
Matplotlib

Introduction

- Matplotlib is a low level graph plotting library in python that serves as a visualization utility.
 - Line, Scatter, Bar, Histogram, Pie Chart
- Installation
 - `pip install matplotlib`
- Import
 - `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`

Plot

- 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.

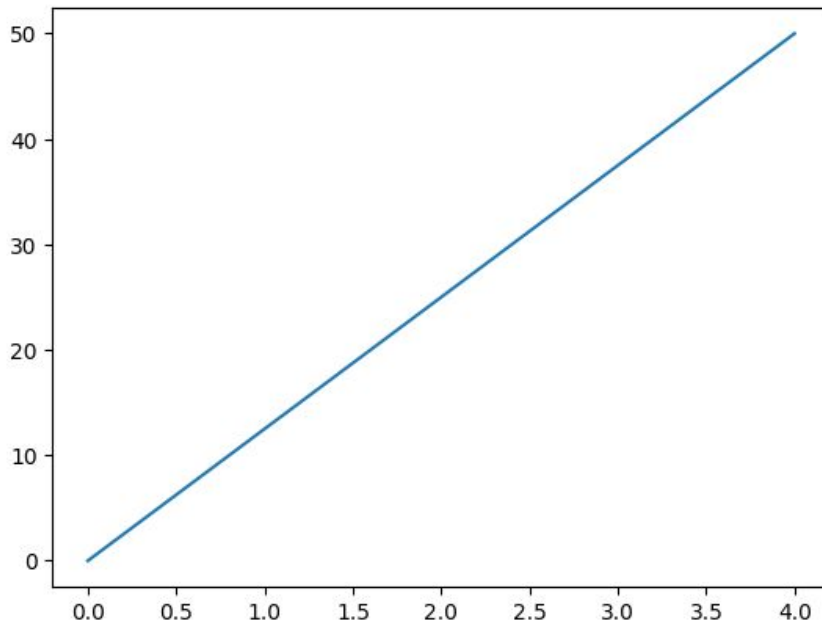
Example - Plotting x and y points - Line

Draw a line in a diagram from position (0,0) to position (4,50):

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

xpoints = np.array([0, 4])
ypoints = np.array([0, 50])

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

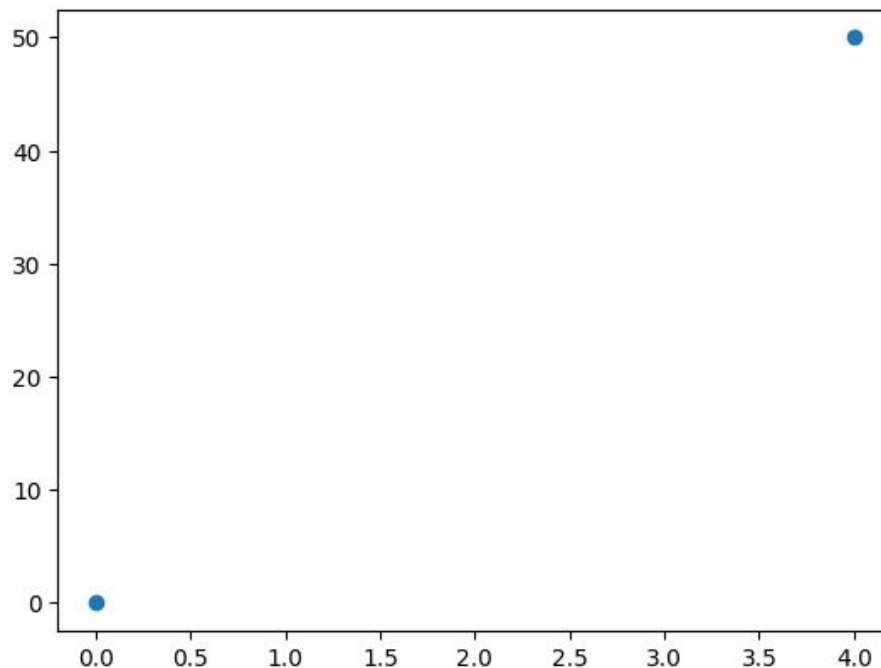


Example - Plotting x and y points - Without Line

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

xpoints = np.array([0, 4])
ypoints = np.array([0, 50])

