# *ASSIGNMENT 1*

**NUMPY:**

1. Getting Started With Numpy:

Numpy stands for Numerical Python. It is essentially used in Python for scientific computing and has been highly useful in the math library for making numerical computation containing statistics, linear and non-linear equations. Numpy packages provides the following operations:

* A powerful N-Dimensional Array Object.
* Sophisticated Functions.
* Fourier Transforms, Linear Algebra, Random Numbers and Routines for shape manipulation.

1. Data Manipulation with Numpy:

* Installation of Numpy:

Using Conda, we can install it with:

conda install numpy

Using Pip, we can install it with:

pip install numpy

* Creating and Saving Numpy ndarray:

1. Importing Numpy:

**Code:**

****

1. Creating 1-Dimensional(1D) ndarray and 2-Dimensional(2D) ndarray:

**Code & Output:**

****

1. Using Built-in Functions on ndarrays:

List of Built-in Functions and Workings:

* np.array(): We can create the Numpy ndarray object by using the array() function.

**Code & Output:**

****

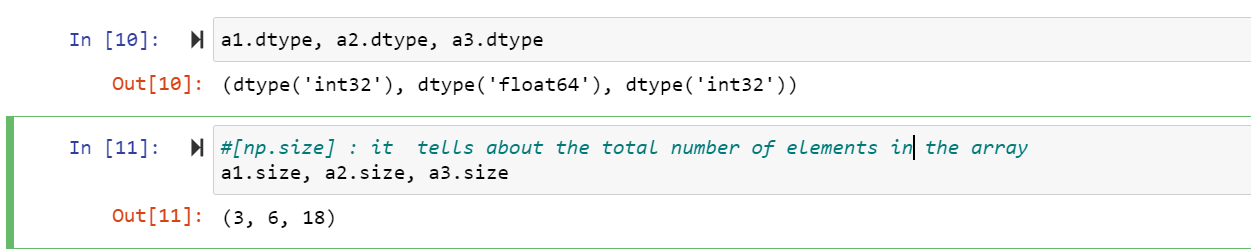
* np.shape: It’s used to indicate the dimension of an array/list.

**Code & Output:**



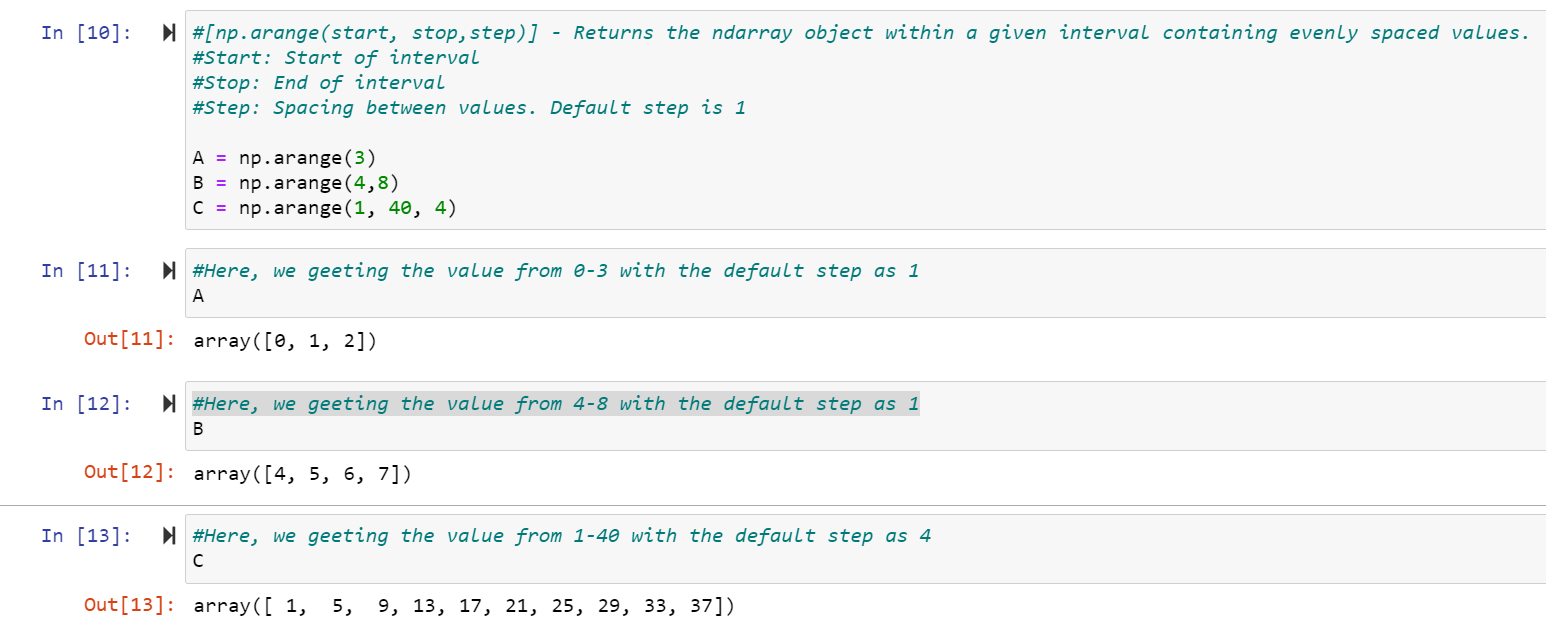
* np.dtype: It’s tells us the type of data stored in the array.
* np.size:

