NumPy Features – Why use NumPy?

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays

NumPy (Numerical Python) is an open-source core Python library for scientific computations. It is a general-purpose array and matrices processing package.

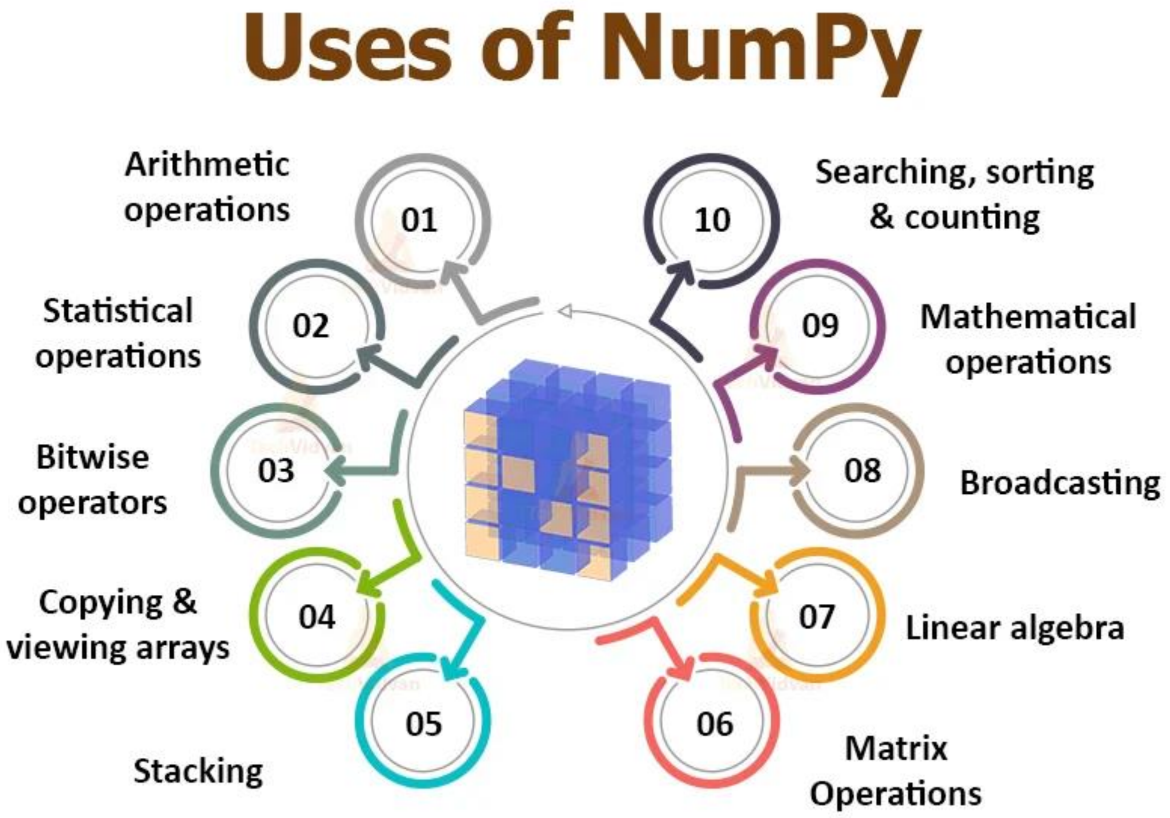
Python is slower as compared to Fortran and other languages to perform looping. To overcome this, we use NumPy that converts monotonous code into the compiled form.

NumPy is one of the most useful **external libraries** available in Python. It has a wide variety of **functions** to work with **arrays** and a **powerful multi-dimensional array object**. It has **operations** that are **applicable** to a **vast range** of **platforms**.

Numpy can be put to use for **storing**, **manipulation**, and **deletion** of **array elements**. We can use it for **sorting**, **indexing**, and **stacking** of the **array elements**. It has modules regarding various operations:

- Arithmetic operations
- Statistical Operations
- Bitwise Operators
- Linear Algebra
- Copying and viewing arrays
- Stacking
- Searching, Sorting, and counting, etc.
- Mathematical Operations

- Broadcasting
- Matplotlib for graphical representations
- Matrix Operations, etc.



NumPy Features

These are the important features of NumPy:

1. High-performance N-dimensional array object

- This is the most important feature of the NumPy library.
- It is the homogeneous array object.
- We perform all the operations on the array elements.
- The arrays in NumPy can be one dimensional or multidimensional.

a. One dimensional array

The one-dimensional array is an array consisting of a single row or column. The elements of the array are of homogeneous nature.

b. Multidimensional array

In this case, we have various rows and columns. We consider each column as a dimension. The structure is similar to an excel sheet. The elements are homogenous.

