```
import numpy as np
z=np.ones((3,5),dtype=float)
print(z)
```

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

```
z=np.zeros(10,dtype=int)
print(z)
```

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

```
i3=np.eye(3)
print(i3)
```

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

Creating an Array of five value evenly spaced between 0 an 1

```
ls=np.arange(0,20,2)
print(ls)
```

```
[ 0  2  4  6  8 10 12 14 16 18]
```

```
ls=np.linspace(0,1,5)
print(ls)
```

```
[0.   0.25 0.5  0.75 1.  ]
```

```
xu=np.random.random((3,3))
print(xu)
```

```
[[0.85674141 0.10460297 0.61371774]
 [0.9565524  0.86768755 0.36726997]
 [0.38921593 0.04690021 0.20189765]]
```

Creating a 30X30 array of normally distibution random value with mean 0 and standard derivation 1

```
xn=np.random.normal(0,1,(30,30))
print(np.mean(xn))
print(np.std(xn))
```

```
0.005583209892403907
0.9917750803796332
```

creating an 3X3 array filled with Rnadom Numbers

```
xi=np.random.randint(0,10,(3,3))
print(xi)
```

```
[[8 8 0]
 [3 9 8]
 [5 4 6]]
```

Creatuing an Empty Array

```
xe=np.empty((3,4))
print(xe)
```

```
[[1.25063899e-311 3.16202013e-322 0.00000000e+000 0.00000000e+000]
 [1.01855798e-312 2.35916043e+184 5.16320581e-066 9.20669347e+169]
 [8.60784795e-043 4.23042234e+175 3.54143886e-033 4.91605345e-062]]
```

Attributes of NUmpy

```
print(xe.dtype)
print(xe.size)
print(xe.size)
print("itemsize: ",xe.itemsize,"bytes")
print("ntypes: ",xe.nbytes,"bytes")
```

```
float64
12
12
```

```
itemsize:  8 bytes
ntypes:  96 bytes
```

Creatiung array from list

Array Operation

```
arra_1d=np.array([1,2,3])
array_add=array_1d+10
print("\nadding 10 to 1D array: ")
print(array_add)

array_2d=np.array([[10,20,30],[40,50,60]])
array_mul=array_2d*2
print("\nMultiply by 2 to 2D array: ")
print(array_mul)
```

```
adding 10 to 1D array:
[11 12 13]

Multiply by 2 to 2D array:
[[ 20  40  60]
 [ 80 100 120]]
```

Array Indexing and slicing

```
# array_2d=np.array([[10,20,30],[40,50,60]])
print("\nELement at index 2 in 1D array: ",array_1d[2])
print("Element from index 1 to 3 in 1D Array:",array_1d[1:4])
print("\n Element at row 1,column 2in 2D Array:",array_2d[1,2])
print("Element in first twp rows and column 1 and 2: ")
print(array_2d[:2,1:3])
```

```
ELement at index 2 in 1D array:  3
Element from index 1 to 3 in 1D Array: [2 3]

 Element at row 1,column 2in 2D Array: 60
Element in first twp rows and column 1 and 2:
```

```
[[20 30]
 [50 60]]
```

Array BroadCasting IN Numpy

```
result=array_2d+array_1d
print("Result of broadcasting 1D array to 2D array: ",result)
```

```
Result of broadcasting 1D array to 2D array:  [[11 22 33]
 [41 52 63]]
```

Reshapping anf Flattening Array

```
array_reshaped=array_1d.reshape((1,3))
print("\n REshapes 1D array to 2D array: ")
print(array_reshaped)

matrix=np.random.randint(1,10,(4,5))
print("Original 10X2 Matrix:")
print(matrix)

reshaped_matrix=matrix.reshape(10,2)
print("Reshaped 10X2 Matrix:")
print(reshaped_matrix)

array_flattend=array_2d.flatten()
print("\nFlattened 2D array to 1D array:")
print(array_flattend)
```

```
 REshapes 1D array to 2D array:
[[1 2 3]]
Original 10X2 Matrix:
[[7 7 5 8 8]
 [1 6 4 1 4]
 [3 1 5 5 2]
 [2 9 7 8 7]]
Reshaped 10X2 Matrix:
[[7 7]
 [5 8]
 [8 1]
 [6 4]
 [1 4]
 [3 1]
 [5 5]
 [2 2]
 [9 7]
```

```
 [8 7]]

Flattened 2D array to 1D array:
[10 20 30 40 50 60]
```

Combining Array

```
array_vstack=np.vstack((array_1d,array_1d))
print("\nVerical Stack of 1D Array:")
print(array_vstack)

array_hstack=np.hstack((array_1d,array_1d))
print("\nHorizonatal Stack od 1D array:")
print(array_hstack)
```

```
Verical Stack of 1D Array:
[[1 2 3]
 [1 2 3]]

Horizonatal Stack od 1D array:
[1 2 3 1 2 3]
```

Mathematical Operation on Arrays

```
array1=np.array([1,2,3])
array2=np.array([4,5,6])
sum_array=array1 + array2
diff_array=array1 - array2
product_array=array1 * array2
quotient_array=array1 / array2
print(sum_array)
print(diff_array)
print(product_array)
print(quotient_array)
```

```
[5 7 9]
[-3 -3 -3]
[ 4 10 18]
[0.25 0.4  0.5 ]
```

.Dot Product

```
dot_product=np.dot(array1,array2)
print(dot_product)
```

```
32
```

```
matrix1=np.array([[1,2],[3,4,]])
matrix2=np.array([[5,6],[7,8]])

matrix_product=np.matmul(matrix1,matrix2)
print(matrix_product)
```

```
[[19 22]
 [43 50]]
```

Statistcal Operation

```
mean_value=np.mean(array1)
median_value=np.median(array1)
std_derivation=np.std(array1)
total_sum=np.sum(array1)
cum_sum=np.cumsum(array1)
min_value=np.cumsum(array1)
max_value=np.max(array1)
print(f"sum={total_sum},mean={mean_value},media={median_value},standar
d Deviation={std_derivation}")
print(f"MInimun={min_value},Maximum={max_value}")
```

```
sum=6,mean=2.0,media=2.0,standard Deviation=0.816496580927726
MInimun=[1 3 6],Maximum=3
```