

```
import numpy as np
```

```
arr = np.array([1,2,3,4,5])
print(arr)
```

```
[1 2 3 4 5]
```

```
import numpy as np
```

```
arr = np.array([[1,2,3,1,1],[1,2,3,4,5],[3,1,1,4,5]])
```

```
output = repr(arr).count("1, 1")
print(output)
```

```
2
```

```
import numpy as np
```

```
arr = np.array([[1,2,3,4,5],
                [1,2,3,4,5],
                [1,2,3,4,5]])
```

```
arr.T
```

```
array([[1, 1, 1],
       [2, 2, 2],
       [3, 3, 3],
       [4, 4, 4],
       [5, 5, 5]])
```

```
import numpy as np
```

```
arr = np.array([0,0,1,2,3,4,5,0,0])
np.trim_zeros(arr)
```

```
array([1, 2, 3, 4, 5])
```

```
import numpy as np
```

```
arr = np.array([1,2,3,4,5])
print("original array")
print(arr)
print("reverse array")
arr=arr[::-1]
print(arr)
```

```
original array
[1 2 3 4 5]
reverse array
[5 4 3 2 1]
```

```
import numpy as np
```

```
a=np.zeros(11)
print("before any change")
print(a)
a[4]=1
print("after first change")
print(a)
a.setflags(write=False)
print("now the array wont accept any value because it has been set to read ")
a[7]=1
```

```

before any change
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
after first change
[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
now thw array wont accept any value because it has been set to read

```

```

-----
ValueError                                Traceback (most recent call last)
<ipython-input-29-135782f9eeaa> in <cell line: 10>()
      8 a.setflags(write=False)
      9 print("now thw array wont accept any value because it has been set to read ")
--> 10 a[7]=1
      11

```

```
import numpy as np
```

```

arr = np.array([75, 42, 60])
print("Given array:")
print(arr)

```

```

print("\nReplace all elements of array which are less than 50. to 15")
arr[arr < 50.] = 15.50

```

```

print("New array :\n")
print(arr)

```

```

Given array:
[75 42 60]

```

```

Replace all elements of array which are less than 50. to 15
New array :

```

```
[75 15 60]
```

```
import numpy as np
```

```

arr = np.array([[20, 67, np.nan],
                [41, 57, np.nan]])

```

```

print("Given array:")
print(arr)

```

```

print("\nRemove all columns containing non-numeric elements ")
print(arr[:, ~np.isnan(arr).any(axis=0)])

```

```

Given array:
[[20. 67. nan]
 [41. 57. nan]]

```

```

Remove all columns containing non-numeric elements
[[20. 67.]
 [41. 57.]]

```

```
import numpy as geek
```

```

b = geek.empty(2, dtype = int)
print("Matrix b : \n", b)

```

```

a = geek.empty([2, 2], dtype = int)
print("\nMatrix a : \n", a)

```

```

Matrix b :
[0 1]

```

```

Matrix a :
[[4626322717216342016 4630967054332067840]
 [4634415122796773376 4633218854145753088]]

```

```
import numpy as np
```

```

arr = np.array([2, 0, 1, 5,
                4, 1, 9])

```

```

print("Given array:", arr)

sorted_index_array = np.argsort(arr)

sorted_array = arr[sorted_index_array]

print("Sorted array:", sorted_array)

n = 3

rslt = sorted_array[-n : ]

print("{} largest value:".format(n),
      rslt[0])

    Given array: [2 0 1 5 4 1 9]
    Sorted array: [0 1 1 2 4 5 9]
    3 largest value: 4

import numpy as np

input_arr = np.array([-1.8, -1.6, -0.5, 0.5,
                     1.6, 1.8, 3.0])
print(input_arr)

floor_values = np.floor(input_arr)
print("\nFloor values : \n", floor_values)

ceil_values = np.ceil(input_arr)
print("\nCeil values : \n", ceil_values)

trunc_values = np.trunc(input_arr)
print("\nTruncated values : \n", trunc_values)

    [-1.8 -1.6 -0.5  0.5  1.6  1.8  3. ]

    Floor values :
    [-2. -2. -1.  0.  1.  1.  3.]

    Ceil values :
    [-1. -1. -0.  1.  2.  2.  3.]

    Truncated values :
    [-1. -1. -0.  0.  1.  1.  3.]

import numpy as np

matrix1 = np.array([[1, 2], [0, 2]])
matrix2 = np.array([[0, 1], [3, 4]])

print("Original matrix:")
print(matrix1)
print(matrix2)

result = np.einsum("mk,kn", matrix1, matrix2)

print("Einstein's summation convention of the two matrix:")
print(result)

    Original matrix:
    [[1 2]
     [0 2]]
    [[0 1]
     [3 4]]
    Einstein's summation convention of the two matrix:
    [[6 9]
     [6 8]]