**Code & Output:**

****

* np.arange(start,stop,step):

**Code & Output:**

****

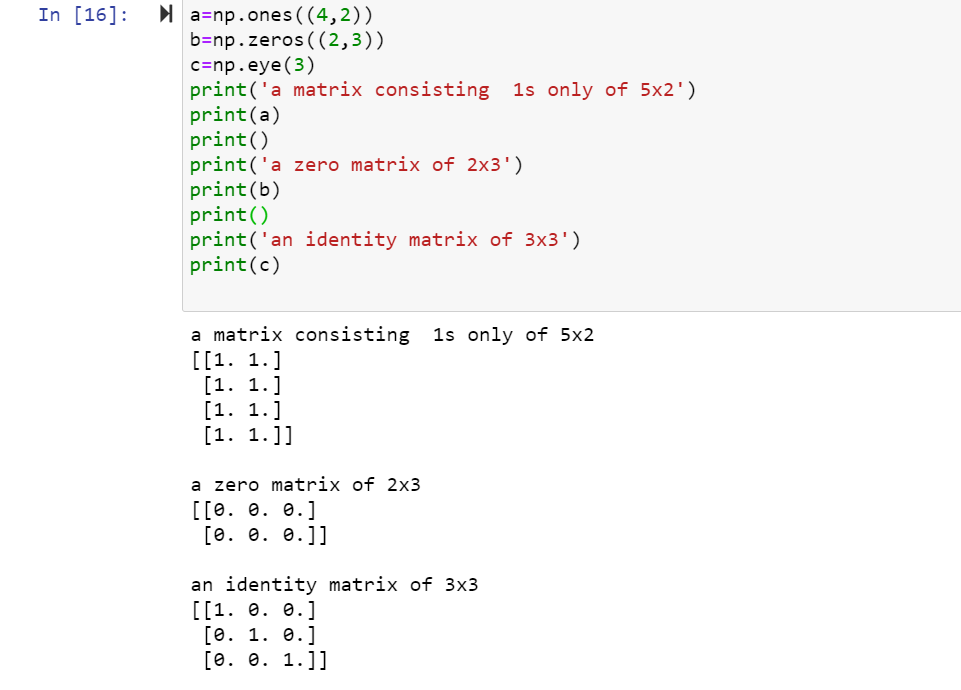
* np.zeros(), np.ones(), np.eye():

zeros()- It’s used to create an array of specified length with all elements as 0.

ones()- It’s used to create an array of specified length with all elements as 1.

