In [ ]:
```
#numpy --> numerical + py

# Numpy stand for numerical python
# 2005 created Travis Oliphant
# numpy is open source and free avaialble package
```

In [1]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\dhruv\appdata\local\programs
\python\python38\lib\site-packages (1.24.2)
```

In [3]:
```
import numpy as np
```

In [4]:
```
np.__version__
```

Out[4]:
```
'1.24.2'
```

In [6]:
```
# 1d
# 2d
# 3d

lst = [20,30,40]
print(type(lst))
myarray = np.array(lst)
print(myarray)
print(type(myarray))
```

```
<class 'list'>
[20 30 40]
<class 'numpy.ndarray'>
```

In [7]:
```
myarray
```

Out[7]:
```
array([20, 30, 40])
```

In [9]:
```
myarray1 = np.array(range(1,10))
myarray1
```

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

In [10]:
```
myarray1.ndim
```

Out[10]:
```
1
```

In [14]:
```
myarray1.shape
```

Out[14]:
```
(9,)
```

2d

In [12]:
```
d2 = np.array([[1,2,3],[4,5,6]])
print(d2)
```

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

In [13]:
```
d2.ndim
```

Out[13]: 2

In [15]:
```
d2.shape
```

Out[15]: (2, 3)

In [18]:
```
d2.itemsize
```

Out[18]: 4

In [19]:
```
d2.dtype
```

Out[19]: dtype('int32')

In [20]:
```
d2.size
```

Out[20]: 6

In [22]:
```
d2.max()
```

Out[22]: 6

In [25]:
```
d2 = np.array([[True,False],[False,True]])
print(d2)
```

```
[[ True False]
 [False  True]]
```

3d

In [26]:
```
d3 = np.array([[[1,2,3],[4,5,6]]])
print(d3)
```

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

In [27]:
```
d3.ndim
```

Out[27]:
```
3
```

In [28]:
```
d3.shape
```

Out[28]:
```
(1, 2, 3)
```

In [32]:
```
d3 = np.array([[[1,2,3],[4,5,6]],[[4,5,6],[4,5,6]]])
print(d3.shape)
```

```
(2, 2, 3)
```

In [33]:
```
d3
```

Out[33]:
```
array([[[1, 2, 3],
        [4, 5, 6]],

       [[4, 5, 6],
        [4, 5, 6]]])
```

Random

In [38]:
```
np.random.rand()
```

Out[38]:
```
0.20789701876391808
```

In [42]:
```
np.random.rand(4,3)
```

Out[42]:
```
array([[0.23575927, 0.95854699, 0.22175755],
       [0.70938507, 0.71985166, 0.00562971],
       [0.41516134, 0.67159305, 0.62369849],
       [0.60490596, 0.86393861, 0.30844395]])
```

In [46]:
```
np.random.randint(1,100)
```

Out[46]:
```
75
```

In [52]:
```
np.random.randint(1,100,4)
```

Out[52]:
```
array([38, 18, 90, 98])
```

Numpy function

In [55]:
```
np.zeros(4)
```

Out[55]:
```
array([0., 0., 0., 0.])
```

In [56]:
```
np.zeros((2,3))
```

Out[56]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [57]:
```
np.ones(4)
```

Out[57]:
```
array([1., 1., 1., 1.])
```

In [58]:
```
np.ones((2,3))
```

Out[58]:
```
array([[1., 1., 1.],
       [1., 1., 1.]])
```

In [62]:
```
np.ones((2,3),dtype='int32') * 5 * 5 *5
```

Out[62]:
```
array([[125, 125, 125],
       [125, 125, 125]])
```

In [65]:
```
np.full((3,3), 20)
```

Out[65]:
```
array([[20, 20, 20],
       [20, 20, 20],
       [20, 20, 20]])
```

In [69]:
```
z = np.identity(4)
z
```

Out[69]:
```
array([[1., 0., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       [0., 0., 0., 1.]])
```

In [70]:
```
np.diag(z)
```

Out[70]:
```
array([1., 1., 1., 1.])
```

Array Indexing

In [71]:
```
a = np.array(range(1,11))
```

In [72]:
```
a
```

Out[72]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [73]:
```
a[3]
```

Out[73]:
```
4
```

In [78]:
```
a[1: 4]
```

Out[78]:
```
array([2, 3, 4])
```

In [79]:
```
a[-1]
```

Out[79]:
```
10
```

In [82]:
```
a[:-3]
```

Out[82]:
```
array([1, 2, 3, 4, 5, 6, 7])
```

In [ ]:
```
#what is the differernce bwteen deep copy and shallow copy?
```

Deep Copy

In [85]:
```
b = a.copy()
b
```

Out[85]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [86]:
```
b[:] = 100
```

In [87]:
```
b
```

Out[87]:
```
array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100])
```

In [88]:
```
a
```

Out[88]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

Shallow copy

In [89]:
```
c = a
```

In [90]:
```
c
```

Out[90]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

In [91]:
```
c[:]=100
```

In [92]:
```
c
```

Out[92]:
```
array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100])
```

In [93]:
```
a
```

Out[93]:
```
array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100])
```

Array Math

In [96]:
```
a = np.array([1,2,3,4,5])
a
```

Out[96]:
```
array([1, 2, 3, 4, 5])
```

In [98]:
```
a + 10
```

Out[98]:
```
array([11, 12, 13, 14, 15])
```

In [99]:
```
a -1
```

Out[99]:
```
array([0, 1, 2, 3, 4])
```

In [100]:
```
a * 10
```

Out[100]:
```
array([10, 20, 30, 40, 50])
```

In [101]:
```
a = np.array([1,2,3,4,5])
b = np.array([1,2,3,4,5])
```

In [102]:
```
a + b
```

Out[102]:
```
array([ 2,  4,  6,  8, 10])
```

In [103]:
```
a-b
```

Out[103]:
```
array([0, 0, 0, 0, 0])
```

In [104]:
```
a *b
```

Out[104]:
```
array([ 1,  4,  9, 16, 25])
```

In [105]:
```
a /b
```

Out[105]:
```
array([1., 1., 1., 1., 1.])
```

In [106]:
```
a + b
```

Out[106]:
```
array([ 2,  4,  6,  8, 10])
```

In [107]:
```
np.sqrt(a)
```

Out[107]:
```
array([1.        , 1.41421356, 1.73205081, 2.        , 2.23606798])
```

In [108]:
```
np.min(a)
```

Out[108]:
```
1
```

In [ ]:
```

```