```
import numpy as np

#Scalar or 0D TENSOR
arr0 = np.array(4)
```

Tensor and ndimensional arrays are same . If you come from maths or physics background then they are called as tensor

O-D = scalar

```
arr0
```

```
array(4)
```

```
arr0.ndim
#here you see that the dimension of the array is 0 therefore it is a
```

```
0
```

1D TENSOR = VECTOR = 1D ARRAY = ARRAY

it's NDIM is 1

Number of dimension's is equal to number of number of axes

How much is the dim = to number of axes

If there are 2D then there are 2 axes

Number of axes = rank = Dimension

```
arr1 = np.array(1,2,3,4)
# Array takes one or 2 positional argument but the argument given is
# This where you see array is either 0D, 1D or 2D
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-17-7e0f7206d7f7> in <module>
----> 1 arr1 = np.array(1,2,3,4)
      2 # Array takes one or 2 positional argument but the argument given is 4
      3 # This where you see array is either 0D, 1D or 2D

TypeError: array() takes from 1 to 2 positional arguments but 4 were given

SEARCH STACK OVERFLOW
```

```
arr1 = np.array([1,2,3,4])
#this is a 1D array, as you have passed only 1 argument []
# You can check the same by printing the dimensions
```

```
# its rank would be 1
# its dimensions will be 1
arr1
```

```
array([1, 2, 3, 4])
```

```
arr1.ndim
#n dimension is 1
#rank is 1
# axis is 1
```

```
1
```

1D tensor/ Vector

READ IT CAREFULLY

WHAT IS THE DIMENSION OF THIS VECTOR ?

IT IS A 4D VECTOR BUT 1D TENSOR/VECTOR

IT IS 4D BECAUSE YOU HAVE PASSED 4 INPUTS SIMILARLY IT WOULD HAVE BEEN 5D

VECTOR IF YOU PASS 5 INPUT EXAMPLE [1,2,3,4,5,]

Notice you added all number of scalar values to create a vector 1 scalar 2 scalar [1,2] vector

So vector are basically collection of more and more that is n numbers of scalar

So Matrices are basically collection of more and more that is n numbers of Vectors

MATRICES ARE COLLECTION OF VECTORS

ITS DIMENSION IS 2D

BECAUSE THERE ARE 2 DIMENSIONS

2 RANKS = 2 AXIS = 2 DIMENSIONS

[1,2,3] [4,5,6] [7,8,9]

MATRICS MADHE

[1,2,3

4,5,6

7,8,9]

Double-click (or enter) to edit

```
# 2 AXeS = ROWS AND COLOUMNS
arr2 = np.array([[1,2,3],[4,5,6],[7,8,9]])
#see the brackets carefully
arr2
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
arr2.ndim
```

```
2
```

- So if you want to make higher dimensional matrics

  4D tensor is the collection of 3D tensor

  Normally you move from 0D- 5D tensors

  RANK AND AXIS AND NUMBER OF DIMENSIONS ARE SAME

  SHAPE MEANS HOW MANY ITEMS YOU HAVE IN A PARTICULAR AXIS

  FOR EXAMPLE IN 3*3 MATRIX THERE ARE 2 AXIS, ROW AXIS AND COLOUMNS AXIS

  therefore the shape of matrix is (3*3)

  RANK AND AXIS FOR (2*3) MATRIX IS

  RANK IS 2 AS THERE ARE STILL ROWS AND COLOUMNS BUT THE SHAPE IS NOW (2*3) WHERE THERE ARE TWO ROWS AND THREE COLOUMNS FIRST NUBER INDICATES NUMBER OF ROWS SECOND NUMBER INDICATE NUMBER OF COLOUMNS

  SIZE

  SIZE is always multiplication of shape of the tensor

  so first find shape and multiply them

  Exception for scalar the size is always 1