

## Farbe in Tabellen

4.1 colortbl . . . . .	161
4.2 xcolor . . . . .	166
4.3 T <sub>E</sub> Xnisches . . . . .	172

Durch die weiterhin zunehmende Verbreitung von Farbdruckern kommt der Anwendung von Farbe in Tabellen eine große Bedeutung zu, denn dies ist ein weiteres Stilmittel, um Zelleninhalte besser lesbar zu gestalten. Grundsätzlich sollte man gleich das xcolor-Paket laden, da es zum Einen mit der Paketooption table das colortbl-Paket lädt und zum Anderen eine weitaus bessere Unterstützung bei der Farbauswahl bietet.

### 4.1 colortbl

Das Paket colortbl von David Carlisle steht namentlich für »color table« und ermöglicht das Einbringen von Farbe in Tabellen in Form von gefärbten Spalten, Zeilen, Zellen oder Linien. Das Paket color wird dazu benötigt und entsprechend automatisch geladen. Das Paket xcolor erweitert die Eigenschaften dieses Paketes um einige zusätzliche Makros. Weiteres findet man im Abschnitt 4.2 auf Seite 166.

#### 4.1.1 Spalten

Die Festlegung von farbigen Spalten kann nur über die Spaltendefinition der Tabelle erfolgen. Das entsprechende Makro hat folgende Syntax:

<code>\columncolor</code> <i>[Farbmodell]</i> <i>{Farbe}</i> <i>[Überhang Links]</i> <i>[Überhang Rechts]</i>
---

Unter *Farbmodell* werden alle durch das Paket `color` oder `xcolor` definierten Modelle verstanden, beispielsweise `rgb` oder `cmyk`. *Farbe* bezeichnet einen vordefinierten Namen, beispielsweise `red`, oder eine durch entsprechende Zahlenwerte definierte Farbe. Mit den beiden optionalen Angaben zu *Überhang links* und *Überhang rechts* kann eine Verbreiterung der Spalte erreicht werden, die sich jedoch nur auf das Setzen der Farbe bezieht und daher keine Auswirkung auf die eigentliche Breite der Spalte hat. Hierbei ist aber unbedingt zu beachten, dass ohne Angabe eines Überhangs *beide* Werte auf `\tabcolsep` gesetzt werden. Wird nur *ein* optionales Argument angegeben, so bezieht es sich automatisch auf *beide*! So ist die Angabe von `[0pt]` identisch zu `[0pt][0pt]` (vergleiche Beispiel 04-01-2).

Überhang 

Eins	Zwei	Drei
A	B	C

```
\usepackage{colortbl}
\definecolor{gray70}{gray}{0.7}
\definecolor{gray90}{gray}{0.9}

\begin{tabular}{>{\columncolor{gray70}}c
               >{\columncolor{rgb}{1,0,0}}c
               >{\columncolor{gray90}}c}
Eins & Zwei & Drei\\ A    & B    & C
\end{tabular}
```

04-01-1

Dem Beispiel kann entnommen werden, dass die gesamte Breite der jeweiligen Spalte gefärbt wird. Möchte man dies nur auf die tatsächliche Textbreite der Spalte beziehen, so kann der intern definierte Überhang ignoriert werden; dieser ist so voreingestellt, dass links und rechts jeweils `\tabcolsep` zur normalen Breite addiert wird. Nur um dies deutlicher werden zu lassen, sind im folgenden Beispiel die senkrechten Linien zusätzlich angegeben.

Eins	Zwei	Drei
A	B	C

```
\usepackage{colortbl}
\definecolor{gray70}{gray}{0.7}
\definecolor{gray90}{gray}{0.9}

\begin{tabular}{|>{\columncolor{gray70}[0pt]}c|
               >{\columncolor{rgb}{1,0,0}[0pt]}c|
               >{\columncolor{gray90}[0pt]}c|}
Eins & Zwei & Drei\\ A    & B    & C
\end{tabular}
```

04-01-2

Wird eine Tabelle mit `@{}` links- und rechtsbündig gesetzt, so ist dies entsprechend bei den Spalten zu berücksichtigen; einmal darf links (erste Spalte) und einmal darf rechts (letzte Spalte) kein Überhang entstehen.

Eins	Zwei	Drei
A	B	C

```
\usepackage{colortbl}
\definecolor{gray70}{gray}{0.7}
\definecolor{gray90}{gray}{0.9}

\begin{tabular}{@{>{\columncolor{gray70}[0pt][\tabcolsep]}c
               >{\columncolor{rgb}{1,0,0}}c
               >{\columncolor{gray90}[\tabcolsep][0pt]}c@{}}
Eins & Zwei & Drei\\ A    & B    & C
\end{tabular}
```

04-01-3

### 4.1.2 Zeilen

Das Färben von Zeilen erfolgt immer nach dem Färben von Spalten, d. h. dass die Farbfestlegung einer Zeile immer die der jeweiligen Spalte überschreibt. Dies ergibt sich bereits aus der Tatsache, dass farbige Spalten schon im Tabellenkopf festgelegt werden. Einzelne Zellen einer Zeile lassen sich wiederum mit dem `\cellcolor`-Befehl farblich gestalten (siehe Abschnitt 4.1.3 auf der nächsten Seite).

`\rowcolor` [*Farbmodell*] {*Farbe*} [*Überhang Links*] [*Überhang Rechts*]

`\rowcolor` hat dieselbe Syntax wie `\columncolor`, jedoch mit unterschiedlicher Auswirkung. Die optionalen Argumente beziehen sich nicht auf die linke und rechte Seite der Zeile, sondern auf die der Spalten und sind somit für die Zeile prinzipiell unbrauchbar. `\rowcolor` muss immer am Anfang der ersten Spalte der betreffenden Zeile erscheinen und gilt dann auch nur für diese eine Zeile. Streng genommen ist `\rowcolor` einfach ein Aneinanderreihen von einzelnen `\columncolor`- bzw. `\cellcolor`-Befehlen.

04-01-4

Eins	Zwei	Drei
A	B	C
uno	due	tre

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage[table]{xcolor}
```

```
\begin{tabular}{ccc}
\rowcolor[gray]{0.6}Eins & Zwei & Drei\\
\rowcolor[gray]{0.7}A & B & C \\
\rowcolor[gray]{0.9}uno & due & tre
\end{tabular}\\[6pt]
%
\begin{tabular}{|>\columncolor{gray!70}[0pt]c|
>\columncolor[rgb]{1,0,0}[0pt]c|
>\columncolor{gray!90}[0pt]c|}
\rowcolor[gray]{0.6}Eins & Zwei & Drei\\
\rowcolor{cyan}A & B & C \\
uno & due & tre
\end{tabular}
```

04-01-5

Eins	Zwei	Drei
A	B	C
uno	due	tre

Eins	Zwei	Drei
A	B	C
uno	due	tre

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}
```

```
\begin{tabular}{||@{}
>\columncolor[gray]{.7}[0pt][\tabcolsep]ccc@{||}
\rowcolor[gray]{0.6}Eins & Zwei & Drei\\
\rowcolor{cyan}A & B & C\\ uno & due & tre
\end{tabular}\\[8pt]
\begin{tabular}{||@{}
>\columncolor[gray]{.7}[0pt][\tabcolsep]ccc@{||}
\rowcolor[gray]{0.6}[0pt]Eins & Zwei & Drei\\
\rowcolor{cyan}[0pt]A & B & C\\ uno & due & tre
\end{tabular}\\[8pt]
\begin{tabular}{||@{}
>\columncolor[gray]{.7}[0pt][\tabcolsep]cc
>\columncolor{white}[\tabcolsep][0pt]c@{||}
\rowcolor[gray]{0.6}Eins & Zwei & Drei\\
\rowcolor{cyan}A & B & C\\ uno & due & tre
\end{tabular}
```

Das Beispiel 04-01-5 auf der vorherigen Seite zeigt in der ersten Tabelle, dass `\rowcolor` zwar zur einem richtigen linken, jedoch falschen rechten Rand führt. Es berücksichtigt nicht den abschließenden @-Operator, sondern nimmt einfach den internen Wert für den rechten Überhang, der standardmäßig auf `\tabcolsep` gesetzt ist. Benutzt man die optionalen Argumente für `\rowcolor` so hat dies wiederum auch Auswirkungen auf die einzelnen Spalten. Erst die dritte Tabelle zeigt das erwartete Ergebnis; mit einer weißen Spaltenfarbe und den optionalen Argumenten `[\tabcolsep][Opt]` kann man für den korrekten Wert in der letzten Spalte sorgen, wobei `\rowcolor` ohne optionales Argument benutzt wird. Gleiches müsste man auch für die erste Spalte vornehmen, wenn dort keine Färbung vorhanden wäre.

### 4.1.3 Zellen

In der Reihenfolge der Farbsetzung erscheint die Zelle an letzter Stelle, diese Festlegung überschreibt daher Farbfestlegungen für Spalte und Zeile. Prinzipiell ist der Befehl `\cellcolor` nur eine Abkürzung für die `\multicolumn`-Anweisung, die den aktuellen Spaltentyp übernimmt und nur über `\rowcolor` eine Farbe setzt.

`\cellcolor` *[Farbmodell]* *{Farbe}*

Im Gegensatz zu den anderen Makros für Spalten und Zeilen hat `\cellcolor` keine optionalen Argumente für den Überhang, sondern übernimmt die aktuellen Vorgaben. In der ersten Spalte muss `\cellcolor` grundsätzlich nach einem eventuellen `\rowcolor` auftreten.