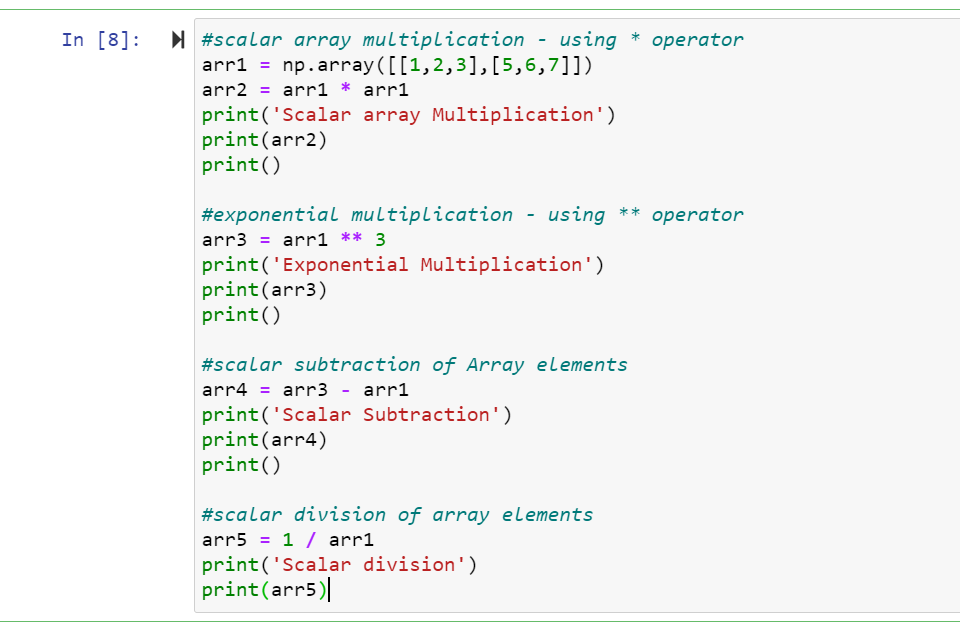
eye()- It’s used to create an Identity matrix of a specified length.

