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:

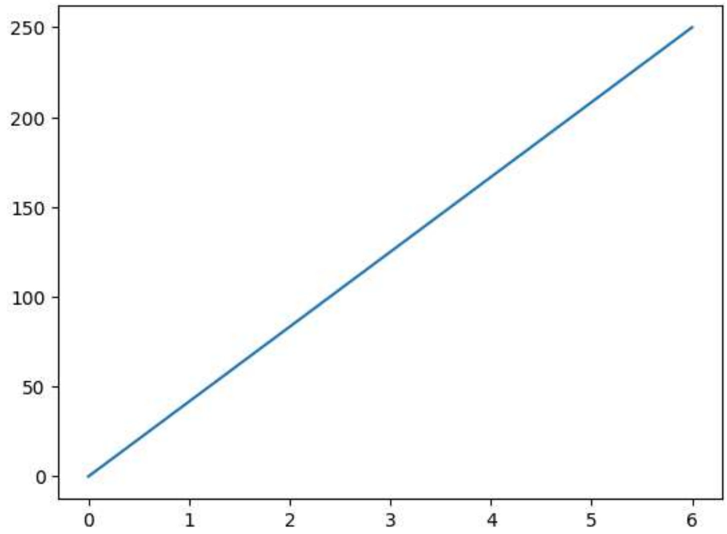
```
import matplotlib.pyplot as plt
```

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.

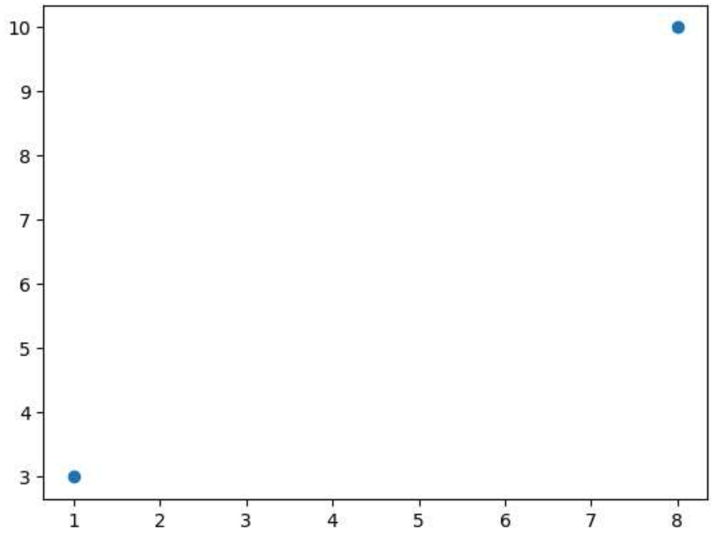
Plotting Without Line

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()
```



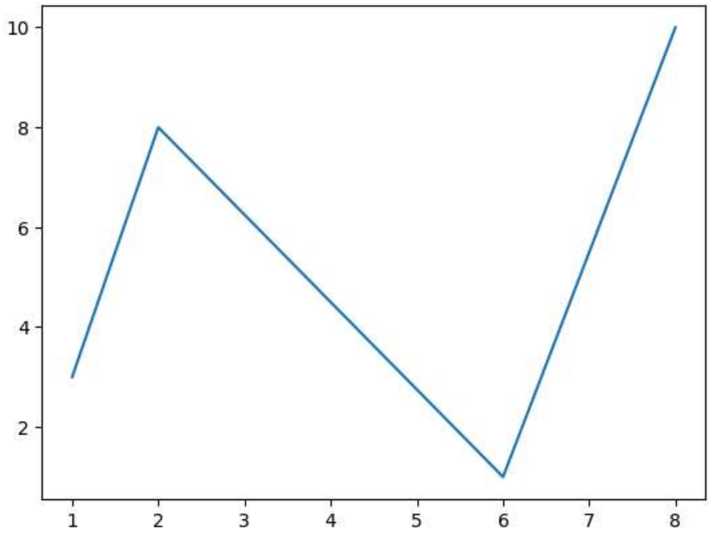
Multiple Points

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

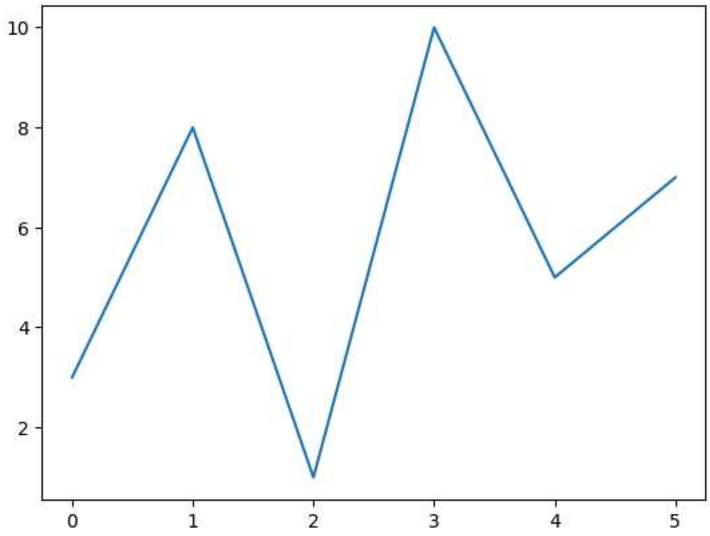
If we do not specify the points in the x-axis, they will get the default values 0, 1, 2, 3, (etc. depending on the length of the y-points.

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()
```



