

## Werkzeuge

|   |     |
|---|-----|
| 14.1 Farben mit <code>xcolor</code> . . . . .                       | 669 |
| 14.2 Verbatimmodus . . . . .  | 680 |
| 14.3 Programmlistings mit dem Paket <code>listings</code> . . . . . | 687 |
| 14.4 Zahlen und Einheiten . . . . .                                 | 700 |
| 14.5 Systemtest . . . . .   | 706 |

### 14.1 Farben mit `xcolor`

$\text{\LaTeX}$  kennt bekanntermaßen selbst keine Farben, was in der Vergangenheit zur Entwicklung verschiedener Pakete führte, von denen das Paket `xcolor` von Uwe Kern die beste Unterstützung bietet. Das Paket erlaubt eine ganze Reihe von Optionen (Tabelle 14.1), die sich vorrangig auf das Farbmodell (Tabelle 14.2 auf der nächsten Seite) beziehen.

Tabelle 14.1: Zusammenstellung der wichtigsten Paketooptionen von `xcolor`

| Option                  | Bedeutung  |
|-------------------------|--|
| <code>natural</code>    | (Standard) Benutze alle Farben innerhalb ihres Modells, mit Ausnahme von RGB (konvertiert to <code>rgb</code> ), HSB (konvertiert to <code>hsb</code> ), und Gray (konvertiert to <code>gray</code> ). |
| <code>rgb</code>        | Konvertiert alle Farben in das <code>rgb</code> -Modell.   |
| <code>cmY</code>        | Konvertiert alle Farben in das <code>cmY</code> -Modell.   |
| <code>cmYk</code>       | Konvertiert alle Farben in das <code>cmYk</code> -Modell.  |
| <code>hsb</code>        | Konvertiert alle Farben in das <code>hsb</code> -Modell.   |
| <code>gray</code>       | Konvertiert alle Farben in das <code>gray</code> -Modell.  |
| <code>RGB</code>        | Konvertiert alle Farben in das <code>RGB</code> -Modell (und danach <code>rgb</code> ).  |
| <code>HTML</code>       | Konvertiert alle Farben in das <code>HTML</code> -Modell (und danach <code>rgb</code> ).   |
| <code>HSB</code>        | Konvertiert alle Farben in das <code>HSB</code> -Modell (und danach <code>hsb</code> ).  |
| <code>Gray</code>       | Konvertiert alle Farben in das <code>Gray</code> -Modell (und danach <code>gray</code> ).  |
| <code>dvipsnames</code> | Lädt die vordefinierten DVIPS-Farben.  |
| <code>svgnames</code>   | Lädt die vordefinierten SVG-Farben   |

Fortsetzung ...

| Option     | Bedeutung   |
|------------|---|
| x11names   | Lädt die vordefinierten X11-Farben  |
| prologue   | schreibt die Liste der Farbnamen (dvipsnames) in den PS-Header, was für die Dokumentenerstellung via DVIPS wichtig ist. |
| table      | Lädt das Paket colortbl für farbige Tabellenzeilen.   |
| hyperref   | Unterstützung des hyperref Packets.   |
| prologue   | Schreibt PS-Prolog-Datei, um vordefinierte Farbnamen für DVIPS zu ermöglichen.  |
| showerrors | (Standard) Gibt eine Meldung bei undefinierten Farben aus.  |
| hideerrors | Gibt nur eine Fehlermeldung aus, wenn die undefinierte Farbe benutzt wird und setzt sie dann auf Schwarz.               |

Tabelle 14.2: Unterstützte Farbmodelle.

| Name | Grundfarben                  | Parameterbereich    |
|------|------------------------------|---------------------|
| 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,...,255}       |
| HTML | RRGGBB                       | {000000,...,FFFFFF} |
| HSB  | Hue, Saturation, Brightness  | {0,1,...,240}       |
| Gray | Gray                         | {0,1,...,15}        |
| wave | lambda (nm – nanometer)      | [363,814]           |

14.1.1 Definition von Farben über numerische Werte

```
\color[Modell]{Spezifikation}
\textcolor[Modell]{Spezifikation}{Text}
```

Nun ist der Text grün und nun ein wenig magenta und jetzt blau und nun wieder grün.

```
\usepackage{xcolor}

Nun ist der Text \color[rgb]{0,1,0} grün und
nun \textcolor[cmYk]{0,1,0,0}{ein wenig magenta}
und {\color[rgb]{0,0,1}}jetzt blau} und nun wieder grün.
```

14-01-1

Die Farben black, white, red, green, blue, cyan, magenta und yellow sollten von jedem beliebigen Treiber definiert werden, sodass der Anwender sie direkt über ihre Namen benutzen können sollte. xcolor definiert die weiteren Grundfarben gray, lightgray, darkgray, brown, lime, olive, orange, pink, purple, violet und teal. Das Makro \definecolor ist bereits durch das color-Paket definiert, wird jedoch durch xcolor in seinen Möglichkeiten erweitert. Es stehen die folgenden Makros für weitere Farbdefinitionen zur Verfügung:

```

\definecolor [Typ] {Name} {Modell} {Farbspezifikation}
\providecolor [Typ] {Name} {Modell} {Farbspezifikation}
\colorlet {Name} [num Modell] {Farbe}
\definecolorset [Typ] {Modell} {Präfix} {Suffix} {Setspezifikation}
\providecolorset [Typ] {Modell} {Präfix} {Suffix} {Setspezifikation}
\definecolorseries {Name} {Basismodell} {Methode} [b-Modell] {b-Spezif.}
    [s-Modell] {s-Spezif.}
\resetcolorseries [Anzahl] {Name}

```

Weiterhin definiert xcolor die aktuelle Farbe durch den Punkt, auf die in derselben Weise zugegriffen werden kann, wie über normale Farbnamen. Im folgenden Beispiel wird mehrere Male hintereinander die aktuelle (Punkt-)Farbe auf 80% (`\color{.!80}`) ihres derzeitigen Wertes verändert, was letztlich zur Farbe Weiß führt. Die folgende `\colorbox` wiederum benutzt die komplementäre aktuelle Farbe (`-.`) für den Hintergrund, um `\iCol` lesbar zu gestalten.

14-01-2



```

\usepackage{xcolor,multido}
\newcommand*\CBox[1]{\color{.!80}\colorbox{.}{%
  \rule[-3ex]{0pt}{7ex}{\color{-.}#1}}}
\color{red}\multido{\iCol=1+1}{11}{\CBox{\large\iCol}}

```

Im Gegensatz zu Makronamen dürfen Farbnamen ohne weiteres Ziffern enthalten, was insbesondere für Farbnamen wie *Grau40* hilfreich sein kann. Andererseits können diese prozentualen Angaben auch direkt bei der Farbauswahl angegeben werden: `\textcolor{black!40}{black40}→black40`. xcolor erlaubt weitere Zeichen, wovon man allerdings keinen Gebrauch machen sollte, um für andere Pakete oder zukünftige Entwicklungen kompatibel zu bleiben.

Die Wirkungsweise der beiden Makros `\providecolor` und `\providecolorseries` ist analog zum bekannten  $\text{\LaTeX}$  Makro `\providecommand`. Wenn eine Farbe oder eine Farbserie bereits unter dem angegebenen Namen existiert, erfolgt keine Neudefinitor, womit ein Überschreiben verhindert werden kann.

```

\definecolor{MyOrange}{cmyk}{0,0.42,1,0} \definecolor[named]{Blue}{rgb}{0,0,0.8}
\providecolor{MyGrey}{gray}{0.75} \definecolor{MyBlack}{named}{Black}
\colorlet{MyRGBO}{rgb}{MyOrange}

```

Mit der Definition obiger Farben stehen *Myorange*, *Blue*, *MyGrey*, *MyBlack* und *MyRGBO* allgemein zur Verfügung und können zusätzlich zu den vordefinierten benutzt werden. Dies ist im folgenden Beispiel zu sehen, welches über die Paketoptionen `dvipsnames` und `prologue` die zusätzlichen Farbnamen des dvips Programms verfügbar macht. `\colorlet` benutzt die vorher im CMYK-Farbmodell definierte Farbe *MyOrange* als Grundlage. Dazu wird diese in das CMYK-Modell umgerechnet und dann als *MyRGBO* bezeichnet.