**Code & Output:**

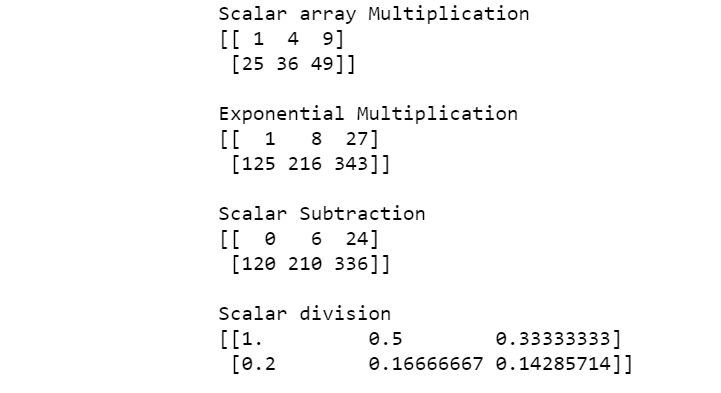
****

1. Scalar Operations on ndarrays: Matrix Multiplication, Subtraction, Division and Exponential Multiplication.

**Code:**

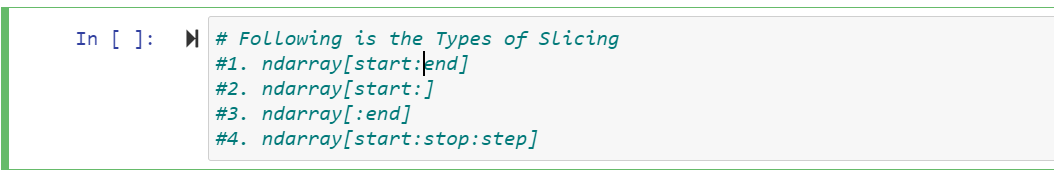
****

**Output:**

****

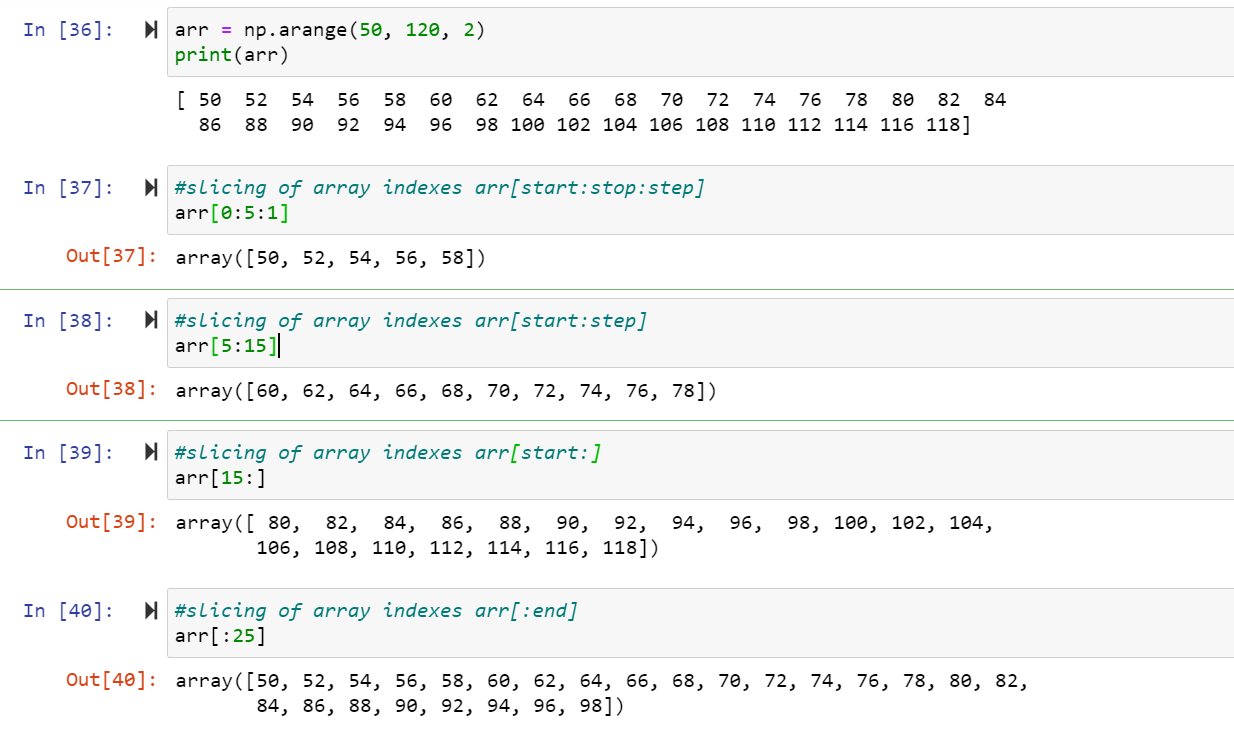
1. Array Indexing in 1-Dimension and multi-dimension ndarrays: It is possible to access elements using indices within square brackets []. In order to access individual (single) elements at a time, Numpy provides a method known as slicing.

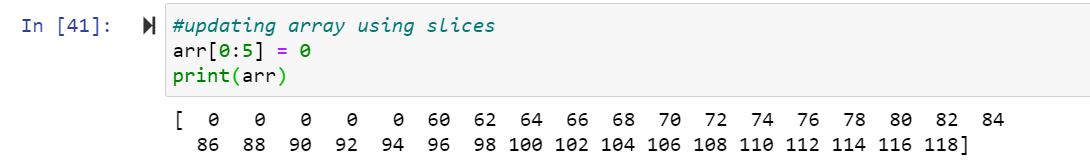
**Code:**

****

Indexing Numpy array(1-D):