```

```
import numpy as np

array = np.arange(10)
print(array)
r = np.var(array)
print("\nvariance: ", r)

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

variance: 8.25
```

```
import numpy as np

array = np.arange(10)
print(array)
r = np.std(array)
print("\nstd: ", r)

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

std: 2.8722813232690143
```

```
import numpy as np

array1 = np.array([0, 1, 2])
array2 = np.array([3, 4, 5])

result = np.corrcoef(array1, array2)

print(result)

[[1. 1.]
 [1. 1.]]
```

```
import numpy as np

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

row_mean = np.mean(arr, axis=1)

row1_mean = row_mean[0]
print("Mean of Row 1 is", row1_mean)

row2_mean = row_mean[1]
print("Mean of Row 2 is", row2_mean)

row3_mean = row_mean[2]
print("Mean of Row 3 is", row3_mean)

column_mean = np.mean(arr, axis=0)

column1_mean = column_mean[0]
print("Mean of column 1 is", column1_mean)

column2_mean = column_mean[1]
print("Mean of column 2 is", column2_mean)

column3_mean = column_mean[2]
print("Mean of column 3 is", column3_mean)
```

```

Mean of Row 1 is 2.0
Mean of Row 2 is 5.0
Mean of Row 3 is 8.0
Mean of column 1 is 4.0
Mean of column 2 is 5.0
Mean of column 3 is 6.0

```

```

import numpy as P
x = (10,20,30)
y = (30,40,50)
print("Add one polynomial to another:")
print(P.polyadd(x,y))
print("Subtract one polynomial from another:")
print(P.polysub(x,y))
print("Multiply one polynomial by another:")
print(P.polymul(x,y))
print("Divide one polynomial by another:")
print(P.polydiv(x,y))

```

```

Add one polynomial to another:
[40 60 80]
Subtract one polynomial from another:
[-20 -20 -20]
Multiply one polynomial by another:
[ 300 1000 2200 2200 1500]
Divide one polynomial by another:
(array([0.33333333]), array([ 6.66666667, 13.33333333]))

```

```

import numpy
List = [1, 2, 3, 4, 5]
Array = numpy.array(List)

print('Array:\n', Array)
file = open("file1.txt", "w+")

content = str(Array)
file.write(content)
file.close()

file = open("file1.txt", "r")
content = file.read()

print("\nContent in file1.txt:\n", content)
file.close()

```

```

Array:
[1 2 3 4 5]

Content in file1.txt:
[1 2 3 4 5]

```

```

import numpy as np
import os

File_data = np.loadtxt("/content/example.txt.txt", dtype=int)
print(File_data)

1234

```

```

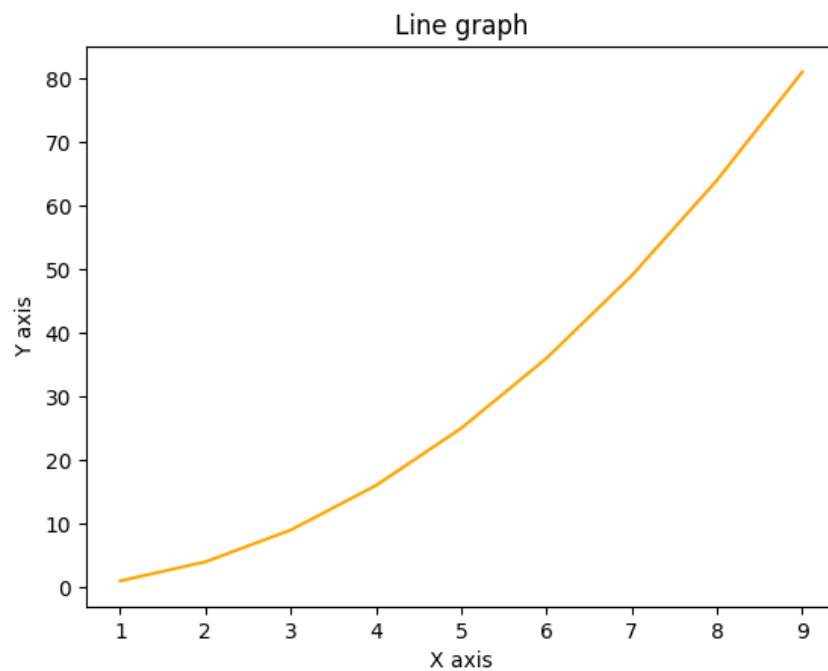
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(1, 10)
y = x * x

plt.title("Line graph")
plt.xlabel("X axis")

```

```
plt.ylabel("Y axis")
plt.plot(x, y, color="orange")
plt.show()
```



```
import numpy as np
```

```
a = np.random.randint(100, size=(50))
```

```
np.histogram(a, bins = [0, 10, 20, 30, 40,
                        50, 60, 70, 80, 90,
                        100])
```

```
hist, bins = np.histogram(a, bins = [0, 10,
                                     20, 30,
                                     40, 50,
                                     60, 70,
                                     80, 90,
                                     100])
```

```
print()
print(hist)
print(bins)
print()
```

```
[3 4 7 2 7 8 5 8 2 4]
[ 0 10 20 30 40 50 60 70 80 90 100]
```

 0s

completed at 3:29 AM