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}

\begin{tabular}{ccc}
\rowcolor[gray]{0.6}Eins & Zwei & Drei\\
\rowcolor[gray]{0.7}A & & \cellcolor{white}B & C \\
\rowcolor[gray]{0.9}\cellcolor{red}uno & due & \cellcolor{white}tre
\end{tabular}
```

04-01-6

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}
\definecolor{gray70}{gray}{0.7}
\definecolor{gray90}{gray}{0.9}

\begin{tabular}{>{\columncolor{gray70}[0pt]}c|
>{\columncolor{rgb}{1,0,0}[0pt]}c|
>{\columncolor{gray90}[0pt]}c|}
\rowcolor[gray]{0.6}Eins & Zwei & \cellcolor{white}Drei\\
\rowcolor{cyan} A & & \cellcolor{white}B & C \\
\cellcolor{white}uno & due & & tre
\end{tabular}
```

04-01-7

04-01-8

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}
\definecolor{gray70}{gray}{0.7}
\definecolor{gray90}{gray}{0.9}

\begin{tabular}{@{}
>{\columncolor{gray70}[0pt][\tabcolsep]}c
>{\columncolor{rgb}{1,0,0}}c
>{\columncolor{gray90}[\tabcolsep][0pt]c@{}}
\cellcolor{white}Eins & Zwei & Drei\\
\rowcolor{cyan} A & \cellcolor{white}B & C \\
\rowcolor[gray]{0.9}uno & due & \cellcolor{white}tre
\end{tabular}
```

4.1.4 Linien

Am einfachsten sind farbige vertikale Linien zu erreichen, da der übliche |-Operator einfach durch `!\color{Farbe}\vline` ersetzt wird, wobei für *Farbe* ein entsprechender Wert einzusetzen ist.

04-01-9

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}

\begin{tabular}{!{\color[gray]{0.2}\vline}c
!{\color[gray]{0.4}\vline}c!{\color[gray]{0.6}\vline}c
!{\color[gray]{0.7}\vline}!{\color[gray]{0.9}\vline}}
Eins & Zwei & Drei\\
A & B & C \\
uno & due & tre
\end{tabular}
```

Bei doppelten Linien bleibt allerdings der Zwischenraum weiß, was allerdings keinen Nachteil darstellt, denn man kann auch gleich eine dickere Linie wählen, die eine Breite von `2\arrayrulewidth+\doublerulesep` hat.

04-01-10

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}
\newlength{Dicke}\setlength{Dicke}{2\arrayrulewidth}
\addtolength{Dicke}{\doublerulesep}

\begin{tabular}{
!{\color[gray]{0.2}\vline}!{\color[gray]{0.2}\vline}c
!{\color[gray]{0.2}\vrule width \Dicke}c
!{\color{red}\vline}!{\color{red}\vline}c
!{\color{cyan}\vrule width \Dicke}}
Eins & Zwei & Drei\\
A & B & C \\
uno & due & tre
\end{tabular}
```

Für alle Linientypen stellt colortbl die beiden Makros

```
\arrayrulecolor [Farbmodell] {Farbe}
\doublerulesepcolor [Farbmodell] {Farbe}
```

zur Verfügung. Diese setzen die Farben global, sodass sie an beliebiger Stelle erscheinen können. Logischerweise können sie nicht rückwirkend wirksam sein; werden sie nach dem Tabellenkopf (Spaltendefinition) gesetzt, können sie nicht

mehr die Farben der senkrechten Linien ändern und die der horizontalen nur danach.

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}
\setlength\arrayrulewidth{2pt}
\arrayrulecolor{red}\doublerulesepcolor{blue}

\begin{tabular}{||c|||c|||c||}\hline\hline
Eins & Zwei & Drei\\\hline\arrayrulecolor[gray]{0.9}
A & B & C \\\hline\hline
uno & due & tre \\\hline\hline
\end{tabular}
```

04-01-11

Eins	Zwei	Drei
A	B	C
uno	due	tre

```
\usepackage{colortbl}
\setlength\arrayrulewidth{2pt}
\arrayrulecolor{red}\doublerulesepcolor[gray]{0.6}

\begin{tabular}{||c|||c|||c||}\hline\hline
Eins & Zwei & Drei\\\cline{1-2}
& \arrayrulecolor[gray]{0.9}
A & B & C \\\cline{1-1}\cline{3-3}
& \arrayrulecolor{black}
uno & due & tre \\\cline{2-2}
\end{tabular}
```

04-01-12

Obiges Beispiel zeigt deutlich, dass die Eckverbindungen insbesondere bei Anwendung von `\cline` nicht immer optimal sind. Die Anwendung des Paketes `hhline` ist hier empfehlenswert.

## 4.2 xcolor

Zwischen den Paketen `xcolor` von Uwe Kern und `color` von David Carlisle existiert ein wesentlicher Unterschied bei der Handhabung der Option `dvips`, denn `color` aktiviert grundsätzlich die Option `dvipsnames`, sobald einer der Treiber `dvips`, `oztex` oder `xdvi` ausgewählt wird. Dies kann zu Problemen führen, wenn man das Dokument mit `pdftex` übersetzt, welches dann undefinierte Farben reklamiert. Dies ist der Grund warum `xcolor` grundsätzlich vom Anwender eine explizite Angabe der Option `dvipsnames` erwartet, wenn die entsprechenden vordefinierten Farben benutzt werden sollen, beispielsweise:

```
\usepackage[dvipsnames,prologue]{xcolor}
```

Im Abschnitt 4.1 auf Seite 161 wurde bereits eingehend das Paket `colortbl` behandelt, welches auch automatisch von `xcolor` bei vorhandener Paketoption `table` (siehe Tabelle 4.1 auf der nächsten Seite) geladen wird.

Tabelle 4.1: Zusammenstellung der Paketooptionen von xcolor

Option	Bedeutung
natural	(Standard) Benutze alle Farben innerhalb ihres Modells, mit Ausnahme von RGB (konvertiert nach rgb), HSB (konvertiert nach hsb), und Gray (konvertiert nach gray).
rgb	Konvertiert alle Farben in das rgb-Modell.
cmy	Konvertiert alle Farben in das cmy-Modell.
cmk	Konvertiert alle Farben in das cmk-Modell.
hsb	Konvertiert alle Farben in das hsb-Modell.
gray	Konvertiert alle Farben in das gray-Modell.
RGB	Konvertiert alle Farben in das RGB-Modell und danach ins rgb-Modell.
HTML	Konvertiert alle Farben in das HTML-Modell und danach ins rgb-Modell.
HSB	Konvertiert alle Farben in das HSB-Modell und danach ins hsb-Modell.
Gray	Konvertiert alle Farben in das Gray-Modell und danach ins gray-Modell.
dvipsnames	Lädt die vordefinierten dvips-Farben.
svgnames	Lädt die vordefinierten SVG-Farben.
prologue	Schreibt die Liste der Farbnamen (dvipsnames) in den PS-Header, was für die Dokumentenerstellung via dvips wichtig ist.
table	Lädt das colortbl Paket, um farbige Tabellenzeilen zu ermöglichen.
hyperref	Unterstützung von hyperref.
showerrors	(Standard) Gibt eine Meldung bei undefinierten Farben aus.
hideerrors	Gibt nur eine Fehlermeldung aus, wenn eine undefinierte Farbe benutzt wird und setzt sie dann auf Schwarz.

xcolor erweitert die Möglichkeiten von colortbl insbesondere im Hinblick auf das alternierende Färben von Tabellenzeilen. Der andere Vorteil betrifft vorrangig das Festlegen von Farben, welches aber prinzipiell unabhängig von Tabellen ist und auch für andere Anwendungsfälle gilt. Die Farbmodelle, die von xcolor in der aktuellen Version unterstützt werden, sind in Tabelle 4.2 auf der nächsten Seite zusammengefasst.

Prinzipiell macht es keinen Unterschied, ob man die einzelnen Zellen einer Tabelle oder einer Matrix farbige hinterlegen will.

04-02-1

$$\underline{A} = \begin{pmatrix} A & B & C \\ A & BBB & C \\ A & B & C \end{pmatrix}$$

```
\usepackage[table]{xcolor}

\[\underline{A} =
\left(\begin{array}{c>\columncolor{magenta}}cc
A & B & C\\
\rowcolor{cyan} A & \cellcolor{white} BBB & C\\
A & B & C
\end{array}\right)\]
```

Mit den Makros \columncolor, \rowcolor und \cellcolor kann jede individuelle Kombination hergestellt werden. Alternativ zu \cellcolor kann auch eine Zelle mit \multicolumn definiert werden, für die dann wieder \columncolor möglich ist.

