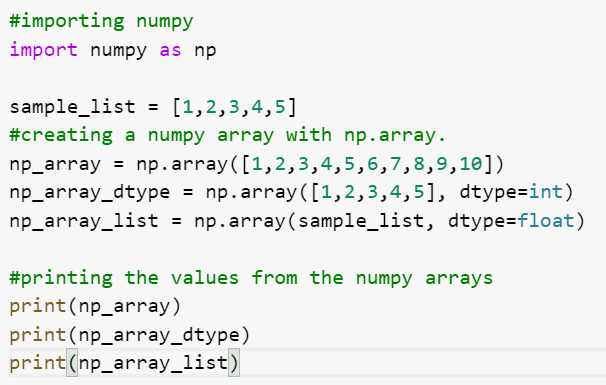
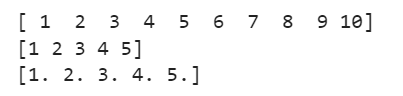
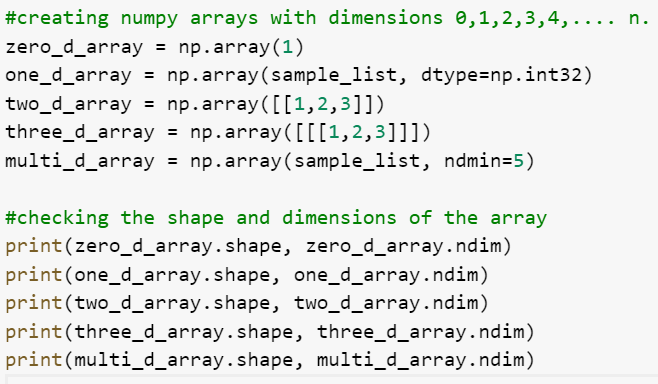
1. Import the numpy package with the alias np that we will be using in this session.
2. To create a numpy array, we are going to use **np.array** - here we can either give the sequence as the parameter or simply write the values. We can also specify the data type in the array, like we have done for the **np\_array\_dtype** as **integer** and for the **np\_array\_list** we have used the **float** data type.



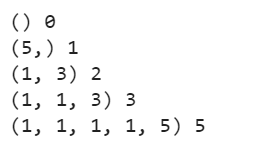
1. The output for the above code will give us the numpy arrays.



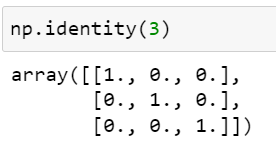
1. Now that we have understood the basics of array creation using numpy, let’s understand how we can create numpy arrays with various dimensions as well.



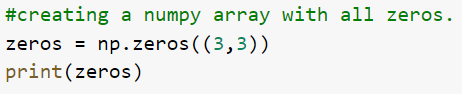
1. The output shows the shape and dimensions of the arrays created.



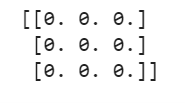
1. Creating a 3x3 identity matrix using numpy



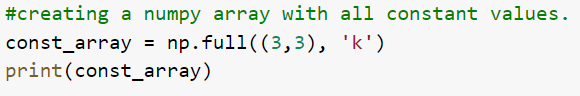
1. Creating a numpy array with all the values as zero



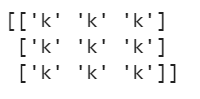
1. The output returns a 3x3 matrix with all zeros.



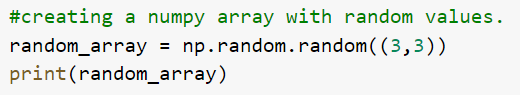
1. Creating a numpy array with all the constant values



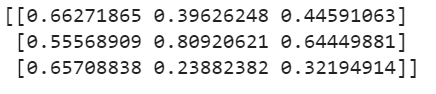
1. The output will give a 3x3 matrix with all values as ‘k’



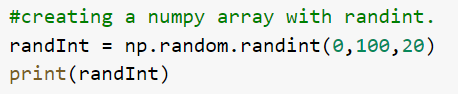
1. Creating a numpy array with all the random values



1. The output will produce a numpy array with random values in the shape 3x3.



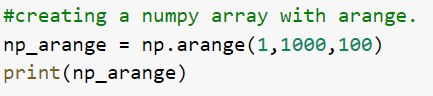
1. Creating a numpy array with random integers.



1. The output will produce a numpy array with random integer values in the range 0,100 and a size 20.



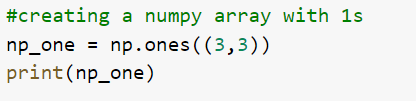
1. Creating a numpy array with arange.



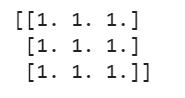
1. The output will produce a numpy array in a range 1 to 1000, and a step 100.