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

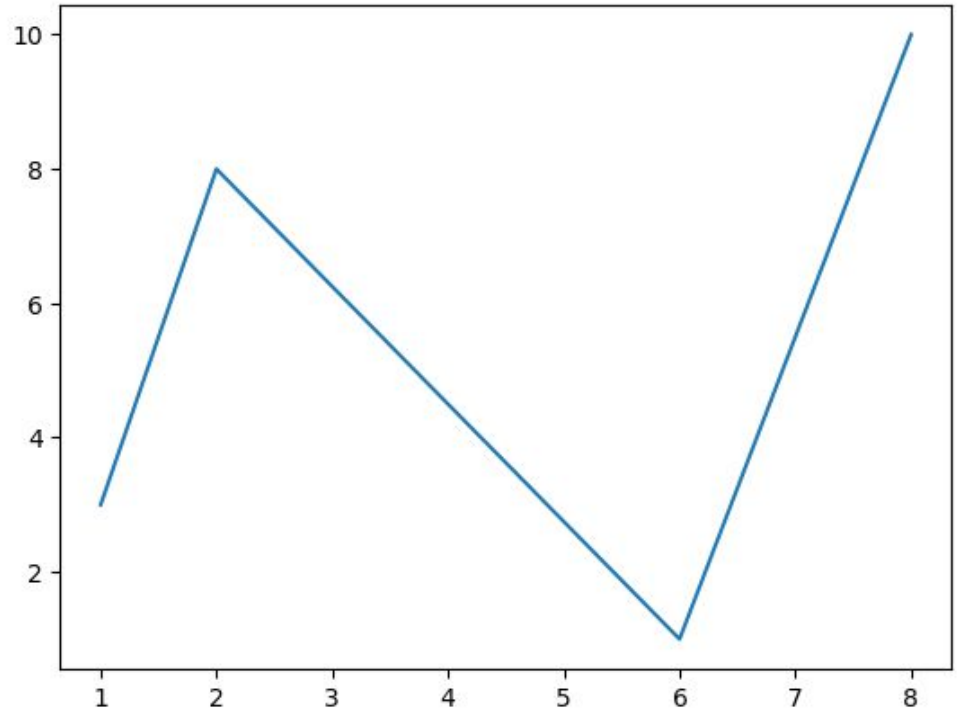


Example - Plotting Multiple Points

```
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()
```



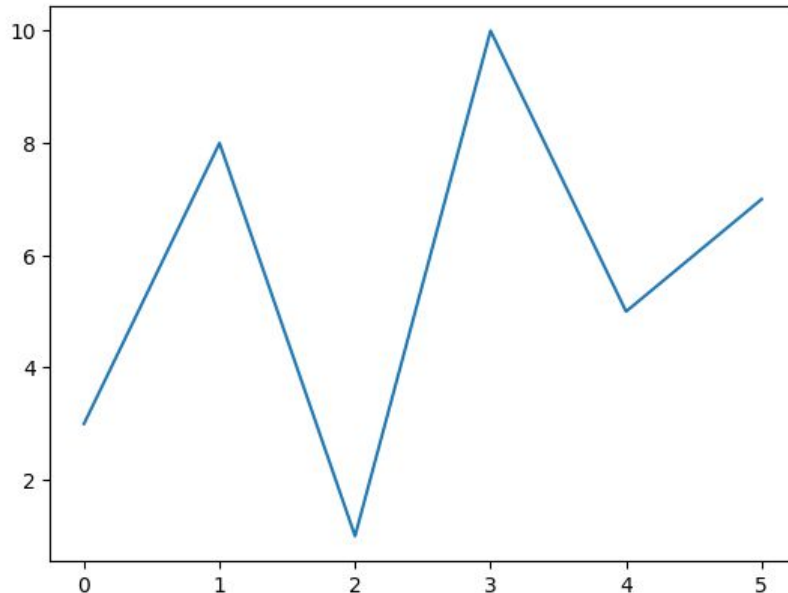
Example - Plotting with Default X-Points

