Data Science & AI
&
NIC - Param

Python-For Data Science

Numpy

Lecture No.- 02

By- Pankaj Sharma Sir

Recap of Previous Lecture

Topic NumPy Part 01

np.zeros()
np.ones()
np.full()
np.empty()
np.array()
np.arange()
np.linspace()

np.identity()
np.eye()

Topics to be Covered

Topic NumPy Part 02

slicing, indexing,

boolean indexing

→ imp ✓

Topic : NumPy

✓

$$\begin{array}{l} \quad\;\; 0 \quad\; 1 \quad\; 2 \quad\; 3 \quad\; 4 \\ l = [\boxed{10, 20, 30}, 40, 50] \end{array}$$

n_arr = np.array(l ,dtype = int)

$\left\{ \begin{array}{l} \text{l[0]} \\ \text{n\_arr[0]} \end{array} \right\}$ same $\left\{ \begin{array}{l} \text{print(l [0 : 3])} \\ \text{print(n\_arr [0 : 3])} \end{array} \right.$ → some result l1[2:4][1]

l1[2:4][4]

Error

2, 3

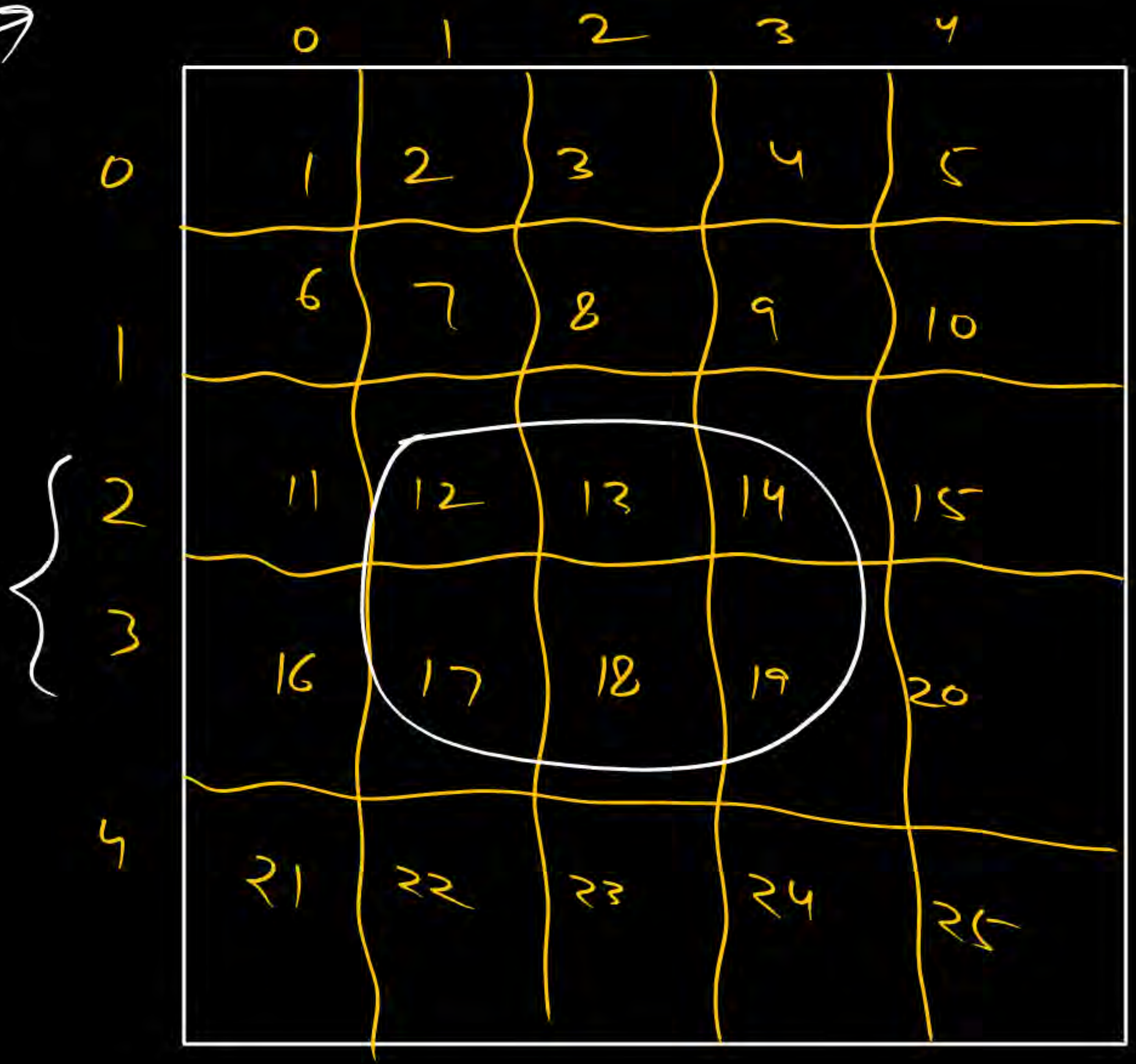
$$\left[\begin{array}{l}\overset{0}{[11, 12, 13, 14, 15]},\\ \underset{1}{[16, 17, 18, 19, 20]}\end{array}\right]$$

