

01-01-13

| Personen |         | Beruf   |      |
|----------|---------|---------|------|
| Anzahl   | Geschl. | Angest. | Frei |
| 222      | m       | 160     | 62   |
| 2        | n       | 1       | 2    |
| 97       | w       | 70      | 27   |
| Summen   |         |         |      |
| 319      |         | 230     | 89   |
|          |         | 319     |      |

```
\begin{tabular}{@{}|cc|cc|@{}}
\multicolumn{2}{|c|}{\textbf{Personen}} &
\multicolumn{2}{|c|}{\textbf{Beruf}} & \\\hline
Anzahl & Geschl. & Angest. & Frei & \\
222 & m & 160 & 62 & \\
2 & n & 1 & 2 & \\
97 & w & 70 & 27 & \\
\multicolumn{4}{|c|}{Summen} & \\
319 & & 230 & 89 & \\
& & \multicolumn{2}{|c|}{319} & \\
& & & & \\
& & \multicolumn{2}{|c|}{319} & \\
\end{tabular}
```

### 1.1.3 Verschachtelte Tabellen

Tabellen können beliebig ineinander verschachtelt werden, wobei schon ab der zweiten Stufe auf eine sinnvolle Code-Anordnung geachtet werden sollte, denn Verschachtelungen erschweren sehr häufig die Übersicht und führen zu Fehlern, die dann nur mit hohem Zeitaufwand korrigiert werden können. Andererseits erlauben verschachtelte Tabellen faktisch jegliche auch noch so komplizierte Anordnung, ohne auf das `multirow`-Paket (siehe Abschnitt 2.3 auf Seite 28) zurückgreifen zu müssen, welches selbst nicht immer optimale Ergebnisse liefert. Im folgenden Beispiel sind die Zellen mit kleinen Buchstaben jeweils Teil einer geschachtelten Tabelle, die man zudem mit dem `\cline`-Makro einfach einrahmen kann.

01-01-14

|    |    |     |    |    |
|----|----|-----|----|----|
| A1 | A1 | A1  | A1 | A1 |
| A2 | a1 | a2  | A  | a3 |
|    | a1 | a2  |    | a4 |
| A3 | A3 | A3  | A3 | A3 |
| A4 | A4 | A4* |    | A4 |
| A5 | A5 | A5  | A5 | A5 |

```
\begin{tabular}{ccccc}
A1 & A1 & A1 & A1 & A1 \\
A2 & \begin{tabular}{|c|} \hline a1 \\ a1 \\ a1 \\ \hline \end{tabular} & a2 & A & a3 \\
& & a2 & & a4 \\
\end{tabular} &
\begin{tabular}{c} a2 \\ a2 \\ a2 \\ \end{tabular} & &
A &
\begin{tabular}{|c|} \hline a3 \\ a3 \\ a4 \\ \hline \end{tabular} \\
A3 & A3 & A3 & A3 & A3 \\
A4 & A4 & \multicolumn{2}{|c|}{A4*} & A4 \\
A5 & A5 & A5 & A5 & A5 \\
\end{tabular}
```

Das folgende Beispiel zeigt allerdings ein unbefriedigendes Ergebnis, denn die beiden senkrechten Linien liegen nicht übereinander. Der Grund liegt in dem Abstand `\tabcolsep`, der jeder Tabelle vorangestellt wird, was zu einem zusätzlichen Abstand führt, wenn eine Tabelle Teil einer anderen Tabelle ist. Entsprechend Tabelle 1.1 auf Seite 4 ergibt sich in solchen Fällen dann ein Abstand von `3\tabcolsep` oder `4\tabcolsep` zwischen zwei Spalten. Dieser Effekt ist deutlich im folgenden Beispiel zu sehen, wo die Spalten einen deutlich zu großen Abstand zueinander haben.

|        |          |        |
|--------|----------|--------|
| Sp1 Z1 | Sp2 Z1   | Sp3 Z1 |
| Sp1 Z2 | Sp2 Z2/3 | Sp4 Z2 |
| Sp1 Z3 |          | Sp4 Z3 |
| Sp1 Z4 | Sp2 Z4   | Sp3 Z4 |

```
\begin{tabular}{|ccc|}
Sp1 Z1 & Sp2 Z1 & Sp3 Z1\\
\begin{tabular}{c}Sp1 Z2\\Sp1 Z3\end{tabular}
& \begin{tabular}{|c|}\hline
Sp2 Z2/3\\\hline \end{tabular}
& \begin{tabular}{c}Sp4 Z2\\Sp4 Z3\end{tabular}\\
Sp1 Z4 & Sp2 Z4 & Sp3 Z4
\end{tabular}
```

