

# PYTHON PANDAS

classmate

Date

Page

Pandas: (Panel Data System): is an open source library. PANDAS offers easy to use data structures and data Analysis tools.

NumPy Array: is an open source module of Python that offers functions and routines for fast mathematical computations on array and matrices.

NumPy Array comes in two forms:

- Vectors &
- Matrices.

Vectors have single row.

Matrices " multiple row/column.

NumPy Data Types: NumPy array can have elements in data type supported by NumPy. few are

<u>Data Type</u>	<u>Size (Byte)</u>
int16	2B
int64	8B
float64	"
string	1B per character
Unicode	no. of Bytes Platform specific

~~Note~~  
- NumPy is <sup>value</sup> mutable but size is immutable.



## Ways to create NumPy Array:-

1) Creating 'empty Array': `empty()` creates array with any random garbage value.

```
arr1 = np.empty([3,2])
```

2) Creating arrays filled with zeros:

```
np.zeros([2,3], dtype=np.int64, order='F')
```

default dtype is float.

default order is 'C' (C-type).

```
o/p: array([[0, 0, 0],
            [0, 0, 0]], dtype=int64)
```

3) Creating array filled with 1

```
arr4 = np.ones([2,3], dtype=np.float32)
```

```
o/p: array([[1., 1., 1.],
            [1., 1., 1.]], dtype=float32)
```

We can create a new array similar to existing array having similar dtype, order etc. fn are:

`empty` like (`<existing array>`)

`zeros` like (`"`)

`ones` like (`"`)



e.g: `arr2 = np.ones_like(arr3)`

o/p:

`array([[1, 1, 1],  
[1, 1, 1]], dtype=int64)`

4) Using `arange(start, stop, step, dtype)`

e.g: `arr5 = np.arange(7)`

o/p: `array([0, 1, 2, 3, 4, 5, 6])`

5) Using `array()`

e.g: `L = [1, 2, 3, 4]`

`arr6 = np.array(L)` Note

o/p: `array([1, 2, 3, 4])`

6) Using `linspace(start, stop, no. of values to be generated)`

e.g: `arr6 = np.linspace(2.5, 5, 6)`

Pandas Data structures: Pandas DS are enhanced version of NumPy structured array in which rows and columns can be identified and accessed with labels rather than integer indices.

Two basic DS are:

Series & Dataframe.