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

numpy

Rahul sir

narr1[2:4, 1:4]



	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

numpy → 3   read this

file handling

```
file_object = Open( filename, 'r' )
                               'w'
                               'a'

s = file-object.read()

list_of_lines = file-object.readlines()
```

Numpy ✓

file handling ✓

Pandas → 3-4 lectures

```
In [1]: import numpy as np
        l=[10,20,30,40,50,60]
        narr=np.array(l,dtype=int)
```

```
In [2]: print(l)
```

```
[10, 20, 30, 40, 50, 60]
```

```
In [3]: print(narr)
```

```
[10 20 30 40 50 60]
```

```
In [4]: #collection of pointers/refrences
        #data ===>first element
        #shape
        #dtype
        #strides
```

```
In [5]: print(narr.shape)
```

```
(6,)
```

```
In [6]: print(narr.data)
        print(narr.shape)#1D ==>6 * 1 ===> (6,)
        print(narr.dtype)#4 bytes
        print(narr.strides)# ==>how many bytes to cross to move from 1 element to other
```

```
<memory at 0x0000024AB8CE6680>
(6,)
int32
(4,)
```

```
In [7]: l1=[[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15],[16,17,18,19,20],[21,22,23,24,25]]
        narr1=np.array(l1,dtype=int)
```

```
In [8]: print(l1)
        print(narr1)
```

```
[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15], [16, 17, 18, 19, 20], [21,
22, 23, 24, 25]]
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]
 [21 22 23 24 25]]
```

```
In [9]: print(narr1.data)
        print(narr1.shape)#2D array ==>5 rows and 5 cols
        print(narr1.dtype)
        print(narr.strides)
```

```
<memory at 0x0000024AB916F920>
(5, 5)
int32
(4,)
```

```
In [11]: l1[1][3]  #===>2nd row ka 4th element(col)
```

Out[11]:
```
9
```

In [12]:
```
narr1[1][3]
```

Out[12]:
```
9
```

In [13]:
```
narr1[1,3] #same as narr1[1][3]
```

Out[13]:
```
9
```

In [14]:
```
l1[1][1:4]  #2nd,3rd,4th col of row index 1(2nd row)
```

Out[14]:
```
[7, 8, 9]
```

In [17]:
```
print(narr1[1][1:4])
```

```
[7 8 9]
```

In [20]:
```
l1[2:4][0]
```

Out[20]:
```
[11, 12, 13, 14, 15]
```

In [22]:
```
print(narr1[2:4,4])#it will work
```

```
[15 20]
```

In [23]:
```
narr1[2:4,1:4]
```

Out[23]:
```
array([[12, 13, 14],
       [17, 18, 19]])
```

In [24]:
```
#operations
```

In [25]:
```
l2=[10,20,30,40,50,60]
```

In [26]:
```
x=np.random.randint(1,30,6)
y=np.random.randint(1,30,6)
print(l2)
print(x)
print(y)
```

```
[10, 20, 30, 40, 50, 60]
[26  4 21 26 22 17]
[ 5 11  3 20 24 23]
```

