

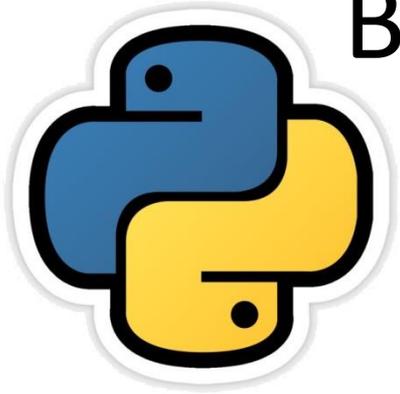
# Plotting with Pyplot-I

## Bar Graphs and Scatter Charts

सीबीएसई पाठ्यक्रम पर आधारित

कक्षा -12

### अध्याय -3 (IP)



द्वारा:

संजीव भदौरिया

स्नातकोत्तर शिक्षक (संगणक विज्ञान )

के० वि० बाराबंकी (लखनऊ संभाग)

# Data Visualization क्या है?

- जैसा की हम सब जानते हैं की आजकल big data का युग है ।
- और data किसी भी organization के लिए decision making के लिहाज़ से अत्यंत महत्वपूर्ण है ।
- इतने बड़े data के संग्रह से data प्राप्त कर उसका analysis करने के लिए उसका visualization अत्यंत महत्वपूर्ण तरीका होता है ।
- “Data Visualization एक प्रकार से information का graphical या visual representation होता है जोकि chart, graph या map इत्यादि हो सकता है ।”

# Data Visualization

- “Data Visualization एक प्रकार से information का graphical या visual representation होता है जोकि chart, graph या map इत्यादि हो सकता है |”

“Data Visualization basically refers to the graphical or visual representation of information and data using visual elements like charts, graphs or maps.

- इसके लिए हम इस अध्याय के द्वारा पाइथन में Pyplot के प्रयोग के बारे में जानेंगे |
- Pyplot के प्रयोग से data को किस प्रकार visualize किया जाता है यह भी जानेंगे |

# MATPLOTLIB Library के **Pyplot** का प्रयोग

- matplotlib एक python library होती है जो MATLAB के जैसे ही 2D graphics दर्शाने के लिए कई सारे functions प्रदान करती है ।
- इसे हम python की high quality plotting library कह सकते हैं ।
- matplotlib कई सारे नामित (named) collections प्रदान करता है जिनमे ढेरों methods समाहित होते हैं - **Pyplot** भी उन collections में से एक है ।
- Pyplot, matplotlib के अंतर्गत methods का एक collection है जिनकी सहायता से 2D plots को आसानी से बनाया जा सकता है ।

# Matplotlib को install और import करना

CA

Command Prompt

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\SHAURYA>pip install matplotlib
Collecting matplotlib
  Using cached matplotlib-1.5.1-cp27-none-win32.whl
Collecting numpy>=1.6 (from matplotlib)
  Using cached numpy-1.11.1-cp27-none-win32.whl
Collecting python-dateutil (from matplotlib)
  Using cached python_dateutil-2.5.3-py2.py3-none-any.whl
Collecting pytz (from matplotlib)
  Using cached pytz-2016.6.1-py2.py3-none-any.whl
Collecting cycler (from matplotlib)
  Using cached cycler-0.10.0-py2.py3-none-any.whl
Collecting pyparsing!=2.0.4,>=1.5.6 (from matplotlib)
  Using cached pyparsing-2.1.5-py2.py3-none-any.whl
Collecting six>=1.5 (from python-dateutil->matplotlib)
  Using cached six-1.10.0-py2.py3-none-any.whl
Installing collected packages: numpy, six, python-dateutil, pytz, cycler, pyparsing, matplotlib
```

# PyPlot को import करना

- Pyplot को import करने के लिए निम्न syntax लिखना होगा-

```
import matplotlib.pyplot
```

अथवा

```
import matplotlib.pyplot as pl
```

- उसके बाद कमांड्स को pl के साथ (.) लगाकर प्रयोग करते हैं।
- लेकिन उसके पहले हमें थोडा numpy के बारे में जानना होगा।
- Numpy plotting के लिए बहुत आवश्यक function प्रदान करता है ।
- Numpy vectorized functions को भी सपोर्ट करता है ।

# NumPy Arrays

- NumPy (“Numerical Python” या Numeric Python”) भी पाइथन का एक open source module है जो arrays और matrices पर तेज़ गणितीय फंक्शन प्रदान करता है |
- NumPy को प्रयोग करने के लिए इसे import करना होता है | जिसका statement निम्न है