Tabelle 4.2: Unterstützte Farbmodelle ( $L, M, N$  sind natürliche Zahlen)

Name	Grundfarben	Parameterbereich	Standard
rgb	red, green, blue	[0, 1]	
cmY	cyan, magenta, yellow	[0, 1]	
cmYk	cyan, magenta, yellow, black	[0, 1]	
hsb	hue, saturation, brightness	[0, 1]	
gray	gray	[0, 1]	
RGB	Red, Green, Blue	{0, 1, ..., L}	$L = 255$
HTML	RRGGBB	{000000, ..., FFFFFF}	
HSB	Hue, Saturation, Brightness	{0, 1, ..., M}	$M = 240$
Gray	Gray	{0, 1, ..., N}	$N = 15$
wave	lambda (nm)	[363, 814]	

Die Syntax der drei Farbbefehle unterscheidet sich nicht von denen aus `colortbl`, wobei `xcolor` weitere drei Befehle für das alternierende Färben von Zeilen zur Verfügung stellt.

```
\columncolor [Farbmodell] {Farbe} [Überhang Links] [Überhang Rechts]
\rowcolor [Farbmodell] {Farbe} [Überhang Links] [Überhang Rechts]
\cellcolor [Farbmodell] {Farbe}
\rowcolors * [Befehl] {Startzeile}{Farbe-ungerade Zeile}{Farbe-gerade Zeile}
\showrowcolors
\hiderowcolors
```

Das optionale Argument von `\rowcolors *` kann dazu benutzt werden, um die üblichen hier zulässigen Befehle auszuführen, wie `\hline` oder `\noalign{...}`. Der Vorteil des optionalen Arguments ist, dass man die Anwendung dieser Befehle durch die Verwendung der Sternversion für Zeilen unterdrücken kann, für die `\rowcolors` nicht aktiv ist (siehe auch Beispiel 04-02-6 auf Seite 171). Die Argumente für die Farbe der ungeraden und geraden Zeilen kann auch jeweils leer bleiben.

Tabelle 1		
A	BBB	C
A	B	C

Tabelle 2		
A	BBB	C
A	B	C

```
\usepackage[table]{xcolor}

\rowcolors{1}{}{blue!30}
\begin{tabular}{|ccc|}\hline
\multicolumn{3}{|c|}{Tabelle 1}\hline
A & BBB & C\hline
\end{tabular} \par\bigskip
\rowcolors*[\hline]{1}{}{blue!30}
\begin{tabular}{|ccc|}
\multicolumn{3}{|c|}{Tabelle 2}\hline
A & BBB & C\hline
\end{tabular}
```

04-02-2



Das folgende Beispiel zeigt eine Anwendung der einfachen Farbbefehle für Zeilen, Spalten und Zellen, wobei hier wegen des ausschließlichen mathematischen Inhalts der einzelnen Zellen gleich eine array-Umgebung verwendet wird.

```
\usepackage[table]{xcolor}
\definecolor{umbra}{rgb}{0.8,0.8,0.5}
\newcommand*{\zero{\multicolumn{1}{>{\columncolor{white}}c}{0}}
\newcommand*{\colCell[2]{\multicolumn{1}{>{\columncolor{#1}}c}{#2}}

\left[ \begin{array}{*{5}{>{\columncolor[gray]{0.95}}c}}
h_{k,1,0}(n) & h_{k,1,1}(n) & h_{k,1,2}(n) & \zero & \zero \\
h_{k,2,0}(n) & h_{k,2,1}(n) & h_{k,2,2}(n) & \zero & \zero \\
h_{k,3,0}(n) & h_{k,3,1}(n) & h_{k,3,2}(n) & \zero & \zero \\
h_{k,4,0}(n) & \colCell{umbra}{h_{k,4,1}(n)} & h_{k,4,2}(n) & \zero & \zero \\
\zero & h_{k,1,0}(n-1) & h_{k,1,1}(n-1) & h_{k,1,2}(n-1) & \zero \\
\zero & h_{k,2,0}(n-1) & h_{k,2,1}(n-1) & h_{k,2,2}(n-1) & \zero \\
\zero & h_{k,3,0}(n-1) & h_{k,3,1}(n-1) & h_{k,3,2}(n-1) & \zero \\
\zero & \colCell{gray}{h_{k,4,0}(n-1)} & h_{k,4,1}(n-1) & h_{k,4,2}(n-1) & \zero \\
\zero & \zero & h_{k,1,0}(n-2) & h_{k,1,1}(n-2) & h_{k,1,2}(n-2) \\
\rowcolor{gray}{0.75}\zero & \zero & h_{k,2,0}(n-2) & h_{k,2,1}(n-2) & h_{k,2,2}(n-2) \\
\zero & \zero & h_{k,3,0}(n-2) & h_{k,3,1}(n-2) & h_{k,3,2}(n-2) \\
\zero & \zero & h_{k,4,0}(n-2) & h_{k,4,1}(n-2) & h_{k,4,2}(n-2)
\end{array} \right]_{12 \times 5}
```

04-02-3

$h_{k,1,0}(n)$	$h_{k,1,1}(n)$	$h_{k,1,2}(n)$	0	0
$h_{k,2,0}(n)$	$h_{k,2,1}(n)$	$h_{k,2,2}(n)$	0	0
$h_{k,3,0}(n)$	$h_{k,3,1}(n)$	$h_{k,3,2}(n)$	0	0
$h_{k,4,0}(n)$	$h_{k,4,1}(n)$	$h_{k,4,2}(n)$	0	0
0	$h_{k,1,0}(n-1)$	$h_{k,1,1}(n-1)$	$h_{k,1,2}(n-1)$	0
0	$h_{k,2,0}(n-1)$	$h_{k,2,1}(n-1)$	$h_{k,2,2}(n-1)$	0
0	$h_{k,3,0}(n-1)$	$h_{k,3,1}(n-1)$	$h_{k,3,2}(n-1)$	0
0	$h_{k,4,0}(n-1)$	$h_{k,4,1}(n-1)$	$h_{k,4,2}(n-1)$	0
0	0	$h_{k,1,0}(n-2)$	$h_{k,1,1}(n-2)$	$h_{k,1,2}(n-2)$
0	0	$h_{k,2,0}(n-2)$	$h_{k,2,1}(n-2)$	$h_{k,2,2}(n-2)$
0	0	$h_{k,3,0}(n-2)$	$h_{k,3,1}(n-2)$	$h_{k,3,2}(n-2)$
0	0	$h_{k,4,0}(n-2)$	$h_{k,4,1}(n-2)$	$h_{k,4,2}(n-2)$

Für eine alternierende Färbung der Zeilen muss der Befehl `\rowcolors` mit den entsprechenden Argumenten *vor* der Tabelle aufgerufen werden.

```
\usepackage[table]{xcolor}
\definecolor{umbra}{rgb}{0.8,0.8,0.5}
\newcommand*{\zero{\multicolumn{1}{>{\columncolor{white}}c}{0}}

\left[ \begin{array}{*{5}{>{\columncolor[gray]{0.95}}c}}
h_{k,1,0}(n) & h_{k,1,1}(n) & h_{k,1,2}(n) & \zero & \zero \\
h_{k,2,0}(n) & h_{k,2,1}(n) & h_{k,2,2}(n) & \zero & \zero \\
h_{k,3,0}(n) & h_{k,3,1}(n) & h_{k,3,2}(n) & \zero & \zero \\
h_{k,4,0}(n) & h_{k,4,1}(n) & h_{k,4,2}(n) & \zero & \zero \\
\zero & h_{k,1,0}(n-1) & h_{k,1,1}(n-1) & h_{k,1,2}(n-1) & \zero \\
\zero & h_{k,2,0}(n-1) & h_{k,2,1}(n-1) & h_{k,2,2}(n-1) & \zero \\
\zero & h_{k,3,0}(n-1) & h_{k,3,1}(n-1) & h_{k,3,2}(n-1) & \zero \\
\zero & h_{k,4,0}(n-1) & h_{k,4,1}(n-1) & h_{k,4,2}(n-1) & \zero \\
\zero & \zero & h_{k,1,0}(n-2) & h_{k,1,1}(n-2) & h_{k,1,2}(n-2) \\
\zero & \zero & h_{k,2,0}(n-2) & h_{k,2,1}(n-2) & h_{k,2,2}(n-2) \\
\zero & \zero & h_{k,3,0}(n-2) & h_{k,3,1}(n-2) & h_{k,3,2}(n-2)
\end{array} \right]_{12 \times 5}
```

```
\zero & \zero & h_{k,4,0}(n-2) & h_{k,4,1}(n-2) & h_{k,4,2}(n-2)
\end{array} \, , \right]_{12\times 5} \backslash]
```

$h_{k,1,0}(n)$	$h_{k,1,1}(n)$	$h_{k,1,2}(n)$	0	0
$h_{k,2,0}(n)$	$h_{k,2,1}(n)$	$h_{k,2,2}(n)$	0	0
$h_{k,3,0}(n)$	$h_{k,3,1}(n)$	$h_{k,3,2}(n)$	0	0
$h_{k,4,0}(n)$	$h_{k,4,1}(n)$	$h_{k,4,2}(n)$	0	0
0	$h_{k,1,0}(n-1)$	$h_{k,1,1}(n-1)$	$h_{k,1,2}(n-1)$	0
0	$h_{k,2,0}(n-1)$	$h_{k,2,1}(n-1)$	$h_{k,2,2}(n-1)$	0
0	$h_{k,3,0}(n-1)$	$h_{k,3,1}(n-1)$	$h_{k,3,2}(n-1)$	0
0	$h_{k,4,0}(n-1)$	$h_{k,4,1}(n-1)$	$h_{k,4,2}(n-1)$	0
0	0	$h_{k,1,0}(n-2)$	$h_{k,1,1}(n-2)$	$h_{k,1,2}(n-2)$
0	0	$h_{k,2,0}(n-2)$	$h_{k,2,1}(n-2)$	$h_{k,2,2}(n-2)$
0	0	$h_{k,3,0}(n-2)$	$h_{k,3,1}(n-2)$	$h_{k,3,2}(n-2)$
0	0	$h_{k,4,0}(n-2)$	$h_{k,4,1}(n-2)$	$h_{k,4,2}(n-2)$

04-02-4

Die folgenden beiden Beispiele zeigen zum Einen noch einmal die Anwendung des `\rowcolors`-Befehls und zum Anderen die Bedeutung der Sternversion, sowie die Anwendung des Zählers `rownum`, der intern die Nummer der jeweiligen Zeile speichert.