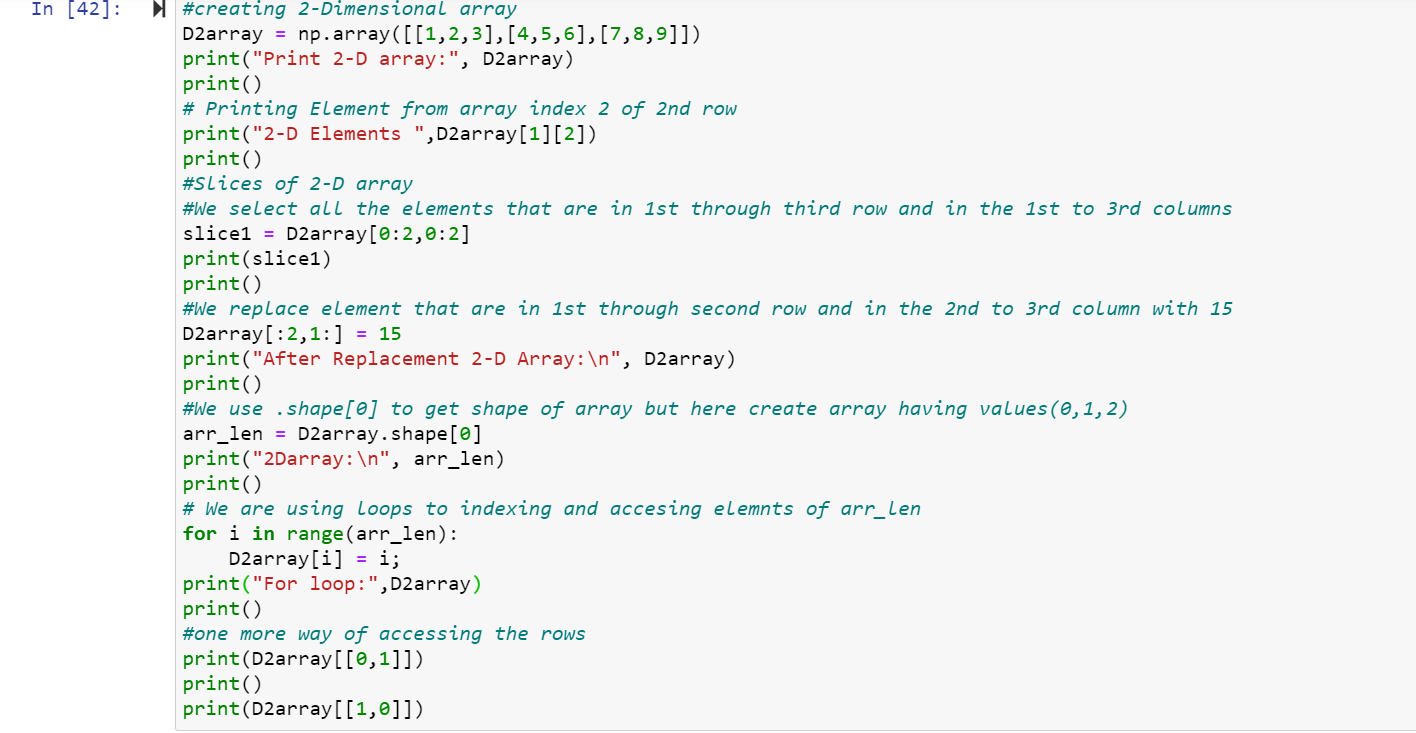
**Code & Output:**

****

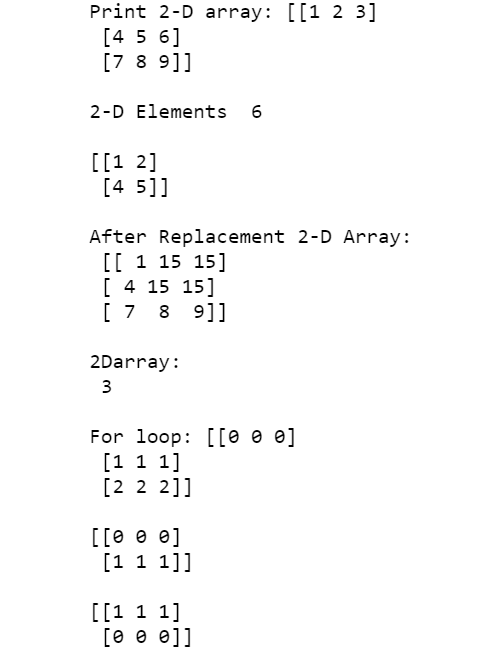
****

Indexing Numpy array(2-D):