```
>>> import numpy as np
```

(इसमें np, numpy का बाद में उसे किये जाने वाला नाम है जो की वैकल्पिक होता है |)

- NumPy arrays 2 रूपों में होता है -
  - 1-D array –  
इसे Vectors के नाम से भी जानते हैं |
  - Multidimensional arrays –  
जिन्हें हम Matrices के नाम से भी जानते हैं |

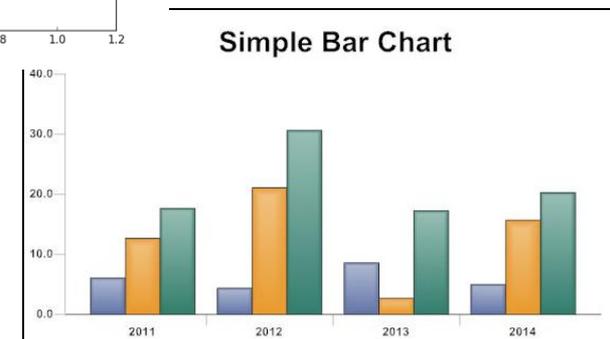
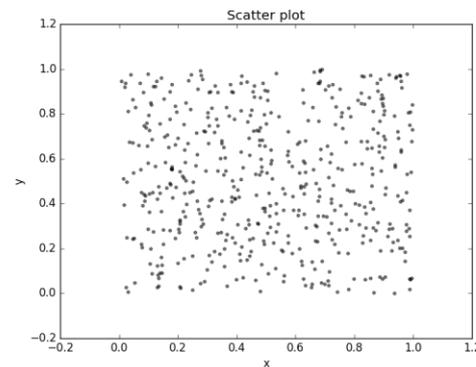
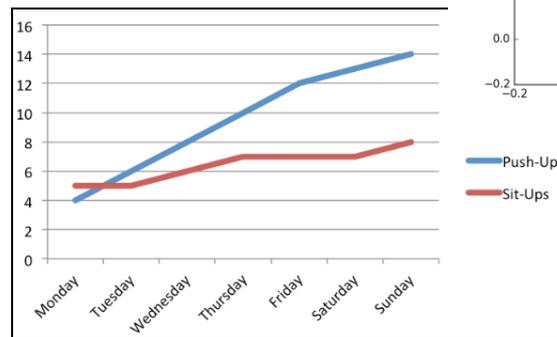
```
>>> import numpy as np
>>> lst = [1,2,3,4]
>>> a1=np.array(lst)
>>> lst
[1, 2, 3, 4]
>>> print(a1)
[1 2 3 4]
>>> a1
array([1, 2, 3, 4])
```

List और array में अन्तर देखिये

# Basics of Simple Plotting

- Compiled data का graphical representation ही data visualization कहलाता है |
- अतः chart और graph data visualization के लिए बहुत ही महत्वपूर्ण tool हैं |
- Pyplot का प्रयोग करके आप विभिन्न प्रकार के graph और chart बना सकते हैं |
- पाठ्यक्रम के अनुसार हमें निम्न प्रकार के chart का अध्ययन करना है |

- Line chart
- Bar Chart
- Scatter Plot



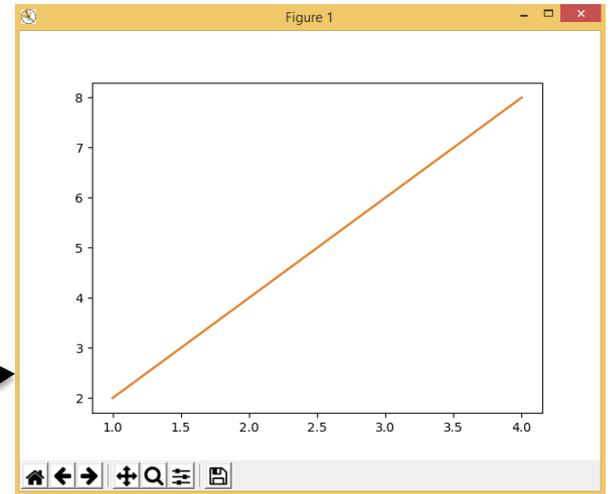
# Line Chart बनाना

- एक line graph या line chart एक प्रकार का chart होता है जो data points के series में data का प्रदर्शन करता है जिन्हें हम markers कहते हैं और ये markers एक सीधी रेखा द्वारा जुड़े होते हैं |
- इसके लिए pyplot interface एक plot() function प्रदान करता है | एक उदाहरण से यह आसानी से समझ आ जायेगा -

```
>>> import matplotlib.pyplot as plt
>>> a=[1,2,3,4]
>>> b=[2,4,6,8]
>>> plt.plot(a,b)
```

```
[<matplotlib.lines.Line2D object at 0x00000021D8F979E8>]
```

```
>>> plt.show()
```



यदि हमें labels लगाने हैं तो इसका अध्ययन हम अगली स्लाइड में करेंगे

# Line Chart बनाना

- एक उदाहरण लेते हैं – जहाँ 5 overs में बनाये गए runs का data है | इसके साथ ही हम X axis का नाम overs रखेंगे और Y axis का नाम runs रखेंगे |

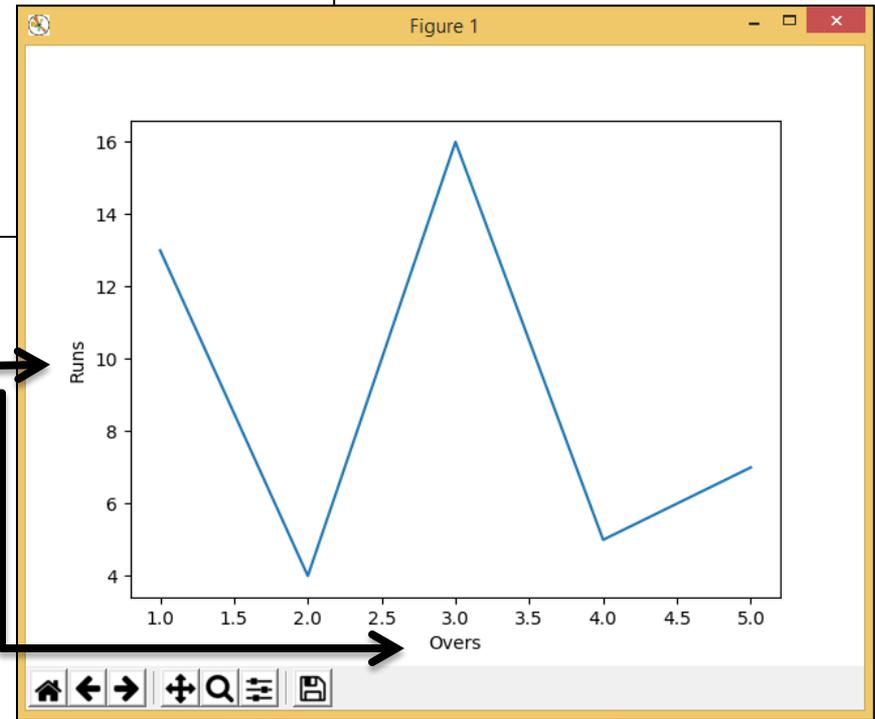
```
import matplotlib.pyplot as plt
over = [1,2,3,4,5]
run = [13,4,16,5,7]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.plot(over,run)
plt.show()
```

Labels लगाने के लिए इन functions का प्रयोग किया गया है

रिजल्ट वाले chart चित्र में दोनों labels दिखाई दे रहे हैं |

**Important :**

जिस axis का आप नाम नहीं देना चाहते हैं उस axis के function को कॉल मत करिए |



# Line color, width और style को बदलना

- इसके लिए syntax है -

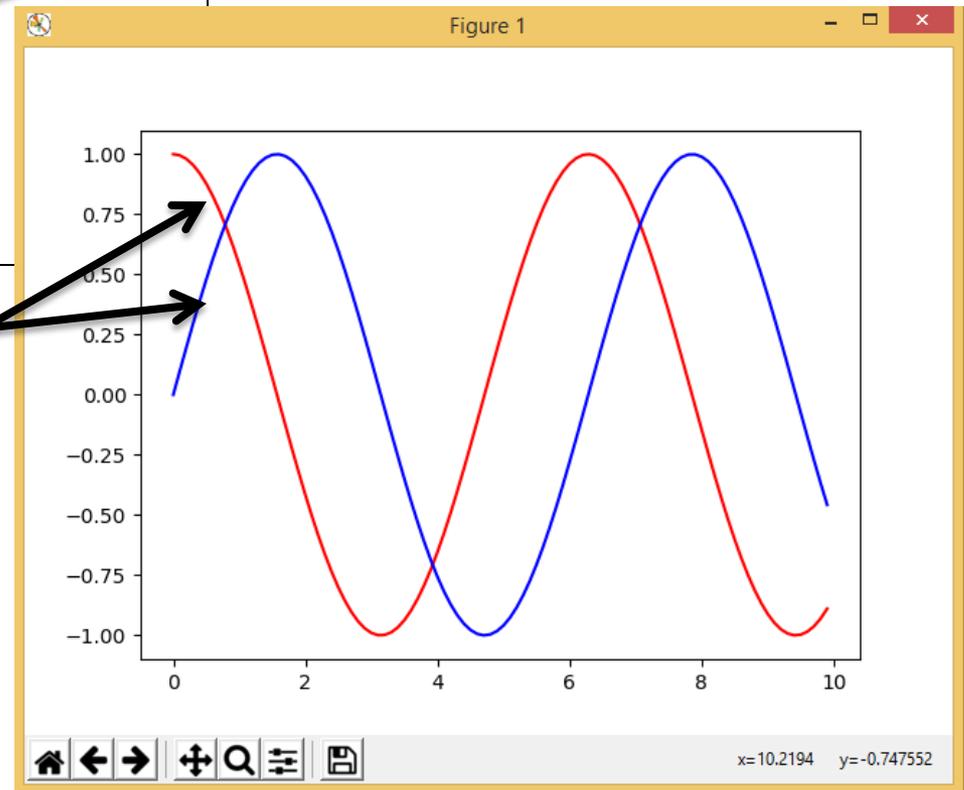
`matplotlib.pyplot.plot(<data1>,<data2>,<color code>)`