`\definecolor` wirkt grundsätzlich lokal, `\definecolorseries` aber global!



```
\usepackage[dvipsnames,prologue]{xcolor}
\definecolor{MyOrange}{cmyk}{0,0.42,1,0}
\definecolor[named]{Blue}{rgb}{0,0,0.8}
\definecolor{MyGrey}{gray}{0.75}
\definecolor{MyBlack}{named}{Black}
\colorlet{MyRGB0}{rgb}{MyOrange}
\newcommand*\col[1]{\color{#1}\rule{3cm}{5mm}}

{\col{MyOrange}}\{\col{Blue}}\
{\col{MyGrey}}\{\col{MyBlack}}\{\col{MyRGB0}}
```

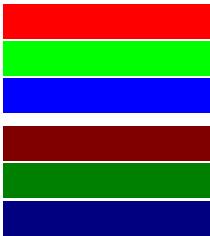
14-01-3

Die Option `table` ist nur bei farbigen Tabellen notwendig, wie beispielsweise in Tabelle 14.1 auf Seite 669 und 14.2 auf Seite 670.

`xcolor` unterstützt die Definition von Farbserien mit zwei Makros, wobei diese nichts weiter als eine erweiterte Version von `\definecolor` sind.

```
\definecolorset{rgb}{}{}{red,1,0,0;green,0,1,0;blue,0,0,1}
\providecolorset{rgb}{}{H}{red,0.5,0,0;green,0,0.5,0;blue,0,0,0.5}
```

Das erste Beispiel definiert die drei RGB-Basisfarben und das zweite drei neue Basisfarben mit dem Suffix *H*: *redH*, *greenH* und *blueH*. `\providecolorset` hätte man auch durch `\definecolorset` ersetzen können, denn es existierten noch keine entsprechenden Farbnamen. Umgekehrt wäre das nicht möglich gewesen, denn die drei RGB-Basisfarben *red*, *green* und *blue* sind durch `xcolor` bereits definiert.



```
\usepackage{xcolor}
\definecolorset{rgb}{}{}{
  red,1,0,0;green,0,1,0;blue,0,0,1}
\providecolorset{rgb}{}{H}{%
  red,0.5,0,0;green,0,0.5,0;blue,0,0,0.5}
\newcommand*\col[1]{\color{#1}\rule{3cm}{5mm}}

{\col{red}}\{\col{green}}\
{\col{blue}}\{\col{redH}}\{\col{greenH}}\{\col{blueH}}
```

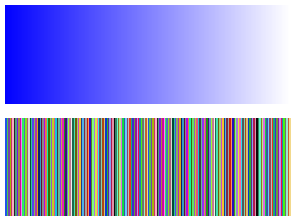
14-01-4

Eine ganze Farbserie lässt sich mit `\definecolorseries` festlegen, wobei der Anwender sowohl Farbanfang und Farbende als auch die Anzahl der Zwischenschritte vorgeben kann, beispielsweise eine Serie von Schwarz nach Weiß in 200 Schritten:

```
\definecolorseries{testA}{rgb}{last}{black}{white}
\resetcolorseries[200]{testA}% definiert eine Serie mit 200 Farben
```

Mit dem Makro `\definecolorseries` wird diese Serie »logisch« definiert, während `\resetcolorseries` dies erst »physikalisch« durchführt, indem es die Zwischenschritte berechnet und alle Teilfarben intern in einer Art Feld definiert. Dabei kann man jederzeit auf einzelne Farben des Feldes zugreifen: `\testA!!` [Index].

14-01-5



```
\usepackage{xcolor,multido}
\definecolorseries{testA}{rgb}{last}{blue}{white}
\resetcolorseries[200]{testA}% 200 Farbschritte
\definecolorseries{testB}{rgb}{step}{rgb}{%
    0.95,0.85,0.55}{0.17,0.47,0.37}
\resetcolorseries[200]{testB}
\linethickness{0.004\linewidth}

\multido{\nC=1+1}{200}{\hspace*{0.004\linewidth}%
    \color{testA!![\nC]}\line(0,1){40}}\line(0,1){40}}
\multido{\nC=1+1}{200}{\hspace*{0.004\linewidth}%
    \color{testB!![\nC]}\line(0,1){40}}
```

Wie die Zwischenwerte berechnet werden, hängt von der gewählten Methode ab:

- *{b-Modell}{b-Spezif.}* Angabe der ersten Farbe.
- *{s-Modell}{s-Spezif.}* Berechnung der Zwischenschritte in Abhängigkeit von method:
  - step, grad: Das optionale Argument ist bedeutungslos und *{s-Spezif.}* ist ein Vektor, dessen Dimension durch *{Basismodell}* festgelegt ist, bsp. [hsb]{0.1,-0.2,0.3}.
  - last: Angabe der letzten Farbe, beispielsweise [rgb]{0.1,0.5,0.5}.

Wie die Berechnung der Zwischenschritte genau erfolgt, kann der Dokumentation zu xcolor entnommen werden. [14] Die einfachste Methode eine Farbserie zu definieren, ergibt sich bei Verwendung der last-Option, wie es auch in den folgenden beiden Beispielen gezeigt wird. Das erste bezieht sich auf das CMYK und das zweite auf das HSB-Modell.

```
\usepackage{xcolor,multido}
\definecolorseries{C}{cmyk}{last}{white}[cmyk]{1,0,0,0}\resetcolorseries[10]{C}
\definecolorseries{M}{cmyk}{last}{white}[cmyk]{0,1,0,0}\resetcolorseries[10]{M}
\definecolorseries{Y}{cmyk}{last}{white}[cmyk]{0,0,1,0}\resetcolorseries[10]{Y}
\definecolorseries{K}{cmyk}{last}{white}[cmyk]{0,0,0,1}\resetcolorseries[10]{K}

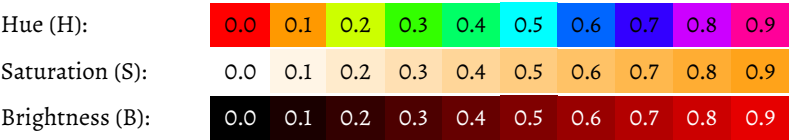
\makebox[30mm][l]{cyan (C):}%
\multido{\nColr=0+1}{10}{\colorbox{C!![\nColr]}\{0.\nColr}}\line(0,1){40}}
\makebox[30mm][l]{magenta (M):}%
\multido{\nColr=0+1}{10}{\colorbox{M!![\nColr]}\{0.\nColr}}\line(0,1){40}}
\makebox[30mm][l]{yellow (Y):}%
\multido{\nColr=0+1}{10}{\colorbox{Y!![\nColr]}\{0.\nColr}}\line(0,1){40}}
\makebox[30mm][l]{black (K):}%
\multido{\nColr=0+1}{10}{\colorbox{K!![\nColr]}\{0.\nColr}}
```