01-01-15

Hier muss in der Regel der Spaltenoperator @ benutzt werden, um diese zusätzlichen Abstände zu korrigieren. In diesem Fall wird bereits in der übergeordneten Tabelle dafür gesorgt, dass alle Spalten (von außen betrachtet) keinen Abstand haben: `{|@{}c @{}c@{} c@{}|}`. Dass sich dennoch zwischen den Spalten `2\tablecolsep` als Abstand ergibt, liegt an den inneren (geschachtelten) Tabellen. Man beachte die Reihenfolge von | und @{}, die besagt, erst die senkrechte Linie, dann kein weiterer Abstand. Im umgekehrten Fall (@{} und |) ist die Bedeutung von @{} wirkungslos, sie wird durch den folgenden senkrechten Strich aufgehoben. Denn diesem folgt in der Regel `\tablecolsep`, wenn nichts anderes vereinbart, wie beispielsweise wieder ein folgendes @{}.

@{} und  
@{} / sind  
unterschied-  
lich

|        |          |        |
|--------|----------|--------|
| Sp1 Z1 | Sp2 Z1   | Sp3 Z1 |
| Sp1 Z2 | Sp2 Z2/3 | Sp4 Z2 |
| Sp1 Z3 |          | Sp4 Z3 |
| Sp1 Z4 | Sp2 Z4   | Sp3 Z4 |

```
\begin{tabular}{|@{}c @{}c@{} c@{}|}
Sp1 Z1 & Sp2 Z1 & Sp3 Z1\\
\begin{tabular}{c} Sp1 Z2\\Sp1 Z3 \end{tabular} &
\begin{tabular}{|c|}\hline
Sp2 Z2/3\\\hline \end{tabular} &
\begin{tabular}{c} Sp4 Z2\\Sp4 Z3 \end{tabular}\\
Sp1 Z4 & Sp2 Z4 & Sp3 Z4
\end{tabular}
```

01-01-16

Je nachdem wie die Tabellen aufgebaut sind und Linien gesetzt werden, kann ein unterschiedliches Vorgehen notwendig sein, beispielsweise innere Tabellen ohne Spaltenabstand zu setzen oder den Rahmen von außen um die innere Tabelle zu setzen, wie im folgenden Beispiel gezeigt wird:

|        |          |        |
|--------|----------|--------|
| Sp1 Z1 | Sp2 Z1   | Sp3 Z1 |
| Sp1 Z2 | Sp2 Z2/3 | Sp4 Z2 |
| Sp1 Z3 |          | Sp4 Z3 |
| Sp1 Z4 | Sp2 Z4   | Sp3 Z4 |

```
\begin{tabular}{@{}c @{}c@{} c@{}}
Sp1 Z1 & Sp2 Z1 & Sp3 Z1\\\cline{2-2}
\begin{tabular}{c} Sp1 Z2\\Sp1 Z3 \end{tabular} &
\begin{tabular}{c} Sp2 Z2/3 \end{tabular} &
\begin{tabular}{|c} Sp4 Z2\\Sp4 Z3 \end{tabular}
\\\cline{2-2}
Sp1 Z4 & Sp2 Z4 & Sp3 Z4
\end{tabular}
```

01-01-17

Weiterhin kann es hilfreich sein, sich aus Gründen der Übersichtlichkeit, Abkürzungen zu definieren, sodass insbesondere bei verschachtelten Tabellen Fehler vermieden werden können. Im folgenden Beispiel wurden die erste Zeile nach oben und die letzte nach unten gestreckt, sodass eine bessere Anordnung des Rahmens die Folge ist. Die Bedeutung von `\rule` kann man [57] entnehmen.

```
\newcommand\Btab{}% sicherstellen, dass es noch nicht existiert
\renewcommand\Btab[2][c]{\tabular[#1]{#2}}
\newcommand\Etab{}% dito
\renewcommand\Etab{\endtabular}
```