```
import matplotlib.pyplot as plt
import numpy as np
x=np.arange(0,10,0.1)
a=np.cos(x)
b=np.sin(x)
plt.plot(x,a,'r')
plt.plot(x,b,'b')
plt.show()
```

Red color के लिए 'r' और blue color के लिए 'b' का प्रयोग किया गया है।

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

रिजल्ट वाले chart चित्र में दोनों color दिख रहे हैं।

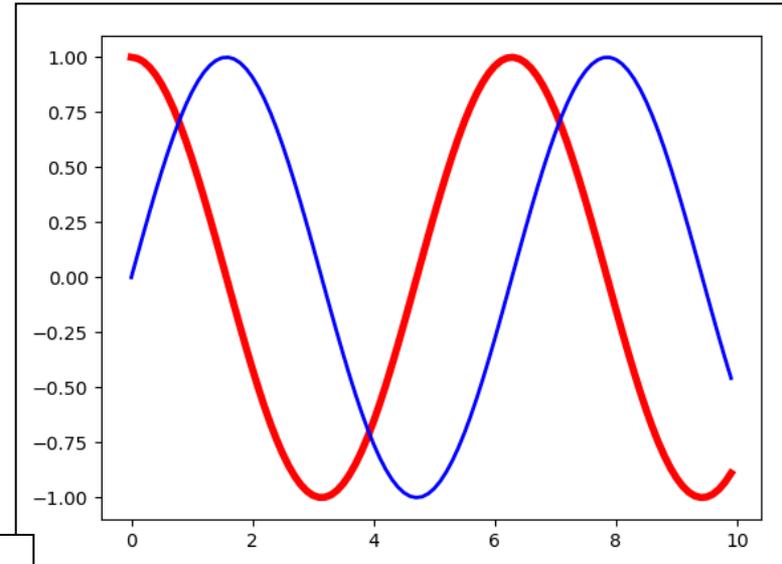


# Line color, width और style को बदलना

- इसके लिए syntax है -

`matplotlib.pyplot.plot(<data 1>, <data 2>, linewidth=<val> ...)`

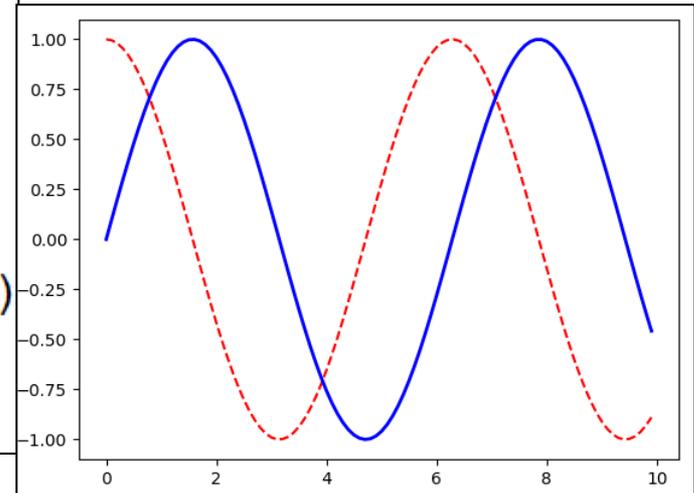
```
import matplotlib.pyplot as pl
import numpy as np
x=np.arange(0,10,0.1)
a=np.cos(x)
b=np.sin(x)
pl.plot(x,a,'r',linewidth=4)
pl.plot(x,b,'b',linewidth=2)
pl.show()
```



```
import matplotlib.pyplot as pl
import numpy as np
x=np.arange(0,10,0.1)
a=np.cos(x)
b=np.sin(x)
pl.plot(x,a,'r',linestyle='dashed')
pl.plot(x,b,'b',linewidth=2)
pl.show()
```

अलग अलग line  
स्टाइल्स के लिए ये  
प्रयोग करें |

',' '\_', '\_\_', '.\_'



# Marker type, size और color को बदलना

- इसके लिए syntax है -

`matplotlib.pyplot.plot(<data1>,<data2>,linestyle=<val>...)`

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

```
over=[1,2,3,4,5]
```

```
run=[13,5,7,16,4]
```

```
plt.xlabel("Overs")
```

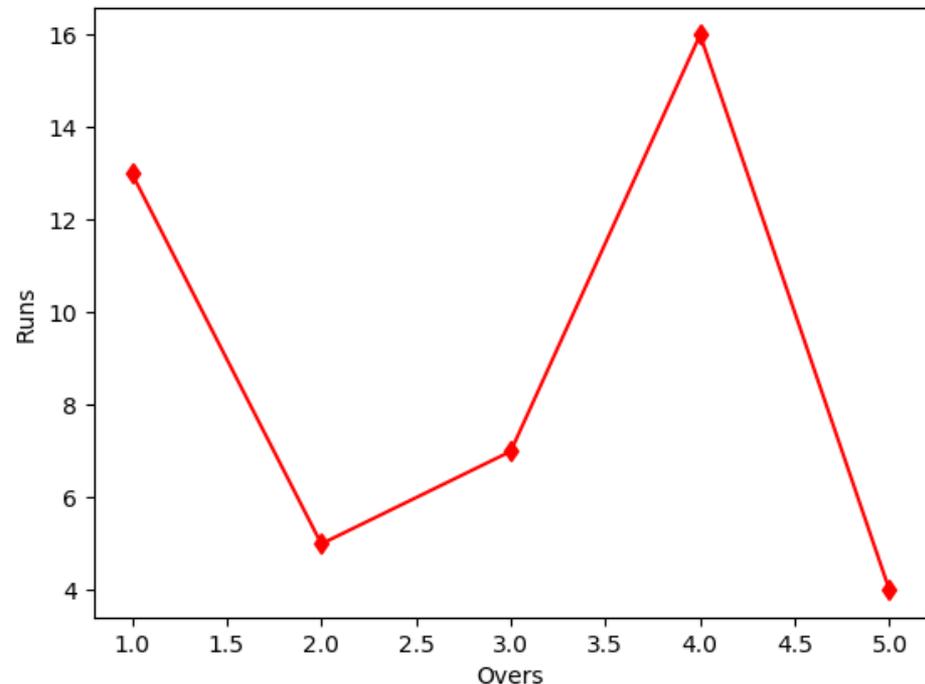
```
plt.ylabel("Runs")
```