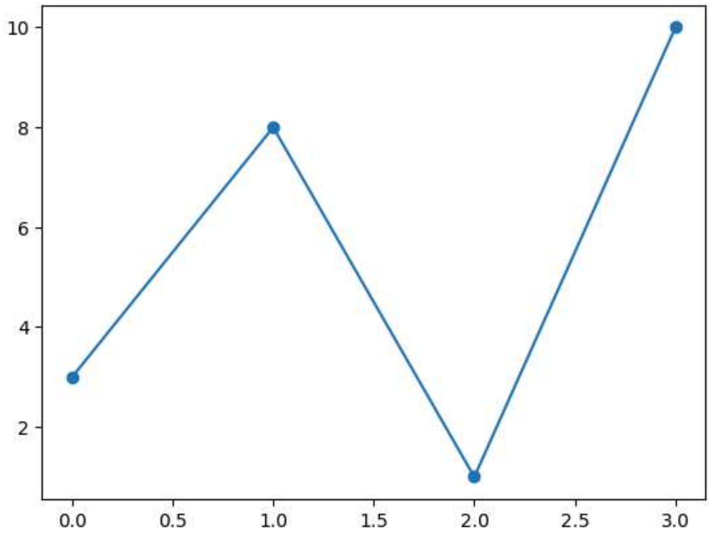
Matplotlib Markers

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

Example 6.

Mark each point with a circle:

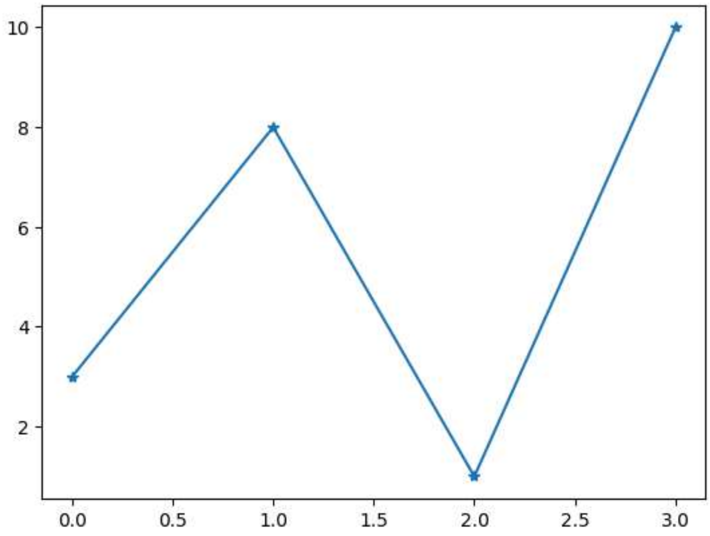
```
import matplotlib.pyplot as plt
import numpy as np

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

Marker	Description	Marker	Description	Marker	Description
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.

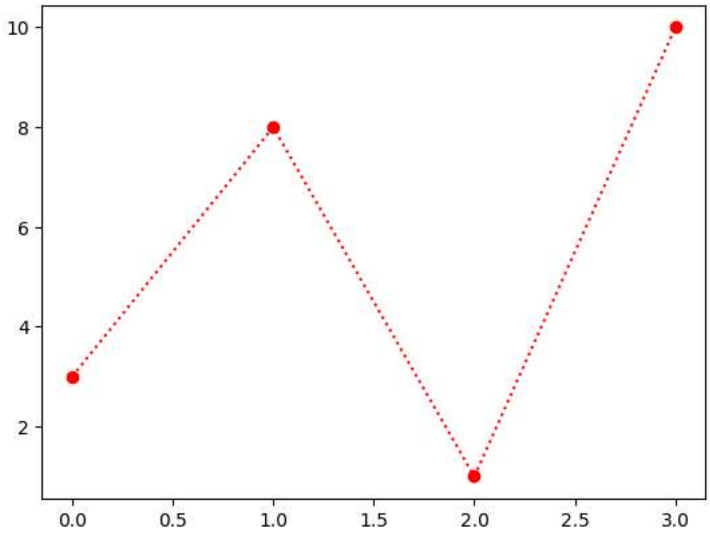
This parameter is also called fmt, and is written with this syntax:

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

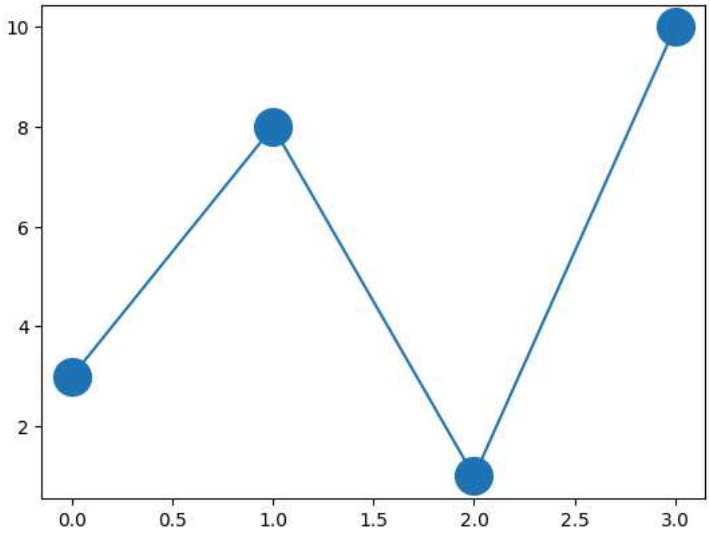
Marker Size

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:

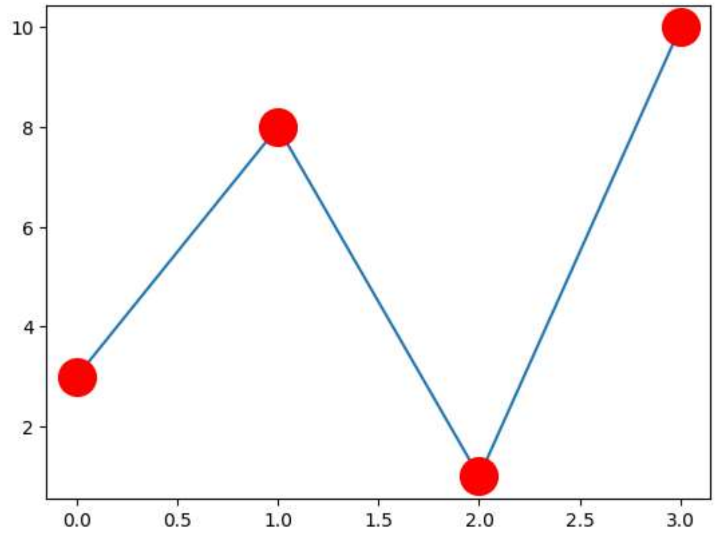
```
import matplotlib.pyplot as plt
import numpy as np

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])

plt.plot(ypoints, marker = 'o', ms = 20, mec = '#4CAF50', mfc = '#4CAF50')
plt.show()
```

Or any of the 140 supported color names.

Example 14.

Mark each point with the color named "hotpink":

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

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

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

Exercise:

Exercise 1: Create data frame for month and total profit, Read Total profit of all months and show it using a line plot Total profit data provided for each month.

Generated line plot must include the following properties: –

X label name = Month Number

Y label name = Total profit

Exercise 2: Get total profit of all months and show line plot with the following Style properties

Generated line plot must include following Style properties: –

- Line Style dotted and Line-color should be red
- Show legend at the lower right location.
- X label name = Month Number
- Y label name = Sold units number
- Add a circle marker.
- Line marker color as read
- Line width should be 3

BAR GRAPH