14-01-6

|              |     |     |     |     |     |     |     |     |     |     |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| cyan (C):    | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| magenta (M): | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| yellow (Y):  | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| black (K):   | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |

```
\usepackage{xcolor,multido}
\definecolorseries{H}{hsb}{last}{hsb}{0,1,1}{hsb}{1,1,1}\resetcolorseries[10]{H}
\definecolorseries{S}{hsb}{last}{hsb}{.1,0,1}{hsb}{.1,1,1}\resetcolorseries[10]{S}
\definecolorseries{B}{hsb}{last}{hsb}{1,1,0}{hsb}{1,1,1}\resetcolorseries[10]{B}

\makebox[30mm][l]{Hue (H):}%
\multido{\nColr=0+1}{10}{\colorbox{H!{\nColr}}{0.\nColr}}\
\makebox[30mm][l]{Saturation (S):}%
\multido{\nColr=0+1}{10}{\colorbox{S!{\nColr}}{0.\nColr}}\
\makebox[30mm][l]{Brightness (B):}%
\multido{\nColr=0+1}{10}{\colorbox{B!{\nColr}}{\color{white}0.\nColr}}\
```



14-01-7

### 14.1.2 Farbspezifikation

- »specification by name«, wobei die aktuelle Farbe den Punkt (.) als »Namen« hat.
- »specification by expression« mit der Syntax:  
*PräfixName!WertI!NameI!...!WertN!NameNSuffix*

*Präfix* Ist dieses ein Minuszeichen (»-«), dann erfolgt vor Anwendung eine Umwandlung in die Komplementärfarbe.

*Name* Modell- und Farbparameter von *Name* bilden eine temporäre Farbe *\temp*.

*WertI!NameI* Die neue Farbe ergibt sich aus *WertI*% der Farbe *temp* und (100 – *WertI*)% von der Farbe *NameI* und wird dann wieder als neue temporäre Farbe *\temp* gespeichert. Dieser Schritt wird für alle weiteren *!Wert!Name* wiederholt. Ist ein *Suffix* definiert, dann ist die temporäre Farbe die zur Farbserie *Name* korrespondierende.

*Suffix* Kann eine der Formen *!+*, *!!++*, *!!!++*, usw. annehmen. Die Anzahl der Pluszeichen (+) bezieht sich auf die zugrundeliegende Farbserie.

Tabelle 14.3 zeigt einige Beispiele für Farbausdrücke und ihren äquivalenten Ausdruck als RGB-Code. Die rechte Hälfte der Tabelle zeigt die Anwendung der komplementären Farben, die jeweils zu denen der ersten Hälfte addiert werden. Farbe plus Komplementärfarbe addiert, ergibt bekanntlich Weiß mit dem RGB-Wert »1 1 1«.

Tabelle 14.3: Farbausdrücke und ihr entsprechendes RGB-Tripel

| Farbausdruck           | RGB-Tripel    | Farbausdruck            | RGB-Tripel    |
|------------------------|---------------|-------------------------|---------------|
| red                    | 1 0 0         | -red                    | 0 1 1         |
| red!75                 | 1 0.25 0.25   | -red!75                 | 0 0.75 0.75   |
| red!75!blue!100        | 0.75 0 0.25   | -red!75!blue!100        | 0.25 1 0.75   |
| red!75!blue!405        | 0.9 0.6 0.7   | -red!75!blue!40         | 0.1 0.4 0.3   |
| red!75!blue!40!cyan!50 | 0.95 0.8 0.55 | -red!75!blue!40!cyan!50 | 0.05 0.2 0.45 |

14-01-8



```
\usepackage{pstricks}

\psframebox[linecolor={red!70!green},
fillcolor=yellow!90!cyan, fillstyle=solid,
doubleline=true, doublesep=5pt, framesep=10pt,
doublecolor=-yellow!90!cyan]{\Large RGB%
\textcolor{red!72.75}{RGB}\color{-green}RGB}
```

Zusätzlich zu den »normalen« Farbausdrücken, kennt xcolor noch die »erweiterten« Farbausdrücke, die sich an einer Farbmischung orientieren. Jede Farbe in dem Ausdruck enthält einen ganzzahligen Faktor, der die Gewichtung der Farbe darstellt.

14-01-9

|  |                                  |  |                                   |
|--|----------------------------------|--|-----------------------------------|
|  | red                              |  | - red                             |
|  | red!75                           |  | - red!75                          |
|  | red!75!green                     |  | - red!75!green                    |
|  | red!75!green!50                  |  | - red!75!green!50                 |
|  | red!75!green!50!blue             |  | - red!75!green!50!blue            |
|  | red!75!green!50!blue!25          |  | - red!75!green!50!blue!25         |
|  | red!75!green!50!blue!25!gray     |  | - red!75!green!50!blue!25!gray    |
|  | rgb: red,1;white,2               |  | cmymk: red,2;white,2              |
|  | rgb: red,5;green,2;yellow,10     |  | cmymk: red,5;green,2;yellow,10    |
|  | rgb: red,11;red,5;green,2;cyan,1 |  | cmymk: red,11;red,5;cyan,2;cyan,1 |

### 14.1.3 Anwendung vordefinierter Farben (»named colors«)

Die so genannten »named colors«, die der zugrundeliegende Farbtreiber, beispielsweise dvips zur Verfügung stellt, erlauben eine vereinfachte Anwendung bei der Neudefinition von Farben: `\color[named]{SpringGreen}` wählt die vordefinierte Farbe SpringGreen. Alternativ kann auch eine neue Farbe mit gleicher Eigenschaft definiert werden. xcolor erlaubt die Verwendung der vordefinierten Farbnamen für alle Ausgabetreiber.

```
\definecolor{MyGreen}{named}{SpringGreen}
```

Danach existiert dann die Farbe `\color{MyGreen}`. Sämtliche »named colors« können über die Paketoption angefordert werden. Eine Zusammenstellung der verfügbaren Farbnamen für das dvipsnames- als auch die svgnames- und x11names-Farbmodelle zeigen die Tabellen 14.4, 14.5 und 14.6.

Tabelle 14.4: Liste der verfügbaren Farbnamen für das dvipsnames-Modell

|  |           |  |               |  |             |  |                |  |             |
|--|-----------|--|---------------|--|-------------|--|----------------|--|-------------|
|  | Apricot   |  | Aquamarine    |  | Bittersweet |  | Black          |  | Blue        |
|  | BlueGreen |  | BlueViolet    |  | BrickRed    |  | Brown          |  | BurntOrange |
|  | CadetBlue |  | CarnationPink |  | Cerulean    |  | CornflowerBlue |  | Cyan        |
|  | Dandelion |  | DarkOrchid    |  | Emerald     |  | ForestGreen    |  | Fuchsia     |
|  | Goldenrod |  | Gray          |  | Green       |  | GreenYellow    |  | JungleGreen |
|  | Lavender  |  | LimeGreen     |  | Magenta     |  | Mahogany       |  | Maroon      |
|  | Melon     |  | MidnightBlue  |  | Mulberry    |  | NavyBlue       |  | OliveGreen  |
|  | Orange    |  | OrangeRed     |  | Orchid      |  | Peach          |  | Periwinkle  |



