**NUMERICAL PYTHON**

**NUMPY**

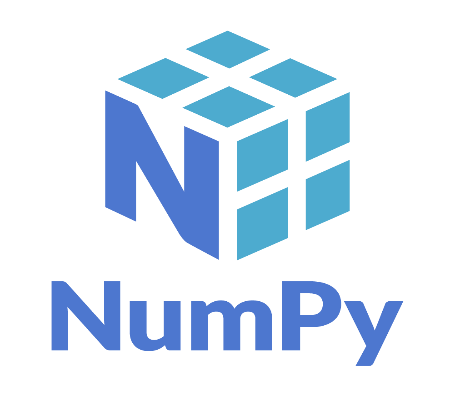
Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with python. It is open-source software.

**Features of Numpy**

NumPy has various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy in Python can also be used as an efficient multi-dimensional container of generic data. Arbitrary data types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

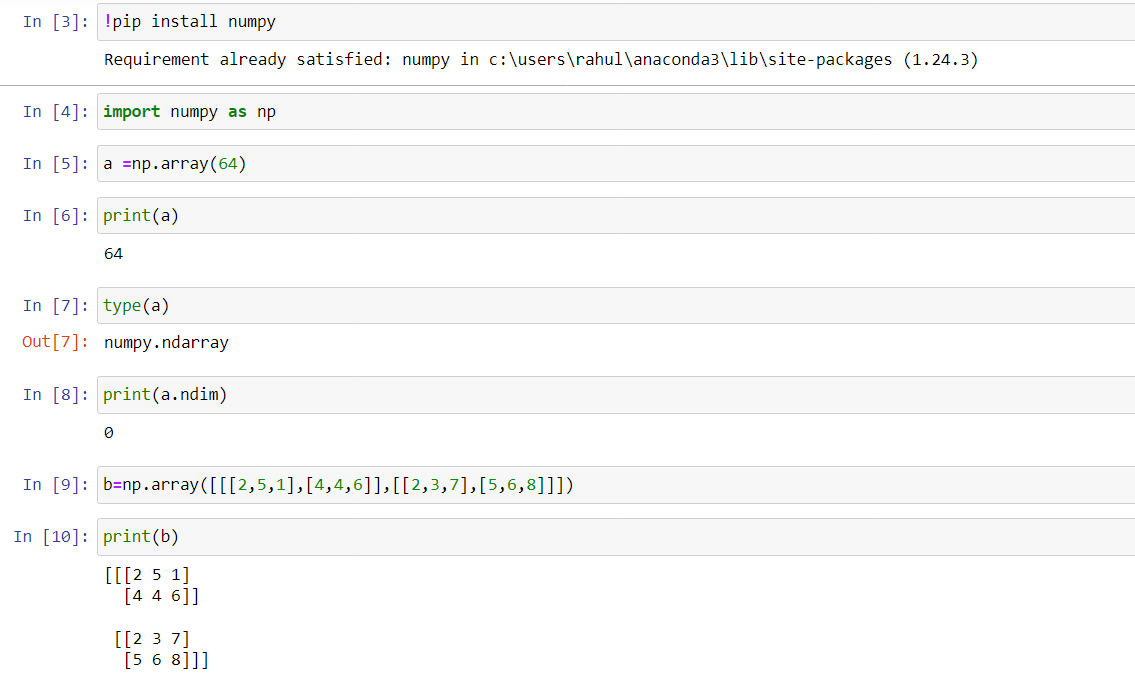


**Array in Numpy**

Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In Numpy, number of dimensions of the array is called rank of the array.A tuple of integers giving the size of the array along each dimension is known as shape of the array. An array class in Numpy is called as **ndarray**. Elements in Numpy arrays are accessed by using square brackets and can be initialized by using nested Python Lists.

**Creating a Numpy Array**

Arrays in Numpy can be created by multiple ways, with various number of Ranks, defining the size of the Array. Arrays can also be created with the use of various data types such as lists, tuples, etc. The type of the resultant array is deduced from the type of the elements in the sequences.