2. NumPy contains tools for integrating code from C/C++ and Fortran

We can use the functions in NumPy to work with code written in other languages. This helps implement inter-platform functions.

3. NumPy contains a multidimensional container for generic data

Here generic data refers to the parameterized data type of arrays. NumPy can perform functions on the generic data types.

The arrays in NumPy are of homogenous nature. These array elements are assigned parameters. The parameters help increase the diversity of the arrays.

4. Additional linear algebra, Fourier transform, and random number capabilities

NumPy has the capability to perform complex operations on array elements like linear algebra, Fourier transform, etc.

- We have the linalg module for linear algebra functions.
- Similarly, we have fft functions for Fourier Transform in NumPy.
- We have a matrix module for applying functions on matrices.
- We also have special functions for plotting graphs in the matplotlib module of NumPy.

5. NumPy consists of broadcasting functions

The broadcasting of array is a very useful concept when we work with arrays of uneven shapes.

NumPy can broadcasts the shape of smaller arrays according to the larger ones.

The broadcasting of arrays has some rules and limitations in its implementation.

For broadcasting one of the arrays needs to be one-dimensional or both the arrays are supposed to be of the same shape.

There are also a few other limitations on the shape of the arrays.

6. NumPy has data type definition capabilities to work with varied databases

We can work with arrays of different data types. We can use the dtype(…) function to determine the data type and hence get a clear idea about the available data set.

With the array definition, we have an additional dtype argument to perform array functions. The knowledge of the data type of an array is very important due to the restrictions on NumPy operations.

Installation of Numpy

If you already have python, NumPy can be installed with:

```
pip install numpy
```

If you don’t have python yet, you might want to consider using Anaconda. Anaconda is the easiest way for getting started and has all the major packages pre-installed.

Importing NumPy

To use any package or library in your code, it needs to be made accessible.

We can start using NumPy and its available functionalities using the import statement:

```
import numpy as np
```

We import it as np to save time and for standardization of the code.

Difference between NumPy Array and Python Lists

NumPy is an alternative for lists in Python as it holds less memory, has faster processing, and is more convenient to use. The difference between the elements is that the NumPy array has to be homogenous.

We can maintain homogeneity for the efficient application of the mathematical functions. Arrays in NumPy are more compact compare to lists and the data type specification, which leads to code optimization.

Numpy works along with other Packages

We can combine NumPy with other basic packages like SciPy and Mat-plotlib. The combination implements scientific computations and plotting graphs respectively.

NumPy Array Basics

The most important feature is the N-dimensional array object is called ndarray. It describes the collection of homogeneous elements that uses zero-based indexing. All the elements of the ndarray have equal size of the memory block. Each element in ndarray is the object of a data type object (called dtype).

NumPy Applications

NumPy **provides highly efficient multi-dimensional arrays**. It also contains the necessary tools to manipulate and perform operations on these arrays. It is one of the best packages to use for data science implementation.

1. An alternative for lists and arrays in Python

Arrays in NumPy are equivalent to lists in Python.

The most important feature of NumPy arrays is that like lists in Python, the NumPy arrays are homogenous sets of elements; they are homogenous in nature.

This differentiates NumPy arrays from Python arrays. This maintains uniformity for mathematical operations that would not be possible with heterogeneous elements.

Another benefit of using NumPy arrays is there are a large number of functions that are applicable to these arrays.

These functions could not be performed when applied to Python arrays due to their heterogeneous nature.

2. NumPy maintains minimal memory

Arrays in NumPy are objects. Python deletes and creates these objects continually, as per the requirements. Hence, the memory allocation is less as compared to Python lists.

NumPy has features to avoid memory wastage in the data buffer.

NumPy has functions like copies, view, and indexing that helps in saving memory. Indexing helps to return the view of the original array, that implements reuse of the data. It also specifies the data type of the elements which leads to code optimization.

3. Using NumPy for multi-dimensional arrays

We can also create multi-dimensional arrays in NumPy.