```
Start coding or generate with AI.
```

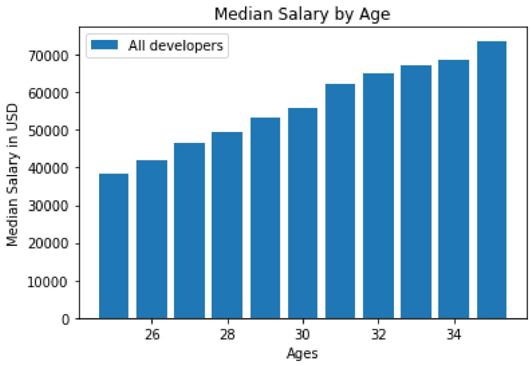
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]
```

```
devs_y = [38496, 42000, 46752, 49320, 53200, 56000, 62316, 64928, 67317, 68748, 73752]
```

1. Plotting the bar plot

```
plt.bar(x, devs_y, label="All developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.legend()
plt.show()
```

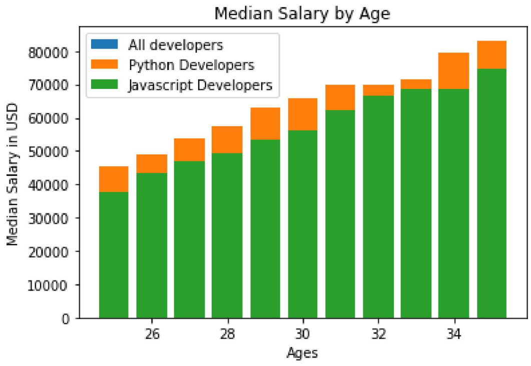


2. Adding more bars to the same plot

```
py_devs_y = [45372, 48876, 53850, 57287, 63016, 65998, 70003, 70000, 71418, 79674, 83238]
```

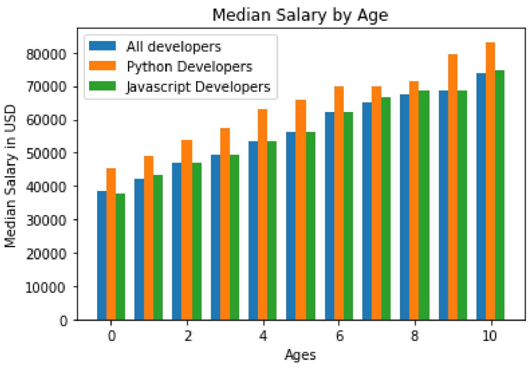
```
js_devs_y = [37810, 43515, 46823, 49293, 53437, 56373, 62375, 66674, 68745, 68746, 74583]
```

```
plt.bar(x, devs_y, label="All developers")
plt.bar(x, py_devs_y, label="Python Developers")
plt.bar(x, js_devs_y, label="Javascript Developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.legend()
plt.show()
```



3. Adjusting the width of the plot

```
import numpy as np

x_indexes = np.arange(len(x))


width = 0.25


plt.bar(x_indexes - width, devs_y, width=width, label="All developers")
plt.bar(x_indexes, py_devs_y, width = width, label="Python Developers")
plt.bar(x_indexes + width, js_devs_y, width=width, label="Javascript Developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.legend()
plt.show()
```



4. Changing the xlabels

```
plt.bar(x_indexes - width, devs_y, width=width, label="All developers")
plt.bar(x_indexes, py_devs_y, width = width, label="Python Developers")
plt.bar(x_indexes + width, js_devs_y, width=width, label="Javascript Developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.xticks(ticks=x_indexes, labels=x) #changing the xlabel
plt.legend()
plt.show()
```

5. Plotting the bar plot from pandas dataframe

```
import pandas as pd

from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
data = pd.read_csv('/content/drive/My Drive/data/data.csv')
```

```
data.head()
```

	Responder_id	LanguagesWorkedWith
0	1	HTML/CSS;Java;JavaScript;Python
1	2	C++;HTML/CSS;Python
2	3	HTML/CSS
3	4	C;C++;C#;Python;SQL
4	5	C++;HTML/CSS;Java;JavaScript;Python;SQL;VBA

