**Name:** Jaydeep Solanki

**Roll No:** 22BEC059

**Subject:** Introduction To AI-ML

**Date:** 25/03/2023

## Practical 1: Basics of NumPy and Matplotlib

In [1]:

**import** numpy **as** np

In [2]:

print("Hello World") a, b **=** 2, 5

print(a,b)

print("Value of a =", a)

Hello World 2 5

Value of a = 2

In [3]:

str **=** 'Hello World!' print (str[**-**1])

print (str[**-**3:**-**1])

print (str[**-**12:])

!

ld

Hello World!

In [4]:

list **=** [ 'abcd', 786 , 2.23, 'john', 70.2 ] tinylist **=** [123, 'john']

print (list)

print (list[0]) print (list[1:3])

print (list[2:]) print (tinylist **\*** 2)

print (list **+** tinylist)

['abcd', 786, 2.23, 'john', 70.2]

abcd

[786, 2.23]

[2.23, 'john', 70.2]

[123, 'john', 123, 'john']

['abcd', 786, 2.23, 'john', 70.2, 123, 'john']

In [5]:

tuple **=** ( 'abcd', 786 , 2.23, 'john', 70.2 ) tinytuple **=** (123, 'john')

print (tuple)

print (tuple[0]) print (tuple[1:3]) print (tuple[2:]) print (tinytuple **\*** 2)

print (tuple **+** tinytuple)

('abcd', 786, 2.23, 'john', 70.2)

abcd

(786, 2.23)

(2.23, 'john', 70.2)

(123, 'john', 123, 'john')

('abcd', 786, 2.23, 'john', 70.2, 123, 'john')

In [6]:

dict **=** {}

dict['one'] **=** "This is one" dict[2] **=** "This is two"

tinydict **=** {'name': 'john','code':6734, 'dept': 'sales'}

print (dict['one']) print (dict[2]) print (tinydict)

print (tinydict**.**keys())

This is one This is two

{'name': 'john', 'code': 6734, 'dept': 'sales'}

dict\_keys(['name', 'code', 'dept'])

In [7]:

a **=** [[1,2,3], [4,5,6]]

print(a[0], a[1][0])

[1, 2, 3] 4

In [8]:

var1 **=** 100

**if** var1:

print ("1 - Got a true expression value") print (var1)

var2 **=** 0

**if** var2:

print ("2 - Got a true expression value") print (var2)

print ("Good bye!")

1 - Got a true expression value 100

In [9]:

amount**=**int(input("Enter amount: "))

**if** amount**<**1000: discount**=**amount**\***0.05

print ("Discount",discount)

**else**:

discount**=**amount**\***0.10

print ("Discount",discount)

Enter amount: 799

Discount 39.95

In [10]:

amount**=**int(input("Enter amount: "))

**if** amount**<**1000: discount**=**amount**\***0.05

print ("Discount",discount)

**elif** amount**<**5000: discount**=**amount**\***0.10

print ("Discount",discount)

**else**:

discount**=**amount**\***0.15

print ("Discount",discount)

print ("Net payable:",amount**-**discount)

Enter amount: 5999

Discount 899.85

Net payable: 5099.15

In [11]:

num**=**int(input("enter number"))

**if** num**%2**==0:

**if** num**%3**==0:

print ("Divisible by 3 and 2")

**else**:

print ("divisible by 2 not divisible by 3")

**else**:

**if** num**%3**==0:

print ("divisible by 3 not divisible by 2")

**else**:

print ("not Divisible by 2 not divisible by 3")

enter number67584 Divisible by 3 and 2

In [12]:

count **=** 0

**while** count **<** 11:

print ('The count is:', count) count **=** count **+** 1

print ("Good bye!")

|  |  |  |  |
| --- | --- | --- | --- |
| The | count | is: | 0 |
| The | count | is: | 1 |
| The | count | is: | 2 |
| The | count | is: | 3 |
| The | count | is: | 4 |
| The | count | is: | 5 |
| The | count | is: | 6 |
| The | count | is: | 7 |
| The | count | is: | 8 |
| The | count | is: | 9 |
| The | count | is: | 10 |

Good bye!