01-01-18

|        |          |        |
|--------|----------|--------|
| Sp1 Z1 | Sp2 Z1   | Sp3 Z1 |
| Sp1 Z2 | Sp2 Z2/3 | Sp4 Z2 |
| Sp1 Z3 |          | Sp4 Z3 |
| Sp1 Z4 | Sp2 Z4   | Sp3 Z4 |

```
\newcommand\Btab{}
\renewcommand\Btab[2][c]{\tabular[#1]{#2}}
\newcommand\Etab{}% \renewcommand\Etab{\endtabular}

\Btab{|@{}c @{}c@{} c@{}}|\hline
\rule{0pt}{3ex}% besserer Abstand nach Oben
Sp1 Z1 & Sp2 Z1 & Sp3 Z1\\ \cline{2-2}
\Btab{c|} Sp1 Z2\\ Sp1 Z3 \Etab &
\Btab{c} Sp2 Z2/3\\ \Etab &
\Btab{c} Sp4 Z2\\ Sp4 Z3 \Etab \\ \cline{2-2}
\rule[-1.5ex]{0pt}{2ex}% besserer Abstand nach Unten
Sp1 Z4 & Sp2 Z4 & Sp3 Z4\\ \hline
\Etab
```

Weitere Ausführungen zum Thema »Linien« finden sich unter anderem im:

- ▷ Beispiel 03-03-1 auf Seite 122 für farbige Linien,
- ▷ Abschnitt 2.2 auf Seite 26 für gestrichelte Linien,
- ▷ Abschnitt 2.10 auf Seite 62 für nicht überlappende doppelte (Teil-)Linien.

### 1.1.4 Fußnoten

Der Befehl `\footnote` kann innerhalb einer Tabelle zwar benutzt werden, führt jedoch nicht zum erwarteten Ergebnis; der Fußnotentext fehlt. Es bieten sich zum einen die Pakete an, die Fußnoten in Tabellen unterstützen oder zum anderen die Verwendung der Kombination `\footnotemark` und `\footnotetext`. Letzteres funktioniert faktisch in allen Anwendungsfällen, bei denen der Befehl `\footnote` nicht erwartungsgemäß funktioniert. Zu beachten ist lediglich, dass `\footnotemark` den Fußnotenzähler inkrementiert, sodass dieser wieder zurückgesetzt werden muss, bevor der Fußnotentext mit `\footnotetext` geschrieben wird.

01-01-19

|                |                |                |
|----------------|----------------|----------------|
| L <sup>1</sup> | C              | R              |
| L              | C              | R              |
| L              | C <sup>2</sup> | R              |
| L              | C              | R              |
| L              | C              | R <sup>3</sup> |

```
\begin{center}
\begin{tabular}{lcr}\hline
L\footnotemark & C & R\\
L & C & R\\
L & C\footnotemark & R\\
L & C & R\\
L & C & R\footnotemark\\ \hline
\end{tabular}
\addtocounter{footnote}{-2}
\footnotetext{Linksbündig}
\stepcounter{footnote}\footnotetext{Zentriert}
\stepcounter{footnote}\footnotetext{Rechtsbündig}
\end{center}
Normale Textzeile nach der zentrierten Tabelle
mit normaler Fußnote\footnote{Normale Fußnote}.
```

Normale Textzeile nach der zentrierten Tabelle mit normaler Fußnote<sup>4</sup>.

<sup>1</sup>Linksbündig

<sup>2</sup>Zentriert

<sup>3</sup>Rechtsbündig

<sup>4</sup>Normale Fußnote

## 1.2 Die tabbing-Umgebung

Die `tabbing`-Umgebung erzeugt im eigentlichen Sinne keine Tabelle, ermöglicht aber tabellarische Anordnungen, weshalb sie auch immer wieder im Zusammenhang mit Tabellen aufgeführt wird. Sie hat insbesondere den Vorteil, dass sie sowohl Fußnoten als auch Seitenumbrüche ermöglicht. Der Name der Umgebung weist schon auf die alte Tabulatortaste hin, wie sie schon bei der ersten Schreibmaschinen zur Anwendung kam. Das Prinzip dieser »Tabulatortaste« ist in der `tabbing`-Umgebung erhalten geblieben. Für die Steuerung der Tabulatoren stehen mehrere Kurzbefehle zur Verfügung, die alle in Tabelle 1.2 zusammengestellt sind.

```
\begin{tabbing}
Eine \=Musterzeile \=... \kill \\
Die \>formatierten \>Zeilen... \\
...
\end{tabbing}
```

Tabelle 1.2: Zusammenstellung der Tabulatorbefehle für die `tabbing`-Umgebung.