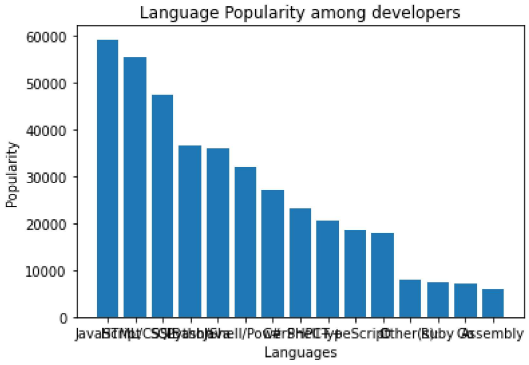
```
from collections import Counter
```

```
ids = data['Responder_id']
language_responses = data['LanguagesWorkedWith']
```

```
language_counter = Counter()


for response in language_responses:
    language_counter.update(response.split(";"))


languages = []
popularity = []

for item in language_counter.most_common(15):
    languages.append(item[0])
    popularity.append(item[1])


print(languages)
print(popularity)
```

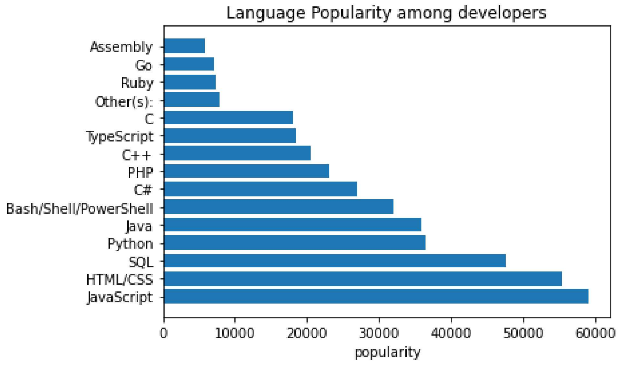
```
['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java', 'Bash/Shell/PowerShell', 'C#', 'PHP', 'C++', 'TypeScript', 'C', 'Other(s):', 'Ru
[59219, 55466, 47544, 36443, 35917, 31991, 27097, 23030, 20524, 18523, 18017, 7920, 7331, 7201, 5833]
```

```
plt.bar(languages, popularity)
plt.xlabel("Languages")
plt.ylabel("Popularity")
plt.title("Language Popularity among developers")
plt.show()
```

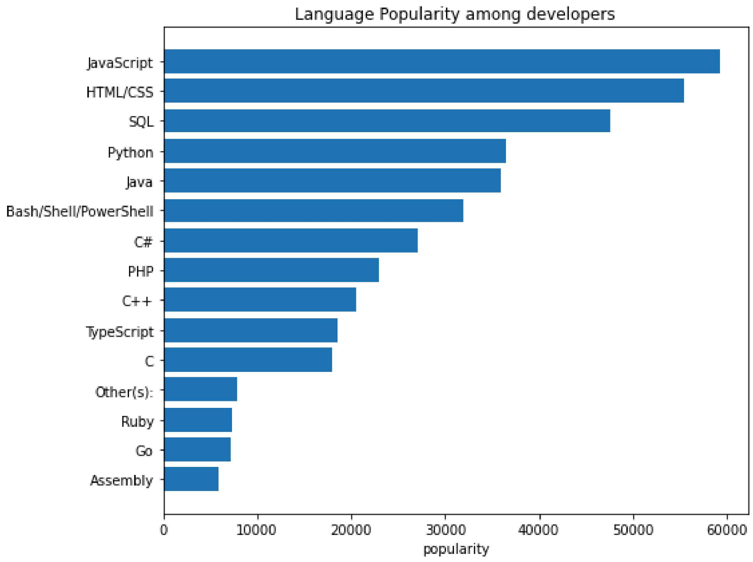


6. Plotting Horizontal bar chart

```
plt.barh(languages, popularity)
plt.xlabel("popularity")
plt.title("Language Popularity among developers")
plt.show()
```



```
languages.reverse()
popularity.reverse()


# https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages

plt.figure(figsize=(8,6))
plt.barh(languages, popularity)
plt.xlabel("popularity")
plt.title("Language Popularity among developers")
plt.tight_layout()
plt.show()
```



Show Your Creativity

Automobile Land Speed Records (GR 5-10)

In the first recorded automobile race in 1898, Count Gaston de Chasseloup-Laubat of Paris, France, drove 1 kilometer in 57 seconds for an average speed of 39.2 miles per hour(mph) or 63.1 kilometers per hour (kph). In 1904, Henry Ford drove his Ford Arrow across frozen Lake St. Clair, MI, at an average speed of 91.4 mph. Now, the North American Eagle is trying to break a land speed record of 800 mph. The Federation International deL'Automobile (FIA), the world's governing body for motor sport and land speed records,recorded the following land speed records.

