Matplotlib

1. What is Matplotlib?

- Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
- Matplotlib was created by John D. Hunter.
- Matplotlib is open source and we can use it freely.
- Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility
- It consists of several plots like line, bar, scatter, histogram, etc.

2. How to install Matplotlib?

Steps to install Matplotlib:

- If you have Python and PIP already installed on a system, then installation of Matplotlib is very easy.
- Install it using this command:

  C:\Users*Your Name*>pip install matplotlib

- Once the installation is completed, go to your IDE (e.g. jupyter notebook) and simply import it by typing: "import matplotlib"
-

3. Checking Matplotlib version

- The version string is stored under `__version__` attribute

```
import matplotlib

print(matplotlib.__version__)
```

4.Matplotlib Pyplot

- Most of the Matplotlib utilities lies under the `pyplot` submodule, and are usually imported under the `plt` alias:

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

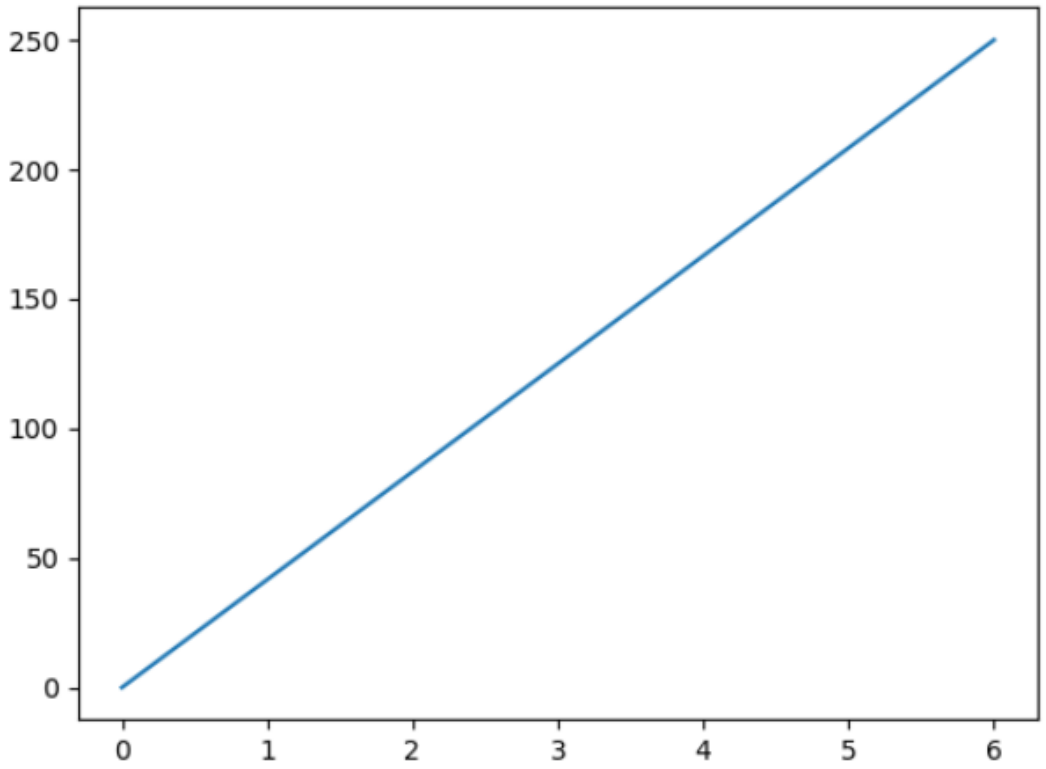
- Now the Pyplot package can be referred to as `plt`

Example1:

Draw a line in a diagram from position (0,0) to position (6,250):

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

xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```

Output:



5. Plotting x and y points

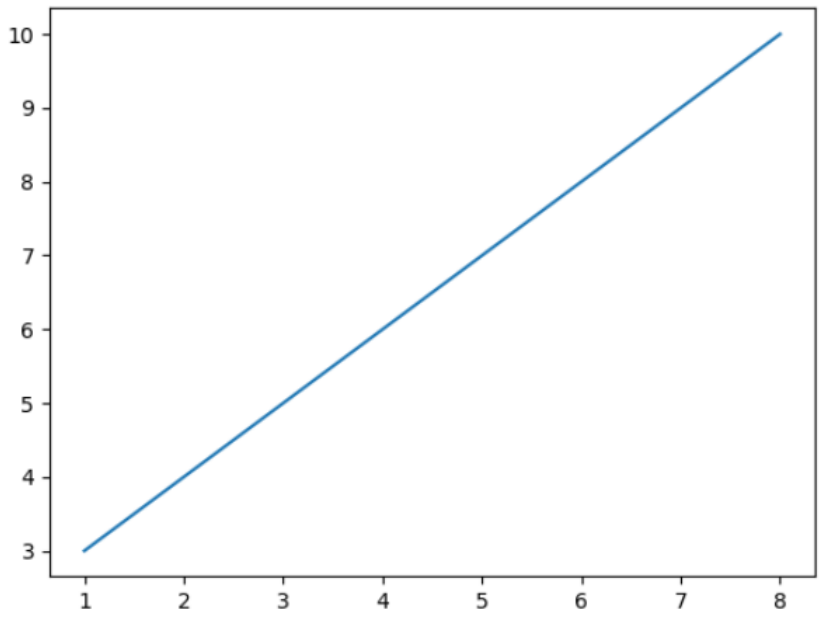
- The plot() function is used to draw points (markers) in a diagram.
- By default, the plot() function draws a line from point to point..
- The function takes parameters for specifying points in the diagram.
- Parameter 1 is an array containing the points on the **x-axis**.
- Parameter 2 is an array containing the points on the **y-axis**

Example2: Draw a line in a diagram from position (1, 3) to position (8, 10)

- If we need to plot a line from (1, 3) to (8, 10), we have to pass two arrays [1, 8] and [3, 10] to the plot function

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

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints)
plt.show()
```

Output:



:

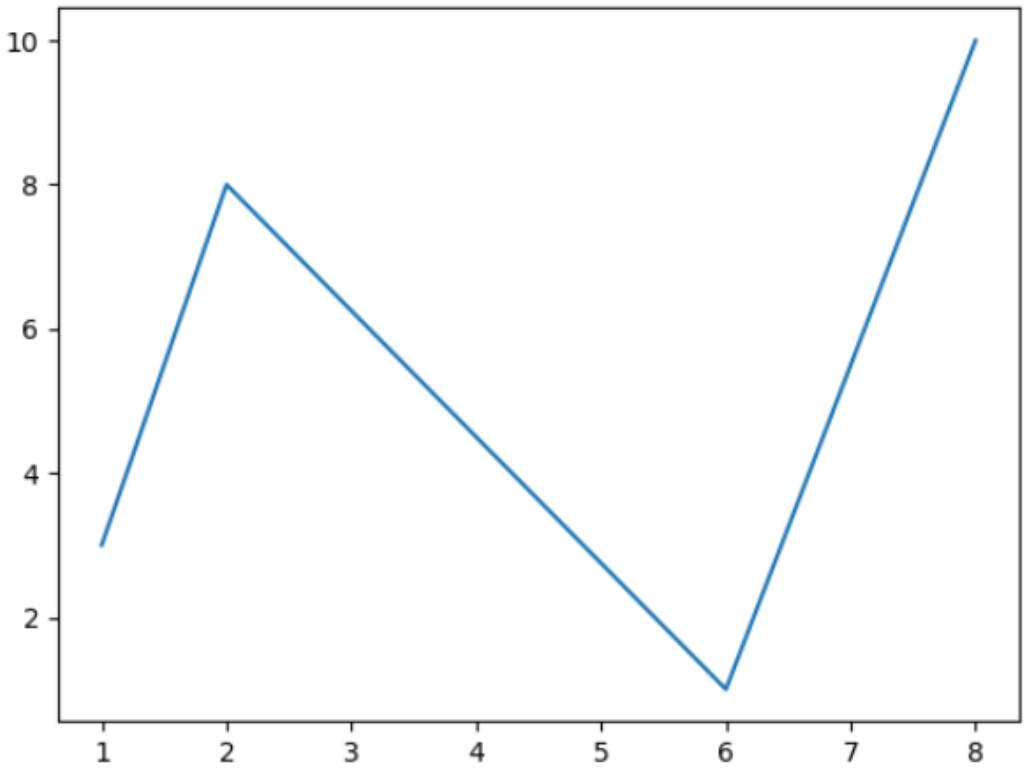
4.Plotting Multiple points

- You can plot as many points as you like, just make sure you have the same number of points in both axis.

Example3: Draw a line in a diagram from position (1, 3) to (2, 8) then to (6, 1) and finally to position (8, 10)

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

xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])

plt.plot(xpoints, ypoints)
plt.show()
```

