Scales

API

```
ax.set_[xy]scale(scale,…)
```

linear
any values

log
values > 0

symlog
any values

logit
0 < values < 1

Projections

API

```
subplot(…,projection=p)
```

p='polar'

p='3d'

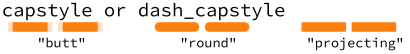
p=Orthographic()
API
from cartopy.crs import Cartographic

Lines

API

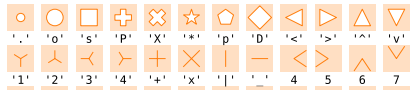
linestyle or ls

capstyle or dash_capstyle

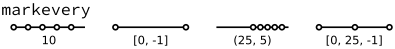


Markers

API

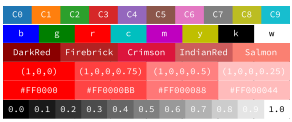


markevery



Colors

API



'Cn'
'x'
'name'
(R,G,B[,A])
'#RRGGBB[AA]'
'x.y'

Colormaps

API

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

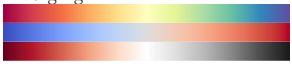
Uniform

viridis
magma
plasma

Sequential

Greys
YlOrBr
Wistia

Diverging



Spectral
coolwarm
RdGy

Qualitative

tab10
tab20

Cyclic

twilight

Tick locators

API

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

```
ticker.NullLocator()
ticker.MultipleLocator(0.5)
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

API

```
from matplotlib import ticker
ax.[xy]axis.set_[minor|major]_formatter(formatter)
```

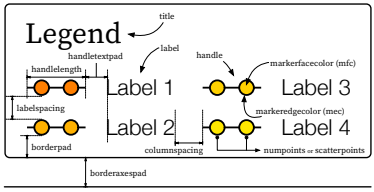
```
ticker.NullFormatter()
ticker.FixedFormatter(['', '0', '1', ...])
ticker.FuncFormatter(lambda x, pos: "[%.2f]" % x)
ticker.FormatStrFormatter('>%d<')
ticker.ScalarFormatter()
ticker.StrMethodFormatter('{x}')
ticker.PercentFormatter(xmax=5)
```

Ornaments

API

```
ax.legend(…)
```

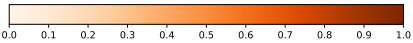
handles, labels, loc, title, frameon



API

```
ax.colorbar(…)
```

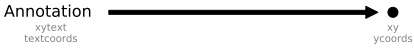
mappable, ax, cax, orientation



API

```
ax.annotate(…)
```

text, **xy**, **xytext**, xycoords, textcoords, arrowprops



Event handling

API

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