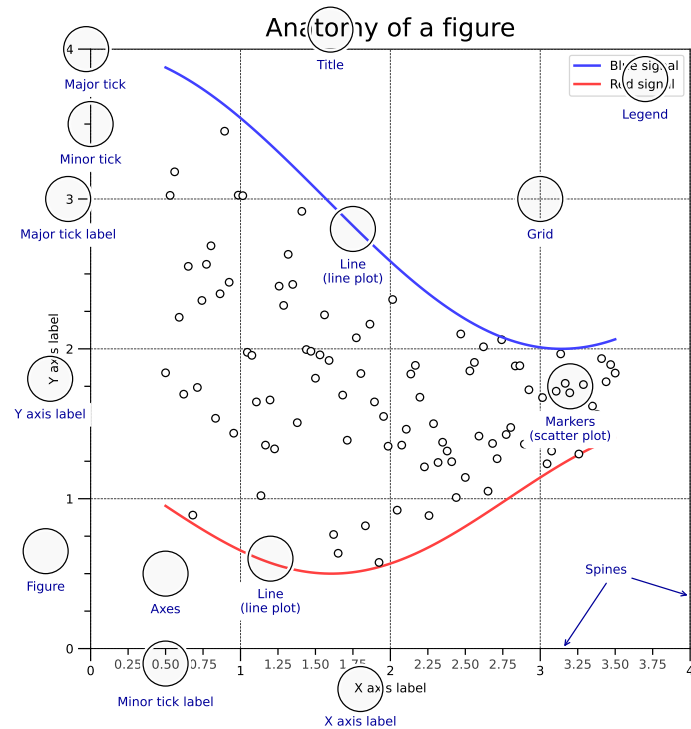


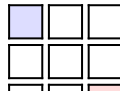
# Matplotlib for intermediate users

A matplotlib figure is composed of a hierarchy of elements that forms the actual figure. Each element can be modified.

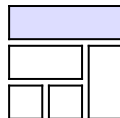


## Figure, axes & spines

```
fig, axs = plt.subplots(3, 3)
axs[0, 0].set_facecolor("#dddfff")
axs[2, 2].set_facecolor("#ffffdd")
```



```
gs = fig.add_gridspec(3, 3)
ax = fig.add_subplot(gs[0, :])
ax.set_facecolor("#dddfff")
```

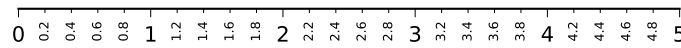


```
fig, ax = plt.subplots()
ax.spines["top"].set_color("None")
ax.spines["right"].set_color("None")
```



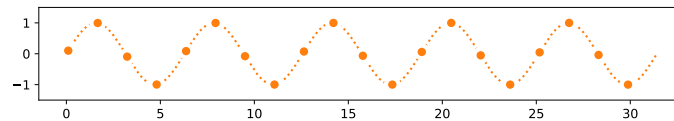
## Ticks & labels

```
from mpl.ticker import MultipleLocator as ML
from mpl.ticker import ScalarFormatter as SF
ax.xaxis.set_minor_locator(ML(0.2))
ax.xaxis.set_minor_formatter(SF())
ax.tick_params(axis='x', which='minor', rotation=90)
```



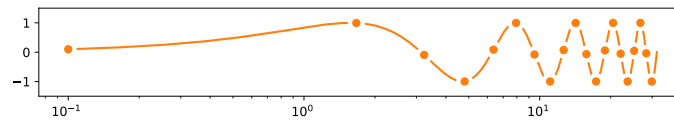
## Lines & markers

```
X = np.linspace(0.1, 10*np.pi, 1000)
Y = np.sin(X)
ax.plot(X, Y, "C1o:", markevery=50, mec="1.0")
```



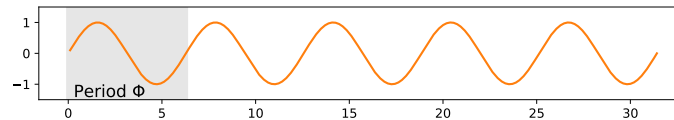
## Scales & projections

```
fig, ax = plt.subplots()
ax.set_xscale("log")
ax.plot(X, Y, "C1o-", markevery=50, mec="1.0")
```



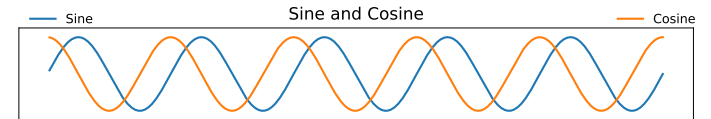
## Text & ornaments

```
ax.fill_betweenx([-1, 1], [0], [2*np.pi])
ax.text(0, -1, r"Period $\Phi$")
```



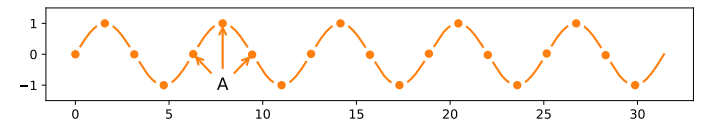
## Legend

```
ax.plot(X, np.sin(X), "C0", label="Sine")
ax.plot(X, np.cos(X), "C1", label="Cosine")
ax.legend(bbox_to_anchor=(0, 1, 1, .1), ncol=2,
         mode="expand", loc="lower left")
```



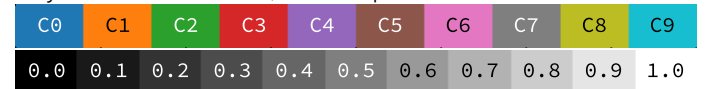
## Annotation

```
ax.annotate("A", (X[250], Y[250]), (X[250], -1),
          ha="center", va="center", arrowprops={
            "arrowstyle": "->", "color": "C1"})
```



## Colors

Any color can be used, but Matplotlib offers sets of colors:



## Size & DPI

Consider a square figure to be included in a two-column A4 paper with 2 cm margins on each side and a column separation of 1 cm. The width of a figure is  $(21 - 2 \times 2 - 1) / 2 = 8$  cm. One inch being 2.54 cm, figure size should be 3.15 × 3.15 in.

```
fig = plt.figure(figsize=(3.15, 3.15), dpi=50)
plt.savefig("figure.pdf", dpi=600)
```