

Цвет в L^AT_EX 2_ε*

Описаны команды, введённые в пакетах `color` из коллекции `graphics` и `colortbl`. Оба пакета написал David Carlisle. Пакет `color` позволяет нам выбирать по своему усмотрению цвет текста и фона как отдельного блока на странице, так и всей страницы печатного документа. Пакет `colortbl` позволяет раскрашивать таблицы.

Содержание

1	Выбор драйвера	1
2	Выбор цвета	2
2.1	Цветовые модели	2
2.1.1	Модель <code>named</code>	2
2.1.1.1	Цвета, именованные в драйвере <code>dvips</code> .	2
2.1.2	Модель <code>rgb</code>	3
2.1.3	Модель <code>gray</code>	4
2.1.4	Модель <code>cmuk</code>	4
2.2	Определение нового имени цвета	4
3	Цветной текст	4
4	Цветные боксы	5
5	Цветной фон страницы	5
6	Цветные таблицы	5
	Алфавитный указатель	8

1 Выбор драйвера

При загрузки пакетов надо в необязательном аргументе команды `\usepackage` указать драйвер или, проще говоря, программу, которая «понимает» команды переключения цвета. Драйвером по умолчанию задан `dvips`¹. Программа `dvips`, которую написал Tomas Rokicki, переводит `dvi`-файл, полученный после обработки L^AT_EX'ом входного файла, в файл, в котором документ описан на языке PostScript. `ps`-файл можно с помощью программы `GhostScript`² просмотреть на экране монитора и распечатать на обычном принтере. Кроме того, `GhostScript` умеет конвертировать `ps`-файл в файл формата PDF. Если же вы создаёте документ в формате PDF непосредственно с помощью `pdfLATEX`'а, написанного Hàn Thê Thành, то вам следует при загрузки пакетов указывать в виде опции драйвер `pdftex`.

*© 2001 Владимир Сюткин. Замечания приветствуются: syutkin@ns.kinetics.nsc.ru

¹Драйвер по умолчанию задаётся в аргументе команды `\ExecuteOptions` в файле настройки `color.cfg`.

²`GhostScript` является интерпретатором языка PostScript и используется в среде Windows под оболочкой `GSview`.

Остальные драйверы, которые в настоящее время декларированы для пакетов из коллекции `graphics`: `xdvi`, `dvipdf`, `dvipdfm`, `dvipsone`, `dviwindo`, `emtex`, `dviwin`, `pctexps`, `pctexwin`, `pctexhp`, `pctex32`, `truettex`, `tcidvi`, `vtex`, `oztex`, `textures`.

На заметку Команды включения цвета игнорируются, если пакет загружен с опцией `monochrome`. Используется, когда выходное устройство не поддерживает цвета.








2 Выбор цвета

Все команды переключения цвета имеют опцию, в которой указывается цветовая модель, и обязательный аргумент(ы), в котором задаётся спецификация цвета. Синтаксис спецификации зависит от выбранной модели. В этом разделе описаны все цветовые модели, которые поддерживает пакет `color`. В примерах используются команды, описанные в разделе 3.

2.1 Цветовые модели

2.1.1 Модель `named`

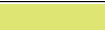

























`named` является цветовой моделью по умолчанию. Спецификация цвета в этой модели задаётся по имени цвета. В пакете `color` определены имена восьми цветов, приведённые в следующей таблице вместе с самими цветами³:

Имя	Цвет	Имя	Цвет
<code>black</code> (чёрный)		<code>white</code> (белый)	
<code>red</code> (красный)		<code>green</code> (зелёный)	
<code>blue</code> (синий)		<code>cyan</code> (голубой)	
<code>magenta</code> (пурпурный)		<code>yellow</code> (жёлтый)	

Пример с командой `\textcolor`⁴:

`\textcolor{blue}{Синий \fbox{x^2}.}` Синий x^2 .











































2.1.1.1 Цвета, именованные в драйвере `dvips`. В файле `dvipsnam.def` содержится определение имён 68 цветов. Они перечислены в следующей таблице.

