

# The colortbl package\*

David Carlisle

(Übersetzung: Anne-Katrin Leich & Christine Römer)

20. Februar 2024

## Zusammenfassung

Durch das flexible Setzen farbiger ‘Felder’ ermöglicht dieses Paket die Hintergrundgestaltung definierter Spalten in Tabellen. Dazu werden das `array`- und das `color`-Paket benötigt.

## 1 Einleitung

Das Paket `colortbl` dient dem Einfärben von Tabellen (d. h. der farbigen Gestaltung von Flächen hinter Tabelleneinträgen). Es ähnelt Timothy Van Zandt’s `colortab`-Paket. Die innere Anwendungsstruktur ist zwar anders, `colortab` benutzt aber wie `colortbl` nicht nur Tabellen-Konstruktionen von  $\text{\LaTeX}$ , sondern auch die anderer Formate. `colortbl` basiert also auf  $\text{\LaTeX}$  (und dessen `color`- und `array`-Pakete).

Zum Vergleich zunächst eine einfache Tabelle:

```
\begin{tabular}{|l|c|}
  eins & zwei \\
  drei & vier \\
\end{tabular}
```

eins	zwei
drei	vier

## 2 Der `\columncolor`-Befehl

Die folgenden Beispiele demonstrieren verschiedene Anwendungsmöglichkeiten des durch `colortbl` eingeführten `\columncolor`-Befehls. Die vertikalen Linien, durch `|` definiert, werden bewusst in allen Beispielen verwendet, um die Positionierung der Spalten zu verdeutlichen. Auch wenn Sie vielleicht letztendlich nicht farbige Felder *und* vertikale Linien verwenden möchten.

---

\*1. Übersetzungsversion, letzte Änderung 30. 7. 2022

Der hier genannte `\columncolor`-Befehl sollte (nur) als Argument im Sinne der `>` column-Definition gebraucht werden, um ein farbiges Feld hinter die definierte Spalte zu legen. Er kann in der einleitenden Präambel als Argument von `array-`, `tabular-` und auch in `\multicolumn`-Definitionen eingesetzt werden.

Der elementare Quelltext lautet:

```
\columncolor[<color model>]{<colour>} [<left overhang>][<right overhang>]
```

Das erste Argument (oder die ersten beiden, falls das optionale Argument in Anspruch genommen wird) ist das übliche Argument des `color`-Pakets wie es auch bei `\color` auftritt.

Die letzten beiden Argumente geben an, nach welchem Abstand das Feld hinter dem breitesten Tabelleneintrag endet. Wenn das Argument *rechter Überhang* nicht in Anspruch genommen wird, entspricht es dem Argument *linker Überhang*. Bleiben beide undefiniert, entsprechen sie dem vorgegebenen Wert von `\tabcolsep` (im `tabular`-Paket) oder `\arraycolsep` (im `array`-Paket).

Setzt man die Überhang-Argumente auf 0pt, tritt folgender Effekt ein:

```
|>{\columncolor[gray]{.8}[0pt]}1|
>{\color{white}%
\columncolor[gray]{.2}[0pt]}1|
```

eins	zwei
drei	vier

Der voreingestellte Überhang von `\tabcolsep` produziert folgendes Layout:

```
|>{\columncolor[gray]{.8}}1|
>{\color{white}%
\columncolor[gray]{.2}}1|
```

eins	zwei
drei	vier

Möglicherweise bedarf man einer Definition, die zwischen diesen Extrembeispielen liegt. Ein Betrag von `.5\tabcolsep` sieht dann so aus:

```
|>{\columncolor[gray]{.8} [.5\tabcolsep]}1|
>{\color{white}%
\columncolor[gray]{.2} [.5\tabcolsep]}1|
```

eins	zwei
drei	vier

`colortbl` sollte mit den meisten anderen Paketen kompatibel sein, die mit der Syntax des `array`-Pakets vereinbar sind. Im Einzelfall arbeitet es mit `longtable` und `dcolumn`, wie es das folgende Beispiel zeigt:

3.3 so wie im folgenden verbatim-Text zu verwenden ist am besten, aber dann besteht die Notwendigkeit die Version vom Juni 1996 von `dcolumn` zu benutzen, dann nehme man hier -1.

