

In [22]: *#Understanding Basics of Numpy (Gaurav Dev)*

```
import numpy as np
li = [1,2,3,4,5]
a = np.random.randint(1,10,5)
b = np.random.randint(1,8,5)
print(li)
print(a)
print(b)
```

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

In [10]:

```
print(a.max())
print(a.min())
print(a.argmax())
print(a.argmin())
```

```
19
1
1
8
```

In [11]:

```
print(a==b)
```

```
False
```

C:\Users\Gaurav\AppData\Local\Temp\ipykernel_30464\1502103540.py:1: Deprecati
onWarning: elementwise comparison failed; this will raise an error in the fut
ure.
print(a==b)

In [12]:

```
a==b
```

C:\Users\Gaurav\AppData\Local\Temp\ipykernel_30464\464372380.py:1: Deprecatio
nWarning: elementwise comparison failed; this will raise an error in the futu
re.
a==b

Out[12]: False

In [13]:

```
li = [i+1 for i in li]
li
```

Out[13]: [2, 3, 4, 5, 6]

In [14]:

```
print(li)
```

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

```
In [17]: #li = li+5
         #print(li)
         print(a)
```

```
[ 6 20 17  9 12  6  7  3  2 11]
```

```
In [16]: a=a+1
         a
```

```
Out[16]: array([ 6, 20, 17,  9, 12,  6,  7,  3,  2, 11])
```

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

```
In [18]: a>b
```

```
-----
ValueError
Cell In[18], line 1
----> 1 a>b
```

Traceback (most recent call last)

ValueError: operands could not be broadcast together with shapes (10,) (4,)

```
In [19]: a = np.random.randint(1,10,5)
         b = np.random.randint(1,8,5)
         print(a)
         print(b)
```

```
[3 4 1 8 1]
[1 3 3 7 3]
```

```
In [23]: a = np.random.randint(1,10,5)
         b = np.random.randint(1,8,5)
         print(a)
         print(b)
```

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

```
In [24]: c = a+b
         print(c)
```

```
[ 8 12 10  7  9]
```

```
In [25]: a > b
```

```
Out[25]: array([False,  True,  True,  True,  True])
```

```
In [26]: a==b
```

```
Out[26]: array([ True, False, False, False, False])
```

```
In [27]: c=a|b  
print(c)
```

```
[ 4 11  6  7  5]
```

```
In [28]: d = a/b  
print(d)
```

```
[1.          3.          1.5          1.33333333 1.25          ]
```

```
In [29]: d = a//b  
print(d)
```

```
[1 3 1 1 1]
```

```
In [31]: np.arange(10)
```

```
Out[31]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [32]: np.arange(5,15,dtype=float)
```

```
Out[32]: array([ 5.,  6.,  7.,  8.,  9., 10., 11., 12., 13., 14.])
```

```
In [37]: np.arange(4,5,0.2)
```

```
Out[37]: array([4. , 4.2, 4.4, 4.6, 4.8])
```

```
In [30]: # Lets Learn about 2D arrays
```

```
In [38]: np.eye(2)
```

```
Out[38]: array([[1., 0.],  
               [0., 1.]])
```

```
In [40]: np.eye(4,4)
```

```
Out[40]: array([[1., 0., 0., 0.],  
               [0., 1., 0., 0.],  
               [0., 0., 1., 0.],  
               [0., 0., 0., 1.]])
```

```
In [46]: np.vander(np.linspace(0,4,5),2)
```

```
Out[46]: array([[0., 1.],
               [1., 1.],
               [2., 1.],
               [3., 1.],
               [4., 1.]])
```

```
In [47]: np.vander((1,2,3,4),4)
```

```
Out[47]: array([[ 1,  1,  1,  1],
               [ 8,  4,  2,  1],
               [27,  9,  3,  1],
               [64, 16,  4,  1]])
```

```
In [ ]: # 3D arrays
```

```
In [49]: np.zeros((3,4))
```

```
Out[49]: array([[0., 0., 0., 0.],
               [0., 0., 0., 0.],
               [0., 0., 0., 0.]])
```

```
In [50]: np.zeros((2,4,5))
```

```
Out[50]: array([[[0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.]],
                [[0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.]])
```

```
In [ ]: #similarly we can use np.ones function that will create an array filled with 1
```

```
In [51]: np.ones((3,4))
```

```
Out[51]: array([[1., 1., 1., 1.],
               [1., 1., 1., 1.],
               [1., 1., 1., 1.]])
```

```
In [52]: np.ones((2,3,5))
```

```
Out[52]: array([[1., 1., 1., 1., 1.],  
                [1., 1., 1., 1., 1.],  
                [1., 1., 1., 1., 1.]],  
               [[1., 1., 1., 1., 1.],  
                [1., 1., 1., 1., 1.],  
                [1., 1., 1., 1., 1.]])
```