Tabelle 14.5: Liste der verfügbaren Farbnamen für svgnames

|  |   |  |   |
|--|---|--|---|
|  AliceBlue           |  AntiqueWhite      |  Aqua                   |  Aquamarine        |
|  Azure               |  Beige             |  Bisque                 |  Black             |
|  BlanchedAlmond      |  Blue              |  BlueViolet             |  Brown             |
|  BurlyWood           |  CadetBlue         |  Chartreuse             |  Chocolate         |
|  Coral               |  CornflowerBlue    |  Cornsilk               |  Crimson           |
|  Cyan                |  DarkBlue          |  DarkCyan               |  DarkGoldenrod     |
|  DarkGray            |  DarkGreen         |  DarkGrey               |  DarkKhaki         |
|  DarkMagenta         |  DarkOliveGreen    |  DarkOrange             |  DarkOrchid        |
|  DarkRed             |  DarkSalmon        |  DarkSeaGreen           |  DarkSlateBlue     |
|  DarkSlateGray       |  DarkSlateGrey     |  DarkTurquoise          |  DarkViolet        |
|  DeepPink            |  DeepSkyBlue       |  DimGray                |  DimGrey           |
|  DodgerBlue          |  FireBrick         |  FloralWhite            |  ForestGreen       |
|  Fuchsia             |  Gainsboro         |  GhostWhite             |  Gold              |
|  Goldenrod           |  Gray              |  Grey                   |  Green             |
|  GreenYellow         |  Honeydew          |  HotPink                |  IndianRed         |
|  Indigo             |  Ivory            |  Khaki                 |  Lavender         |
|  LavenderBlush     |  LawnGreen       |  LemonChiffon         |  LightBlue       |
|  LightCoral        |  LightCyan       |  LightGoldenrodYellow |  LightGray       |
|  LightGreen        |  LightGrey       |  LightPink            |  LightSalmon     |
|  LightSeaGreen     |  LightSkyBlue    |  LightSlateGray       |  LightSteelBlue  |
|  LightYellow       |  Lime            |  LimeGreen            |  Linen           |
|  Magenta           |  Maroon          |  MediumAquaMarine     |  MediumBlue      |
|  MediumOrchid      |  MediumPurple    |  MediumSeaGreen       |  MediumSlateBlue |
|  MediumSpringGreen |  MediumTurquoise |  MediumVioletRed      |  MidnightBlue    |
|  MintCream         |  MistyRose       |  Moccasin             |  NavajoWhite     |
|  Navy              |  OldLace         |  Olive                |  OliveDrab       |
|  Orange            |  OrangeRed       |  Orchid               |  PaleGoldenrod   |
|  PaleGreen         |  PaleTurquoise   |  PaleVioletRed        |  PapayaWhip      |
|  PeachPuff         |  Peru            |  Pink                 |  Plum            |
|  PowderBlue        |  Purple          |  Red                  |  RosyBrown       |
|  RoyalBlue         |  SaddleBrown     |  Salmon               |  SandyBrown      |
|  SeaGreen          |  Seashell        |  Sienna               |  Silver          |
|  SkyBlue           |  SlateBlue       |  SlateGray            |  SlateGrey       |
|  Snow              |  SpringGreen     |  SteelBlue            |  Tan             |
|  Teal              |  Thistle         |  Tomato               |  Turquoise       |
|  Violet            |  Wheat           |  White                |  WhiteSmoke      |
|  Yellow            |  YellowGreen     |  |   |



Tabelle 14.6: Liste der verfügbaren Farbnamen für x11names

|   |   |   |   |
|---|---|---|---|
|  AntiqueWhite1     |  AntiqueWhite2     |  AntiqueWhite3     |  AntiqueWhite4     |
|  Aquamarine1       |  Aquamarine2       |  Aquamarine3       |  Aquamarine4       |
|  Azure1            |  Azure2            |  Azure3            |  Azure4            |
|  Bisque1           |  Bisque2           |  Bisque3           |  Bisque4           |
|  Blue1             |  Blue2             |  Blue3             |  Blue4             |
|  Brown1            |  Brown2            |  Brown3            |  Brown4            |
|  Burlywood1        |  Burlywood2        |  Burlywood3        |  Burlywood4        |
|  CadetBlue1        |  CadetBlue2        |  CadetBlue3        |  CadetBlue4        |
|  Chartreuse1       |  Chartreuse2       |  Chartreuse3       |  Chartreuse4       |
|  Chocolate1        |  Chocolate2        |  Chocolate3        |  Chocolate4        |
|  Coral1            |  Coral2            |  Coral3            |  Coral4            |
|  Cornsilk1         |  Cornsilk2         |  Cornsilk3         |  Cornsilk4         |
|  Cyan1             |  Cyan2             |  Cyan3             |  Cyan4             |
|  DarkGoldenrod1    |  DarkGoldenrod2    |  DarkGoldenrod3    |  DarkGoldenrod4    |
|  DarkOliveGreen1   |  DarkOliveGreen2   |  DarkOliveGreen3   |  DarkOliveGreen4   |
|  DarkOrange1       |  DarkOrange2       |  DarkOrange3       |  DarkOrange4       |
|  DarkOrchid1       |  DarkOrchid2       |  DarkOrchid3       |  DarkOrchid4       |
|  DarkSeaGreen1     |  DarkSeaGreen2     |  DarkSeaGreen3     |  DarkSeaGreen4     |
|  DarkSlateGray1    |  DarkSlateGray2    |  DarkSlateGray3    |  DarkSlateGray4    |
|  DeepPink1         |  DeepPink2         |  DeepPink3         |  DeepPink4         |
|  DeepSkyBlue1      |  DeepSkyBlue2      |  DeepSkyBlue3      |  DeepSkyBlue4      |
|  DodgerBlue1       |  DodgerBlue2       |  DodgerBlue3       |  DodgerBlue4       |
|  Firebrick1        |  Firebrick2        |  Firebrick3        |  Firebrick4        |
|  Gold1            |  Gold2            |  Gold3            |  Gold4            |
|  Goldenrod1      |  Goldenrod2      |  Goldenrod3      |  Goldenrod4      |
|  Green1          |  Green2          |  Green3          |  Green4          |
|  Honeydew1       |  Honeydew2       |  Honeydew3       |  Honeydew4       |
|  HotPink1        |  HotPink2        |  HotPink3        |  HotPink4        |
|  IndianRed1      |  IndianRed2      |  IndianRed3      |  IndianRed4      |
|  Ivory1          |  Ivory2          |  Ivory3          |  Ivory4          |
|  Khaki1          |  Khaki2          |  Khaki3          |  Khaki4          |
|  LavenderBlush1  |  LavenderBlush2  |  LavenderBlush3  |  LavenderBlush4  |
|  LemonChiffon1   |  LemonChiffon2   |  LemonChiffon3   |  LemonChiffon4   |
|  LightBlue1      |  LightBlue2      |  LightBlue3      |  LightBlue4      |
|  LightCyan1      |  LightCyan2      |  LightCyan3      |  LightCyan4      |
|  LightGoldenrod1 |  LightGoldenrod2 |  LightGoldenrod3 |  LightGoldenrod4 |
|  LightPink1      |  LightPink2      |  LightPink3      |  LightPink4      |
|  LightSalmon1    |  LightSalmon2    |  LightSalmon3    |  LightSalmon4    |
|  LightSkyBlue1   |  LightSkyBlue2   |  LightSkyBlue3   |  LightSkyBlue4   |
|  LightSteelBlue1 |  LightSteelBlue2 |  LightSteelBlue3 |  LightSteelBlue4 |
|  LightYellow1    |  LightYellow2    |  LightYellow3    |  LightYellow4    |
|  Magenta1        |  Magenta2        |  Magenta3        |  Magenta4        |
|  Maroon1         |  Maroon2         |  Maroon3         |  Maroon4         |
|  MediumOrchid1   |  MediumOrchid2   |  MediumOrchid3   |  MediumOrchid4   |
|  MediumPurple1   |  MediumPurple2   |  MediumPurple3   |  MediumPurple4   |

Fortsetzung ...

