

Day - 2

Numpy &

Pandas



CONTENT

1. Numpy

- a. Basic Function of numpy
- b. Random Number in numpy
- c. Slicing in numpy n - dim arrays!
- d. Reshaping the Array
- e. Sorting an array
- f. Data Loading using numpy

2. Pandas

- a. Series Data Types in Pandas
- b. DataFrames Data Types in Pandas

3.

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.



NUMPY

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.



NUMPY Array

Basic Functions of Numpy

```
[2] array1 = np.array([1,2,3,4,5])
```

```
[3] type(array1)
```

```
↳ numpy.ndarray
```

```
[4] array2 = np.array([1,2,3,4,5])
```

```
[5] array2
```

```
↳ array([1, 2, 3, 4, 5])
```

```
[6] array1
```

```
↳ array([1, 2, 3, 4, 5])
```

```
[7] array1 + array2
```

```
↳ array([ 2,  4,  6,  8, 10])
```

Basic Functions of Numpy

```
[18] np.zeros((3,3))
```

```
↳ array([[0., 0., 0.],  
         [0., 0., 0.],  
         [0., 0., 0.]])
```

```
[19] np.ones((3,3))
```

```
↳ array([[1., 1., 1.],  
         [1., 1., 1.],  
         [1., 1., 1.]])
```

```
[20] c = np.full((3,3), 1)
```

```
[21] c.dtype
```

```
↳ dtype('int64')
```


Basic Functions of Numpy

```
[23] np.add(a, b)
```

```
↳ array([[2, 4, 6],  
        [5, 7, 9]])
```

```
[24] np.subtract(a,b)
```

```
↳ array([[ 0,  0,  0],  
        [-3, -3, -3]])
```

```
[25] np.multiply(a,b)
```

```
↳ array([[ 1,  4,  9],  
        [ 4, 10, 18]])
```

```
[26] np.divide(a,b)
```

```
↳ array([[1. , 1. , 1. ],  
        [0.25, 0.4 , 0.5 ]])
```

Basic Functions of Numpy

```
[27] np.eye(5, 5)
```

```
↳ array([[1., 0., 0., 0., 0.],  
         [0., 1., 0., 0., 0.],  
         [0., 0., 1., 0., 0.],  
         [0., 0., 0., 1., 0.],  
         [0., 0., 0., 0., 1.]])
```

```
[28] np.arange(1, 10, 5)
```

```
↳ array([1, 6])
```

```
[29] example_list = []
```

```
    for i in range(100000):  
        example_list.append(i)
```

```
[30] numpy_example_list = np.arange(0,100000)
```

```
[31] type(numpy_example_list), type(example_list)
```

```
↳ (numpy.ndarray, list)
```


Random Functions in Numpy

```
[38] np.random.seed(1)  
      np.random.randint(1,15, size = (5,5))
```

```
↳ array([[ 6, 12, 13,  9, 10],  
         [12,  6,  1,  1,  2],  
         [13,  8, 14, 13,  7],  
         [10,  3,  5,  6,  3],  
         [ 5, 12, 13, 11, 13]])
```

```
[39] np.random.seed(5)  
      np.random.random(size=(5,5))
```

```
↳ array([[0.22199317, 0.87073231, 0.20671916, 0.91861091, 0.48841119],  
         [0.61174386, 0.76590786, 0.51841799, 0.2968005 , 0.18772123],  
         [0.08074127, 0.7384403 , 0.44130922, 0.15830987, 0.87993703],  
         [0.27408646, 0.41423502, 0.29607993, 0.62878791, 0.57983781],  
         [0.5999292 , 0.26581912, 0.28468588, 0.25358821, 0.32756395]])
```

Slicing in Numpy

Slicing: Similar to Python lists, numpy arrays can be sliced. Since arrays may be multidimensional, you must specify a slice for each dimension of the array:

```
[40] ndim_array_example = np.random.randint(1,15, size = (3,3))
```

```
[41] ndim_array_example
```

```
↳ array([[ 5,  5, 10],  
         [ 4, 12,  3],  
         [ 5,  7, 10]])
```

```
[42] ndim_array_example[ 2:, 1:2 ]
```

```
↳ array([[7]])
```

```
[43] ndim_array_example[ 2:3, 1: ]
```

```
↳ array([[ 7, 10]])
```

```
[44] ndim_array_example[1:2, 2:3] + ndim_array_example[1:2, 0:1]
```

```
↳ array([[7]])
```

Reshaping the Numpy Array

[45] a

```
↳ array([1, 2, 3])
```

[46] a.shape

```
↳ (3,)
```

[47] a = a.reshape((1,3))

[48] a.shape

```
↳ (1, 3)
```

[49] a

```
↳ array([[1, 2, 3]])
```

[50] a, b

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

Reshaping the Numpy Array

```
[51] a * b
```

```
↳ array([[ 1,  4,  9],  
         [ 4, 10, 18]])
```

```
[52] a1 = np.random.randint(1, 3, size = (2,2))
```

```
[53] a1
```

```
↳ array([[2, 2],  
         [1, 2]])
```

```
[54] a = 5  
     b = 6
```

```
[55] a, b
```

```
↳ (5, 6)
```

```
[56] a, b = b, a
```

Reshaping the Numpy Array

```
[ ] a1, b1 = np.random.randint(1, 3, size = (2,5)),  
        np.random.randint(1, 3, size = (2,5))
```

```
[ ] a1, b1
```

```
↳ (array([[2, 2, 1, 1, 2],  
          [1, 1, 2, 2, 2]]), array([[1, 2, 2, 2, 2],  
          [1, 2, 2, 2, 2]]))
```

```
[ ] a1.dot(b1)
```

```
↳ -----  
ValueError                                Traceback (most recent call last)  
<ipython-input-60-37f3a801ac37> in <module>()  
----> 1 a1.dot(b1)
```

