

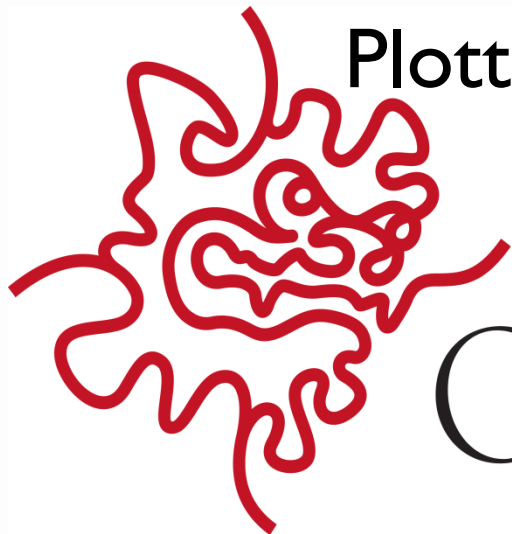


**SKILLPILLS**

# SKILL PILL: Programming with Python

Lecture:

Plotting in Python with matplotlib



OIST

# Create and plot some data

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

```
# Create data
```

```
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
```

```
Y1, Y2 = np.cos(X), np.sin(X)
```

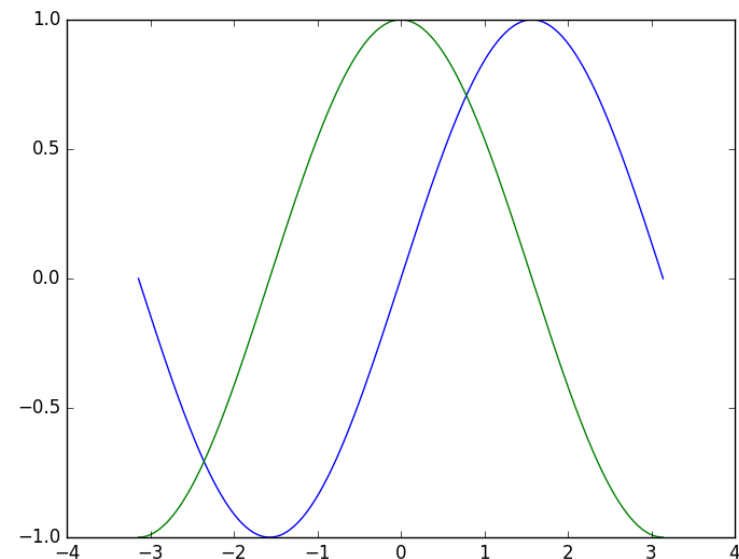
```
# Plot data
```

```
plt.plot(X, Y1)
```

```
plt.plot(X, Y2)
```

```
# Show plot
```

```
plt.show()
```



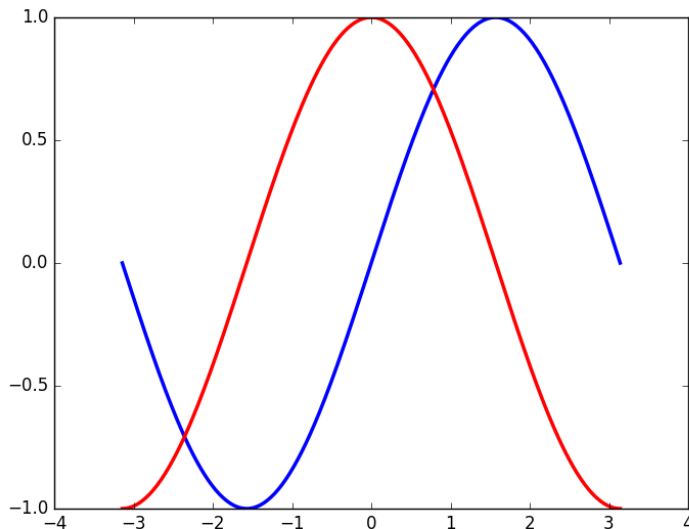
...

*# Change colour, width and style of lines*

```
plt.plot(X, Y1, color="blue", linewidth=2.5, linestyle="-")
```

```
plt.plot(X, Y2, color="red", linewidth=2.5, linestyle="-")
```

...



