

Kapitel 9

Pakete

9.1 fontenc	183
9.2 inputenc	184
9.3 graphicx	184
9.4 caption	186
9.5 hvfloat	194
9.6 Subfloats	202
9.7 hyperref	204
9.8 array	209
9.9 xcolor	213

In diesem Kapitel werden nur ein paar ausgewählte Pakete behandelt, deren Anwendung mittlerweile Standard ist und faktisch per Definition von jedem zu erstellenden Dokument geladen werden können. Trotzdem ist auch diese kurze Zusammenstellung willkürlich, denn die Grenze zu den weniger häufig angewendeten Paketen ist fließend.

9.1 fontenc

```
\usepackage[Fontkodierung]{fontenc}
```

Bei Anwendung von pdfL^AT_EX können Unicode-Schriften nicht direkt genutzt werden, sodass man wegen der Limitierung von 256 Zeichen pro Schrift auf unterschiedliche Kodierungen angewiesen ist. Standardmäßig wird man T1 verwenden, die auch entsprechende Zeichen für korrekte Trennungen von Wörtern mit Umlauten bereitstellt und nichts mit den Type1-Schriften von PS zu tun hat. Die allgemein gültigen Schriftkodierungen enthält Tabelle 8.1 auf Seite 158.

Aufgrund der Tatsache, dass fast nur Vektorschriften verwendet werden, sollte das Paket `fontenc` mit der korrekten Schriftkodierung standardmäßig geladen werden. Im Allgemeinen mit der Option `T1`.

9.2 inputenc

Mit diesem Paket wird die sogenannte Eingabekodierung festgelegt, was sich lediglich auf die nicht im Standard-ASCII vorhandenen Zeichen bezieht.

```
\usepackage[Encoding-Name]{inputenc}
\inputencoding{Encoding-Name}
```

Das Makro `\inputencoding` gestattet das Umschalten der Kodierung für einzelne Textpassagen, beispielsweise für mehr oder weniger kurze CJK-Sequenzen (Chinese–Japanese–Korean).

Die wichtigsten Kodierungen für das Input-Encoding sind

- `ascii` ASCII für den Zeichenbereich 32–127.
- `latin1` ISO Latin-1 Encoding.
- `latin9` ISO Latin-9 Encoding (mit Euro-Zeichen).
- `utf8` 8-bit Unicode Transformation Format (L^AT_EX-Standard).
- `applemac` Macintosh Encoding.
- `ansinew` Windows ANSI Encoding, Erweiterung von Latin-1 (synonym für `cp1252`)

Weitere kann man der Dokumentation zum Paket entnehmen → `texdoc inputenc`. Insbesondere im Zusammenhang mit einer Mehrautorenschaft kann es zu Problemen kommen, wenn verschiedene Autoren unterschiedliche Eingabekodierungen verwenden. Das Programm `recode` (oder ein vergleichbares) gestattet ein einfaches Umdefinieren in das allgemeingültige L^AT_EX-Format, welches nur Makros verwendet:

```
recode --diacritics --touch --verbose latin1..LaTeX <Dateiname>
```

In diesem Beispiel würde eine `latin1`-Kodierung nach L^AT_EX konvertiert, also beispielsweise `ä→\ "a`. Alternativ hätte man mit `\inputencoding` eine Umschaltung vornehmen können.

Standard

Die standardmäßige Eingabekodierung ist für pdfL^AT_EX `utf8`, sodass diese nicht gesondert angegeben werden muss. Für X_LL^AT_EX und LuaL^AT_EX ist die Kodierung per Definition `utf8`, sodass das Laden von `inputenc` überflüssig ist.

9.3 graphicx

9.3.1 \includegraphics

Der Befehl `\includegraphics*` benötigt das Paket `graphicx`, welches selbst automatisch das Paket `graphics` lädt. Nach dem Laden steht dann unter anderen der folgende Befehl zur Verfügung:

```
\usepackage [Optionen] {graphicx}
...
\includegraphics* [Optionen] {Dateiname}
```

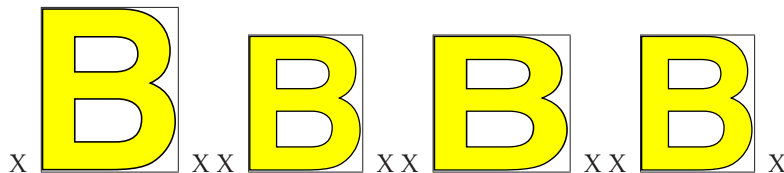
- Eine Angabe des optionalen Treibers kann in der Regel entfallen, da die Konfigurationsdatei den verwendeten Modus selbst feststellen kann, beispielsweise dvips oder pdftex
- Der Dateiname sollte ohne Dateiendung angegeben werden, dann kann der Treiber die richtige auswählen, beispielsweise mit Endung .eps, wenn der L^AT_EX-Modus benutzt wird. Enthält der Dateiname selbst einen Punkt oder andere spezielle Zeichen, dann sollte das Paket grffile geladen werden.
- Ohne Angabe eines Pfades sucht T_EX erst im Dokumentenverzeichnis, dann im lokalen T_EX-Baum und danach im normalen T_EX-Baum.
- Über \graphicspath oder die Umgebungsvariable TEXINPUTS kann T_EX ein weiterer Suchpfad mitgeteilt werden.
- Die Sternversion beschneidet die Grafik entsprechend der Bounding Box.

Die Beispiele beziehen sich alle auf dieselbe Ausgangsgrafik (siehe Beispiel 09-03-2).

```
\usepackage{graphicx}
```

```
X\_\fbox{\includegraphics[width=2cm]{B}}\_X
X\_\fbox{\includegraphics[height=2cm]{B}}\_X
X\_\fbox{\includegraphics[width=2cm,height=2cm,keepaspectratio=false]{B}}\_X
X\_\fbox{\includegraphics[width=2cm,height=2cm,keepaspectratio]{B}}\_X
```

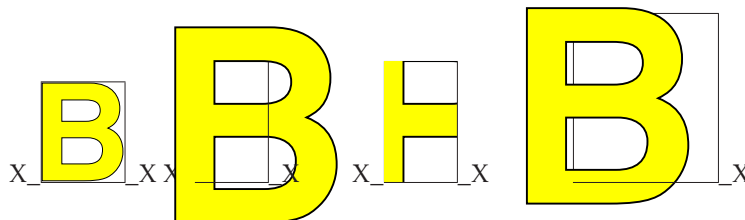
09-03-1



```
\usepackage{graphicx}
```

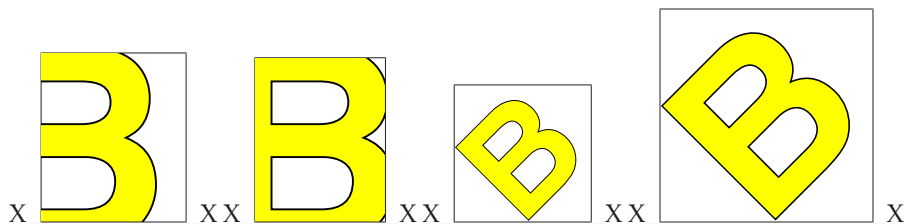
```
X\_\fbox{\includegraphics[scale=0.5]{B}}\_X
X\_\fbox{\includegraphics[trim=9 18 30 15]{B}}\_X\quad
X\_\fbox{\includegraphics[trim=9 18 30 15, clip]{B}}\_X\quad
X\_\fbox{\includegraphics[viewport=20 10 80 80]{B}}\_X
```

09-03-2



```
\usepackage{graphicx}
```

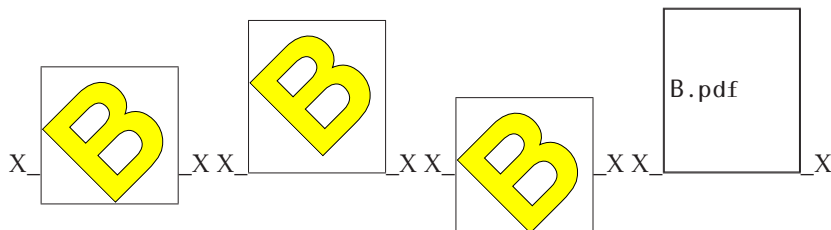
```
X\_\fbox{\includegraphics[viewport=20 10 80 80,clip]{B}}\_X
X\_\fbox{\includegraphics[trim=10 10 5 5, clip]{B}}\_X
X\_\fbox{\includegraphics[angle=45, width=2cm]{B}}\_X
X\_\fbox{\includegraphics[width=2cm, angle=45]{B}}\_X
```



09-03-3

```
\usepackage{graphicx}
```

```
X\_\fbox{\includegraphics[origin=rt,angle=45,width=2cm]{B}}\_X
X\_\fbox{\includegraphics[origin=lc,angle=45,width=2cm]{B}}\_X
X\_\fbox{\includegraphics[origin=rb,angle=45,width=2cm]{B}}\_X
X\_\fbox{\includegraphics[draft,width=2cm]{B}}\_X
```



09-03-4

Weist der Dateiname der Grafik Zeichen auf, die für $\TeX$ keinen normalen alphanumerischen Zeichen entsprechen, so kann mit dem Paket `grffile` von Heiko Oberdiek das Problem behoben werden. Damit sind dann auch mehrere Punkte oder Leerzeichen in Dateinamen möglich.

9.4 caption

Das Paket `caption` von Axel Sommerfeld sollte grundsätzlich geladen werden, da es den Satz von Bild- und Tabellenbeschriftungen erheblich verbessert; die Möglichkeiten, die Standard- $\LaTeX$ hier bietet, sind sehr eingeschränkt.

KOMA-Script

Bei Verwendung einer der KOMA-Script-Klassen kann auf das Laden von `caption` verzichtet werden, da KOMA-Script im Wesentlichen alle Einstellungen auch durch eigene Makros erlaubt. Grundsätzlich besteht aber kein Problem bei der Anwendung von `caption` mit einer Klasse von KOMA-Script.

Die Einstellungen für die Beschriftungen können über das optionale Argument von `\usepackage` oder mit dem Makro `\captionsetup` verändert werden. Innerhalb einer Gleitumgebung bleiben diese Angaben lokal, anderenfalls gelten sie solange für alle folgenden

`\caption`, bis eine weitere Änderung erfolgt. Das Paket `caption` hat keinerlei Auswirkungen auf den Ort der Beschriftung; soll diese beispielsweise oberhalb einer Tabelle erscheinen, so muss sie auch *vor* der Tabelle gesetzt werden. Eine automatische Anordnung ist mit dem Paket `floatrow` möglich. Die Optionen, die nur in der Präambel eines Dokumentes erscheinen dürfen, zeigt Tabelle 9.1. Diese können als Teil des optionalen Arguments von `\usepackage` oder als Parameter von `\captionsetup` gesetzt werden.

```
\captionsetup [Typ] {Einstellungen}
\caption [Verzeichniseintrag] {Beschriftung}
\caption*{Beschriftung}
\captionof{Typ} [Verzeichniseintrag] {Beschriftung}
\captionof*{Typ}{Beschriftung}
```

- Die Sternversionen verhindern jeweils einen Eintrag in das entsprechende Verzeichnis. Dasselbe erreicht man mit einem leeren optionalen Argument, beispielsweise `\caption[]`{Eine Tabelle}.
- Mit `\captionof` kann unabhängig vom Typ der übergeordneten Gleitumgebung ein anderer Typ erzwungen werden.

Tabelle 9.1: Zusammenstellung der Paketoptionen für `caption`, die nur in der Präambel auftreten dürfen.

Name	Vorgabe	Bedeutung
<code>compatibility</code>	<code>true</code>	Kompatibel zu anderen Paketen bleiben.
<code>figurename</code>	<code>\figurename</code>	Labelnamen festlegen, der in der Regel von der Dokumentenklasse vorgegeben wird.
<code>figureposition</code>	–	Platzierungsparameter für Objekte in einer Umgebung <code>figure</code> . Die Vorgabe ist abhängig von der Dokumentenklasse und Paketen.
<code>figurewithin</code>	–	Aktiviert übergeordnete Zähler für die Umgebung <code>figure</code> .
<code>tablename</code>	<code>\tablename</code>	Labelnamen festlegen, der in der Regel von der Dokumentenklasse vorgegeben wird.
<code>tableposition</code>	–	Platzierungsparameter für Objekte in einer Umgebung <code>table</code> . Die Vorgabe ist abhängig von der Dokumentenklasse und Paketen.
<code>tablewithin</code>	–	Aktiviert übergeordnete Zähler für die Umgebung <code>table</code> .