****

# **Basics of NumPy Arrays**

Numpy stands for Numerical Python. It is a Python library used for working with an array. In Python, we use the list for purpose of the array but it’s slow to process. NumPy array is a powerful N-dimensional array object and its use in linear algebra, Fourier transform, and random number capabilities. It provides an array object much faster than traditional Python lists.

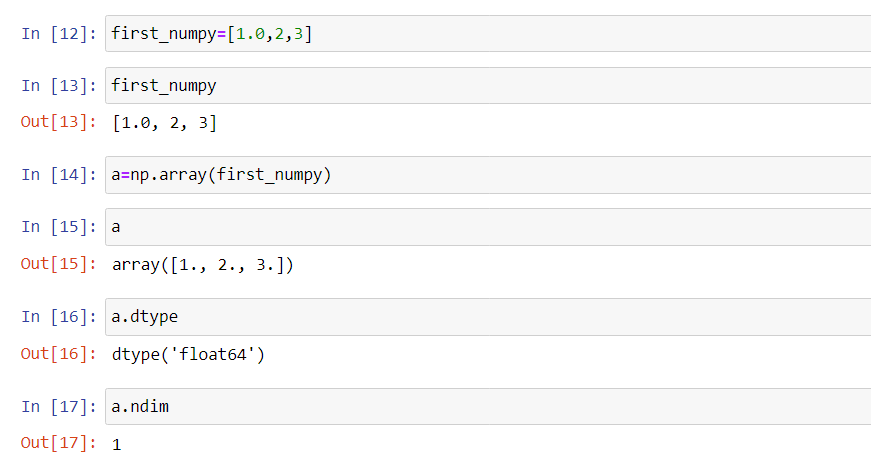
#### **Types of Array:**

1. One Dimensional Array
2. Multi-Dimensional Array

#### **One Dimensional Array:**

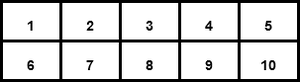
A one-dimensional array is a type of linear array.

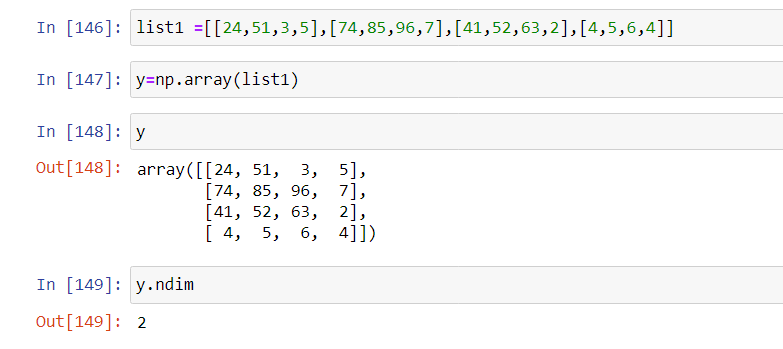
Lightbox

****

#### **Multi-Dimensional Array:**

Data in multidimensional arrays are stored in tabular form.





**Functions in python**

Here are some of the important NumPy functions in Python which every Data scientist.

**np.dtype**

This parameter specifies the data type of the returned array. It is an optional parameter and its default value is None. If dtype is set to None, the data type of the returned array will be the same as the input array.

**np.size()**

In Python, numpy.size() function count the number of elements along a given axis.

**np.arange():**

 This function is used to create an array with a range of values.

**np.zeros():**

 This function is used to create an array filled with zeros.

**np.ones():**

 This function is used to create an array filled with ones.

**np.linspace():**

 This function is used to create an array with a specified number of evenly spaced values.

