**MIT xPRO Data Engineering Certificate**

**NumPy**

**What Is a NumPy Array?**

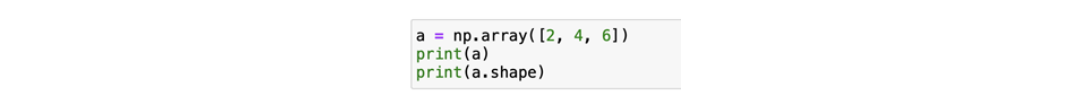
The central NumPy data structure is the "ndarray" — an n-dimensional array data structure.

An ndarray is a (usually fixed-size) multidimensional container of items of the same type and size. The number of dimensions and items in an array is defined by its shape, which is a tuple of n non-negative integers that specify the sizes of each dimension.

In contrast to Python's built-in list data structure, all elements of a single array must be of the same type.

Consider a one-dimensional NumPy array.

You can define an array like so:



The output of the code above is



where the first print statement outputs the elements of the array a, and the second one returns the shape of it. Notice that the tuple in the output only has one defined entry. This means that a is one-dimensional with size 3.

**How Can I Handle NumPy Arrays?**

As described above, NumPy is extremely powerful. It follows that the library includes not only arrays but also a wide range of functions and methods to handle these data structures and perform more advanced mathematical computations.

A comprehensive list of all these functions can be found in the NumPy application programming interface (API) documentation Links to an external site.. Here is an overview of the most common ones:

np.array(n) Creates n-dimensional arrays

np.zeros(n) Creates an array of length n with entries that are all zeros

np.ones(n) Creates an array of length n with entries that are all ones

np.eye(n) Creates an array of size n with an identity matrix

np.linspace(a,b,n) Creates an array with n equally spaced entries from a to b

np.random.rand(n) Creates an array with n random float entries

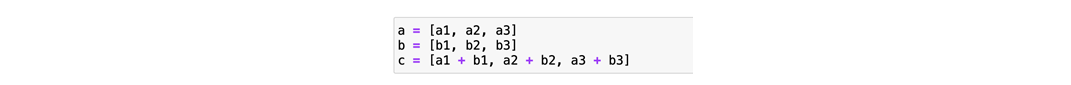
np.random.randint(n) Creates an array with n random int entries

**Broadcasting**

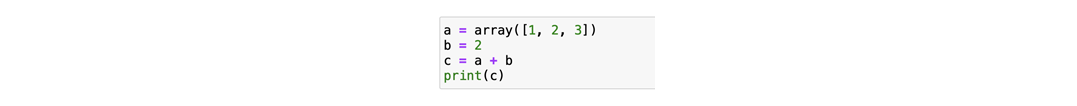
In the NumPy environment, broadcasting refers to the action of array arithmetic between arrays of the same shape and size.

Looking at some examples in NumPy will help make this concept more concrete. Note that the examples in this section are not exhaustive but instead are common types of broadcasting you may see or implement.

Consider the following:

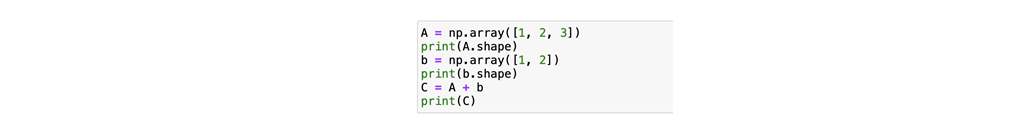


In NumPy, operations such as addition, subtraction, multiplication, and division are element-wise. For instance,

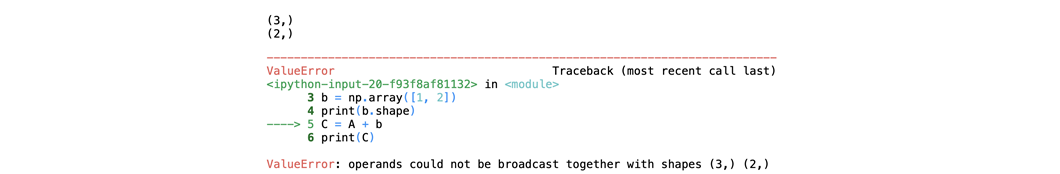


returns

On the other hand,



returns



Running the example first prints the shapes of the arrays and then raises an error when attempting to broadcast because the dimensions don’t match.