Spalte 1	Zeile 1
Spalte 1	Zeile 2
Spalte 1	Zeile 3
Spalte 1	Zeile 4
Spalte 1	Zeile 5
Spalte 1	Zeile 6
Spalte 1	Zeile 7
Spalte 1	Zeile 8
Spalte 1	Zeile 9
Spalte 1	Zeile 10
Spalte 1	Zeile 11
Spalte 1	Zeile 12
Spalte 1	Zeile 13

```
\usepackage[table]{xcolor}
\newcommand*\Nr{\number\rownum}

\rowcolors[\hline]{3}{green!25}{yellow!50}
\arrayrulecolor{red!75!gray}
\begin{tabular}{ll}
Spalte 1 & Zeile \number\rownum\\
Spalte 1 & Zeile \Nr\\
Spalte 1 & Zeile \Nr\\
\arrayrulecolor{black}
Spalte 1 & Zeile \Nr\\
Spalte 1 & Zeile \Nr\\
\rowcolor{blue!25} Spalte 1 & Zeile \Nr\\
\hiderowcolors% Farbe unterdrücken
Spalte 1 & Zeile \Nr\\
Spalte 1 & Zeile \Nr\\
\showrowcolors% Farbe aktivieren
Spalte 1 & Zeile \Nr\\
\multicolumn{1}{>{\columncolor{red!40}}1}{Spalte 1}
& Zeile \number\rownum\\
\end{tabular}
```

04-02-5

Die Zeilen 8-10 sind durch Anwendung von `\hiderowcolors` und `\showrowcolors` von der Farbgebung ausgenommen. Durch Anwendung des `\multicolumn`-Befehls kann jede Zelle wieder von der normalen Farbgebung ausgenommen werden. Die Farbe der Linien kann über den von `colortbl` bereitgestellten Befehl `\arrayrulecolor` an jeder beliebigen Stelle der Tabelle vorgenommen werden. Aus reinen Platzgründen wurde in den folgenden Beispielen das Makro `\Nr` definiert, wobei die jeweils erste Tabellenzeile zum Vergleich die Ersetzung `\numberrownum` verwendet.

Die Anwendung der Sternversion von `\rowcolors` führt jetzt dazu, dass das optionale Argument nicht beachtet wird, wenn die jeweilige Zeile vor der Startzeile von


`\rowcolors` oder innerhalb von `\hiderowcolors... \showrowcolors` liegt. Ohne die Sternversion wird dieses optionale Argument bei *jeder* Zeile beachtet.

04-02-6

Spalte 1	Zeile 1
Spalte 1	Zeile 2
Spalte 1	Zeile 3
Spalte 1	Zeile 4
Spalte 1	Zeile 5
Spalte 1	Zeile 6
Spalte 1	Zeile 7
Spalte 1	Zeile 8
Spalte 1	Zeile 9
Spalte 1	Zeile 10
Spalte 1	Zeile 11
Spalte 1	Zeile 12
Spalte 1	Zeile 13

```
\usepackage[table]{xcolor}
\newcommand*{\Nr{\number\rownum}}

\rowcolors*[\hline]{3}{green!25}{yellow!50}
\arrayrulecolor{red!75!gray}
\begin{tabular}{ll}
Spalte 1 & Zeile \Nr\rownum\\
Spalte 1 & Zeile \Nr\\
Spalte 1 & Zeile \Nr\\
\arrayrulecolor{black}
Spalte 1 & Zeile \Nr\\
\rowcolor{blue!25}
Spalte 1 & Zeile \Nr\\
\hiderowcolors
Spalte 1 & Zeile \Nr\\
\showrowcolors
Spalte 1 & Zeile \Nr\\
\multicolumn{1}{>{\columncolor{red!40}}1}{Spalte 1}
& Zeile \Nr\\
\end{tabular}
```

Zu beachten ist, dass der Zeilenzähler für `\rowcolors` beim Beenden der Tabelle *nicht* zurückgesetzt wird, sodass folgende Tabellen eine andere alternierende Färbung der Zeilen erhalten. 

04-02-7

11	12	51	52
21	22	11	12
31	32	21	22
41	42	31	32

```
\usepackage[table]{xcolor}

\rowcolors{1}{black!15}{black!5}
\begin{tabular}{cc}
11 & 12 \\
21 & 22 \\
31 & 32 \\
41 & 42
\end{tabular}

\quad

\begin{tabular}{cc}
51 & 52 \\
11 & 12 \\
21 & 22 \\
31 & 32
\end{tabular}
```

Setzt man den Zähler `rownum` nach der ersten Tabelle auf Null zurück, erhält man eine gleiche Färbung.

04-02-8

11	12	51	52
21	22	11	12
31	32	21	22
41	42	31	32

```
\usepackage[table]{xcolor}

\rowcolors{1}{black!15}{black!5}
\begin{tabular}{cc}
11 & 12 \\
21 & 22 \\
31 & 32 \\
41 & 42
\end{tabular}

\quad\rownum=0 % Zurücksetzen
\begin{tabular}{cc}
51 & 52 \\
11 & 12 \\
21 & 22 \\
31 & 32
\end{tabular}
```

Für die Färbung der Zeilen kann auch eine Farbserie verwendet werden, die sich über den Befehl `\definecolorseries` festlegen lässt und dann wieder über den `\rowcolors`-Befehl auf einfache Weise durch das Argument `{CS!+}` inkrementieren lässt. Weitere Informationen dazu findet man in der Beschreibung zum Paket `xcolor`. [31]

1
2
3
4
5
6
7
8
9
10
11
12
13

```
\usepackage[table]{xcolor}
\usepackage{array}
\newcommand*\Nr{\makebox[1cm]{%
  \textcolor{CL!+}{\number\rownum}}}

\definecolorseries{CS}{rgb}{last}{yellow}{blue}
\definecolorseries{CL}{rgb}{last}{blue}{yellow}
\resetcolorseries[13]{CS}
\resetcolorseries[13]{CL}
\rowcolors[\hline]{1}{CS!+}{CS!+}
\begin{tabular}{c}
  \Nr\ \Nr\ \Nr\ \Nr\ \Nr\
  \Nr\ \Nr\ \Nr\ \Nr\ \Nr\
  \Nr\ \Nr\ \Nr
\end{tabular}
```

04-02-9

### 4.3 T<sub>E</sub>Xnisches

Das Paket `colortbl` unterstützt den Befehl `\arrayrulecolor{Farbe}`, der allerdings grundsätzlich global wirkt. Jede Änderung der Linienfarbe wirkt sich daher auf alle folgenden Linienbefehle aus, unabhängig davon, ob `\arrayrulecolor` durch Gruppierung lokal gehalten wird. Im folgenden Beispiel erscheint daher die zweite mit `\hline` gesetzte Linie in der Farbe rot, da die erste `\Chline`-Anweisung die Farbe global ändert.

Ägypten	30.06.1995
Albanien	08.09.2000
Angola	23.11.1996
Argentina	01.01.1995
Antillen	21.01.1996

```
\usepackage[table]{xcolor}
\newcommand\Chline[1]{\arrayrulecolor{#1}\hline}

\begin{tabular}{@{}l@{}}\\ \hline
  Ägypten & 30.06.1995 \\ \Chline{red}
  Albanien & 08.09.2000 \\ \hline
  Angola & 23.11.1996 \\ \Chline{blue}
  Argentina & 01.01.1995 \\ \Chline{red!40}
  Antillen & 21.01.1996 \\ \hline \hline
\end{tabular}
```

04-03-1

Man kann aber einfach das Makro ergänzen und die Farbe wieder auf den Standardwert zurücksetzen, sodass Änderungen nicht mehr global wirksam sind.

04-03-2

marching band	AN	9.80
presentation	AN	8.66
commity	AN	13.31

```
\usepackage[table]{xcolor}
\newcommand\Chline[1]{\arrayrulecolor{#1}%
\hline\arrayrulecolor{black}}
\newcommand\Ccline[2]{\arrayrulecolor{#1}%
\cline{#2}\arrayrulecolor{black}}

\begin{tabular}{l r r} \\ \Chline{blue}
marching band & AN & 9.80 \\ \hline
presentation & AN & 8.66 \\ \Ccline{red}{1-2}
commity & AN & 13.31 \\ \Ccline{blue}{2-3}
\end{tabular}
```

Bei der Anwendung der beamer-Klasse muss dem von beamer standardmäßig geladenen Paket xcolor die Option table entweder über die Klassenoption der Dokumentenklasse oder über das Makro \PassOptionsToPackage (vor dem Laden der Dokumentenklasse) mitgeteilt, beziehungsweise übergeben werden.