**np.random.rand():**

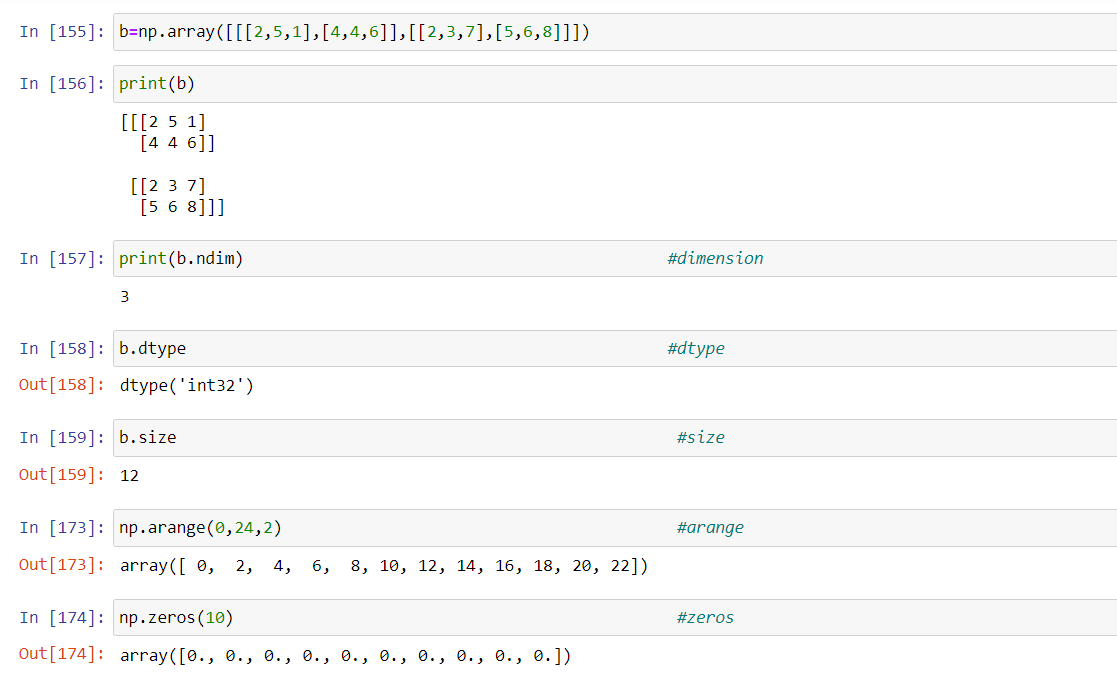
This function is used to create an array with random values between 0 and 1.

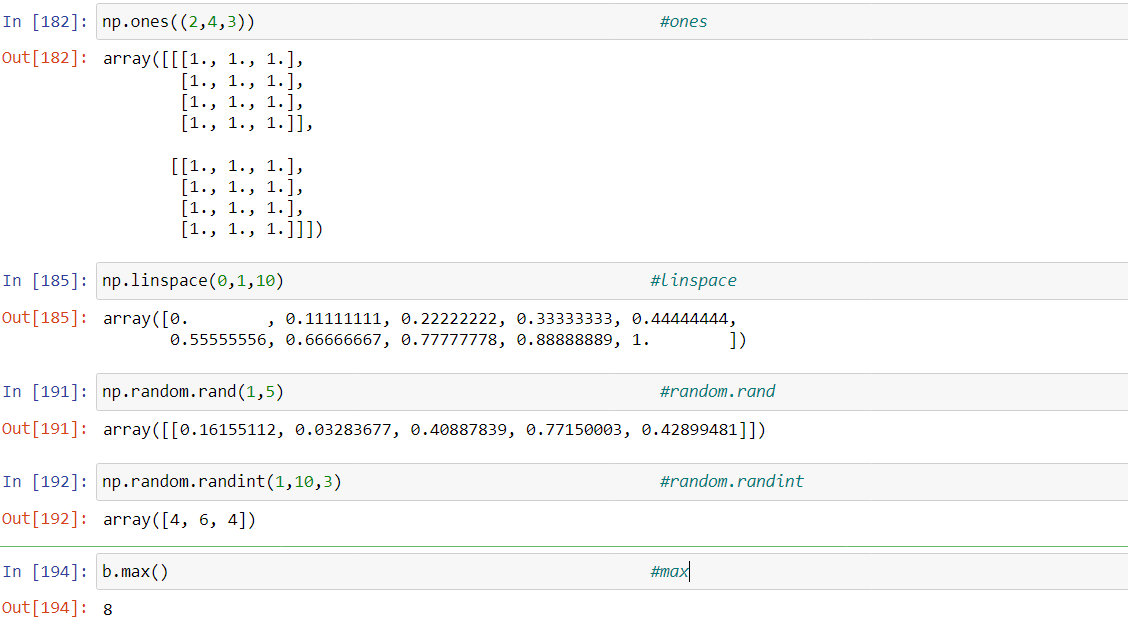
**np.random.randint():**

 This function is used to create an array with random integer values between a specified range.

