

Ouick start

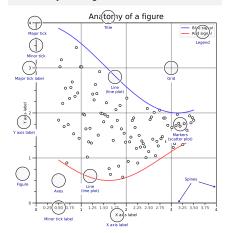
import numpy as np import matplotlib as mpl import matplotlib.pyplot as plt

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") plt.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) API ax = d.new_horizontal('10%')

Getting help

matplotlib.org

github.com/matplotlib/matplotlib/issues

discourse.matplotlib.org

stackoverflow.com/questions/tagged/matplotlib https://gitter.im/matplotlib/matplotlib

y twitter.com/matplotlib

✓ Matplotlib users mailing list



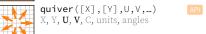
scatter(X,Y,...) X, Y, [s]izes, [c]olors, marker, cmap



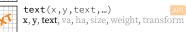










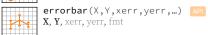




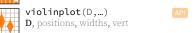
Advanced plots

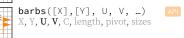
API





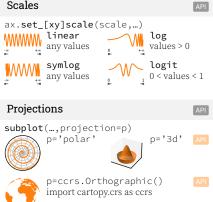


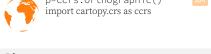




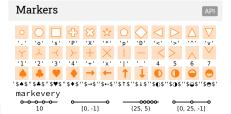




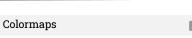






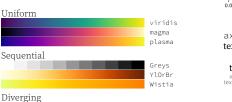




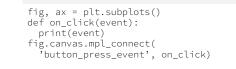


plt.get_cmap(name)

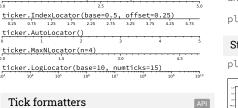
Cyclic

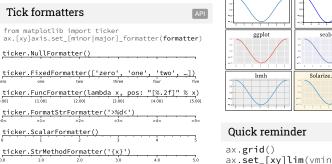






```
from matplotlib import ticker
ax.[xy]axis.set [minor|major] locator(locator)
ticker.NullLocator()
                                                    S = np.sin(T)
ticker.MultipleLocator(0.5)
  0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
                                                     def animate(i):
ticker.FixedLocator([0, 1, 5])
ticker.LinearLocator(numticks=3)
ticker.IndexLocator(base=0.5, offset=0.25)
                                                     plt.show()
ticker.AutoLocator()
                                                     Styles
ticker.MaxNLocator(n=4)
```



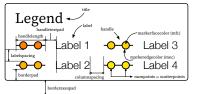




ticker.PercentFormatter(xmax=5)

Tick locators

ax.legend(...) handles, labels, loc, title, frameon







0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9



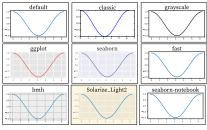
Event handling

Animation

import matplotlib.animation as mpla

```
T = np.linspace(0, 2*np.pi, 100)
line, = plt.plot(T, S)
    line.set_ydata(np.sin(T+i/50))
anim = mpla.FuncAnimation(
    plt.gcf(), animate, interval=5)
```

plt.style.use(style)



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) 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 + s Save ctrl + w Close plot r Reset view f Fullscreen 0/1

b View back

f View forward p Pan view

O Zoom to rect x X pan/zoom y Y pan/zoom

g Minor grid 0/1

G Major grid 0/1

X axis log/linear L 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

8. Avoid "chartiunk"

9. Message trumps beauty 10. Get the right tool