... Fortsetzung x11names

|  |  |  |  |
|--|--|--|--|
|  MistyRose1     |  MistyRose2     |  MistyRose3     |  MistyRose4     |
|  NavajoWhite1   |  NavajoWhite2   |  NavajoWhite3   |  NavajoWhite4   |
|  OliveDrab1     |  OliveDrab2     |  OliveDrab3     |  OliveDrab4     |
|  Orange1        |  Orange2        |  Orange3        |  Orange4        |
|  OrangeRed1     |  OrangeRed2     |  OrangeRed3     |  OrangeRed4     |
|  Orchid1        |  Orchid2        |  Orchid3        |  Orchid4        |
|  PaleGreen1     |  PaleGreen2     |  PaleGreen3     |  PaleGreen4     |
|  PaleTurquoise1 |  PaleTurquoise2 |  PaleTurquoise3 |  PaleTurquoise4 |
|  PaleVioletRed1 |  PaleVioletRed2 |  PaleVioletRed3 |  PaleVioletRed4 |
|  PeachPuff1     |  PeachPuff2     |  PeachPuff3     |  PeachPuff4     |
|  Pink1          |  Pink2          |  Pink3          |  Pink4          |
|  Plum1          |  Plum2          |  Plum3          |  Plum4          |
|  Purple1        |  Purple2        |  Purple3        |  Purple4        |
|  Red1           |  Red2           |  Red3           |  Red4           |
|  RosyBrown1     |  RosyBrown2     |  RosyBrown3     |  RosyBrown4     |
|  RoyalBlue1     |  RoyalBlue2     |  RoyalBlue3     |  RoyalBlue4     |
|  Salmon1        |  Salmon2        |  Salmon3        |  Salmon4        |
|  SeaGreen1      |  SeaGreen2      |  SeaGreen3      |  SeaGreen4      |
|  Seashell1      |  Seashell2      |  Seashell3      |  Seashell4      |
|  Sienna1        |  Sienna2        |  Sienna3        |  Sienna4        |
|  SkyBlue1       |  SkyBlue2       |  SkyBlue3       |  SkyBlue4       |
|  SlateBlue1     |  SlateBlue2     |  SlateBlue3     |  SlateBlue4     |
|  SlateGray1     |  SlateGray2     |  SlateGray3     |  SlateGray4     |
|  Snow1          |  Snow2          |  Snow3          |  Snow4          |
|  SpringGreen1   |  SpringGreen2   |  SpringGreen3   |  SpringGreen4   |
|  SteelBlue1    |  SteelBlue2    |  SteelBlue3    |  SteelBlue4    |
|  Tan1         |  Tan2         |  Tan3         |  Tan4         |
|  Thistle1     |  Thistle2     |  Thistle3     |  Thistle4     |
|  Tomato1      |  Tomato2      |  Tomato3      |  Tomato4      |
|  Turquoise1   |  Turquoise2   |  Turquoise3   |  Turquoise4   |
|  VioletRed1   |  VioletRed2   |  VioletRed3   |  VioletRed4   |
|  Wheat1       |  Wheat2       |  Wheat3       |  Wheat4       |
|  Yellow1      |  Yellow2      |  Yellow3      |  Yellow4      |
|  Gray0        |  Green0       |  Grey0        |  Maroon0      |
|  Purple0      |  |  |  |

### 14.1.4 Farben in Boxen

Farbdefinitionen, die innerhalb einer `\savebox` erfolgen, werden grundsätzlich beim Schreiben in die Box ebenfalls in dieser gespeichert und sind somit lokal und können nicht mehr von außen überschrieben werden.

```
\usepackage{xcolor}
\newsavebox{\X}\sbox{\X}{[Schwarz] und \color[cm]{0,0.6,0.8,0}[Orange]}
```

Start mit `\usebox{\X}`, und zurück zu Schwarz.`\par`  
`\color{green}`Start mit Grün, siehe `\usebox{\X}` und wieder Grün.

14-01-10

Start mit [Schwarz] und [Orange], und zurück zu Schwarz.  
 Start mit Grün, siehe [Schwarz] und [Orange] und wieder Grün.

### 14.1.5 Seiten- und Boxhintergrundfarbe

```
\pagecolor{Name}
```

Die Hintergrundfarbe der gesamten Seite kann mit `\pagecolor` verändert werden, wobei dieselbe Syntax wie bei `\color` verwendet wird. Sowohl die aktuelle als auch alle folgenden Seiten werden auf die angegebene Hintergrundfarbe gesetzt, wobei diese Definition grundsätzlich global wirkt und somit innerhalb einer `minipage` oder Gruppe keinen Sinn macht. Analog zum bekannten Makro `\fbox` existieren zwei Makros zum Setzen der Hintergrundfarbe von Boxen, wobei `\fcolorbox` zusätzlich das Setzen der Rahmenfarbe ermöglicht.

```
\colorbox{Hintergrundfarbe}{Text}
\fcolorbox{Rahmenfarbe}{Hintergrundfarbe}{Text}
```

```
\usepackage{xcolor}
\definecolor{Light}{gray}{.80} \definecolor{Dark}{gray}{.20}

\colorbox{red!40}{Schwarz auf 40\%-Rot}\hfill%
\fcolorbox{red}{cyan!40}{Schwarz -- Text, cyan -- Hintergrund, Rot -- Rahmen}\\
\colorbox{Light}{\textcolor{Dark}{Heller Hintergrund}}\hfill%
\fcolorbox{red}{cyan!60}{\color{white}Weiß -- Text, cyan -- Hintergrund, Rot -- Rahmen}
```

14-01-11

Schwarz auf 40% Rot

Heller Hintergrund

Schwarz – Text, cyan – Hintergrund, Rot – Rahmen

Weiß – Text, cyan – Hintergrund, Rot – Rahmen

Das folgende Beispiel zeigt die Handhabung der `\fbox`-Parameter `\fboxrule` und `\fboxsep`, welche zum einen die Liniendicke und zum anderen den Abstand zwischen innerem Box-Text und Rahmen festlegen.

```
\usepackage{xcolor}

\setlength{\fboxsep}{10pt}\setlength{\fboxrule}{6pt}
\colorbox{yellow!40}{\color{blue}Spaß mit xcolor}
\fcolorbox{red!40}{yellow}{Spaß mit xcolor}
\setlength{\fboxrule}{1pt}\colorbox{green!40}{Spaß mit xcolor}
\fcolorbox{blue}{green!40}{\color{blue}Spaß mit xcolor}
```

14-01-12

Spaß mit xcolor

Spaß mit xcolor

Spaß mit xcolor

Spaß mit xcolor

### 14.1.6 Farbwerte bestimmen

Mithilfe des `xcolor`-Pakets lassen sich auf einfache Art und Weise Farbwerte oder auch Farbserien bestimmen (Abschnitt 14.1.1). Möchte man beispielsweise eine Farbe analog zu einer gegebenen HTML-Seite definieren, so kann dies mit dem gleichnamigen Modell geschehen, wobei man sich die entsprechenden Werte in einem anderen Modell mithilfe von `\convertcolorspec` transformieren und ausgeben lassen kann.

HTML-Farbe **FF2006**  
 rgb : **1,0.12549,0.02353**  
 cmyk:**0,0.87451,0.97647,0**  
 hsb : **0.0174,0.97647,1**

```
\usepackage{xcolor}

\definecolor{HTMLFarbe}{HTML}{FF2006}% #FF2006
HTML-Farbe \ttfamily\textcolor{HTMLFarbe}{FF2006}\\
rgb :\convertcolorspec{HTML}{FF2006}{rgb}\RGBFarbe
\textcolor[rgb]{\RGBFarbe}{\RGBFarbe}\\
cmyk:\convertcolorspec{HTML}{FF2006}{cmyk}\CMYKFarbe
\textcolor[cmyk]{\CMYKFarbe}{\CMYKFarbe}\\
hsb :\convertcolorspec{HTML}{FF2006}{hsb}\HSBFarbe
\textcolor[hsb]{\HSBFarbe}{\HSBFarbe}
```

14-01-13

## 14.2 Verbatimmodus

Der Verbatimmodus ignoriert bis auf das Zeilenendezeichen alle aktiven Zeichen, sodass diese ohne weiteres benutzt werden können. Dafür stehen ein Makro und eine Umgebung zur Verfügung:

```
\verb*⟨Begrenzer⟩ Text ⟨Begrenzer⟩
\begin{verbatim}*}
Text
\end{verbatim}*}
```

Der Zeilenvariante `\verb` wird durch zwei *gleiche* Begrenzer der verbatim zu setzende Text mitgeteilt. Die Begrenzer können formal beliebige Zeichen sein, beispielsweise `\verb$\verb$`  $\Rightarrow$  `\verb.` Ausgenommen sind das Sternchen und Buchstaben, wobei auch der Backslash ein zulässiges Zeichen ist: `\verb\^~\`  $\Rightarrow$  `^~`. Das Begrenzersymbol selbst darf natürlich dann nicht innerhalb des verbatim zu setzenden Textes auftreten. Die Sternvariante gibt Leerzeichen durch `\textvisiblespace` aus: `\verb*|\text{A B C}|`  $\Rightarrow$  `\text{A_B_C}`.