**np.max():**

This function is used to find the maximum value in an array.





**argmax( ) and argmin( )**

functions are used to return the index of the maximum element along axis = 0 and axis = 1 , the elements from top to bottom along the column come under axis = 0 , whereas the elements from left to right along each row come under axis = 1

**np.reshape()**

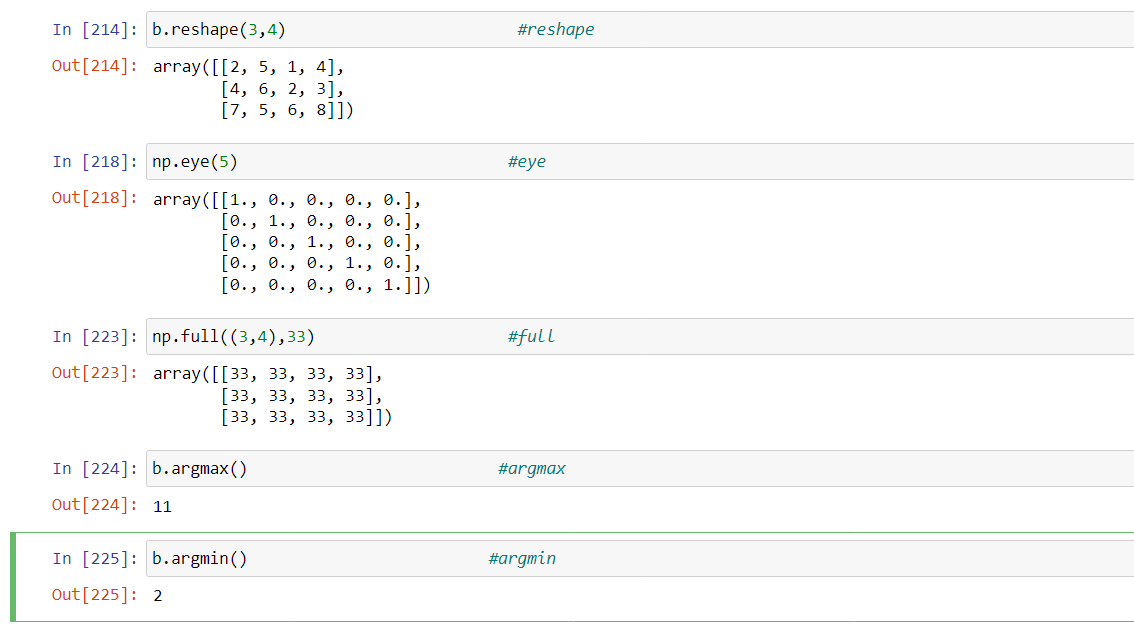
 Reshape is a method is used to reshape the matrix into required number of rows and columns. But, the total number of elements in the matrix should be equal to product of rows and columns.

**np.eye()**

Create a square N x N identity matrix (1’s on the diagonal and 0’s elsewhere)