04-03-3

```
\documentclass[xcolor=table,12pt]{beamer}
\usetheme{Malmo} \useoutertheme{sidebar} \usecolortheme{dove}
\newcommand\Chline[1]{\arrayrulecolor{#1}\hline\arrayrulecolor{black}}

\begin{frame}{Beispiel}{Farbige Linien}
\begin{center}\Large
\begin{tabular}{l >{\columncolor{red!30}}r r} \\ \Chline{blue}
\rowcolor{magenta!40} \emph{Name} & \emph{Type} & \emph{Value} \\ \Chline{blue}
marching band & AN & 9.80 \\ \Chline{red} presentation & AN & 8.66 \\ \Chline{green}
commity & AN & 13.31 \\ \Chline{blue} food & AN & 11.01 \\ \hline
\end{tabular}
\end{center}
\end{frame}
```

Beispiel  
Farbige Linien

Name	Type	Value
marching band	AN	9.80
presentation	AN	8.66
commity	AN	13.31
food	AN	11.01

\usepackage[table]{xcolor}\usepackage{ragged2e}% Definitionen in der Praeambel -> CTAN!

```
\begin{tabular}{>{\Centering}p{1.75cm}>{\columncolor{cyan}}c>{\columncolor{magenta}}c
>{\columncolor{yellow}}c>{\columncolor{black}}c >{\Centering}p{2.5cm}
>{\columncolor{red}}c>{\columncolor{green}}c>{\columncolor{blue}}c >{\Centering}p{1.75cm}}
CMYK & C & M & Y & \color{white}K & Bezeichnung & R & G & \color{white}B & RGB \\ \hline
\TZeile{RAL1023}{0}{10}{90}{0}{Verkehrsgelb}{100}{90}{10} & \cellcolor{rgb}{1,0.9,0.1} \\
\TZeile{RAL2009}{5}{70}{100}{0}{Verkehrsoranger}{95}{30}{0} & \cellcolor{rgb}{0.95,0.3,0} \\
\TZeile{RAL3020}{0}{100}{100}{10}{Verkehrsrrot}{90}{0}{0} & \cellcolor{rgb}{0.9,0,0} \\
\TZeile{RAL4006}{50}{100}{0}{10}{Verkehrspurpur}{40}{0}{90} & \cellcolor{rgb}{0.4,0,0.9} \\
\TZeile{RAL5017}{100}{20}{5}{40}{Verkehrsblau}{0}{40}{55} & \cellcolor{rgb}{0,0.4,0.55} \\
\TZeile{RAL6024}{90}{10}{80}{10}{Verkehrsgrün}{0}{80}{10} & \cellcolor{rgb}{0,0.8,0.1} \\
\TZeile{RAL7042}{30}{10}{20}{40}{Verkehrsgrau A}{30}{50}{40} & \cellcolor{rgb}{0.3,0.5,0.4} \\
\TZeile{RAL7043}{30}{10}{20}{80}{Verkehrsgrau B}{0}{10}{0} & \cellcolor{rgb}{0,0.1,0} \\
```

```
\TZeile{RAL9016}{3}{0}{0}{0}{Verkehrswei}{97}{100}{100} & \cellcolor{rgb}{0.97,1,1}\\
\TZeile{RAL9017}{100}{90}{100}{95}{Verkehrsschwarz}{0}{0}{0} & \cellcolor{rgb}{0,0,0}
\end{tabular}
```

CMYK	C	M	Y	K	Bezeichnung	R	G	B	RGB	04-03-4
	0	10	90	0	RAL1023 Verkehrsgelb	100	90	10		
	5	70	100	0	RAL2009 Verkehrsorange	95	30	0		
	0	100	100	10	RAL3020 Verkehrsrot	90	0	0		
	50	100	0	10	RAL4006 Verkehrspurpur	40	0	90		
	100	20	5	40	RAL5017 Verkehrsblau	0	40	55		
	90	10	80	10	RAL6024 Verkehrsgrün	0	80	10		
	30	10	20	40	RAL7042 Verkehrsgrau A	30	50	40		
	30	10	20	80	RAL7043 Verkehrsgrau B	0	10	0		
	3	0	0	0	RAL9016 Verkehrsweiß	97	100	100		
	100	90	100	95	RAL9017 Verkehrsschwarz	0	0	0		

Das obige Beispiel stellt exemplarisch den Zusammenhang dar zwischen der RGB- und CMYK-Notation der sogenannten deutschen Verkehrsfarben.

```
\usepackage{array,booktabs,ragged2e}
\usepackage[table]{xcolor} \definecolor{gold}{rgb}{.99,1,.9}
\definecolor{lgrey}{gray}{.95} \definecolor{blue}{rgb}{.92,.97,1}
\newcolumnntype{C}[1]{>{\columncolor{#1}[0pt][\tabcolsep]\Centering$}c<{}}
\newcolumnntype{M}[2]{>{\columncolor{#1}[2\ptabcolsep][0pt]\Centering$}m#2<{}}

\begin{tabular}{@{} C{yellow!40} *{10}{>{\mathsf}m{12pt}<{}}
M{lgrey}{14pt}M{lgrey}{12pt} @{}}
\toprule
\rowcolor{blue!40}\rule[-2pt]{0pt}{2.7ex}%
Tabelle1&2&3&4&5&6&7&8&9&10&Y_t&Z_t\\midrule
0 &\circ&\circ&\circ&\circ&\circ&\circ&\circ&* &\circ&\circ &1 &— \\
1 &\circ&\circ&\circ&\circ&\circ&\circ&\circ&† &* &\circ &1 &1 \\
2 &\circ&\circ&\circ&\circ&* &* &\circ&† &† &\circ &2 &2 \\
3 &\circ&\circ&\circ&\circ&† &* &\circ&† &† &\circ &1 &0 \\
4 &\circ&\circ&\circ&\circ&† &† &\circ&† &† &\circ &0 &0 \\
\end{tabular}
```

Tabelle	1	2	3	4	5	6	7	8	9	10	Y <sub>t</sub>	Z <sub>t</sub>	04-03-5
0	◦	◦	◦	◦	◦	◦	◦	*	◦	◦	1	—	
1	◦	◦	◦	◦	◦	◦	◦	†	*	◦	1	1	
2	◦	◦	◦	◦	*	*	◦	†	†	◦	2	2	
3	◦	◦	◦	◦	†	*	◦	†	†	◦	1	0	
4	◦	◦	◦	◦	†	†	◦	†	†	◦	0	0	

Auch bei Verwendung des booktabs-Paket sind farbige Linien einfach zu realisieren.

04-03-6

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

```
\usepackage[table]{xcolor}\usepackage{booktabs}
\arrayrulecolor{red}

\begin{tabular}{@{}lrr@{}}\toprule
\multicolumn{2}{c}{Item}\\
\cmidrule(r){1-2}\morecmidrules\cmidrule(r){1-2}
Animal & Description & Price (\$)\\
\arrayrulecolor{blue}\cmidrule(r){1-1}
\cmidrule[2pt](l{1em}r{1em}){2-2}\cmidrule(l){3-3}
Gnat & per gram & 13.65 \\
& each & 0.01 \\
Gnu & stuffed & 92.50 \\
Emu & stuffed & 33.33 \\
Armadillo & frozen & 8.99\\
\bottomrule
\end{tabular}
```





## Tipps und Tricks

5.1 Tabelle – Allgemein . . . . .	177
5.2 tabbing-Umgebung . . . . .	180
5.3 Zweispaltenmodus . . . . .	180
5.4 Tabellenüberschriften . . . . .	181
5.5 Mathematischer Modus . . . . .	183
5.6 Excel- und LibreOffice-Dateien . . . . .	184
5.7 Tabelle neben einer Abbildung . . . . .	186

In diesem Kapitel werden verschiedene Tipps und Tricks angegeben, die bislang nicht behandelt wurden oder nicht direkt von einem Paket unterstützt werden. Einige der Tricks wurden den üblichen T<sub>E</sub>X-Mailinglisten oder Newsgroups entnommen. In diesen Fällen ist dann der jeweilige Autor angegeben.

### 5.1 Tabelle – Allgemein

Am Anfang einer Tabellenzeile sucht T<sub>E</sub>X intern nach bestimmten Makros (Primitiven), die die folgende Formatierung beeinflussen, wie beispielsweise `\span`, `\omit` oder `\noalign`. Dies kann zu Problemen führen, wenn für die erste Spalte ein Spaltentyp verwendet wird, der eine explizite Umschaltung der Kodierung vornimmt. In diesen Fällen würde es einen Fehler geben, da T<sub>E</sub>X auf der Suche nach den bestimmten Befehlen das betreffende Makro in der Spalte expandiert, aber danach erst den Spaltentyp beachtet. Im Folgenden von Bernd Raichle angegebenen Beispiel sorgt der `\relax`-Befehl dafür, das T<sub>E</sub>X den anschließenden Befehl `\textepsilon` aus dem `tipa`-Paket nicht expandiert. *Trick*

```
\usepackage[T3,T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{array,dcolumn,tabularx,textcomp,ragged2e}
\usepackage[noenc]{tipa}
\newcolumnmtype{C}{>{\Centering}X}

\begin{tabularx}{0.85\linewidth}{|>{\tipaencoding}c|l|
>{\RaggedRight}X|D{.}{.}{-1}|}\hline
\multicolumn{1}{|c|}{Sound} & \multicolumn{1}{c|}{Examples}
& \multicolumn{1}{C|}{Place and manner of articulation}
& \multicolumn{1}{C|}{Occurence frequency (\%)} \\\hline
\relax\textepsilon & lait, jouet, merci & front, half-open & 5.3 \\\hline
\end{tabularx}
```

Sound	Examples	Place and manner of articulation	Occurence frequency (%)
ε	lait, jouet, merci	front, half-open	5.3

05-01-1

*Tip* Bei leeren oder nicht vollständig gefüllten Zeilen und senkrechten Linien ist darauf zu achten, dass alle betreffenden Spalten aufgeführt werden; ansonsten fehlt ein senkrechter Strich.

foo	bar	baz
foo		
foo	bar	baz

foo	bar	baz
foo		
foo	bar	baz

```
\begin{tabular}{|c|c|c|}\hline
foo & bar & baz\\ foo \\
foo & bar & baz\\\hline
\end{tabular}\par\medskip

\begin{tabular}{|c|c|c|}\hline
foo & bar & baz\\
foo & & \\
foo & bar & baz\\\hline
\end{tabular}\par\medskip
```

05-01-2

*Trick* Bei einer Tabelle kann man am Ende der Zeile durch das optionale Argument von `\\` beliebigen vertikalen Abstand einfügen. Fügt man noch zusätzlich eine horizontale Linie ein, dann führt TeX's Bearbeitungsreihenfolge »zuerst Vorschub, dann Linie« zu einem unbefriedigenden Ergebnis. Der TeX-Befehl `\noalign` erlaubt nach einem Zeilenende das Einfügen von vertikalem Material, sodass zuerst die horizontale Linie gezeichnet und danach ein entsprechender vertikaler Vorschub eingefügt wird.