Ehe die Tabelle beginnt, sollte ein kleiner Leerraum eingefügt werden:

```
\setlength\minrowclearance{2pt}
```

<b>Bsp. für eine lange Tabelle</b>		
die ersten beiden Spalten p-type		die dritte Spalte D-type (dcolumn)
P-column	und eine	12·34
Total	(falsch)	100·6
Ein etwas längerer Text in der ersten Spalte	bbb	1·2
aaa	und etwas mehr Text in der zweiten Spalte	1·345
Total	(falsch)	100·6
aaa	bbb	1·345
Beachten Sie, dass sich die farbigen Linien der Breite der längsten Tabelleneinträge anpassen.	bbb	1·345
aaa	bbb	100

**Fortsetzung folgt...**

Bsp. für eine lange Tabelle (Fortsetzung)		
die ersten beiden Spalten p-type		die dritte Spalte D-type (dcolumn)
aaa	Abhängig vom Treiber entstehen dort, wo sich Felder beeinflussen, unansehnliche Lücken oder Linien. Sie können dann durch die Definition von Überhang-Komponenten angrenzende Flächen derselben Farbe erzeugen oder durch <code>noalign</code> 'negative Felder' zwischen Zeilen einfügen.	12·4
aaa	bbb	45·3
<b>Ende</b>		

Dieses Beispiel zeigt ein wenig ansprechendes Layout, ist jedoch farbenfroh gestaltet. Für den vollständigen Quelltext öffnen sie die Quelldatei `colortbl.dtx`. Die verwendeten Spalten-Typen finden Sie aber auch hier:

```

\newcolumntype{A}{%
  >{\color{white}\columncolor{red} [.5\tabcolsep]%
  \raggedright}%
  p{2cm}}
\newcolumntype{B}{%
  >{\columncolor{blue} [.5\tabcolsep]%
  \color{yellow}\raggedright}
  p{3cm}}
\newcolumntype{C}{%
  >{\columncolor{yellow} [.5\tabcolsep]}%
  D{.}{\cdot}{3.3}}
\newcolumntype{E}{%
  >{\large\bfseries
  \columncolor{cyan} [.5\tabcolsep]}c}
\newcolumntype{F}{%
  >{\color{white}

```

```

\columncolor{magenta} [.5\tabcolsep]}c}
\newcolumntype{G}{%
>{\columncolor[gray]{0.8} [.5\tabcolsep] [\tabcolsep]}1}
\newcolumntype{H}{>{\columncolor[gray]{0.8}}1}
\newcolumntype{I}{%
>{\columncolor[gray]{0.8} [\tabcolsep] [.5\tabcolsep]}%
D{.}{\cdot}{3.3}}

```

### 3 Benutzung der ‘Überhang’-Argumente für `tabular*`

Die bisher aufgeführten Optionen eignen sich für `tabular`, aber wie sieht es mit `tabular*` aus?

In diesem Fall ist die Gestaltung farbiger Felder schwieriger. Die Anwendung des  $\TeX$ -Befehls `\leader`, der zum Einfügen breiterer farbiger Felder dient, ähnelt *glue*. `\tabskip glue`, das bei `tabular*` (und in diesem Fall auch bei `longtable`) zwischen den Spalten eingefügt wird, muss ein ‘wirklicher glue-Befehl’ sein, keine ‘leader-Anweisung’.