Das Makro `\caption` darf nur innerhalb einer Gleitumgebung auftreten, anderenfalls gibt es eine Fehlermeldung »! LaTeX Error: `\caption` outside float.«. Dagegen ist `\captionof` sowohl innerhalb als auch außerhalb einer Gleitumgebung erlaubt, sodass damit auch Beschriftungen von nicht-gleitenden Objekten möglich sind.

 `\caption`
versus
`\captionof`

Eine Zusammenstellung der möglichen Optionen, die sowohl global in der Präambel, als auch lokal im Text durch `\captionsetup` verändert werden können, zeigt Tabelle 9.2 auf Seite 192.

9.4.1 Globale Formatierung der Beschriftung

Eine Beschriftung besteht prinzipiell aus drei Teilen, dem Bezeichner, dem Trenner und dem Text. Die globale Ausrichtung kann über die Option `format` gesteuert werden. Zulässige Werte sind:

- plain** Standard-L^AT_EX-Verhalten; die Beschriftung wird ohne weitere Formatierung als eigener Absatz gesetzt. Der Einzug ab der zweiten Zeile kann über die Option `indention` geändert werden, wobei positive und negative Einzüge möglich sind.
- hang** Der Beschriftungstext wird als eigener Absatz neben Bezeichner und Trenner gesetzt. Der Einzug ab der zweiten Zeile kann über die Option `indention` geändert werden, beispielsweise `indention=-2em`. Die Werte beziehen sich auf den Beginn des Beschriftungstextes.
- ... Für eigene Formate reserviert, die mit `\DeclareCaptionFormat` definiert werden können.

```
\DeclareCaptionFormat{Name}{Code mit #1, #2 und #3}
```

Die Definition muss zwingend in der Präambel erfolgen und *kann* die drei Parameter #1 für den Bezeichner, #2 für den Trenner und #3 für den Text verwenden. Der *Name* kann willkürlich gewählt werden, sollte aber sinnvollerweise nicht bereits vergeben sein.

9.4.2 Label

Das Label, welches sich aus dem Bezeichner und der fortlaufenden Nummer zusammensetzt, wird so gesetzt, wie es die verwendete Dokumentenklasse vorgibt, in der Regel einfach der Bezeichner und die durch ein Leerzeichen getrennte Nummer. Dieses Verhalten kann über die Option `labelformat` geändert werden. Mögliche Werte sind:

- default** Der Bezeichner wird so gesetzt, wie es die verwendete Dokumentenklasse vorsieht.
- empty** Der Bezeichner wird nicht ausgegeben; es erscheint nur die Nummer.
- simple** Das Label ist ohne weitere Formatierung aus dem Bezeichner und der Nummer zusammengesetzt.
- brace** Der Bezeichner wird mit einer einzelnen (rechten) Klammer abgeschlossen.
- parens** Die Nummer des Labels wird in runde Klammern gesetzt.
- ... Für eigene mit `\DeclareCaptionLabelFormat` definierte Werte reserviert.

```
\DeclareCaptionLabelFormat{Name}{Code mit #1 und #2}
```

Der Abstand zwischen Bezeichner und Nummer kann mit der Option `labelsep` beeinflusst werden. Mögliche Werte sind:

- none** Es wird kein Trenner vorgesehen.
- colon** Der Trenner besteht aus einem Doppelpunkt und einem folgenden Leerzeichen.
- period** Der Trenner besteht aus einem Punkt und einem folgenden Leerzeichen.
- space** Der Trenner besteht nur aus einem folgenden Leerzeichen. Ist kein Bezeichner vorhanden, so entfällt es.
- quad** Der Trenner besteht aus dem Makro `\quad`.
- newline** Der Trenner besteht aus dem Zeilenumbruch, wobei dieser nicht mit dem Format `hang` funktioniert.
- endash** Als Trenner wird der Gedankenstrich – verwendet.
- ... Für eigene mit `\DeclareCaptionLabelSeparator` definierte Werte reserviert.

```
\DeclareCaptionLabelSeparator{Name}{Code}
```

9.4.3 Beschriftungstext

Mit der Option `textformat` wird nicht die eigentliche Formatierung des Beschriftungstextes bezeichnet, sondern das, was an das Ende des Textes angehängt werden soll. Damit kann man sicherstellen, dass beispielsweise alle Beschriftungen grundsätzlich mit einem Punkt enden. Mögliche Werte sind:

- simple** Es wird keine Veränderung vorgenommen.
- period** Der Text wird mit einem Punkt abgeschlossen.
- ... Für eigene mit `\DeclareCaptionTextFormat` definierte Werte reserviert.

```
\DeclareCaptionTextFormat{Name}{Code mit #1}
```

Der Parameter #1 steht hier für den Beschriftungstext.

Die Ausrichtung des Beschriftungstextes, beziehungsweise die Absatzformatierung, kann mit der Option `justification` festgelegt werden. Zulässige Werte sind:

- justified** Der standardmäßige Blocksatz.
- centering** Der Text wird ohne Trennungen zentriert.
- centerlast** Die letzte Zeile des Textes wird zentriert.
- centerfirst** Die erste Zeile des Textes wird zentriert.
- raggedright** Der Text wird ohne Trennungen linksbündig gesetzt.
- RaggedRight** Der Text wird mit Trennungen linksbündig gesetzt, wozu das Paket `ragged2e` geladen wird. Ob dieses Paket benötigt wird oder nicht, wird von `caption` zur Laufzeit ermittelt, sodass ein weiterer L^AT_EX-Lauf erforderlich sein kann.
- raggedleft** Der Text wird ohne Trennungen rechtsbündig gesetzt.
- ... Für eigene mit `\DeclareCaptionJustification` definierte Werte reserviert.

Einzeilige Beschriftungen werden in Standard-L^AT_EX automatisch zentriert, im Gegensatz zu mehrzeiligen, die immer im Blocksatz gesetzt werden. Das Paket `caption` ermöglicht das Unterbinden der Überprüfung des Beschriftungstextes, sodass dieser auch bei nur einer Zeile den Parameter `justification` auswertet. Notwendig ist dazu die Option `singlelinecheck` auf `false` zu setzen, wobei auch alternativ die Werte `no`, `off` und `0` möglich sind.

Die Zeichenformatierung erfolgt über drei optionale Argumente. `font` ist die übergeordnete Option und gilt für die gesamte Beschriftung: Bezeichner, Nummer und Beschriftungstext. `labelfont` gilt nur für den Bezeichner und Trenner und `textfont` gilt nur für den Beschriftungstext. Beide überschreiben dabei jeweils die Vorgaben aus der übergeordneten Option `font`. Die durch das »+«-Zeichen ergänzten alternativen Optionen dienen zum Erweitern bestehender Definitionen. Alle drei, beziehungsweise sechs Varianten, können die folgenden Werte annehmen, wobei Kombinationen durch Komma zu trennen und eventuell in {...} zu setzen sind:

Standardvorgabe Mit `normal` kann der Standard aus den Kombinationen `normalcolor`, `normalfont`, `normalsize` und `singlespacing` gesetzt werden.

Schriftgröße `scriptsize`, `footnotesize`, `small`, `normalsize`, `large` und `Large`

Schrift	normalfont	Normale Schriftart	up	Aufrechte Schriftart
	it	<i>Italic Schriftart</i>	sl	<i>Slanted Schriftart</i>
	sc	SMALL CAPS SCHRIFTART	md	Medium Schriftserie
	bf	Fette Schriftserie	rm	Roman Schriftfamilie
	sf	Sans serif Schriftfamilie	tt	Typewriter Schriftfamilie

Zeilenabstand singlespacing, onehalfspacing, doublespacing und stretch, wobei letzterem ein Wert zugewiesen werden muss! Wird das Paket setspace geladen, so ist für Beschriftungen automatisch der Modus *einzeilig* eingestellt.

Farbe normalcolor oder color=<Farbe>. Setzt das Paket color oder xcolor voraus.

... Für eigene mit \DeclareCaptionFont definierte Werte reserviert.

Alle Kombinationen können beliebig miteinander kombiniert werden, solange die aktuelle Schrift die Kombinationen auch unterstützt.

margin Die Ränder können alternativ über die explizite Angabe durch margin oder implizite
width Angabe durch width (Textbreite der Beschriftung) beeinflusst werden. Die Werte müssen
oneside absolute Längenangaben sein, relative Werte wie Breite der Tabelle oder Abbildung sind nicht möglich. Dies kann jedoch mit dem Paket floatrow oder auch threeparttable erfolgen. Wird nur ein Wert für margin angegeben, so bezieht er sich automatisch auf die dann symmetrischen Ränder links und rechts. Zwei getrennte Werte für links und rechts sind durch Komma getrennt in {...} zu setzen, beispielsweise margin={0pt, 10pt}. Bei
oneside zweiseitigen Dokumenten und unterschiedlichen Rändern für Innen und Außen, werden
twoside diese automatisch an gerade und ungerade Seiten angepasst. Dieses Verhalten kann mit der Option oneside abgeschaltet und mit twoside wieder aktiviert werden. Immer dann, wenn die Breite der Abbildung oder Tabelle bekannt ist, kann man dieselbe Breite für die Beschriftung wählen.

9.4.4 Horizontale Abstände

*margin** Die Option margin* setzt die Ränder nur dann, wenn nicht vorher schon mit width eine
minmargin Textbreite gesetzt wurde, die entsprechende Ränder zur Folge hatte. Bei global gesetzten
maxmargin Rändern kann es Probleme geben, wenn zwischenzeitlich in einen Mehrspaltensatz gewechselt wird. Für diese Fälle können minmargin und maxmargin minimale und/oder maximale Ränder vereinbart werden, beispielsweise maxmargin=0.1\linewidth.

parskip Das Paket caption erlaubt als Beschriftungstext auch mehrere Absätze, was bei Standard-
hangindent L^AT_EX nicht möglich ist. Für das Verzeichnis der Gleitumgebung ist allerdings über das optionale Argument von \caption ein Ersetzungstext ohne Absätze vorzusehen. Der Absatzabstand kann über die Option parskip und der hängende Einzug nach der ersten Zeile durch hangindent festgelegt werden.

9.4.5 Vertikale Abstände

Der standardmäßige Wert für den Abstand \belowcaptionskip, beziehungsweise entsprechend \abovecaptionskip, kann über die Option skip verändert werden. Die Position dieses Abstandes wird mit position festgelegt. Zulässige Werte sind top oder above (Objekt oben – Beschriftung unten) bottom oder below (Objekt unten – Beschriftung oben)

und auto. Zu beachten ist, dass sich die Angabe der Position nur auf skip bezieht, nicht jedoch auf die Anordnung der Beschriftung; sie erscheint analog zur Position im Quelltext. Eine Automatik ist nur mit Paketen wie floatrow oder threeparttable möglich. Bei auto wird lediglich *versucht*, den Wert für skip korrekt auszuwerten. Eine Zuordnung zu einem bestimmten Typ von Gleitumgebung kann über das optionale Argument von \captionsetup erfolgen, beispielsweise:

```
\captionsetup[table]{position=below} .
```

Damit diese Möglichkeit auch schon über das optionale Argument von \usepackage erfolgen kann, gibt es die Optionen figureposition und tableposition.

```
\usepackage[tableposition=below]{caption} .
```

Die Festlegung des Abstandes durch das Paket caption ist abhängig vom Verhalten der jeweiligen Dokumentenklasse und anderer Pakete, die Gleitumgebungen tangieren und Einfluss auf die beiden entscheidenden Längen \belowcaptionskip und \abovecaptionskip nehmen. Es ist daher nicht ungewöhnlich, wenn es Probleme bei der eindeutigen Zuordnung des Abstandes gibt.

9.4.6 Stile

Mehrere Optionen können zu einem Stil zusammengefasst werden, um damit eine einfachere Einstellung von Optionen zu erreichen. Zwei Stile sind bereits vom Paket caption definiert:

base: format=plain, labelformat=default, margin=0pt, labelsep=colon, font={}, labelfont={}, justification=justified, textfont={}, indentation=0pt, parindent=0pt, hangindent=0pt, singlelinecheck=true.

default: format=default, labelformat=default, margin=0pt, font=default, labelsep=default, textfont=default, indentation=0pt, labelfont=default, parindent=0pt, hangindent=0pt, justification=default, singlelinecheck=true.

