

matplotlib

Fork me on GitHub

[home](#) | [examples](#) | [gallery](#) | [pyplot](#) | [docs](#) » [Matplotlib Examples](#) » [previous](#) | [next](#) | [modules](#) | [index](#)

[color Examples](#) »

color example code: colormaps_reference.py

[\(Source code\)](#)

Depsy 100th percentile

Travis-CI: build passing

Related Topics

Perceptually Uniform Sequential colormaps



[notation overview](#)

[matplotlib Examples](#)

[color Examples](#)

Previous: [color example code:](#)

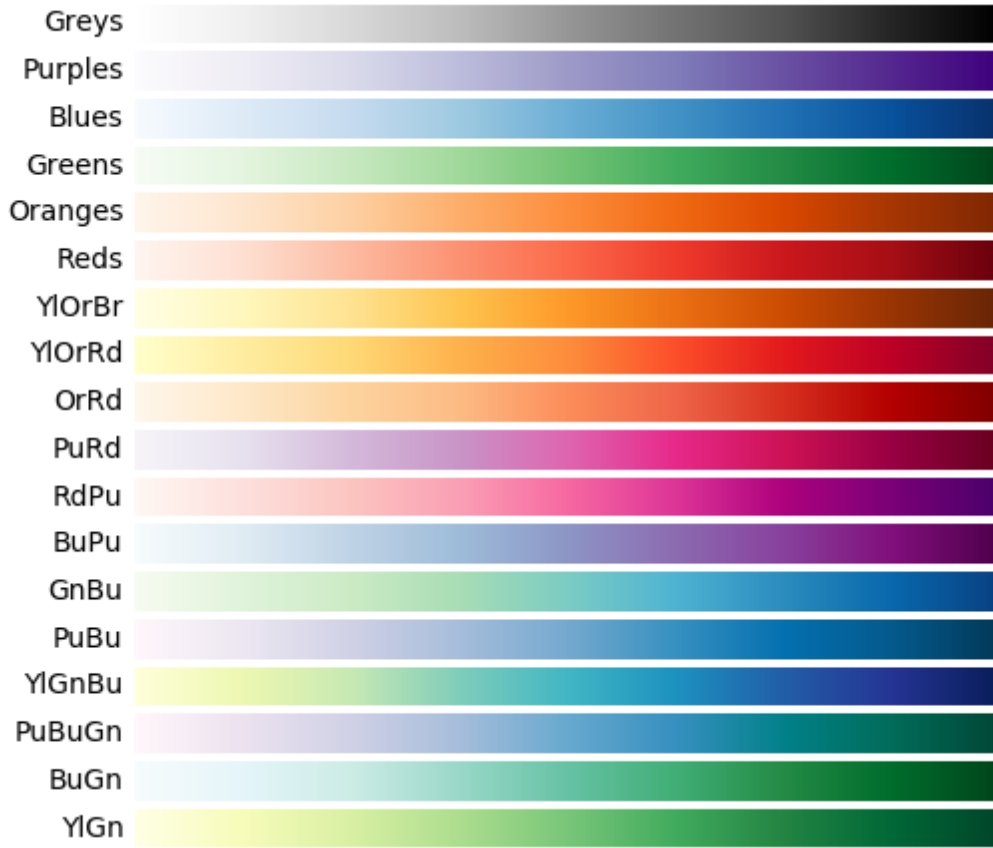
[color_cycle_demo.py](#)

Next: [color example code: named_colors.py](#)

[source](#)

