Data Science & AI
&
NIC - Param

Python-For Data Science

Numpy

Lecture No.- 03

By- Pankaj Sharma Sir

Recap of Previous Lecture

Topic NumPy Part 02

Topics to be Covered

Topic NumPy Part 03

Topic : NumPy

np.reshape()
np.resize()

broadcasting

x → (4,4)
y → (4,4)

x + y
x - y
x / y
x // y
x ** y

All are valid

$x \rightarrow (3,2)$
$y \rightarrow (2,3)$
} × { different Shapes }

$x+y$
$x-y$
} Invalid

① Equal

② if one dim in an array is 1

$x - (3,1)$
$y - (3,4)$
} =

(4,4)

4 ⇒ (1,4)

broadcasting

Rule 2

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

+

x	y	u	v
x	y	u	v
x	y	u	v
x	y	u	v

$(3,1)$

x

1	1	1
2	2	2
3	3	3

$+$

$(3,3)$

2	1	3
3	2	2
3	1	2

(1, 3)

1	2	3
1	2	3
1	2	3

+

(3, 1)

1	1	1
2	2	2
3	3	3

data manipulation

CSV (comma sep. values)

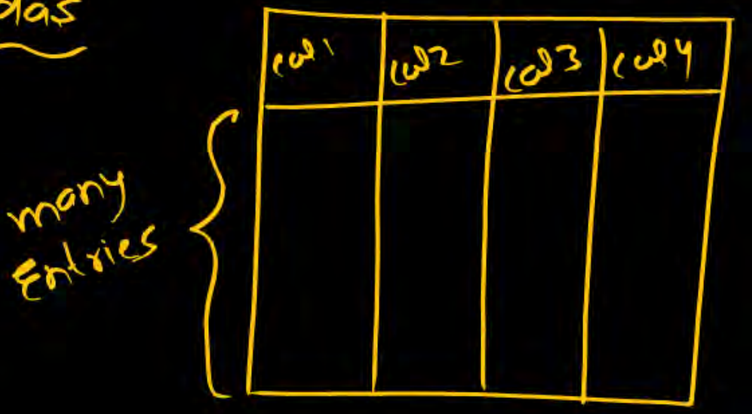
JSON

HTML

⋮

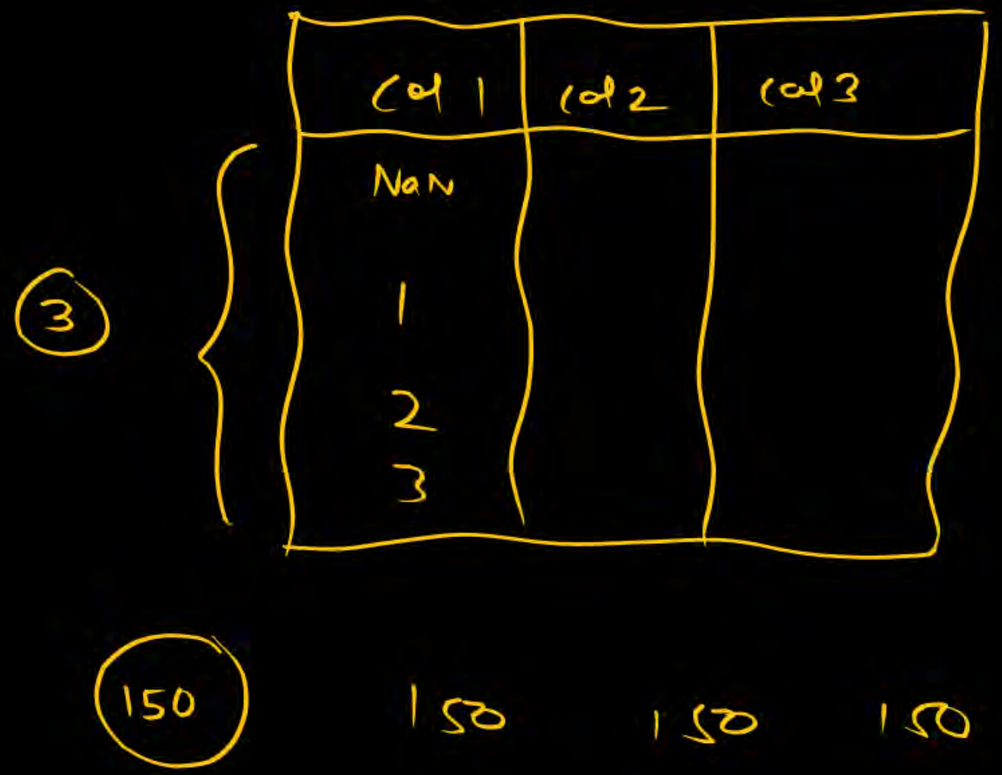
Text ⟶ Open(—— , 'r' 'w' 'a')

