

Matplotlib

```
import matplotlib.pyplot as plt
```

```
matplotlib.__version__
```

```
#Getting all available styles  
plt.style.available
```

```
#Using a style  
plt.style.use('ggplot')
```

```
#Controlling figure size  
plt.figure(figsize=(x_size, y_size))
```

Markers, Colors, and Linestyles

Marker: Indicates data points in different shapes.

Style : Indicates style of the line.

Color : Sets color to line

Marker:

character	description
-----------	-------------

.....
'.'	point marker

','	pixel marker
-----	--------------

'o'	circle marker
-----	---------------

'v'	triangle_down marker
-----	----------------------

'^'	triangle_up marker
-----	--------------------

'<'	triangle_left marker
-----	----------------------

'>'	triangle_right marker
-----	-----------------------

'1'	tri_down marker
-----	-----------------

'2'	tri_up marker
-----	---------------

'3'	tri_left marker
-----	-----------------

'4'	tri_right marker
-----	------------------

's'	square marker
-----	---------------

'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin diamond marker
' '	vline marker
'_'	hline marker

Line Style:

character	description
.....
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style

Color:

character	color
.....
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black

'w' white

Use Case:

i) plt.plot(x, y, '<--m', label=label) # Any [marker][style][color] among this anything can be omitted. Ex: '--c'

ii) plt.plot(x, y, color='m', linestyle='--', marker='|', label=label)

Legend Locations

plt.legend(loc='upper right')

lower right, lower left, upper left, lower center, upper center

1) Line Plot

plt.plot(x, y, color, marker, linestyle, label)

plt.title('Title')

plt.xlabel('x-label')

plt.ylabel('y-label')

plt.legend()

plt.grid(True)

plt.savefig('plot.png/jpeg/pdf')

plt.show()

2) BarPlot

vertical plot

plt.bar(x, y, color, label)

horizontal plot

plt.barh(x, y, color, label)

3) Pie Chart

plt.pie(x, labels, wedgeprops={'edgecolor':'black'}, colors, explode, shadow=True, autopct='%1.1f%%')

4) Histogram

```
plt.hist(x, edgecolor='black', bins)
```

bins - a number or a list

5) Scatter Plot

```
plt.scatter(x, y, s, c, marker, edgecolor, alpha, cmap)
```

s - value/list (size), c-color

6) Subplots : *One figure multiple axis*

#Default is 1row, 1col

```
fig, ax = plt.subplots()
```

```
ax.plot(x, y, label, color')
```

```
ax.set_xlabel('x label')
```

```
ax.set_ylabel('y label')
```

```
ax.set_title('Title')
```

```
ax.legend()
```

```
plt.show()
```

```
fig, (ax1, ax2) = plt.subplots(nrows=2, ncols=1)
```

```
ax1.plot(x, y)
```

```
ax2.plot(x, y)
```

```
ax1.set_xlabel('x')
```

```
ax1.set_ylabel('y')
```

```
ax1.set_title('T')
```

```
ax2.set_xlabel('x')
```

```
ax2.set_ylabel('y')
```

```
ax2.set_title('T')
```

```
ax1.legend()
```

```
ax2.legend()
```

```
plt.show()
```

```

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(nrows=2, ncols=2)

ax1.plot(age_x, dev_y, label='All Dev', c='#321E1E')
ax2.plot(age_x, js_dev_y, label='JS Dev', c='#F86F03')
ax3.plot(age_x, py_dev_y, label='Python Dev', c='#090580')
ax4.plot(age_x, jv_dev_y, label='Java Dev', c='#B31312')

ax1.set_ylabel('All Dev')
ax2.set_ylabel('JS Dev')
ax3.set_ylabel('Python Dev')
ax4.set_ylabel('Java Dev')

ax3.set_xlabel('age')
ax4.set_xlabel('age')

fig.savefig('figax.png')

plt.show()

```

7) Subplots : *Multiple figure*

```

# Both figures will be shown simultaneously in different window

fig1, ax1 = plt.subplots()
fig2, ax2 = plt.subplots()

ax1.plot(age_x, dev_y, label='All Dev', c='#321E1E')

ax2.plot(age_x, js_dev_y, label='JS Dev', c='#F86F03')

ax2.legend(loc='upper left')

ax1.set_ylabel('Salary')
ax1.set_xlabel('age')

ax2.set_ylabel('Salary')
ax2.set_xlabel('age')

fig1.savefig('fig1.jpg')
fig2.savefig('fig2.jpg')

plt.show()

```