ValueError: shapes (2,5) and (2,5) not aligned: 5 (dim 1) \neq 2 (dim 0)

SEARCH STACK OVERFLOW

Reshaping the Numpy Array

```
[ ] a1.shape
```

```
↳ (2, 5)
```

```
[ ] b1.shape
```

```
↳ (2, 5)
```

```
[ ] a1.dot(b1.reshape((5,2)))
```

```
↳ array([[14, 15],  
         [15, 14]])
```

```
[ ] a1.dot(b1.T)
```

```
↳ array([[14, 14],  
         [15, 15]])
```

```
[ ] b1.T
```

```
↳ array([[1, 1],  
         [2, 2],  
         [2, 2],  
         [2, 2],  
         [2, 2]])
```


Sorting an array in Numpy

```
[ ] a1.sort()
```

```
[ ] a1
```

```
↳ array([[1, 1, 2, 2, 2],  
         [1, 1, 2, 2, 2]])
```

Data Loading using Numpy

```
data = np.loadtxt("rp.txt", dtype = "str", delimiter=",")
```

```
data
```

```
array(['Generally',  
      ' solution to Travelling Salesman Problem(TSP) using any algorithm results only in a singl  
e path which is optimal path according the algorithm. This can be sometimes not the best path for  
the user',  
      ' because it is possible that the user wants to visit any other city first irrespective of  
some increase in cost for the travelling route. So',  
      ' the proposed algorithm for the TSP will display all the possible paths in a given scenar  
io with their respective costs using dynamic programming. Based on the cost and user preference',  
      ' a user can select any of the path displayed.'], dtype='<U195')
```

Pandas is a fast, powerful, flexible and
easy to use open source data analysis
and manipulation tool



PANDAS

Pandas

Series Data Types in Pandas

```
example_pandas_series = pd.Series(["Data", "Science", "Training"])
```

```
type(example_pandas_series)
```

```
pandas.core.series.Series
```

```
example_pandas_series2 = pd.Series(["Data", "Science", "Training"], index=["D", "S", "T"])
```

```
example_pandas_series2
```

```
D      Data  
S      Science  
T      Training  
dtype: object
```

Series data type in Pandas continues...

```
type(example_pandas_series2)
```

```
pandas.core.series.Series
```

```
colours = pd.Series(["Red", "Green", "Yellow"])
```

```
cars = pd.Series(["Lambo", "Audi", "BMW"])
```

Data Frames in Pandas

```
example_dataframe = pd.DataFrame([1,2,3,4,5], index=["1","2","3","4","5"], columns = ["Values"])
```

example_dataframe

	Values
1	1
2	2
3	3
4	4
5	5

```
example_dataframe2 = pd.DataFrame({"colour":colours, "cars":cars})
```

example_dataframe2

	colour	cars
0	Red	Lambo
1	Green	Audi
2	Yellow	BMW

WHAT SETS US APART?

Despite being red, Mars is actually a cold place. It's full of iron oxide dust



MARS

Neptune is the fourth-largest planet in our Solar System



NEPTUNE

Venus has a beautiful name and is the second planet from the Sun



VENUS

NAME OF YOUR SECTION

01

You could enter a subtitle
here if you need it

WHAT DO WE DO?



Mercury is the smallest planet in our Solar System



Venus has a beautiful name, but it's terribly hot



Jupiter is the biggest planet in our Solar System



Saturn is composed of hydrogen and helium

HOW DO WE DO IT?

MERCURY

Mercury is the smallest planet in our Solar System

VENUS

Venus has a beautiful name, but it's terribly hot

SATURN

Saturn is composed of hydrogen and helium



WHAT DO WE OFFER?

01	02	03	04
Mercury is the smallest planet in our Solar System	Jupiter is the biggest planet in our Solar System	Venus has a beautiful name, but it's terribly hot	Saturn is composed of hydrogen and helium

OUR CLIENTS

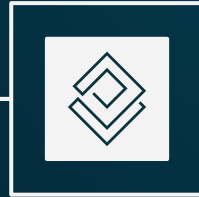
Mercury is the smallest planet in our Solar System

MERCURY



JUPITER

Jupiter is the biggest planet in our Solar System



Venus has a beautiful name, but it's terribly hot

VENUS

SATURN

Saturn is composed of hydrogen and helium



WHAT DO THEY SAY ABOUT US?

Despite being red, Mars is actually a cold place

MARS

Venus has a beautiful name, but it's terribly hot

VENUS

Mercury is the closest planet to the Sun

MERCURY

Neptune is the farthest planet from the Sun

NEPTUNE

Saturn is composed of hydrogen and helium

SATURN

Jupiter is the biggest planet in our Solar System

JUPITER

CASE STUDIES

Venus is the
second planet
from the Sun

VENUS

Saturn is the
ringed planet and
a gas giant

SATURN



MARS

Despite being red,
Mars is actually a
cold place




MERCURY

Mercury is the
closest planet to
the Sun



NEPTUNE

Neptune is the
farthest planet
from the Sun



**AWESOME
WORDS**

OUR PROGRESS



NEPTUNE

Neptune is the farthest planet from the Sun



VENUS

Venus is the second planet from the Sun



MERCURY

Mercury is the closest planet to the Sun



MARS

Despite being red, Mars is actually a cold place

MEET THE TEAM



JENNA DOE

You can replace the image on the screen with your own



JAMES PATTERSON

You can replace the image on the screen with your own

SOFTWARE DESKTOP



Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than our Moon

THANKS!

Do you have any questions?

youremail@freepik.com

+91 620 421 838

yourcompany.com



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.