Create NumPy Array

Summary: in this tutorial, you'll learn how to create NumPy arrays including one-dimensional, two-dimensional, and three-dimensional arrays.

The array is the core data structure of the NumPy library. A NumPy array is a grid of values with the same type and indexed by a tuple of non-negative integers.

All arrays are instances of the `ndarray` class. To create a new NumPy array, you use the `array()` function of the NumPy library.

Creating one-dimensional arrays

The following example uses the `array()` function to create a one-dimensional (1-D) array:

```
import numpy as np

a = np.array([1, 2, 3])

print(type(a))
print(a)
```

Output:

```
<class 'numpy.ndarray'>
[1 2 3]
```

How it works.

First, import the `numpy` library as `np`:

```
import numpy as np
```

Second, create a 1D array by passing a list (https://www.pythontutorial.net/python-basics/python-list/) of three integers:

```
a = np.array([1, 2, 3])
```

The `array()` function returns a new instance of the `ndarray` type. Therefore, the `type(a)` returns `<class 'numpy.ndarray'>`.

A 1-D array is known as a *vector*.

Getting the dimension of an array

To get the number of dimensions of an array, you use the `ndim` property. In NumPy, dimensions are called axes. For example:

```
import numpy as np

a = np.array([1, 2, 3])

print(a.ndim)
```

Output:

```
1
```

In this example, The `ndim` property returns one as expected.

Getting the data type of array elements

To get the data type of the elements of an array, you use the `dtype` property. For example:

```
import numpy as np

a = np.array([1, 2, 3])

print(a.dtype)
```

Output:

```
int32
```

In this example, the type of the elements is `int32`. If you want to set the type of the array's elements, you can use the `dtype` argument of the `array()` function. For example:

```
import numpy as np

a = np.array([1, 2, 3], dtype=np.float64)

print(a)
print(a.dtype)
```

Output:

```
[1. 2. 3.]
float64
```

In this example, the numbers of the array have the decimal point (`.`) and the data type of its elements is `float64`.

Creating two-dimensional arrays

The following example uses the `array()` function to create a two-dimensional (2-D) array:

```
import numpy as np

b = np.array(
    [
        [1, 2, 3],
        [4, 5, 6]
    ]
)

print(b)
print(b.ndim)
```

Output:

```
[[1 2 3]
 [4 5 6]]

2
```

In this example, we pass a list of a list of integers to the `array()` function. The `ndim` property returns 2 as expected.

A good tip to get the number of dimensions of an array is that you count the square brackets (`[`) until you encounter the first number. The number of square brackets is the number of dimensions or axes.

A two-dimensional array is also called a *matrix*.

Creating three-dimensional array

The following example uses the `array()` function to create a three-dimensional (3-D) array:

```
import numpy as np

```

```
c = np.array(
    [
        [
            [1, 2, 3],
            [4, 5, 6]
        ],
        [
            [7, 8, 9],
            [10, 11, 12]
        ],
    ]
)

print(c.ndim)
```

Output:

```
3
```

Note that a 3-D array is also called a tensor.

Getting shapes of arrays

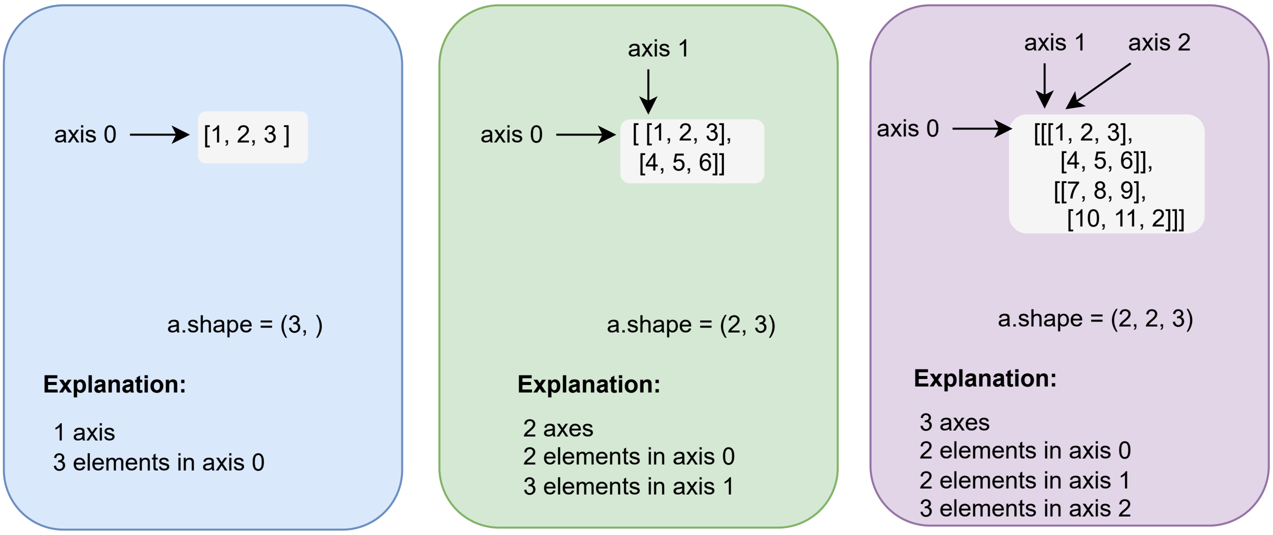
To find the number of axes and the number of elements on each axis of an array, you use the `shape` property. For example:

```
import numpy as np



a = np.array([1, 2, 3])
print(a.shape)  # (3,)

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

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

Output:

```
(3,)
(2, 3)
(2, 2, 3)
```

The following picture explains the shape of each array a, b, and c:



The `shape` property returns a tuple:

- The number of elements in the tuple is the number of axes.
- Each tuple element stores the number of elements of the corresponding axis.

Summary

- A numpy array is a grid of values with the same type and is indexed by a tuple of non-negative values.
- Numpy arrays have the type of `ndarray` .
- Use the `array()` function to create a numpy array.
- Use the `dtype` property to get the data type of array's elements.
- Use the `ndim` property to get the number of dimensions or the number of axes.
- Use the `shape` property to get the number of dimensions as well as the number of elements in each dimension.