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

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

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

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

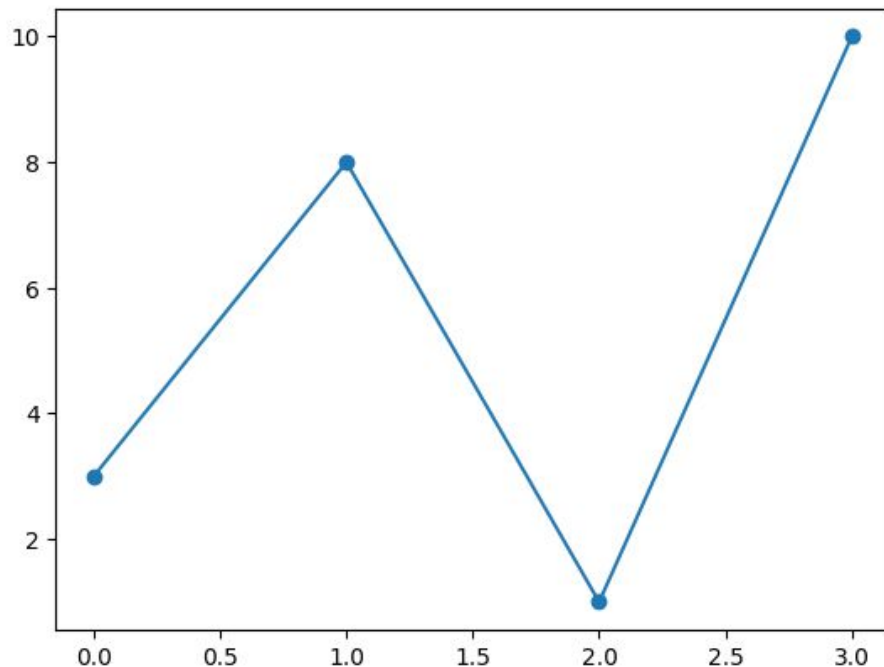


Example - Markers

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

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

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

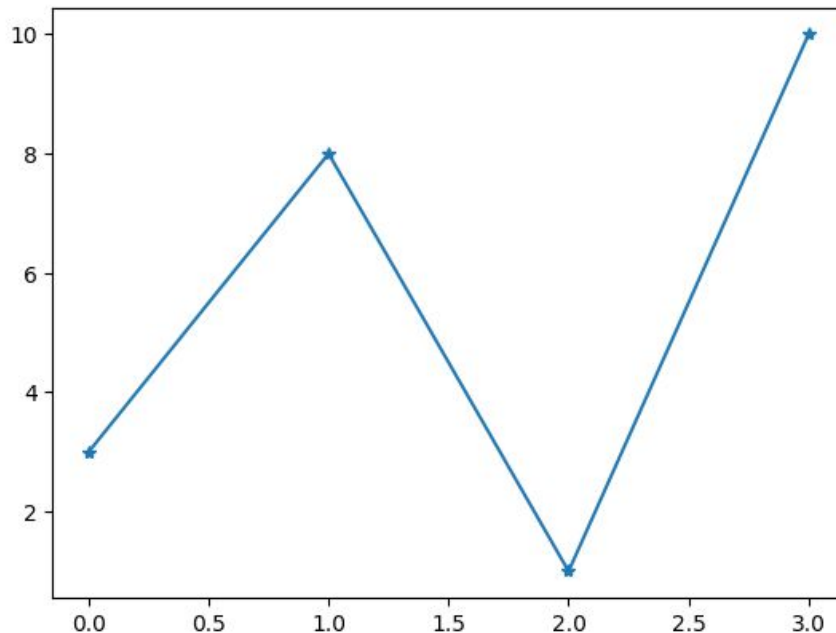


Example - Markers - Star

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

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

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



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

Marker	Description