**Code:**

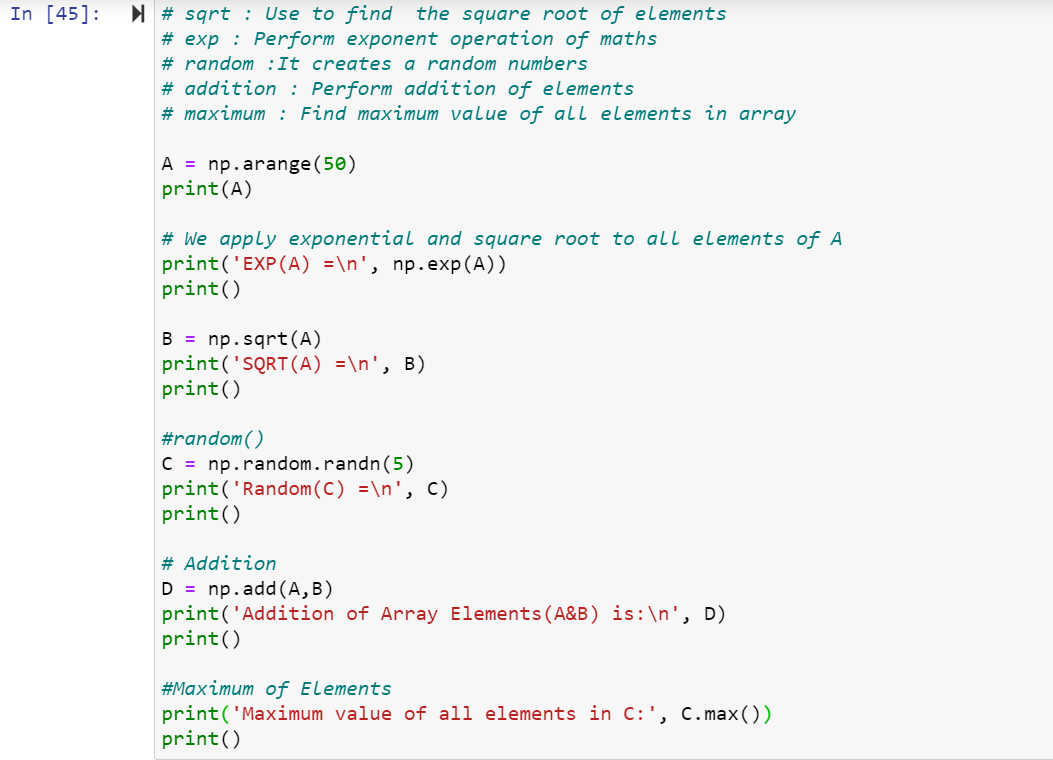
****

**Output:**