Имя	Цвет	Имя	Цвет
<code>GreenYellow</code>		<code>Yellow</code>	
<code>Goldenrod</code>		<code>Dandelion</code>	
<code>Apricot</code>		<code>Peach</code>	
<code>Melon</code>		<code>YellowOrange</code>	
<code>Orange</code>		<code>BurntOrange</code>	
<code>Bittersweet</code>		<code>RedOrange</code>	
<code>Mahogany</code>		<code>Maroon</code>	
<code>BrickRed</code>		<code>Red</code>	
<code>OrangeRed</code>		<code>RubineRed</code>	
<code>WildStrawberry</code>		<code>Salmon</code>	
<code>CarnationPink</code>		<code>Magenta</code>	
<code>VioletRed</code>		<code>Rhodamine</code>	
<code>Mulberry</code>		<code>RedViolet</code>	

продолжение на следующей странице

³Восприятие цвета зависит от выходного устройства: даже на экран монитора GhostScript и Acrobat Reader выводят разные цвета с одной и той же спецификацией.

⁴Здесь и далее в левой части примера показан исходный текст, а в правой — результат его обработки L^AT_EX'ом.

(продолжение)			
Имя	Цвет	Имя	Цвет
Fuchsia		Lavender	
Thistle		Orchid	
DarkOrchid		Purple	
Plum		Violet	
RoyalPurple		BlueViolet	
Periwinkle		CadetBlue	
CornflowerBlue		MidnightBlue	
NavyBlue		RoyalBlue	
Blue		Cerulean	
Cyan		ProcessBlue	
SkyBlue		Turquoise	
TealBlue		Aquamarine	
BlueGreen		Emerald	
JungleGreen		SeaGreen	
Green		ForestGreen	
PineGreen		LimeGreen	
YellowGreen		SpringGreen	
OliveGreen		RawSienna	
Sepia		Brown	
Tan		Gray	
Black		White	

Если пакет `color` загружен с опцией `usenames`:

```
\usepackage[usenames]{color}
```

то цвета драйвера `dvips` можно использовать точно так же, как и цвета пакета `color`, указывая их имена в качестве спецификации цвета. Пример:

```
\textcolor{Orange}{Оранжевый  $\bigcup$ .} \quad \text{Оранжевый } \bigcup.
```

Если опция пакета `usenames` опущена, то при использовании цветов драйвера `dvips` надо указывать опцию команд переключения цвета `named` в качестве цветовой модели:

```
\textcolor[named]{Orange}{Оранжевый  $\oint$ .} \quad \text{Оранжевый } \oint.
```

На заметку Имена цветов драйвера `dvips` распознаются драйвером `pdftex`, если пакет `color` загружен с опцией `dvipsnames`.

2.1.2 Модель `rgb`

В модели `rgb` любой цвет получается в результате смешения лучей света трёх базовых цветов: красного (`red`), зелёного (`green`) и синего (`blue`). Поэтому спецификация цвета в этой модели задаётся тремя перечисленными через запятую числами от 0 до 1, которые соответствуют абсолютной интенсивности базовых составляющих света. Например, жёлтый цвет задаётся как `1, 1, 0`, а белый — как `1, 1, 1`. `0, 0, 0` даёт чёрный цвет.

Задавая цвет, надо помнить, что его восприятие зависит не только от длины волны света, но и от насыщенности и яркости излучения. Например, свет, содержащий лучи только красного цвета, от тусклого источника воспринимается не как бледно красный, а как чёрный. Бледно красный цвет получается при большой интенсивности света, в котором красная составляющая несколько превышает остальные.

2.1.3 Модель `gray`

Серый (`gray`) цвет получается при смешении в равных пропорциях базовых цветов из модели `rgb` с интенсивностью, меньшей единицы. Спецификация цвета в модели `gray` задаётся одним числом от 0 до 1, например, 0.5 вместо 0.5, 0.5, 0.5 в модели `rgb`. Чёрному цвету соответствует 0, а белому — 1. Пример со спецификацией `gray` можно найти в разделе 2.2.

2.1.4 Модель `смук`

Если на лист бумаги нанести красную краску и затем осветить его белым светом, то только красная составляющая света отразится от бумаги. Нанесём теперь на бумагу красную краску густо посаженными точками, а затем заполним все промежутки между красными точками зелёной краской. Теперь от листа бумаги отразится как красный, так и зелёный свет. Издали различить отдельные точки нельзя, поэтому глаз получит смесь красного и зелёного света и бумага будет выглядеть жёлтой⁵. На этом принципе основана цветовая модель `смук`. В этой модели спецификация цвета задаётся четырьмя перечисленными через запятую числами от 0 до 1, которые соответствуют «количеству» голубой (`cyan`), пурпурной (`magenta`), жёлтой (`yellow`) и чёрной (`black`) краски на белом листе бумаги. Теоретически при смешении первых трёх цветов в равной пропорции должен получиться чёрный цвет. В действительности краски поглощают свет не полностью и поэтому смесь трёх основных цветов выглядит тёмно-коричневой. По этой причине в модель введена ещё и чёрная краска.

