

Цвета в Delphi - тип TColor



- [Цвета в Delphi - тип TColor](#)
- [Цвета по названию](#)
 - [Стандартные](#)
 - [Расширенные](#)
 - [Системные](#)
 - [Название как строка](#)
- [Два представления: #RrGgBb и #BbGgRr](#)

Последнее изменение: 03.10.2016

Начнём с того, что цвета в компьютере представимы в различных цветовых моделях, таких как [RGB](#), [CMYK](#), [LAB](#), [HSV](#), [HSL](#) и другие. Здесь я буду рассматривать только RGB. Сейчас расскажу почему.

В Delphi имеется встроенный тип данных для хранения цветов: Graphics.TColor (он же Windows.COLORREF, DWORD и Longint). Это целочисленный тип с диапазоном $-\$7FFFFFFF-1..\$7FFFFFFF$ (т.е. -2147483648..2147483647). Достаточно много, учитывая что компьютер поддерживает около 17 миллионов цветов. Почему именно столько? RGB подразумевает представления цвета в виде трёх компонент: Красной (Red), Зелёной (Green) и Синей (Blue). Каждая из них задаётся числом от 0 до 255. Несложно посчитать, что цветов будет 256^3 , т.е. 16^777^216 . Для того, чтобы "собрать" TColor из трёх компонент есть стандартная функция Windows.RGB:

```
function RGB(r, g, b: Byte): COLORREF;  
begin  
  Result := (r or (g shl 8) or (b shl 16));  
end;
```

Если не очень понятно, то можно переписать так:

```
function RGB(r, g, b: Byte): COLORREF;  
begin  
  Result := r + g * 256 + b * 256 * 256;  
end;
```

Заметили как просто? Однако, будут небольшие проблемы. Но позже.

Естественно, есть и обратные функции, извлекающие нужные компоненты цвета - Windows.GetRValue, Windows.GetGValue и Windows.GetBValue:

```
function GetRValue(rgb: DWORD): Byte;  
begin  
  Result := Byte(rgb);  
end;  
  
function GetGValue(rgb: DWORD): Byte;  
begin  
  Result := Byte(rgb shr 8);  
end;  
  
function GetBValue(rgb: DWORD): Byte;  
begin  
  Result := Byte(rgb shr 16);  
end;
```

Их тоже можно написать проще, используя целочисленное деление (div) и остаток от деления (mod), предлагаю подумать как.

Цвета по названию

Наиболее используемые цвета объявлены в Delphi (кстати, уже в модуле "Graphics") именными константами - названиями цветов с префиксом "cl". Их не так много, как предусматривает [HTML](#), но всё же что-то. Их можно разделить на три группы.

Стандартные

16 цветов, знакомых каждому. О колонке #RRGGBB в таблице расскажу чуть позже.

cbStandardColors

###	Название	Цвет	Значение	R, G, B	#RrGgBb
	clBlack	Чёрный	0	0, 0, 0	#000000
	clMaroon	Тёмно-красный	128	128, 0, 0	#800000
	clGreen	Зелёный	32`768	0, 128, 0	#008000
	clOlive	Оливковый	32`896	128, 128, 0	#808000
	clNavy	Тёмно-синий	8`388`608	0, 0, 128	#000080
	clPurple	Пурпурный	8`388`736	128, 0, 128	#800080
	clTeal	Стальной	8`421`376	0, 128, 128	#008080
	clGray	Серый	8`421`504	128, 128, 128	#808080
	clSilver	Серебряный	12`632`256	192, 192, 192	#C0C0C0
	clRed	Красный	255	255, 0, 0	#FF0000
	clLime	Ярко-зелёный	65`280	0, 255, 0	#00FF00
	clYellow	Жёлтый	65`535	255, 255, 0	#FFFF00
	clBlue	Синий	16`711`680	0, 0, 255	#0000FF
	clFuchsia	Фиолетовый	16`711`935	255, 0, 255	#FF00FF
	clAqua	Бирюзовый	16`776`960	0, 255, 255	#00FFFF
	clWhite	Белый	16`777`215	255, 255, 255	#FFFFFF

Расширенные

Зачем-то разработчики выделили в отдельную категорию ещё 4 более редких цвета.

cbExtendedColors

###	Название	Цвет	Значение	R, G, B	#RrGgBb
	clMoneyGreen	Money Green	12`639`424	192, 220, 192	#C0DCC0
	clSkyBlue	Sky Blue	15`780`518	166, 202, 240	#A6CAF0
	clCream	Cream	15`793`151	255, 251, 240	#FFFBF0
	clMedGray	Medium Gray	10`789`024	160, 160, 164	#A0A0A4

Системные

Они имеют названия, но само значение цвета зависит от текущей версии ОС и выбранной темы. Как это реализовано? Расскажу после таблицы, в которой представлены системные цвета для стандартных тем Windows XP и Windows 7.