Die Umgebung `verbatim` setzt ihren Text in einen eigenen Absatz, wobei die Sternvariante wieder alle Leerzeichen mit einem Symbol ausgibt. Analog zur Einschränkung, dass ein Begrenzer nicht Teil des Textes sein kann, kann auch `\end{verbatim}` nicht innerhalb der Umgebung auftreten; es definiert eindeutig das Ende und kann daher nur durch einen kleinen Trick dargestellt werden, falls man die Umgebung selbst als Code darstellen will.

14-02-1

|                               |  |
|-------------------------------|--|
| Anfang der Umgebung.          | Anfang der Umgebung.                           |
| <code>\begin{verbatim}</code> | <code>\begin{verbatim}</code>                  |
| <code>\foo_\bar_\baz</code>   | <code>\foo \bar \baz</code>                    |
| <code>\end{verbatim}</code>   | <code>\end{verbatim}% beendet die Umg.</code>  |
| Ende der Umgebung.            | <code>\vspace*{-0.75\baselineskip}</code>      |
|                               | <code>\verb=\end{verbatim}= % der Trick</code> |
|                               | <code>\par\bigskip</code> Ende der Umgebung.   |

Der Inhalt von `\verb` und `verbatim` wird durch die mit `\verbatim@font` definierte Schrifteinstellung dargestellt, standardmäßig mit `\normalfont\ttfamily`. Eine Änderung ist durch einfaches Redefinieren des Makros möglich, wobei dieses in `\makeatletter ... \makeatother` einzuschließen ist. Die Möglichkeiten sind aber ohnehin eingeschränkt, sodass eine Anwendung des Paketes `fancyvrb` angebracht erscheint.

Das Makro `\verb` und auch die Umgebung können nicht Teil eines Arguments eines anderen Makros oder Umgebung sein, beispielsweise einer Fußnote. Dies ist eine wesentliche Einschränkung, die zum einen mit dem Paket `fancyvrb` oder der Anwendung des Makros `\texttt` umgangen werden kann.



14.2.1 Das Paket fancyvrb

Das Paket `fancyvrb` von Timothy Van Zandt bietet neben dem Paket `listings` die umfangreichsten Möglichkeiten verbatim Code einzubinden. Alle Einstellungen lassen sich über das Makro `\fvset` global vornehmen.

```
\fvset{Key/Value-Angaben}
```

Die möglichen Parameter sind alle in Tabelle 14.7 zusammengestellt.

Tabelle 14.7: Zusammenstellung der möglichen Optionen für das Paket `fancyvrb`.

| Name        | Bedeutung  |
|-------------|--|
| commentchar | Zeichen, welches einen Kommentar im Originaltext einleitet und daher nicht ausgegeben wird, wobei alle $\LaTeX$ -Sonderzeichen mit einem Backslash »\« beginnen müssen.  |
| gobble      | Die Anzahl an Zeichen, die zu Beginn einer jeder Zeile unterdrückt werden (maximal Neun).  |
| formatcom   | Makros, die vor dem Beginn der Umgebung <code>Verbatim</code> ausgeführt werden.   |
| fontfamily  | Definiert die anzuwendende Schriftfamilie, wobei das Paket hier seine eigenen Definitionen vornimmt. Standardmäßig sind <code>tt</code> (Vorgabe), <code>helvetica</code> und <code>courier</code> definiert.  |
| fontsize    | Festlegung der Schriftgröße, wobei die aktuelle Einstellung die Vorgabe ist.   |
| fontshape   | Festlegung der Schriftform, wobei die aktuelle Einstellung die Vorgabe ist.  |
| fontseries  | Festlegung der Schriftserie, wobei die aktuelle Einstellung die Vorgabe ist.   |
| frame       | Festlegung eines Rahmens um den Text. Mögliche Werte sind <code>none</code> (Vorgabe), <code>leftline</code> , <code>bottomline</code> , <code>lines</code> oder <code>single</code> . Die Linien werden im Abstand von <code>\fboxsep</code> gesetzt. |

Fortsetzung ...

... Fortsetzung

| <i>Name</i>                   | <i>Bedeutung</i>  |
|-------------------------------|---|
| <code>framerule</code>        | Dicke der Rahmenlinien (Vorgabe 0.4 pt).  |
| <code>framesep</code>         | Abstand zwischen Text und Linie (Vorgabe ist <code>\fboxsep</code> ).   |
| <code>rulecolor</code>        | Linienfarbe (Vorgabe ist <code>black</code> ).  |
| <code>fillcolor</code>        | Füllfarbe zwischen Rahmen und Text (Vorgabe ist <code>none</code> ).  |
| <code>label</code>            | Legt die Beschriftung für den Textrahmen fest, wobei <code>label</code> ein optionales Argument zugewiesen werden kann, welches dann eine Beschriftung für die obere <i>und</i> untere Rahmenlinie erlaubt. Das Argument muss dann gesondert geklammert werden: <code>label={ [Oben]Unten }</code> . Ob beide letztlich ausgegeben werden, hängt von dem Parameter <code>labelposition</code> ab. |
| <code>labelposition</code>    | Festlegung, wo die Labelbeschriftung positioniert wird. Mögliche Werte sind <code>topline</code> , <code>bottomline</code> und <code>all</code> (Vorgabe).  |
| <code>numbers</code>          | Festlegung, ob eine Zeilennummerierung eingefügt werden soll, wozu der Zähler <code>FancyVerbLine</code> verwendet wird. Mögliche Werte sind <code>none</code> , <code>left</code> und <code>right</code> .   |
| <code>numbersep</code>        | Der Abstand zwischen Zeilennummer und Text, wobei die Nummer immer außerhalb des Textes gesetzt wird.   |
| <code>firstnumber</code>      | Zeilennummer der ersten Zeile als ganze Zahl. Bei Zuweisung von <code>last</code> wird eine Zählung bei der nächsten Verbatimumgebung fortgeführt und bei <code>auto</code> (Vorgabe) beginnt jede Zählung von vorne mit Zeile 1. Ansonsten kann jeder ganze Zahlenwert zugewiesen werden.  |
| <code>stepnumber</code>       | Die Schrittweite der Nummerierung (Vorgabe ist 1).  |
| <code>numberblanklines</code> | Festlegung, ob Leerzeilen ebenfalls nummeriert werden sollen (Vorgabe ist <code>true</code> ).  |
| <code>firstline</code>        | Erste Zeile, ab der der Text angegeben wird.  |
| <code>lastline</code>         | Letzte Zeile, bis zu der der Text ausgegeben wird.  |
| <code>showspaces</code>       | Festlegung, ob Leerzeichen mit <code>\_</code> dargestellt werden sollen (Vorgabe ist <code>false</code> ).   |
| <code>showtabs</code>         | Festlegung, ob Tabulatorzeichen angezeigt werden sollen (Vorgabe ist <code>false</code> ).  |
| <code>obeytabs</code>         | Festlegung, ob die Tabulatorzeichen zur aktuellen Position addiert werden sollen oder nicht. Der Standard ist <code>false</code> .  |
| <code>baselinestretch</code>  | Streckfaktor für den Zeilenabstand, wobei die aktuelle Einstellung die Vorgabe ist.   |
| <code>commandchars</code>     | Festlegung der drei Zeichen, die ein Makro definieren, bei speziellen $\LaTeX$ -Zeichen durch einen vorausgehenden <code>\</code> zu maskieren. Das erste Zeichen ist der Steuercode, das zweite legt den Beginn des Blockes und das dritte das Ende des Blockes fest. Im Standardfall sind dies die »drei« Zeichen <code>\{ \{ \}</code> , die dann nicht im Text vorkommen dürfen.              |
| <code>xleftmargin</code>      | Zusätzlicher linker Rand (Vorgabe ist 0 pt).  |
| <code>xrightmargin</code>     | Zusätzlicher rechter Rand (Vorgabe ist 0 pt).   |
| <code>resetmargins</code>     | Zusätzliche Ränder löschen. Nur sinnvoll, wenn die Umgebung <code>Verbatim</code> Teil anderer Umgebungen ist, die ebenfalls den Rand einrücken.  |
| <code>hfuzz</code>            | Zur Verhinderung von Meldungen der Art »Überfull <code>\hbox ...</code> «, wenn der Text über den rechten Rand ragt (Vorgabe ist 2 pt).   |
| <code>samepage</code>         | Festlegung, ob innerhalb der Umgebung <code>Verbatim</code> ein Seitenumbruch nicht erlaubt ist (Vorgabe ist <code>false</code> ).  |
| <code>codes</code>            | Zur Änderung von Catcodes einzelner Zeichen.  |
| <code>defineactive</code>     | Festlegung von aktiven Zeichen.   |