Из описанного выше ясно, что в модели `смук` белому цвету соответствует спецификация 0, 0, 0, 0 (на белом листе бумаги нет никакой краски), а чёрному — 0, 0, 0, 1.

2.2 Определение нового имени цвета

Команда

```
\definecolor{name}{model}{spec}
```

позволяет определить имя *name* для любого цвета. Здесь *model* — цветовая модель, *spec* — спецификация цвета. Используя имя *name* в качестве спецификации цвета, цветовую модель можно не указывать, поскольку такой способ переключения цвета относится к модели по умолчанию `named`.

Пример:

```
\definecolor{faded}{gray}{0.7}
\textcolor{faded}{Блѐклый\dots}
```

Блѐклый...

3 Цветной текст

Изменить цвет текста *text* в документе можно либо командой

```
\textcolor[model]{spec}{text}
```

либо декларацией

```
{\color[model]{spec}text}
```

Здесь *model* — цветовая модель, *spec* — спецификация цвета. Пример с разными цветовыми моделями:

```
\textcolor{red}{Красный,}
\textcolor[смук]{0,1,1,0}{опять красный,}
\textcolor[rgb]{1,0,0}{и ещё раз красный!}
```

Красный,
опять красный,
и ещё раз красный!

⁵Если сначала смешать красную и зелёную краски, а затем нанести смесь на бумагу, то получится тёмный цвет с красноватым оттенком.

4 Цветные боксы

Команда

```
\colorbox[model]{spec}{lr-text}
```

помещает, подобно команде `\mbox`, текст *lr-text* в бокс, у которого цвет фона задан аргументами *model* и *spec*. Здесь *model* — цветовая модель, *spec* — спецификация цвета. Команда

```
\fcolorbox[model]{fr-spec}{spec}{lr-text}
```

дополнительно обводит этот бокс рамкой цвета *fr-spec*. Пример:

```
\fcolorbox{red}{yellow}{Текст\dotsc}
```



В этом примере опция, задающая цветовую модель опущена, поскольку мы используем цвета из пакета `color`. Список таких цветов приведён в разделе 2.

На заметку В качестве толщины линий рамки и ширины промежутка между рамкой и текстом в боксе команда `\fcolorbox` использует значения параметров `\fboxrule` и `\fboxsep` стандартного L^AT_EX'a.

5 Цветной фон страницы

Изменить цвет страницы можно декларацией

```
\pagecolor[model]{spec}
```

Здесь *model* — цветовая модель, *spec* — спецификация цвета. Область действия декларации не ограничивается никакими скобками. Чтобы вернуть белый цвет страниц, надо вызвать команду `\pagecolor{white}`.

6 Цветные таблицы

Пакет `colortbl`, используя пакеты `color` и `array`, позволяет раскрашивать таблицы. Цвета задаются точно так же, как в пакете `color` через цветовую модель *model* и спецификацию цвета *spec* (см. раздел 2).

Команду

```
\columncolor[model]{spec}[left-overhang][right-overhang]
```

можно использовать в спецификации `>{...}` (её вводит пакет `array`) в преамбуле окружений `tabular` и `array`. Она окрашивает весь столбец таблицы цветом, который задаётся первыми двумя аргументами *model* и *spec*. Опции *left-overhang* и *right-overhang* задают расстояния, соответственно, слева и справа между краями окрашенной области и текстом. Если указан только один аргумент, то он задаёт оба расстояния. По умолчанию значения аргументов задаются командными длинами `\tabcolsep` для окружения `tabular` и `\arraycolsep` для окружения `array`. Следующий пример показывает, как оставить в средней колонке неокрашенными полосы шириной `.4\tabcolsep`:

```
\begin{tabular}{%  
|>\color{white}\columncolor{black}l|  
>\columncolor{yellow}[.6\tabcolsep]c|  
>\columncolor[gray]{.8}r|}  
один & два & три \\ четыре & пять & шесть  
\end{tabular}
```

один	два	три
четыре	пять	шесть

Команда

```
\rowcolor[model]{spec}[left-overhang][right-overhang]
```

используется для окрашивания целой строки таблицы. Её место — в самом начале строки:

```
\begin{tabular}{|l|c|}
\rowcolor[gray]{.9} один & два \\
\rowcolor[gray]{.6} три & четыре \\
\end{tabular}
```