Unabhängig von der Wahl des Stils wird automatisch justification=centering und indentation=0pt gewählt werden, wenn die Beschriftung in eine einzelne Zeile passt und der Anwender nichts anderes vereinbart hat. Dies kann über das Argument *zusätzliche Optionen* beim Makro \DeclareCaptionStyle geschehen.

```
\DeclareCaptionStyle{Name}[zusätzliche Optionen]{Optionen}
```

Alle neu definierten Stile basieren immer auf dem Stil base, sodass nur davon abweichende Werte anzugeben sind. Die in *zusätzliche Optionen* definierten Werte kommen nur dann für einzelilige Beschriftungen zum Tragen, wenn nicht mit singlelinecheck=false der Test darauf ausgeschaltet wurde.

9.4.7 Verzeichnisse

Es wurde bereits darauf hingewiesen, dass man mit einem leeren optionalen Argument bei \caption einen Eintrag in das Verzeichnis der jeweiligen Gleitumgebung verhindern *list*
listformat
Paket subfig

kann. Soll dies global für das Dokument oder ab einem bestimmten Punkt gelten, kann dies mit der Option `list` erfolgen, wenn dieser der Wert `false`, `no` oder `off` zugewiesen wird. Mit `list=true` kann wieder das normale Verhalten erreicht werden. Das Paket `subfig` unterstützt diese Option nicht. Stattdessen sind die Zähler `lofdepth` und/oder `lotdepth` auf Null zu setzen. Das Format des Verzeichnisses wird über die Option `listformat` gesteuert. Folgende Werte sind möglich:

- empty** Es wird keine Nummer der Gleitumgebung angegeben.
- simple** Die Nummer wird (mit Zähler-Präfix) angegeben.
- parens** Die Nummer wird (mit Präfix) in Klammern angegeben.
- subsimple** Wie `simple`, aber ohne ein Präfix (Standard).
- subparens** Wie `parens`, aber ohne ein Präfix.
- ...** Eigenes Listenformat mit dem Makro `\DeclareCaptionListFormat` definieren.

Werden nicht mit dem Paket `subcaption` oder `subfig` Unterabbildungen erzeugt, ist im Normalfall kein Präfix vorgesehen, so dass `simple` und `subsimple`, sowie `parens` und `subparens` gleiche Ergebnisse liefern. Im Gegensatz zu den Beschriftungen selbst, erfolgt in Verzeichnissen keine Angabe des Typs, wie beispielsweise »Abbildung«, da sich dies bereits aus der Verzeichnisüberschrift ergibt.

```
\DeclareCaptionListFormat{Name}{Code mit #1 und #2}
```

Im Laufe des Dokumentes wird #1 dann mit dem Bezeichner-Präfix (z. B. `\p@figure` – Abbildung), und #2 mit der Referenznummer (z. B. `\thefigure`) ersetzt.

Tabelle 9.2: Zusammenstellung der optionalen Argumente für das Paket `caption`, die lokal über das Makro `\captionsetup` verändert werden können. Für *boolsche* Variablen sind die Werte `true`, `yes`, `on` oder `1`, beziehungsweise `false`, `no`, `off` oder `0`, möglich.

Name	Vorgabe	Bedeutung
aboveskip	<code>\abovcaptionskip</code>	Abstand oberhalb der Beschriftung.
belowskip	<code>\belowcaptionskip</code>	Abstand unterhalb der Beschriftung.
font	–	Globale Schrift für die Beschriftung. Mögliche Werte sind <code>scriptsize</code> , <code>footnotesize</code> , <code>normalsize</code> , <code>large</code> , <code>Large</code> , <code>normalfont</code> , <code>up</code> , <code>it</code> , <code>sl</code> , <code>sc</code> , <code>md</code> , <code>bf</code> , <code>rm</code> , <code>sf</code> , <code>tt</code> , <code>singlespacing</code> , <code>onehalfspacing</code> , <code>doublespacing</code> , <code>stretch=<Wert></code> , <code>normalcolor</code> , <code>color=<Farbe></code> und <code>normal</code> .
font+	–	Zusätzliche Option zur aktuellen Festlegung. Mögliche Werte siehe <code>font</code> .
format	<code>plain</code>	Formatierung der Beschriftung, mögliche Werte sind <code>plain</code> und <code>hang</code> .
FPList	<code>caption</code>	Angabe des Verweises bei einer Umgebung <code>FPfigure</code> oder <code>FPtable</code> . Mögliche Werte sind <code>caption</code> und <code>figure</code> .
FPref	<code>figure</code>	dito, für eine Referenz.
hangindent	<code>0pt</code>	Setzt den Wert für den hängenden Einzug nach der ersten Zeile.
hypcap	<code>true</code>	Setzt den Anker für <code>hyperref</code> auf den Anfang der Gleitumgebung.
hypcapspace	<code>0pt</code>	Vertikaler Abstand zwischen Link und Objekt.
indention	<code>0pt</code>	Setzt den Wert für den hängenden Einzug.

Fortsetzung ...

... Fortsetzung

<i>Name</i>	<i>Vorgabe</i>	<i>Bedeutung</i>
justification	justified	Horizontale Formatierung der Beschriftung. Mögliche Werte sind justified, centering, centerlast, centerfirst, raggedright, RaggedRight und raggedleft.
labelfont	–	Formatierung des Labels, Werte siehe font.
labelfont+	–	Formatierung des Labels, Werte siehe font+.
labelformat	default	Formatierung des Labels, mögliche Werte sind default, empty, simple, brace und parens.
labelsep	colon	Zeichen zwischen Label und Beschriftungstext. Mögliche Werte sind none, colon, period, space, quad, endash und newline.
labelseparator	colon	Identisch zu labelsep.
list	on	Übernahme der Beschriftungen in die Listen.
listfigurename	–	Überschrift der Liste der Abbildungen, wobei die Vorgabe von der verwendeten Dokumentenklasse abhängt.
listformat	subsimple	Formatierung der Nummer in der Liste der Abbildungen und der der Tabellen. Mögliche Werte sind empty, simple, parens, subsimple, subparens.
listformat	subsimple	Formatierung der Nummer in der Liste der Abbildungen und der der Tabellen. Mögliche Werte sind empty, simple, parens, subsimple, subparens.
listtablename	–	Analog zu listfigurename.
margin	Opt	Festlegung des linken und rechten Randes der Beschriftungen. Bei einer einzelnen Angabe werden symmetrische Ränder gewählt. Zwei Angaben für {Links,Rechts} sind durch Komma getrennt in geschweifte Klammern zu setzen.
margin*	Opt	Ränder werden nur gesetzt, wenn width nicht definiert ist.
maxmargin	–	Maximaler Rand
minmargin	–	Minimaler Rand
name	–	Analog zu figurename und tablename, jedoch nur gültig für die aktuelle Gleitumgebung.
oneside	–	Setzt die Dokumentenoption auf einseitig.
parindent	\parindent	Setzt den Absatzeinzug
parskip	\parskip	Setzt den dynamischen Absatzabstand
position	auto	Angabe über die Position der Beschriftung. Mögliche Werte sind d, default, t, top, above, b, bottom, below, a und auto.
singlelinecheck	true	Besondere Behandlung von einzeiligen Beschriftungen.
skip	10pt	Abstand zwischen Beschriftung und Objekt.
strut	on	Anwendung des Makros \struts.
style	–	Zuweisung eines definierten Anwenderstils.
subtype	–	Setzt den Typ einer Teil-Gleitabbildung.
textfont	–	Setzt die Schrift für den Beschriftungstext (siehe font).
textfont+	–	Zusätzliche Schriftattribute (siehe font+).
textformat	plain	Format der Beschriftung, mögliche Werte sind plain und hang.
twoside	–	Setzt die Dokumentenoption auf zweiseitig.
type	–	Setzt den aktuellen Typ der Gleitumgebung und einen Hyperlink.
type*	–	dito, nur ohne Hyperlink.
width	\linewidth	Setzt die Breite einer Beschriftung.

9.5 hvfloat

Das Paket hvfloat unterstützt die Anordnung von Abbildung/Tabelle und Beschriftung und kann mit den Optionen fbox (zu Testzwecken jeweils ein Rahmen um Abbildung und Caption) und hyperref (automatisches Laden von hypcap) geladen werden. hvfloat selbst lädt die Pakete caption, subcaption, atbegshi, expl3, multido, graphicx, xkeyval, ifoddpage und afterpage.

Die eigentliche Formatierung einer Beschriftung wird hierbei nicht beeinflusst und ist daher mit einem der hier angegebenen Pakete vorzunehmen, wobei hvfloat standardmäßig caption und subcaption benutzt. Mit hvfloat lässt sich eine getrennte oder gemeinsame Rotation von Objekt und Caption erreichen. Auch ganzseitige oder spaltenübergreifende Abbildungen werden unterstützt.

9.5.1 Einfache Gleitumgebungen

Hierunter versteht man Gleitumgebungen, die jeweils nur *eine* Beschriftung für ein oder mehrere Objekte aufweisen.

```
\hvFloatSet{key=value list}
\setDefault
\defhvstyle{name}{setting}
\hvFloat * [Optionen] + {Typ}{Gleitobject} [CAption-Kurz] {Caption-Lang}{Label}
```

Tabelle 9.3: The optional keywords for the macro \hvFloat

Option	Vorgabe	Beschreibung
floatPos	htb	Positionierungsparameter der Gleitumgebungen.
rotAngle	0	Rotation sowohl für das Objekt als auch die Beschriftung.
capWidth	n	Breite der Beschriftung. Kann die Werte »n« für natürliche Breite, »w« für die Breite des Objekts, »h« für die Höhe des Objekts oder eine Dezimalzahl für die relative Skalierung in Bezug auf \columnwidth.
capAngle	0	Rotation der Beschriftung (im mathematischen Sinn).
capPos	before	Anordnung der Beschriftung relativ zum Objekt: before: immer vor (links) dem Objekt; left: immer vor (links) dem Objekt, im Zweispaltensatz aber auf derselben Seite wie das Objekt; after: immer nach (rechts) dem Objekt; right: immer nach (rechts) dem Objekt, im Zweispaltensatz aber auf derselben Seite wie das Objekt; inner: im twoside-Modus in der inneren Spalte setzen. outer: im twoside-Modus in der äußeren Spalte setzen. evenPage: im twoside-Modus und der Option fullpage die Beschriftung auf eine gerade Seite setzen. oddPage: dito., aber ungerade Seite.
capVPos	c	Nur interessant für capPos=left right, nur dann kann die Beschriftung vertikal unterschiedlich angeordnet werden: bottom, center und top.
objectPos	center	Die horizontale Anordnung des Objekts relativ zur Textbreite. Mögliche Werte sind (l)eft (c)enter (r)ight.
objectAngle	0	Rotation des Objekts (im mathematischen Sinn).
floatCapSep	5pt	Abstand zwischen Objekt und Beschriftung (falls nebeneinander).

Option	Vorgabe	Beschreibung
use0Box	false	Anstelle das Objekt als Parameter an \hvFloat zu übergeben, kann es vorher in der Box \hv0Box gespeichert werden und durch use0Box=true benutzt werden.
nonFloat	false	Umgebung als nicht-gleitend definieren. Objekt und Beschriftung erscheinen als normale Absätze, wobei der jeweilige Zähler (Abbildung/Tabelle) normal inkrementiert wird.
wide	false	Für die Umgebung kann die Breite \textwidth+\marginparwidth benutzt werden.
objectFrame	false	Setzt mit \frame einen Rahmen um das Objekt (ohne Abstand).
style	---	Anwenderdefinierter Stil, der vorher festgelegt sein muss.
fullpage	false	Setzt die Beschriftung unten auf die vorhergehend oder folgende Seite, während das Objekt den gesamten Textbereich beansprucht.
FULLPAGE	false	dito., nur wird die Papiergröße für das Objekt benutzt. Der Seitenstil ist dabei automatisch empty.
multiFloat	false	Mehrere Gleitobjekte zusammen als ein fullpage-Objekt mit jeweils eigener Beschriftung setzen (9.5.2 auf Seite 200).
subFloat	false	Mehrere Gleitobjekte zusammen als ein fullpage-Objekt mit einer Hauptbeschriftung und jeweiliger Teilbeschriftung (\subcaption) setzen (9.6 auf Seite 202).
separatorLine	true	Setze eine horizontale Linie (analog zu einer Funotenlinie) mit der gegebenen Dicke von 0.4pt zwischen Text und folgender Beschriftung. Nur sinnvoll bei einem fullpage-Objekt im Modus twocolumn.
capFormat	---	Alle Angaben werden an das Paket caption weitergeleitet (siehe Dokumentation zu caption).
subcapFormat	---	dito., für das Paket \subcaption.

Im einfachsten Fall wird man wie in Beispiel 09-05-2 mit \hvFloat eine normale Gleitumgebung erzeugen, die zuerst das Objekt und dann die Beschriftung darunter anordnet.