## Fußnoten und das Makro `\verb`

In Abschnitt 14.2 auf Seite 680 wurde darauf hingewiesen, dass das Makro `\verb` nicht als Argument eines anderen Makros auftreten kann. Das Paket `fancyvrb` stellt jedoch ein Makro `\VerbatimFootnotes` bereit, welches dies ermöglicht.

```
\VerbatimFootnotes
```

14-02-2

Eine verb-Fußnote<sup>1</sup>

<sup>1</sup>mit `\VerbatimFootnotes` ist `\verb` in der Fußnote möglich

```
\usepackage{fancyvrb}\VerbatimFootnotes
```

```
Eine verb-Fußnote\footnote{mit
\verb|\VerbatimFootnotes| ist
\verb|\verb| in der Fußnote
möglich}
```

## Verbatim in der Zeile

Das Makro `\Verb` erlaubt im Gegensatz zum standardmäßigen `\verb` die Anwendung eines optionalen Arguments, mit dem unter anderem Schrifteinstellungen möglich sind.

```
\Verb [Optionen]
\DefineShortVerb{\<Zeichen>}
\UndefineShortVerb{\<Zeichen>}
```

In Texten mit vielen Code-Sequenzen kann durch die Definition einer Kurzform auf `\Verb` oder `\verb` verzichtet werden. Das Zeichen darf dann allerdings nicht in einem anderen Zusammenhang benutzt werden, wenn es nicht über `\UndefineShortVerb` vorher deaktiviert wurde.

14-02-3

Das Makro `\verb` kann durch eine Kurzform mit `\DefineShortVerb` vereinfacht werden|.

```
\usepackage[scaled=0.88]{helvet}
\usepackage{fancyvrb}\DefineShortVerb{\|}
```

Das Makro `|verb|` kann durch eine Kurzform mit `\Verb[fontfamily=helvetica]|DefineShortVerb|` vereinfacht `\UndefineShortVerb{\|}|werden|.`

## Verbatim als Absatz

Die Umgebung `Verbatim` wird ähnlich zum Makro `\Verb` um ein optionales Argument erweitert, mit welchem umfangreiche lokalen Festlegungen möglich sind. Daneben gibt es die Umgebungen `BVerbatim`, welche seinen Inhalt in eine Box setzt und die Umgebung `LVerbatim`, welche den Inhalt in eine horizontale LR-Box setzt. Im ersten Fall besteht die Möglichkeit die Breite der Box über `boxwidth` und die vertikale Ausrichtung durch `baseline` (mögliche Werte sind c, b und t) vorzugeben. Das Makro `\VerbatimInput` verhält sich analog zur Umgebung `Verbatim`, nur dass es auf externe Dateien angewendet werden kann.

```

\begin{Verbatim} [Optionen]
...
\end{Verbatim}
\FancyVerbFormatLine{Definitionen}
\begin{BVerbatim} [Optionen]
...
\end{BVerbatim}
\begin{LVerbatim} [Optionen]
...
\end{LVerbatim}
\VerbatimInput [Optionen] {Dateiname}

```

Die in Tabelle 14.7 auf Seite 681 zusammengefassten Parameter erlauben eine große Variation an möglichen Ausgabeformaten, die hier aus Platzgründen nicht alle angegeben werden können. Das Makro `\FancyVerbFormatLine` wird zu Beginn einer jeden Zeile ausgeführt und kann für eigene Definitionen benutzt werden.

Eine Beschriftung kann automatisch mit einer Zählung versehen werden, wenn der zugehörige Zähler über `\refstepcounter` entsprechend inkrementiert wird (siehe auch Abschnitt A.2 auf Seite 861)

\_\_\_\_\_ Lua-Beispiel 1 \_\_\_\_\_

```

#!/usr/bin/lua
lines = {
  luaH_set = 10,
  luaH_get = 24,
  luaH_present = 46}

```

Wie Beispiel 1 zeigt ...

```

\usepackage{fancyvrb}\newcounter{bsp}

\begin{Verbatim}[frame=single,fontsize=\small,
formatcom=\refstepcounter{bsp}%
  \label{Bsp-\arabic{bsp}},
label=Lua-Beispiel~\thebsp]
#!/usr/bin/lua
lines = {
  luaH_set = 10,
  luaH_get = 24,
  luaH_present = 46}
\end{Verbatim}
Wie Beispiel~\ref{Bsp-1} zeigt \ldots

```

14-02-4

Externe Dateien können mit dem Befehl `\VerbatimInput` in der gleichen Weise dargestellt werden, wie der Inhalt einer Umgebung `Verbatim`, wobei auch die jeweiligen B- und L-Varianten existieren.

\_\_\_\_\_ Lua \_\_\_\_\_

```

#!/usr/bin/lua
lines = {
  luaH_set = 10,

```

```

\usepackage{fancyvrb}

\VerbatimInput[frame=single,
firstline=1,lastline=3,
label=Lua]{foo.lua}

```

14-02-5

Ein wesentlicher Vorteil des Paketes ist die Möglichkeit, sich selbst eigene `Verbatim`-Umgebungen zu definieren, die bereits entsprechende Parametersetzungen berücksichtigen.



```
\RecustomVerbatimEnvironment{Umgebung}{Umgebung}{Key/Value-Angaben}
\DefineVerbatimEnvironment{Umgebung-Neu}{Umgebung}{Key/Value-Angaben}
```

Mit `\RecustomVerbatimEnvironment` kann man Parametersetzungen permanent zu einer bestehenden Umgebung hinzufügen, während der zweite Fall aus einer gegebenen Umgebung eine neue mit den angegebenen Parametern erstellt.

```
\usepackage{fancyvrb}
\DefineVerbatimEnvironment{FVerbatim}{LVerbatim}{numbers=left,numbersep=2mm,
  frame=lines,framerule=0.8mm,label=Lua,fontsize=\small}

\begin{minipage}{0.45\linewidth}
\begin{FVerbatim}
#!/bin/lua
-- Kommentarzeile,
\end{FVerbatim}
\end{minipage} \begin{minipage}{0.45\linewidth}
\begin{FVerbatim}[framerule=0.5pt]
#!/bin/lua
local m=1 -- Variable
\end{FVerbatim}
\end{minipage}
```