Mit einigen Einschränkungen kann aber auch hier die Überhang-Funktion genutzt werden. Beachten Sie nachfolgend die erste Beispieltabelle. Mit `tabular*` kann in der Präambel eine Breite von 3 cm festgelegt werden:

```

\begin{tabular*}{3cm}{%
@{\extracolsep{\fill}}
>{\columncolor[gray]{.8} [0pt] [20mm]}1
>{\columncolor[gray]{.8} [5mm] [0pt]}1
@{}}

```

eins	zwei
drei	vier

Das Feld kann auf 4 cm verbreitert werden, aber fordern Sie Ihr Glück nicht mit einer weiteren Verbreiterung auf 5 cm heraus ...

eins	zwei	eins	zwei
drei	vier	drei	vier

### 4 Der `\rowcolor`-Befehl

Wie demonstriert, kann die Farbe von definierten Zeilen einer Tabelle mit Hilfe des `\multicolumn`-Befehls verändert werden. Besteht Ihre Tabelle hingegen prinzipiell aus *rows*, könnten Sie dies als unvorteilhaft empfinden. Aus diesem Grund wurde der Befehl `\rowcolor` eingeführt<sup>1</sup>.

<sup>1</sup>Zum Teil auf Kosten der Komplexität von `colortbl`.

`\rowcolor` arbeitet mit den gleichen Argument-Strukturen wie `\columncolor`. Der Befehl muss zu *Beginn* der Zeile eingefügt werden. Spart man die optionalen Überhang-Argumente wieder aus, entsprechen diese den Definitionen der `\columncolor`-Befehle der entsprechenden Spalte, bzw. der Definition von `\tabcolsep` (oder `\arraycolsep` im `array`-Paket).

Konkurrieren bei einem Tabelleneintrag eine `\columncolor`-Definition aus der Tabellen-Präambel und eine `\rowcolor`-Festlegung vom Beginn der jeweiligen Zeile miteinander, setzt sich der `\rowcolor`-Befehl durch. Der `\multicolumn`-Befehl darf `>\rowcolor...` enthalten, sodass die voreingestellten Farben der betreffenden Zeile und Spalte aufgehoben werden.

```
\begin{tabular}{|l|c|}
\rowcolor[gray]{.9}
eins & zwei\\
\rowcolor[gray]{.5}
drei & vier
\end{tabular}
```

eins	zwei
drei	vier

## 5 Der `\rowcolors` Befehl

Der `\rowcolors`-Befehl und seine Dokumentation stammen aus dem `xcolor`-Paket von Dr. Uwe Kern.

```
\rowcolors
[commands]{row}{odd-row color}{even-row color}
\rowcolors*
[commands]{row}{odd-row color}{even-row color}
```

Einer dieser Befehle muss ausgeführt werden, *bevor* eine Tabelle gestartet wird. `row` gibt die Nummer der ersten Zeile an, die nach dem Schema `odd-row color` und `even-row color` gefärbt werden soll. Jedes der Farbgargumente kann auch leer gelassen werden (= keine Farbe). In der Version mit Stern werden Befehle `commands` in Zeilen mit inaktivem `rowcolors status` (siehe unten) ignoriert, während in der Version ohne Stern *Befehle* auf jede Zeile der Tabelle angewendet werden. Solche optionalen Befehle können `\hline` oder `\noalign{stuff}` sein.

**`\showrowcolors`, `\hiderowcolors`**

Der `rowcolors status` ist standardmäßig aktiviert (d.h. das Farbschema wird verwendet), er wird durch den Befehl `\hiderowcolors` deaktiviert (d.h. das Farbschema wird ignoriert) und kann mit `\showrowcolors` wieder aktiviert werden.

**`\rownum`**

Der Zähler `\rownum` (oder  $\LaTeX$  Zähler `rownum` kann innerhalb einer solchen Tabelle verwendet werden, um auf die aktuelle Zeilennummer zuzugreifen.

Derzeit wird der `rownum`-Zähler nur in Tabellen mit `\rowcolors` inkrementiert.

```

\rowcolors[\hline]{3}{green}{yellow} \arrayrulecolor{red}
\begin{tabular}{ll}
test & row \therownum\\
test & row \therownum\\
test & row \therownum\\
test & row \therownum\\
\arrayrulecolor{black}
test & row \therownum\\
test & row \therownum\\
\rowcolor{blue}
test & row \therownum\\
test & row \therownum\\
\hiderowcolors
test & row \therownum\\
test & row \therownum\\
\showrowcolors
test & row \therownum\\
test & row \therownum\\
\multicolumn{1}{%
  >{\columncolor{red}}l}{test} & row \therownum\\
\end{tabular}