09-05-1



```
\usepackage{hvfloat}

\hvFloat{figure}%
{\includegraphics{images/crose}}%
{\texttt{\textbackslash hvFloat} Ohne
 weitere Optionen angewendet.}%
{fig:0}
```

Abbildung 1: \hvFloat Ohne weitere Optionen angewendet.

Tabellen können als normales Argument oder alternativ über eine Box eingefügt werden (Option use0Box). Das direkte Einfügen ist bei einfachen Tabellen die bessere Variante:

```
\usepackage{hvfloat}
\usepackage{booktabs}
\def\CMD#1{\texttt{\textbackslash#1}}

\hvFloat[capVPos=top, capPos=outer]{table}{%
\begin{tabular}{@{}l l@{}}\toprule
\emph{Name}      & \emph{Typ} \\ \midrule
\CMD{hvFloat}    & command \\
\CMD{hvFloatEnv} & environment \\
\CMD{figcaption} & command \\
\CMD{tabcaption} & command \\
\CMD{setDefaults}& command \\
\CMD{defhvstyle} & command \\ \bottomrule
\end{tabular}%
}{Eine einfache Tabelle, die die Beschriftung außen erhält und zusätzlich an
der Oberkante der Tabelle ausgerichtet, was der Standard sein sollte.}{tab:0}
```

<i>Name</i>	<i>Typ</i>	Tabelle 1: Eine einfache Tabelle, die die Beschriftung außen erhält und zusätzlich an der Oberkante der Tabelle ausgerichtet, was der Standard sein sollte.
<code>\hvFloat</code>	command	
<code>hvFloatEnv</code>	environment	
<code>\figcaption</code>	command	
<code>\tabcaption</code>	command	
<code>\setDefaults</code>	command	
<code>\defhvstyle</code>	command	

09-05-2

```
\usepackage{hvfloat}

\hrulefill Die Textbreite \hrulefill\par
\hvFloat[floatPos=!htb, capPos=after, objectFrame, capFormat={format=plain, font={sf, small}}]{%
{figure}{\includegraphics{images/crose}}}%
}{Caption rechts daneben mit einer \emph{natürlichen} Breite, welche durch die Breite
des Objekts, des Zwischenabstands und der Zeilenbreite selbst gegeben ist.}{fig:width1}
```

Die Textbreite

09-05-3



Abbildung 1: Caption rechts daneben mit einer *natürlichen* Breite, welche durch die Breite des Objekts, des Zwischenabstands und der Zeilenbreite selbst gegeben ist.

Anstelle der individuellen Angabe der Position (im Beispiel 09-05-3 floatPos=!htb kann auch das Paket hvfloat-fps mit der Option all=!htb geladen werden, wodurch *jeder* Gleitumgebung die Option !htb übergeben wird.

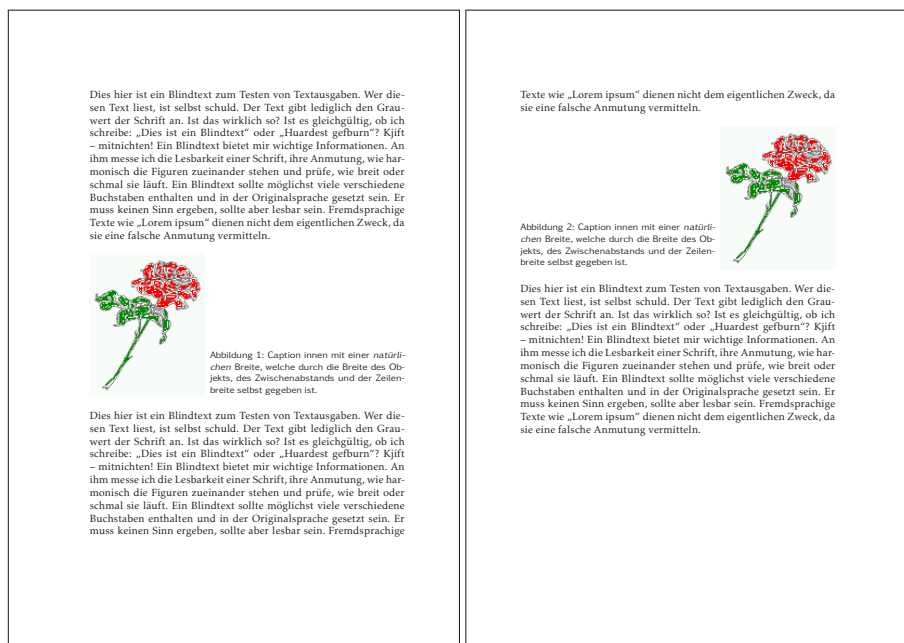
Die Beschriftung kann an allen vier Seiten des Objekts oder bei zweiseitigen Dokumenten jeweils innen oder außen angeordnet werden. Um wiederholende Angaben zu vermeiden, können die Angaben auch global oder lokal mit \hvset gesetzt werden. Das folgende Beispiel 09-05-4 zeigt die Seiten zwei und drei mit der Option capPos=inner.

09-05-4

```
\usepackage{blindtext}
\usepackage{hvfloat}
\hvFloatSet{capPos=inner,capVPos=b,floatPos=htb,capFormat={format=plain,font={sf,small}}}
\blindtext

\hvFloat{figure}{\includegraphics{images/crose}}%
{Caption innen mit einer \emph{natürlichen} Breite, welche durch die Breite
des Objekts, des Zwischenabstands und der Zeilenbreite selbst gegeben ist.}{fig:width2}
\blindtext

\hvFloat{figure}{\includegraphics{images/crose}}%
{Caption innen mit einer \emph{natürlichen} Breite, welche durch die Breite
des Objekts, des Zwischenabstands und der Zeilenbreite selbst gegeben ist.}{fig:width3}
\blindtext
```



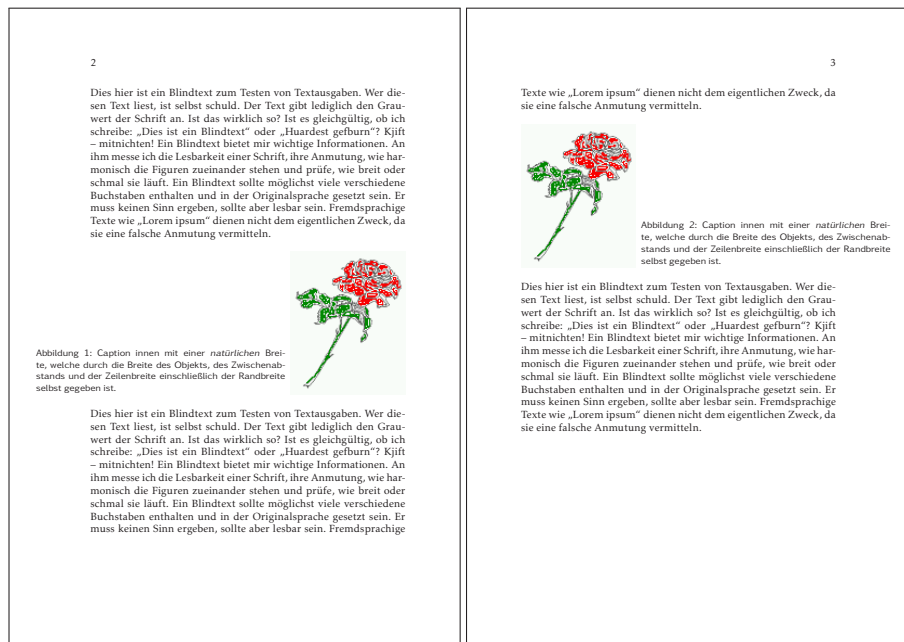
Gleiches läßt sich auch mit capPos=outer erreichen, wobei Beispiel 09-05-5 zusätzlich die Option wide nutzt, womit die gesamte Papierbreite genutzt wird.

09-05-5

```
\usepackage{blindtext}
\usepackage{hvfloat}
\hvFloatSet{capPos=outer,wide,capVPos=b,floatPos=htb,capFormat={format=plain,font={sf,small}}}
\blindtext

\hvFloat{figure}{\includegraphics{images/crose}}%
{Caption innen mit einer \emph{natürlichen} Breite, welche durch die Breite
des Objekts, des Zwischenabstands und der Zeilenbreite einschließlich der Randbreite
selbst gegeben ist.}{fig:width2}
\blindtext

\hvFloat{figure}{\includegraphics{images/crose}}%
{Caption innen mit einer \emph{natürlichen} Breite, welche durch die Breite
des Objekts, des Zwischenabstands und der Zeilenbreite einschließlich der Randbreite
selbst gegeben ist.}{fig:width3}
\blindtext
```



Bei Grafiken die wegen ihrer Größe entweder den gesamten Textbereich oder auch die gesamte Seite für sich beanspruchen, muss die Beschriftung auf die vorhergehende oder folgende Seite verschoben werden. Sinnigerweise sollte diese dann unten auf der Seite platziert werden. Für `\hvFloat` und `\includegraphics` gibt es die beiden Optionen `fullpage` und `FULLPAGE`, um größtmögliche und ganzseitige Abbildungen zu erstellen. Bei einseitigen und einspaltigen Dokumenten spielt die Anordnung in der Regel keine große Rolle. Bei zweiseitigen Dokumenten sollten dabei Objekt und Beschriftung auf gegenüberliegenden Seiten erscheinen.

```
\usepackage{blindtext,xcolor} \usepackage{hvfloat}
```

```
\Blindtext\blindtext\par
```

```
\fbox{Die Abb.-\ref{fig:fullpage0-cap} hat die Beschriftung auf Seite-\pageref{fig:fullpage0-cap} und die Abbildung aber auf Seite-\pageref{fig:fullpage0}.}
```

09-05-6

```
\hvFloat[fullpage,capPos=evenPage]{
```

```
{figure}%
```

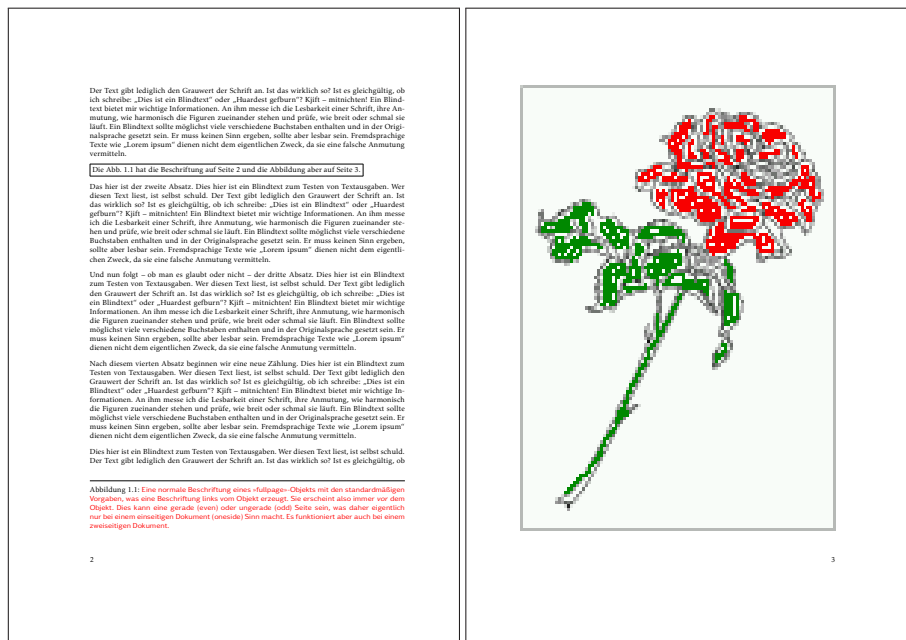
```
{\includegraphics[fullpage]{froze}}%
```

```
[Ein »fullpage« float mit normaler Caption]%
```

```
{\color{red}Eine normale Beschriftung eines »fullpage«-Objekts mit den standardmäßigen Vorgaben, was eine Beschriftung links vom Objekt erzeugt. Sie erscheint also immer \emph{vor} dem Objekt. Dies kann eine gerade (even) oder ungerade (odd) Seite sein, was daher eigentlich nur bei einem einseitigen Dokument (oneside) Sinn macht. Es funktioniert aber auch bei einem zweiseitigen Dokument.}%
```

```
{fig:fullpage0}
```

```
\Blindtext\Blindtext
```