Other allowed arguments for colour:

“green”, “black”, “white”, “cyan”,  
“magenta”, “yellow”

Other allowed arguments for linestyle:

“-”, “—”, “-.”, “:”, “None”, “ ”



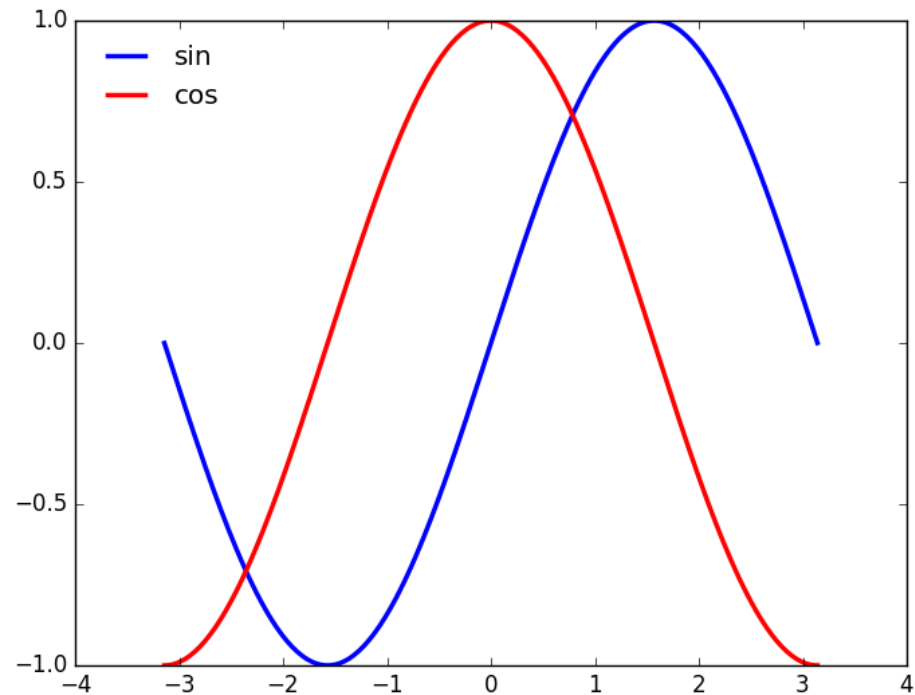
# Add a legend

...

*# Add a legend*

`plt.legend(loc='upper left', frameon=False)`

...



# Add some space

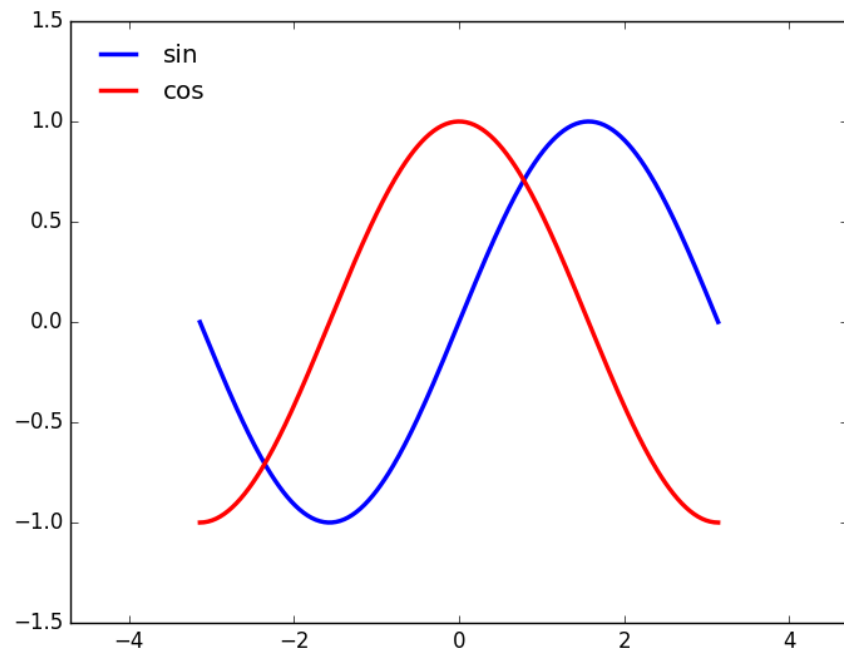
...

*# Add some space*

```
plt.xlim(X.min()*1.5, X.max()*1.5)
```

```
plt.ylim(Y1.min()*1.5, Y1.max()*1.5)
```

...



# Change the ticks

```
...  
# Change the position of ticks  
plt.xticks( [-np.pi, -np.pi/2, 0, np.pi/2, np.pi])  
plt.yticks([-1, 0, +1])  
  
# Label the ticks  
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi],  
           [r'$-abc$', r'$-abc/2$', r'$0$', r'$+abc/2$', r'$+abc$'])  
  
plt.yticks([-1, 0, +1],  
           [r'$minus$', r'$0$', r'$plus$'])  
...
```

# Move the spines...

...

*# Move the spines*

```
ax = plt.gca()
```

```
ax.spines['right'].set_color('none')
```

```
ax.spines['top'].set_color('none')
```

```
ax.xaxis.set_ticks_position('bottom')
```

```
ax.spines['bottom'].set_position(('data',0))
```

```
ax.yaxis.set_ticks_position('left')
```

```
ax.spines['left'].set_position(('data',0))
```

...