```

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

## 6 Der `\cellcolor` Befehl

Die Einstellung der Hintergrundfarbe kann auch auf eine einzelne Zelle beschränkt werden, indem zu Beginn der Befehl `\multicolumn{1}{>\rowcolor...}`, (oder `\columncolor`, wenn keine Zeilenfarbe eingestellt ist) eingegeben wird. Hier treten jedoch Defizite auf: 1) Es hindert die Daten in der Zelle, die Färbung auszulösen; 2) die Definitionen für die Ausrichtung der Tabelle müssen aus dem Tabellenkopf kopiert werden und sind anfällig für Fehlermeldungen, insbesondere bei  $p$  Spalten; 3) die Anweisung `\multicolumn{1}` ist unsinnig. Ersatzweise gibt es den `\cellcolor`-Befehl, der wie `\columncolor` und `\rowcolor` funktioniert, aber beide aufhebt. `\cellcolor` kann auf jede einzufärbende Tabellenzelle angewendet werden.

## 7 Linien einfärben

Sie benötigen auch farbige Linien?

Das Einfärben von Linien bedarf keiner speziellen Befehle. Verwenden Sie einfach `!\color{green}\vline` an Stelle von `|`. Die Leerstelle zwischen `||` ist im Normalfall weiß. Um diese farbig zu gestalten, erweitern Sie die Überhangeinstellung der vorangehenden Spalte (zu `\tabcolsep + \arrayrulewidth + \doublerulesep`). Oder entfernen Sie die glue-Regel bzw. ersetzen Sie diese durch eine farbige Linie der erforderlichen Stärke, wie nachfolgend:

```

{\color{green}\vline}
@{\color{yellow}\vrule width \doublerulesep}
!{\color{green}\vline}

```

Es sollte sich der gleiche Abstand wie bei `||` ergeben, nur mit entsprechender Farbigkeit.

Allerdings stellt sich das Einfärben von `\hline` und `\cline` als etwas kniffliger heraus. Deshalb wurden extra Befehle eingeführt (die dann auch auf vertikale Linien angewendet werden können).

## 8 `\arrayrulecolor`

`\arrayrulecolor` benötigt die gleichen Argumentfestlegungen wie `\color`. Es handelt sich um eine globale Deklaration, die alle folgenden horizontalen und vertikalen Linien in Tabellen betrifft. Sie kann folgendermaßen definiert werden: Wird außerhalb einer jeden Tabelle, zu Beginn einer Zeile oder als `>` Definition innerhalb einer Tabellenpräambel in der Tabellenmitte eine Regel eingefügt, gilt diese nur für alle folgenden Linien. Alle vertikalen Linien vor der Regel erhalten diejenige Farbe, welche in der Tabellenpräambel festgelegt wurde.

## 9 `\doublerulesepcolor`

Wenn die Linien bunt sind, möchten Sie möglicherweise die weißen Lücken, die durch `||` und `\hline\hline` entstanden sind, auch farbig gestalten. `\doublerulesepcolor` funktioniert wie `\arrayrulecolor`. Zu beachten ist, dass `longtable` den Leerraum, der zwischen `\hline\hline` entsteht, bei einem Seitenumbruch beibehält. (T<sub>E</sub>X löscht diesen Leerraum automatisch, jedoch die gefärbte Fläche, welche vorher von `\doublerulesep` genutzt wurde, ist im Prinzip eine dritte Linie in einer anderen Farbe als die beiden anderen Linien. Linien sind aber hingegen nicht so einfach zu löschen.)

```

\setlength\arrayrulewidth{2pt}\arrayrulecolor{blue}
\setlength\doublerulesep{2pt}\doublerulesepcolor{yellow}
\begin{tabular}{||1||c||}
  \hline\hline
  eins & zwei\\
  drei & vier\\
  \hline\hline
\end{tabular}

