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**.

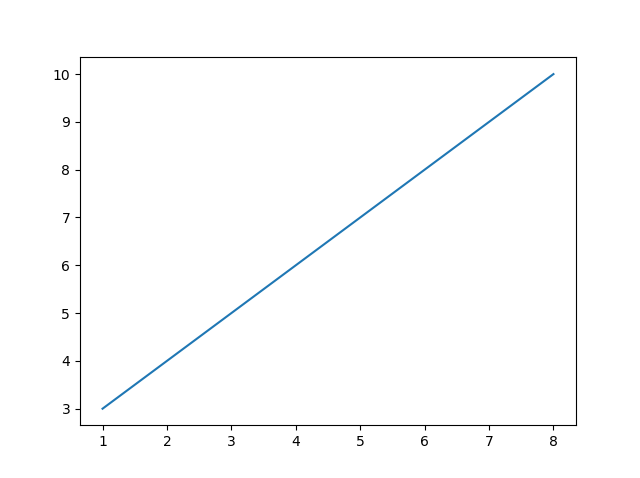
Parameter 2 is an array containing the points on the **y-axis**.

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**

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.

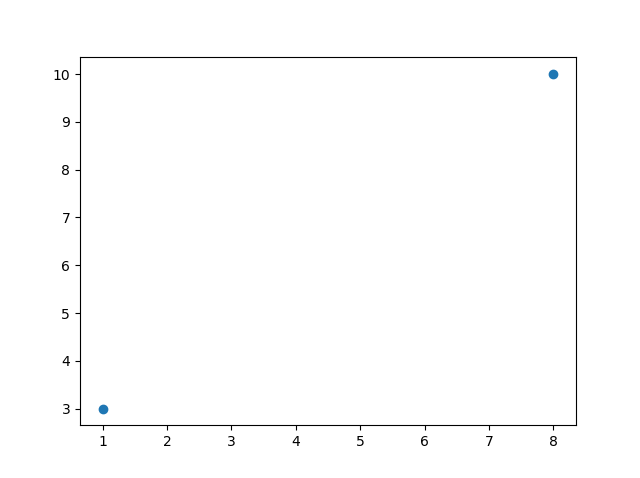
## **Plotting Without Line**

To plot only the markers, you can use shortcut string notation parameter 'o', which means 'rings'.

### **Example**

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()



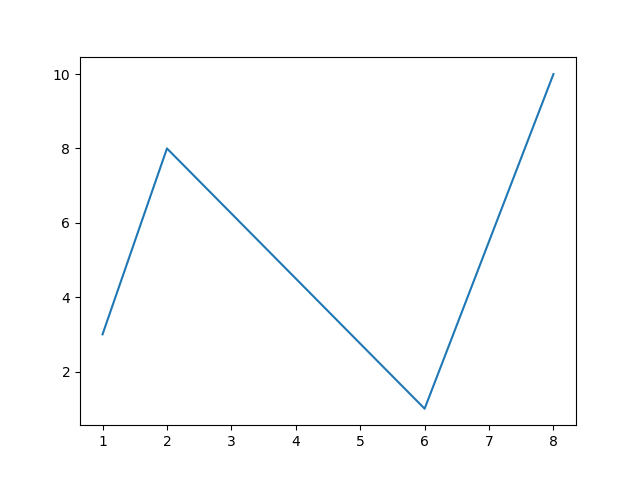
## **Multiple Points**

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

### **Example**

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**

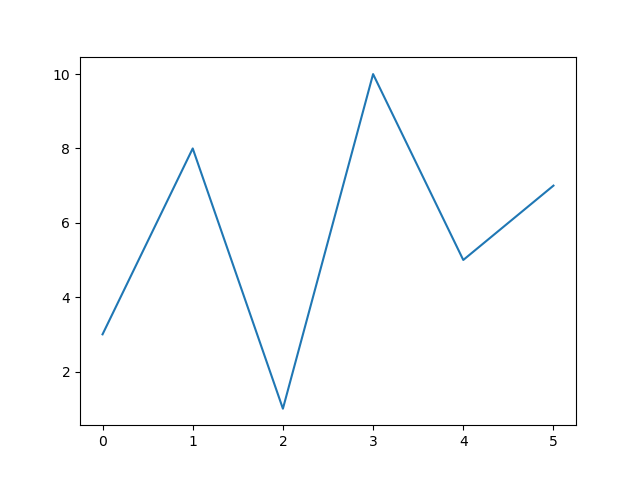
If we do not specify the points in the x-axis, they will get the default values 0, 1, 2, 3, (etc. depending on the length of the y-points.

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

### **Example**

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

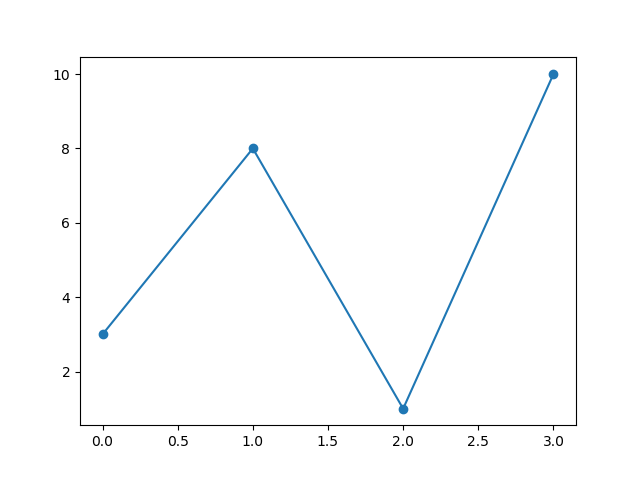
## **Markers**

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

### **Example**

Mark each point with a circle:

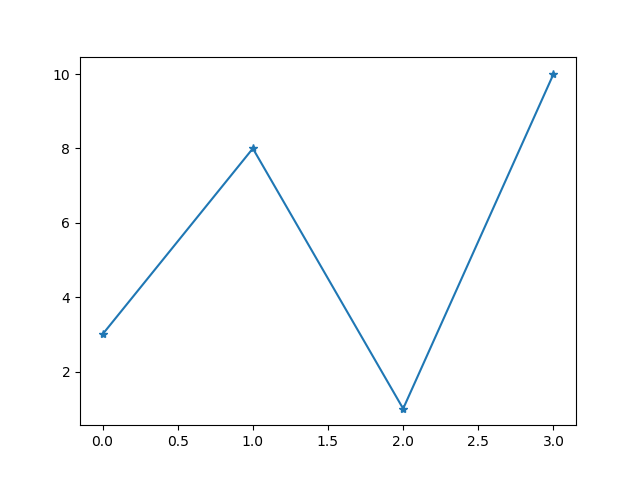
import matplotlib.pyplot as plt  
import numpy as np  
  
ypoints = np.array([3, 8, 1, 10])  
  
plt.plot(ypoints, marker = 'o')  
plt.show()



### **Example**

Mark each point with a star:

...  
plt.plot(ypoints, marker = '\*')  
...



## **Marker Reference**

You can choose any of these markers:

|  |  |
| --- | --- |
| **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.

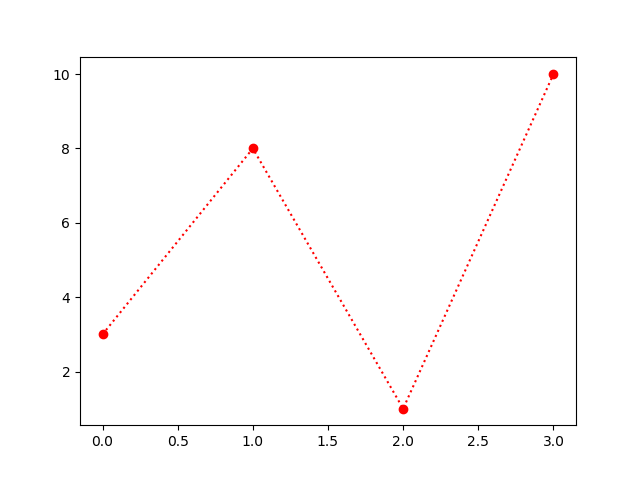
This parameter is also called fmt, and is written with this syntax:

marker|line|color

### **Example**

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()



## **Line Reference**

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

## **Color Reference**

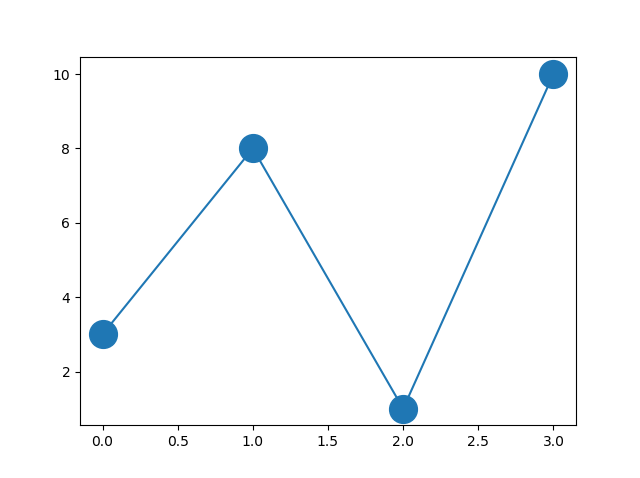
|  |  |
| --- | --- |
| **Color Syntax** | **Description** |
| 'r' | Red |
| 'g' | Green |
| 'b' | Blue |
| 'c' | Cyan |
| 'm' | Magenta |
| 'y' | Yellow |
| 'k' | Black |
| 'w' | White |

## **Marker Size**

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

### **Example**

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()

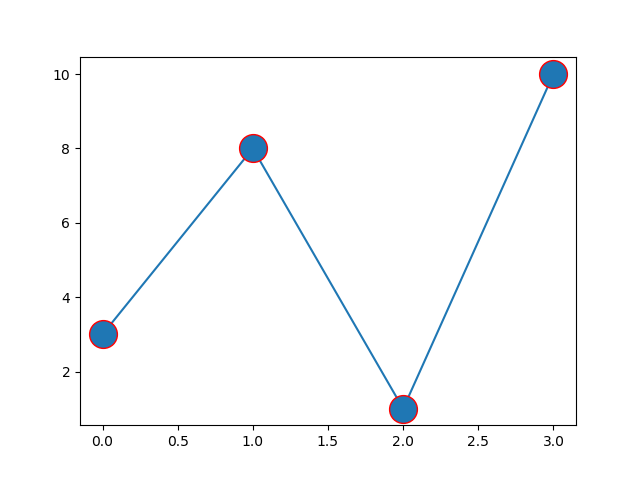
## **Marker Color**

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

### **Example**

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()



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

### **Example**

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()