# ... and make the ticks pretty

...

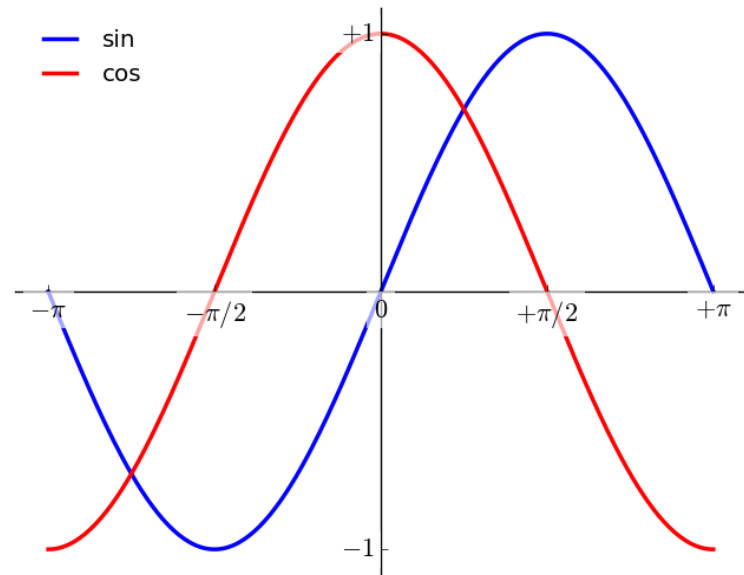
*# Add some space*

```
for label in ax.get_xticklabels() + ax.get_yticklabels():
```

```
    label.set_fontsize(16)
```

```
    label.set_bbox(dict(facecolor='white', edgecolor='None', alpha=0.65))
```

...





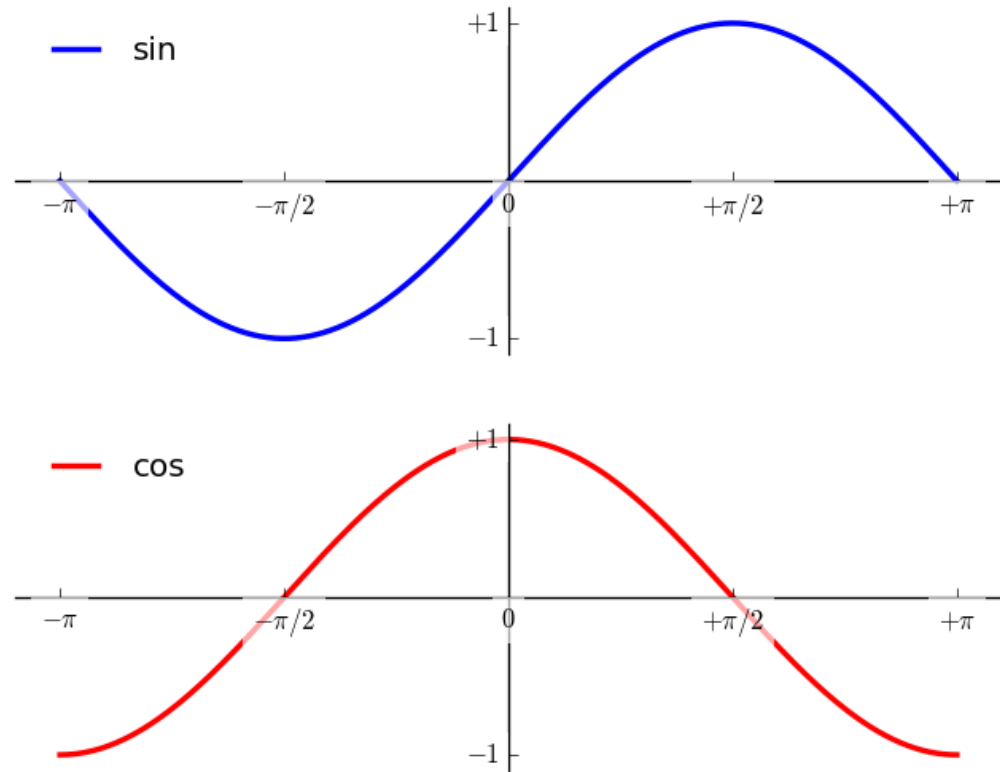
# Split the data over two plots

...

*# Subplot*

```
plt.subplot(211)
```

...



# Make a scatter plot

...

```
X_discret = np.arange(-np.pi, (np.pi+0.1), (np.pi/4))
```

```
Y1_discret, Y2_discret = np.sin(X_discret), np.cos(X_discret)
```

```
# Subplot
```

```
plt.subplot(121)
```

```
plt.scatter(X_discret, Y1_discret, color="cyan", label="sin")
```

...

# Some practice