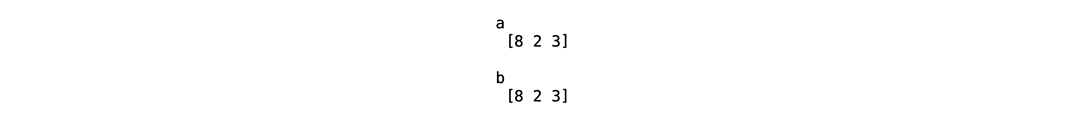
**How Do I Copy Arrays?**

You can copy one NumPy array into another. Copying an array means a new instance is created, and the contents of the original array are copied into this newly created array.

To copy array data using Python’s NumPy library, you can use the .copy() function. For example,



returns



**What Are NumPy 2D Arrays?**

Two-dimensional arrays consist of an array within an array. In this type of array, the position of a data element is referred to by two indices instead of one. They are like a table (or matrix) with rows and columns of data (Tutorialspoint 2021).

A two-dimensional NumPy array can be defined as follows:



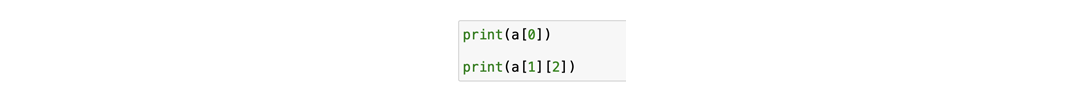
Naturally, these arrays can contain as many entries as you need.

**Accessing Values in a Two-Dimensional Array**

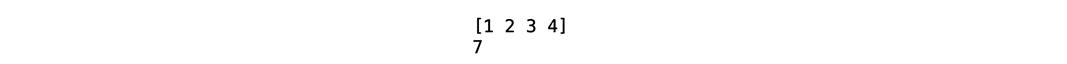
The data elements in two-dimensional arrays can be accessed using two indices. One index refers to the position of the data element in the outer array, and the other refers to its position in the inner array (Tutorialspoint 2021).

Consider the array defined in the previous paragraph.

The statements

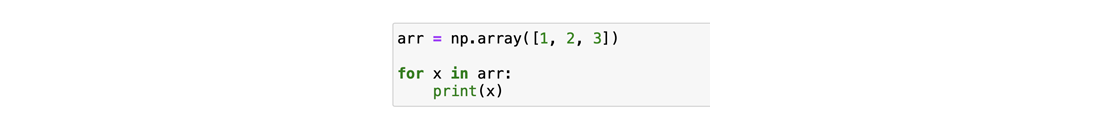


return



**How Do I Iterate over Arrays?**

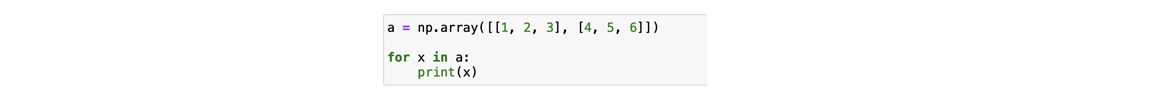
It is possible to iterate over arrays by using a loop. For example,



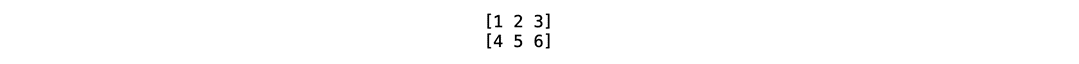
returns



On the other hand,



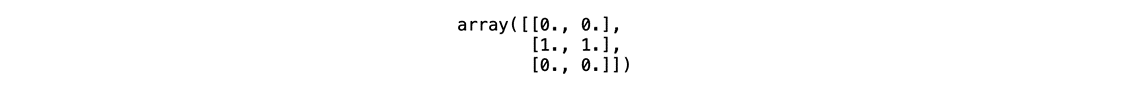
gives



**How Do I Insert and Modify Elements in an Array?**

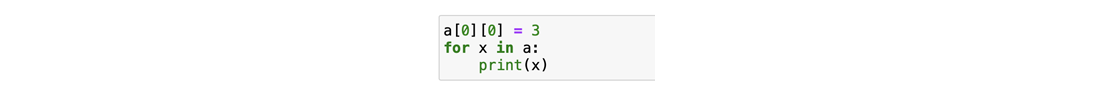
NumPy offers functions that let you insert and modify elements in an array.

To insert new data elements at specific positions in an array, you can use the np.insert() function, like so:



You can update data elements of the inner array by reassigning the values using the array index.

For example,



returns

