NumPy

Numerical Python(NumPy) is the core library of python used for numerical operations.

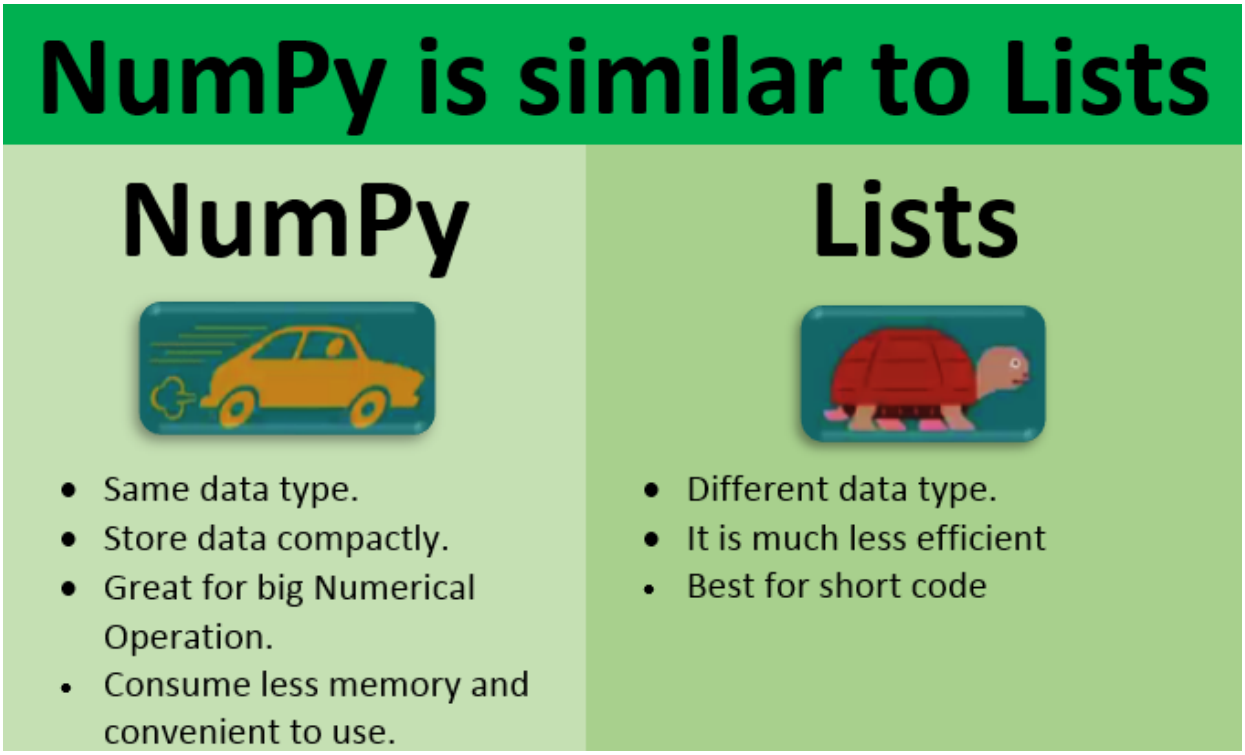
How to install NumPy is VsCode?

To install Numpy run this command **"pip install numpy"** in terminal window of VsCode. Not in OutPut window.

NumPy is already installed Jupyter Notebook.

How to use NumPY

To Access NumPy and its functions import it in your python code like this. **"import numpy as np"**



What is an Array? گوروں کا طریقہ

- **An array is central data structure of the NumPy library.**
- **An array is a grid of values and it contains information about the raw data,how to locate an element and how to interpret an element.**
- **It has a grid of elements that can be indexed in various ways.**
- **The elements are all of the same type, referred to as the array dtype.**
- **An array can be indexed by a tuple of non-negative integers, by booleans, by another array or by integers.**

What is rank of an array?

The rank of the array is the number of dimensions.

What is the shap of an Array?

The shape of the array is a tuple of integers giving the size of the array along each dimension."

Tuple

tup1=(90, "Chilla_version_2.0", True, 3.5)

What is an Array? بابا جی کا طریقہ

Examples

In [4]:
```
import numpy as np
a = np.array([7,5,4,3,7])
```

In [3]:
```
type(a)
```

Out[3]:
```
numpy.ndarray
```

In [5]:
```
#List of lists
import numpy as np
b = np.array([[10,11,12,20],[30,33,55,76],[90,78,65,34]])
b
```

Out[5]:
```
array([[10, 11, 12, 20],
       [30, 33, 55, 76],
       [90, 78, 65, 34]])
```