'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

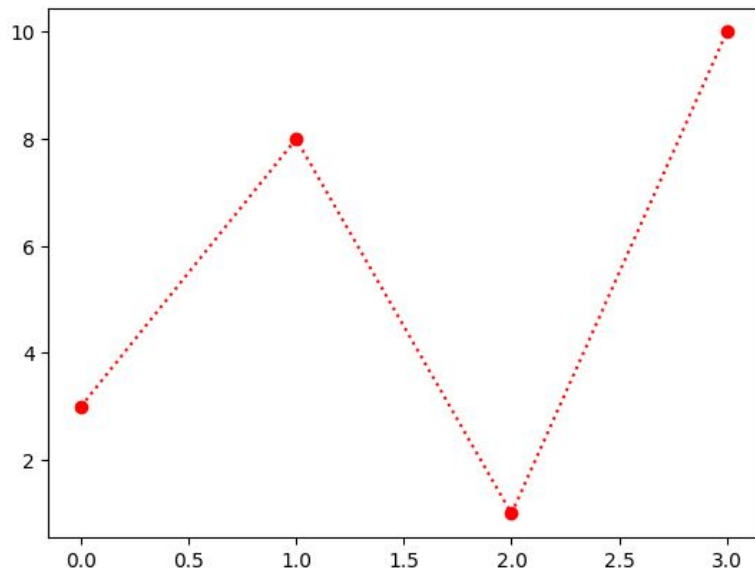
Example - Markers - Format Strings fmt

- You can 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

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

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

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



Line Reference

Line Syntax	Description
'_'	Solid line
'.'	Dotted line
'-'	Dashed line
'-.'	Dashed / dotted line

Color Reference

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

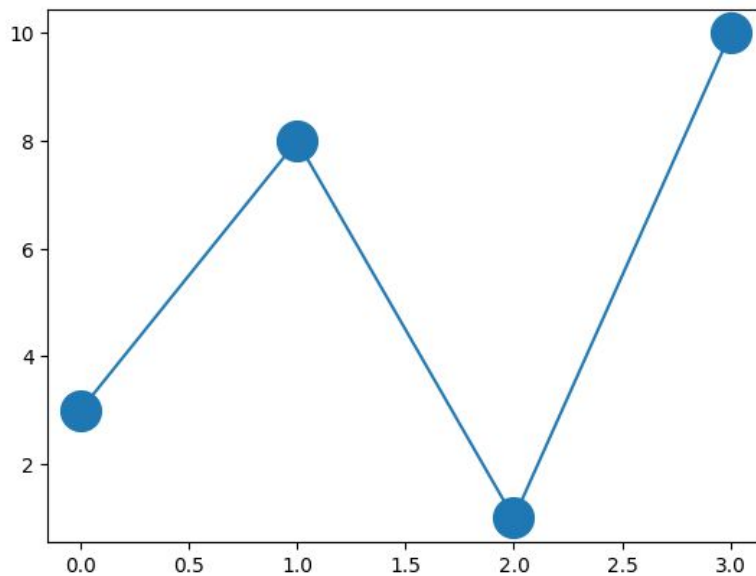
Example - Markers Size

- Use the keyword argument `markersize` or the shorter version, `ms` to set the size of the markers

```
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()
```