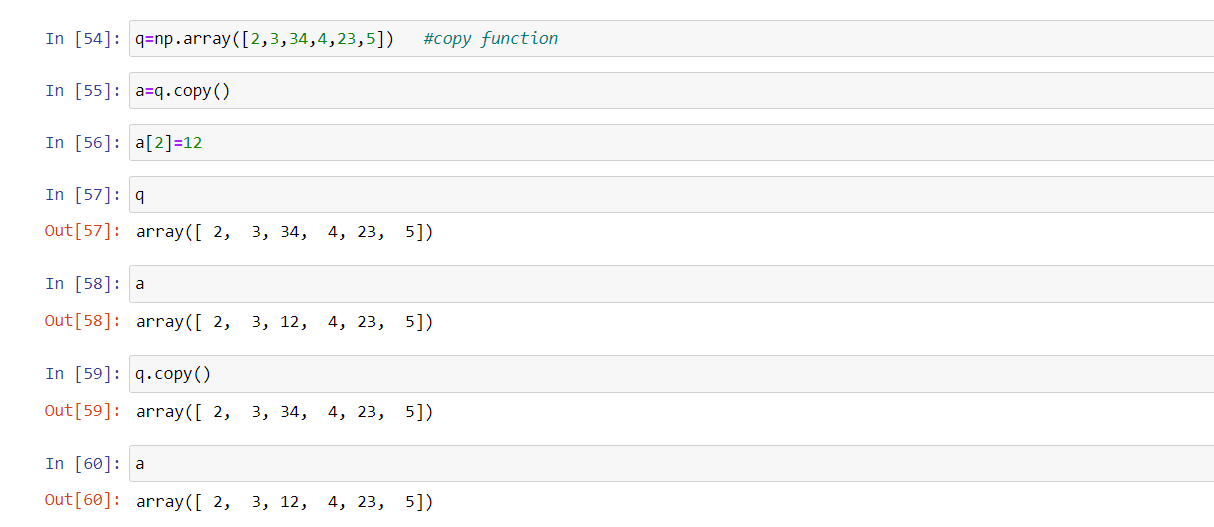
**np.full()**

**np.full(shape, fill\_value, dtype = None, order = ‘C’):**Return a new array with the same shape and type as a given array filled with a fill\_value.

****

**Copy()**

Use np.copy() function to copy Python NumPy array (ndarray) to another array. This method takes the array you wanted to copy as an argument and returns an array copy of the given object. The copy owns the data and any changes made to the copy will not affect the original array. Alternatively, you can also try ndarray.copy() function.

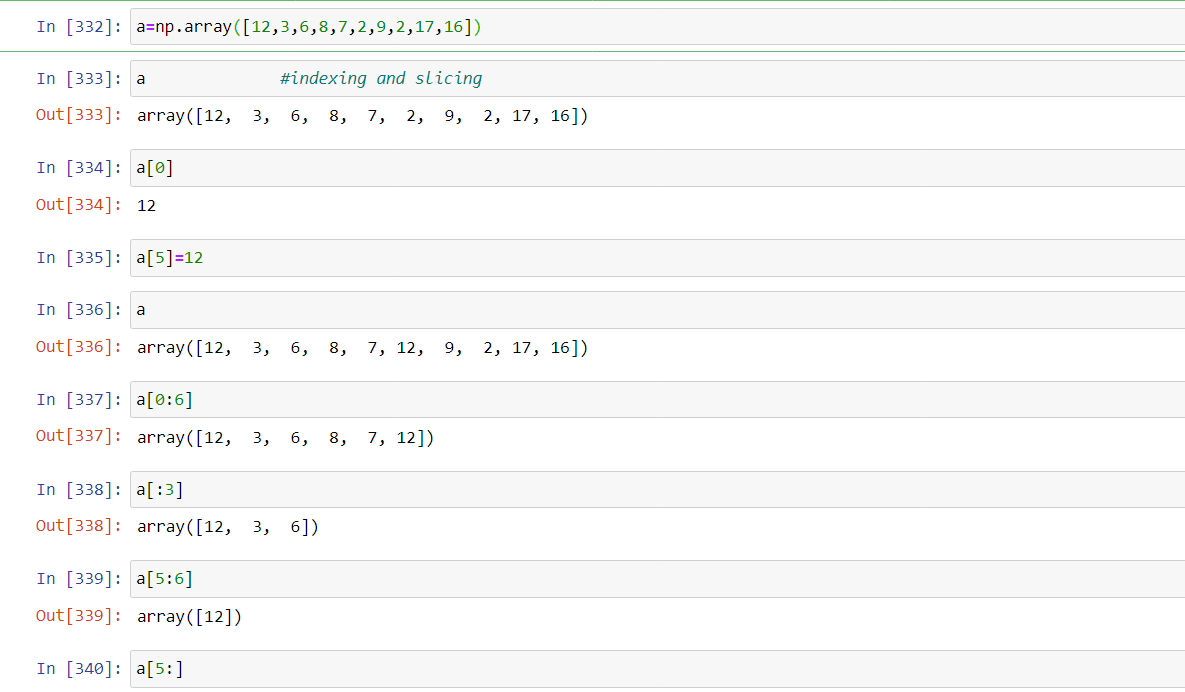
****

**Indexing and slicing**

Contents of ndarray object can be accessed and modified by indexing or slicing, just like Python's in-built container objects.

