

qocolours.sty Package

Peter Brookes Chambers

August 17, 2021

Contents

| | | |
|----------|-------------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Useage | 2 |
| 2.1 | Additional Commands | 2 |

1 Introduction

This package defines a set of colour schemes which resemble many of those provided by the Microsoft Office suite of software, in addition to 386 other schemes. Each scheme defines 10 base colours, as well as 5 variations of each. These are then used by some of other packages within the QoL^AT_EX family of packages. The colours defined by each scheme are as follows, with a short description of their general usage in the QoL^AT_EX family of packages.

| | |
|--------------------------|--|
| ForegroundColour | The default text colour for all non-paragraph text (i.e., in diagrams, box titles and similar). |
| BackgroundColour | The default background or page colour for all non-paragraph text (i.e., in diagrams, box titles and similar). |
| Accent1 | The primary accent colour |
| Accent2 | The secondary accent colour. Also used for hyperlinks in the <code>---</code> package by default. |
| Accent3 | Further accent colours. |
| Accent4 | |
| Accent5 | |
| Accent6 | |
| Hyperlink | An alternative colour for external links. This is not always provided uniquely for all schemes; for schemes without a unique Hyperlink colour, a blue colour is used as default. |
| FollowedHyperlink | An alternative colour for external links. In MS Office, this would be used for links which have already been clicked. This is not always provided uniquely for all schemes; for schemes without a unique Hyperlink colour, a dark blue colour is used as default. |

Each colour then has 5 variants, indicated by a “_1” to “_5” suffix. These colour variants are derived from the core colour by changing only the lightness of the colour in the HSL colour model; the hue and saturation are unchanged. Note that the HSL model is different to the HSB or HSV colour models. For these variations, “20% darker” means that the lightness is brought 20% closer to 0, and “20% lighter” means that the lightness is 20% closer to 1. Thus, unless a core colour has a lightness of 0 or 1, the variants will never have a lightness of 0 or 1 (except after rounding). The variants are defined in one of three ways depending on the lightness of the core colour.

- For most core colours, there are two darker variants (25% and 50% darker), and three lighter variants (40%, 60%, and 80% lighter).
- For core colours with lightness above $l = 0.8$, 5 darker variants are defined at 10%, 25%, 50%, 75%, and 90% darker.
- For core colours with lightness below $l = 0.2$, 5 lighter variants are defined at 10%, 25%, 50%, 75%, and 90% lighter.

This approximates the behaviour of the Microsoft Office colour themes as closely as possible.

Each scheme has two variants; “light” and “dark”, which for now simply swap the **ForegroundColour** and **BackgroundColour** and change the order of the core colour variants. In the “light” variant of each scheme, these variants are all arranged from darkest to lightest, such that **Accent1_1** will be the darkest variant, and **Accent1_5** the lightest. In the “dark” variants of each scheme, this is reversed; this should mean that relative contrast is conserved as closely as possible when switching between light and dark variants.

2 Usage

The package can be loaded with `\usepackage{qocolours}`. By default, this will load the `light` variant of the `twilight` scheme. To load a different scheme, one can use the keyword `scheme`. To minimise overhead when loading the package, four optional arguments are provided; `microsoft`, `iterm`, `slidehelper`, and `all`. To have access to any of the 79 Microsoft Office-like themes, the flag `microsoft` must be included. For access to the 237 themes from <https://iterm2colorchemes.com/>, use the flag `iterm`. For the 150 themes from slidehelper.com, use the flag `slidehelper`. The flag `all` provides access to all themes. For the light and dark variants, two flags are provided; `light` and `dark`. So, to load the scheme which closely resembles the MS Office theme *aspect* in its dark variant, one can use `\usepackage[scheme = aspect, microsoft, dark]{qocolours}` or `\usepackage[scheme = aspect, all, dark]{qocolours}`. The flags `light` and `dark` can also take boolean values, so one can also use `light = false` or similar.

In addition, the scheme can be changed part way through a document using the command `\UseColourScheme`, which takes one required argument; the name of the new scheme. For example, the scheme can be changed to resemble the MS Office theme *Black Tie* using `\UseColourScheme{black_tie}`. Unfortunately, the scheme cannot be changed from the light to the dark variant or back except in the call to `\usepackage`.

2.1 Additional Commands

There are a few commands defined for use within the package. Although these are not intended to be used except by the package itself, they are documented here anyway. There is no guarantee that these commands will exist or act the same in future versions.

| | |
|-----------------------------------|--|
| <code>\getRed c</code> | Expands to the two characters representing the red value of the RGB hex string <code>c</code> . Strictly, <code>c</code> is not a single argument but 6 arguments, so care should be taken with order of expansions here. |
| <code>\getGreen c</code> | Similar to <code>\getRed c</code> except that the green value is found. |
| <code>\getBlue c</code> | Similar to <code>\getRed c</code> except that the blue value is found. |
| <code>\lighten{c}{p}{name}</code> | This command takes three arguments; a colour <code>c</code> (as an RGB hex string), a percentage <code>p</code> , and a colour name <code>name</code> . A new colour <code>name</code> will be created, which is the colour <code>c</code> lightened by <code>p%</code> . This uses <code>pgfmath</code> and is not especially fast or efficient, so calls to this command should be minimised where possible. |
| <code>\darken{c}{p}{name}</code> | This acts identically to <code>\lighten</code> , except that the new colour <code>name</code> will be <code>p%</code> darker than <code>c</code> . |
| <code>\getLightness{c}</code> | This takes a colour <code>c</code> (as an RGB hex string) and creates a <code>pgfmath</code> macro <code>\lightness</code> which is set to the lightness of the colour <code>c</code> . Internally, it also sets values for <code>pgfmath</code> macros <code>\r</code> , <code>\g</code> , and <code>\b</code> . |