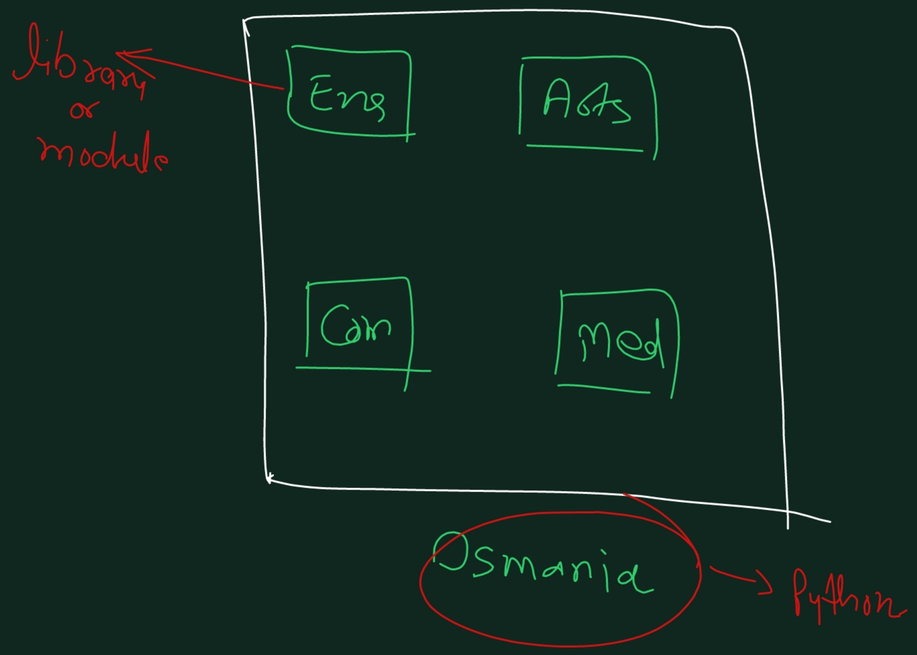
Python for DA

→ It has vast collection of libraries.



① Numpy → Numerical Python

② Pandas → Analysis / Manipulation

③ Matplotlib → Plotting / Vizualization

④ Seaborn → Advanced Visualization

① Numpy : Numerical Python → Data Structure → Array : Homogeneous (similar Data)

List	Vs	Array
Hetero		Homo
Slower		Faster
Storage		Calculation

② Create

import numpy (or) import numpy as np

numpy.array( ) np.array( )

→ Collection
- list
- tuple
- set
- dict →
- range( )

Create

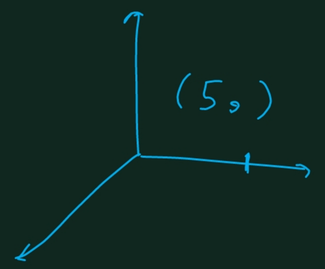
```
arr = np.array([1,2,3,4])
```

↳ o/p → array([1,2,3,4])

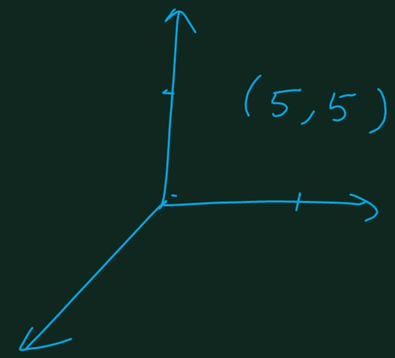
1) type(arr) → numpy.ndarray() ✓
2) arr.ndim → no. of dimension , 1
3) arr.dtype → datatype of arr , int
4) arr.size → no. of elements , 4
5) arr.shape → shape of an array , (4,)

Types

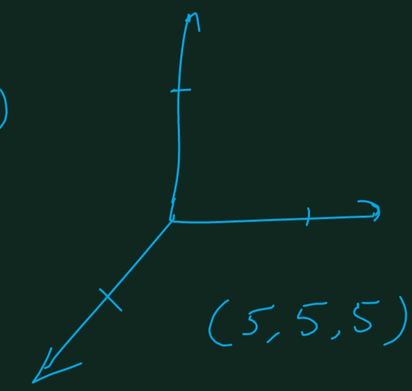
① 1D



② 2D



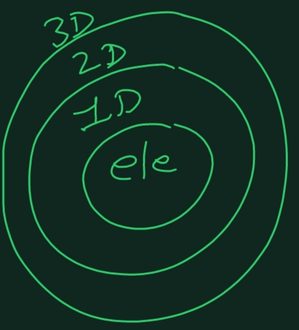
③ 3D



Shape :

1D → (Rows,)
2D → (Rows, Columns)
3D → (Layer, Rows, Columns)

① Manual



1D

(Rows,)

[0, 1, 2, 3]

(4,)

[Row]

No. of elements

[1, 2, 3, 4, 5, 6]

(6,)

2D

(Rows, Columns)

[[0, 1, 2, 3.]]

(1, 4)

Rows — no. of 1D

Columns — no. of ele in any 1D

[[0, 1, 2], [2, 3, 4]]

(2, 3)

Note : No. of ele in all 1D must be same.

3D

(Layer, Rows, Columns)

[[[0, 1, 2, 3,]],]

(1, 1, 4)

Layer — no. of 2D

Rows — no. of 1D in any 2D

Columns — no. of ele in any 1D

[[[0, 1, 2], [2, 3, 4]],
[[1, 1, 1], [2, 2, 2]]]

(2, 2, 3)

② Dynamic

reshape(shape)

arr = array([0,1,2,3,4,5])

↳ (6,) → 6

arr2D = arr.reshape(R,C)

1 x 6
6 x 1
3 x 2
2 x 3
} → 6

arr3 = arr.reshape(L,R,C)

1,6,1
1,1,6
6,1,1
1,3,2
2,3,1
⋮
} → 6

Note: Multiplication of shape must be same.