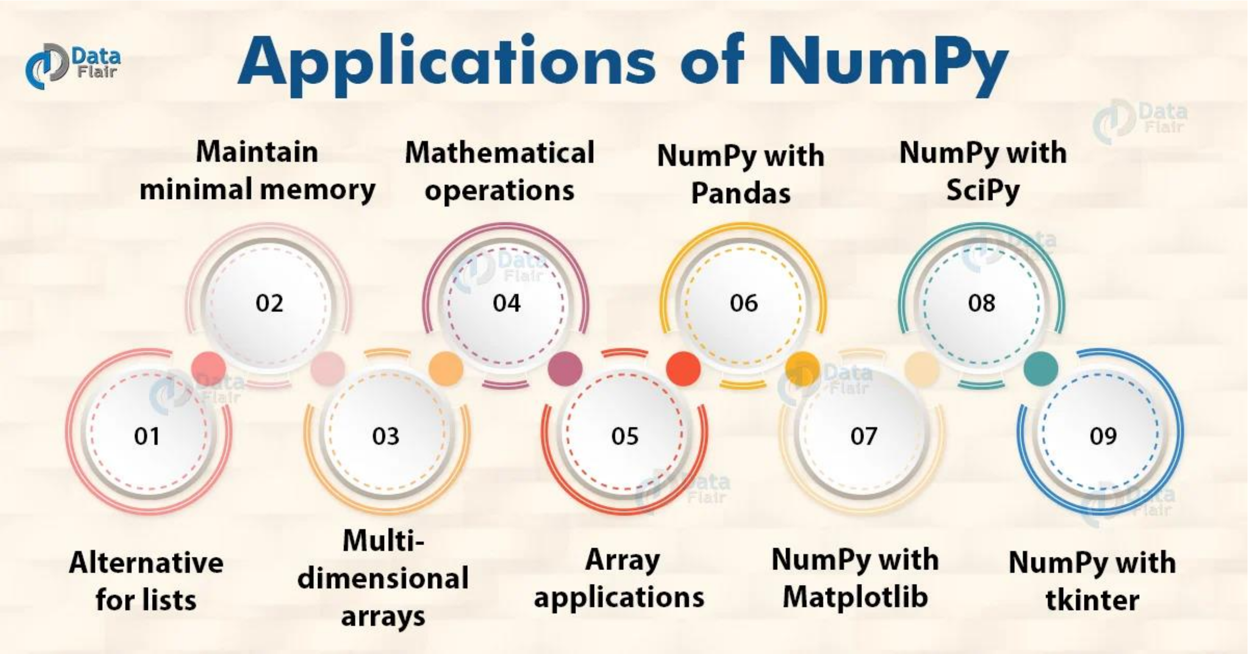
- These arrays have multiple rows and columns.
- These arrays have more than one column that makes these multi-dimensional.
- Multi-dimensional arrays implement the creation of matrices. These matrices are easy to work with.
- With the use of matrices, the code also becomes memory efficient.
- NumPy has a matrix module to perform various operations on these matrices.

4. Mathematical operations with NumPy

NumPy also includes easy to use functions for mathematical computations on array datasets, with modules for basic and special mathematical functions in NumPy.

There are functions for Linear Algebra, bitwise operations, Fourier transform, arithmetic operations, string operations, etc.

NumPy Array Applications



1. Shape Manipulations

Users can change array dimensions at runtime if the output produces the same number of elements.

We apply np.reshape(...) function on the array.

The reshape function is useful for performing various operations, like when we want to broadcast two dissimilar arrays.

2. Array Generation

We can generate array datasets for implementing various functions.

We can also generate a predefined set of numbers for array elements using the np.arrange(...) function.

Reshape function is useful to generate a different set of dimensions.

Use the random function to generate an array having random values.

Similarly, use linspace(…) function to generate arrays having similar spacing in elements.

NumPy an create arrays with pre-filled ones or zeroes.

The default data type is set to be float64 but we can edit the data type using dtype option.

3. Array Dimensions

NumPy consists of both one and multidimensional arrays.

Some functions have restrictions on multidimensional arrays. It is then necessary to transform those arrays into one-dimensional arrays.

We can transform multi-dimensional to single dimension using np.ravel(…)

NumPy Applications work with Other Python Libraries

1. NumPy with Pandas

Pandas is one of the most important libraries in Python for data analysis. Pandas provide high performance, fast analysis, and data cleaning.

Pandas is used to manipulate data structures and has data analysis tools.

Pandas consists of a data frame object.

Pandas interoperates with NumPy for faster computations. Both libraries together are very helpful resources for scientific computations.

2. NumPy with Matplotlib

Matplotlib is a module in NumPy.

Matplotlib is a very helpful tool to work with graphical representations. It consists of a wide range of functions to plot and manipulate graphs.

This combination can replace the functionalities of MatLab.

Matplotlib is used to generate the graphs of data manipulation results.

We enhance it further with the use of graphic toolkits like PyQt and wxPython.

3. NumPy with SciPy

SciPy is the most important open-source scientific library in Python.

It has been built upon the functionalities of NumPy. There are advanced functionalities in SciPy for scientific computations.

We can combine it with NumPy for greater mathematical performance. The combination helps in the implementation of complex scientific operations.

4. NumPy with Tkinter

Tkinter is a standard library for the GUI representation of the NumPy data.

Its combination with NumPy can implement fast and easy GUIs.

The use of Tkinter along with NumPy is user friendly. We can easily convert the array objects into image objects.