aber: alle Vorteile von LuaTeX (für diesen Lauf) verloren!

Umgebungen

```
\begin{umgebung}[opt. Argumente]{evtl. Argumente}
   .
   .
\end{umgebung}
```

- Jede Umgebung ist eine Gruppierung (wie {})
  ⟹ alle Einstellungen innerhalb einer Umgebung sind lokal

Umgebungen

```
\begin{umgebung}[opt. Argumente]{evtl. Argumente}
   .
   .
\end{umgebung}
```

- Jede Umgebung ist eine Gruppierung (wie `{}`)
  $\Longrightarrow$ alle Einstellungen innerhalb einer Umgebung sind lokal

- am Anfang und am Ende werden Befehle ausgeführt:

$\Longrightarrow$ Abstände (horizontal, vertikal), Schriftumstellungen, Setzen von Zählern

- Argumente beeinflussen das Aussehen der Umgebung
  (Anzahl/Ausrichtung von Tabellenspalten, Anordnung von Gleitobjekten, Breiten von Objekten, etc.)

Umgebungen: einfache Listen

```
\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 der Dokumentenklasse bestimmt!

Wichtige Standardumgebungen

Dokument, Abstract	`document`, `abstract`
Aufzählungen	`itemize`
Nummerierungen	`enumerate`
wörtliche Wiedergabe	`verbatim(*)`
zweispaltiger Satz	`twocolumn`
Zitate	`quotation`
zentiert	`centering`
abgeschlossene Einheit	`minipage`
Tabellen	`tabular` u. ä. (⇒ Vorlesung Tabellen)
Gleitumgebung Tabellen	`table` (⇒ Vorlesung Tabellen)
Gleitumgebung Bilder	`figure` (⇒ Vorlesung Abbildungen)
Beamerfolie	`frame` (⇒ Vorlesung Präsentationen)
Gleichungen	`align` (⇒ Vorlesung Mathe)
Matrizen	`matrix` (⇒ Vorlesung Mathe)

⇒ see `.TeXworks/completion/tw-latex.txt` for a very uncomplete list

Neue Umgebungen

- `xparse` bietet Definition von Umgebungen ganz analog zu Makrodefinitionen
  (Umgebungen *sind* lediglich zwei Makros am Anfang und Ende)
- Definition mittels

```
\NewDocumentEnvironment{name}{argument specifier}
  {start code}
  {end code}
```

- Sowohl `{start code}` als auch `{end code}` kann auf die Argumente zugreifen, mittels `#1, #2 ...`.
  (*nicht* möglich mit LaTeX 2$_\varepsilon$ `\newenvironment`!)

Nützliche Pakete

- Pakete definieren neue Befehle und Umgebungen
- können auch wichtige Dokumenteigenschaften ändern
- jedes Paket hat (mehr oder weniger) hilfreiche Dokumentation
- `texdoc paketname`
- bei Fragen zunächst dort nachsehen, meist ist Inhaltsverzeichnis ausreichend, um alles zu finden
- ansonsten: im pdf suchen!

Nützliche Pakete (unvollständige Liste)

`fontspec, microtype`	(Schriften und Feinheiten)
`babel, blindtext`	(Vielsprachigkeit, Blindtext)
`geometry, hyperref`	(Satzspiegel, Hyperlinks)
`scrpage2, fancyhdr, …`	(Kopf- und Fußzeilen)
`xparse, xspace`	(Befehlsdefinitionen)
`graphicx, xcolor`	(Abbildungen, Farben)
`biblatex`	(Bibliographien)
`paralist`	(Listenumgebungen)
`amsmath, siunitx`	(Mathe- und Einheitensatz)
`unicode-math`	(OpenType-Schriften im Mathesatz)
`tikz, ps-tricks`	(Diagramme, Zeichnungen, ...)
`luacode, luatexbase`	(einfachere Eingabe von Lua Code)

Listung aller Pakete:

`www.ctan.org/tex-archive`

⇒ macros ⇒ latex ⇒ contrib

Was tun, wenn LaTeX anhält?

1. Ruhe bewahren (Dateien können nicht beschädigt werden!)
2. überlegen, was man gerade geändert hat
3. Fehlermeldung lesen

Was tun, wenn LaTeX anhält?

1. Ruhe bewahren (Dateien können nicht beschädigt werden!)
2. überlegen, was man gerade geändert hat
3. Fehlermeldung lesen
4. Fehlermeldung *lesen*!
   (Im Editor evtl. Einstellungen so anpassen, dass Fehler korrekt angezeigt werden.)
5. (offensichtliche) Schreibfehler korrigieren
6. stückweise rückbauen, bis es wieder funktioniert
   (mittes `Strg + z`)
7. wieder vorbauen, dann den Fehler korrigieren

Was tun, wenn LaTeX anhält?

1. Ruhe bewahren (Dateien können nicht beschädigt werden!)
2. überlegen, was man gerade geändert hat
3. Fehlermeldung lesen
4. Fehlermeldung *lesen*!
   (Im Editor evtl. Einstellungen so anpassen, dass Fehler korrekt angezeigt werden.)
5. (offensichtliche) Schreibfehler korrigieren
6. stückweise rückbauen, bis es wieder funktioniert
   (mittes `Strg + z`)
7. wieder vorbauen, dann den Fehler korrigieren
8. Internetsuche nach dem Fehler
9. log-Datei lesen
10. nicht überfliegen, sodern lesen (von hinten)
11. wenn alles nicht hilft: Frage in Mailinglisten, Foren, … (s. u.)

Anhalten bei Fehlern

- manche Editoren kompilieren im nonstop-mode
- Fehler werden ausgegeben, aber ignoriert.
- „Ich habe 100 Fehler in meinem Dokument“ ist keine sinnvolle Aussage

Anhalten bei Fehlern

- manche Editoren kompilieren im nonstop-mode
- Fehler werden ausgegeben, aber ignoriert.
- „Ich habe 100 Fehler in meinem Dokument“ ist keine sinnvolle Aussage
- „Ich habe nur noch fünf Fehler!“ ist ebenfalls sinnlos:
- sobald ein Fehler auftritt, ist das Dokument meist wertlos.

Fehlerausgabe

Typische Fehlermeldung: (mit `pdflatex` oder `xelatex`)

