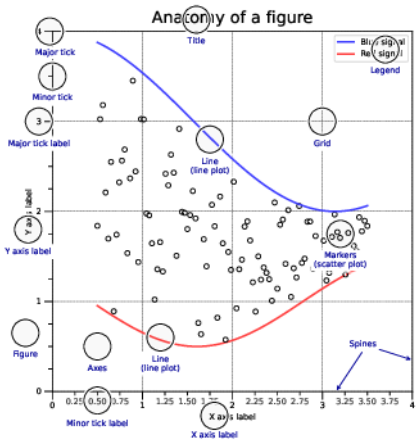
matplotlib Cheat sheet

Version 3.5.0

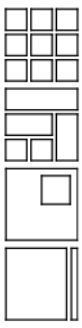
Quick start

```
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,…)
ax = G[0,:]

ax.inset_axes(extent)

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

Getting help

- matplotlib.org
- github.com/matplotlib/matplotlib/issues
- discourse.matplotlib.org
- stackoverflow.com/questions/tagged/matplotlib
- gitter.im/matplotlib
- twitter.com/matplotlib
- Matplotlib users mailing list

Basic plots

```
plot([X],Y,[fmt],…)
X, Y, fmt, color, marker, linestyle
```

```
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

```
step(X,Y,[fmt],…)
X, Y, fmt, color, marker, where
```

```
boxplot(X,…)
X, notch, sym, bootstrap, widths
```

```
errorbar(X,Y,xerr,yerr,…)
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,…)
```

linear — any values
log — values > 0
symlog — any values
logit — 0 < values < 1

Projections

```
subplot(…,projection=p)
```

p='polar'

p='3d'

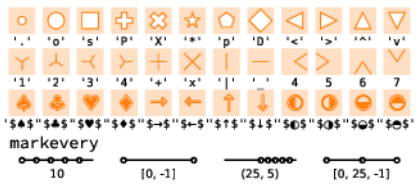
p=Orthographic()
from cartopy.crs import Cartographic

Lines

```
linestyle or ls
```


```
capstyle or dash_capstyle
```

Markers



Colors

Colormaps

```
plt.get_cmap(name)
```