Matplotlib

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

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
```

Import Matplotlib

```
import matplotlib
```

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**.

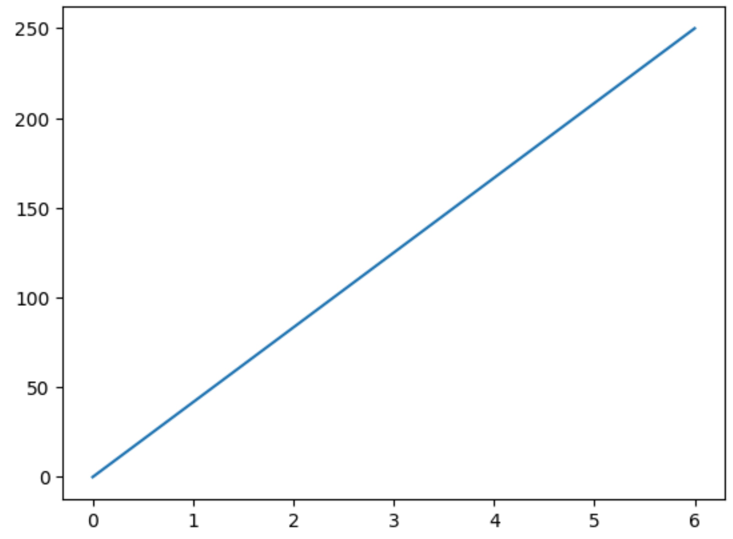
Example 1.

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()
```



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.

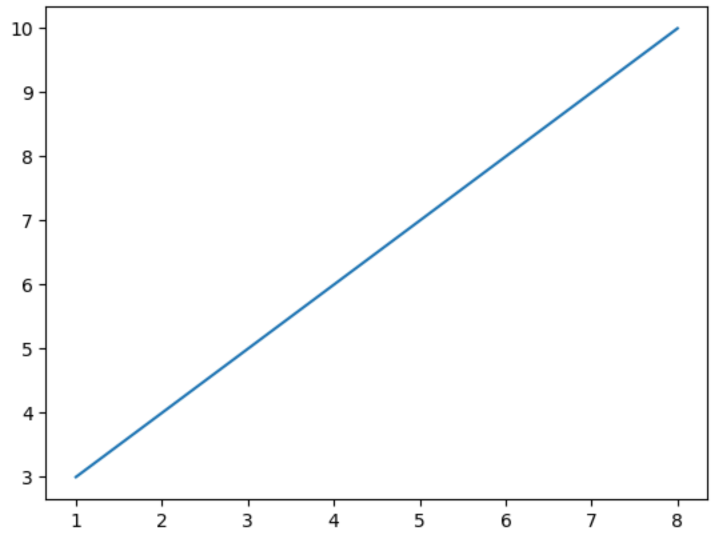
Example 2.

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

```
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()
```



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`.

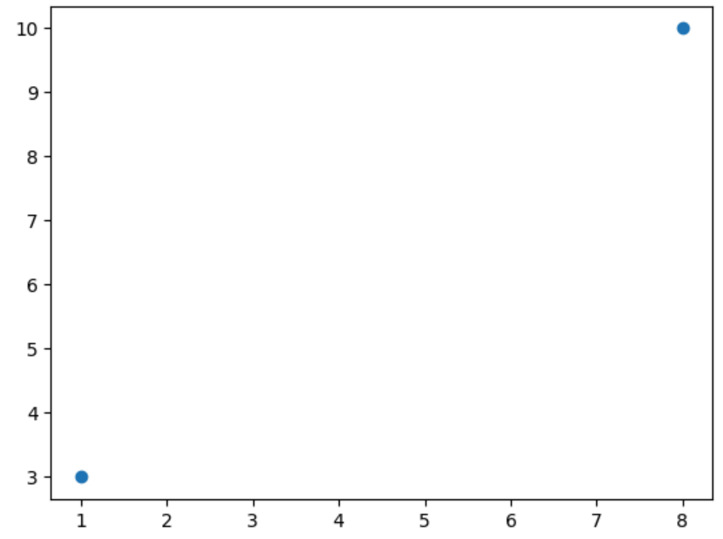
Example 3.

