

# Chapter 1 - NumPy Arrays

—

By Beena Nair

# LEARNING OBJECTIVES

This presentation will help you to analyse and comprehend about the following topics:

- 1. Introduction to NumPy**
- 2. Arrays v/s Lists**
- 3. Creation of NumPy Arrays**
- 4. Various functions of arrays**
- 5. Arithmetic Operations on Arrays**
- 6. Statistical operations on Arrays**
- 7. Applications of Arrays**

- Numpy is a python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy stands for Numerical Python.

- NumPy provides an `ARRAY` object.
- 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.
- Data Science: is a branch of computer science where we study how to store, use and analyze data for deriving information from it.

Watch this video to  
understand the meaning of  
NumPy -

[https://www.youtube.com/watch?v=8JfDAm9y\\_7s](https://www.youtube.com/watch?v=8JfDAm9y_7s)

# Comparison of NumPy with Lists

1. NumPy arrays are stored at one continuous place.
  2. More efficient than Lists.
  3. NumPy arrays works on homogenous data types
  4. Hence it is faster.
  5. NumPy arrays does not support addition and removal of elements from an existing array
  6. This behavior is called locality of reference in computer science.Hence NumPy is faster than
1. Lists are stored at different places in memory .
  2. Less efficient than arrays.
  3. Lists works on heterogeneous data types.
  4. Hence it is slower.
  5. NumPy arrays does not support addition and remove operations.
  6. This behavior is called locality of reference in computer science.
  - 7.

# Create a NumPy ndarray Object

NumPy is used to work with arrays. The array object in NumPy is called **ndarray**.

We can create a NumPy **ndarray** object by using the **array()** function.

For example:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5])
```

```
print(arr)
```

```
print(type(arr))
```



# Try yourself :

[https://www.w3schools.com/python/numpy\\_creating\\_arrays.asp](https://www.w3schools.com/python/numpy_creating_arrays.asp)

---

Create a 1D array containing the values 4,5,6,7:

```
import numpy as np
```

```
arr = np.array([4, 5, 6, 7])
```

```
print(arr)
```

Create a 2D array containing the values 4,5,6,7:

```
import numpy as np
```

```
arr = np.array([[4, 5], [6, 7]])
```

```
print(arr)
```

Check how many dimensions are there in the arrays a,b,c and d.

```
import numpy as np
a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
print(a.ndim)
print(b.ndim)
print(c.ndim)
print(d.ndim)
```

**NumPy Arrays provides the `ndim` attribute that returns an integer that tells us how many dimensions the array have.**

### **Array Indexs:**

**You can access any array elements using the index number. The index number starts from 0. Relation between index number and size of array is  $\text{size} = \text{last index number} + 1$**

How to get the third and fourth elements of the array and add them. Display the output.

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

Ans is 9

Complete the program to Display the 5th element of 2 dim.

```
import numpy as np
```

```
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
```

```
print(_____)
```

Print the last element of 2 dim.

```
import numpy as np
```

```
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
```

```
print('Last element from 2nd dim: ', arr[1, -1])
```

Negative indexing is also possible. It starts from the last element.



This program code helps to create an array with all zeros.

```
import numpy as np
```

```
arr = np.zeros([3,2])
```

```
print(arr)
```

The output will be

```
[[ 0 0 ]
```

```
[ 0 0 ]
```

```
[0 0 ]]
```

This program code helps to create an array with all ones.

```
import numpy as np
```

```
arr = np.ones([3,2])
```

```
print(arr)
```

The output will be

```
[[ 1 1 ]
```

```
[ 1 1 ]
```

```
[1 1 ]]
```

This program code helps to create an array with all elements with same value.

```
import numpy as np
```

```
arr = np.full([3,2],5)
```

```
print(arr)
```

The output will be

```
[[ 5 5 ]
```

```
 [ 5 5 ]
```

```
[ 5 5 ]]
```

This program code helps to create an array from a range using `arange()` function.

```
import numpy as np
```

```
arr = np.arange(3,7)
```

```
print(arr)
```

The output will be

```
[ 3 4 5 6]
```

This program code helps to create an array from a range using `linspace()` function.

```
import numpy as np  
  
arr = np.linspace(2,3,6)  
  
print(arr)
```

The output will be

```
[ 2.  2.2  2.4  2.6  2.8  3.]
```

This program code helps to copy an array from another array using `copy()` function.

```
import numpy as np
arr = np.array([2,3,6])
arr1=np.copy(arr)
print(arr)
print(arr1)
arr[2]=10
print(arr)
print(arr1)
```

The output will be

```
[ 2 3 6 ]
```

```
[2 3 6]
```

```
[2 3 10]
```

```
[2 3 6]
```

`copy()` function

**The main difference between a copy and a view of an array is that the copy is a new array, and the view is just a view of the original array.**

**The copy *owns* the data and any changes made to the copy will not affect original array, and any changes made to the original array will not affect the copy.**

This program code helps to create an array from another array using reshape() function.

```
import numpy as np
arr = np.array([2,3,6,12])
arr1=np.reshape(arr,(2,2))
print(arr)
print(arr1)
```

The output will be

```
[ 2 3 6 12]
```

```
[[2 3 ]
```

```
[6 12] ]
```

reshape() function is used to change the arrangement of elements of the array.



# NumPy Data Types

- **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 )
-

# Operations on ARRAY

- SLICING
  - JOIN
  - TRANSPOSE
  - ARITHMETIC  
OPERATIONS
  - STATISTICAL  
OPERATIONS
-

## 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].
```

Slice elements from the beginning to index 4 (not included):

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7])
```

```
print(arr[:4])
```

## Negative Slicing

Use the minus operator to refer to an index from the end:

Slice from the index 3 from the end to index 1 from the end:

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

Return every other element from index 1 to index 5:

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7])
```

```
print(arr[1:5:2])
```

```
import numpy as np
```

```
arr = np.array([1, 2, 3, 4, 5, 6, 7])
```

```
print(arr[::2])
```

What is the output of the following program.

```
import numpy as np
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
print(arr[0:2, 2])
```

In 2D array the first parameter is the row number and the second parameter is column number.

This program code displays the index 1 to index 4(not included).

```
import numpy as np
```

```
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9,  
10]])
```

```
print(arr[0:2, 1:4])
```



# ASSIGNMENT

1. Write a Python program to create an array of 5 integers and display the array items. Access individual element through indexes
2. Make 5X2 int array in a range b/w 500 to 320 having difference between each element is 10?
3. From a 3\*3 numPy array return items in the second column from all rows?
4. Make a 4X2 integer array of unsigned int16 and Print shape, dimensions and Length of each element ?
5. Create a 3 x 4 two-dimensional ndarray from the range of integers 13..24.

# ASSIGNMENT

6. Write code to create an ndarray having six zeros in it. Write statements to change 3rd and 5th elements of this ndarray to 15 and 25 respectively.
7. Which functions allow you to extract only rows or columns from an ndarray ?
8. Create an 9X3 array from a range between 13 to 40 such that the difference between each element is 1 and then Split the array into three equal-sized sub-arrays.

**ALL THE ABOVE QUESTIONS TO BE PART OF PRACTICAL FILE ALSO.YOU WRITE THE CODE USING W3SCHOOL.COM OR JUPYTER.ORG SITES**