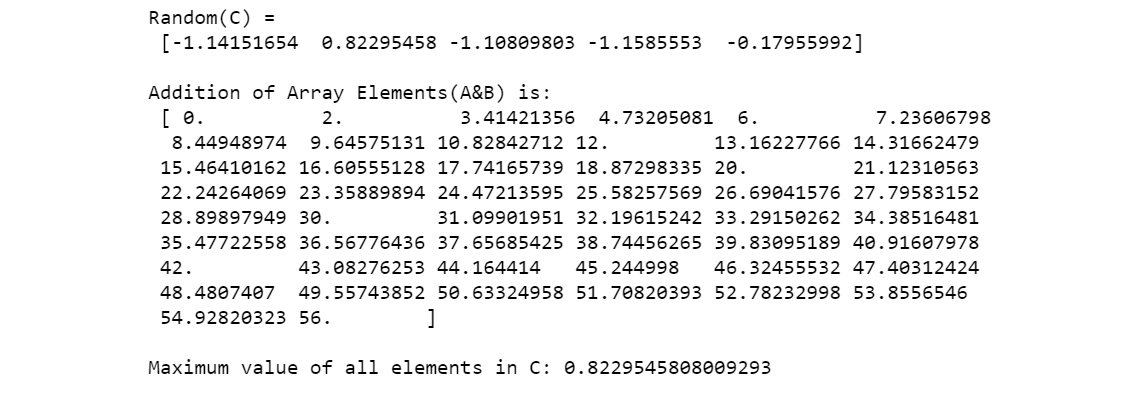
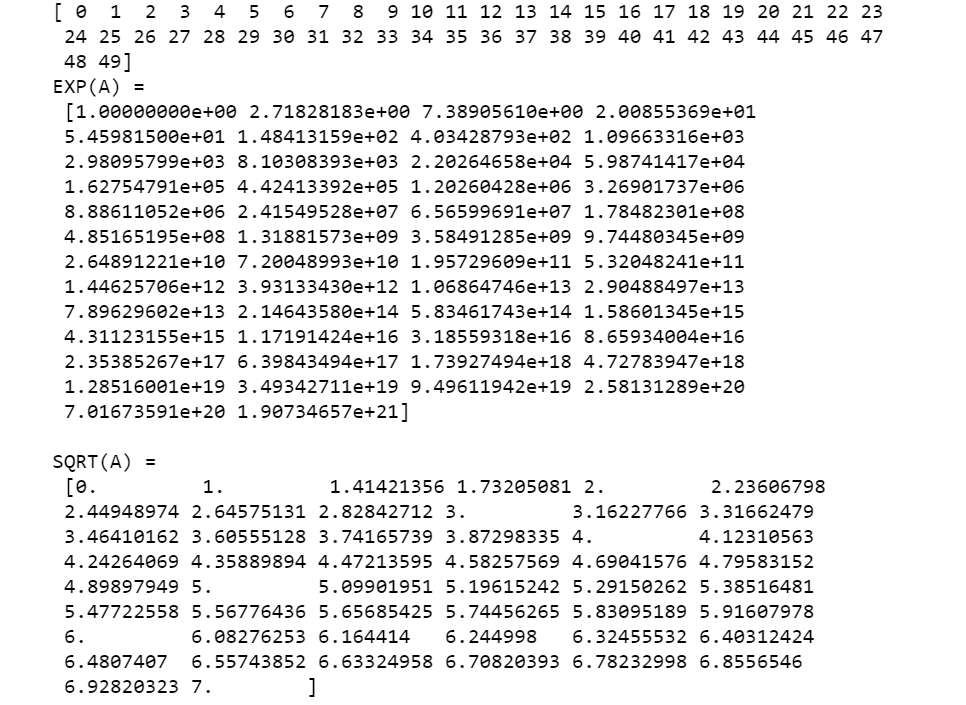
****