```
plt.plot(over,run,'r',marker='d', markersize=6,markeredgecolor='red')
```

```
plt.show()
```

[https://matplotlib.org/2.1.1/api/as\\_gen/matplotlib.pyplot.plot.html](https://matplotlib.org/2.1.1/api/as_gen/matplotlib.pyplot.plot.html) A use full Link to understand pyplot

character	description	character	description
'-'	solid line style	'p'	pentagon marker
'--'	dashed line style	'*'	star marker
'-.'	dash-dot line style	'h'	hexagon1 marker
':'	dotted line style	'H'	hexagon2 marker
'.'	point marker	'+'	plus marker
','	pixel marker	'x'	x marker
'o'	circle marker	'D'	diamond marker
'v'	triangle_down marker	'd'	thin_diamond marker
'^'	triangle_up marker	' '	vline marker
'<'	triangle_left marker	'_'	hline marker
'>'	triangle_right marker		
'1'	tri_down marker		
'2'	tri_up marker		
'3'	tri_left marker		
'4'	tri_right marker		
's'	square marker		

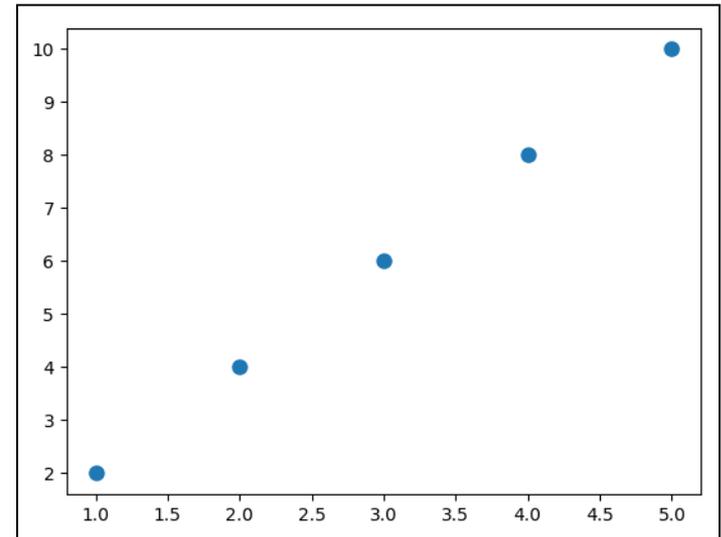


# Scatter Chart बनाना

- बिन्दुओं के द्वारा 2 axes पर बना हुआ graph ही scatter chart कहलाता है जी दो data set के मध्य सम्बन्ध दर्शाता है |
- दो तरीकों से scatter chart बनाया जा सकता है |
  - plot( ) function से
  - scatter( ) function से
- plot() function का syntax है –

`matplotlib.pyplot.plot(a,b,<point style >, markersize=<value>)`

```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[2,4,6,8,10]
plt.plot(a,b,"o",markersize=8)
plt.show()
```

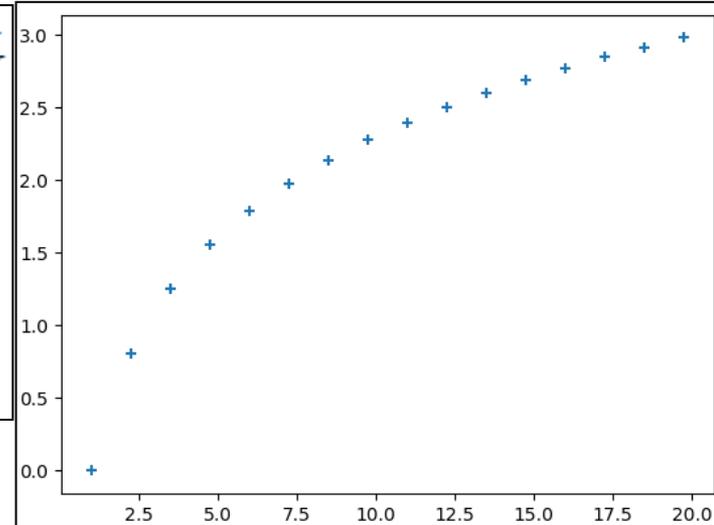


# Scatter Chart बनाना

- scatter () function का syntax है –

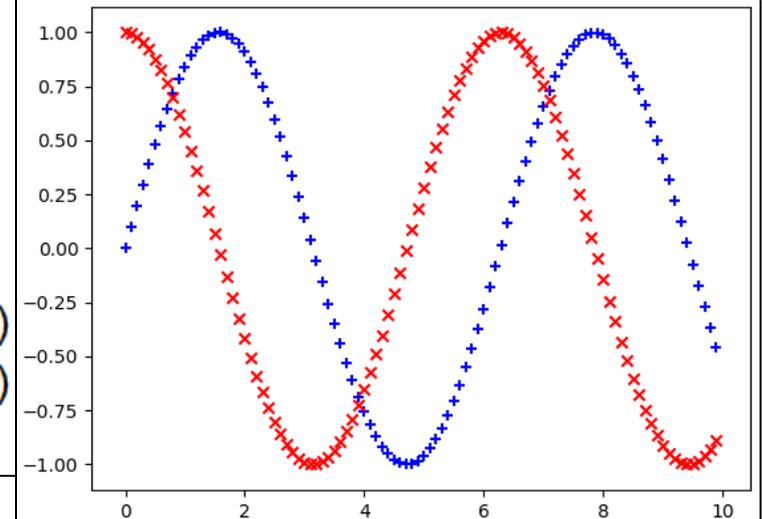
`matplotlib.pyplot.scatter(a, b, marker=<type>)`

```
import matplotlib.pyplot as plt
import numpy as np
a=np.arange(1,20,1.25)
b=np.log(a)
plt.scatter(a,b,marker="+")
plt.show()
```



- Line color Change करना -