In [13]:

**for** var **in** range(5): print(var)

|  |  |  |
| --- | --- | --- |
|  | | 0 |
| 1 |
| 2 |
| 3 |
| 4 |
| In | [14]: | **for** letter **in** 'Python': |
|  | | **if** letter **==** 'h':  **continue**  print ('Current Letter :', letter) |

|  |  |  |  |
| --- | --- | --- | --- |
| Current | Letter | : | P |
| Current | Letter | : | y |
| Current | Letter | : | t |
| Current | Letter | : | o |
| Current | Letter | : | n |

|  |  |  |
| --- | --- | --- |
| In | [15]: | **def** sum (**\***args):  **return** sum(args) |
|  |  |  |
| In | [16]: | print(sum (1,2,3)) |
|  |  | 6 |
| In | [17]: | x **=** np**.**array([[1,2,3],[4,5,6]], float) print(x, type(x)) |
|  |  | [[1. 2. 3.] |

[4. 5. 6.]] <class 'numpy.ndarray'>

In [18]:

x **=** np**.**array(range(10), float)**.**reshape((2,5)) print(x)

[[0. 1. 2. 3. 4.]

[5. 6. 7. 8. 9.]]

In [19]:

x **=** np**.**array(range(10), float)**.**reshape((5,2)) print(x, x**.**transpose(), sep **=** "\n")

|  |  |
| --- | --- |
| [[0. | 1.] |
| [2. | 3.] |
| [4. | 5.] |
| [6. | 7.] |
| [8. | 9.]] |
| [[0. | 2. 4. 6. 8.] |
| [1. | 3. 5. 7. 9.]] |

In [20]:

x **=** x**.**flatten() print(x)

[0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]

In [21]:

a **=** np**.**array([[1,2], [3,4]], float)

b **=** np**.**array([[4,5], [2,5]], float) print(np**.**concatenate((a,b), axis **=** 1)) print(np**.**concatenate((a,b), axis **=** 0))

|  |  |
| --- | --- |
| [[1. | 2. 4. 5.] |
| [3. | 4. 2. 5.]] |
| [[1. | 2.] |
| [3. | 4.] |
| [4. | 5.] |
| [2. | 5.]] |

In [22]:

x**.**sort() print(x**.**mean())

print(x**.**var())

print(x**.**std()) print(x) print(x**.**min()) print(x**.**argmin()) print(np**.**median(x))

4.5

8.25

2.8722813232690143

[0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]

0.0

0

4.5

In [23]:

a **=** np**.**array([[1,2,3,4,5], [2,3,4,5,5]], float) np**.**corrcoef(a)

np**.**cov(a)

Out[23]:

In [27]:

array([[2.5, 2. ],

[2. , 1.7]])

print(x[x**>=**6]) print([x**>=**6])

[6. 7. 8. 9.]

[array([False, False, False, False, False, False, True, True, True, True])]

In [1]:

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

In [2]:

x **=** np**.**arange(0,10)

In [11]:

y **=** 2**\***x **+** 4

plt**.**title("Linear") plt**.**xlabel("X-Axis") plt**.**ylabel("Y-Axis")

plt**.**plot(x, y, color**=**'m', linestyle**=**'--', marker**=**'.')

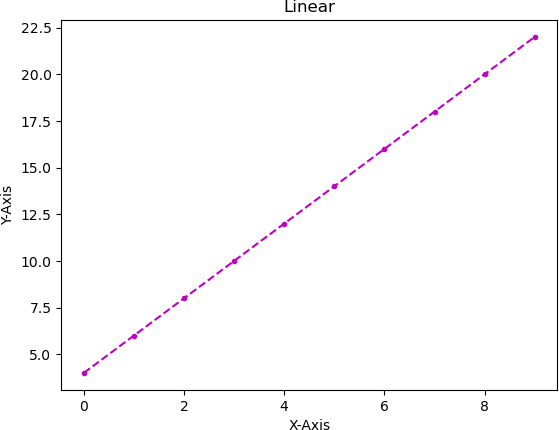
Out[11]:

In [13]:

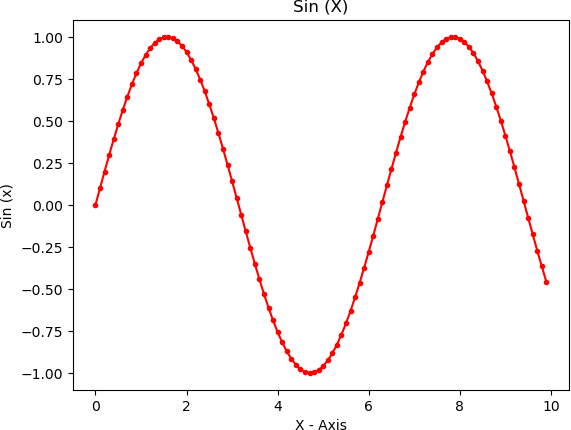
x **=** np**.**arange(0,10, 0.1) y **=** np**.**sin(x)

plt**.**title("Sin (X)") plt**.**xlabel("X - Axis") plt**.**ylabel("Sin (x)")

plt**.**plot(x,y, linestyle**=**'-', marker**=**'.', color**=**'r')