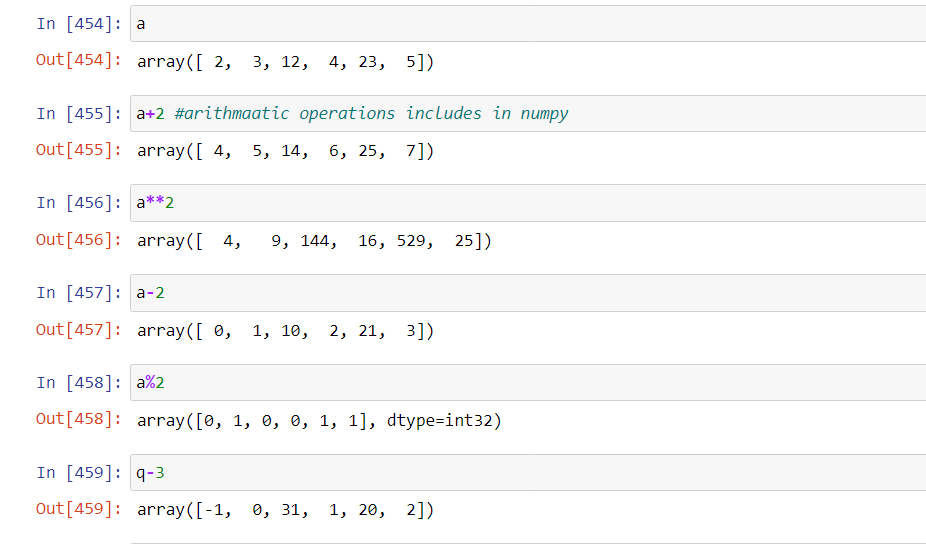
As mentioned earlier, items in ndarray object follows zero-based index. Three types of indexing methods are available − **field access, basic slicing** and **advanced indexing**.

Basic slicing is an extension of Python's basic concept of slicing to n dimensions. A Python slice object is constructed by giving **start, stop**, and **step** parameters to the built-in **slice** function. This slice object is passed to the array to extract a part of array.

****

# **Arithmetic Operations**

| **Function** | **Description** |
| --- | --- |
| Add | Add corresponding elements in arrays |
| Subtract | Subtract elements in second array from first array |
| Multiply | Multiply array elements |
| divide, floor\_divide | Divide or floor divide (truncating the remainder) |
| Power | Raise elements in first array to powers indicated in second array |