| Symbol                 | Bedeutung  |
|------------------------|--|
| <code>\=</code>        | Setzt einen Tabstop an der aktuellen Position.   |
| <code>\&gt;</code>     | Springt zum nächsten Tabstop.  |
| <code>\&lt;</code>     | Ermöglicht Text links vom linken Rand zu setzen ohne diesen zu ändern.   |
| <code>\+</code>        | Der linke Rand wird einen Tabstop nach rechts versetzt.  |
| <code>\-</code>        | Der linke Rand wird einen Tabstop nach links versetzt.   |
| <code>\'</code>        | Der vor dem <code>\'</code> -Befehl stehende Text wird rechtsbündig und der folgende Text linksbündig gesetzt. |
| <code>\‘</code>        | Der dem <code>\‘</code> -Befehl folgende Text wird rechtsbündig in der Zeile gesetzt.                          |
| <code>\a</code>        | Zum Setzen von Akzenten innerhalb der <code>tabbing</code> -Umgebung, beispielsweise <code>\a 'e→é</code> .    |
| <code>\kill</code>     | Löscht die aktuelle (Muster-)Zeile und behält die darin definierten Tabstops.                                  |
| <code>\pushtabs</code> | Speichert oder setzt alle aktuellen Tabulatoren, je nach Reihenfolge.  |

Die Musterzeile kann „gekilled“ werden  
 erster Tabstop  
 zweiter Tabstop  
 dritter  
 vierter

```
\usepackage{pst-node}
```


```
\begin{tabbing}
Die \=\rnode{A}{}Musterzeile \=\rnode{B}{}kann
\=\rnode{C}{},,gekilled‘‘ \=\rnode{D}{}werden\\
\>\rnode{a}{}erster Tabstop\\
\>\>\rnode{b}{}zweiter Tabstop\\
\>\>\>\rnode{c}{}dritter\\
\>\>\>\>\rnode{d}{}vierter
\end{tabbing}
\psset{nodesepA=-1ex,nodesepB=1ex}
\ncline{->}{A}{a}\ncline{->}{B}{b}
\ncline{->}{C}{c}\ncline{->}{D}{d}
```

01-02-1

Die sogenannte Musterzeile erlaubt zusammen mit dem `\kill`-Befehl auf einfache Weise eine Festlegung der einzelnen Tabulatoren; `\kill` löscht zwar die Zeile, nicht jedoch die mit `\=` festgelegten Positionen.

01-02-2

|   |                 |   |   |
|---|-----------------|---|---|
|   | erster Tabstop  |   | <code>\begin{tabbing}</code>  |
|   | zweiter Tabstop |   | Die <code>\=Musterzeile \=kann</code>                               |
|   | dritter         |   | <code>\=,,gekillt'' \=werden\kill\\</code>                          |
|   | vierter         |   | <code>\&gt;erster Tabstop\\ \&gt;\&gt;zweiter Tabstop\\</code>      |
| 0 | 1               | 2 | <code>\&gt;\&gt;\&gt;dritter\\ \&gt;\&gt;\&gt;\&gt;vierter\\</code> |
|   |                 | 3 | <code>0 \&gt;1 \&gt;2 \&gt;3 \&gt;4</code>                          |
|   |                 | 4 | <code>\end{tabbing}</code>  |

Ein Problem ergibt sich bei der Anwendung einiger Kurzbefehle, wenn diese gleichzeitig als Akzent definiert sind. In diesen Fällen müssen die Akzente innerhalb einer `tabbing`-Umgebung durch `\a` eingeleitet werden (vergleiche dazu auch das Paket `Tabbing` im Abschnitt 2.29 auf Seite 109). 

01-02-3

|  |              |        |  |
|--|--------------|--------|--|
|  | erstes Café  |        | <code>\begin{tabbing}</code>                                   |
|  | zweites Café |        | Die <code>\=Musterzeile \=kann</code>                          |
|  | Mâitre       |        | <code>\=,,gekillt'' \=werden\kill\\</code>                     |
|  | Crème        | brûlée | <code>\&gt;erstes Caf\a'e\\ \&gt;\&gt;zweites Caf\a'e\\</code> |
|  |              |        | <code>\&gt;\&gt;\&gt;M\a^aitre\\</code>                        |
|  |              |        | <code>\&gt;Cr\a'eme\&gt;\&gt;\&gt;br^\a^ul\a'ee</code>         |
|  |              |        | <code>\end{tabbing}</code>                                     |

Wenn in folgenden Zeilen auch jeweils ein zum aktuellen Tabulator folgender angesprungen werden soll, so kann man den Befehl `\+` anwenden, der den linken Rand auf die folgende Tabulatorposition setzt. Damit ist dann ein `\>` überflüssig geworden, da der aktuelle linke Textrand dann der Tabulatorposition entspricht. Dieser Effekt lässt sich mit `\-` umkehren; der aktuelle linke Rand wird genau einen Tabstop nach links gesetzt. Diese beiden Möglichkeiten bieten sich besonders bei Algorithmen an, wenn man das sonst übliche Paket `listings` nicht verwenden will oder kann.