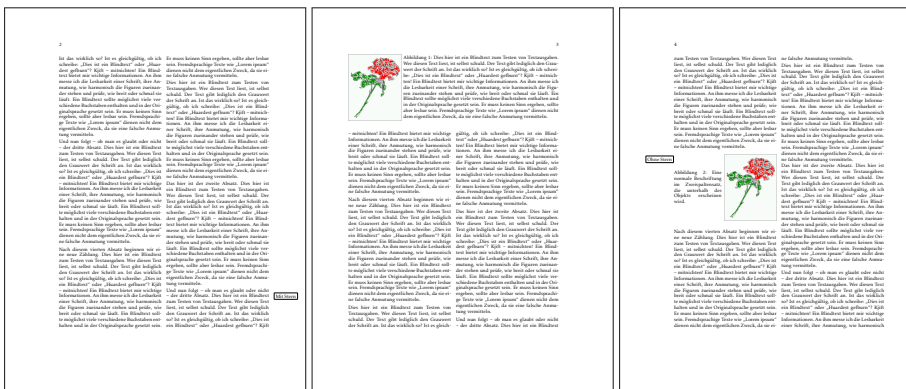
Bei zweiseitigem und zweiseitigem Satz kann mit der Sternvariante das gleiche Verhalten wie bei den Umgebungen `tabular*` und `figure*` erreicht werden; das Objekt wird zusammen mit der Beschriftung über beide Spalten gesetzt. In der Regel oben auf der der Definition folgenden Seite.

```
\usepackage{blindtext} \usepackage{hvfloat}

\Blindtext \Blindtext
\marginnote{\fbox{Mit Stern}}
\hvFloat*[capPos=right]{figure}{\includegraphics{frose}}{\blindtext}{fig:0}

\Blindtext
\marginnote{\fbox{Ohne Stern}}
\hvFloat[capPos=b]{figure}{\includegraphics{frose}}%
{Eine normale Beschriftung im Zweispaltenatz, die unterhalb des Objekts erscheinen
wird.}{figl}

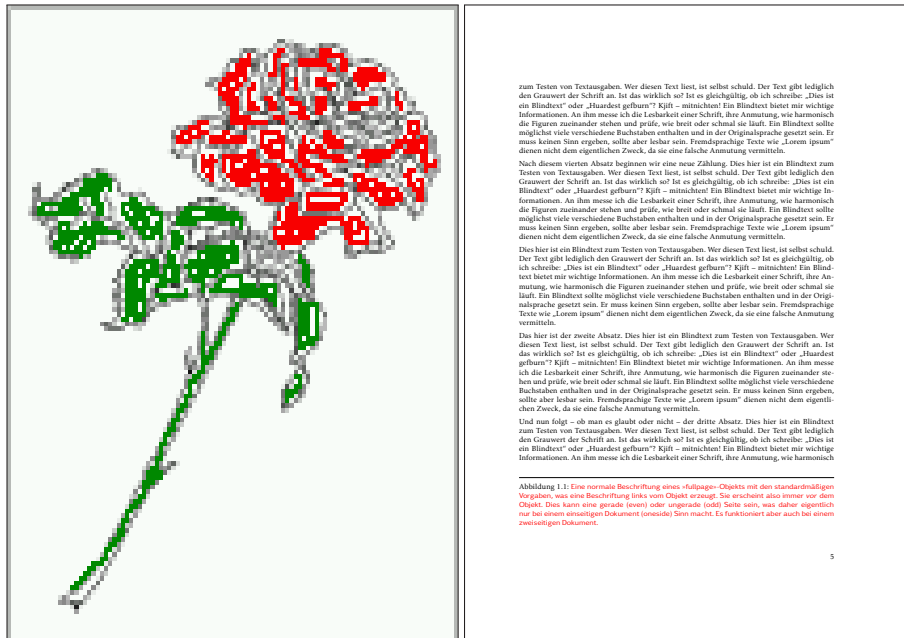
\Blindtext \Blindtext
```



Bei zweiseitigen Dokumenten wird mit den Platzierungsoptionen `capPos=even` und `capPos=odd` immer eine Anordnung auf *gegenüberliegenden* Seiten erreicht.

```
\usepackage{blindtext,xcolor} \usepackage{hvfloat}
\Blindtext\blindtext\par
\fbbox{Die Abb.-\ref{fig:fullpage1-cap} hat die Beschriftung auf Seite-\pageref{fig:fullpage1-cap}
und die Abbildung aber auf Seite-\pageref{fig:fullpage1}.}

\hvfloat[FULLPAGE, capPos=oddPage]%
{figure}%
{\includegraphics[FULLPAGE]{froese}}%
[Ein »fullpage« float mit normaler Caption]%
{\color{red}Eine normale Beschriftung eines »fullpage«-Objekts mit den standardmäßigen Vorgaben, was
eine Beschriftung links vom Objekt erzeugt. Sie erscheint also immer \emph{vor} dem Objekt. Dies kann
eine gerade (even) oder ungerade (odd) Seite sein, was daher eigentlich nur bei einem einseitigen
Dokument (oneside) Sinn macht. Es funktioniert aber auch bei einem zweiseitigen Dokument.}%
{fig:fullpage1}
\Blindtext \Blindtext \Blindtext
```



Bei ganzseitigen Objekten ergibt sich ein Problem mit dem Marke (`\label`), falls man sich mit `\pageref` auf die zugehörige Seitenzahl beziehen möchte, denn Beschriftung und Objekt sind auf unterschiedlichen Seiten. Aus diesem Grund definiert das Paket `hvfloat` eine zusätzliche Marke mit der Endung `-cap`, die sich ausschließlich auf die Beschriftung bezieht, sodass man sich, wie in den Beispielen 09-05-6 und 09-05-8 zu sehen ist, auf unterschiedliche Seiten beziehen kann. Bezüglich der Abbildungsnummer besteht bei beiden Marken kein Unterschied.

9.5.2 Multifloats

Hierunter versteht man die Ausgabe einer Gleitumgebung mit mehreren Objekten, die jeweils eine Hauptbeschriftung und mehrere Teilbeschriftungen aufweisen. Die Syntax

ist daher weitaus komplexer, wobei die Option multiFloat zwingend angegeben werden muss (siehe Beispiel 09-05-9):

```
\hvFloat [Optionen] +{float type}{floating object} [short caption] {long caption}{label}
+{float type}{floating object} [short caption] {long caption}{label}
+...
+{float type}{floating object} [short caption] {long caption}{label}
```

Das Symbol + definiert ein jeweils folgendes Teilobjekt. Hierbei muss der Anwender dafür Sorge tragen, dass die Objekte in der Summe auf eine Seite passen; ein Seitenumbruch ist nicht möglich. Das folgende Beispiel zeigt die Seiten 6, 7, 10 und 11 des einseitigen und einspaltigen Dokuments.

09-05-9

```
\usepackage{blindtext,varioref} \usepackage{hvfloat}

\title{Example for fullpage floats}\author{Herbert Voß}\maketitle \tableofcontents
\blinddocument
\section{File \texttt{\jobname}}
The fullpage multifloat-\vref{img:demo2} has a caption-\vpageref{img:demo2-cap}.

\captionsetup{singlelinecheck=false}
\hvFloat[fullpage,capPos=l,multiFloat]%
+{figure}{\includegraphics[width=\linewidth]{images/CTAN}}%% no 1
[Short caption A]%
{A Caption A of a ``fullpage'' object, which follows on the left or
right column. This can be an even or odd page. And some more text which has no
real meaning because it fills only the space for a long caption.}%
{img:demo0}%
+{table}{\begin{tabular}{lrcp{3cm}}\hline % no 2
Linksbündig & Rechtsbündig & Zentriert & Parbox\\\hline
L & R & C & P\\
left & right & center & Text with possible linebreaks\\
\multicolumn{4}{c}{Multicolumn over all columns}\\\hline
\end{tabular}}%
[Short Caption B]%
{A Caption B of a ``fullpage'' object, which follows on the left or
right column. This can be an even or odd page.}%
+{figure}{\includegraphics[width=\linewidth]{images/CTAN}}%% no 3
{A Caption C of a ``fullpage'' object, which follows on the left or
right column.}%
{img:demo1}
+{figure}{\includegraphics[width=\linewidth]{images/CTAN}}%% no 4
{A Caption D of a ``fullpage'' object, which follows on the left or
right column.}%
{img:demo2}

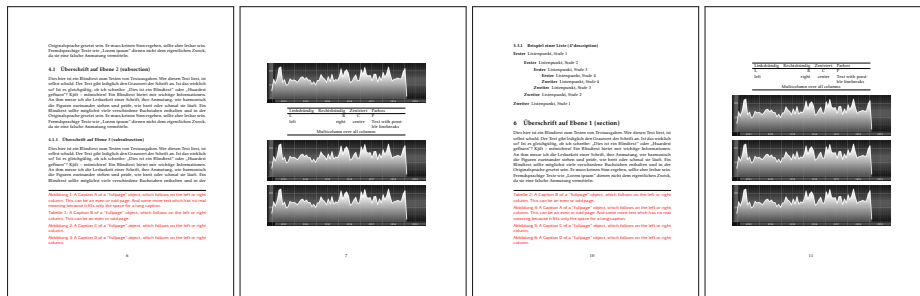
\blinddocument

\captionsetup{singlelinecheck=false}
\hvFloat[fullpage,capPos=l,multiFloat]%
+{table}{\begin{tabular}{lrcp{3cm}}\hline % no 2
Linksbündig & Rechtsbündig & Zentriert & Parbox\\\hline
L & R & C & P\\
left & right & center & Text with possible linebreaks\\
\multicolumn{4}{c}{Multicolumn over all columns}\\\hline
\end{tabular}}%
[Short Caption B]%
{A Caption B of a ``fullpage'' object, which follows on the left or
```

```

right column. This can be an even or odd page.}}%
+{figure}{\includegraphics[width=\linewidth]{images/CTAN}}}% no 1
[Short caption A]%
{A Caption A of a ``fullpage'' object, which follows on the left or
right column. This can be an even or odd page. And some more text which has no
real meaning because it fills only the space for a long caption.}%
{img:demo3}%
+{figure}{\includegraphics[width=\linewidth]{images/CTAN}}}% no 3
{A Caption C of a ``fullpage'' object, which follows on the left or
right column.}%
{img:demo4}%
+{figure}{\includegraphics[width=\linewidth]{images/CTAN}}}% no 4
{A Caption D of a ``fullpage'' object, which follows on the left or
right column.}%
{img:demo5}%
\blinddocument \Blindtext \blindtext

```



9.6 Subfloats

Im Gegensatz zu »Multifloats« können bei »Subfloats« die einzelnen Teilobjekte eine eigene Beschriftung erhalten, die jeweils direkt unter das Teilobjekt gesetzt wird. Standardmäßig kommt dabei das Paket subcaption von Axel Sommerfeld zum Einsatz. Die Syntax ist ähnlich komplex und erwartet zwingend das Schlüsselwort subFloat. Da die erste Angabe von `{float type}` für das gesamte Objekt gilt, können die folgenden Angaben entfallen; sie werden ohnehin von `\hvFloat` ignoriert. Die Teilobjekte müssen jeweils vom gleichen Typ sein.

```

\hvFloat [Optionen] +{float type}{<empty>} [short caption] {long caption}{label}
+{<empty>}{floating object} [short caption] {long caption}{label}
+...
+{<empty>}{floating object} [short caption] {long caption}{label}

```

Die Option `singlelinecheck` sollte für die Teilabbildungen standardmäßig aktiviert werden: `\captionsetup[sub]{singlelinecheck}`.