Output:



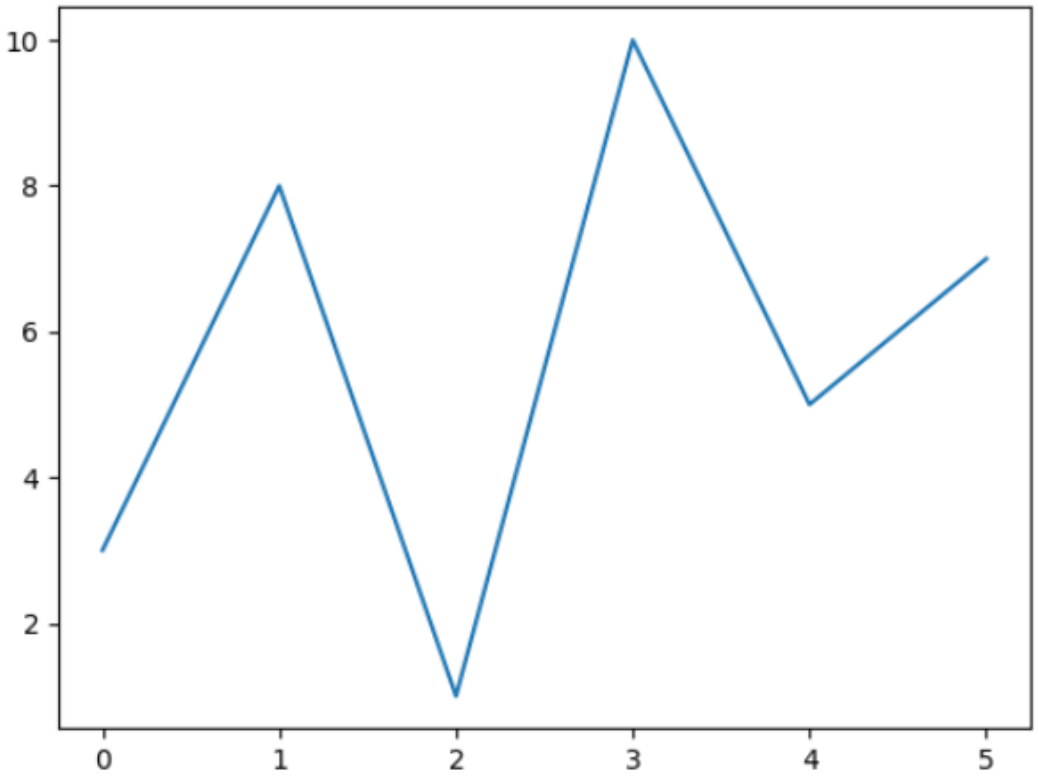
5.Default X points

- If we do not specify the points in the x-axis, they will get the default values 0, 1, 2, 3, (etc. depending on the length of the y-points.