Ähnlich einer normalen Tabelle werden die einzelnen Elemente zwischen den Tabulatoren in einer Gruppe gesetzt, sodass eine Schriftumschaltung für die ganze `tabbing`-Umgebung grundsätzlich vor der Umgebung zu erfolgen hat. Die Tabulatoren `\=` können nach der Musterzeile nach Belieben ergänzt werden, sie werden jedoch immer in die vorhandene Liste zusätzlich eingefügt; bestehende werden nicht gelöscht.

01-02-4

|  |  |  |   |
|--|--|--|---|
|  | Eine normale Zeile...                            |  | <code>\usepackage[scaled]{luximono}</code>                    |
|  | <code>function fact(n: integer): integer;</code> |  | Eine normale Zeile\ldots \ttfamily\small                      |
|  | <code>begin</code>                               |  | <code>\begin{tabbing}</code>                                  |
|  | <code>if n &gt; 1 then</code>                    |  | <code>function \=\textbf{fact}(n: integer): integer;\\</code> |
|  | <code>fact:= n * fact(n-1)</code>                |  | <code>\&gt; begin \=\+ \\\% linker Rand auf Tabstop</code>    |
|  | <code>else</code>                                |  | <code>\&gt; if \=n &gt; 1 then \+ \\</code>                   |
|  | <code>fact:= 1;</code>                           |  | <code>fact:= n * fact(n-1) \- \\</code>                       |
|  | <code>end;</code>                                |  | <code>else \+ \\</code>                                       |
|  | Eine normale Zeile...                            |  | <code>fact:= 1; \-\-\% 2 Tabs zurueck</code>                  |
|  |  |  | <code>end;</code>   |
|  |  |  | <code>\end{tabbing}\normalfont\normalsize</code>              |
|  |  |  | Eine normale Zeile\ldots                                      |

Die Festlegung der Tabulatoren muss nicht notwendigerweise über einen Mustertext erfolgen, es können auch mithilfe von `\hspace` Abstände festgelegt werden.

```
\usepackage[scaled]{luximono}
```

Eine normale Zeile\ldots

```
\begingroup\small\ttfamily % Schriftumschaltung lokal halten
\begin{tabbing}
\=\hspace{0.25in} \=\hspace{0.25in} \=\hspace{0.25in} \=\hspace{0.25in}\kill
\> <category> \+ \\\
  \> <pattern>WHAT IS A *</pattern> \+ \\\
  \> <template><srail>DEFINE <star index = "1"></srail> \\\
  \> </template> \- \- \\\
\> </category>  \|[0.5\normalbaselineskip]
\> <category> \+ \\\
  \> <pattern>DO YOU KNOW WHAT A * IS</pattern> \+ \\\
\> <template><srail>DEFINE <star index = "1"></srail></template> \- \- \\\
\> </category>
\end{tabbing}
\endgroup
Eine normale Zeile\ldots
```

Eine normale Zeile...

01-02-5

```
<category>
  <pattern>WHAT IS A *</pattern>
  <template><srail>DEFINE <star index = "1"></srail>
  </template>
</category>

<category>
  <pattern>DO YOU KNOW WHAT A * IS</pattern>
  <template><srail>DEFINE <star index = "1"></srail></template>
</category>
```

Eine normale Zeile...

Die beiden Tabulatorbefehle `\'` und `\'` ermöglichen es, Wörter links- oder rechtsbündig zu setzen, wobei sich die Rechtsbündigkeit bei `\'` immer auf die Zeile als Ganzes bezieht; weitere Tabulatoren werden nach dem Befehl `\'` ignoriert. Dagegen bezieht sich der Befehl `\'` auf den aktuellen Tabulator, was deutlich im folgenden Beispiel zu sehen ist. Der Abstand zwischen den beiden rechts- und linksbündigen Zeichenfolgen ist durch `\tabbingsep=5.0pt` festgelegt und kann beliebig geändert werden. Dabei ist darauf zu achten, dass innerhalb der `tabbing`-Umgebung die Längenänderung global vorgenommen werden muss; der Inhalt zwischen zwei Tabstops wird intern lokal gehalten und die Änderung der Länge hätte sonst keine Auswirkung nach außen.