```

eins	zwei
drei	vier



## 10 Mehr Spaß mit `\hline`

Die obigen Befehle arbeiten mit `\hline` des `hline`-Pakets. Wie auch immer `hline` geladen wird, es gibt neben diesem Paket noch eine andere Möglichkeit. Es kann `>\ldots` genutzt werden, um Definitionen hinzuzufügen, welche zu den - oder = `column`-Regel passen. Insbesondere können `\arrayrulecolor`- oder `\doublerulesepcolor`-Festlegungen ergänzt werden. Viele Stilhandbücher warnen davor, innerhalb von Tabellen Regeln einzufügen. Ich vermag es nicht, mir vorzustellen, was jene Kritiker aus dem folgenden Regenbogen-Beispiel gemacht hätten:



```

\newcommand\rainbowline[1]{%
\hline{%
>\arrayrulecolor {red}\doublerulesepcolor[rgb]{.3,.3,1}%
|#1:=%
>\arrayrulecolor{orange}\doublerulesepcolor[rgb]{.4,.4,1}%
=%
>\arrayrulecolor{yellow}\doublerulesepcolor[rgb]{.5,.5,1}%
=%
>\arrayrulecolor {green}\doublerulesepcolor[rgb]{.6,.6,1}%
=%
>\arrayrulecolor {blue}\doublerulesepcolor[rgb]{.7,.7,1}%
=%
>\arrayrulecolor{indigo}\doublerulesepcolor[rgb]{.8,.8,1}%
=%
>\arrayrulecolor{violet}\doublerulesepcolor[rgb]{.9,.9,1}%
=:#1|}%
}}
\arrayrulecolor{red}
\doublerulesepcolor[rgb]{.3,.3,1}%
\begin{tabular}{||*7{>\columncolor[gray]{.9}}c||}
\rainbowline{t}%
\arrayrulecolor{violet}\doublerulesepcolor[rgb]{.9,.9,1}
Richard & of & York & gave & battle & in &
\multicolumn{1}{>\columncolor[gray]{.9}}c||}{vain}\\
\rainbowline{%
1 & 2 & 3 & 4 & 5 & 6 &

```

```

\multicolumn{1}{>{\columncolor[gray]{.9}c||}{7}}\
\rainbowline{b}%
\end{tabular}

```

## 11 Weniger Spaß mit `\cline`

Mit `\cline` erzeugte Linien können durch `\arrayrulecolor` eingefärbt werden. Tritt jedoch in der folgenden Zeile ein Befehl zur Erzeugung eines Farbfelds auf, überdeckt dieser die Linienfärbung. Das ist ein kleines ‘Feature’ von `\cline`. Wenn Sie `colortbl` verwenden, sollten Sie innerhalb des `\hline`-Arguments anstelle von `\cline` besser den - Linientyp verwenden

## 12 Der `\minrowclearance` Befehl

Weil `colortbl` jeden Tabelleneintrag verpacken und berechnen muss, um zu ermitteln wie lang die Linien gezogen werden müssen, dachte ich daran, das `\minrowclearance`-Feature zu ergänzen. Denn manchmal berühren Einträge eine vorhergehende `\hline` oder den Anfang eines Farbfeldes, das durch dieses Layout definiert wurde. Um sicher zu gehen, dass das nicht passiert, sollten `\extrarowsep` und `\arraystretch` ergänzt werden. Dies reguliert den Abstand der Linien angemessen. Manchmal möchte man aber trotzdem über einem großen Eintrag einen extra Platzhalter einfügen. Für einen kleinen Leerraum können sie den Befehl `\minrowclearance` einfügen. (Die Höhe einer Tabellenzeile sollte die Höhe eines Großbuchstabens plus dieses Leerraums aber nicht überschreiten, sonst wirkt die Tabellenaufteilung unvorteilhaft.)

Donald Arseneaus Paket `tabls` verwendet einen ähnlichen `\tablinesep`-Befehl. Ich gab meinem Befehl den gleichen Namen, um eine Kompatibilität mit `tabls` zu ermöglichen. Aber `tabls` ist, wenn man es einbindet, recht schwierig und verhält sich vermutlich anders. Deshalb verwende ich jetzt einen anderen Namen.