

Quick start

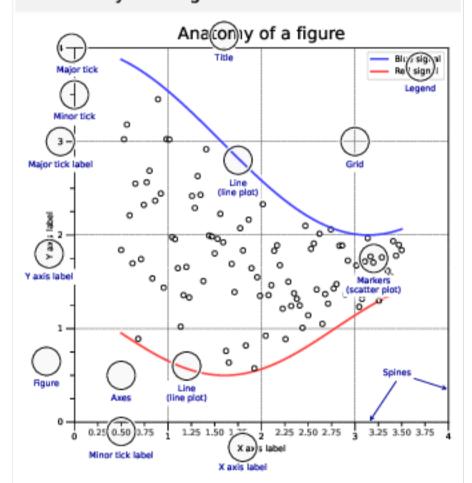
import numpy as np import matplotlib as mpl import matplotlib.pyplot as plt

X = np.linspace(0, 2*np.pi, 100)Y = np.cos(X)

fig, ax = plt.subplots() ax.plot(X, Y, color='green')

fig.savefig("figure.pdf") fig.show()

Anatomy of a figure



Subplots layout

subplot[s](rows,cols,...) fig, axs = plt.subplots(3, 3)

> G = gridspec(rows,cols,...) API ax = G[0,:]

ax.inset_axes(extent)

d=make_axes_locatable(ax) ax = d.new_horizontal('10%')

Getting help

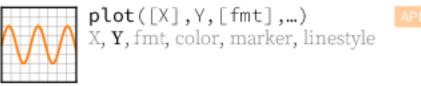
matplotlib.org

pithub.com/matplotlib/matplotlib/issues O discourse.matplotlib.org

stackoverflow.com/questions/tagged/matplotlib

| gitter.im/matplotlib twitter.com/matplotlib

Basic plots



scatter(X,Y,...) X, Y, [s]izes, [c]olors, marker, cmap

bar[h](x,height,...) x, height, width, bottom, align, color

imshow(Z,…) Z, cmap, interpolation, extent, origin

contour[f]([X],[Y],Z,...) X, Y, Z, levels, colors, extent, origin

pcolormesh([X],[Y],Z,...) X, Y, Z, vmin, vmax, cmap

quiver([X],[Y],U,V,...) X, Y, U, V, C, units, angles

pie(X,...) Z, explode, labels, colors, radius

text(x,y,text,...) x, y, text, va, ha, size, weight, transform

fill[_between][x](...) X, Y1, Y2, color, where

Advanced plots

API

step(X,Y,[fmt],...) X, Y, fmt, color, marker, where boxplot(X,...)

X, notch, sym, bootstrap, widths

errorbar(X,Y,xerr,yerr,...) API X, Y, xerr, yerr, fmt

hist(X, bins, ...) X, bins, range, density, weights

violinplot(D,...) **D**, positions, widths, vert

barbs([X],[Y], U, V, ...) X, Y, U, V, C, length, pivot, sizes

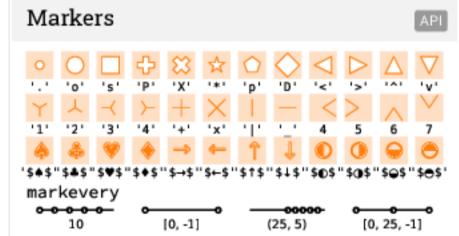
eventplot(positions,…) positions, orientation, lineoffsets

hexbin(X,Y,C,...) X, Y, C, gridsize, bins

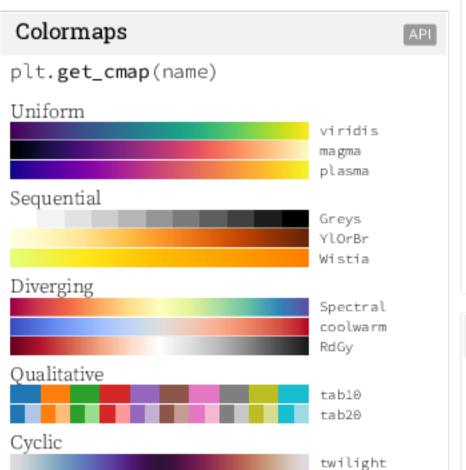
Scales ax.set_[xy]scale(scale,...) MMMMMM. linear log values > 0 any values M symlog **∕∖**/∕ logit 0 < values < 1 any values