↳ Pandas

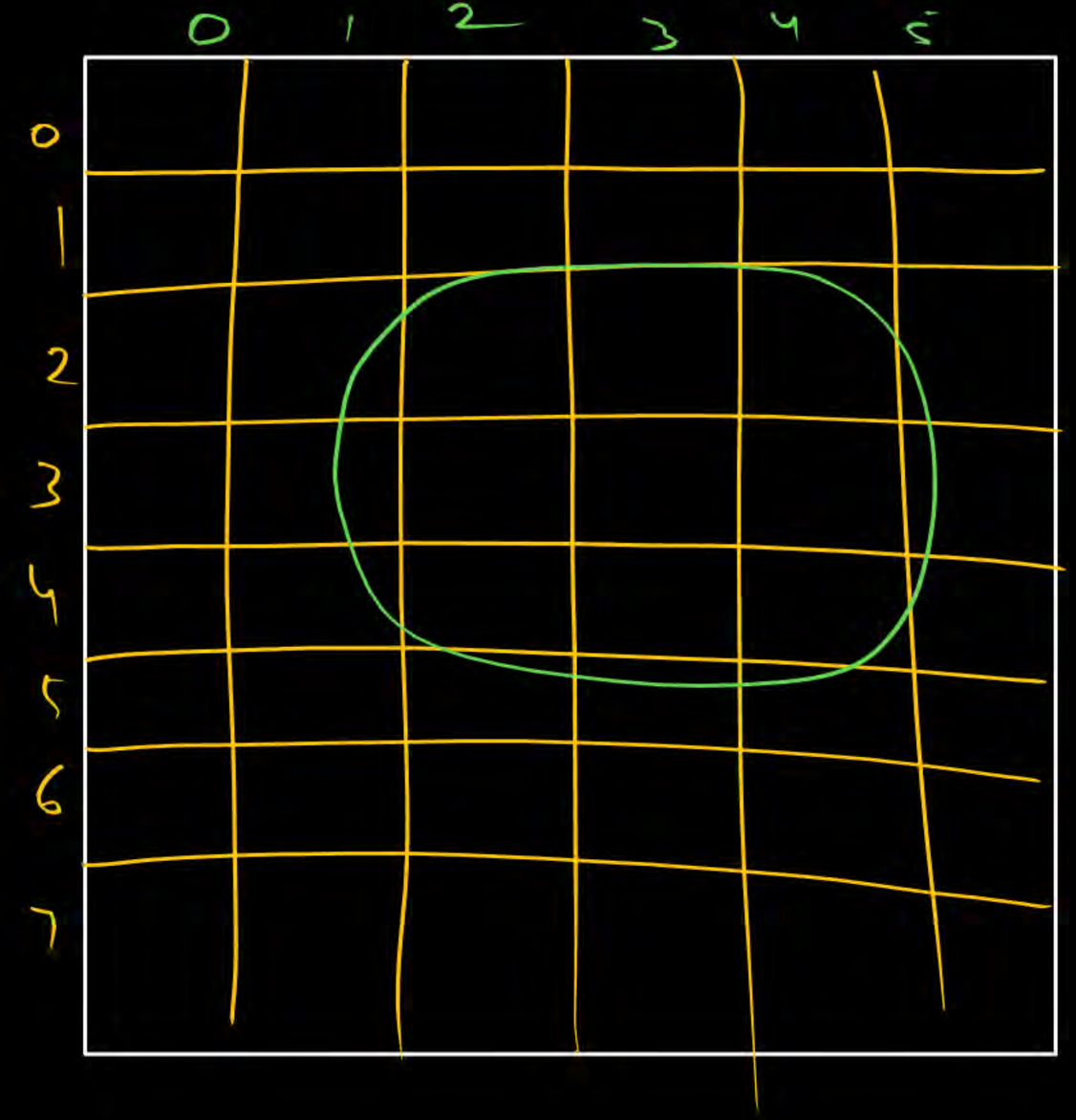


```
import Pandas

Pandas.read_csv(   )
```

```
import Pandas as Pd

obj = Pd.read_csv(_)
      .read_json(_)
      .read_html(_)
```


col 1
col 2
col 3
NaN
1
2
3
3
150
150
150
150

df.iloc[2:5, 2:5]



① dataframe

② series

pd · read_csv
· read_json
· read_html

HW

pipe seperated values

03/04/05

Numpy

1. read_csv(filename)
2. df = read_csv(filename)
3. df.describe()
4. df.shape
   df.isnull()
   df.isnull().sum()

df.head() → 5 rows

header →

t.me/PWpankajsirP

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

In [2]:
```
x=np.random.randint(10,20,(4,4))
y=np.random.randint(10,20,(4,4))
```

In [3]:
```
print(x)
print(y)
```

```
[[10 10 12 13]
 [17 11 11 12]
 [11 14 12 14]
 [14 15 19 14]]
[[15 10 12 16]
 [19 18 19 15]
 [13 19 17 12]
 [15 12 18 16]]
```

In [4]:
```
x+y
```

Out[4]:
```
array([[25, 20, 24, 29],
       [36, 29, 30, 27],
       [24, 33, 29, 26],
       [29, 27, 37, 30]])
```

In [5]:
```
x-y
```

Out[5]:
```
array([[-5,  0,  0, -3],
       [-2, -7, -8, -3],
       [-2, -5, -5,  2],
       [-1,  3,  1, -2]])
```

In [6]:
```
x*y
```

Out[6]:
```
array([[150, 100, 144, 208],
       [323, 198, 209, 180],
       [143, 266, 204, 168],
       [210, 180, 342, 224]])
```

In [7]:
```
x/y
```

Out[7]:
```
array([[0.66666667, 1.        , 1.        , 0.8125    ],
       [0.89473684, 0.61111111, 0.57894737, 0.8       ],
       [0.84615385, 0.73684211, 0.70588235, 1.16666667],
       [0.93333333, 1.25      , 1.05555556, 0.875     ]])
```

In [8]:
```
x//y
```

Out[8]:
```
array([[0, 1, 1, 0],
       [0, 0, 0, 0],
       [0, 0, 0, 1],
       [0, 1, 1, 0]])
```

In [9]:
```
#same shape hai ==>no problem at all
```

In [10]:
```
x=np.random.randint(10,20,(4,4))
y=np.random.randint(10,20,(4))
```

In [11]:
```
print(x)
print(y)
```

```
[[17 19 13 14]
 [11 15 11 12]
 [19 13 18 18]
 [17 16 12 16]]
[10 19 12 17]
```

In [12]:
```
x+y
```

Out[12]:
```
array([[27, 38, 25, 31],
       [21, 34, 23, 29],
       [29, 32, 30, 35],
       [27, 35, 24, 33]])
```

In [13]:
```
a=np.random.randint(10,20,(3,4))
b=np.random.randint(10,20,(4,3))
```

In [14]:
```
print(a)
print(b)
```

```
[[19 10 17 15]
 [16 12 19 14]
 [10 12 14 14]]
[[12 15 10]
 [11 17 11]
 [17 16 13]
 [10 13 11]]
```

