```
#Install numpy-pip3 install numpy|| pip install numpy


#Numpy-Numeric python
#Most powerful package-library
#Numpy-Dimensions
#A-one Dimensional Array
#[A,B]||[A,B]-Two Dimensional array-Row||Column
#Three Dimensional
#Array-Storing,Manipulating,Similar Data type
#Faster than any mathematical functions
#convinient compared to other packages
#Memory efficiency
#Support multidimensional array
#Install numpy- pip3 install numpy||pip install numpy


# create a numpy array
#1D array
import numpy as np
arr1D=np.array([1,2,3,4,5])


print(arr1D)
```

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

```
#creating a 2D array
arr2D=np.array([[1,2,3],[4,5,6]])


print(arr2D)
```

```
[[1 2 3]
 [4 5 6]]
```

```
#creating array with default values
#creating matrix with all values of elements as 0
allZeros=np.zeros((2,3))
print(allZeros)
```

```
[[0. 0. 0.]
 [0. 0. 0.]]
```

```
#creating matrix with all values of elements as '1'
allOnes=np.full((2,3),6)
print(allOnes)
```

```
[[6 6 6]
 [6 6 6]]
```

```
#creating matrix with all values of elements as 'constant values-n-fill values'
allConstant=np.full((2,3),6)
print(allConstant)
```

```
[[6 6 6]
 [6 6 6]]
```

```
#creating identity matrix -I-square matrix
#Identity matrix-Aii=1&&Aij=0
#For all belong to R MxN
identity=np.eye(4)
print(identity)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

```
#creating  random matrices
#between 0 and 1
rand1=np.random.rand(3,3)
print(rand1)
#between a and b
rand2=np.random.randint(10,20,(3,3))
print(rand1)
```

```
[[0.05441007 0.93430754 0.10664118]
 [0.06934699 0.13980198 0.23463465]
 [0.27809089 0.44033312 0.96279481]]
[[0.05441007 0.93430754 0.10664118]
 [0.06934699 0.13980198 0.23463465]
 [0.27809089 0.44033312 0.96279481]]
```

```
#array properties
arr=np.array([[1, 2,3,4],[2, 3,4,5]])
print(arr)
#shape of array
print(f"The shape of the array is:{arr.shape}")
#size of array
print(f"The size of the array is:{arr.size}")
#data type
print(f"The data type of the array is:{arr.dtype}")
#dimensions of array
print(f"The dimensions of the array is:{arr.ndim}")
```

```
[[1 2 3 4]
 [2 3 4 5]]
The shape of the array is:(2, 4)
The size of the array is:8
The data type of the array is:int64
The dimensions of the array is:2
```

```
#Indexing and slicing
#Access the elements in 1D array
arr=np.array([1,2,3,4])
print(arr)
print(arr[0])
#slicing the elements
print(arr[1:3])
```

```
#Accessing elements in 2D array
arr2=np.array([[1,2,3],[2,3,4]])
print(arr2)
#slicing the elements
print(arr2[1,1])
print(arr2[0,:])# first row with all columns
print(arr2[:,0])# first column with all rows
```

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

```
l1=[]
[l1.append(_) for _ in range (10)]
print(l1)
#alternative
l2=np.arange(0,10)
print(l2)
l3=np.arange(10,0,-1)
print(l3)
l4=np.arange(10,-10,-3)
print(l4)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0 1 2 3 4 5 6 7 8 9]
[10  9  8  7  6  5  4  3  2  1]
[10  7  4  1 -2 -5 -8]
```

```
#mathematical operation
a=np.array([1,2,3])
b=np.array([3,4,5])
print(f"The sum of a and b is {a+b}")
print(f"The difference of a and b is {a-b}")
print(f"The product of a and b is {a*b}")
print(f"The division of a and b is {a/b}")
```

```
The sum of a and b is [4 6 8]
The difference of a and b is [-2 -2 -2]
The product of a and b is [ 3  8 15]
The division of a and b is [0.33333333 0.5        0.6       ]
```

```
#Matrix Multiplication
b1=np.array([[1,2,3],[2,3,4]])
b2=np.array([[5,6],[7,8],[1,3]])
#Dot product---product=AB-----A m xn----B n x p
c=np.dot(b1,b2)
print(c)
```

```
[[22 31]
 [35 48]]
```