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.

Installation of Matplotlib

Step1: pip install matplotlib

```
Command Prompt

C:\Users\Anusha>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.3.4-cp36-cp36m-win_amd64.whl (8.5 MB)
     |□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□| 8.5 MB 6.4 MB/s
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.3.1-cp36-cp36m-win_amd64.whl (51 kB)
     |□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□| 51 kB 4.1 MB/s
Collecting cycler>=0.10
  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)
Requirement already satisfied: numpy>=1.15 in c:\users\anusha\appdata\local\programs\python\python36\lib\site-packages
(from matplotlib) (1.19.5)
Collecting pillow>=6.2.0
  Downloading Pillow-8.4.0-cp36-cp36m-win_amd64.whl (3.2 MB)
     |□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□| 3.2 MB 6.8 MB/s
Requirement already satisfied: python-dateutil>=2.1 in c:\users\anusha\appdata\local\programs\python\python36\lib\site-
packages (from matplotlib) (2.8.2)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3
  Downloading pyparsing-3.0.6-py3-none-any.whl (97 kB)
     |□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□| 97 kB 6.8 MB/s
Requirement already satisfied: six>=1.5 in c:\users\anusha\appdata\local\programs\python\python36\lib\site-packages (fr
om python-dateutil>=2.1->matplotlib) (1.16.0)
Installing collected packages: pyparsing, pillow, kiwisolver, cycler, matplotlib
Successfully installed cycler-0.11.0 kiwisolver-1.3.1 matplotlib-3.3.4 pillow-8.4.0 pyparsing-3.0.6

C:\Users\Anusha>_
```

Step2: import 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
```

Matplotlib Plotting

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**.

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.

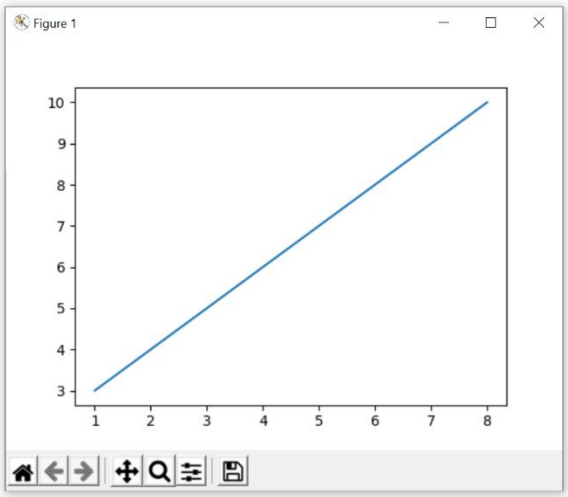
Example1:

```
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 SAMPLE



NOTE:

The **x-axis** is the horizontal axis.
The **y-axis** is the vertical axis.

Plotting Without Line: To plot only the markers, you can use *shortcut string notation* parameter 'o', which means 'rings'.

Example2:

```
import matplotlib.pyplot as plt

import numpy as np
```

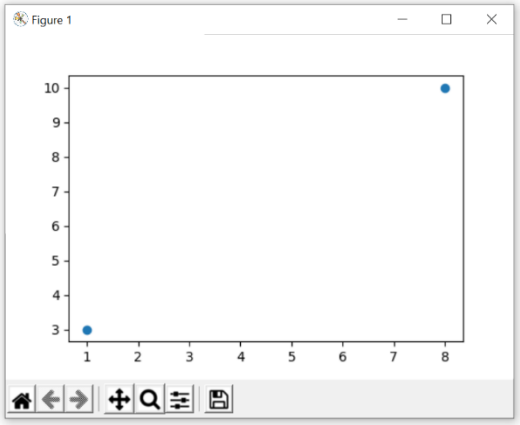
```
xpoints = np.array([1, 8])

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


plt.plot(xpoints, ypoints, 'o')

plt.show()
```

OUTPUT SAMPLE



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()
```

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.

Example4: Note- The **x-points** in the example below are [0, 1, 2, 3, 4, 5].

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

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

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

Matplotlib Markers-

You can use the keyword argument marker to emphasize each point with a specified marker:

Example1: Markers

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o')        #circle
#plt.plot(ypoints, marker = '*')        #star
#plt.plot(ypoints, marker = '+')        #plus + or 'P'
#plt.plot(ypoints, marker = 'x')        # x OR X
#plt.plot(ypoints, marker = 'p')        # pentagon
#plt.plot(ypoints, marker = 'D')        #Diamond

# use the shortcut string notation parameter to specify the marker.   TRY IT!
#       marker|line|color
        #plt.plot(ypoints, marker = '*')
        #plt.plot(ypoints, '*:g')
plt.show()
```

Matplotlib Line

Example1: Line Reference

```
import matplotlib.pyplot as plt

import numpy as np

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

#Example: Line Reference

plt.plot(ypoints, 'o-')       #Line

#plt.plot(ypoints, 'o:')      #Dot

#plt.plot(ypoints, 'o--')      #Dash

#plt.plot(ypoints, 'o-.')      #dash-dot

plt.show()
```

Example2:

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

ypoints = np.array([3, 8, 1, 10])
#linestyle, or ls
plt.plot(ypoints, linestyle = 'dotted')
#plt.plot(ypoints, linestyle = 'dashed')
#plt.plot(ypoints, ls = ':')    #shorthand notation

#Line-Color or c
#plt.plot(ypoints, color = 'r')
#plt.plot(ypoints, c = 'cyan')

#Line-width or lw
#plt.plot(ypoints, linewidth = '20.5')
plt.show()
```