

L^AT_EX, beamer, tikz und Co.

94. Benutzung von csv-Dateien

Thomas Worsch

Fakultät für Informatik
Karlsruher Institut für Technologie

Wintersemester 2016/2017

Überblick

Was ist das?

Erzeugung von tabular-Umgebungen

Struktur von csv-Dateien

- ▶ Abkürzung von
 - ▶ **c**omma **s**eparated **v**alues
 - ▶ **c**haracter **s**eparated **v**alues
- ▶ «Tabellen»
 - ▶ jede Tabellenzeile beschrieben durch eine Zeile der Datei
 - ▶ Einträge für die Spalten durch «Komma» getrennt
- ▶ keine allgemein anerkannte Definition
 - ▶ <https://tools.ietf.org/html/rfc4180> (relativ strikt)
 - ▶ «das, was Programm XYZ erzeugt/akzeptiert»

Struktur von csv-Dateien (2)

nicht Konsens:

- ▶ Zeilen mit unterschiedlich vielen Einträgen?
- ▶ erlaubte Separatoren
 - ▶ Komma, Semikolon, Leerzeichen, Strich, Tab?
- ▶ Separator in Tabelleneintrag? wie?
- ▶ Kommentare?
- ▶ optional: erste Zeile Spaltennamen
 - ▶ wie erkennbar?

Überblick

Was ist das?

Erzeugung von tabular-Umgebungen

Paket `csvsimple`

- ▶ `\usepackage{csvsimple}`
- ▶ einfache Methode
 - ▶ `\csvautotabular[⟨options⟩]{⟨csv file name⟩}`
 - ▶ setzt *ganze* csv-Tabelle in `tabular`-Umgebung
- ▶ flexiblere Methode
 - ▶ `\csvreader[⟨options⟩]{⟨csv file name⟩}`
`{⟨assignments⟩}{⟨row template⟩}`
 - ▶ iteriert über alle Zeilen der csv-Datei
 - ▶ Auswahl zu setzender Zeilen
 - ▶ Auswahl zu setzender Spalten

Besonderheiten von csvsimple

aus der `csvsimple`-Dokumentation

- ▶ *every line of the file has to contain an identical amount of comma separated values*
 - ▶ *every line which contains a higher or lower number of entries is ignored during standard processing*
 - ▶ es gibt Optionen um Einfluss zu nehmen
- ▶ *curly braces `{ }` can be used to mask a block which may contain commas not to be processed as separators*
 - ▶ statt *commas* sind hier hoffentlich *separators* gemeint
- ▶ *first line of such a CSV file is by default `[...]` processed as a header line which contains the identifiers for each column*
 - ▶ abschalten mit Option `head=false` bzw. `no head`

Beispiel mit csvsimple: Eingabedatei stadt.csv

```
stadt.csv
```

```
Stadt | Land | Fluss
```

```
%-----
```

```
Schwerte | Schweden | Saale
```

```
Braunschweig | Brunei | Brenta
```

Beispiel mit csvsimple: Eingabedatei stadt.csv

```
stadt.csv
```

```
Stadt | Land | Fluss  
%-----  
Schwerte | Schweden | Saale  
Braunschweig | Brunei | Brenta
```

- ▶ Beobachtung: `csvsimple` «versteht» \LaTeX -Kommentare
 - ▶ nicht explizit dokumentiert
 - ▶ in einem Beispiel verwendet
 - ▶ *If needed, the data file may contain macro code. Note that the first character of a data line is not allowed to be the backslash \backslash .*

Beispiel mit `\csvautotabular`

```
\csvautotabular[separator=pipe]{stadt.csv}
```

Stadt	Land	Fluss
Schwerte	Schweden	Saale
Braunschweig	Brunei	Brenta

Beispiel mit `\csvautobooktabular`

```
\csvautobooktabular[separator=pipe]{stadt.csv}
```

Stadt	Land	Fluss
Schwerte	Schweden	Saale
Braunschweig	Brunei	Brenta

Beispiel (2): Eingabedatei `daten.csv`

`daten.csv`

```
1 , 1  
2 , 4  
3 , 9  
4 , 16
```

Beispiel (2): \csvreader in tabular Umgebung

```
\begin{tabular}{r@{ ist }l}  
  \csvreader[no head,late after line=\\]%  
    {daten.csv}{1=\x,2=\y}%  
    {die Wurzel aus  $\$ \y$$  &  $\$ \x$$  }  
\end{tabular}
```

die Wurzel aus 1 ist 1
die Wurzel aus 4 ist 2
die Wurzel aus 9 ist 3
die Wurzel aus 16 ist 4

Beispiel (2): \csvreader ohne tabular

```
\csvreader[no head,late after line={, },  
          late after first line={,},  
          late after last line={.},  
          before first line=Es ist]%  
{daten.csv}{1=\x,2=\y}%  
{$\sqrt{\y}=\x$}
```

Es ist $\sqrt{1} = 1, \sqrt{4} = 2, \sqrt{9} = 3, \sqrt{16} = 4$.