Out[13]:



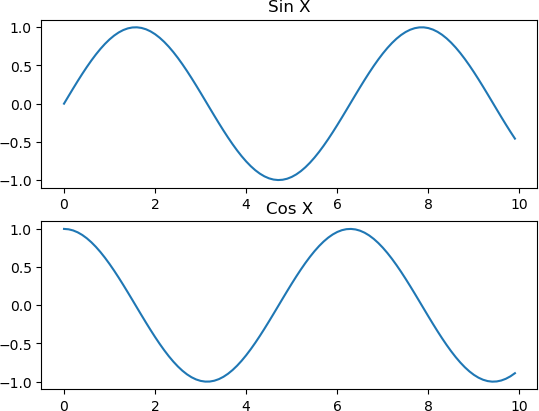
In [14]:

plt**.**subplot(2,1,1) plt**.**title("Sin X") plt**.**plot(x,y)

plt**.**subplot(2,1,2) plt**.**title("Cos X")

plt**.**plot(x, np**.**cos(x))

Out[14]:



In [21]:

x **=** np**.**arange(0, 3**\***np**.**pi, 0.01) sin **=** np**.**sin(x)

cos **=** np**.**cos(x)

plt**.**subplot(2,2,1) plt**.**plot(x, sin)

plt**.**title("Sine") plt**.**subplot(2,2,2) plt**.**plot(x, cos)

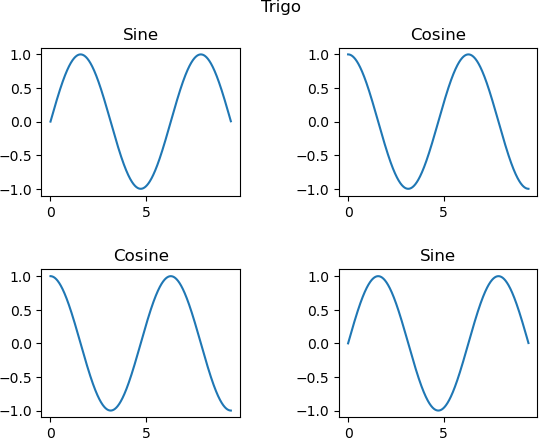
plt**.**title("Cosine") plt**.**subplot(2,2,3) plt**.**plot(x, cos)

plt**.**title("Cosine") plt**.**subplot(2,2,4) plt**.**plot(x, sin)

plt**.**title("Sine")

plt**.**subplots\_adjust(hspace**=**0.5, wspace **=** 0.5) plt**.**suptitle("Trigo")

Out[21]:



In [28]:

y **=** np**.**sin(x) z **=** np**.**cos(x)

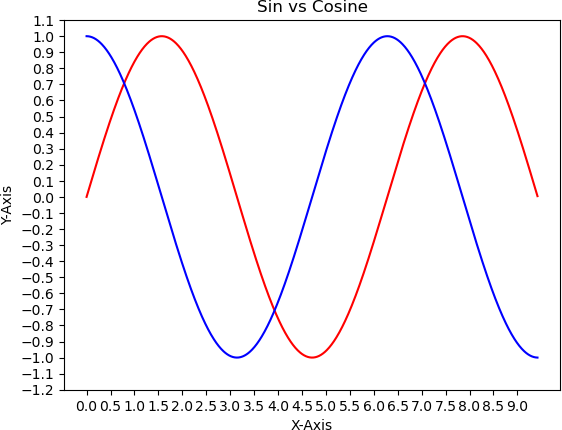
plt**.**title("Sin vs Cosine") plt**.**xlabel("X-Axis")

plt**.**ylabel("Y-Axis")

plt**.**plot(x, y, color**=**'r', label**=**"Sin") plt**.**plot(x, z, color**=**'b', label**=**"Cos") plt**.**xticks(np**.**arange(0,3**\***np**.**pi, 0.5))

plt**.**yticks(np**.**arange(**-**1.2,1.2, 0.1))

Out[28]:



In [32]:

s1 **=** [19, 16, 20, 19]