```
import matplotlib.pyplot as plt
import numpy as np
x=np.arange(0,10,0.1)
a=np.cos(x)
b=np.sin(x)
plt.scatter(x,b,c='b', marker="+")
plt.scatter(x,a,c='r', marker="x")
plt.show()
```



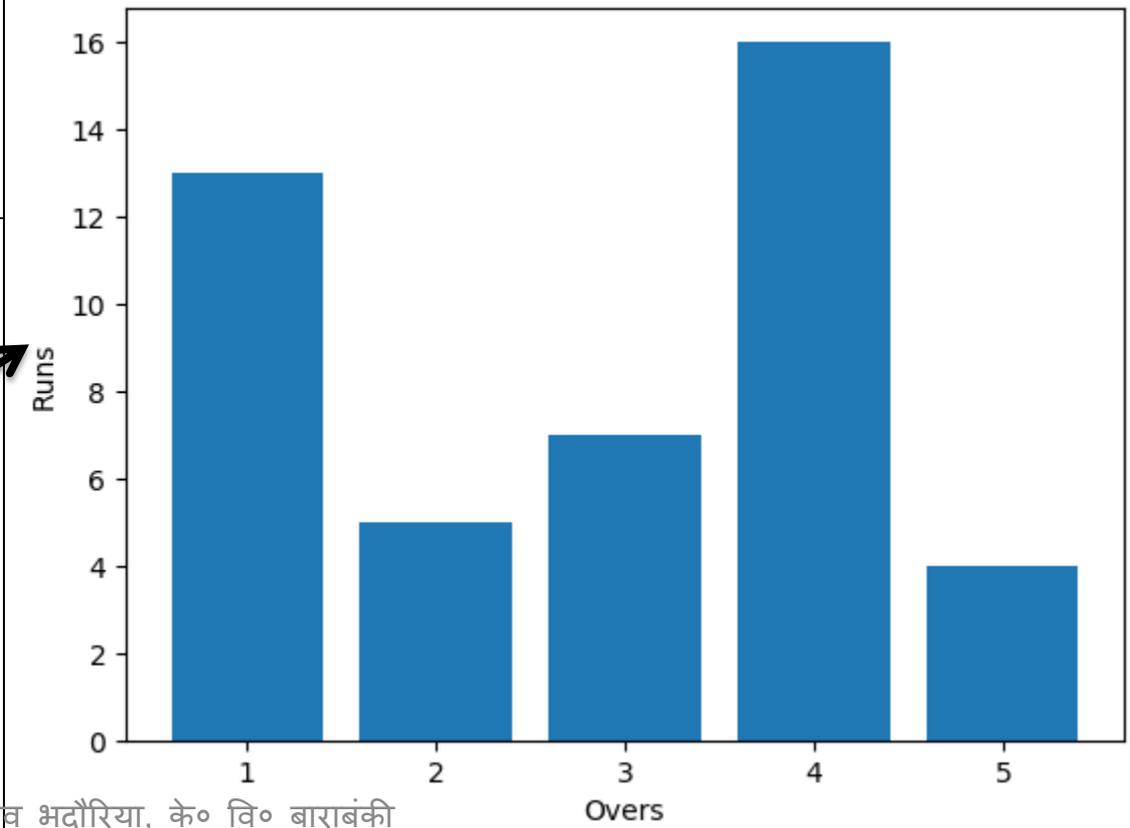
# Bar Chart बनाना

- विभिन्न ऊंचाइयों के स्तम्भ (Bar) से यदि data को प्रदर्शित करें तो वह bar chart कहलाता है | इसका syntax है – `matplotlib.pyplot.bar(a,b)`

```
import matplotlib.pyplot as plt
over=[1,2,3,4,5]
run=[13,5,7,16,4]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.bar(over,run)
plt.show()
```

Lables लगाने के लिए इन functions का प्रयोग किया गया है

रिजल्ट वाले chart चित्र में दोनों lables दिखई दे रहे हैं |  
**Important :**  
जिस axis का आप नाम नहीं देना चाहते हैं उस axis के function को कॉल मत करिए |



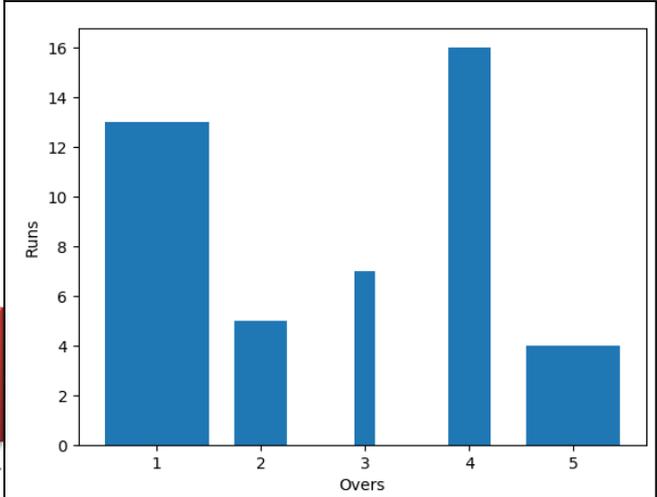
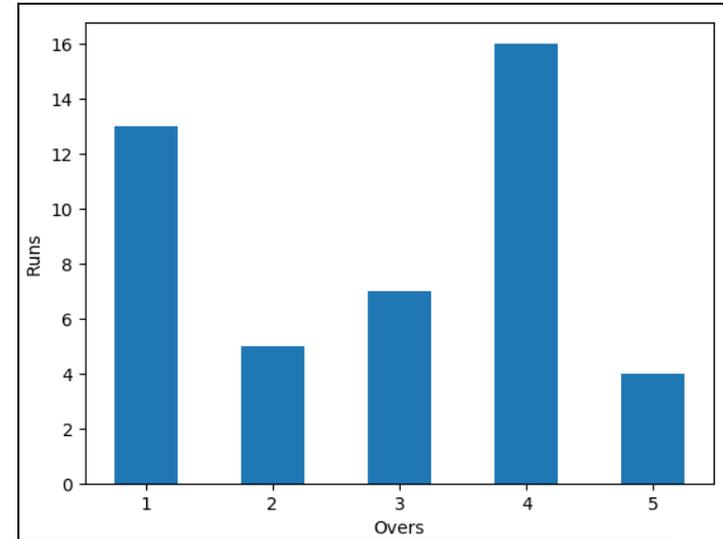
# Bar width को बदलना

- विभिन्न ऊंचाइयों के स्तम्भ (Bar) से यदि data को प्रदर्शित करें तो वह bar chart कहलाता है | इसका syntax है –

`matplotlib.pyplot.bar(a, b, width=<Value>)`

