Numpy Crash Cousre--------->30/10/2025

```
In [1]: import numpy as np
```

```
In [2]: np.__version__
```

```
Out[2]: '1.26.4'
```

Creating List

```
In [3]: mylist=[1,2,3,4,5]
        mylist
```

```
Out[3]: [1, 2, 3, 4, 5]
```

```
In [4]: type(mylist)
```

```
Out[4]: list
```

```
In [5]: # List to Array

```

```
In [6]: arr=np.array(mylist)
        arr
```

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

```
In [8]: type(arr)    # n-D array
```

```
Out[8]: numpy.ndarray
```

```
In [9]: #np.tab------> functions of numpy
```

```
In [11]: np.arange(10)  #int range
```

```
Out[11]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [12]: np.arange(5.0) # float range
```

```
Out[12]: array([0., 1., 2., 3., 4.])
```

```
In [13]: np.arange(0,5) # start and end
```

```
Out[13]: array([0, 1, 2, 3, 4])
```

In [15]:
```
np.arange(20,10)  ## 1stArg > 2nd Arg ,so empty array

#MUST  1st Arg < 2nd Arg
```
Out[15]:
```
array([], dtype=int32)
```

In [16]:
```
np.arange(-20,10)
```
Out[16]:
```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
        -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
         6,   7,   8,   9])
```

In [17]:
```
ar=np.arange(-20,10)
ar                           ## Always this should be wriiten  np.arange(-2
```
Out[17]:
```
array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
        -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
         6,   7,   8,   9])
```

In [18]:
```
ar=np.arange()
ar                    ## requires arguments
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [18], in <cell line: 1>()
----> 1 ar=np.arange()
      2 ar

TypeError: arange() requires stop to be specified.
```

In [20]:
```
ar=np.arange(-20,10,5)
ar                                ## step index
```
Out[20]:
```
array([-20, -15, -10,  -5,   0,   5])
```

In [22]:
```
np.arange(-20,10,5,4)       ##  Pass MAX 3 ARG-----> Strat , stop, Step I
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [22], in <cell line: 1>()
----> 1 np.arange(-20,10,5,4)

TypeError: Cannot interpret '4' as a data type
```

In [23]:
```
np.zeros(10)       ### 10 zeros with float data type--------> BCS IT IS P
```
Out[23]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

In [24]:
```
np.zeros(3,dtype=int)     ### HYPERPARAMETER
```
Out[24]:
```
array([0, 0, 0])
```

In [25]:
```
np.zeros((2,2),dtype=int)        ##(2,2) ----->2Rows and 2 colums
```

Out[25]:
```
array([[0, 0],
       [0, 0]])
```

In [26]:
```
zero=np.zeros([3,3])
print(zero)

print("###")
print(type(zero))
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
###
<class 'numpy.ndarray'>
```

In [27]:
```
np.zeros((2,5),dtype=int)     ## 2rows 10 columns
```

Out[27]:
```
array([[0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0]])
```

In [28]:
```
np.zeros((5,5),dtype=int)
```

Out[28]:
```
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]])
```

In [29]:
```
np.ones(3)          ## 3 times 1's are printed
```

Out[29]:
```
array([1., 1., 1.])
```

In [30]:
```
np.ones(3,dtype=int)
```

Out[30]:
```
array([1, 1, 1])
```

In [31]:
```
np.two(3,dtype=int)         ## np Allows 0,1 only ===== Bcs it is IN-Buil
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [31], in <cell line: 1>()
----> 1 np.two(3,dtype=int)

File ~\AppData\Roaming\Python\Python39\site-packages\numpy\__init__.py:362,
in __getattr__(attr)
    359     "Removed in NumPy 1.25.0"
    360     raise RuntimeError("Tester was removed in NumPy 1.25.")
--> 362 raise AttributeError("module {!r} has no attribute "
    363                      "{!r}".format(__name__, attr))

AttributeError: module 'numpy' has no attribute 'two'
```

OTP num are generated multiple times randomly---------rand fun is used

np ----is package

random --- is Modu

rand ---is function

```
In [33]: np.random.rand(5)
```

```
Out[33]: array([0.965608  , 0.63114456, 0.68162199, 0.45487365, 0.32785785])
```

```
In [34]: np.random.rand(5)
```

```
Out[34]: array([0.46533716, 0.98574595, 0.17516561, 0.08394594, 0.23178947])
```

```
In [35]: np.random.rand(3,5)
```

```
Out[35]: array([[4.17455325e-01, 2.04462612e-02, 5.16123846e-01, 7.92416826e-01,
                 3.38844582e-01],
                [6.14989827e-01, 3.32251897e-01, 6.69120518e-04, 5.61597641e-01,
                 6.00571218e-01],
                [4.71845461e-01, 4.98946810e-01, 7.60727841e-01, 2.26866734e-01,
                 5.34391406e-01]])
```

3-rows, 5-colums randomly generated

```
In [36]: np.random.randint(2,5)
```

```
Out[36]: 2
```