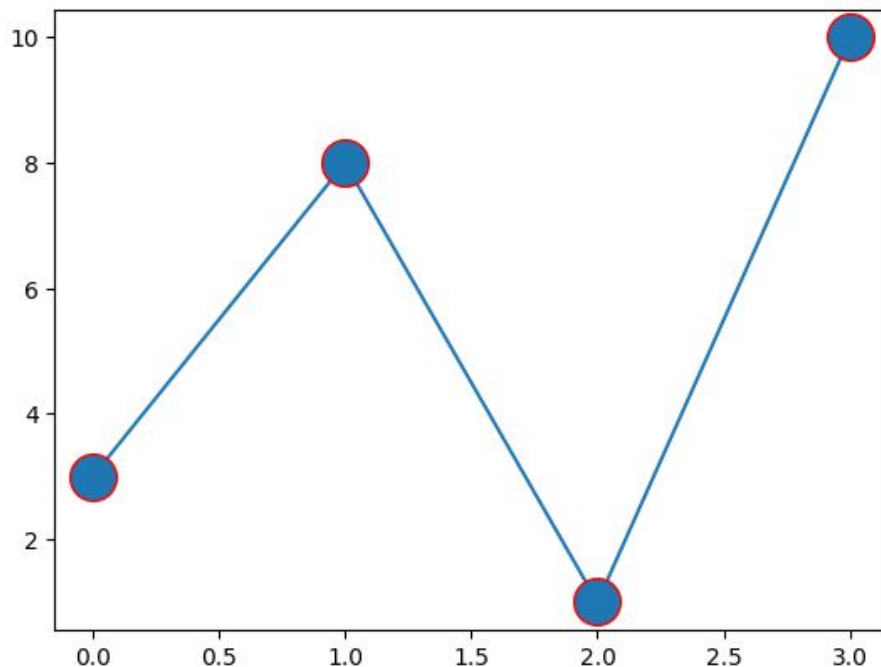
Example - Markers Color

- Use the keyword argument `markeredgcolor` or the shorter `mec` to set the color of the edge of the markers

```
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()
```



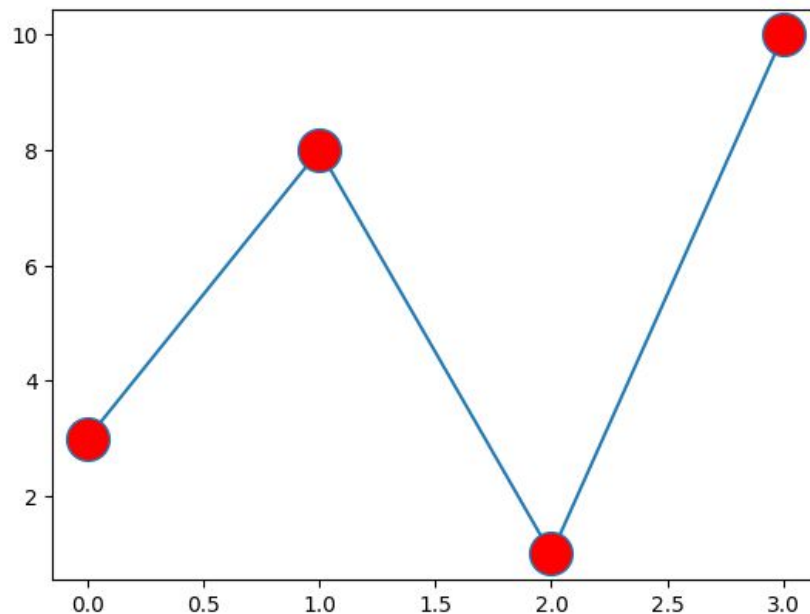
Example - Markers face Color

- Use the keyword argument `markerfacecolor` or the shorter `mfc` to set the color inside the edge of the markers:

```
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()
```