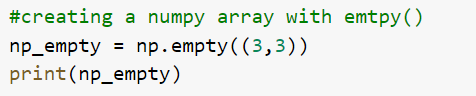
1. Creating a numpy array with all 1s as the elements.



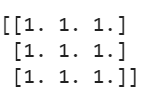
1. The output will produce a numpy array with all values as 1.



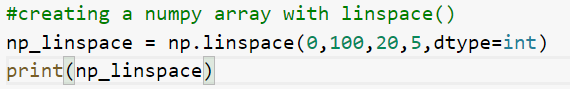
1. Creating a numpy array with empty()



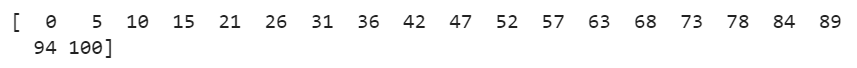
1. The output will produce a matrix in the shape 3x3 without initializing any values.



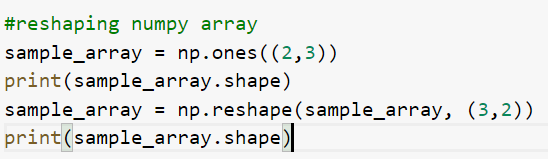
1. Creating a numpy array with linspace()



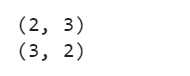
1. The output will produce a numpy array with a range 0-100, sample size as 20 and a step 5 with data type int.



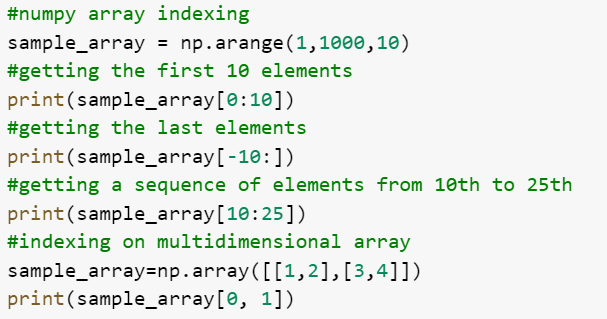
1. Reshaping a numpy array



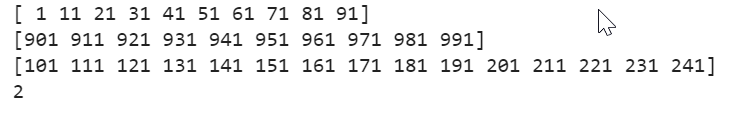
1. The output of the above code will change the shape of the sample\_array.



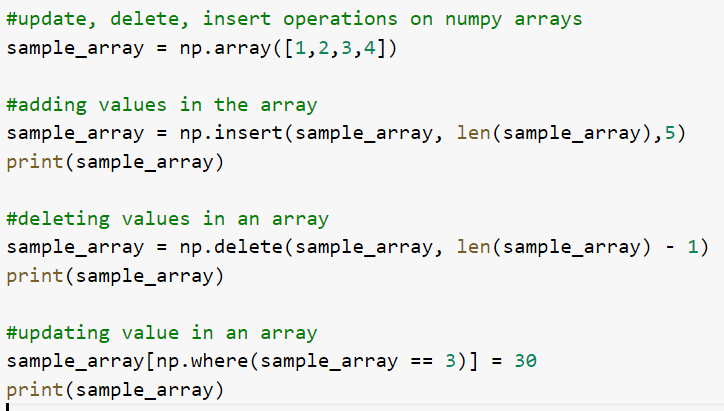
1. Let’s understand how indexing works in numpy.



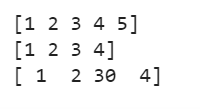
1. The output returns the results for the above indexing operations.



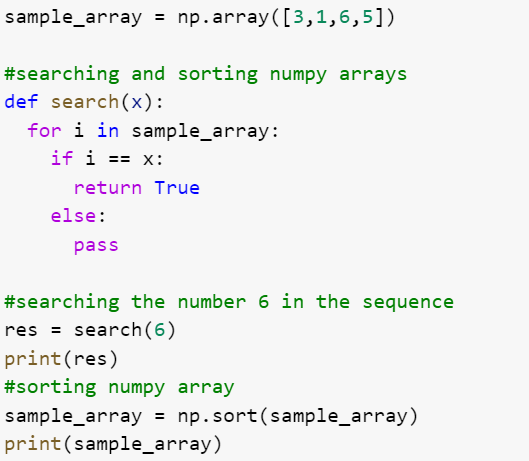
1. CRUD operations for a numpy array.



1. The output returns the updated arrays, with values inserted, deleted and updated for specific conditions.



1. Searching and sorting on the same sample\_array



1. The output shows the result for the searching and sorting operations.