In [27]:
```
#doubling each element of list
l2=[2*i for i in l2]
l2
```

Out[27]:
```
[20, 40, 60, 80, 100, 120]
```

In [28]:
```
x=x*2
x
```

Out[28]:
```
array([52,  8, 42, 52, 44, 34])
```

```
In [29]: print(x)
         print(y)

         [52  8 42 52 44 34]
         [ 5 11  3 20 24 23]
```

```
In [30]: z=x+y #it will work fine
```

```
In [31]: z

Out[31]: array([57, 19, 45, 72, 68, 57])
```

```
In [32]: n1=x+y
         n2=x-y
         n3=x*y
```

```
In [33]: n1

Out[33]: array([57, 19, 45, 72, 68, 57])
```

```
In [34]: n2

Out[34]: array([47, -3, 39, 32, 20, 11])
```

```
In [35]: n3

Out[35]: array([ 260,   88,  126, 1040, 1056,  782])
```

```
In [36]: n4=x/y
```

```
In [37]: n4

Out[37]: array([10.4       ,  0.72727273, 14.        ,  2.6       ,  1.83333333,
                 1.47826087])
```

```
In [38]: n5=x**y
```

```
In [39]: n5

Out[39]: array([380204032,         0,     74088,         0,         0, 947912704])
```

```
In [40]: x=np.array([2,3,4,5,6],dtype=int)
         y=np.array([1,2,3,4,5],dtype=int)
         x**y

Out[40]: array([   2,    9,   64,  625, 7776])
```

```
In [41]: x

Out[41]: array([2, 3, 4, 5, 6])
```

```
In [42]: x.sum()

Out[42]: 20
```

```
In [43]:  x.mean()

Out[43]:  4.0

In [44]:  x.min()

Out[44]:  2

In [45]:  x.max()

Out[45]:  6

In [46]:  x.argmin() #===>index of min element

Out[46]:  0

In [47]:  x.argmax()

Out[47]:  4

In [48]:  #narray ===>relational

In [49]:  x=np.random.randint(1,30,6)
          y=np.random.randint(1,30,6)

In [50]:  print(x)

          [ 3  6 18 19 18 14]

In [51]:  print(y)

          [12  9 21  6 27 22]

In [52]:  x>y #[3>12  6>9  18>21  19>6   18>27   14>22]

Out[52]:  array([False, False, False,  True, False, False])

In [53]:  x<y

Out[53]:  array([ True,  True,  True, False,  True,  True])

In [54]:  x==y

Out[54]:  array([False, False, False, False, False, False])

In [55]:  x!=y

Out[55]:  array([ True,  True,  True,  True,  True,  True])

In [56]:  #logical operator #np.logical_or   np.logical_and    np.logical_not
          x=np.array([12,3,44,0,56,90],dtype=int)
          y=np.array([1,2,3,4,0,23],dtype=int)
```

In [57]:
```
print(x)
print(y)
```
```
[12  3 44  0 56 90]
[ 1  2  3  4  0 23]
```

In [58]:
```
print(np.logical_or(x,y))
```
```
[ True  True  True  True  True  True]
```

In [59]:
```
print(np.logical_and(x,y))
```
```
[ True  True  True False False  True]
```

In [60]:
```
print(np.logical_not(x))
```
```
[False False False  True False False]
```

In [61]:
```
y
```
Out[61]:
```
array([ 1,  2,  3,  4,  0, 23])
```

In [62]:
```
print(np.logical_not(y))
```
```
[False False False False  True False]
```

In [63]:
```
#boolean indexing
x=np.array([1,23,4,56,7,89,900],dtype=int)
y=np.random.randint(1,100,7)
```

In [64]:
```
print(x)
print(y)
```
```
[  1  23   4  56   7  89 900]
[26 30 35 42 62 58 43]
```

In [65]:
```
print(y>30) #boolean array ayega ===>[26>30    30>30   35>30 .......]
```
```
[False False  True  True  True  True  True]
```

In [66]:
```
boolean_array=y>30
```

In [67]:
```
print(boolean_array)
```
```
[False False  True  True  True  True  True]
```

In [68]:
```
arr1=y[boolean_array]
```

In [69]:
```
arr1
```
Out[69]:
```
array([35, 42, 62, 58, 43])
```

In [70]:
```
arr2=y[y>30]#y[condition]
```

In [71]:
```
arr2
```
Out[71]:
```
array([35, 42, 62, 58, 43])
```

```
In [75]: arr3=y[ (y>30) &  (y<50)]
```

```
In [76]: arr3

Out[76]: array([35, 42, 43])
```

```
In [77]: y

Out[77]: array([26, 30, 35, 42, 62, 58, 43])
```

```
In [78]: x=y
```

```
In [79]: print(x)

[26 30 35 42 62 58 43]
```

```
In [80]: print(y)

[26 30 35 42 62 58 43]
```

```
In [81]: x[0]=126 #1 element got change
```

```
In [82]: x

Out[82]: array([126,  30,  35,  42,  62,  58,  43])
```

```
In [83]: x[0:4] #index 0,1,2,3 element 1st,2nd,3rd,4th

Out[83]: array([126,  30,  35,  42])
```

```
In [84]: x[0:4]=100
```

```
In [85]: x

Out[85]: array([100, 100, 100, 100,  62,  58,  43])
```

```
In [86]: print(x)

[100 100 100 100  62  58  43]
```

```
In [87]: x[x>70]=20
```

```
In [88]: x

Out[88]: array([20, 20, 20, 20, 62, 58, 43])
```

```
In [89]: x==20

Out[89]: array([ True,  True,  True,  True, False, False, False])
```

```
In [90]: indexes=np.where(x==20)
```

```
In [91]: indexes
```

```
Out[91]: (array([0, 1, 2, 3], dtype=int64),)
```

```
In [92]: x=np.random.randint(1,50,(4,5))#4rows 5 cols
```

```
In [93]: print(x)
```

```
[[30 38 25 45 45]
 [ 8 42 40 42 49]
 [30 18 27 13 23]
 [34 28 49  2 11]]
```

```
In [94]: x>25
```

```
Out[94]: array([[ True,  True, False,  True,  True],
                [False,  True,  True,  True,  True],
                [ True, False,  True, False, False],
                [ True,  True,  True, False, False]])
```

```
In [95]: boolean_arr=x>25
```

```
In [96]: boolean_arr
```

```
Out[96]: array([[ True,  True, False,  True,  True],
                [False,  True,  True,  True,  True],
                [ True, False,  True, False, False],
                [ True,  True,  True, False, False]])
```

```
In [97]: print(x[boolean_arr])
```

```
[30 38 45 45 42 40 42 49 30 27 34 28 49]
```

```
In [98]: y=x
```

```
In [99]: x
```

```
Out[99]: array([[30, 38, 25, 45, 45],
                [ 8, 42, 40, 42, 49],
                [30, 18, 27, 13, 23],
                [34, 28, 49,  2, 11]])
```

```
In [100…: y
```

```
Out[100]: array([[30, 38, 25, 45, 45],
                 [ 8, 42, 40, 42, 49],
                 [30, 18, 27, 13, 23],
                 [34, 28, 49,  2, 11]])
```

```
In [101…: boolean_arr
```

```
Out[101]: array([[ True,  True, False,  True,  True],
                 [False,  True,  True,  True,  True],
                 [ True, False,  True, False, False],
                 [ True,  True,  True, False, False]])
```

```
In [102…: #size is same for boolean_arr ,x and y
```

```
In [103…: x[boolean_arr]=0
```

```
In [104…: x
```

```
Out[104]: array([[ 0,  0, 25,  0,  0],
                 [ 8,  0,  0,  0,  0],
                 [ 0, 18,  0, 13, 23],
                 [ 0,  0,  0,  2, 11]])
```

```
In [105…  x=np.random.randint(10,20,(3,3))
```

```
In [106…  y=np.random.randint(10,20,(2,3))
```

```
In [107…  x
```

```
Out[107]: array([[18, 14, 13],
                 [18, 10, 18],
                 [14, 17, 10]])
```

```
In [108…  y
```

```
Out[108]: array([[10, 12, 11],
                 [11, 13, 10]])
```

```
In [109…  boolean_arr=np.array([[True,False],[False,True]])
```

```
In [110…  boolean_arr
```

```
Out[110]: array([[ True, False],
                 [False,  True]])
```

```
In [111…  y[boolean_arr] #y===>         (2,3)
                         #boolean_arr==>(2,2)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[111], line 1
----> 1 y[boolean_arr]

IndexError: boolean index did not match indexed array along dimension 1; dimension is
3 but corresponding boolean dimension is 2
```

```
In [112…  x[boolean_arr]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[112], line 1
----> 1 x[boolean_arr]

IndexError: boolean index did not match indexed array along dimension 0; dimension is
3 but corresponding boolean dimension is 2
```

```
In [ ]:
```

THANK - YOU