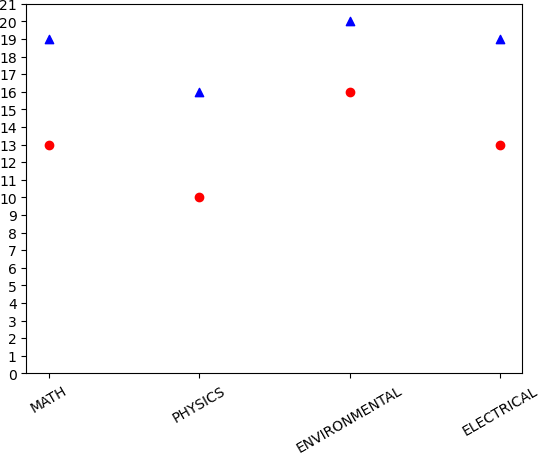
s2 **=** [13, 10, 16, 13]

sub **=** ['MATH', "PHYSICS", "ENVIRONMENTAL", "ELECTRICAL"]

plt**.**scatter(np**.**arange(0, 4), s1, label**=**"Student - 1", color**=**'b', marker**=**'^') plt**.**scatter(np**.**arange(0, 4), s2, label**=**"Student - 1", color**=**'r', marker**=**'o') plt**.**xticks(np**.**arange(0,4), sub, rotation**=**30)

plt**.**yticks(np**.**arange(0,22, 1))

Out[32]:



In [34]:

x **=** [5, 8, 10]

y **=** [12, 16, 6]

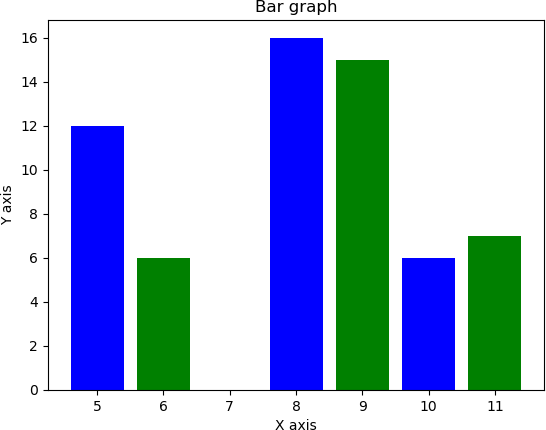
x2 **=** [6, 9, 11]

y2 **=** [6, 15, 7]

plt**.**bar(x, y, color **=** 'b', align **=** 'center') plt**.**bar(x2, y2, color **=** 'g', align **=** 'center') plt**.**title('Bar graph')

plt**.**ylabel('Y axis') plt**.**xlabel('X axis') plt**.**show()

Out[34]:

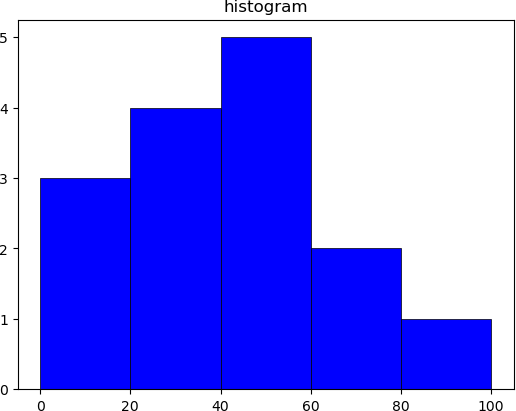


In [35]:

a **=** np**.**array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])

plt**.**hist(a, bins **=** [0,20,40,60,80,100], color **=** 'b', edgecolor**=**'k', linewidt plt**.**title("histogram")

plt**.**show()



In [2]:

**import** cv2

In [3]:

cv2**.**imshow("HEHE",cv2**.**imread("copy.jpg"))

cv2**.**imwrite("copy.jpg",cv2**.**imread("N3.jpeg"))

Out[3]:

In [7]:

**import** pandas **as** pd

**import** numpy **as** np

data**=**pd**.**read\_csv('data.csv',header**=None**) data**=**data**.**values

print(data)

True

[['Student\_id' 'Age' 'Grade' 'Employed' 'marks'] ['1' '19' '1st Class' 'yes' '29']

['2' '20' '2nd Class' 'no' '41']

...

['230' '20' '3rd Class' 'yes' '21']

['231' '19' '1st Class' 'yes' '64']

['232' '20' '3rd Class' 'yes' '30']]

In [9]:

a **=** np**.**array([ [1,2,3], [4,5,6], [7,8,9] ])

df **=** pd**.**DataFrame(a)

df**.**to\_csv("file.csv", header**=None**, index**=False**) df

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Out[9]: |  | 0 | 1 | 2 |
|  | 0 | 1 | 2 | 3 |
|  | 1 | 4 | 5 | 6 |
|  | 2 | 7 | 8 | 9 |