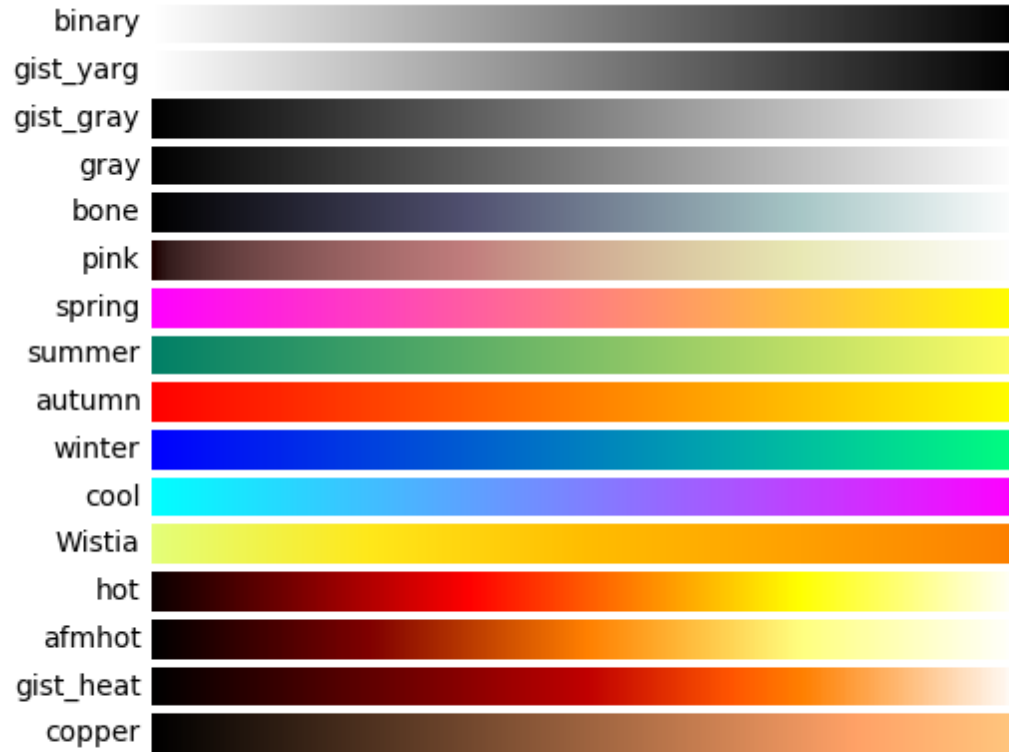
[\(png, pdf\)](#)

Sequential colormaps

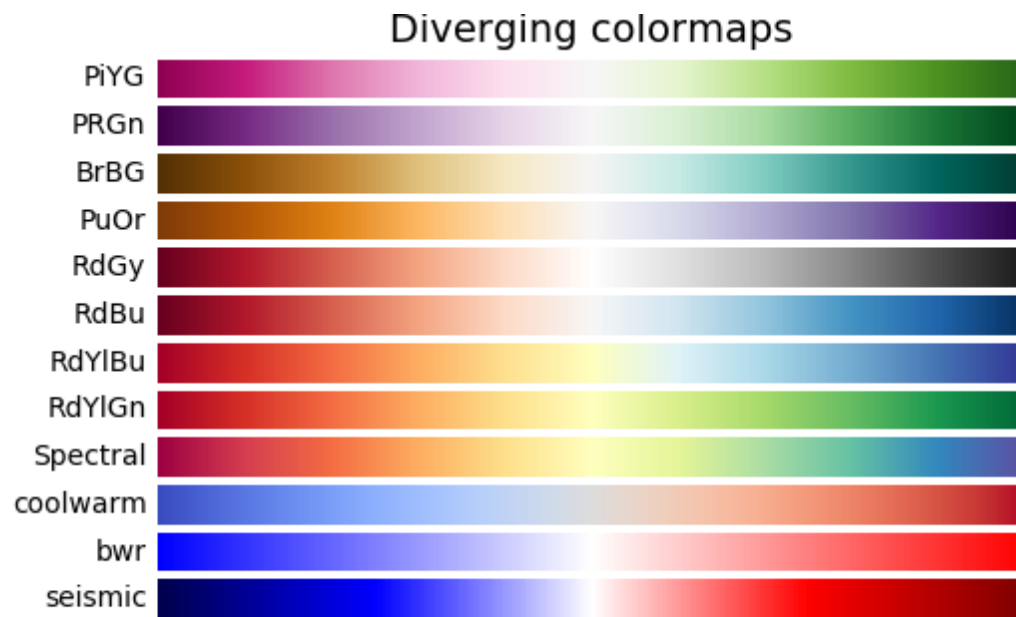


([png](#), [pdf](#))