09-06-1

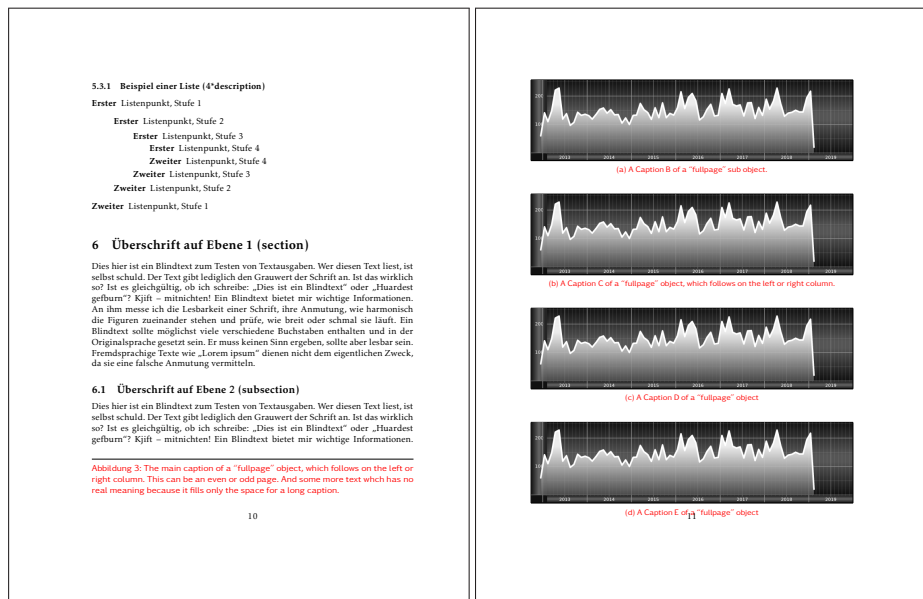
```
\usepackage{blindtext,varioref} \usepackage{hvfloat}

\title{Example for fullpage floats} \author{Herbert Voß} \maketitle \tableofcontents
\blinddocument
\section{File \texttt{\jobname}}
The multi images-\vref{sub:demo0} has a caption-\vpageref{sub:demo0-cap} and the
subimage--\vref{sub:demo1} has a caption-\vpageref{sub:demo1}.

\captionsetup[sub]{singlelinecheck}
\hvFloat[fullpage,capPos=l,objectFrame,subFloat]%
+{}{}[Short main caption of the objects]% main short lsi entry
{The main caption of a ``fullpage'' object, which follows on the left or
right column. This can be an even or odd page. And some more text which has no
real meaning because it fills only the space for a long caption.}% main caption
{sub:demo0}%
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
[Short caption B]%
{A Caption B of a ``fullpage'' sub object.}% subcaption
{}%
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
{A Caption C of a ``fullpage'' object, which follows on the left or right column.}%
{sub:demo1}
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
{A Caption D of a ``fullpage'' object}%
{sub:demo2}
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
{A Caption E of a ``fullpage'' object}%
{sub:demo3}

\blinddocument
\captionsetup[sub]{singlelinecheck}
\hvFloat[fullpage,capPos=l,objectFrame,subFloat]%
+{}{}[Short main caption of the objects]% main short lsi entry
{The main caption of a ``fullpage'' object, which follows on the left or
right column. This can be an even or odd page. And some more text which has no
real meaning because it fills only the space for a long caption.}% main caption
{sub:demo4}%
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
[Short caption B]%
{A Caption B of a ``fullpage'' sub object.}% subcaption
{}%
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
{A Caption C of a ``fullpage'' object, which follows on the left or right column.}%
{sub:demo5}
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
{A Caption D of a ``fullpage'' object}%
{sub:demo7}
+{}{}{\includegraphics[width=\linewidth]{CTAN}}%
{A Caption E of a ``fullpage'' object}%
{sub:demo6}

\blinddocument \Blindtext \blindtext
```



9.7 hyperref

Für Hyperlinks ist das mächtige Paket `hyperref` von Sebastian Rahtz und Heiko Oberdiek zuständig. Es versucht, sich in alle möglichen Referenzen, Links, Fußnoten und andere Verweise einzuklinken um sämtliche Zeiger, die auf irgend etwas zeigen, unter seine Kontrolle zu bringen. Es kann dann Hyperlinks von beispielsweise einer Fußnotenziffer zu dem korrespondierenden Fußnotentext erstellen. Es gilt die allgemeine Regel, dass das Paket `hyperref` als letztes aller Pakete zu laden ist. Einige Ausnahmen bestätigen diese Regel und sollten unbedingt beachtet werden, um eine einwandfreie Funktion zu erreichen. Die Tabelle 9.4 zeigt für die besonderen Fälle, was in diesen zu beachten ist. In allen anderen sollte es keine Probleme geben, wenn `hyperref` als letztes Paket geladen wird. Bei den Dokumentenklassen ist nur für `IEEEtran` darauf zu achten, dass wegen `\@makecaption` eine Version $\geq 1.6b$ zu wählen ist.

Tabelle 9.4: Hinweise zur Reihenfolge von Paketen im Zusammenhang mit `hyperref`.

Paket(e)	Anmerkungen
<code>algorithm</code>	<code>\usepackage{float}</code> <code>\usepackage{hyperref}</code> <code>\usepackage[chapter]{algorithm}</code>
<code>amsmath</code>	<code>\usepackage{amsmath}</code> <code>\let\equation\gather</code> <code>\let\endequation\endgather</code>
<code>amsrefs</code>	<code>\usepackage{hyperref}</code> <code>\usepackage{amsrefs}</code>
<code>arydshln</code>	<code>\usepackage{xltabular}</code>

Fortsetzung ...

<i>Paket(e)</i>	<i>Anmerkungen</i>
xltabular	\usepackage{hyperref} \usepackage{arydshln}
bibentry	\makeatletter\let\saved@bibitem\bibitem\makeatother \usepackage{bibentry} \usepackage{hyperref} \begin{document} \begin{group} \makeatletter\let\bibitem\saved@bibitem \nobibliography{database} \endgroup
chappg	\usepackage{hyperref} \usepackage{chappg}
cite	\usepackage{cite} \makeatletter \def\NAT@parse{\typeout{This is a fake Natbib command to fool Hyperref.}} \makeatother \usepackage[hypertex]{hyperref} \cite arbeitet dann wie gewohnt, aber es werden keine Hyperlinks eingerichtet.
dblacnt	% pdlenc.def oder puenc.def müssen vorher geladen werden \usepackage{hyperref} \usepackage{dblacnt}
easyeqn	Bislang keine Anwendung mit hyperref möglich!
ellipsis	Das Paket redefiniert \textellipses, weswegen es <i>nach</i> hyperref geladen werden muss. \usepackage{hyperref} \usepackage{ellipsis}
float	Mehrere Makros \caption innerhalb einer Gleitumgebung werden nicht unterstützt. Wenn der Stil vom Paket float kontrolliert wird, dann zeigt der Anker von hyperref auf die obere Kante des Objekts. \usepackage{float} \usepackage{hyperref}
endnotes	Wird nicht von hyperref unterstützt.
footnote	Paket wird nicht unterstützt und man muss zusätzlich hyperref mit der Option hyperfootnotes=false laden.
geometry	Probleme gibt es nur bei dem Treiber dvipdfm und dem gleichnamigen Programm zum Konvertieren der DVI-Datei. Es ist grundsätzlich das Programm dvipdfmx zu benutzen.
index	Wegen der Redefinition von \addcontentsline ist eine Version >= 1995/09/28 v4.1 zu wählen.
linguex	\usepackage{hyperref} \usepackage{linguex}
ltabptch	\usepackage{xltabular} \usepackage{ltabptch} \usepackage{hyperref}
nomncl	Beispiele für Links der Seitenzahl: \renewcommand*{\pagedeclaration}[1]{\unskip, \hyperpage{#1}} Für Gleichungsnummern: \renewcommand*{\eqdeclaration}[1]{\hyperlink{equation.#1}{(Equation-#1)}}

Das Verhalten von hyperref wird fast ausnahmslos über Optionen gesteuert, die mit zum einen als Paketoptionen, zum anderen aber auch mit \hypersetup gesetzt werden können. Hierbei können nicht alle Optionen nach dem Laden des Paketes mit \hypersetup geändert werden. hyperref gibt eine Warnung aus, wenn eine Option zu spät geändert

wird und daher nichts mehr bewirken kann.

```
\usepackage[Optionen]{hyperref}
\hypersetup{Key/Value-Liste}
```

Eine alphabetische Zusammenstellung aller derzeit bekannten Optionen zeigt die Tabelle 9.5. Aus reinen Platzgründen können hier davon nicht alle behandelt werden.

Tabelle 9.5: Zusammenstellung aller möglichen Optionen für das Paket hyperref. Eine fehlende Angabe in der Spalte *Vorgabe* entspricht einem leeren Argument.

<i>Name</i>	<i>Vorgabe</i>	<i>Anmerkungen</i>
anchorcolor	black	Ankerfarbe
backref	false	Bibliographische Rückwärtsreferenzen
baseurl		Basis-URL für das Dokument
bookmarks	true	Erstelle Bookmarks
bookmarksnumbered	false	Bookmarks mit Abschnittsnummer
bookmarksopen	false	Zeige Bookmarks im PDF-Viewer aufgeklappt (kontrolliert durch <code>bookmarksopenlevel</code>).
bookmarksopenlevel	\maxdimen	Tiefe, bis zu der die Bookmarks im Viewer geöffnet werden sollen.
bookmarkstype	toc	Festlegung des Typs für die Bookmarks.
breaklinks	true	Interne Angabe, ob der verwendete Treiber Links umbrechen kann.
CJKbookmarks	false	CJK Bookmarks
citebordercolor	0 1 0	Rahmenfarbe für \cite-Befehle.
citecolor	green	Linkfarbe für \cite-Befehle.
colorlinks	false	Farbige Links, statt eines Rahmens. Für tex4ht- und dviwindo-Treiber ist die Vorgabe true.
debug	false	Mehr Informationen in die Logdatei schreiben (analog zur Option verbose).
destlabel	false	Zielmarken werden nach Ankersetzung nach dem ersten \label benannt.
draft	false	Keine Hyperlinks erstellen.
dvipdfm		dvipdfm-Treiber
dvipdfmx		dvipdfmx-Treiber
dvips		dvips-Treiber
dvipsone		dvipsone-Treiber
dviwindo		dviwindo-Treiber
encap		»encap«-Zeichen für den Hyperindex
extension	dvi	Suffix für verlinkte Dateien, kann dvi, pdf, htm oder html sein.
filebordercolor	0 .5 .5	Rahmenfarbe für Dateilinks
filecolor	cyan	Linkfarbe für Dateilinks
final	true	Gegenteil der Option draft
frenchlinks	false	\textsc statt einer Linkfarbe benutzen
hyperfigures	false	Abbildungen verlinken

Fortsetzung ...

<i>Name</i>	<i>Vorgabe</i>	<i>Anmerkungen</i>
hyperfootnotes	true	Fußnoten verlinken
hyperindex	true	Index verlinken
hypertex		HyperTeX-Treiber
hypertextnames	true	Benutze Namen für Links
implicit	true	Redefiniere L ^A T _E X-Intern
latex2html		L ^A T _E X2HTML-Treiber
linkbordercolor	1 0 0	Rahmenfarbe von Links
linkcolor	red	Linkfarbe
linktocpage	false	Verlinke die Seitenzahl anstatt des Textes für alle Verzeichnisse.
menubordercolor	1 0 0	Rahmenfarbe für Menülinks
menucolor	red	Linkfarbe für Menülinks
nativepdf	false	Alias für dvips
naturalnames	false	benutze L ^A T _E X-erstellte Namen für Links
nesting	false	Erlaube verschachtelte Links
pageanchor	true	Setze für jede Seite einen Anker
pagebackref	false	Rückwärtsreferenz mit Seitenzahl.
pdfauthor		Text für PDF-Autor-Feld
pdfborder	0 0 1	Breite eines Linkrahmens, für die Option colorlinks ist die Vorgabe gleich 0 0 0.
pdfcenterwindow	false	Zentrierte Anordnung des Dokumentenfensters auf dem Bildschirm.
pdfcreator		LaTeX with hyperref package
pdfdirection		Text für das PDF-Ersteller-Feld
pdfdisplaydoctitle	false	Richtungsfestlegung
pdfduplex		Dokumententitel statt Dateinamen in der Menüleiste anzeigen
pdfffitwindow	false	Für Duplexdruck
pdfhighlight	/I	Fenstergröße auf Dokumentgröße einstellen.
pdfinfo		»highlighting« von PDF-Links
pdfkeywords		Alternative zum Setzen der Dokumentinformationen.
pdflang		Text für das PDF-Schlüsselwörter-Feld
pdfmark	false	PDF language identifier (RFC 3066)
pdfmenubar	true	Alias für dvips
pdfnewwindow		Menüleiste im PDF-Viewer aktivieren
pdfnonfullscreenpagemode		Links auf andere PDF-Dateien im neuen Fenster öffnen, mögliche Werte sind false und true, sowie ein leeres Argument {}, welches als (absent) interpretiert wird.
pdfnumcopies		Seitenanzeigemodus beim Verlassen vom »full-screen« Modus
		Anzahl Kopien für den Drucker

Fortsetzung ...

