Tabellensatz mit LaTeX

Dr. Klaus Höppner

Herbsttagung und 37. MV von DANTE

Grundlegendes Beispiel

```
\begin{tabular}{lcr}
a   & b   & c\\
aa  & ab  & ac\\
aaa & aab & aac
\end{tabular}
```

a	b	c
aa	ab	ac
aaa	aab	aac

- *Zellen* getrennt durch `&`
- *Zeilen*-Umbruch durch `\\` (besser: `\tabularnewline`)
- Spaltenausrichtung:
  `l`: left, `c`: centered, `r`: right
  Sonderfall: `p{width}` für einen Absatz als Zelle
- `@{...}` zum Definieren eigener Spaltentrenner.

Zusatzpakete

array	neue Spalten-Definitionen `m{width}` und `b{width}` analog zu `p`, jedoch vertikal zentriert bzw. unten bündig; Definition von Befehlen, die vor oder nach jeder Zelle ausgeführt werden.
tabularx	setzt eine Tabelle in festgelegter Breite durch neuen Spaltentyp `X`, der einer `p`-Spalte mit automatisch festgelegter Breite entspricht.
booktabs	für schöne Linien in Tabellen.
colortbl	für farbige Tabellen.
dcolumn	zum Setzen von Zahlenkolonnen mit Ausrichtung am Dezimalpunkt/-komma.

Globale Kommandos für alle Zellen einer Spalte

Beispiel: Eine Spalte in Schreibmaschinenschrift, eine im Mathe-Modus

```
\begin{tabular}{>{\ttfamily\textbackslash}c>{$}c<{$}}
\multicolumn{1}{c}{Befehl} & \multicolumn{1}{c}{Symbol}\\
\hline
alpha & \alpha\\
beta & \beta\\
gamma & \gamma
\end{tabular}
```

Verwendung von `\multicolumn`, damit Zellen in Titelzeile normal erscheinen.

Ausgabe des Beispiels

Befehl	Symbol
\alpha	α
\beta	β
\gamma	γ

Automatische Berechnung der Spaltenbreite

Aufgabe: Folgende Tabelle soll über die gesamte Textbreite gesetzt werden, wobei die rechte Spalte den Raum einnimmt, der von der linken übrig gelassen wird.

Label	Text
Eins	Dies ist ein Blindtext ohne Bedeutung, der nicht zum Lesen gedacht ist.
Zwei	Noch mehr Blindtext ohne Bedeutung, der nur Platz verbraucht. Auch nicht zum Lesen gedacht.
Drei	Ein weiterer bedeutungsloser Text. Seine Bestimmung ist reiner Platzverbrauch.
Vier	Ob der Blindtext weiß, dass er nicht gelesen werden soll?

Benutzen von tabularx

Die vorige Tabelle wurde mit dem tabularx-Paket gesetzt.

Dieses setzt Tabellen in festgelegter Breite. Im Gegensatz zu tabular* (dort wird der Spalten*zwischen*raum gedehnt, um die Zielbreite zu erhalten), berechnet tabularx automatisch die Breite einer/mehrerer X-Spalten.

Quellcode für die Tabelle:

```
\begin{tabularx}{\linewidth}{lX}
Label & Text\\
\hline
Eins & Dies ist ein Blindtext ohne Bedeutung, der nicht zum
     Lesen gedacht ist.\\
Zwei & Noch mehr Blindtext ohne Bedeutung, [...] \\
[...]
\end{tabularx}
```

Textausrichtung in schmalen Spalten

In schmalen Spalten ist Verwenden von `\raggedright` sinnvoll (sonst gibt es viele underfull hboxes).
Problem: `\raggedright` definiert `\\` um. Daher führt das Verwenden von `\raggedright` mit `\\` am Ende der Tabellenzeile zu seltsamen Compiler-Fehlern!

Dagegen helfen 3 Möglichkeiten:

- Verwende `\tabularnewline` statt `\\`:

```
\begin{tabularx}{\linewidth}{l>{\raggedright}X}
Label & Text\tabularnewline
```

- Nutze `\arraybackslash`, um `\\` die ursprüngliche Bedeutung zurück zu geben:

```
\begin{tabularx}{\linewidth}%
        {l>{\raggedright\arraybackslash}X}
Label & Text\\
```

- Verwende das Paket `ragged2e`

Ausbalancieren von X-Spalten

Bei mehreren X-Spalten in einer Tabelle, z. B. `\begin{tabularx}{width}{lXX}`, erhalten diese alle die selbe Breite.

Um diese *manuell* auszubalancieren, kann die Länge `\hsize` entsprechend geändert werden, z. B.

```
\begin{tabularx}{\linewidth}%
    {l>{\setlength\hsize{.67\hsize}}X%
      >{\setlength\hsize{1.33\hsize}}X}
```

(Die Gesamtsumme aller `\hsize`s muss wieder die Zahl der X-Spalten ergeben!)