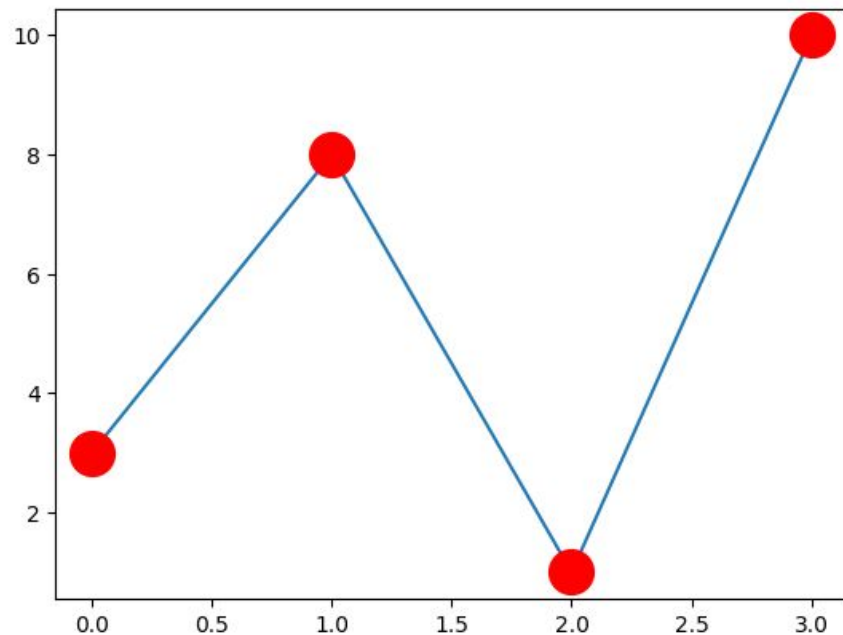


Example - Markers edge and face Color

```
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()
```



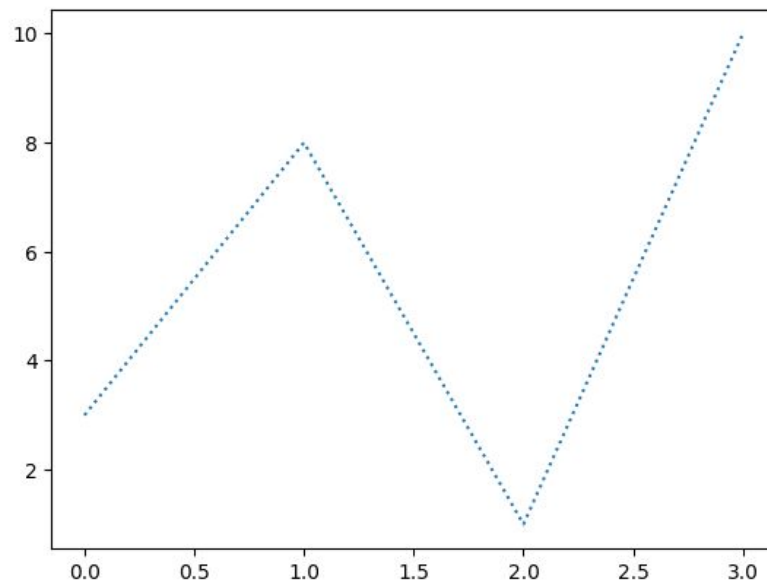
Example - Linestyle

Use the keyword argument `linestyle`, or shorter `ls`, to change the style of the plotted line