```
import matplotlib.pyplot as plt
over=[1,2,3,4,5]
run=[13,5,7,16,4]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.bar(over,run,width=1/2)
plt.show()
```

```
import matplotlib.pyplot as plt
over=[1,2,3,4,5]
run=[13,5,7,16,4]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.bar(over,run,width=[1,0.5,0.2,0.4,0.9])
plt.show()
```



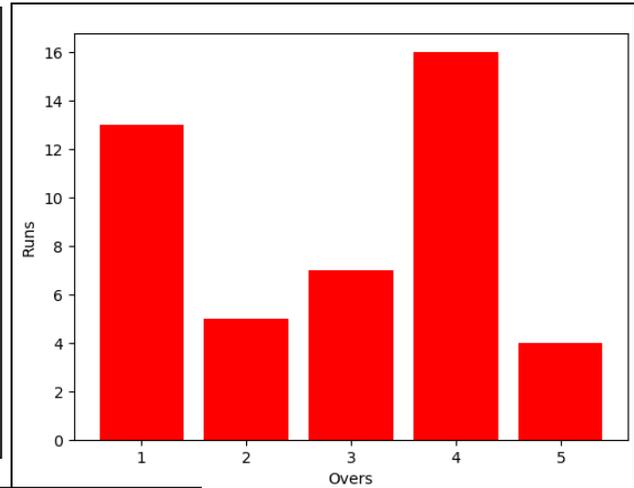
अलग अलग data के bar की चौड़ाई भी अलग अलग की जा सकती है |

# Bar color को बदलना

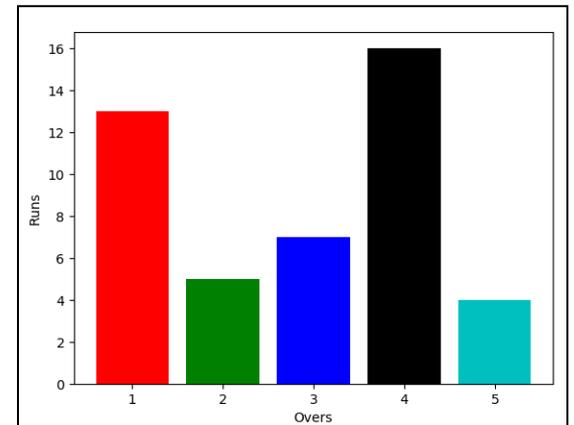
- विभिन्न ऊंचाइयों के स्तम्भ (Bar) से यदि data को प्रदर्शित करें तो वह bar chart कहलाता है | इसका syntax है –

`matplotlib.pyplot.bar(a, b, color=<code>)`

```
import matplotlib.pyplot as plt
over=[1,2,3,4,5]
run=[13,5,7,16,4]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.bar(over,run,color='red')
plt.show()
```

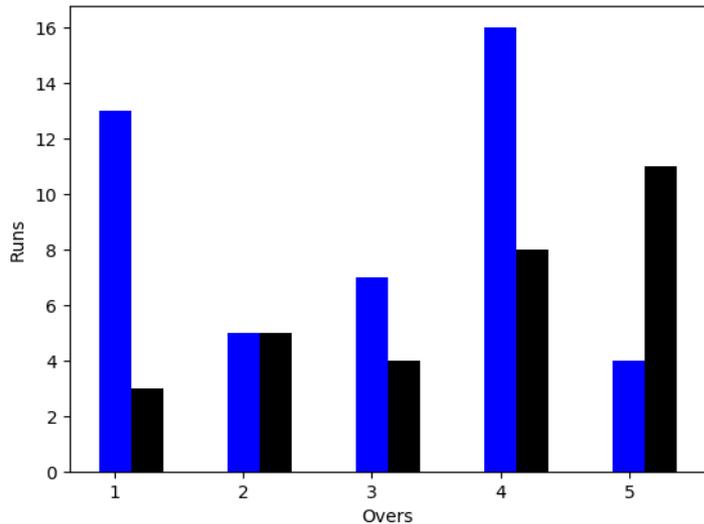


```
import matplotlib.pyplot as plt
over=[1,2,3,4,5]
run=[13,5,7,16,4]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.bar(over,run,color=['r','g','b','k','c'])
plt.show()
```



# Multiple Bar Chart को बनाना

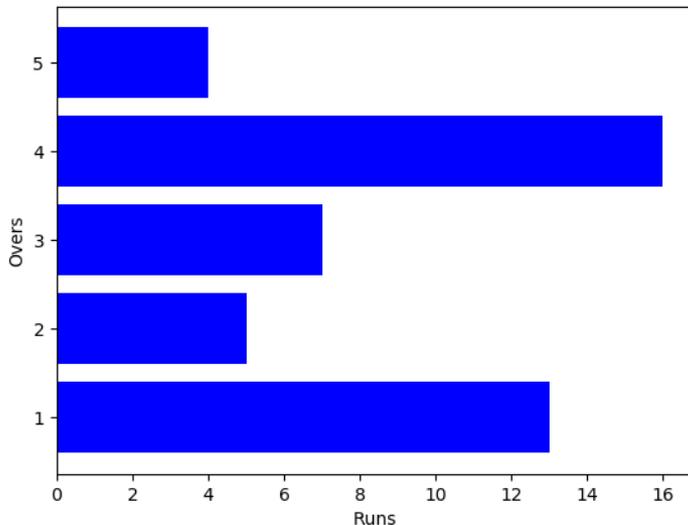
```
import matplotlib.pyplot as plt
import numpy as np
over=np.arange(1.0,6.0,1.0)
Ind=[13,5,7,16,4]
Nz=[3,5,4,8,11]
plt.xlabel("Overs")
plt.ylabel("Runs")
plt.bar(over,Ind,color='b',width=0.25)
plt.bar(over+0.25,Nz,color='k',width=0.25)
plt.show()
```



यहाँ समझने लायक कोड इस स्थान पर है। Show() Function को कॉल करने से पहले आप जितनी बार bar function को प्रयोग करेंगे वह एक ही chart में जुड़ता जायेगा।

# Horizontal Bar Chart को बनाना

