

```
In [ ]: import numpy as np
```

```
In [ ]: # numpy array
arr=np.array([1,2,3,4])
print(arr)
```

```
In [ ]: #array with only zeroes
arr_z=np.zeros((2,4))
print(arr_z)
```

```
In [ ]: #array with 1
arr_1=np.ones((3,3))
print(arr_1)
```

```
In [ ]: #empty array
arr_empty=np.empty((2,4))
print(arr_empty)
```

```
In [ ]: # arrange method
arr_arange=np.arange(-10)
print(arr_arange)

arr_arange1=np.arange(10)
print(arr_arange1)
```

```
In [ ]: # changing the shape
arr=np.array([1,2,3,4])
arr.reshape(2,2)
```

```
In [ ]: # linspace
lin_space=np.linspace(10,100,10)
print(lin_space)
```

```
In [ ]: # two-dimensional array
twoD_array = arr.reshape(2, 2)
print(twoD_array)
```

```
In [ ]: # three-dimensional array
threeD_array = np.arange(27).reshape(3,3,3)
print(threeD_array)
```

```
In [ ]: # displaying the no. of dimensions
print(arr.ndim)
print(threeD_array.ndim)
print(twoD_array.ndim)
```

```
In [ ]: # shape of array
print(threeD_array.shape)
print(twoD_array.shape)
print(arr.shape)
```

```
In [ ]: # size of array
print(twoD_array.size)
print(arr.size)
print(threeD_array.size)
```

## operations

```
In [ ]: a=[1,2,3,4,5]
b=[6,7,8,9,10]
```

```
In [ ]: print(a+b)
```

```
In [ ]: # converting to array and then adding
an=np.array(a)
bn=np.array(b)
print(an+bn)
```

```
In [ ]: print(np.add(4,5))
print(np.add(a,5))
```

```
In [ ]: # comparision operator
c=np.array([34,65,43,21,12,32,13])
print(c[c>4])
```

```
In [ ]: # logical operator
np.logical_and(threeD_array>10,threeD_array<20)
```

```
In [ ]: np_trial_data = np.zeros((3, 3))
print(np_trial_data)
print("\n")
trial_data = np_trial_data[:,1:3]
print(trial_data)
```

```
In [ ]: # printig specifc values
print(threeD_array)
print("\n")
print(threeD_array[2,:,2]) #matrix 2, all rows , 2nd column
```

```
In [ ]: # ravel - single line
arr2=np.array([[54,67,43,21,21,356],[35,87,654,33,67,88]])
arr2.ravel()
```

```
In [ ]: # reshape - does not change original array
arr2.reshape((3,4))
```

```
In [ ]: # resize - changes original array
arr2.resize(3,4)
print(arr2)
```

```
In [ ]: # hsplit - splits array
np.hsplit(arr2,4)
```

```
In [ ]: #hstack - combines array
print(arr)
print(arr2)
np.hstack((arr,arr))
```

```
In [ ]: import numpy as np
np_trial_data_1 = np.array([[1,2,3],[4,5,6], [7,8,9]])
print(np_trial_data_1)
```

```
In [ ]: a=np.array([1,2,3,4])
b=np.array([2,3,4,5])
a*b # should be same length
```

```
In [ ]: np_arr_1 = np.array([12,45,8])
var = [2,3,4] # same length or 1 number
np_arr_1 * var
```

```
In [ ]: np_arr_1*5
```

```
In [ ]: a = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
print(a)
```

```
In [ ]: x=np.array([[1,2,3], [4,5,6], [7,8,9]])
np.linalg.inv(x)
```

```
In [ ]: # sum of diagonal
np.trace(x)
```

```
In [ ]:
```