Agypten	30.06.1995
Albanien	08.09.2000
Angola	23.11.1996
Argentina	01.01.1995
Antillen	21.01.1996

Agypten	& 30.06.1995 \\[10pt]\hline
Albanien	& 08.09.2000 \\
Angola	& 23.11.1996 \\
Argentina	& 01.01.1995 \\
Antillen	& 21.01.1996

05-01-3

*Trick* Eine `longtable` benötigt in der Regel mehrere Durchläufe, da sie selbst Informationen über die aktuelle Breite in die `.aux`-Datei schreibt und wieder einliest. Möchte man innerhalb seines eigenen Textes auf diese Breite zurückgreifen, um beispielsweise die Tabellenüberschrift oder nachfolgenden Text in der Tabellenbreite zu setzen, so kann man den von Heiko Oberdiek angegebenen Trick verwenden. Den entsprechenden Code für das Setzen der Länge `\LongTableWidth` findet man in der hier unsichtbaren Präambel des Beispiels von Heiko Oberdiek.

05-01-4

Tabelle 1: Caption für eine  
longtable .....

Hallo	Welt und etwas mehr
foo	bar

Hier kommt jetzt Text in der  
Breite der longtable

```
\usepackage{longtable}
\newlength\LongtableWidth% Siehe Beispielcode

\begin{longtable}{|l|l|}
\caption{Caption für eine \texttt{longtable}\dotfill}\\
Hallo & Welt und etwas mehr\\ \foo & bar
\end{longtable}
\begin{center}
\begin{minipage}{\LongtableWidth}
Hier kommt jetzt Text in der Breite
der \texttt{longtable}
\end{minipage}
\end{center}
```

Möchte man eine Tabelle über den `\input`-Befehl einlesen, so sollte man diese *Tip* komplett auslagern, denn sonst kann es, wie in folgendem Fall, Probleme geben. Die Sequenz

```
\begin{tabular}{c c}
\input{Tabellenzeilen}
\end{tabular}
```

führt zu einem Fehler, wenn die erste Zeile der externen Datei *Tabellenzeilen* einen `\multicolumn`-Befehl aufweist:

```
! Misplaced \omit.
\multispan ->\omit
                \@multispan
1.1 \multicolumn{2}{c}{A}
                \\
?
```

Dies kann man umgehen, indem auch `\begin{tabular}` und `\end{tabular}` in die Datei geschrieben werden.

05-01-5

A
a b
a b

```
\begin{filecontents*}{Tabelle.tex}
\begin{tabular}{c c}
\multicolumn{2}{c}{A}\\
a & b\\ a & b
\end{tabular}
\end{filecontents*}

\input{Tabelle}
```

## 5.2 tabbing-Umgebung

*Trick* Möchte man innerhalb einer tabbing-Umgebung den Zwischenraum bis zum nächsten Tabulator mit `\dotfill` oder `\hrulefill` auffüllen, so ergibt sich das Problem, dass diese nicht sichtbar sind, da sie in eine horizontale Box natürlicher Breite gesetzt werden. Beide Füllmakros haben aber eine »natürliche« Breite von 0 pt, bleiben somit unsichtbar. Heiko Oberdiek hat einen Vorschlag für ein entsprechendes `\rtab`-Makro gemacht, welches anstelle von `\>` verwendet werden kann. Alternativ kann man das interne Makro `\@rtab` mit `\rtab` überschreiben, sodass dann `\>` sich wie `\rtab` verhält.

blabla    blabla  
bla       blabla

blabla ... blabla  
bla\_\_\_\_\_blabla

blabla ... blabla  
bla\_\_\_\_\_blabla

```
\begin{tabbing}
blablabla \= \kill \\
blabla \dotfill \> blabla\% keine Wirkung!
bla\hrulefill \> blabla    % keine Wirkung!
\end{tabbing}

\begin{tabbing}
blablabla \=\kill\\blabla\dotfill\rtab blabla\bla\hrulefill\rtab blabla
\end{tabbing}% Siehe Präambel im Beispiel fuer \rtab
\makeatletter\let\@rtab\rtab\makeatother

\begin{tabbing}
blablabla \=\kill\\blabla\dotfill\> blabla\bla\hrulefill\>blabla
\end{tabbing}
```

05-02-1

## 5.3 Zweispaltenmodus

*Tip* Die einzige Umgebung, die im `\twocolumn`-Modus spaltenübergreifende Tabellen ermöglicht, ist `supertabular` aus dem gleichnamigen Paket (siehe auch Abschnitt 3.5 auf Seite 144). Bei Anwendung des Paketes `multicol` ist der Zweispaltenmodus nicht mehr möglich. Für diesen Fall bleibt dann nur die Anwendung der tabbing-Umgebung.

twocolumn-Demo									
twocolumn-Modus									
L	Z	R	l	c	r				
l	c	r	l	c	r				
l	c	r	l	c	r				
l	c	r	l	c	r				
l	c	r	l	c	r				
l	c	r							
l	c	r							
l	c	r							

Seite 1

```
\usepackage{supertabular}

\twocolumn[\centering%
\texttt{twocolumn}-Modus\bigskip]
\begin{supertabular}{l c r }\hline
L & Z & R \\\hline
l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ l & c & r \\ \end{supertabular}
```

05-03-1

Die tabbing-Umgebung hat zwar den Nachteil, dass eine horizontale Ausrichtung *Tipp* nur linksbündig erfolgt, aber den Vorteil, dass sie in sowohl im \twocolumn-Modus als auch der multicols-Umgebung erfolgreich angewendet werden kann.

05-03-2

twocolumn-Demo

twocolumn-Modus

L Z R	l c r
l c r	l c r
l c r	l c r
l c r	
l c r	
l c r	
l c r	
l c r	
l c r	
l c r	

Seite 1

```
\twocolumn[\centering%
\texttt{twocolumn}-Modus\bigskip]
\begin{tabbing}
L \=Z \=R \kill
L \>Z \>R \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
\end{tabbing}
```

05-03-3

twocolumn-Demo

multicols-Modus

L Z R	l c r
l c r	l c r
l c r	l c r
l c r	l c r
l c r	l c r
l c r	l c r
l c r	

Normaler einspaltiger Text.

Seite 1

```
\usepackage{multicol}
\subsection*{\texttt{multicols}-Modus}
\begin{multicols}{2}
\begin{tabbing}
L \=Z \=R \kill
L \>Z \>R \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
1 \>c \>r \\\ 1 \>c \>r \\\ 1 \>c \>r \\\
\end{tabbing}
\end{multicols}\par
Normaler einspaltiger Text.
```

5.4 Tabellenüberschriften

In manchen Anwendungsfällen (Tabellen mit möglichem Seitenumbruch) kann es vorkommen, dass zwischen Tabellenüberschrift (Caption) und der folgenden Tabelle ein Seitenumbruch eingefügt wird. Hier ist es Aufgabe des Anwenders

selbst zu testen, ob der auf der Seite noch vorhandene Platz ausreicht, um die Tabellenüberschrift und mindestens drei oder mehr Tabellenzeilen aufzunehmen, bevor ein Seitenumbruch erfolgt. Donald Arseneau hat dafür das Makro `\need` angegeben, welches auf einfache Weise überprüft, ob noch Platz auf der Seite ist und anderenfalls einen Seitenumbruch einfügt.