<i>Name</i>	<i>Vorgabe</i>	<i>Anmerkungen</i>
pdfpagelayout		Wert /PageLayout festlegen, möglich sind SinglePage, OneColumn, TwoColumnLeft und TwoColumnRight,
pdfpagemode		Den Standardmodus für die PDF-Anzeige setzen
pdfpagelabels	true	Erstelle PDF-Seitenlabels
pdfpagescrop		Setze »crop«-Seitengröße des PDF-Dokument
pdfpagetransition		Seitenübergangsstil setzen
pdfpicktrackbypdfsize		Optionen für den Druckdialog setzen
pdfprintarea		Setze /PrintArea in den Viewer-Präferenzen
pdfprintclip		Setze /PrintClip in den Viewer-Präferenzen
pdfprintpagerange		Setze /PrintPageRange in den Viewer-Präferenzen
pdfprintscaling		Setze /PrintScaling in den Viewer-Präferenzen
pdfproducer		Text für das »PDF Producer« Feld
pdfremotestartview	Fit	Startansicht entfernter PDF-Dokumente
pdfstartpage	1	Startseite mit der das PDF-Dokument geöffnet wird
pdfstartview	Fit	Startmodus des PDF-Dokuments, mögliche Werte sind Fit, FitB, FitH top, FitBH top, FitR x1 y1 x2 y2, FitV left, FitBV left und XYZ links rechts zoom.
pdfsubject		Text für das PDF-Betreff-Feld
pdftex		pdfTeX-Treiber
pdftitle		Text für das PDF-Titel-Feld
pdftoolbar	true	PDF-Toolbar ist sichtbar
pdftrapped		Trapped Dokumentinformation (mögliche Werte sind True, False und Unknown. Ein leerer Eintrag verweist auf die Nichtsetzung.
pdfview	XYZ	PDF-»view«, wenn einem Link gefolgt wird
pdfviewarea		Setze /ViewArea in den Viewer-Präferenzen
pdfviewclip		Setze /ViewClip in den Viewer-Präferenzen
pdfwindowui	true	Elemente eines benutzerdefinierten Interfaces sichtbar machen
plainpages	false	Seitenanker als arabische Zahlen setzen
ps2pdf		Benutze ps2pdf-Treiber
raiselinks	false	Links »erhöhen« (für HyperTeX-Treiber)
runbordercolor	0 .7 .7	Rahmenfarbe der Programmlinks
runcolor	filecolor	Farbe der Programmlinks
setpagesize	true	Setze Seitengröße mit den passenden Treiber-Befehlen
tex4ht		TeX4ht-Treiber
textures		Textures-Treiber
unicode	false	PDF-Strings in Unicode, siehe auch pdfencoding. Sollte zwingend für Xe _L TeX/Lua _L TeX angegeben werden.

Fortsetzung ...

Name	Vorgabe	Anmerkungen
<code>urlbordercolor</code>	<code>0 1 1</code>	Rahmenfarbe der URL-Links
<code>urlcolor</code>	<code>magenta</code>	Farbe der URL-Links
<code>verbose</code>	<code>false</code>	Mehr Ausgaben im Logfile
<code>vtex</code>		VT _E X-Treiber
<code>xetex</code>		X _E T _E X-Treiber

Einige wichtige von `hyperref` unterstützte Befehle zeigt die folgende kurze Zusammenstellung.

Name	Beschreibung
<code>\href{Link}{Text}</code>	<i>Text</i> verweist intern auf <i>Link</i> .
<code>\hypersetup{Argument}</code>	<i>Argument</i> enthält Wertzuweisungen für die <code>hyperref</code> -Optionen.
<code>\nolinkurl{URL}</code>	<i>URL</i> wird wie ein Argument von <code>\url</code> behandelt, jedoch nicht verlinkt.
<code>\pdfbookmark[Ebene]{BookmarkText}{Ankername}</code>	<i>BookmarkText</i> wird mit <i>Ankername</i> verlinkt. Die <i>Ebene</i> 0 entspricht dabei chapter für die book-Klasse.
<code>\currentpdfbookmark{Text}{Name}</code>	Bookmark für die aktuelle Ebene.
<code>\subpdfbookmark{Text}{Name}</code>	Bookmark eine Ebene tiefer als die aktuelle.
<code>\belowpdfbookmark{Text}{Name}</code>	dito., ohne Änderung des Ebene.
<code>\texorpdfstring{T_EX}{PDF}</code>	Bookmarks können keine Sonderzeichen darstellen, sodass <i>T_EX</i> für die normale Darstellung und <i>PDF</i> für die Bookmarks verwendet wird, beispielsweise in Überschriften.

9.8 array

Mit dem `array`-Paket stehen vielfältige Möglichkeiten für eine bessere Einteilung einer Tabelle zur Verfügung. Zu empfehlen ist in jedem Fall noch die Anwendung von `booktabs`. Ausgehend von der folgenden Tabelle wird zum Vergleich jeweils eine der neuen Möglichkeiten gezeigt, wobei der Vollständigkeit halber auch »normale« Spaltendefinitionen zur Anwendung kommen. Es sei darauf hingewiesen, dass die Vielzahl der horizontalen und vertikalen Linien nicht der »Lesbarkeit« einer Tabelle förderlich ist; es dient hier nur zur besseren Erklärung der einzelnen Optionen. Im allgemeinen sollte man vertikale Linien gar nicht und horizontale nur sparsam einsetzen.

09-08-1

Links	Rechts	Zentriert	Box
l	r	c	p{2cm}

```
\begin{tabular}{|l|r|c|p{2cm}|}\hline
Links & Rechts & Zentriert & Box\\\hline
l & r & c & p{2cm}\\\hline
\end{tabular}
```

`\extrarowheight` vergrößert die **Höhe** der Zeile ohne Einfluss auf die Tiefe zu nehmen, sodass dieser Abstand nicht zu groß gewählt werden sollte. Er dient im Wesentlichen nur zur Vermeidung des Anstoßens von Großbuchstaben an eine obere Linie.

Links	Rechts	Zentriert	Box
l	r	c	p{1.75cm}

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

\setlength\extrarowheight{8pt}
\begin{tabular}{|l|r|c|p{1.75cm}|}\hline
Links & Rechts & Zentriert & Box\\\hline
\rowcolor{cyan!40}
l & r & c & p{1.75cm}\\\hline
\end{tabular}
```

09-08-2

`\arraybackslash` redefiniert den standardmäßigen Befehl für einen Tabellenzeilenumbruch, den Doppelbackslash `\`, der bei gleichzeitiger Anwendung eines der Formatierungsmakros `\raggedright`, `\raggedleft` oder `\centering` ohne `\arraybackslash` in sogenannten p-Spalten nicht mehr seine ursprüngliche Bedeutung hätte. Für eine neue Zeile in einer Spalte benutzt man dann `\newline` und für eine neue Tabellenzeile dann `\` oder alternativ das Makro `\tabularnewline`.

L	R	Z	Box
l	r	c	p{2.5cm}
l	r	c	neue Zeile mit <code>\newline</code> neue Tabellenzeile mit <code>\</code>
wie jetzt			

```
\usepackage{array,ragged2e}
\usepackage[table]{xcolor}

\begin{tabular}{|>{\columncolor{yellow}}l|r|c|}
>{\RaggedRight\arraybackslash}p{2.5cm}|}\hline
L & R & Z & Box\\\hline
l & r & c & p{2.5cm}\newline
neue Zeile mit \verb+\newline+\\
l & r & c & neue Tabellenzeile mit \verb+\\+\\
wie jetzt &&& \\\hline
\end{tabular}
```

09-08-3

L	R	Z	Box
l	r	c	p{2cm}
l	r	c	neue Zeile mit <code>\newline</code> neue Tabellenzeile mit <code>\tabu-</code> <code>larnewline</code>
wie jetzt			

```
\usepackage{array,ragged2e}
\usepackage[table]{xcolor}

\begin{tabular}
{||r|c|>{\RaggedRight}p{2.5cm}|}\hline
L & R & Z & Box\tabularnewline\hline
l & r & c & \cellcolor{magenta!40}p{2cm}\newline
neue Zeile mit \verb+\newline+\\
l & r & c & neue Tabellen\zeile
mit \Lcs\tabularnewline\\
wie jetzt&&&\tabularnewline\hline
\end{tabular}
```

09-08-4

`>{...}`, wobei hier `»...«` für eine beliebige Code-Sequenz steht, die vor dem Beginn der jeweiligen Spalte in jeder Zeile ausgeführt wird, in diesem Beispiel wird die Fettschrift gewählt.

Links	Rechts	Zentriert	Box
l	r	c	p{1.5cm}

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

\begin{tabular}{|>{\bfseries}l|r|c|p{1.5cm}|}
\rowcolor{red!30}Links&Rechts&Zentriert&Box\\
\rowcolor{red!60}l & r & c & p{1.5cm}\}
\end{tabular}
```

09-08-5

`<{...}`, wobei hier »...« wieder für eine beliebige Code-Sequenz steht, die diesmal *vor dem Ende* der jeweiligen Spalte in jeder Zeile ausgeführt wird, in diesem Beispiel wird der mathematische Modus gewählt, indem `>{}` den Inlinemodus ein- und `<{}` ausschaltet.

09-08-6	Links	Rechts	Zentriert	Box	<code>\usepackage{array}</code> <code>\usepackage[table,dvipsnames]{xcolor}</code> <code>\begin{tabular}{l>{\columncolor{Tan}\$}r<{\$}</code> <code>c p{1.5cm} }</code> <code>Links & Rechts & Zentriert & Box\\hline</code> <code>l & r & c & p\{1.5cm\}</code> <code>\end{tabular}</code>
	l	r	c	p{1.5cm}	

`!{...}`, wobei hier »...« wieder für eine beliebige Code-Sequenz, die diesmal *anstelle* einer Trennlinie in jeder Zeile eingefügt wird, wobei links und rechts der gleiche Abstand zu den beiden Spalten bestehen bleibt. In diesem Beispiel wird anstelle der Trennlinie ein Doppelpunkt bzw. ein Pfeil eingefügt.

09-08-7	Links	Rechts	: Zentriert	→ Box	<code>\usepackage[table]{xcolor}</code> <code>\begin{tabular}{l!{\color{red}}\vline}r</code> <code>!{:}c!{\color{blue}\$\rightarrow\$p{1.5cm}}</code> <code>Links & Rechts & Zentriert & Box\\hline</code> <code>\rowcolor[cmk]{0,.5,.7,0}l&r&c&p\{1.5cm\}</code> <code>\end{tabular}</code>
	l	r	:	c → p{1.5cm}	

`@{...}`, wobei hier »...« wieder für eine beliebige Code-Sequenz steht, die diesmal *anstelle* einer Trennlinie in jeder Zeile eingefügt wird, wobei links und rechts *kein* Abstand mehr zwischen den beiden Spalten besteht. In diesem Beispiel wird wieder anstelle der Trennlinie ein Doppelpunkt bzw. ein Pfeil eingefügt. Der Unterschied zu obigem Beispiel ist offensichtlich.

09-08-8	Links	Rechts:Zentriert	→ Box	<code>\usepackage{xcolor}</code> <code>\begin{tabular}{@{}l r@{:}c@{\protect%</code> <code>\colorbox{cyan}{\$\rightarrow\$p{1.5cm}}@{}}</code> <code>Links & Rechts & Zentriert & Box\\cline{2-4}</code> <code>l & r & c & p\{1.5cm\}\\cline{1-1}</code> <code>\end{tabular}</code>
	l	r: c	→ p{1.5cm}	

`*{Anzahl}{Spaltentyp}`, wobei hier *Anzahl* für den Wiederholungsfaktor und *Spaltentyp* für eine beliebige Code-Sequenz steht.

09-08-9

l	l	l	r	r	r	r
L	L	L	R	R	R	R

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

\begin{tabular}
{|*{3}>{\columncolor{blue!40}}l|*{4}{r|}}\cline{1-3}
l&l&l&r&r&r&r\\ L&L&L&R&R&R&R\\ \cline{1-3}
\end{tabular}
```

Das Makro `\newcolumnntype` erlaubt beliebige neue Spaltentypen, die dann in abgekürzter Form als Spaltendefinition im Tabellenkopf benutzt werden können.

```
\newcolumnntype{Zeichen}[n]{Spaltendefinition}
```

Zeichen kann nur ein Buchstabe sein, der sinnvollerweise von den bereits reservierten abweichen sollte. *n* bezeichnet die Anzahl der verwendeten Parameter und bleibt unberücksichtigt, wenn die Definition ohne Parameter erfolgt. Die *Spaltendefinition* muss auf einen bereits existierenden Spaltentyp Bezug nehmen. Die folgenden Beispiele zeigen jeweils exemplarisch die Anwendung des Makros.