Example 4: Plotting without x-points

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

ypoints = np.array([3, 8, 1, 10, 5, 7])

plt.plot(ypoints)
plt.show()
```

Output:



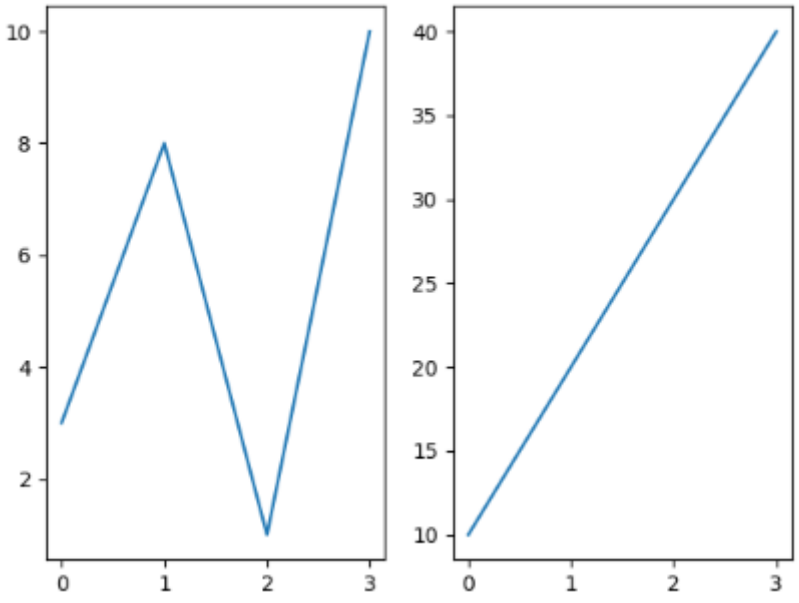
6. Matplotlib Subplot

- With the subplot() function you can draw multiple plots in one figure

Example 5: Draw 2 plots

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

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```

Output:



The subplot() Function:

- The subplot() function takes three arguments that describes the layout of the figure.
- The layout is organized in rows and columns, which are represented by the *first* and *second* argument.
- The third argument represents the index of the current plot.
- plt.subplot(1, 2, 1)
  #the figure has 1 row, 2 columns, and this plot is the *first* plot.

- plt.subplot(1, 2, 2)
  #the figure has 1 row, 2 columns, and this plot is the *second* plot

.

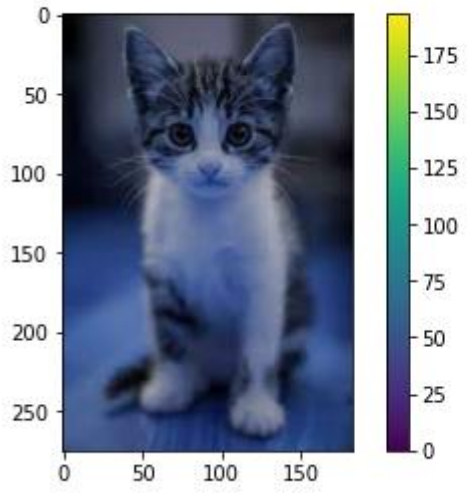
].

Let's try loading an image using matplotlib through cv2 and displaying it.

```
from matplotlib import pyplot as plt
import numpy as np
import cv2
img = cv2.imread('cat.png')
plt.imshow(img)
plt.colorbar()
```

Output:

```
<matplotlib.colorbar.Colorbar at 0x16f52165940>
```



- plt.colorbar() adds a color bar next to the plot. It's helpful to have an idea of what value a color represents

7. References:

- https://www.w3schools.com/python/matplotlib_intro.asp
- https://www.geeksforgeeks.org/matplotlib-tutorial/

8. Video Link:

- https://youtu.be/yZTBMMdPOww