```
! Undefined control sequence.
l.3 Ein \Latex
                -Dokument.
?
```

$\Longrightarrow$ Befehl in Zeile 3 falsch geschrieben.

Fehlerausgabe

Typische Fehlermeldung: (mit `lualatex -file-line-error`)

```
./test.tex:3: Undefined control sequence.
l.3 Ein \Latex
                -Dokument.
?
```

$\Longrightarrow$ Befehl in Zeile 3 falsch geschrieben.

Komplexere Fehlermeldungen

```
./03-Allgemeine Formatierung und Pakete.tex:401: LaTeX
Error: \begin{itemize} o
n input line 401 ended by \end{beamer@framepauses}.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.401 \end{frame}

?
```

$\Longrightarrow$ Eindeutige, verständliche Meldung

Komplexere Fehlermeldungen

```
./03-Allgemeine Formatierung und Pakete.tex:401: LaTeX
Error: \begin{itemize} o
n input line 401 ended by \end{beamer@framepauses}.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...

l.401 \end{frame}

?
```

$\Rightarrow$ Eindeutige, verständliche Meldung ?!

Dank LuaLaTeX wird immerhin Zeile und Datei angegeben!

Erstellen eines Minimalbeispiels

- falls man den Fehler nicht finden/verstehen kann: Minimalbeispiel erstellen!

Erstellen eines Minimalbeispiels

- falls man den Fehler nicht finden/verstehen kann:
  Minimalbeispiel erstellen!
- allen Code wegnehmen, der nichts mit dem Problem zu tun hat
- allen unnützen Text wegnehmen.
  Wenn langer Text nötig ist, Paket `blindtext` verwenden
- alle Pakete entfernen, die nichts mit dem Problem zu tun haben.
- falls die Klasse für das Problem unerheblich ist:
  `\documentclass{minimal}`
- allen Code wegnehmen, der nichts mit dem Problem zu tun hat!

$\Rightarrow$ meist löst sich das Problem dann schon …

Sinn des Minimalbeispiels

- eigenständiges Finden von Fehlern durch Codereduktion (auch möglich mittels `\iffalse` – `\fi`-Konstrukten oder Auskommentieren

- optimal für Fragen:
- erspart den Kampf durch unnötigen Code
- minimales Dokument, das den Fehler noch produziert

$\Rightarrow$ für die meisten Übungsabgaben gefordert