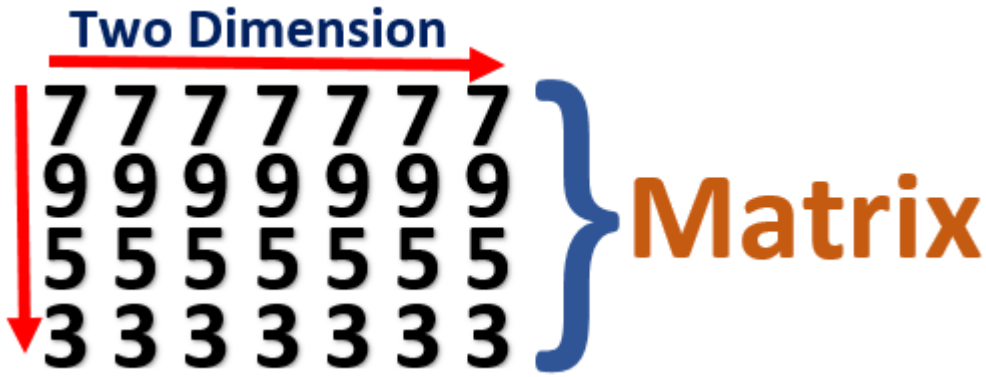
Vector

A vector is an array with a single dimension(There is no difference between row and column

vector.

Matrix

A matrix refer to an array with two dimensions.



1-D Array/Single Dimension Array

In [5]:
```
import numpy as np
c = np.array([89,45,53,34,3,3])
c
```

Out[5]:
```
array([89, 45, 53, 34,  3,  3])
```

In [8]:
```
# finding type
type(c)
```

Out[8]:
```
numpy.ndarray
```

In [9]:
```
#Finding Length of array
len(c)
```

Out[9]:
```
6
```

In [13]:
```
#Finding Index of item.
c[0]
```

Out[13]:
```
89
```

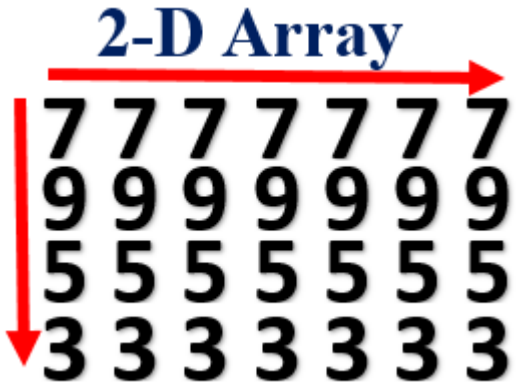
In [15]:
```
#Is trah bhi array print kr skty hain
c[0:]
```

Out[15]:
```
array([89, 45, 53, 34,  3,  3])
```

2-D Array

A matrx refers to an array having two dimensions.

The NumPy ndarray class is used to represent both matrices and vectors.



In [1]:
```
#List of lists
import numpy as np
d = np.array([[10,11,12,20],[30,33,55,76],[90,78,65,34],[35,44,89,55]])
d
```

Out[1]:
```
array([[10, 11, 12, 20],
       [30, 33, 55, 76],
       [90, 78, 65, 34],
       [35, 44, 89, 55]])
```

