

# Numpy

Array is display the items in a particular way

Array object in Numpy is called ndarray

Numpy arrays are stored at one continous place in memory so it can access easily and that behaviour is xalled Locality of reference.

Thats why they are faster than Lists

Numpy source code is located at the github repository  
<https://github.com/numpy/numpy>

```
In [1]: # Import numpy
```

```
import numpy
```

```
In [2]: # example
```

```
import numpy
```

```
arr=numpy.array([1,2,3,4,5])
```

```
print(arr)
```

```
[1 2 3 4 5]
```

```
In [3]: # numpy as np ( alias is the alternate name to refer samething)
```

```
import numpy as np
```

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

```
print(arr)
```

```
[1 2 3 4 5]
```

```
In [4]: # checking numpy version
```

```
import numpy as np
```

```
print(np.__version__)
```

```
1.23.5
```

## NumPy creating Arrays

the array object in numpy is called ndarray

we create NumPy ndarray object by using the array() function.

In [ ]:

In [10]:

```
import numpy as np

arr= np.array([1,2,3,4,5])
print(arr)
print(type(arr))

[1 2 3 4 5]
<class 'numpy.ndarray'>
```

In [11]:

```
arr1=(1,2,3)
print(type(arr1))

<class 'tuple'>
```

In [13]:

```
import numpy as np

arr1=np.array((1,2,3))
print(type(arr1))

<class 'numpy.ndarray'>
```

## Dimensions in arrays

In [14]:

```
# 0 D arrays

import numpy as np

arr=np.array(56)
print(arr)

56
```

In [15]:

```
# 1 D array

import numpy as np

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

print(arr)

[1 2 3 4 5]
```

In [20]:

```
# 2 D Arrays ( 2nd order tensors)

import numpy as np

arr=np.array([[1,2,3,4,5],[6,7,8,9,10]])

print(arr)

[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
```

In [54]:

```
# 3 D array ( 3rd order tensors)

import numpy as np
```

```
arr=np.array([[[[1,2,3,4,5],[6,7,8,9,10]],[[11,12,13,14,15],[16,17,18,19,20]]]])

print(arr)

[[[ 1  2  3  4  5]
  [ 6  7  8  9 10]]

 [[11 12 13 14 15]
  [16 17 18 19 20]]]
```

In [25]: *# check the number of dimensions*

```
import numpy as np

a=np.array(56)
b=np.array([1,2,3,4,5])
c=np.array([1,2,3,4,5],[6,7,8,9,10])
d=np.array([[[[1,2,3,4,5],[6,7,8,9,10]],[[11,12,13,14,15],[16,17,18,19,20]]]])

print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)

0
1
2
3
```

In [34]: *# defining number of dimensions ndmin*

```
import numpy as np

arr1= np.array([1,2,3,4],ndmin=5)
arr2=np.array([1,2,3,4], ndmin=4)
arr3=np.array([1,2,3,4], ndmin=3)
arr4=np.array([1,2,3,4], ndmin=2)
arr5=np.array([1,2,3,4], ndmin=1)

print(arr1)
print(arr2)
print(arr3)
print(arr4)
print(arr5)
#print(" Number of Dimensions: ", arr.ndim)

[[[[[1 2 3 4]]]]]
[[[[[1 2 3 4]]]]]
[[[1 2 3 4]]]
[[1 2 3 4]]
[1 2 3 4]
```

the innermost dimension (5th dim) has 4 elements

the 4 th dim has 1 element that is the vector

the 3 rd dim has 1 element that is a marix with vector

the 2nd dim has 1 element that is the 3d array

the first dimension has 1 element that is the 4d array

vector: single dimensional or one-dimensional array as a list

Matrix: matrix is the 2 D data structure where numbers are arranged in row and columns

3 D array : Multi dimensional array " arrays of arrays( Block size, row size, column size)( an array of 2 D array)

4 D array: an array of 3 D array

```
In [36]: import numpy as np

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

```
[1 2 3 4]
1
```

```
In [38]: import numpy as np

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

```
[[1 2 3 4]
 [1 2 3 4]]
2
```

```
In [40]: import numpy as np

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

print(arr)
print(arr.ndim)
```

```
[[[1 2 3 4]]]
3
```

```
In [53]: import numpy as np

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

```
[[[[1 2 3 4]]]]
4
```

```
In [ ]:
```