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

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

plt.plot(ypoints, linestyle = 'dotted')
plt.show()
```



Linestyle

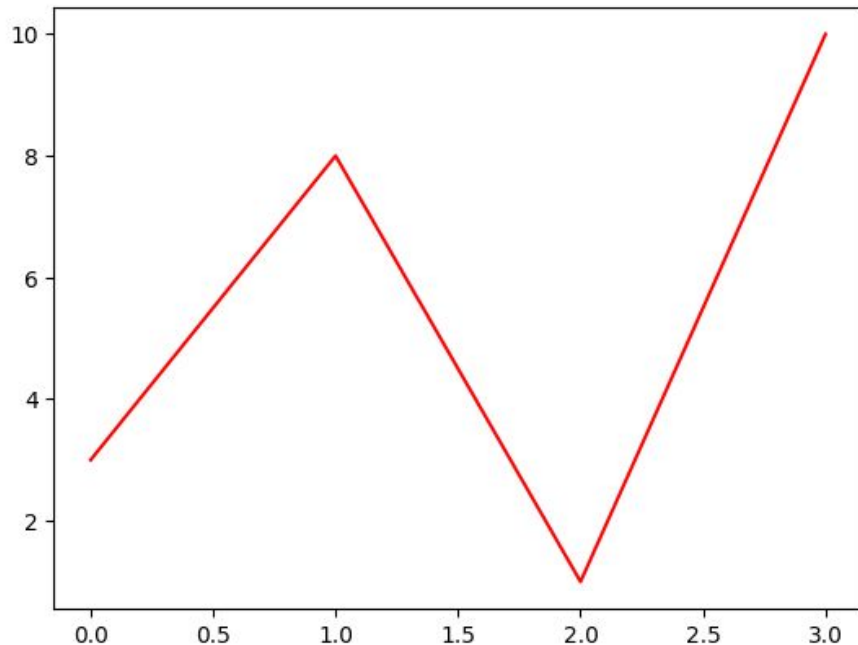
Style	Or
'solid' (default)	'-'
'dotted'	'.'
'dashed'	'--'
'dashdot'	'-.'
'None'	" or ''

Example - Line Color

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

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

plt.plot(ypoints, color = 'r')
plt.show()
```