In [2]:
```
#Type
type(d)
```

Out[2]:
```
numpy.ndarray
```

In [6]:
```
#Length
len(d)
```

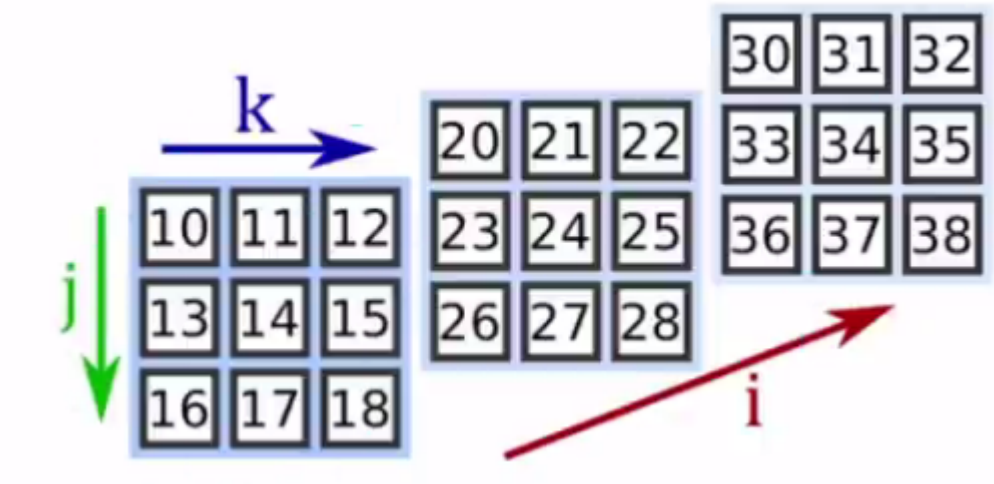
Out[6]:
```
4
```

In [7]:
```
#indexing
d[3]
```

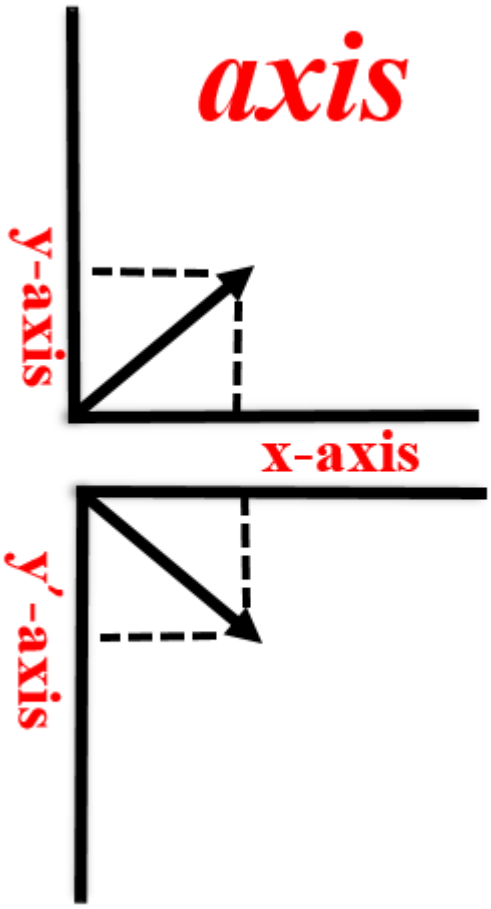
Out[7]:
```
array([35, 44, 89, 55])
```

3-D or higher:

For 3-D or higher dimensional arrays, the term tensor is also commonly used



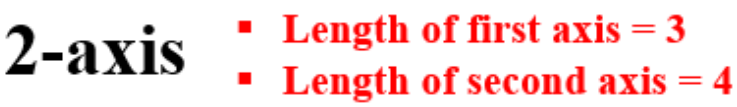
Attributes of Array



- **Dimensions are called axis.**

In [2]:
```
import numpy as np
e = np.array([[7,3,6,4],[5,9,3,3],[7,3,6,4]])
e
```

Out[2]:
```
array([[7, 3, 6, 4],
       [5, 9, 3, 3],
       [7, 3, 6, 4]])
```



Different ways to create 1-D array

1- Using List

In [35]:
```
import numpy as np
f=np.array([9,7,6,9,2,3,2])
f
```

Out[35]:
```
array([9, 7, 6, 9, 2, 3, 2])
```

2- zeros Function

In [38]:
```
g=np.zeros(4)
g
```

Out[38]:
```
array([0., 0., 0., 0.])
```

3- ones Function

In [40]:
```
h=np.ones(2)
h
```

Out[40]:
```
array([1., 1.])
```

4- empty Array

In [99]:
```
np.empty(6)
```

Out[99]:
```
array([0., 0., 0., 0., 0., 0.])
```

In [100…
```
np.empty(3)
```

Out[100…
```
array([1.21734824e-312, 0.00000000e+000, 1.87673412e-152])
```

5- By providing Range

اس میں ہم رینج کے طور ایک نمبر دیتے ہیں جہاں تک ہم ایرے پرنٹ کروانا چاہتے ہیں۔ جیسے اگلی مثال میں 9 رینج ہے

In [49]:
```
j=np.arange(9)
```

In [50]:
```
j
```

Out[50]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
```

6- Array having specific Range

You know what I mean.

For example 4 to 9.

9 is excluded because indexing start from "0".

In [52]:
```
k=np.arange(4,9)
k
```

Out[52]:
```
array([4, 5, 6, 7, 8])
```

7- With Specific Interval

In [54]:
```
l=np.arange(3,30,3)
l
```

Out[54]:
```
array([ 3,  6,  9, 12, 15, 18, 21, 24, 27])
```

8- Line spaced Array

Mtlb ye k hm aik specific range provide krain gy k kaha se kaha tk array print

krni hai or ye bhi btain gy k us range k drmiyan kitni values ayen gi.

In [64]:
```
m=np.linspace(10,100,num=10)
m
# 10-100 k b/w aise 10 number print ho gy jin ka diffrence bilkl same ho gy.
```

Out[64]:
```
array([ 10.,  20.,  30.,  40.,  50.,  60.,  70.,  80.,  90., 100.])
```

9- Specific DataType in Array

In [74]:
```
n=np.zeros(7,dtype=np.int8)
n
```

Out[74]:
```
array([0, 0, 0, 0, 0, 0, 0], dtype=int8)
```

In [76]:
```
o=np.ones(8,dtype=np.float64)
o
```

Out[76]:
```
array([1., 1., 1., 1., 1., 1., 1., 1.])
```

How to Create 2-D Array

In [82]:
```
np.zeros((3,4))
```

Out[82]:
```
array([[0., 0., 0., 0.],
```

```
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [86]:
```
np.zeros((3,4))
```

Out[86]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [87]:
```
np.ones((7,8))
```

Out[87]:
```
array([[1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1., 1., 1.]])
```

In [88]:
```
np.empty((3,4))
```

Out[88]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

How to create 3-D Array

Making and reshaping a 3-D Array

In [96]:
```
np.arange(24).reshape(2,3,4)
```

Out[96]:
```
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],
```

```
[[12, 13, 14, 15],
 [16, 17, 18, 19],
 [20, 21, 22, 23]]])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: