In [2]:

```
import numpy as np
```

In [3]:

```
x=[1,2,3]
x
```

Out[3]:

```
[1, 2, 3]
```

In [4]:

```
type(x)
```

Out[4]:

```
list
```

In [5]:

```
x=np.array([1,2,3])
x
```

Out[5]:

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

In [6]:

```
type(x)
```

Out[6]:

```
numpy.ndarray
```

1 D Array

In [7]:

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

Out[7]:

```
array([1, 2, 3, 4, 5, 6])
```

In [8]:

```
a=[1,2,3,4]                #list to array
arr=np.array(a)
arr
```

Out[8]:

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

In [9]:

```
#take input from user

a=[]                    # empty list
cnt=1                   # count
for i in range(int(input("how many elements you want:"))):
    val=eval(input(f"Enter value {cnt}:"))      #eval = function like input(anothe
    a.append(val)
    cnt+=1
b=np.array(a)
b
```

```
how many elements you want:5
Enter value 1:12
Enter value 2:10
Enter value 3:6
Enter value 4:8
Enter value 5:5
```

Out[9]:

```
array([12, 10,  6,  8,  5])
```

In [11]:

```
a=[]                    # empty list
cnt=1                   # count
for i in range(int(input("how many elements you want:"))):
    val=str(input(f"Enter value {cnt}:"))      #eval = function like input(another
    a.append(val)
    cnt+=1
b=np.array(a)
b
```

```
how many elements you want:5
Enter value 1:a
Enter value 2:b
Enter value 3:c
Enter value 4:d
Enter value 5:e
```

Out[11]:

```
array(['a', 'b', 'c', 'd', 'e'], dtype='<U1')
```

2 D Array

In [14]:

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

Out[14]:

```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [15]:

```
a.shape
```

Out[15]:

```
(2, 3)
```

In [16]:

```
type(a)
```

Out[16]:

```
numpy.ndarray
```

In [17]:

```
#To check dimensions

np.ndim(a)      #ndim=n dimension (use=  to find out dimension)
```

Out[17]:

```
2
```

In [19]:

```
print(np.ndim(a))                #or
print(a.ndim)
```

```
2
2
```

3 D Array

(matrices, rows, columns)

In [20]:

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

Out[20]:

```
array([[[ 1,  2,  3],
        [ 4,  5,  6]],

       [[ 7,  8,  9],
        [10, 11, 12]]])
```

In [21]:

```
a.ndim
```

Out[21]:

```
3
```

```
In [22]: a.shape

Out[22]: (2, 2, 3)
```

Attributes of Numpy

```
In [24]: a=np.array([[1,2,3],[4,5,6]])
         a

Out[24]: array([[1, 2, 3],
                [4, 5, 6]])
```

```
In [25]: #gives shape of a matrix or array in rows and columns
```

```
In [26]: a.shape

Out[26]: (2, 3)
```

```
In [27]: #return total number of observations in an array
```

```
In [37]: a.size    #(Total numbers 0f element)

Out[37]: 6
```

```
In [29]: # to change shape of array
```

```
In [34]: a.reshape(3,2)
         a.T                        #transpose= rows to columns and columns to rows

Out[34]: array([[1, 4],
                [2, 5],
                [3, 6]])
```

```
In [36]: a.T

Out[36]: array([[1, 4],
                [2, 5],
                [3, 6]])
```

Joining of array

```
In [44]: np.arange(1,7)

Out[44]: array([1, 2, 3, 4, 5, 6])
```

```
In [45]: np.arange(1,7).reshape(3,2)

Out[45]: array([[1, 2],
                [3, 4],
                [5, 6]])
```

```
In [46]: np.arange(1,7).reshape(3,2).T          #Transpose

Out[46]: array([[1, 3, 5],
                [2, 4, 6]])
```

```
In [48]: a=np.arange(1,7).reshape(2,3)
         b=np.arange(7,13).reshape(2,3)
```

```
In [49]: a
```

```
Out[49]: array([[1, 2, 3],
                [4, 5, 6]])
```

```
In [50]: b
```

```
Out[50]: array([[ 7,  8,  9],
                [10, 11, 12]])
```

Concat a and b

```
In [54]: c=np.concatenate((a,b))
         c
```

```
Out[54]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])
```

```
In [55]: c.shape
```

```
Out[55]: (4, 3)
```

```
In [56]: type(c)
```

```
Out[56]: numpy.ndarray
```

```
In [57]: np.concatenate((a,b),axis=1)
```

```
Out[57]: array([[ 1,  2,  3,  7,  8,  9],
                [ 4,  5,  6, 10, 11, 12]])
```

```
In [58]: np.concatenate((a,b),axis=0)
```

```
Out[58]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])
```

```
In [60]: np.concatenate((a,b),axis=1,dtype="float")
```

```
Out[60]: array([[ 1.,  2.,  3.,  7.,  8.,  9.],
                [ 4.,  5.,  6., 10., 11., 12.]])
```

```
In [ ]:
```