Sequential (2) colormaps

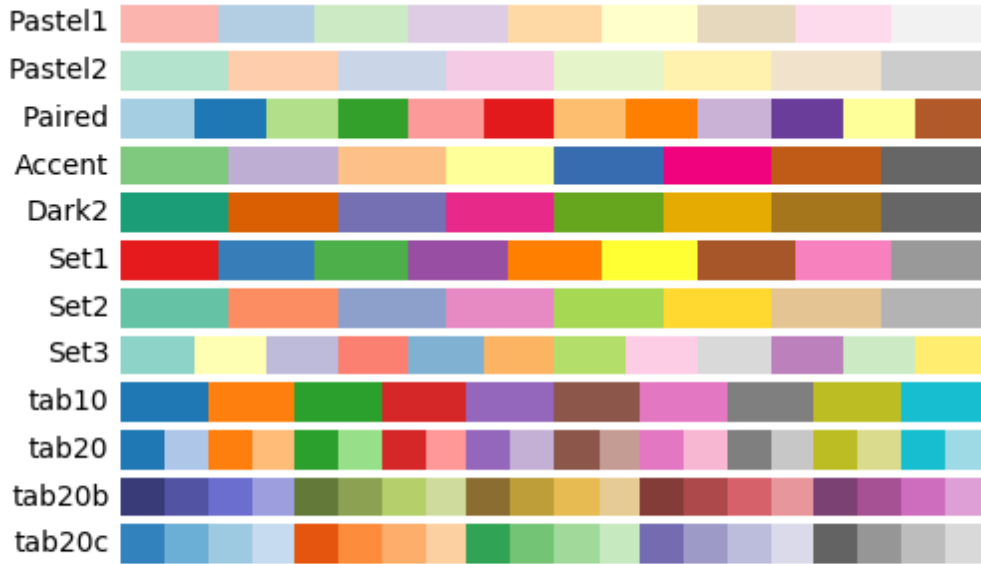


([png](#), [pdf](#))



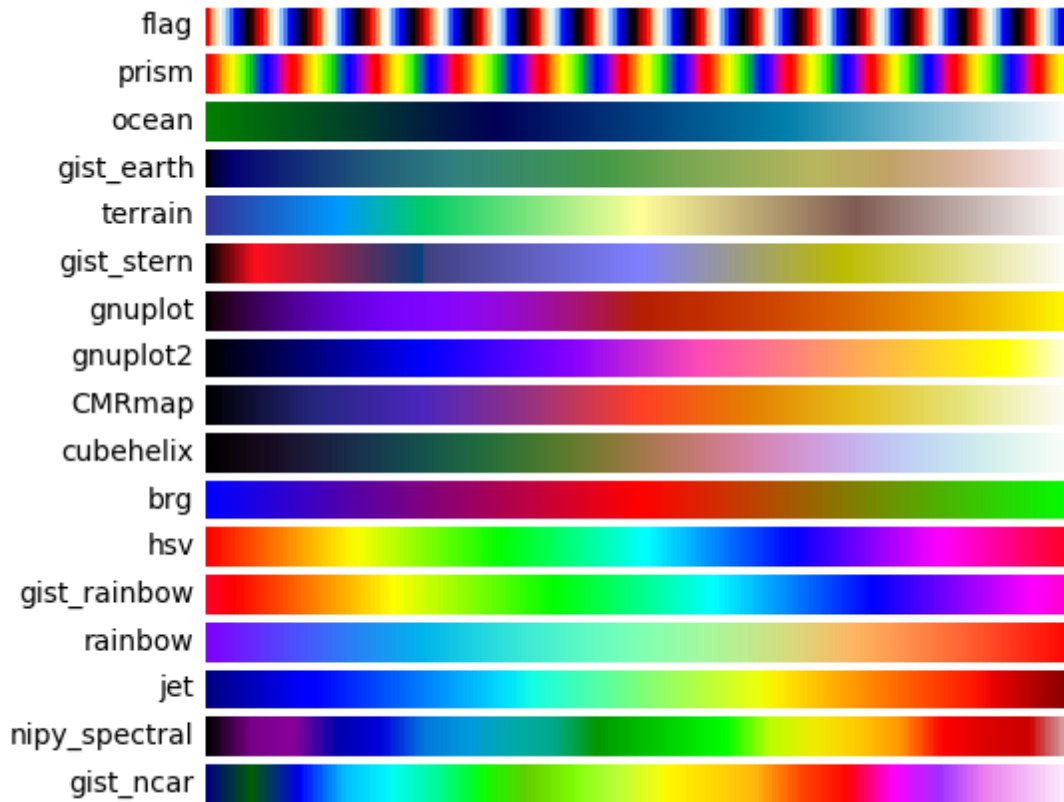
([png](#), [pdf](#))