\need{Länge}

Man kann die Definition der entsprechenden Tabellenumgebung auch erweitern und das `\need`-Makro grundsätzlich integrieren. Die folgenden beiden Beispiele zeigen die ersten beiden Seiten eines Dokuments, wobei im ersten Beispiel das Standardverhalten ohne das `\need`-Makro und danach mit diesem Makro gezeigt wird. Die Größe der notwendigen Länge ist ein Erfahrungswert, wobei man mit `\need{6\normalbaselineskip}` in der Regel das erwünschte Ergebnis erzielen wird; es gibt dann keinen Seitenumbruch zwischen Tabellenüberschrift und Tabellenkörper.

```
\usepackage{supertabular}
\newcommand\demoText{Theoretisch kann man die Definition der Tabellenumgebung
auch erweitern und das \protect\texttt{\textbackslash need}-Makro integrieren.
Das Beispiel zeigt die ersten beiden Seiten eines Dokuments, wobei die erste
Seite einen (ohne \protect\texttt{\textbackslash need}) oder keinen (mit
\protect\texttt{\textbackslash need}) Seitenumbruch aufweist.}

\demoText
\begin{center}
\tablecaption{Überschrift} \tablehead{\hline Spalte1 & Spalte2 \\}
\begin{supertabular}{|c|c|}\hline
  yyy & xxx \\ \hline yyy & xxx \\ \hline yyy & xxx \\ \hline
\end{supertabular}
\end{center}
```

05-04-1

<div>need-Demo</div> <div>Theoretisch kann man die Definition der Tabellenumgebung auch erweitern und das <code>\need</code>-Makro integrieren. Das Beispiel zeigt die ersten beiden Seiten eines Dokuments, wobei die erste Seite einen (ohne <code>\need</code>) oder keinen (mit <code>\need</code>) Seitenumbruch aufweist.</div> <div>Tabelle 1: Überschrift</div> <div>Seite 1</div>	<div>need-Demo</div> <table><tr><th>Spalte1</th><th>Spalte2</th></tr><tr><td>yyy</td><td>xxx</td></tr><tr><td>yyy</td><td>xxx</td></tr><tr><td>yyy</td><td>xxx</td></tr></table> <div>Seite 2</div>	Spalte1	Spalte2	yyy	xxx	yyy	xxx	yyy	xxx
Spalte1	Spalte2								
yyy	xxx								
yyy	xxx								
yyy	xxx								

05-04-2

```
\usepackage{supertabular}
\makeatletter
\newcommand\need[1]{\par \penalty-100 \begingroup % preserve \dimen@
  \dimen@pagegoal \advance\dimen@-\pagetotal % space left
  \ifdim #1>\dimen@ % not enough space left
    % only do \vfil if some space left on page
    \ifdim\dimen@>\z@ \vskip -\pagedepth plus 1fil \fi
    \break
  \fi \endgroup}
\makeatother

\demoText% siehe anderes Beispiel
\need{6\normalbaselineskip}
\begin{center}
\tablecaption{Überschrift}
\tablehead{\hline Spalte1 & Spalte2 \\\}
\begin{supertabular}{|c|c|}\hline
  yyy & xxx \\\hline yyy & xxx \\\hline yyy & xxx \\\hline
\end{supertabular}
\end{center}
```

need-Demo

Theoretisch kann man die Definition der Tabellenumgebung auch erweitern und das `\need-` Makro integrieren. Das Beispiel zeigt die ersten beiden Seiten eines Dokuments, wobei die erste Seite einen (ohne `\need`) oder keinen (mit `\need`) Seitenumbruch aufweist.

Seite 1

need-Demo

Tabelle 1: Überschrift

Spalte1	Spalte2
yyy	xxx
yyy	xxx
yyy	xxx

Seite 2

5.5 Mathematischer Modus

Das Setzen von Zahlenkolonnen mit Ausrichtung am Dezimalpunkt oder Dezimalkomma erfolgt T<sub>E</sub>X-intern durch zwei getrennte Spalten, die das Dezimalzeichen als Spaltentrenner definieren. Da einzelne Tabellenzellen immer als eigene Gruppe gesetzt werden (Definitionen bleiben lokal), hat man das Problem, dass Definitionen vor einer D-Spalte nicht mehr bekannt sind, wenn die Dezimalstellen gesetzt werden. Es muss daher mit einem Trick gearbeitet werden, wenn man sämtliche Ziffern in einer anderen Schriftart setzen möchte. Das folgende Beispiel von Heiko Oberdiek zeigt dies für Ziffern in Sans-Serif.

Test	Test1.0	Test1.2	Test3.1
Test	1	1,22	333,1
Test	2	3,44	444,1

0123456789

```
\usepackage{dcolumn}
\makeatletter \newcolumntype{s}[1]{%
  >{\DC@{.}{\sf\aftergroup\sf,}{#1}\sf}1<{\DC@end}}
\makeatother

\sffamily
\begin{tabular}{ 1 s{1.0} s{1.2} s{3.1} }
  Test & \multicolumn{1}{l}{Test1.0} & &
        \multicolumn{1}{l}{Test1.2} & &
        \multicolumn{1}{l}{Test3.1} \\ \hline
  Test & 1 & 1.22 & 333.1 \\ Test & 2 & 3.44 & 444.1
\end{tabular}\par\medskip
$\mathsf{0123456789}$
```

05-05-1

*Tipp* Wenn in D-Spalten durch den Parameterwert -1 eine Zentrierung am Dezimaltrenner vorgesehen ist, gleichzeitig aber eine Konstante optisch addiert werden soll, so kann man im folgenden Beispiel nicht einfach den Spaltentyp . und den <-Operator benutzen: `\newcolumntype{a}{.<{+6240}}`. Wie dem folgenden Beispiel zu entnehmen ist, wird bei dieser Definition zuerst +6240 angehängt und dann zentriert, womit links ein großer Freiraum entsteht (siehe 2. Spalte).

```
\usepackage{dcolumn}
\makeatletter \newcolumntype{.}{D{.}{.}{-1}} \newcolumntype{a}{.<{+6240}}
\newcolumntype{b}{>{\DC@{.}{.}{-1}}c<{\DC@end+6240}}
\newcolumntype{B}{>{\DC@{.}{.}{-1}}c<{\DC@end+25530}}
\newcommand\interval[1]{\multicolumn{1}{c}{#1}} \makeatother

\begin{tabular}{@{}la..bB@{}}
\interval{S1}&\interval{S2}&\interval{S3}&\interval{S4}&\interval{S5}&\interval{S6}\\
99.0\,\% & 14.40& 438& 5256& 375.60& 182.70\\
99.9\,\% & 1.59& 44 & 526 & 37.60& 18.30\\
99.99\,\% & 0.15& 4.4& 0.53& 3.76 & 1.83
\end{tabular}
```

S1	S2	S3	S4	S5	S6
99.0 %	14.40 + 6240	438	5256	375.60 + 6240	182.70 + 25530
99.9 %	1.59 + 6240	44	526	37.60+6240	18.30+25530
99.99 %	0.15 + 6240	4.4	0.53	3.76+6240	1.83+25530

05-05-2

## 5.6 Excel- und LibreOffice-Dateien

Grundsätzlich kann man von Problemen ausgehen, wenn man Excel- oder LibreOfficeCalc-Dateien in ein L<sup>A</sup>T<sub>E</sub>X-kompatibles Format konvertieren will. Einen Erfolg wird man nur bei Dateien erwarten können, die keine Zellen mit Rechenoperationen aufweisen und daher innerhalb von Excel oder LibreOfficeCalc als Dateien im csv-Format (comma separated values) gespeichert werden können. Derartige Dateien können dann mit dem datatool-Paket im Allgemeinen problemlos eingelesen und weiterverarbeitet werden (siehe dazu Abschnitt 2.9 auf Seite 57). Tabellen mit Rechenoperationen können jedoch innerhalb von Excel oder LibreOfficeCalc durch »copy and paste« und der Option »Werte einfügen« in eine neue Tabelle ohne Rechenoperationen gewandelt werden. Das Paket odsfile (Abschnitt 2.16 auf Seite 81) unterstützt das direkte Einlesen von Tabellenfeldern aus LibreOfficeCalc-Dateien im .ods-Format.



Auf CTAN findet man einen etwas älteren Software-Zusatz von Joachim Marder und George Pearson, der auf Windows-Ebene eine Erweiterung für Excel darstellt, sodass auch Rechenoperationen bei der Ausgabe unterstützt werden. Dieses Software-Tool arbeitet jedoch nur bis einschließlich Office 2007 (<http://www.dante.de/CTAN/support/excel2latex/>). Eine andere Variante ist das Paket `exceltex` von Hans-Peter Doerr, welches eine Kombination aus einem normalen  $\text{\LaTeX}$ -Paket und einem externen Perlprogramm gleichen Namens darstellt. Hier wird die Tabelle nicht vorab konvertiert, sondern innerhalb eines  $\text{\LaTeX}$ -Dokuments eingelesen, wobei der Bereich vorgegeben werden kann. Die eigentliche Konvertierung nimmt dann das Perlprogramm vor, welches grundsätzlich zwischen zwei  $\text{\LaTeX}$ -Läufen aufgerufen werden muss.