2-inclusive and 5- exclusive

2-lower value and 5- Upper value

numbers are generated b/w 2-5

```
In [38]: np.random.rand_int(4,6)           ERROR: "'rand_int'"
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [38], in <cell line: 1>()
----> 1 np.random.rand_int(4,6)

AttributeError: module 'numpy.random' has no attribute 'rand_int'
```

In [39]:
```
np.random.randint(0,9,4)          ## 0-Exclusive
```

Out[39]:
```
array([3, 5, 4, 4])
```

In [40]:
```
np.random.randint(20,10)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [40], in <cell line: 1>()
----> 1 np.random.randint(20,10)

File numpy\\random\\mtrand.pyx:780, in numpy.random.mtrand.RandomState.randi
nt()

File numpy\\random\\_bounded_integers.pyx:2885, in numpy.random._bounded_int
egers._rand_int32()

ValueError: low >= high
```

In [41]:
```
np.random.randint(-20,10)
```

Out[41]:
```
-6
```

In [42]:
```
np.random.randint(-20,10,4)
```

Out[42]:
```
array([  0, -15, -16, -11])
```

In [43]:
```
np.random.randint(10,40,(10,10))       ### GEnerate the element 10-30 wit
```

Out[43]:
```
array([[34, 19, 21, 28, 21, 31, 39, 33, 11, 30],
       [35, 29, 21, 17, 14, 12, 36, 31, 13, 26],
       [33, 29, 18, 19, 29, 34, 28, 13, 39, 21],
       [14, 35, 10, 25, 32, 39, 18, 21, 22, 31],
       [21, 39, 14, 15, 36, 27, 27, 25, 27, 21],
       [35, 10, 25, 34, 15, 35, 22, 36, 37, 32],
       [21, 34, 33, 13, 38, 28, 25, 34, 16, 26],
       [28, 12, 29, 13, 26, 19, 36, 32, 11, 39],
       [29, 27, 15, 27, 34, 38, 36, 30, 27, 23],
       [18, 34, 16, 12, 38, 37, 17, 19, 34, 33]])
```

GEnerate the element 10-30 with 4*4 matrix

Reshaping----as rows and colums

In [44]:
```
np.arange(1,13).reshape(3,4)
```

Out[44]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [45]:
```
np.arange(1,13).reshape(3,5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [45], in <cell line: 1>()
----> 1 np.arange(1,13).reshape(3,5)

ValueError: cannot reshape array of size 12 into shape (3,5)
```

BCS 3*4 = 12

so when we multiply the 3*5 = 15 but thye no-f elements are 12 only i.e---(1,13)

In [47]:
```
np.arange(2,20).reshape(4,6)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [47], in <cell line: 1>()
----> 1 np.arange(2,20).reshape(4,6)

ValueError: cannot reshape array of size 18 into shape (4,6)
```

In [52]:
```
np.arange(2,20).reshape(3,6)
```

Out[52]:
```
array([[ 2,  3,  4,  5,  6,  7],
       [ 8,  9, 10, 11, 12, 13],
       [14, 15, 16, 17, 18, 19]])
```

2-20 there are 18 elements

for reshaping 3*6=18

Slicing in MATRIX

In [3]:
```
b=np.random.randint(10,20,(5,4))
b
```

Out[3]:
```
array([[10, 19, 16, 18],
       [19, 13, 18, 15],
       [17, 16, 17, 10],
       [19, 12, 13, 14],
       [11, 19, 19, 16]])
```

In [4]:
```
b[:]
```
Out[4]:
```
array([[10, 19, 16, 18],
       [19, 13, 18, 15],
       [17, 16, 17, 10],
       [19, 12, 13, 14],
       [11, 19, 19, 16]])
```

In [8]:
```
b[0:]
```
Out[8]:
```
array([[10, 19, 16, 18],
       [19, 13, 18, 15],
       [17, 16, 17, 10],
       [19, 12, 13, 14],
       [11, 19, 19, 16]])
```

In [16]:
```
b[-1]
b[-2]
b[-3]
print(b[-1])
print(b[-2])
print(b[-3])
```
```
[11 19 19 16]
[19 12 13 14]
[17 16 17 10]
```

In [18]:
```
b[1:3]
```
Out[18]:
```
array([[19, 13, 18, 15],
       [17, 16, 17, 10]])
```

In [19]:
```
b[1,3]
```
Out[19]:
```
15
```

Slicing":" will pasre row or column

,----> represent the row and the column

In [20]:
```
b
```
Out[20]:
```
array([[10, 19, 16, 18],
       [19, 13, 18, 15],
       [17, 16, 17, 10],
       [19, 12, 13, 14],
       [11, 19, 19, 16]])
```

In [22]:
```
b[0,3]   ##  [0,3] ----->> 0th row and the 3rd colum 1st eelemnt will be
```
Out[22]:
```
18
```

In [23]:
```
b[0:-2]
```

Out[23]:
```
array([[10, 19, 16, 18],
       [19, 13, 18, 15],
       [17, 16, 17, 10]])
```

In [24]:
```
b[0:5:3]
```

Out[24]:
```
array([[10, 19, 16, 18],
       [19, 12, 13, 14]])
```

No of fun in numpy are 217

**** ---means import al the fun which are in numpy***

In [ ]:
```

```