



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

19AD501- Foundations of Data Science

III Year / V Semester

Unit 2 - DATA SCIENCE USING PYTHON

Topic 1: Numpy





Numpy



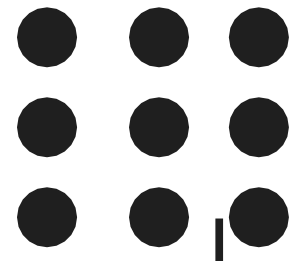
- NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.
- Using NumPy, mathematical and logical operations on arrays can be performed.

Why Use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.



Numpy



Operations using NumPy

Using NumPy, a developer can perform the following operations

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

Example Creating array with numpy

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



Numpy



Data Types in NumPy

NumPy has several data types, and refer to data types with one character, like i for integers, u for unsigned integers etc.

Below is a list of all data types in NumPy and the characters used to represent them.

i - integer

b - boolean

u - unsigned integer

f - float

c - complex float

m - timedelta

M - datetime

O - object

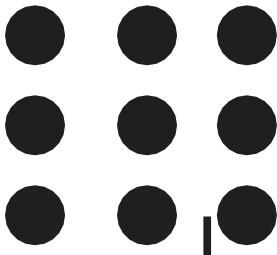
S - string

U - unicode string

V - fixed chunk of memory for other type (void)



Numpy



Example

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr.dtype)
```

Example

```
import numpy as np
arr = np.array(['apple', 'banana', 'cherry'])
print(arr.dtype)
```



Numpy



Slicing arrays

- Slicing in python means taking elements from one given index to another given index.
- We pass slice instead of index like this: `[start:end]`.
- We can also define the step, like this: `[start:end:step]`.
- If we don't pass start its considered 0
- If we don't pass end its considered length of array in that dimension
- If we don't pass step its considered 1



Scipy



- SciPy is a free and open-source Python library used for scientific computing and technical computing.
- It is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python.
- It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

Why use SciPy

- SciPy contains varieties of sub packages which help to solve the most common issue related to Scientific Computation.
- SciPy package in Python is the most used Scientific library only second to GNU Scientific Library for C/C++ or Matlab's.
- Easy to use and understand as well as fast computational power.
- It can operate on an array of NumPy library.



Numpy



Example

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5])
```

Output : 2,3,4,5

Example

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[4:])
```

Output:

5,6,7



Numpy



Python NumPy Operations

ndim:

You can find the dimension of the array, whether it is a two-dimensional array or a single dimensional array. In the below code, with the help of 'ndim' function, I can find whether the array is of single dimension or multi dimension.

```
import numpy as np
a = np.array([(1,2,3),(4,5,6)])
print(a.ndim)
```

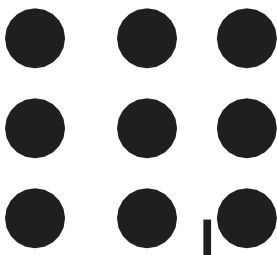
Item size:

You can calculate the byte size of each element. In the below code, I have defined a single dimensional array and with the help of 'itemsize' function, we can find the size of each element.

```
import numpy as np
a = np.array([(1,2,3)])
print(a.itemsize)
```

Output – 4

So every element occupies 4 byte in the above numpy array.



THANK YOU