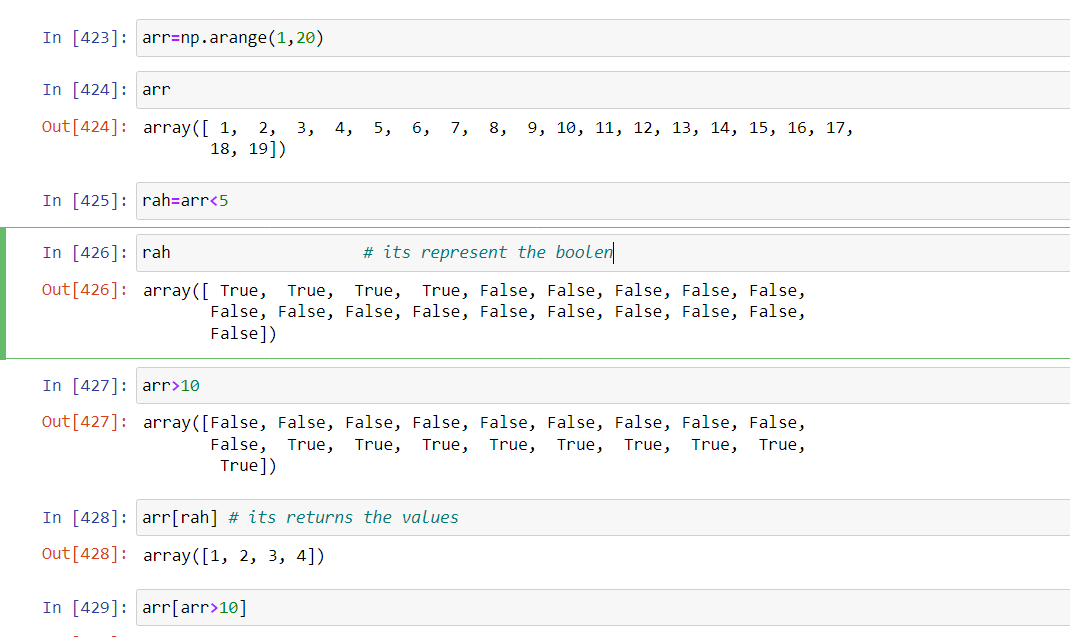
****

**Comparison operators in NumPy**

NumPy provides several comparison and logical operations that can be performed on NumPy arrays.

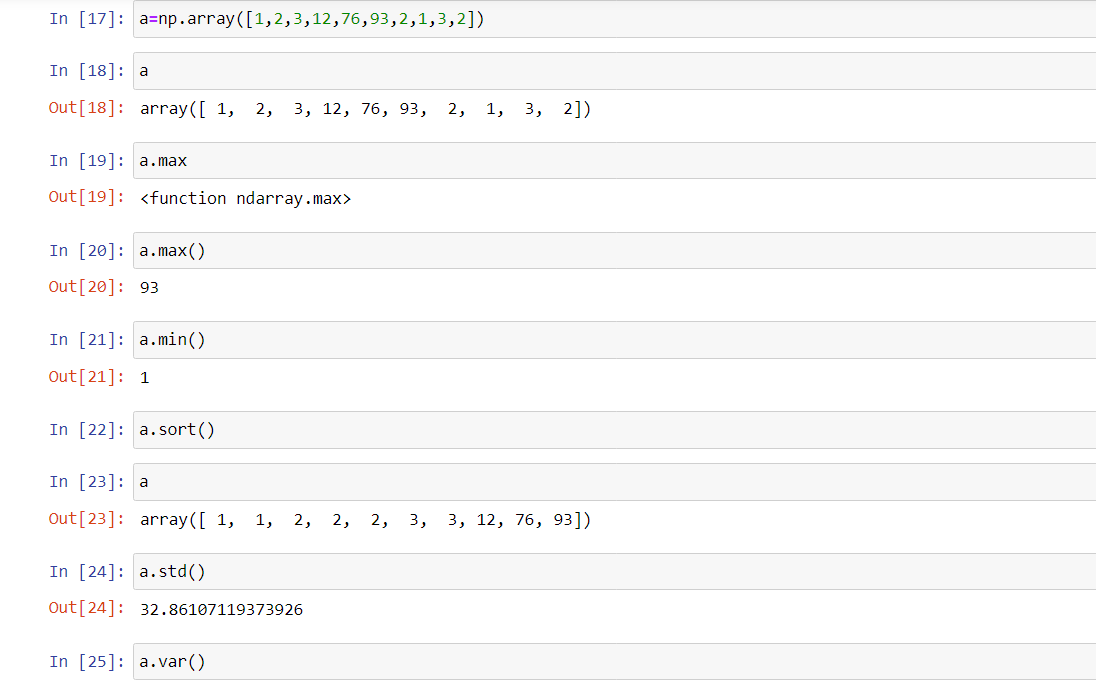
NumPy's comparison operators allow for element-wise comparison of two arrays.

Similarly, logical operators perform boolean algebra, which is a branch of algebra that deals with True and False statements.

****

**Functions**

|  |  |  |
| --- | --- | --- |
| np.sum | np.nansum | Compute sum of elements |
| np.prod | np.nanprod | Compute product of elements |
| np.mean | np.nanmean | Compute mean of elements |
| np.std | np.nanstd | Compute standard deviation |
| np.var | np.nanvar | Compute variance |
| np.min | np.nanmin | Find minimum value |
| np.max | np.nanmax | Find maximum value |

****

****