14-02-6

|  |   |
|--|---|
| <div style="border-top: 1px solid black; width: 100%; margin-bottom: 2px;"></div> <div style="display: flex; justify-content: space-between; font-size: small;"> <span>1</span> <span>#!/bin/lua</span> </div> <div style="display: flex; justify-content: space-between; font-size: small;"> <span>2</span> <span>-- Kommentarzeile,</span> </div> <div style="border-top: 1px solid black; width: 100%; margin-top: 2px;"></div> | <div style="border-top: 1px solid black; width: 100%; margin-bottom: 2px;"></div> <div style="display: flex; justify-content: space-between; font-size: small;"> <span>1</span> <span>#!/bin/lua</span> </div> <div style="display: flex; justify-content: space-between; font-size: small;"> <span>2</span> <span>local m=1 -- Variable</span> </div> <div style="border-top: 1px solid black; width: 100%; margin-top: 2px;"></div> |
|--|---|

Obiges Beispiel definiert die eigene Umgebung `FVerbatim`, die dann in jeweils eine `minipage` gesetzt wurde, um die Breite zu begrenzen. Dies hätte man auch gleich mit einer Umgebung `BVerbatim` als Grundlage machen können, jedoch erlaubt diese Umgebung keine Nummerierungen und Rahmen.

### Text in eine externe Datei schreiben

Mit der Umgebung `VerbatimOut` kann beliebiger Text in eine externe Datei geschrieben werden. Handelt es sich dabei um Programme, so können diese nach einem  $\LaTeX$ -Lauf ausgeführt werden und das Ergebnis in einem weiteren  $\LaTeX$ -Lauf eingelesen werden.

```
\begin{VerbatimOut}{Datei}
...
\end{VerbatimOut}
```

Das folgende Beispiel schreibt den Inhalt der Umgebung `VerbatimOut` in die Datei `CP/VerbatimOut.xltx`, wo sie nach einem  $\LaTeX$ -Lauf selbst mit  $\XeLaTeX$  ausgeführt und mit `pdfcrop` vom weißen Rand befreit wird.

VerbatimOut-Demo

```

\begin{VerbatimOut}{CP/VerbatimOut.xltx}
\documentclass{article}
\usepackage{pst-solides3d}
\pagestyle{empty}% keine Seitenzahl!
\begin{document}
\begin{pspicture}(-3,-3)(3,5)
\psset{viewpoint=60 80 30 rtp2xyz,Decran=60,lightsrc=viewpoint,solidmemory}
\psSolid[object=anneau,h=1.7,R=4.4,r=3.8,name=A1,action=none,hue=0 1](0,0, 0.77)
\psSolid[object=anneau,h=0.6,R=5,r=4.4, name=A2,action=none](0,0,0.46)
\psSolid[object=fusion,base=A1 A2]
\end{pspicture}
\end{document}
\end{VerbatimOut}

```

Über das Makro `\IfFileExists` kann abgefragt werden, ob eine Datei `VerbatimOut.pdf` existiert. Wenn ja, dann wird sie eingelesen:

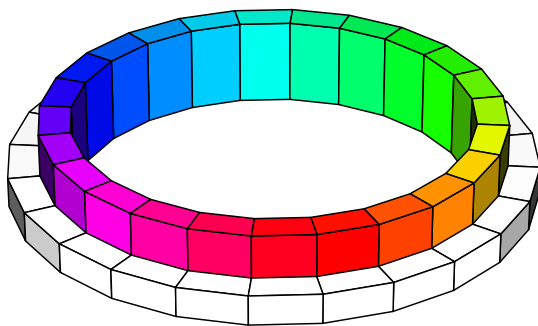


Abbildung 14.1: Das Ergebnis der Beispieldatei `VerbatimOut.xltx` (erstellt mit Xe<sub>La</sub>TeX)

Auf diese Weise lassen sich beliebige Programme aus eine L<sup>A</sup>T<sub>E</sub>X-Dokument heraus erstellen, extern ausführen und anschließend wieder mit ihrem Ergebnis in das Dokument einfügen. [52]

### 14.2.2 Das Paket `alltt`

Das Paket `alltt` von Johannes Braams stellt die gleichnamige Umgebung `alltt` bereit, die der Umgebung `verbatim` ähnlich ist, jedoch L<sup>A</sup>T<sub>E</sub>X-Makros grundsätzlich ausführt. Dies gilt jedoch nicht für die aktiven Zeichen `$`, `^`, `~`, usw.; sie werden als normale Zeichen behandelt.

```

\begin{alltt}
...
\end{alltt}

```

Soll ein Text innerhalb der Zeile im Mathematikmodus gesetzt werden, so geht dies nicht mit `$...$`. Alternativen sind die Sequenz `\(...\)` oder die Umgebung `math`. Der Vorteil der Umgebung `alltt` ist, dass sie auch in einer Fußnote verwendet werden kann, was mit den normalen `Verbatim`-Umgebungen nicht ohne weiteres möglich ist.

14-02-7

Text mit  
 SONDERZEICHEN  
 / " \$ \$ \$ % & ö ä ü ß #  
 und  $\LaTeX$ -Makros!  $E = mc^2$

foo<sup>1</sup>

<sup>1</sup>Auch hier geht eine Umgebung alltt:

$E = mc^2$

```
\usepackage{alltt}

\begin{alltt}
\emph{Text} mit {\rmfamily
\textsc{Sonderzeichen}}
/ " $ $ $ % & ö ä ü ß #
und \LaTeX-Makros! \((E=mc^2\))
\end{alltt}

foo\footnote{Auch hier geht eine
Umgebung \texttt{alltt}:
\begin{alltt}\(E=mc^2\)
\end{alltt}}
```

## 14.3 Programmlistings mit dem Paket listings

Mit dem Paket listings von Carsten Heinz und Brooks Moses lässt sich Quellcode von  $\LaTeX$  oder anderen Programmiersprachen formatiert ausgeben.

```
\begin{lstlisting} [Optionen]
...
\end{lstlisting}
\lstinputlisting [Optionen] {Datei}
\lstinline [Optionen] <Zeichen>...<Zeichen>
\lstset{Parameter}
```

Der Inhalt von der Umgebung `lstlisting`, das Argument von `\lstinline` und die externe Datei von `\lstinputlisting` werden »verbatim« gelesen; *alle* Zeichen haben die Bedeutung eines normalen Textzeichens, solange nichts anderes vereinbart wird. So wird beispielsweise ein Backslash nicht als Einleitung zu einem Makro interpretiert, wie in diesem Fall: `\clearpage`. Wird der Code innerhalb der Zeile ausgegeben so ist dies nur mit dem Makro `\lstinline` zu erreichen: `\lstinline|\clearpage|`. Die begrenzenden Zeichen dürfen keine Buchstaben sein, ansonsten können beliebige gewählt werden, solange es sich nicht um den Backslash handelt.

Die Umgebung `lstlisting` muss zwingend mit `\end{lstlisting}` am Anfang einer Zeile beendet werden, ansonsten kann das Paket das Ende der Umgebung nicht eindeutig bestimmen.

14-03-1

```
\begin{itemize}
\item\today
\end{itemize}
```

```
\usepackage[scaled=0.85]{beramono}
\usepackage{listings}

\begin{lstlisting}[basicstyle=\small\ttfamily,
keywordstyle=\bfseries,language={\LaTeX}TeX}]
\begin{itemize}
\item\today
\end{itemize}
\end{lstlisting}
```