

#### **Ouick** start

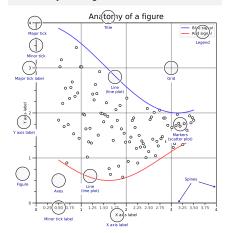
import numpy as np import matplotlib as mpl import matplotlib.pyplot as plt

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

API 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

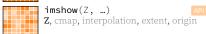
**y** twitter.com/matplotlib ✓ Matplotlib users mailing list

### Basic plots



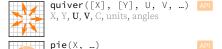
scatter(X, Y, ...) X, Y, [s]izes, [c]olors, marker, cmap

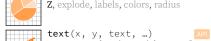












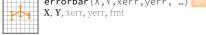


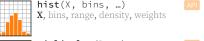
X, Y1, Y2, color, where

# Advanced plots

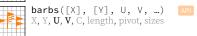




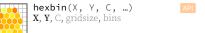










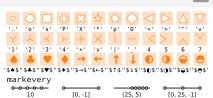


#### Scales ax.set\_[xy]scale(scale, ...) MAMAMAMA linear \/ log any values values > 0 symlog logit 0 < values < 1 any values **Projections** subplot(..., projection=p)

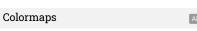












plt.get\_cmap(name)

Cyclic



Spectral

coolwarm

# Event handling

Tick locators

ticker.NullLocator()

ticker.AutoLocator()

ticker.MaxNLocator(n=4)

Tick formatters

ticker.NullFormatter()

ticker.ScalarFormatter()

Ornaments

ax.legend(...)

Legend -

ax.colorbar(...)

from matplotlib import ticker

from matplotlib import ticker

ticker.MultipleLocator(0.5)

ticker.FixedLocator([0, 1, 5])

ticker.LinearLocator(numticks=3)

ax.[xy]axis.set [minor|major] locator(locator)

0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0

ticker.IndexLocator(base=0.5, offset=0.25)

ticker.LogLocator(base=10, numticks=15)

ax.[xy]axis.set\_[minor|major]\_formatter(formatter)

ticker.FixedFormatter(['zero', 'one', 'two', ...])

ticker.FuncFormatter(lambda x, pos: "[%.2f]" % x)

[2.00]

ticker.FormatStrFormatter('>%d<')

ticker.StrMethodFormatter('{x}')

ticker.PercentFormatter(xmax=5)

handles, labels, loc, title, frameon

abel 1

Label 2

mappable, ax, cax, orientation

Label 3

fig, ax = plt.subplots() def on\_click(event): print(event) fig.canvas.mpl\_connect( 'button\_press\_event', on\_click)

### Animation

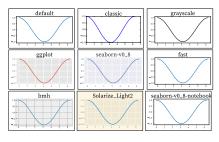
import matplotlib.animation as mpla

```
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

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 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

