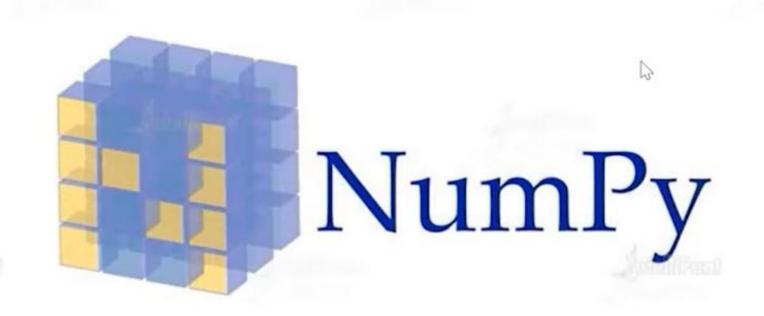# Numpy

# What is Numpy?

The NumPy library is a very popular Python library and the abbreviation is "Numerical Python". The purpose of NumPy library is to do scientific computation and apply them to python applications.

# Why Numpy?

1. First of all, It is a open source Python library

2. It is fast because it is written in C and Python

3. In Python, there is no in-built array capabilities

4. You can use List as an alternative for arrays, but NumPy is better. But how is it better? We will discuss that now.

# Numpy Faeturs:

**1** A durable N-Dimensional array

**2** Tools to Manipulate and work with the array

**3** Perform mathematical and logical operations on the array

**4** Powerful pre-defined functions

# Numpy Operations:

1. Fourier Transform and Shape Manipulation

2. Linear Algebra and Random Number Generation

3. Tools for integrating C/C++

4. Easily integrate with databases

# Numpy Advantages over Lists:

**01** NumPy arrays consume lesser memory due to contiguous memory allocation

**02** Operations complete quicker in NumPy because of its better runtime behavior
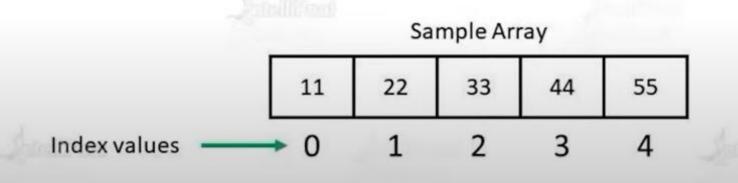
**03** Pre-defined functions for linear algebra operations are available

**04** NumPy arrays have fixed types, so there is no type checking in execution which saves time

# What is Numpy Array?

Array is the data structure which the NumPy library revolves around. To be more precise, it is a matrix/grid of values which are all of the same data type. It is also called a **ndarray** because it is a N-Dimensional array.

The are indexed using non-negative integers starting from 0. NumPy array act similar to a Python list, but is totally different in execution and is relatively a lot quicker.

Sample Array

| 11 | 22 | 33 | 44 | 55 |
|----|----|----|----|----|

Index values ⟶ 0   1   2   3   4

# Applications of Numpy

**01** Can be used of Mathematical operations and also can be an MATLAB alternative

**02** For any Backend code, NumPy is important because Pandas & NumPy work well together

**03** Helps in plotting while using the Matplotlib Python library

**04** A important part of Machine Learning and Data Science algorithms

# Numpy Array creation:

The first step to create a NumPy array is to import the NumPy package

```
import numpy as np
```

**import** – This command is to get access to another module/library to get access to their code

**numpy** – This is the module which we should get access to do any NumPy related operations

# Different ways to create Numpy Array:

**01** Converting other Python structures to arrays (Lists, tuples)

**02** Using NumPy array creation objects or NumPy pre-defined functions (ones, range, zeros, etc)

**03** Use of special Library functions (random)

# Numpy Array Initialization:

The difference between creating an array and initializing the array is that while you create, the array can be empty as well. But when you initialize, you are actually entering values in the array.

You can initialize a NumPy array in all the ways you create an array. While you use those pre-defined functions, they initialize the array after creating it

# Numpy Array Inspection:

Any real world problem involving data will have millions of rows and thousands of columns. So, if you are using or creating a DS algorithm to manipulate that data, then it will be very helpful if you are able to inspect the structure of your arrays

# Numpy Array Inspection cont….

| Inspect Functions | Description |
|---|---|
| ndarray.shape | Gives a tuple with the array dimensions. You can also use this function to resize the array |
| ndarray.size | Returns the count of number of elements in the given array |
| ndarray.ndim | Provides the dimension of the given array |
| ndarray.dtype | Returns the datatype used by the array (eg. Int32, float64) |