

Programming and Problem Solving through Python Language

O Level / A Level

Chapter - 9: NumPy

Introduction to NumPy

- NumPy is python package and it stands for Numerical Python.
- It was created in 2005 by Travis Oliphant. It is an open source project.
- It is core python library used working with arrays and for scientific computing.
- It also has inbuilt functions for working with linear algebra, Fourier transform, matrices and data science.
- It has an N-dimensional array object(**ndarray**) which is in the form of rows and columns.

Lists and Numpy arrays

- Python has the lists that are similar to arrays and serves the purpose, but it is slow to process.
- To work with the large data set, python provides the Numpy arrays, and it provides much faster access to data, than the Python List.
- Numpy arrays are stored at one continuous place in memory unlike the case of Python List, so it provides the fast access and processing. This Numpy follows the Locality of Reference.

Installation of Numpy

- Numpy is not available as bundled package in Python.
- We can install it using PIP. PIP is package management system to install and manage software packages written in Python. PIP stands for Pip Installs Packages.
- Steps to install the NumPy in Python.
- Go to command prompt
 - Type cmd on RUN or click the command prompt
- Change the folder to location where python installed.
 - cd c:\Python38-32
- Move to the folder **Scripts** in **C:\python38-32**.
 - c:\Python38-32> cd scripts
 - c:\Python38-32\Scripts>
- Type the PIP install Numpy and press the enter key. Keep the internet connection active during this process. It will download the package and install it automatically.
 - c:\Python38-32\Scripts> pip install numpy

```
Command Prompt
C:\Python38-32\Scripts>pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/52/2c/bf86d762ae65550dc8a7ab8381ba610bb69af6db619b3755f2b73052c6b9/numpy-1.18.4-cp38-cp38-win32.whl
Installing collected packages: numpy
  WARNING: The script f2py.exe is installed in 'c:\python38-32\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.18.4
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Python38-32\Scripts>
```

Using numpy

- To use the NumPy package, it is required to import the library NumPy.
- NumPy is used to work with arrays. It has functions to manipulate the arrays.
- The array object in NumPy is called ndarray.

Example

```
import numpy
arr1=numpy.array(1)
arr2=numpy.array([1,2,3,4,5])
print('Numpy Version : ', numpy.__version__)
print('Array 1 Values : ', arr1)
print('Array 2 Values : ', arr2)
print('Array 1 Data Type : ', type(arr1))
print('Array 2 Data Type : ',type(arr2))
```

Output

```
Numpy Version : 1.18.4
Array 1 Values : 1
Array 2 Values : [1 2 3 4 5]
Array 1 Data Type : <class 'numpy.ndarray'>
Array 2 Data Type : <class 'numpy.ndarray'>
```

Dimension of arrays in Numpy

- A dimension in arrays is one level of array depth.
- Types of Arrays
 1. 0-D arrays: It has a scalar element in an array. Each value in an array is a 0-D array.
 - e.g. `numpy.array(10)`
 2. 1-D arrays: An array that has 0-D arrays as its elements is called uni-dimensional or 1-D array.
 - e.g. `numpy.array([30 , 40])`
 3. 2-D arrays: An array that has 1-D arrays as its elements is called a 2-D array.
 - e.g. `numpy.array([[30 , 40] , [50 , 60]])`
 4. 3-D arrays: An array that has 2-D arrays (matrices) as its elements is called 3-D array.
 - e.g. `numpy.array([[[30 , 40] , [50 , 60]] ,
[[70 , 80] , [90 , 10]]
])`
- `<array_name>.ndim` used to know the dimension of the array.

Example

```
import numpy

arr1=numpy.array(1)
arr2=numpy.array([1,2,3,4,5])
arr3=numpy.array( [ [1,2,3], [4,5,6] ] )
arr4=numpy.array( [ [ [1,2,3], [4,5,6] ],
                  [ [1,2,3], [4,5,6] ] ] )

print('Array -1')
print(arr1)
print('Array Dim :', arr1.ndim)

print('Array -2')
print(arr2)
print('Array Dim :', arr2.ndim)

print('Array -3')
print(arr3)
print('Array Dim :', arr3.ndim)

print('Array -4')
print(arr4)
print('Array Dim :', arr4.ndim)
```

Output

Array -1

1

Array Dim : 0

Array -2

[1 2 3 4 5]

Array Dim : 1

Array -3

[[1 2 3]

[4 5 6]]

Array Dim : 2

Array -4

[[[1 2 3]

[4 5 6]]

[[1 2 3]

[4 5 6]]]

Array Dim : 3