Draw two points in the diagram, one at position (1, 3) and one in position (8, 10):

```
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()
```



Multiple Points

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

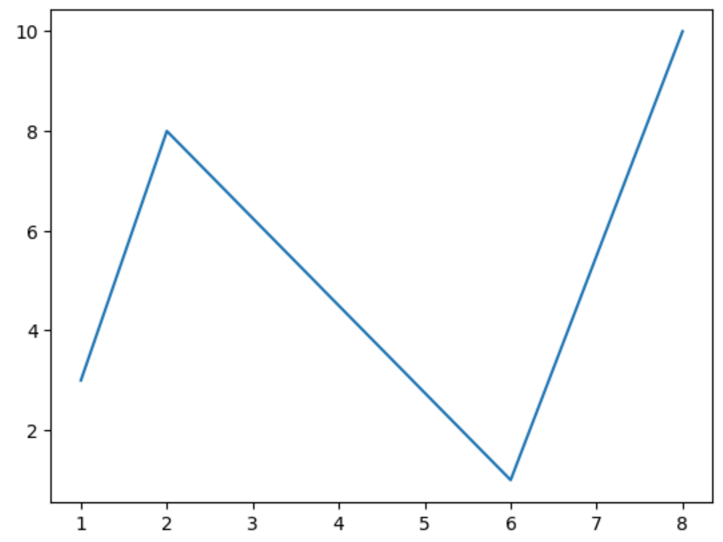
Example 4.

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.

So, if we take the same example as above, and leave out the x-points, the diagram will look like this:

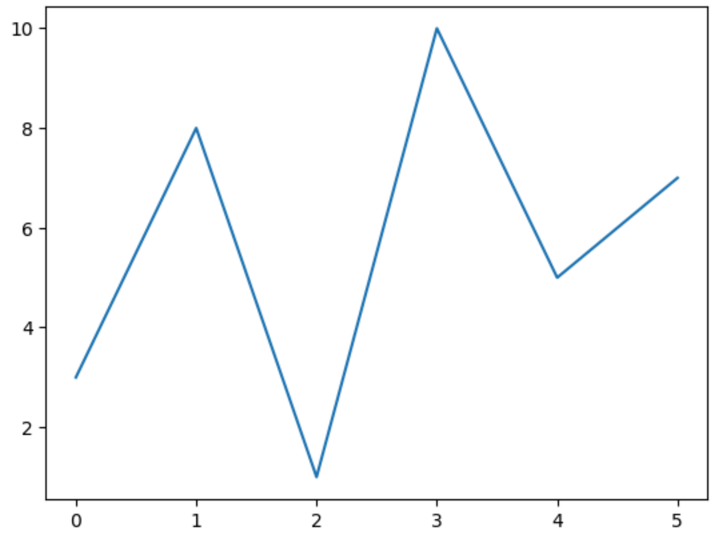
Example 5.

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()
```



Matplotlib Markers

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

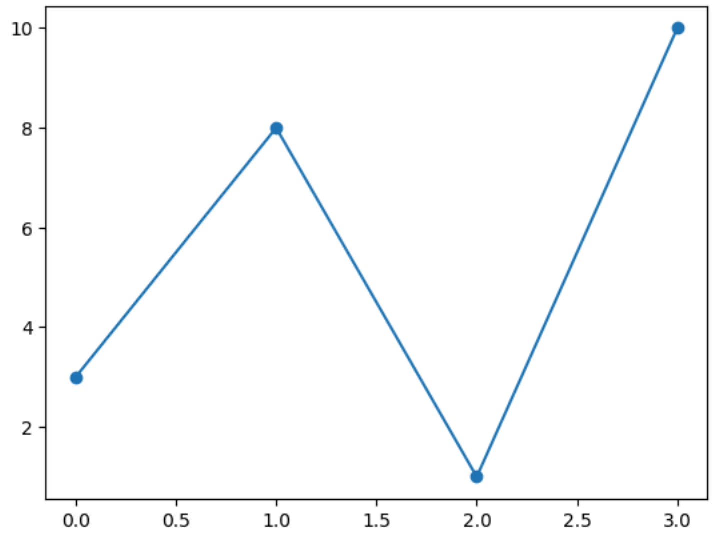
Example 6.

Mark each point with a circle:

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

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

plt.plot(ypoints, marker = 'o')
plt.show()
```



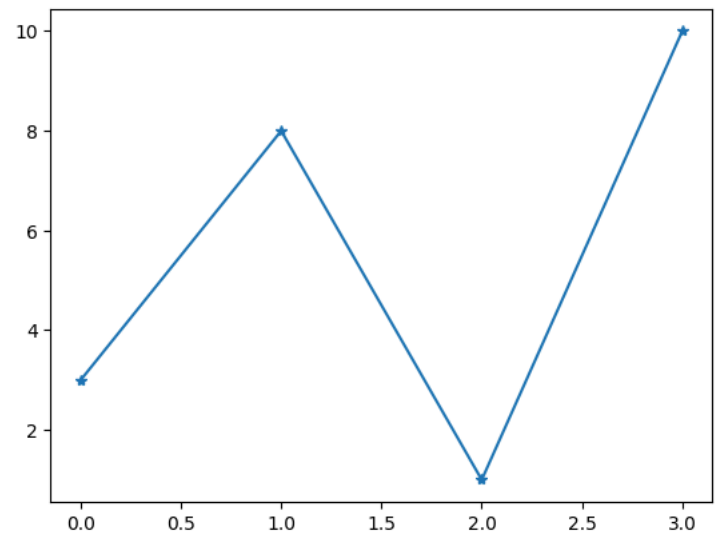
Example 7.

Mark each point with a star:

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

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

plt.plot(ypoints, marker = '*')
plt.show()
```



Marker Reference

You can choose any of these markers:

Marker	Description	Marker	Description	Marker	Description
o	Circle	*	Star	.	Point
,	Pixel	x	X	X	X (filled)
+	Plus	P	Plus (filled)	s	Square
D	Diamond	d	Diamond (thin)	p	Pentagon
H	Hexagon	h	Hexagon	v	Triangle Down
^	Triangle Up	<	Triangle Left	>	Triangle Right
1	Tri Down	2	Tri Up	3	Tri Left
4	Tri Right	\	Vline	_	Hline

Format Strings **fmt**

You can use also use the shortcut string notation parameter to specify the marker.

This parameter is also called fmt, and is written with this syntax:

marker | *line* | *color*

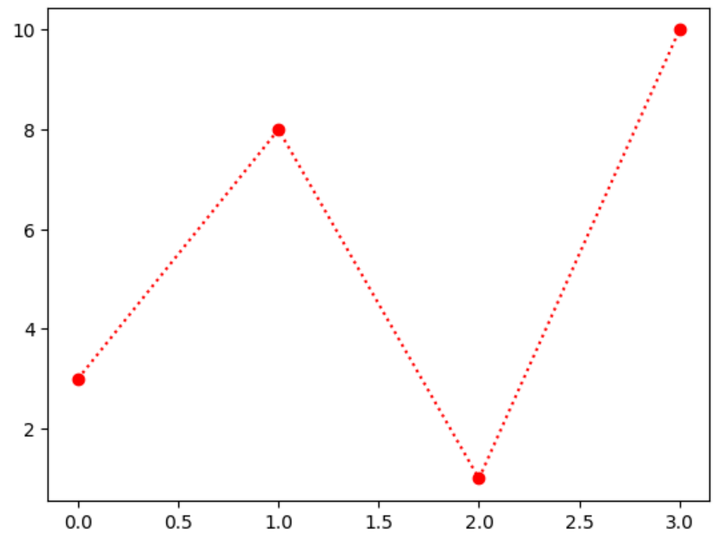
Example 8.

Mark each point with a circle:

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

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

plt.plot(ypoints, 'o:r')
plt.show()
```



The marker value can be anything from the Marker Reference above.

The line value can be one of the following:

Line Reference

Line Syntax	Description
-	Solid line
:	Dotted line
--	Dashed line
-.	Dashed/dotted line

Note: If you leave out the line value in the fmt parameter, no line will be plottet.

The short color value can be one of the following:

Color Reference

Color Syntax	Description	Color Syntax	Description
r	Red	g	Green
b	Blue	c	Cyan
m	Magenta	y	Yellow
k	Black	w	White

Marker Size

You can use the keyword argument **markersize** or the shorter version, `ms` to set the size of the markers:

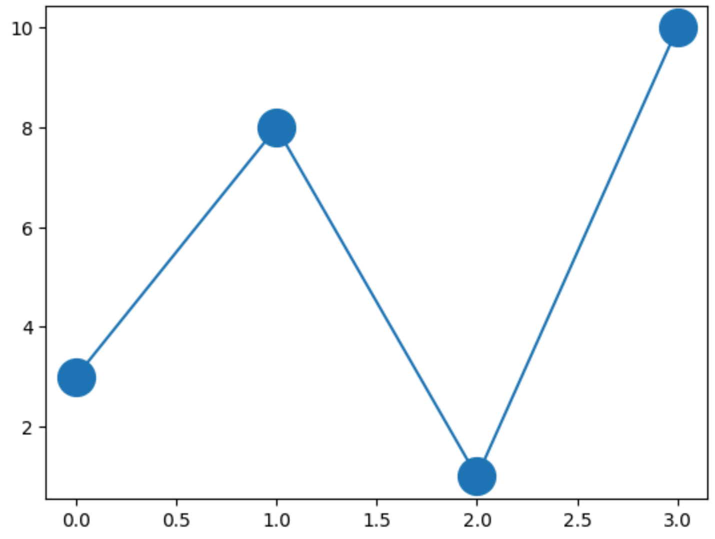
Example 9.

Set the size of the markers to 20:

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

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

plt.plot(ypoints, marker = 'o', ms = 20)
plt.show()
```



Marker Color

You can use the keyword argument `markeredgecolor` or the shorter `mec` to set the color of the edge of the markers:

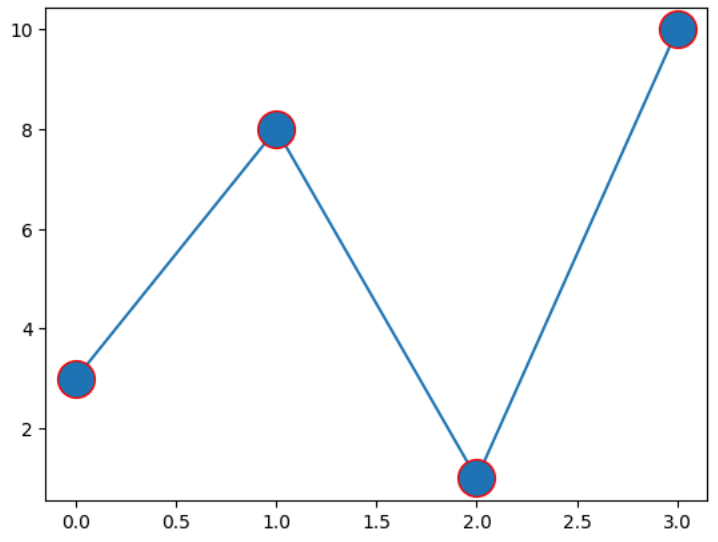
Example 10.

Set the EDGE color to red:

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

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

plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')
plt.show()
```



You can use the keyword argument `markerfacecolor` or the shorter `mfc` to set the color inside the edge of the markers:

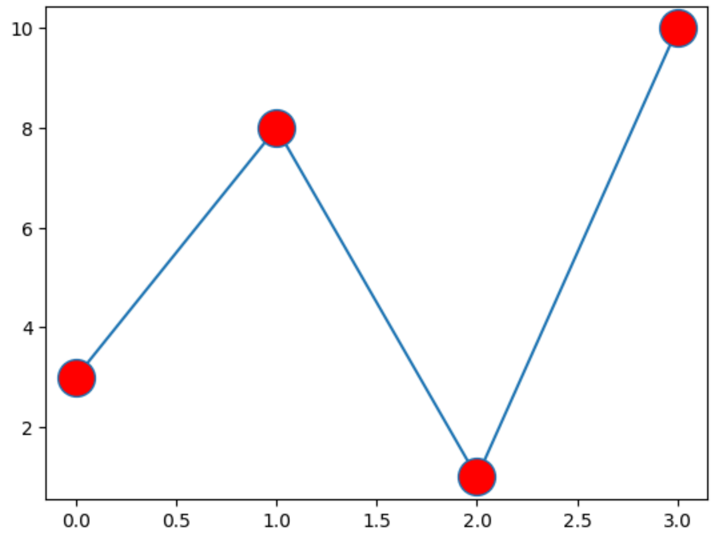
Example 11.

Set the FACE color to red:

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

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

plt.plot(ypoints, marker = 'o', ms = 20, mfc = 'r')
plt.show()
```



Use both the `mec` and `mfc` arguments to color of the entire marker:

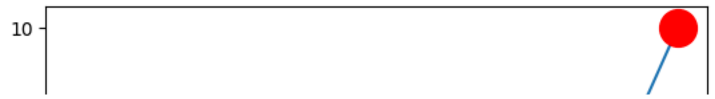
Example 12.

Set the color of both the edge and the face to red:

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

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

plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r', mfc = 'r')
plt.show()
```



You can also use Hexadecimal color values:

Example 13.

Mark each point with a beautiful green color:

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

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