

▼ Matplotlib

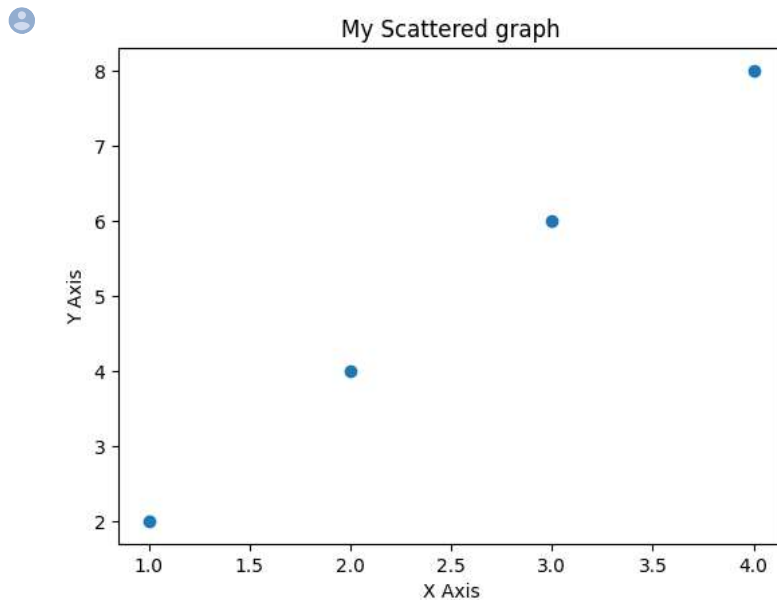
Import Necessary Libraries

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

▼ Scatter Plot

- Plot X values on X axis, Y values on Y axis, using scatter()
- Put title of the Graph, using title()
- Put labels for x and y axis, using xlabel() and ylabel()

```
x=[1,2,3,4]
y=[2,4,6,8]
plt.scatter(x,y)
plt.title('My Scattered graph')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.show()
```



▼ Basic Line Graph

- Plot list of x vs y in a line, using plot()
- Label ticks of x and y axis manually, using xticks() and yticks()
- Show the Graph, using show()
- Add legends for each line, using legend(), and for this, the line plotting must be labeled.
- Style the line using different keyword argument notations inside plot().

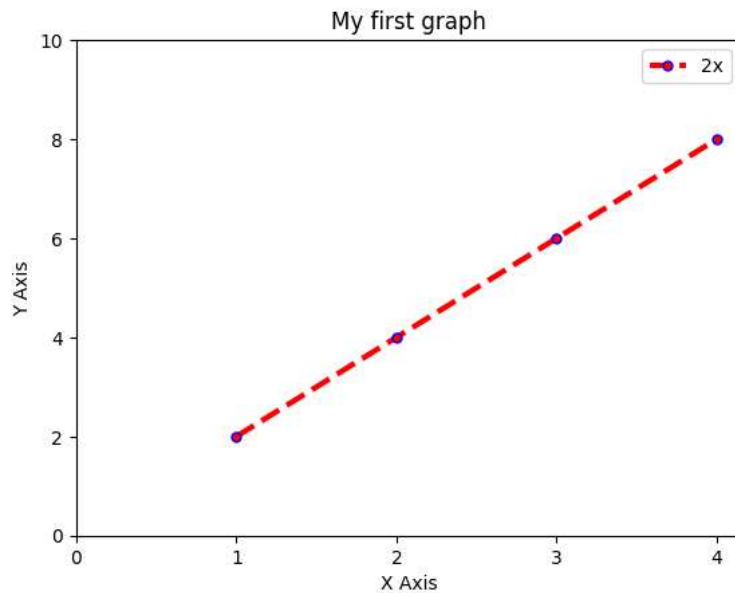
```
x=[1,2,3,4]
y=[2,4,6,8]

#Keyword Argument Notations
plt.plot(x,y,label='2x', color='red', linewidth=3, marker='.',linestyle='--',markersize=10, markededgecolor='blue')

plt.title('My Line graph')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')

#X, Y axis tickmarks (Scale of graph)
plt.xticks([0,1,2,3,4])
plt.yticks([0,2,4,6,8,10])

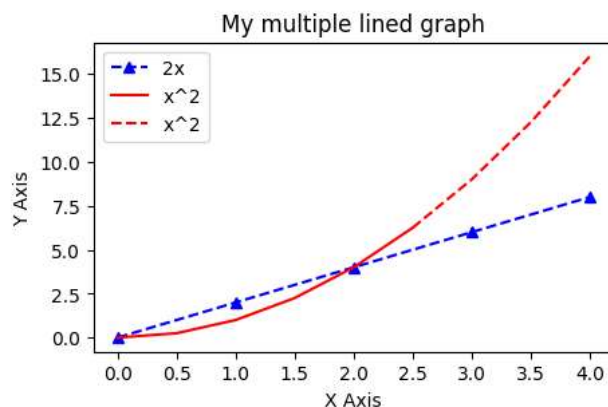
#Add legend
plt.legend()
plt.show()
```



▼ Plot multiple lines

- Resize graph, using `.figure(figsize, dpi)`
- Use Shorthand notation
- Plot multiple lines
- Save a graph, using `.savefig('graph_name.png', dpi)`

```
#Resize Graph
plt.figure(figsize=(5,3), dpi = 100)
#line 1
x=[0,1,2,3,4]
y=[0,2,4,6,8]
#Shorthand notation
#fmt=[color][marker][line]
plt.plot(x,y,'b^--',label='2x')
#line 2
x2=np.arange(0,4.5,0.5)
#Plot part of the graph as line
plt.plot(x2[5:],x2[5:]**2,'r', label='x^2')
#Plot Remaining line as dotted line
plt.plot(x2[5:],x2[5:]**2,'r--', label='x^2')
plt.title('My multiple lined graph')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.legend()
#dpi=300 is good choice for resolution
plt.savefig('my_graph.png',dpi=300)
plt.show()
```



▼ Barcharts

```
plt.figure(figsize=(5,3))
labels = ['A','B','C']
values = [1,4,2]
```

```
plt.title('My Barchart graph')
bars = plt.bar(labels, values)

#Styling the bars
bars[0].set_hatch('/')
bars[1].set_hatch('o')
bars[2].set_hatch('*')

plt.show()
```

