

# palette – Create palettes for colors and symbols

Dennis Chen  
proofprogram@gmail.com

v1.1.0 (2021/07/29)

## Abstract

The `palette` package provides `colorpalette.sty` and `symbolpalette.sty`, which can be used to create and manage palettes of colors and symbols, respectively.

## 1 Overview

We will go over the basic structure of the user interface in broad terms first (i.e. no macros or environments yet). We will use `colorpalette` as our example here, but `symbolpalette` behaves virtually identically.

First, the user can create palette *themes*, which are independent from each other. Then, you can add colors to themes. Finally, you can make a palette associated with a theme and set its colors.

## 2 `colorpalette.sty`

### 2.1 Usage

<code>\newpalettetheme</code>	To create a new palette theme called <code>theme</code> , write <code>\newpalettetheme{theme}</code> .
<code>\addcolortotheme</code>	To add a color <code>color</code> to theme <code>theme</code> , write <code>\addcolortotheme{theme}{color}</code> .
<code>\newpalette</code>	To create a new palette <code>palette</code> based on theme <code>theme</code> , write <code>\newpalette{theme}{palette}</code> . Palette names only need to be unique across the same theme. This means that you can have two palettes with the same name if they are associated with different themes.
<code>\setpalettecolor</code>	To set the color <code>color</code> (which was defined with respect to the theme of the palette) as the color value <code>value</code> in the format <code>encoding</code> (which by default is HTML) of palette <code>palette</code> in theme <code>theme</code> , write <code>\setpalettecolor{theme}{palette}{color}[encoding]{value}</code>
<code>\activepalette</code>	This sets the active palette of a theme. (This is how palettes are “swapped in” with respect to a theme.) To set the active palette of theme <code>theme</code> as <code>palette</code> , write <code>\activepalette{theme}{palette}</code> .
<code>\getcolor</code>	To use, write <code>\getcolor{theme}{color}</code> . This gets the <i>internal</i> color name

from the *active* palette of *theme*. No errors will be thrown even if you pass in an undefined theme or color (with respect to the theme). This limitation is because we must be able to parse `\getcolor`.

`\applycolor` To use, write `\applycolor{theme}{color}`. This applies color *color* from the *active* palette of theme *theme*. (This means that the newly defined color is passed into `\color`.) If *theme* has no active palette, an error will be thrown.

To make it clearer what `\applycolor` does, we will just show the macro definition directly (since it is quite simple). (The macros `\colorpalette@themecheck` and `\colorpalette@colorcheck` are just for error handling.)

```
1 <*color>
2 \newcommand\applycolor[2]{%
3   \colorpalette@themecheck{#1}%
4   \colorpalette@colorcheck{#1}{#2}%
5   \color{\getcolor{#1}{#2}}%
6 }
7 </color>
```

## 2.2 Error Messages

If a nonexistent theme is passed in, the following error will be thrown:

```
The palette passed in is not a defined palette of theme '#1'
```

where *#1* is the name of the theme.

If a palette that does not belong to a theme is passed in, the following error will be thrown:

```
The palette '#2' passed in is not a defined palette of theme '#1'
```

where *#1* is the name of the theme and *#2* is the name of the palette.

If a color that does not belong to a theme is passed in, the following error will be thrown:

```
The color '#2' passed in is not a defined color of theme '#1'
```

where *#1* is the name of the theme and *#2* is the name of the color.

## 3 `symbolpalette.sty`

Despite the name `symbolpalette` (which is used to identify it as one of the files of the package `palette`), the main keyword is a *suite*. This is because the keyword *palette* is already used for `colorpalette`.

This package is nearly analogous to `colorpalette`. A *suite* is analogous to a *palette*, and a *symbol* is analogous to a *color*. *Themes* are used in both packages.

### 3.1 Usage

<code>\newsuitetheme</code>	To create a new suite theme called <code>theme</code> , write <code>\newsuitetheme{theme}</code> .
<code>\addsymboltotheme</code>	To add a symbol <code>symbol</code> to theme <code>theme</code> , write <code>\addsymboltotheme{theme}{symbol}</code> .
<code>\newsuite</code>	To create a new suite <code>suite</code> based on theme <code>theme</code> , write <code>\newsuite{theme}{suite}</code> . Suite names only need to be unique across the same theme. This means that you can have two suites with the same name if they are associated with different themes.
<code>\setsuitesymbol</code>	To set the symbol <code>symbol</code> (which was defined with respect to the theme of the suite) as the icon <code>icon</code> of suite <code>suite</code> in theme <code>theme</code> , write  <code>\setsuitesymbol{theme}{suite}{symbol}{icon}</code>
<code>\activesuite</code>	This sets the active suite of a theme. (This is how suites are “swapped in” with respect to a theme.) To set the active suite of theme <code>theme</code> as <code>suite</code> , write <code>\activesuite{theme}{suite}</code> .
<code>\printsymbol</code>	To use, write <code>\printsymbol{theme}{symbol}</code> . This is similar to <code>colorpalette.sty</code> ’s <code>\applycolor</code> .

### 3.2 Error Messages

If a nonexistent theme is passed in, the following error will be thrown:

```
The palette passed in is not a defined palette of theme ‘#1’
```

where `#1` is the name of the theme.

If a suite that does not belong to a theme is passed in, the following error will be thrown:

```
The suite ‘#2’ passed in is not a defined suite of theme ‘#1’
```

where `#1` is the name of the theme and `#2` is the name of the suite.

If a symbol that does not belong to a theme is passed in, the following error will be thrown:

```
The symbol ‘#2’ passed in is not a defined symbol of theme ‘#1’
```

where `#1` is the name of the theme and `#2` is the name of the symbol.

## Change History

v1.0.0		to symbol, not color . . . . . 3
General: Initial version . . . . . 1	v1.1.0	
v1.0.1		General: Update macrolist (to get compatibility with etoolbox) . . 1
General: Make error message refer		