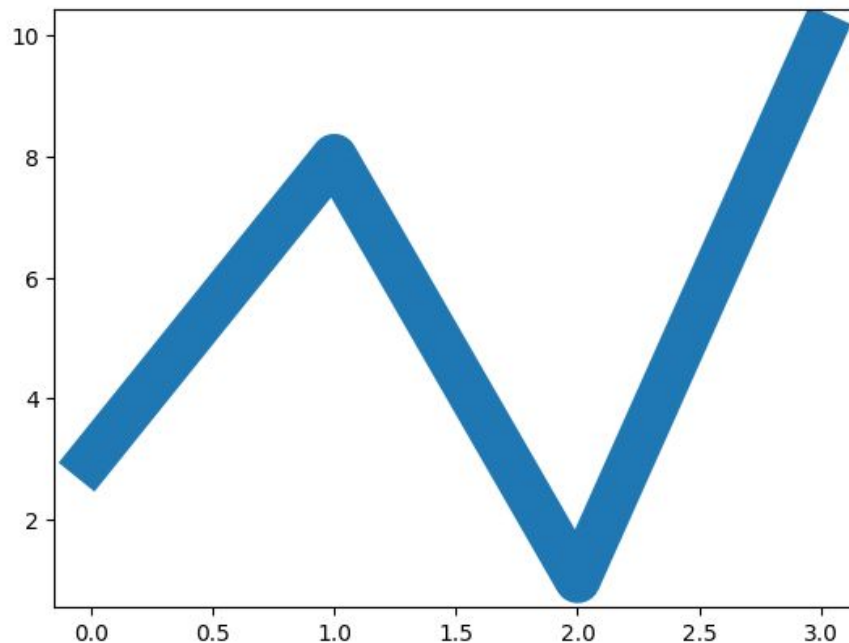


Example - Line Width

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

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

plt.plot(ypoints, linewidth = '20.5')
plt.show()
```



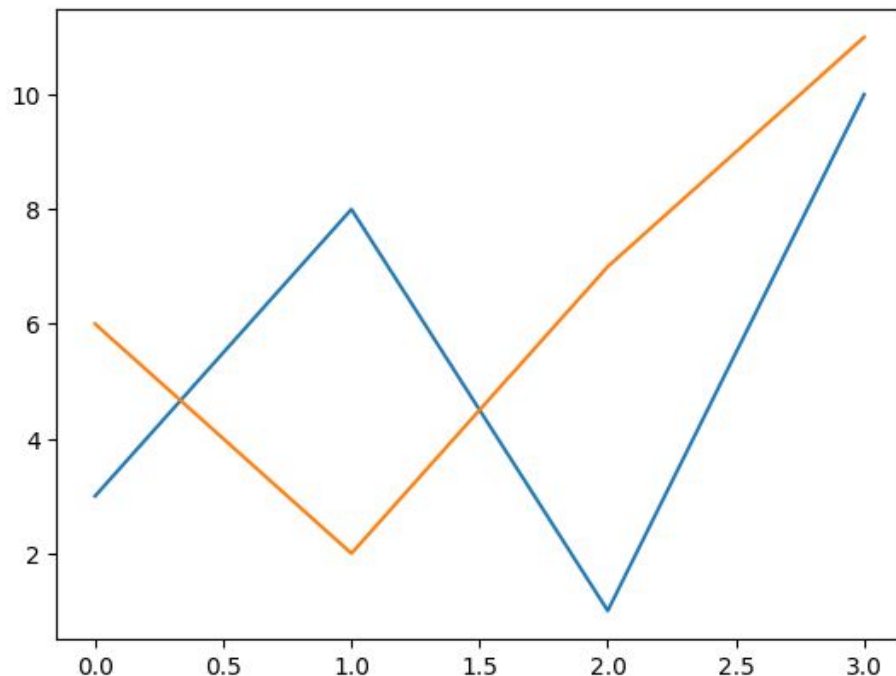
Example - Multiple Line

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

y1 = np.array([3, 8, 1, 10])
y2 = np.array([6, 2, 7, 11])

plt.plot(y1)
plt.plot(y2)

plt.show()
```



Example - Label

Use the `xlabel()` and `ylabel()` functions to set a label for the x- and y-axis

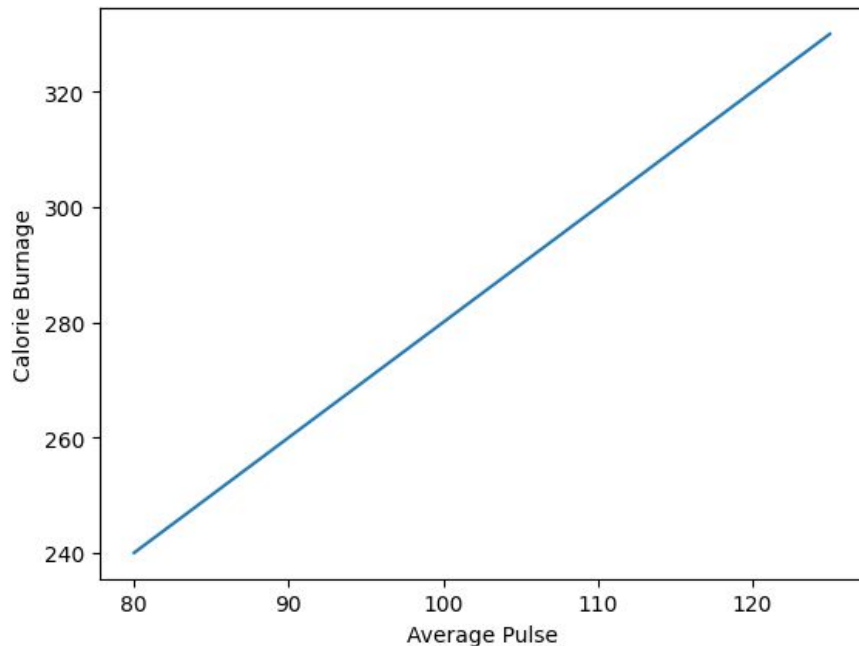
```
import numpy as np
import matplotlib.pyplot as plt

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y)

plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.show()
```



Example - Title

Use the title() function to set a title for the plot

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

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y)

plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.show()
```