один	два
три	четыре

Приведём теперь пример таблицы со слитыми ячейками в строке. В этом случае цвет слитых ячеек нужно задавать в спецификации команды `\multicolumn`. Я сделал это, введя новый тип колонки:

```
\newcolumntype{H}{>{\columncolor{magenta}}c}
\begin{tabular}{%
|>{\columncolor{yellow}}l|
>{\color{white}\columncolor{black}}l|}
\multicolumn{2}{|H|}{один} \\
два & три \\
четыре & пять \\
\end{tabular}
```

один	
два	три
четыре	пять

На заметку Пакет `colortbl` совместим с пакетами `longtable` и `dcolumn` из коллекции `tools`.

Цветные таблицы лучше смотрятся, когда строки в них отделены друг от другой белым промежутком. Для вставки такого промежутка можно определить новую команду. Назовём её `\tabrowsep`. Использую более компактный синтаксис plain TeX'a, определим команду `\tabrowsep` в виде:

```
\def\tabrowsep{\noalign{\vskip 2pt}}
```

Здесь используется команда `\noalign`, которая вставляет бокс (в нашем случае высотой 2 pt) в стопку боксов (в нашем случае это будут строки таблицы). Приведём пример с командой `\tabrowsep`:

```
\newcolumntype{H}{%
>{\columncolor[gray]{.9}}p{1.7cm}}
\begin{tabular}[t]{*2H}
\rowcolor[gray]{.6}один & два\\
три & четыре \\
пять & шесть \\
\end{tabular}
```

один	два
три	четыре
пять	шесть

Глобальная декларация

`\arrayrulecolor[model]{spec}`

задаёт цвет горизонтальных и вертикальных линий, разделяющих ячейки в таблицах. Её можно вводить не только перед таблицей, но и в спецификации `>{...}` в преамбуле таблицы или в начале какой-нибудь строки таблицы. Область действия декларации `\arrayrulecolor` начинается в точке её размещения во входном файле. Так, если ввести декларацию сразу после преамбулы, то вертикальные разделительные линии, заданные в преамбуле, сохранят свой цвет.

Ещё одна глобальная декларация

`\doublerulesepcolor[model]{spec}`

задаёт окрашивание промежутка между двойными вертикальными (`| |`) и двойными горизонтальными (`\hline\hline`) разделительными линиями в таблицах. Ширину линий и промежутка между линиями можно выбрать по своему усмотрению. Пример с обеими декларациями:

```
\setlength\arrayrulewidth{2pt}\arrayrulecolor{blue}
\setlength\doublerulesep{2pt}\doublerulesepcolor{yellow}
\begin{tabular}{||l||c||}
\hline\hline
один & два \\
три & четыре \\
\hline\hline
\end{tabular}
```

один	два
три	четыре

Команда `\hhline` из пакета `hhline`, рисующая горизонтальные линии в таблицах, позволяет модифицировать таблицу из предыдущего примера, например, так:

```
\begin{tabular}{|l|c|}
\hhline{|t:=:t:=:t|}
один & два \\ три & четыре \\
\hhline{|b:=:b:=:b|}
\end{tabular}
```

один	два
три	четыре

Декларации `\arrayrulecolor` и `\doublerulesepcolor` можно использовать в `>{...}` перед `-` и `=` в аргументе команды `\hhline`. Пример:

```
\newcommand\rgbline[1]{\hhline{>{\arrayrulecolor{red}}|#1:=%
>{\arrayrulecolor{green}}=>{\arrayrulecolor{blue}}=#1|}}
```

```
\arrayrulecolor{red}
\begin{tabular}{|*3c|}\rgbline{t}
red & green & \multicolumn{1}{c|}{blue}\\
\rgbline{b}\end{tabular}
```

red	green	blue
-----	-------	------

На заметку Из-за проблем с окраской линий, которые чертит команда `\cline`, рекомендуется вместо неё использовать команду `\hhline` с линейкой `-`.

Алфавитный указатель

a
`\arrayrulecolor` 6

c
`\color` 4
`\colorbox` 5
`\columncolor` 5

d
`\definecolor` 4
`\doublerulesepcolor` 6

p
`\pagecolor` 5

r
`\rowcolor` 5

t
`\textcolor` 4

o
Опции
`dvipsnames` 3
`monochrome` 2
`usenames` 3

ц
Цветовая модель
`смук` 4
`gray` 4
`named` 2
`rgb` 3