

## What is Python Numpy Array?

NumPy arrays are a bit like Python lists, but still very much different at the same time. For those of you who are new to the topic, let's clarify what it exactly is and what it's good for.

As the name kind of gives away, a NumPy array is a central data structure of the numpy library. The library's name is actually short for "Numeric Python" or "Numerical Python".

## Create a NumPy Array

Simplest way to create an array in Numpy is to use Python List

```
myPythonList = [1,9,8,3]
```

To convert python list to a numpy array by using the object np.array.

```
numpy_array_from_list = np.array(myPythonList)
```

To display the contents of the list

```
numpy_array_from_list
```

### Output

```
array([1, 9, 8, 3])
```

In practice, there is no need to declare a Python List. The operation can be combined.

```
a = np.array([1,9,8,3])
```

**NOTE:** Numpy documentation states use of np.ndarray to create an array. However, this the recommended method

You can also create a numpy array from a Tuple

## Mathematical Operations on an Array

You could perform mathematical operations like additions, subtraction, division and multiplication on an array. The syntax is the array name followed by the operation (+,-,\*,/) followed by the operand

Example:

```
numpy_array_from_list + 10
```

### Output:

```
array([11, 19, 18, 13])
```

This operation adds 10 to each element of the numpy array.

## Shape of Array

You can check the shape of the array with the object shape preceded by the name of the array. In the same way, you can check the type with dtypes.

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

```
(3, )
int64
```

An integer is a value without decimal. If you create an array with decimal, then the type will change to float.

```
#### Different type
b = np.array([1.1,2.0,3.2])
print(b.dtype)
```

```
float64
```

## 2 Dimension Array

You can add a dimension with a "," comma

Note that it has to be within the bracket []

```
### 2 dimension
c = np.array([(1,2,3),
              (4,5,6)])
print(c.shape)
(2, 3)
```

## 3 Dimension Array

Higher dimension can be constructed as follows:

```
### 3 dimension
d = np.array([
    [[1, 2, 3],
     [4, 5, 6]],
    [[7, 8, 9],
     [10, 11, 12]]
])
print(d.shape)
(2, 2, 3)
```

## Summary

Below, a summary of the essential functions used with NumPy.

**Objective****Code**

Create array	<code>array([1,2,3])</code>
print the shape	<code>array([.]).shape</code>