```
import matplotlib.pyplot as plt
import numpy as np
over=np.arange(1.0,6.0,1.0)
Ind=[13,5,7,16,4]
plt.xlabel("Runs")
plt.ylabel("Overs")
plt.barh(over,Ind,color='b')
plt.show()
```



इसके लिए barh() function का प्रयोग किया गया है |

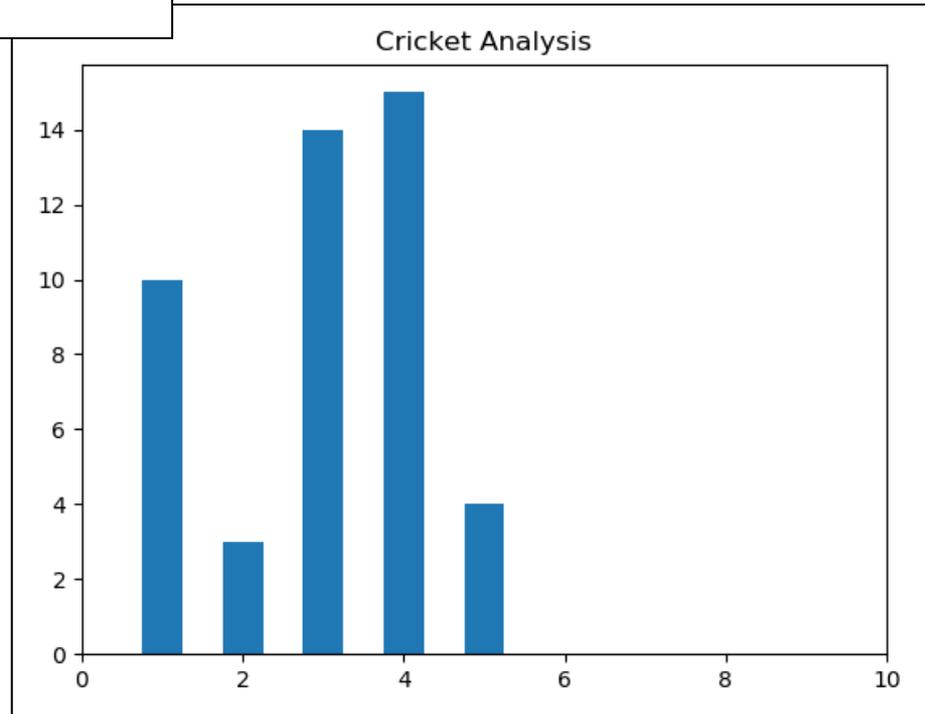
# Chart की anatomy

- Chart का अपना एक ढांचा (structure) होता है | जिसके बारे में हम यहाँ देखेंगे –
- **Figure** – कोई भी चार इसके अंतर्गत ही बनेगा. यह plot का area होता है |
- **Axes** – यह वह area होता है जिसमे वास्तविक plotting होगी |
  - **Axis Label** – यह x-axis और y-axis से मिलकर बनी है |
  - **Limits** – ये x-axis और y-axis पर मार्क की गयी values की सीमा है |
  - **Tick\_Marks** – ये x-axis और y-axis पर individual value होती है |
- **Title** – plot के टॉप पर दिखने वाला text होता है |
- **Legends** – यह उन data के अलग अलग color का सेट है जो plotting के समय प्रयोग किया जाता है इन्हें हम मानक भी कह सकते हैं |

# Title add करना, xlimit और ylimit सेट करना

```
import matplotlib.pyplot as plt
import numpy as np
over=[1,2,3,4,5]
run=[10,3,14,15,4]
plt.xlim(0,10)
plt.title("Cricket Analysis")
plt.bar(over,run,width=1/2)
plt.show()
```

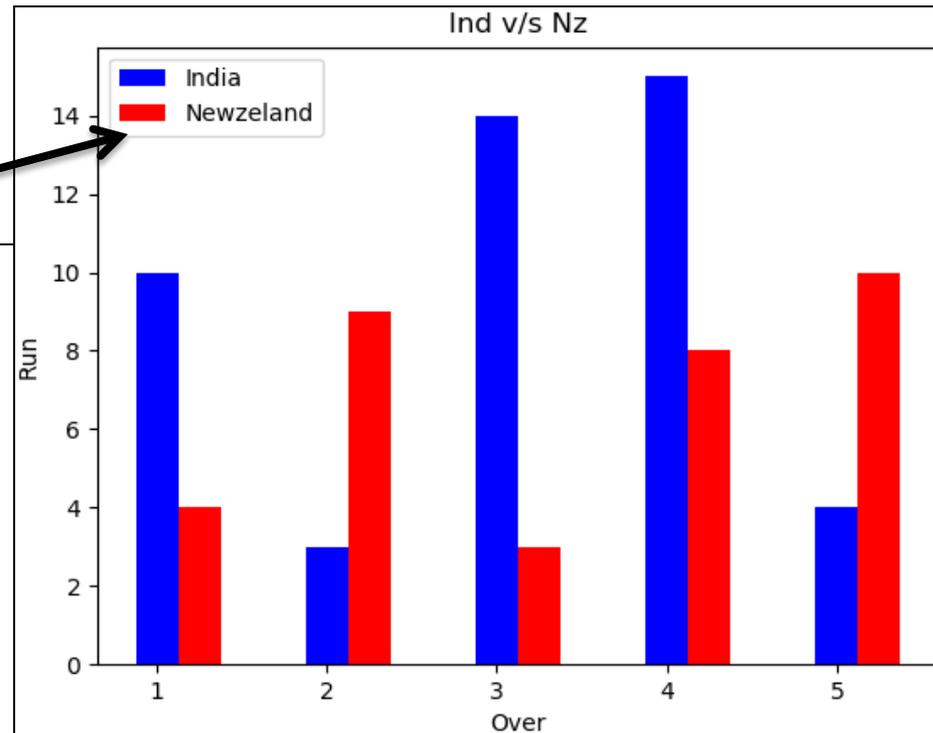
Code को ध्यान से देखिये इसमें `plt.title()` और `plt.xlim()` function का प्रयोग किया गया है |



# Legends को add करना

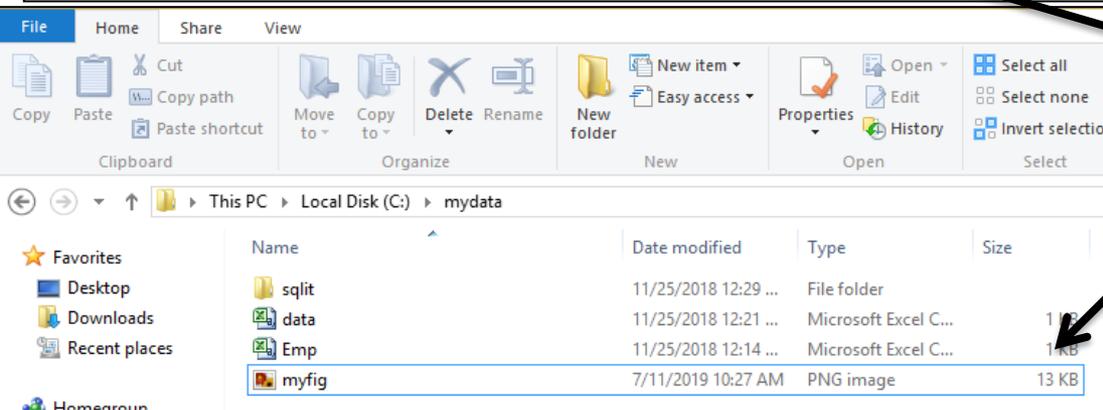
```
import matplotlib.pyplot as plt
import numpy as np
over=np.arange(1.0, 6.0, 1.0)
ind=[10, 3, 14, 15, 4]
nz=[4, 9, 3, 8, 10]
plt.title("Ind v/s Nz")
plt.bar(over, ind, color='b', width=0.25, label='India')
plt.bar(over+0.25, nz, color='r', width=0.25, label='Newzeland')
plt.legend(loc='upper left')
plt.xlabel("Over")
plt.ylabel("Run")
plt.show()
```

Legends



# Saving a Figure