In [15]:
```
a+b
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[15], line 1
----> 1 a+b

ValueError: operands could not be broadcast together with shapes (3,4) (4,3)
```

In [16]:
```
a=np.random.randint(10,20,(3,4))
b=np.random.randint(10,20,(4,3))
```

In [19]:
```
b=b.transpose()
```

In [20]:
```
print(b)
```

```
[[13 19 11 12]
 [11 17 13 16]
 [13 12 12 14]]
```

In [21]:
```
a+b
```

Out[21]:
```
array([[23, 35, 24, 31],
       [21, 35, 25, 27],
       [31, 30, 27, 26]])
```

In [22]:
```
x=np.array([[1],[2],[3]])
```

In [24]:
```
x.shape
```

Out[24]:
```
(3, 1)
```

In [25]:
```
print(x)
```

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

In [26]:
```
y=np.random.randint(1,5,(3,3))
```

In [27]:
```
print(y)
```

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

In [28]:
```
#x ===>(3,1)
#y===>(3,3)
#x+y  ===>valid/invalid broadcasting
```

In [29]:
```
x+y
```

Out[29]:
```
array([[3, 2, 4],
       [5, 4, 4],
       [6, 4, 5]])
```

In [30]:
```
a=np.array([1,2,3])
b=np.array([1])
```

In [31]:
```
a.shape  #(1,3)
```

Out[31]:
```
(3,)
```

In [32]:
```
b.shape #(1,1)
```

Out[32]:
```
(1,)
```

In [33]:
```
#broadcasting  ===> [1 1 1]  new b array
```

In [34]:
```
a+b
```

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

In [35]:
```
x=np.array([1,2,3])
```

In [36]:
```
y=np.array([[1],[2],[3]])
```

In [37]:
```
print(x)
print(y)
```

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

In [38]:
```
#x ====>(1,3)
#y ====>(3,1)
#broadcasting applicable or not
```

In [39]:
```
x +y
```

Out[39]:
```
array([[2, 3, 4],
       [3, 4, 5],
       [4, 5, 6]])
```

In [40]:
```
#(3,1,1)  (1,4,5)  ===>valid bcz of concept of broadcasting
```

In [41]:
```
import pandas as pd
```

In [43]:
```
iris=pd.read_csv('Desktop\petals.csv')
```

In [44]:
```
print(type(iris))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [50]:
```
df=iris
```

In [51]:
```
print(df)
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

In [52]:
```
df.shape
```

Out[52]:
```
(150, 5)
```

In [55]:
```
df.head(2)
```

Out[55]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa

In [56]:
```
df.columns
```

Out[56]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [57]:
```
df.columns=['sl','sw','pl','pw','kind']
```

In [58]:
```
df
```

Out[58]:

	sl	sw	pl	pw	kind
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [60]:
```
df.describe()
```

Out[60]:

	sl	sw	pl	pw
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [61]:
```
df.isnull()
```

Out[61]:

	sl	sw	pl	pw	kind
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [62]:
```
df.isnull().sum()
```

Out[62]:
```
sl      0
sw      0
pl      0
pw      0
kind    0
dtype: int64
```

In [63]:
```
df.sl
```

Out[63]:
```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: sl, Length: 150, dtype: float64
```

In [64]:
```
df["sl"]
```

Out[64]:
```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: sl, Length: 150, dtype: float64
```

In [65]:
```
#slicing
df.iloc[1:3,1:3]
```

Out[65]:

	sw	pl
1	3.0	1.4
2	3.2	1.3

In [70]:
```
df.iloc[1:5,1:4]
```

Out[70]:

	sw	pl	pw
1	3.0	1.4	0.2
2	3.2	1.3	0.2
3	3.1	1.5	0.2
4	3.6	1.4	0.2

In [71]:
```
df1=pd.read_csv('Desktop\petals.csv')
```

In [72]:
```
df1
```

Out[72]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	Nan	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [76]:
```
df1.isnull().sum()
```

Out[76]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [ ]:

THANK - YOU