PIE CHARTS

```
Start coding or generate with AI.
```

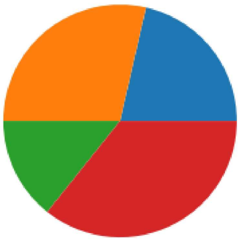
```
from matplotlib import pyplot as plt
```

Recommended to use when we have 5 or less values to plot

1. Plotting the Pie Chart

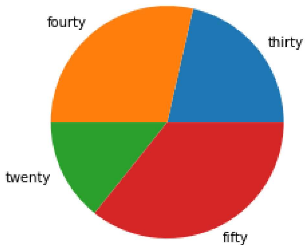
```
slices = [30, 40, 20, 50] #sum needs not be 100
```

```
plt.pie(slices)
plt.show()
```



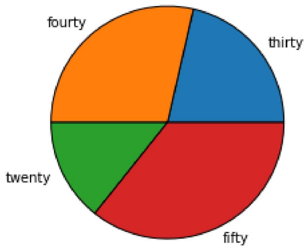
2. Adding labels to the pie chart

```
labels = ['thirty','fourty', 'twenty','fifty']
plt.pie(slices, labels=labels)
plt.show()
```



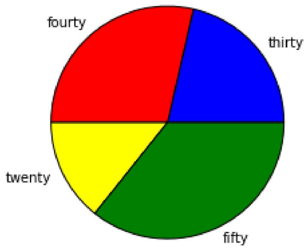
3. setting edge color

```
plt.pie(slices, labels=labels, wedgeprops={'edgecolor':'black'})
plt.show()
```



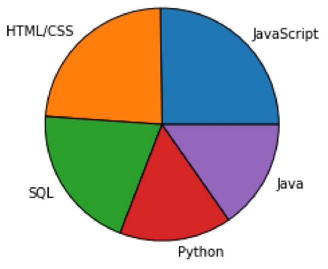
4. setting color of the slices

```
color = ['blue','red','yellow','green']

#hexadecimal color codes can also be used
plt.pie(slices, labels=labels, colors=color, wedgeprops={'edgecolor':'black'})
plt.show()
```



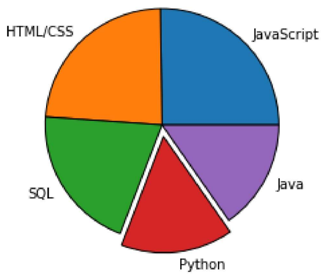
5. plotting real world data

```
labels = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']

slices = [59219, 55466, 47544, 36443, 35917]


plt.pie(slices, labels=labels, wedgeprops={'edgecolor':'black'})
plt.show()
```



6. Exploding the slice

```
explode = [0, 0, 0, 0.1, 0]

plt.pie(slices, labels=labels, explode=explode, wedgeprops={'edgecolor':'black'})
plt.show()
```

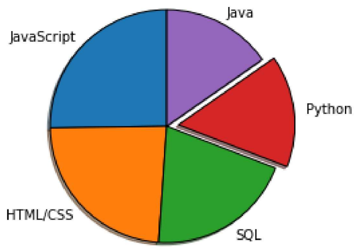


7. adding shadow to the chart

```
plt.pie(slices, labels=labels, explode=explode, shadow=True, wedgeprops={'edgecolor':'black'})
plt.show()
```

8. setting the starting angle

```
plt.pie(slices, labels=labels, explode=explode, shadow=True, startangle=90, wedgeprops={'edgecolor':'black'})
plt.show()
```



9. displaying percentage of each slices

```
plt.pie(slices, labels=labels, explode=explode, shadow=True, startangle=60, autopct="%0.1f%%", wedgeprops={'edgecolor':'black'})
plt.show()
```

Show Your Creativity

Covid 19 India Data as on 5th Sept 2020

```
import matplotlib.pyplot as plt
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps

Enter your authorization code:
··········
Mounted at /content/drive
```

```
data = pd.read_csv('/content/drive/My Drive/data/Covid_19.csv')
```

```
data.head()
```

	State	last_updated	population	tested__last_updated	total__confirmed	total
0	AN	2020-09-05T22:09:31+05:30	397000	9/4/2020	3292	
1	AP	2020-09-05T20:15:29+05:30	52221000	9/5/2020	487331	
2	AR	2020-09-06T00:53:37+05:30	1504000	9/5/2020	4914	

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.