```
import matplotlib.pyplot as plt
import numpy as np
over=np.arange(1.0,6.0,1.0)
ind=[10,3,14,15,4]
nz=[4,9,3,8,10]
plt.title("Ind v/s Nz")
plt.bar(over,ind,color='b',width=0.25,label='India')
plt.bar(over+0.25,nz,color='r',width=0.25,label='Newzeland')
plt.legend(loc='upper left')
plt.xlabel("Over")
plt.ylabel("Run")
plt.savefig("C:\\MyData\\myfig.png")
plt.show()
```

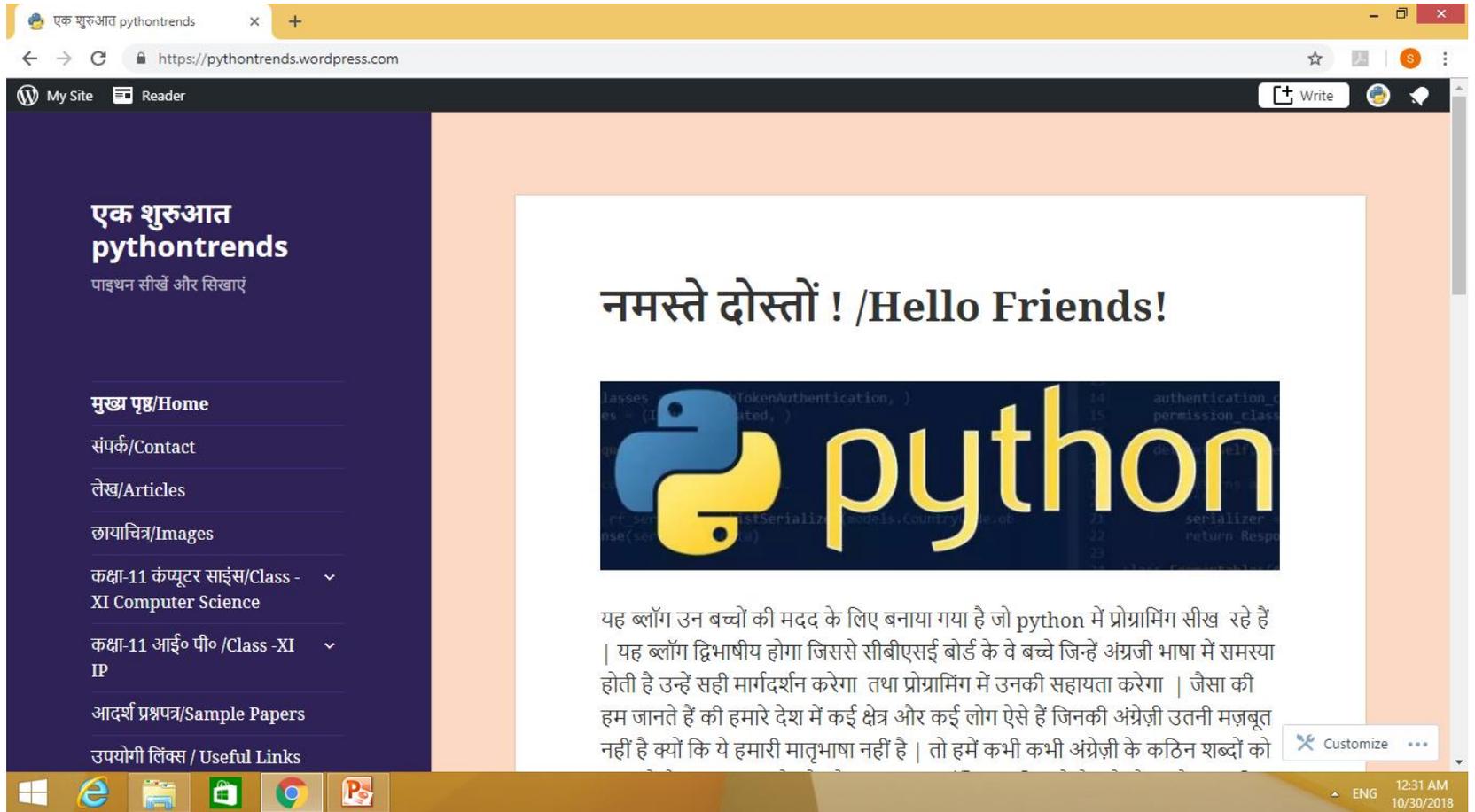


Plotting के figure को सेव करने के लिए यह line लिखनी होती है ।

# धन्यवाद

और अधिक पाठ्य-सामग्री हेतु निम्न लिंक पर क्लिक करें -

[www.pythontrends.wordpress.com](http://www.pythontrends.wordpress.com)



एक शुरुआत pythontrends

पाठ्यन सीखें और सिखाएं

मुख्य पृष्ठ/Home

संपर्क/Contact

लेख/Articles

छायाचित्र/Images

कक्षा-11 कंप्यूटर साइंस/Class - XI Computer Science

कक्षा-11 आई० पी० /Class -XI IP

आदर्श प्रश्नपत्र/Sample Papers

उपयोगी लिंक्स / Useful Links

नमस्ते दोस्तों ! /Hello Friends!

python

यह ब्लॉग उन बच्चों की मदद के लिए बनाया गया है जो python में प्रोग्रामिंग सीख रहे हैं | यह ब्लॉग द्विभाषीय होगा जिससे सीबीएसई बोर्ड के वे बच्चे जिन्हें अंग्रेजी भाषा में समस्या होती है उन्हें सही मार्गदर्शन करेगा तथा प्रोग्रामिंग में उनकी सहायता करेगा | जैसा की हम जानते हैं की हमारे देश में कई क्षेत्र और कई लोग ऐसे हैं जिनकी अंग्रेजी उतनी मज़बूत नहीं है क्यों कि ये हमारी मातृभाषा नहीं है | तो हमें कभी कभी अंग्रेजी के कठिन शब्दों को

Customize ...

ENG 12:31 AM 10/30/2018