

Ouick start

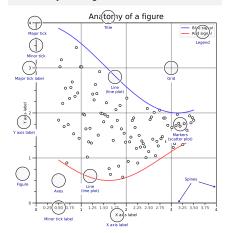
import numpy as np import matplotlib as mpl import matplotlib.pyplot as plt

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

Getting help

matplotlib.org

github.com/matplotlib/matplotlib/issues

discourse.matplotlib.org

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

ax = d.new_horizontal('10%')

y twitter.com/matplotlib

✓ Matplotlib users mailing list



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



X, Y, xerr, yerr, fmt

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

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

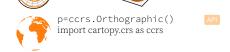
barbs([X],[Y], U, V, ...) X, Y, U, V, C, length, pivot, sizes

eventplot(positions,...) positions, orientation, lineoffsets

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

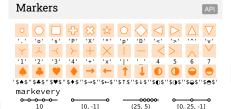
Scales ax.set_[xy]scale(scale,...) MAMAMAMA linear log any values values > 0 symlog logit 0 < values < 1 any values **Projections**

subplot(...,projection=p) p='polar' p='3d'



Lines

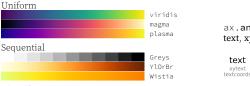






Colormaps

plt.get_cmap(name)







Tick locators

from matplotlib import ticker ax.[xy]axis.set [minor|major] locator(locator)

ticker.NullLocator() ticker.MultipleLocator(0.5) 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 ticker.FixedLocator([0, 1, 5]) ticker.LinearLocator(numticks=3) ticker.IndexLocator(base=0.5, offset=0.25)

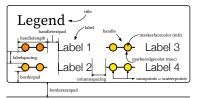
ticker.AutoLocator() ticker.MaxNLocator(n=4) ticker.LogLocator(base=10, numticks=15)

Tick formatters

from matplotlib import ticker ax.[xy]axis.set_[minor|major]_formatter(formatter) ticker.NullFormatter() ticker.FixedFormatter(['zero', 'one', 'two', ...]) ticker.FuncFormatter(lambda x, pos: "[%.2f]" % x) [2.00] ticker.FormatStrFormatter('>%d<') ticker.ScalarFormatter() ticker.StrMethodFormatter('{x}') ticker.PercentFormatter(xmax=5)

Ornaments

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



ax.colorbar(...) mappable, ax, cax, orientation

0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

ax.annotate(...) text, xy, xytext, xycoords, textcoords, arrowprops



Event handling

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

Animation

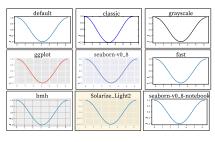
import matplotlib.animation as mpla

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

Styles

API

plt.style.use(style)



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) 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

f View forward

p Pan view

x X pan/zoom

g Minor grid 0/1

G Major grid 0/1 X axis log/linear L Y axis log/linear

b View back

O Zoom to rect

y Y pan/zoom

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