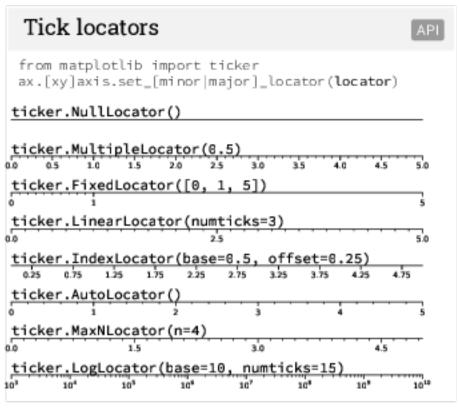
Projections subplot(...,projection=p) p='polar' p=Orthographic() from cartopy.crs import Cartographic

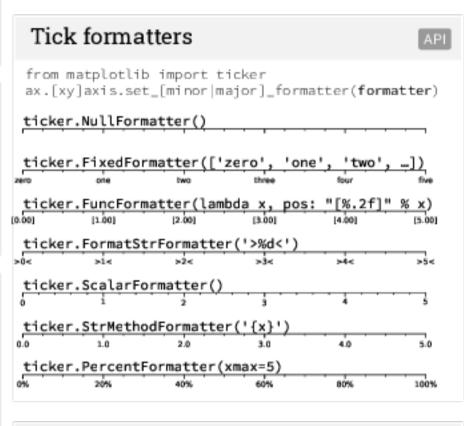




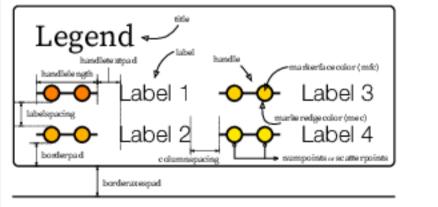


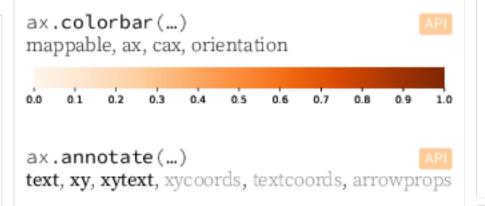








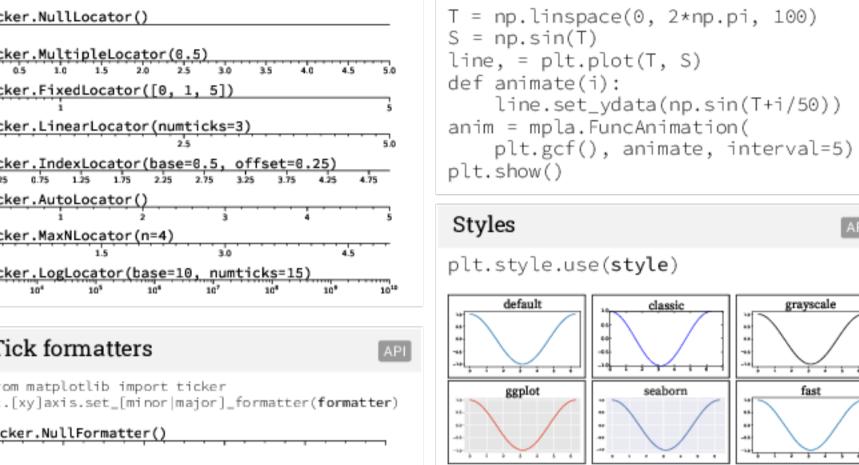






Event handling

fig, ax = plt.subplots() def on_click(event): print(event) fig.canvas.mpl_connect('button_press_event', on_click)



Animation

import matplotlib.animation as mpla

Quick reminder

ax.grid() ax.set_[xy]lim(vmin, vmax) ax.set_[xy]label(label) ax.set_[xy]ticks(ticks, [labels]) ax.set_[xy]ticklabels(labels) ax.set_title(title)

Solarize_Light2

seaborn-notebook

ax.tick_params(width=10, ...) ax.set_axis_[on|off]()

fig.suptitle(title) fig.tight_layout() plt.gcf(), plt.gca() mpl.rc('axes', linewidth=1, ...) [fig|ax].patch.set_alpha(0) text=r'\$\frac{-e^{i\pi}}{2^n}\$'

Keyboard shortcuts

ctrl+ w Close plot ctrl + s Save f Fullscreen 0/1 r Reset view b View back f View forward p Pan view O Zoom to rect X X pan/zoom y Y pan/zoom G Major grid 0/1 g | Minor grid 0/1 X axis log/linear Y axis log/linear

Ten simple rules

1. Know Your Audience

2. Identify Your Message

3. Adapt the Figure

4. Captions Are Not Optional

5. Do Not Trust the Defaults

6. Use Color Effectively

7. Do Not Mislead the Reader

Avoid "Chartjunk"

9. Message Trumps Beauty

10. Get the Right Tool

READ