Qualitative colormaps



[\(png, pdf\)](#)

Miscellaneous colormaps



([png](#), [pdf](#))

```

"""
=====
Colormap reference
=====

Reference for colormaps included with Matplotlib.

This reference example shows all colormaps included with
any colormap listed here can be reversed by appending "_r"
These colormaps are divided into the following categories

Sequential:
    These colormaps are approximately monochromatic color
    between two color tones---usually from low saturation
    saturation (e.g. a bright blue). Sequential colormaps
    representing most scientific data since they show a
    low-to-high values.

Diverging:
    These colormaps have a median value (usually light) and
    smoothly to two different color tones at high and low
    colormaps are ideal when your data has a median value
    (e.g. 0, such that positive and negative values are
    different colors of the colormap).

```

Qualitative:

These colormaps vary rapidly in color. Qualitative colormaps are good for choosing a set of discrete colors. For example::

```
color_list = plt.cm.Set3(np.linspace(0, 1, 12))
```

gives a list of RGB colors that are good for plotting on a dark background.

Miscellaneous:

Colormaps that don't fit into the categories above.

```
"""
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# Have colormaps separated into categories:
```

```
# http://matplotlib.org/examples/color/colormaps_reference.html
```

```
cmaps = [('Perceptually Uniform Sequential', [
    'viridis', 'plasma', 'inferno', 'magma']),
 ('Sequential', [
    'Greys', 'Purples', 'Blues', 'Greens', 'Oranges',
    'YlOrBr', 'YlOrRd', 'OrRd', 'PuRd', 'RdPu',
    'GnBu', 'PuBu', 'YlGnBu', 'PuBuGn', 'BuGn',
 ('Sequential (2)', [
    'binary', 'gist_yarg', 'gist_gray', 'gray',
    'spring', 'summer', 'autumn', 'winter', 'cool',
    'hot', 'afmhot', 'gist_heat', 'copper']),
 ('Diverging', [
    'PiYG', 'PRGn', 'BrBG', 'PuOr', 'RdGy', 'RdBu',
    'RdYlBu', 'RdYlGn', 'Spectral', 'coolwarm',
 ('Qualitative', [
    'Pastel1', 'Pastel2', 'Paired', 'Accent',
    'Dark2', 'Set1', 'Set2', 'Set3',
    'tab10', 'tab20', 'tab20b', 'tab20c']),
 ('Miscellaneous', [
    'flag', 'prism', 'ocean', 'gist_earth', 'terrain',
    'gnuplot', 'gnuplot2', 'CMRmap', 'cubehelix',
    'gist_rainbow', 'rainbow', 'jet', 'nipy_spectral']])
```

```
nrows = max(len(cmap_list) for cmap_category, cmap_list
```

```
gradient = np.linspace(0, 1, 256)
```

```
gradient = np.vstack((gradient, gradient))
```

```
def plot_color_gradients(cmap_category, cmap_list, nrows):
```

```
    fig, axes = plt.subplots(nrows=nrows)
```

```
    fig.subplots_adjust(top=0.95, bottom=0.01, left=0.2,
```

```
    axes[0].set_title(cmap_category + ' colormaps', font
```

```
    for ax, name in zip(axes, cmap_list):
```

```
        ax.imshow(gradient, aspect='auto', cmap=plt.get
```

```
        pos = list(ax.get_position().bounds)
```

```
x_text = pos[0] - 0.01
y_text = pos[1] + pos[3]/2.
fig.text(x_text, y_text, name, va='center', ha='center')

# Turn off all ticks & spines, not just the ones w
for ax in axes:
    ax.set_axis_off()

for cmap_category, cmap_list in cmaps:
    plot_color_gradients(cmap_category, cmap_list, nrow=2)

plt.show()
```

Keywords: python, matplotlib, pylab, example, codex (see [Search examples](#))