```
\usepackage{array}
\usepackage[table,svgnames]{xcolor}
\newcolumnntype{L}{>{\$}l<{\$}} \newcolumnntype{T}{>{\ttfamily\small}l}

\rowcolors{1}{Pink}{0rchid!50}% aus xcolor
\begin{tabular}{\LLT@{\quad}\LLT}
Sinus & \sin(\alpha) & <Wert> sin & Cosinus & \cos(\alpha) & <Wert> cos \\
Tangens & \tan(\alpha) & <Wert> tan & Arcus Tangens & \arctan(x) & <Wert1> <Wert2> atan
\end{tabular}
```

Sinus	$\sin(\alpha)$	<Wert> sin	Cosinus	$\cos(\alpha)$	<Wert> cos
Tangens	$\tan(\alpha)$	<Wert> tan	Arcus Tangens	$\arctan(x)$	<Wert1> <Wert2> atan

09-08-10

Im folgenden Beispiel wird eine Kopfzeile gedreht, um auf diese Weise die schmalen Spalten nicht unnötig zu verbreitern, was zu Lasten der Lesbarkeit der Tabelle ginge. Wenn bei einer Spalte mehrere Zeilen im Kopf erscheinen, dann sollte das Paket `varwidth` von Donald Arseneau eingesetzt werden, welches gegenüber der `minipage`-Umgebung den Vorteil hat, dass man die benötigte Breite nicht explizit angeben muss.

```
\usepackage{array,rotating}
\usepackage[table,svgnames]{xcolor} \rowcolors{2}{Peru}{Peru!50}
\newcolumnntype{B}{>{\bfseries}c} \newcolumnntype{T}{>{\ttfamily}l}
\newcolumnntype{R}[1]{>{\color{Crimson}\begin{turn}{#1}l<\end{turn}}}

\begin{tabular}[b]{BTL}
\multicolumn{1}{R{45}}{Abkürzung} & \multicolumn{1}{R{45}}{Bezugsspalte} & 
\multicolumn{1}{R{45}}{Parameter} \\ \hline
B & c & - \\
T & l & - \\
R & l & 1
\end{tabular} \quad \quad \quad \rowcolors{2}{Thistle}{Wheat}
\begin{tabular}[b]{BTL}
\multicolumn{1}{R{90}}{Abkürzung} & \multicolumn{1}{R{90}}{Bezugsspalte} & 
\multicolumn{1}{R{90}}{Parameter} \\ \hline
B & c & - \\
T & l & - \\
R & l & 1
\end{tabular}
```

Abkürzung	Bezugsspalte	Parameter
B	c	-
T	l	-
R	l	1

Abkürzung	Bezugsspalte	Parameter
B	c	-
T	l	-
R	l	1

09-08-11

Nach Definition trennt $\text{\TeX}$ nicht das erste Wort eines Absatzes. Dies ist in schmalen Spalten aber manchmal wünschenswert. Man definiert einen Spaltentyp, der als erstes

ein `\hspace{0pt}` in die Spalte schreibt, was $\TeX$ als ein Wort der Breite Null erkennt und daraus folgend dann das nächste reguläre Wort trennt. Im folgenden Beispiel erkennt man dies deutlich; die erste Tabelle ist fehlerhaft in der letzten Spalte, die zweite korrekt.

09-08-12	Links	Rechts	Zentriert	Box	<code>\usepackage{array}\usepackage[table]{xcolor}</code>
	l	r	c	Definitionsgemäß	<code>\newcolumntype{P}[1]{% >{\columncolor{green!40}% \hspace{0pt}}p{#1}}</code>
	Links	Rechts	Zentriert	Box	<code>\begin{tabular}{ l r c> {\columncolor{blue!40}}p{2cm}} Links & Rechts & Zentriert & Box\\\hline l & r & c & Definitionsgemäß \end{tabular}\\\{8mm} \begin{tabular}{ l r cP{2cm}} Links & Rechts & Zentriert & Box\\\hline l & r & c & Definitionsgemäß \end{tabular}</code>
	l	r	c	Definitionsgemäß	

Horizontale Ausrichtungen erreicht man mit den bereits mehrfach erwähnten drei Makros `\raggedleft`, `\centering` und `\raggedright`, beziehungsweise den modifizierten Makros `\RaggedLeft`, `\Centering` und `\RaggedRight` des Paketes `ragged2e` von Martin Schröder, die eine Worttrennung erlauben.

```
\usepackage{array,ragged2e} \usepackage[table]{xcolor}
\newcolumntype{L}[1]{>{\color{red}\hspace{0pt}\RaggedRight\arraybackslash}p{#1}}
\newcolumntype{C}[1]{>{\color{green}\hspace{0pt}\Centering\arraybackslash}p{#1}}
\newcolumntype{R}[1]{>{\color{blue}\hspace{0pt}\RaggedLeft\arraybackslash}p{#1}}

\begin{tabular}{@{}L{3cm}C{3cm}R{3cm}@{}}\hline
Eine linksbündige Spalte mit rechtsseitigem Flattersatz
& Eine zentrierte Spalte mit linkem und rechtem Flattersatz
& Eine rechtsbündige Spalte mit linksseitigem Flattersatz\\
L & C & R\\\hline
\end{tabular}
```

09-08-13	Eine linksbündige Spalte mit rechtsseitigem Flattersatz	Eine zentrierte Spalte mit linkem und rechtem Flattersatz	Eine rechtsbündige Spalte mit linksseitigem Flattersatz
	L	C	R

9.9 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 `dvipsnames` Option, 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 »named«-Option, die

dvipsnames erwartet, wenn die entsprechenden vordefinierten Farben (siehe Tabelle 9.10 auf Seite 221) benutzt werden sollen.

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

Tabelle 9.6: Zusammenstellung der wichtigsten Paketoptionen von xcolor.

Option	Bedeutung
natural	(Standard) Benutze alle Farben innerhalb ihres Modells, mit Ausnahme von RGB (konvertiert to rgb), HSB (konvertiert to hsb), und Gray (konvertiert to gray).
rgb	Konvertiert alle Farben in das rgb-Modell.
cmy	Konvertiert alle Farben in das cmy-Modell.
cmyk	Konvertiert alle Farben in das cmyk-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 to gray-Modell).
dvipsnames	Lädt die vordefinierten DVIPS-Farben.
svgnames	Lädt die vordefinierten SVG-Farben
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 colortbl Paket, um farbige Tabellenzeilen zu ermöglichen.
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.

9.9.1 Definition von Farben über numerische Werte

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

Nun ist der Text grün
und nun ein wenig ma-
genta 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.

09-09-1

Name	Grundfarben	Parameterbereich	Standard
rgb	red, green, blue	[0, 1]	
cmv	cyan, magenta, yellow	[0, 1]	
cmv	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, ..., FFFFFFF}	
HSB	Hue, Saturation, Brightness	{0, 1, ..., M}	M = 240
Gray	Gray	{0, 1, ..., N}	N = 15
wave	lambda (nm -- nanometer)	[363, 814]	

Tabelle 9.7: Unterstützte Farbmodelle (L, M, N sind natürliche Zahlen).

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      \providcolor  \colorlet
\definecolorset   \providcolorset
\definecolorseries
```

Weiterhin definiert xcolor die in PS als *current color* bezeichnete 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.

09-09-2



```
\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.

```
\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}
```

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 Neudefinition, womit ein Überschreiben verhindert werden kann.

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

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

09-09-3

Die Option `table` ist nur bei farbigen Tabellen notwendig, wie beispielsweise in Tabelle 9.6 auf Seite 214 und 9.7 auf der vorherigen Seite.

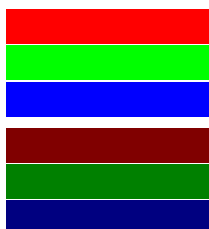
`xcolor` unterstützt die Definition von Farbserien mit drei Makros, wobei zwei nichts weiter als eine erweiterte Version von `\definecolor` sind.

```
\definecolorset{rgb}{}{}{red,1,0,0;green,0,1,0;blue,0,0,1}
\providecolorset{rgb}{}{}{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 sowohl durch xcolor als auch PSTricks bereits definiert.

09-09-4



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

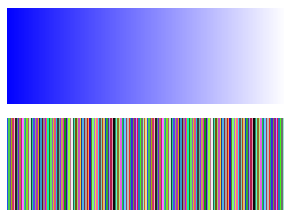
{\col{red}}\{\col{green}}\{\col{blue}}\{\col{redH}}\{\col{greenH}}\{\col{blueH}}
```

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]`.

09-09-5



```
\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}}\{5pt}
\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. (Kern 2016) 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}}\
\makebox[30mm][l]{magenta (M):}%
\multido{\nColr=0+1}{10}{\colorbox{M!![\nColr]}{0.\nColr}}\
\makebox[30mm][l]{yellow (Y):}%
\multido{\nColr=0+1}{10}{\colorbox{Y!![\nColr]}{0.\nColr}}\
\makebox[30mm][l]{black (K):}%
\multido{\nColr=0+1}{10}{\colorbox{K!![\nColr]}{0.\nColr}}
```

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

09-09-6

```
\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}}\
```

Hue (H):	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Saturation (S):	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Brightness (B):	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9

09-09-7

9.9.2 Farbspezifikation

- »specification by name«, wobei die aktuelle Farbe den Punkt (.) als »Namen« hat.
- »specification by expression« mit der Syntax:
PräfixName! Wert1! Name1! ...! WertN! NameN Suffix

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`.

Wert1!Name1 Die neue Farbe ergibt sich aus *Wert1%* der Farbe *temp* und $(100 - \text{Wert1})\%$ von der Farbe *Name1* 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 9.8 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 9.8: 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!40	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

09-09-8



```
\usepackage{pstricks}

\psframebox[linewidth={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:

core model, *suml:expr1,fac1;expr2,fac2;...;exprN,facN*

Jede Farbe in dem Ausdruck enthält einen ganzzahligen Faktor, der die Gewichtung der Farbe darstellt.

09-09-9

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	cmk:red,2;white,2
rgb:red,5;green,2;yellow,10	cmk:red,5;green,2;yellow,10
rgb,11:red,5;green,2;cyan,1	cmk,11:red,5;cyan,2;cyan,1

9.9.3 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 9.10, 9.9 und 9.11.

Tabelle 9.9: Liste der verfügbaren Farbnamen für das svgnames-Modell

 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

Fortsetzung ...

... Fortsetzung des svgnames-Modells

 SkyBlue	 SlateBlue	 SlateGray	 SlateGrey
 Snow	 SpringGreen	 SteelBlue	 Tan
 Teal	 Thistle	 Tomato	 Turquoise
 Violet	 Wheat	 White	 WhiteSmoke
 Yellow	 YellowGreen		

Tabelle 9.10: 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
 PineGreen	 Plum	 ProcessBlue	 Purple	 RawSienna
 Red	 RedOrange	 RedViolet	 Rhodamine	 RoyalBlue
 RoyalPurple	 RubineRed	 Salmon	 SeaGreen	 Sepia
 SkyBlue	 SpringGreen	 Tan	 TealBlue	 Thistle
 Turquoise	 Violet	 VioletRed	 White	 WildStrawberry
 Yellow	 YellowGreen	 YellowOrange		

Tabelle 9.11: Liste der verfügbaren Farbnamen für das x11names-Modell

 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

Fortsetzung ...

... Fortsetzung x11names-Modell

 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
 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

Fortsetzung ...

... Fortsetzung x11names-Modell

 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			

9.9.4 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{cmyk}{0,0.6,0.8,0}[Orange]}
```

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

09-09-10

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

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

09-09-11

Schwarz auf 40% Rot Schwarz – Text, cyan – Hintergrund, Rot – Rahmen
 Heller Hintergrund Weiß – Text, cyan – Hintergrund, Rot – Rahmen

Weiteren Beispiele zeigen 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}
```

Spaß mit xcolor

Spaß mit xcolor

Spaß mit xcolor

Spaß mit xcolor

09-09-12

9.9.6 Farbwerte bestimmen

Mithilfe des `xcolor`-Pakets lassen sich auf einfache Art und Weise Farbwerte oder auch Farbserien bestimmen (Abschnitt 9.9.1). 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 `\convertcolourspec` 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 :\convertcolourspec{HTML}{FF2006}{rgb}\RGBFarbe
\textcolor[rgb]{\RGBFarbe}{\RGBFarbe}\\
cmyk:\convertcolourspec{HTML}{FF2006}{cmyk}\CMYKFarbe
\textcolor[cmyk]{\CMYKFarbe}{\CMYKFarbe}\\
hsb :\convertcolourspec{HTML}{FF2006}{hsb}\HSBFarbe
\textcolor[hsb]{\HSBFarbe}{\HSBFarbe}
```

09-09-13