cbSystemColors

Название	Значение	Win Xp			Win 7		
		###	R, G, B	#RrGgBb	###	R, G, B	#RrGgBb
clActiveBorder	-16`777`206		212, 208, 200	#D4D0C8		180, 180, 180	#B4B4B4
clActiveCaption	-16`777`214		0, 84, 227	#0054E3		153, 180, 209	#99B4D1
clAppWorkSpace	-16`777`204		128, 128, 128	#808080		171, 171, 171	#ABABAB
clBackground	-16`777`215		0, 78, 152	#004E98		0, 0, 0	#000000
clBtnFace	-16`777`201		236, 233, 216	#ECE9D8		240, 240, 240	#F0F0F0
clBtnHighlight	-16`777`196		255, 255, 255	#####		255, 255, 255	#####
clBtnShadow	-16`777`200		172, 168, 153	#ACA899		160, 160, 160	#A0A0A0
clBtnText	-16`777`198		0, 0, 0	#000000		0, 0, 0	#000000
clCaptionText	-16`777`207		255, 255, 255	#####		0, 0, 0	#000000
clGradientActiveCaption	-16`777`189		61, 149, 255	#3D95FF		185, 209, 234	#B9D1EA
clGradientInactiveCaption	-16`777`188		157, 185, 235	#9DB9EB		215, 228, 242	#D7E4F2
clGrayText	-16`777`199		172, 168, 153	#ACA899		109, 109, 109	#6D6D6D
clHighlight	-16`777`203		49, 106, 197	#316AC5		51, 153, 255	#3399FF
clHighlightText	-16`777`202		255, 255, 255	#####		255, 255, 255	#####
clHotLight	-16`777`190		0, 0, 128	#000080		0, 102, 204	#0066CC
clInactiveBorder	-16`777`205		212, 208, 200	#D4D0C8		244, 247, 252	#F4F7FC
clInactiveCaption	-16`777`213		122, 150, 223	#7A96DF		191, 205, 219	#BFCDD8
clInactiveCaptionText	-16`777`197		216, 228, 248	#D8E4F8		67, 78, 84	#434E54
clInfoBk	-16`777`192		255, 255, 225	#####E1		255, 255, 225	#####E1
clInfoText	-16`777`193		0, 0, 0	#000000		0, 0, 0	#000000
clMenu	-16`777`212		255, 255, 255	#####		240, 240, 240	#F0F0F0
clMenuBar	-16`777`186		236, 233, 216	#ECE9D8		240, 240, 240	#F0F0F0
clMenuHighlight	-16`777`187		49, 106, 197	#316AC5		51, 153, 255	#3399FF
clMenuText	-16`777`209		0, 0, 0	#000000		0, 0, 0	#000000
clScrollBar	-16`777`216		212, 208, 200	#D4D0C8		200, 200, 200	#C8C8C8
cl3DDkShadow	-16`777`195		113, 111, 100	#716F64		105, 105, 105	#696969
cl3DLight	-16`777`194		241, 239, 226	#F1EFE2		227, 227, 227	#E3E3E3
clWindow	-16`777`211		255, 255, 255	#####		255, 255, 255	#####
clWindowFrame	-16`777`210		0, 0, 0	#000000		100, 100, 100	#646464
clWindowText	-16`777`208		0, 0, 0	#000000		0, 0, 0	#000000

Ага, заметили, что цвета отрицательные? Их невозможно получить из трёх компонент функцией RGB(...), невозможно извлечь их компоненты, передав такой цвет в GetRValue(...) и ему подобные. Это и позволяет дать возможность системе выбирать значения этих цветов исходя из настроек. Чтобы преобразовать их в обычные, положительные, цвета имеется специальная функция Graphics.ColorToRGB:

```
function ColorToRGB(Color: TColor): Longint;
begin
  if Color < 0 then
    Result := GetSysColor(Color and $000000FF)
  else
    Result := Color;
end;
```

Теперь с ним можно работать, как и прежде.

Название как строка

Кстати, Delphi имеет представление о том, как названы его цветовые константы, и позволяет преобразовать строку с названием цвета в сам цвет, используя функцию Graphics.StringToColor:

```
function ToString(const S: string): TColor;
begin
  if not IdentToColor(S, Longint(Result)) then
    Result := TColor(StrToInt(S));
end;
```

И наоборот, по цвету попробовать узнать его название, используя Graphics.ColorToString:

```
function ColorToString(Color: TColor): string;
begin
  if not ColorToIdent(Color, Result) then
    FmtStr(Result, '%s%.8x', [HexDisplayPrefix, Color]);
end;
```

Особо любознательные могли задаться вопросом: А что будет, если цвет не имеет названия? Ответ: функция вернёт преобразованное в строку число (не забываем, TColor - всего лишь целочисленный тип).

Два представления: #RrGgBb и #BbGgRr

Часто цвета записывают короче, используя [Шестнадцатеричную систему счисления](#). Это позволяет уложиться в 7 символов вида #RRGGBB:

- как отличительный знак такой записи;

RR - две шестнадцатеричных цифры, отвечающих за красный;

GG - за зелёный;

BB - и за синий.

Такая запись достаточно распространена, и используется, например, в [HTML](#).

Запомнили их порядок? А теперь я скажу, что бывает очень похожая запись, вида #BBGRRR, где красный и синий поменялись местами. Именно такой порядок цветов получается при использовании ColorToString из предыдущего абзаца. Плюс и форматирование там немного другое. В общем, я не советую без надобности пользоваться встроенной функцией, лучше написать свою:

```
function ColorToHex(color: TColor): String;
begin
  Result := Format('#%.2x%.2x%.2x', [byte(color), byte(color shr 8), byte(color shr 16)]);
end;
```

Она уже будет приводить любой (положительный!) цвет к виду #RRGGBB, даже если он имеет название.

[Смотреть последнюю версию статьи онлайн](#)

 [Материал этой страницы доступен по лицензии CC BY-SA 4.0](#)

© [diversenok 2015-2016](#)