Das Paket `tabulary` leistet eine (grobe) automatische Ausbalancierung der X-Spalten.

Und nun einfach ein paar Linien . . .

. . . mit | für vertikale Linien in der Spalten-Spezifikation und **\hline** für horizontale Linien,

. . . mit folgendem Resultat (die Ausgabe von LaTeX ist immer schön):

Label	Text	Mehr text
Eins	Dies ist etwas Text ohne Bedeutung.	Noch mehr Text ohne Bedeutung, der nicht zum Lesen gedacht ist.
Zwei	Ein weiterer Text ohne Aussage.	Völlig nutzloser Text, den man nicht lesen soll, so genannter Blindtext.

Extra-Abstand zwischen Zeilen

Zwei generelle Möglichkeiten, um zu verhindern, dass der Inhalt von Zellen an die horizontalen Linien stößt:

- Setze vor der Tabelle die Länge `\extrarowheight` auf einen Wert > 0 (wirkt für alle Zeilen).

```
\setlength\extrarowheight{4pt}
\begin{tabular}{....}
```

- Einfügen von unsichtbaren `\rule`s (Breite 0), die höher als der Inhalt der ersten Zeile der Zelle sind, z. B. `\rule{0pt}{18pt}This is {\huge another} text`.

Etwas bessere Version

Nach Setzen von `\extrarowheight` auf 4 pt und Hinzufügen einer `\rule` (zur Illustration hier in rot), erhalten wir:

Label	Text	More text
Eins	Dies ist etwas Text ohne Bedeutung.	Noch mehr Text ohne Bedeutung, der nicht zum Lesen gedacht ist.
Zwei	Ein weiterer Text ohne Aussage.	Völlig nutzloser Text, den man nicht lesen soll, so genannter Blindtext.

Typografische Sicht

Grundregeln zum Verwenden von Linien:

- Keine vertikalen Linien!
- Keine doppelten Linien!

Für schöne Tabellen ist `booktabs` sehr sinnvoll. Es bietet

- verschiedene Arten horizontaler Liniee: `\toprule`, `\midrule` and `\bottomrule`, mit unterschiedlicher Dicke und angepassten Abständen davor und dahinter,
- `\cmidrule` für horizontale Linien über einzelne Spalten,
- `\addlinespace` für zusätzlichen Zeilenabstand, wo es notwendig ist (ersetzt Herumspielen mit `\rule`)

Beispiel

Nun schaut die Tabelle deutlich besser aus (abgesehen von den offensichtlichen mentalen Problemen des Autors):

Label	Text	More text
Eins	Dies ist etwas Text ohne Bedeutung.	Noch mehr Text ohne Bedeutung, der nicht zum Lesen gedacht ist.
Zwei	Ein weiterer Text ohne Aussage.	Völlig nutzloser Text, den man nicht lesen soll, so genannter Blindtext.

Anderes Beispiel

Hier ein realistischers Beispiel aus der booktabs-Anleitung:

Item		
Animal	Description	Price ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

Tabellen über mehrere Seiten

Es existieren zwei grundlegende Pakete für Tabellen, in denen Seitenumbrüche erlaubt sind:

- `supertabular`
- `longtable`

Beide Pakete machen im Wesentlichen das Gleiche (natürlich mit unterschiedlicher Syntax . . .), mit zwei Unterschieden:

- `longtable` hält die Spaltenbreiten über alle Seiten der Tabelle fest, `supertabular` berechnet die Spaltenbreite je Seite neu.
- `longtable` funktioniert nicht mit two- oderr multicolumn.

Generelle Struktur einer longtable

```
\begin{longtable}{ll}
Label (cont.) & Text (cont.)\\
\endhead
\multicolumn{2}{l}{This is the first head}\\
Label & Text\\
\endfirsthead
\multicolumn{2}{l}{to be continued on next page}
\endfoot
\multicolumn{2}{l}{this is the end (finally!)}
\endlastfoot
One & Some content\\
Two & Another content\\
Three & Yet another content\\
[...]
\end{longtable}
```

ltxtable – tabularx und longtable kombiniert

Wenn man mehrseitige Tabellen möchte und tabularx kennt, möchte man wahrscheinlich beides auf einmal.

Das Paket ltxtable kombiniert die longtable-Umgebung mit tabularx. Zum Nutzen von X-Spalten in mehrseitigen Tabellen verwendet man

```
\LTXtable{width}{filename}
```

wobei die Datei (!) filename.tex die longtable enthält, z. B.:

```
\begin{longtable}{lX}
[...]
\end{longtable}
```

Literatur

Die Pakete, die in diesem Vortrag vorgestellt wurden, sind inklusive Dokumentation Teil der meisten TEX-System bzw. von CTAN erhältlich.
Zusätzlich empfiehlt sich für einen umfangreichen Überblick (auch weiterer Pakete): Der LATEX-Begleiter, 2. Aufl. von F. Mittelbach, M. Goossens, et al. (Pearson Education, ISBN 382737166X) bzw. .