*Tipp*

```
latex beispiel
exceltex beispiel
latex beispiel
```

Das Perlprogramm `exceltex` benötigt das Perlmodul `Spreadsheet::ParseExcel`, welches von CPAN <https://metacpan.org/release/Spreadsheet-ParseExcel> heruntergeladen werden kann, falls es nicht installiert ist. Für Debian-basierte Systeme geht auch

```
sudo apt install libspreadsheet-parseexcel-perl
```

Das  $\text{\LaTeX}$ -Paket `exceltex` ist auf sowohl auf  $\text{\TeX}$ Live als auch  $\text{\MiKTeX}$  vorhanden. Das Paket unterstützt das Einlesen einzelner Zellen und mehrerer Zeilen:

```
\incell{xls-Datei!Tabellenblatt!Zelle}
\inctab{xls-Datei!Tabellenblatt!Startzelle!Endzelle}
```

05-06-1

Einlesen einer Zelle: Nadine Haßemer

```
\usepackage{exceltex}

Einlesen einer Zelle:
\incell{data/excel/test.xls!Zeugnisliste!C3}
\incell{data/excel/test.xls!Zeugnisliste!D3}
```

```
\usepackage{exceltex,booktabs}

Einlesen einer Tabelle:\par\setlength\tabcolsep{2pt}
\begin{tabular}{@{} *{16}{l} @{} }\toprule
Vorname & Nachname & Geb. & \multicolumn{13}{c@{}}{F\"acher} \\
& & & & & & & & & & & & & & & \\
& & & & & & & & & & & & & & & & \\
\inctab{data/excel/test.xls!Zeugnisliste!C2!R5}\bottomrule
\end{tabular}
```

05-06-2

Einlesen einer Tabelle:

Vorname	Nachname	Geb.	Fächer													
			D	U	T	E	F	L	G	K	M	P	C	B	S	
Victoria	Hoene	06.08.80				x					x					
Nadine	Haßemer	18.12.78	x				x									
So-Young	Lee	20.07.80									x				x	
Sebastian	Weigmann	12.06.79											x	x		

exceltex erstellt jeweils ein Unterverzeichnis mit den Dateinamen, gefolgt von -excltx, indem die einzelnen Einträge jeweils als Datei mit beispielsweise folgendem Inhalt gespeichert werden: `\textcolor[rgb]{0, 0, 0}{Nadine}`. Weitere Informationen kann man der Paketbeschreibung entnehmen, insbesondere auch der weiteren Funktionsweise des Perlprogramms. Allerdings kann die Dokumentation nicht mit texdoc aufgerufen werden, da ihr Name nicht gleich dem Paketnamen entspricht.

```
\usepackage{exceltex,array,booktabs}

\begin{tabular}{@{} *{8}{>{\ttfamily}r} @{} \toprule
\multicolumn{8}{c}{Kurse} \midrule
\inctab{data/excel/test.xls!Tabelle1!G2!N19} \bottomrule
\end{tabular}
```

05-06-3

Kurse							
DE-1	de-1	DE-2	de-2	DE-3	de-3	DE-4	de-4
Mu-1	mu-1	Mu-2	mu-2	Mu-3	mu-3	Mu-4	mu-4
Ku-1	ku-1	Ku-2	ku-2	Ku-3	ku-3	Ku-4	ku-4
E1-1	e1-1	E1-2	e1-2	E1-3	e1-3	E1-4	e1-4
F2-1	f2-1	F2-2	f2-2	F2-3	f2-3	F2-4	f2-4
L3-1	l3-1	L3-2	l3-2	L3-3	l3-3	L3-4	l3-4
	tü-1		tü-2		tü-3		tü-4
	ds-1		ds-2		ds-3		ds-4
	pw-1		pw-2		pw-3		pw-4
GE-1	ge-1	GE-2	ge-2	GE-3	ge-3	GE-4	ge-4
	phil-1		phil-2		phil-3		phil-4
MA-1	ma-1	MA-2	ma-2	MA-3	ma-3	MA-4	ma-4
PH-1	ph-1	PH-2	ph-2	PH-3	ph-3	PH-4	ph-4
CH-1	ch-1	CH-2	ch-2	CH-3	ch-3	CH-4	ch-4
BI-1	bi-1	BI-2	bi-2	BI-3	bi-3	BI-4	bi-4
	in-1		in-2		in-3		in-4
SP-1	sp-1	SP-2	sp-2	SP-3	sp-3	SP-4	sp-4
EK-1		EK-2		EK-3		EK-4	


## 5.7 Tabelle neben einer Abbildung


Wenn eine Tabelle neben einer Abbildung erscheinen soll, muss die Frage geklärt werden, wie die horizontale Ausrichtung erfolgen soll. Für den Textsatz ist die Abbildung nichts weiter als eine Box, auch wenn sie formal sehr groß sein kann. Eine Tabelle wird beim Textsatz zwar auch wie eine große Box gehandhabt, jedoch unterscheidet sich ihre Ausrichtung erheblich von der einer Grafik. Das folgende Beispiel zeigt den einfachen Fall, dass eine Grafik direkt nach der Tabelle folgt, wobei diese einmal mit und einmal ohne Kopf- und Fußlinie gesetzt wird. Die Kopf- und Fußzeile einer Tabelle sind formal jeweils eine eigene Zeile, wenn auch mit geringer Zeilenhöhe. Dadurch ergeben sich die im nächsten Beispiel dargestellten unterschiedlichen Formatierungen.


```
\usepackage{graphicx}\setlength{\fboxsep}{0pt}

\begin{tabular}{cc}
\emph{Nr.}&\emph{Tier}\\
\hline 1.& Gelse\\
2.& Gemse\\
3.& Stinktief\\
4.& Ameisenbär\\
5.& Gürteltier
\end{tabular}
\begin{tabular}{cc}
\hline
\emph{Nr.}&\emph{Tier}\\
\hline 1.& Gelse\\
2.& Gemse\\
3.& Stinktief\\
4.& Ameisenbär\\
5.& Gürteltier\\
\hline
\end{tabular}
\end{tabular}
\par\bigskip
\begin{tabular}{b}{cc}
\emph{Nr.}&\emph{Tier}\\
\hline 1.& Gelse\\
2.& Gemse\\
3.& Stinktief\\
4.& Ameisenbär\\
5.& Gürteltier
\end{tabular}
\begin{tabular}{b}{cc}
\hline
\emph{Nr.}&\emph{Tier}\\
\hline 1.& Gelse\\
2.& Gemse\\
3.& Stinktief\\
4.& Ameisenbär\\
5.& Gürteltier\\
\hline
\end{tabular}
\end{tabular}
\par\bigskip
\begin{tabular}{t}{cc}
\emph{Nr.}&\emph{Tier}\\
\hline 1.& Gelse\\
2.& Gemse\\
3.& Stinktief\\
4.& Ameisenbär\\
5.& Gürteltier
\end{tabular}
\begin{tabular}{t}{cc}
\hline
\emph{Nr.}&\emph{Tier}\\
\hline 1.& Gelse\\
2.& Gemse\\
3.& Stinktief\\
4.& Ameisenbär\\
5.& Gürteltier\\
\hline
\end{tabular}
\end{tabular}
```

05-07-1

<i>Nr.</i>	<i>Tier</i>		<i>Nr.</i>	<i>Tier</i>	_Text
1.	Gelse		1.	Gelse	
2.	Gemse		2.	Gemse	
3.	Stinktief		3.	Stinktief	
4.	Ameisenbär		4.	Ameisenbär	
5.	Gürteltier		5.	Gürteltier	

<i>Nr.</i>	<i>Tier</i>		<i>Nr.</i>	<i>Tier</i>	_Text
1.	Gelse		1.	Gelse	
2.	Gemse		2.	Gemse	
3.	Stinktief		3.	Stinktief	
4.	Ameisenbär		4.	Ameisenbär	
5.	Gürteltier		5.	Gürteltier	

<i>Nr.</i>	<i>Tier</i>		<i>Nr.</i>	<i>Tier</i>	_Text
1.	Gelse		1.	Gelse	
2.	Gemse		2.	Gemse	
3.	Stinktief		3.	Stinktief	
4.	Ameisenbär		4.	Ameisenbär	
5.	Gürteltier		5.	Gürteltier	


Im ersten Fall ohne Angabe eines Platzierungsparameters wird die Tabelle vertikal zentriert zur Basislinie angeordnet. Die Abbildung bleibt mit ihrer unteren Kante auf der Basislinie. Im zweiten Fall mit dem Platzierungsparameter [b] wird die untere Zeile der Tabelle an der Basislinie ausgerichtet und im letzten Fall mit dem Platzierungsparameter [t] wird die Basislinie der ersten Zeile der Tabelle an der Basislinie ausgerichtet.

Es ist offensichtlich, dass die Grafik mit ihrer Oberkante nicht an der der Tabellen ausgerichtet ist. Dies lässt sich nur mit einer künstlichen ersten Zeile erreichen, was mit der Anweisung `\vspace{0pt}` möglich ist. Dazu muss die Grafik in eine `minipage` gesetzt werden, damit zusammen mit der Abbildung mehrere Zeilen möglich sind.

```
\usepackage{varwidth,graphicx,booktabs}\setlength\fbboxsep{0pt}

\begin{varwidth}[t]{\linewidth} \vspace{0pt}% künstliche Zeile
\begin{tabular}{cc}
\emph{Nr.}&\emph{Tier}\\\midrule
1.&Gelse\\2.&Gemse\\3.&Stinktier\\4.&Ameisenbär\\5.&Gürteltier
\end{tabular}
\end{varwidth}

%
\begin{minipage}[t]{2cm} \vspace{0pt}% künstliche Zeile
\fbbox{\includegraphics[width=2cm]{tiger}}
\end{minipage}
%
\begin{tabular}[t]{cc}\toprule
\emph{Nr.}&\emph{Tier}\\\midrule
1.&Gelse\\2.&Gemse\\3.&Stinktier\\4.&Ameisenbär\\5.&Gürteltier\\
\bottomrule
\end{tabular}
```

<i>Nr.</i>	<i>Tier</i>		<i>Nr.</i>	<i>Tier</i>
1.	Gelse		1.	Gelse
2.	Gemse		2.	Gemse
3.	Stinktier		3.	Stinktier
4.	Ameisenbär		4.	Ameisenbär
5.	Gürteltier		5.	Gürteltier

05-07-2

Das Paket `varwidth` wurde hier benutzt, da eine `minipage` die explizite Vorgabe der Tabellenbreite erforderlich gemacht hätte. Bei `varwidth` braucht dagegen nur die maximal mögliche Breite angegeben zu werden. Intern wird dann die kleinstmögliche Breite berechnet und berücksichtigt.