1. Premium Array Operations:

**Code:**

****

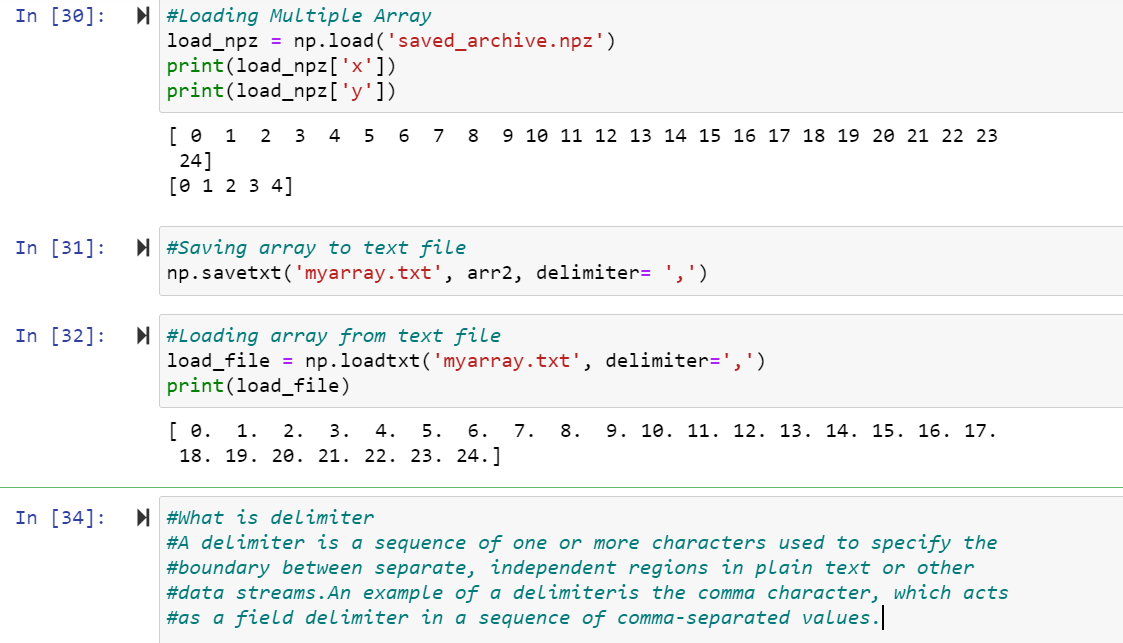
**Output:**

****

1. Saving and Loading Arrays to external Memory:

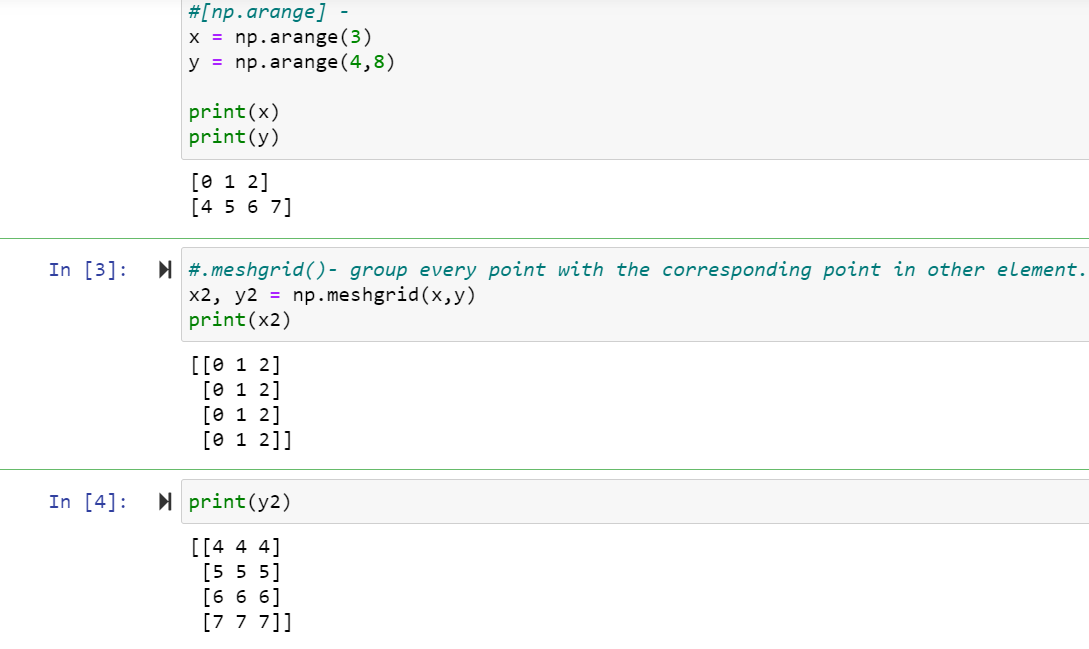
**Code & Output:**

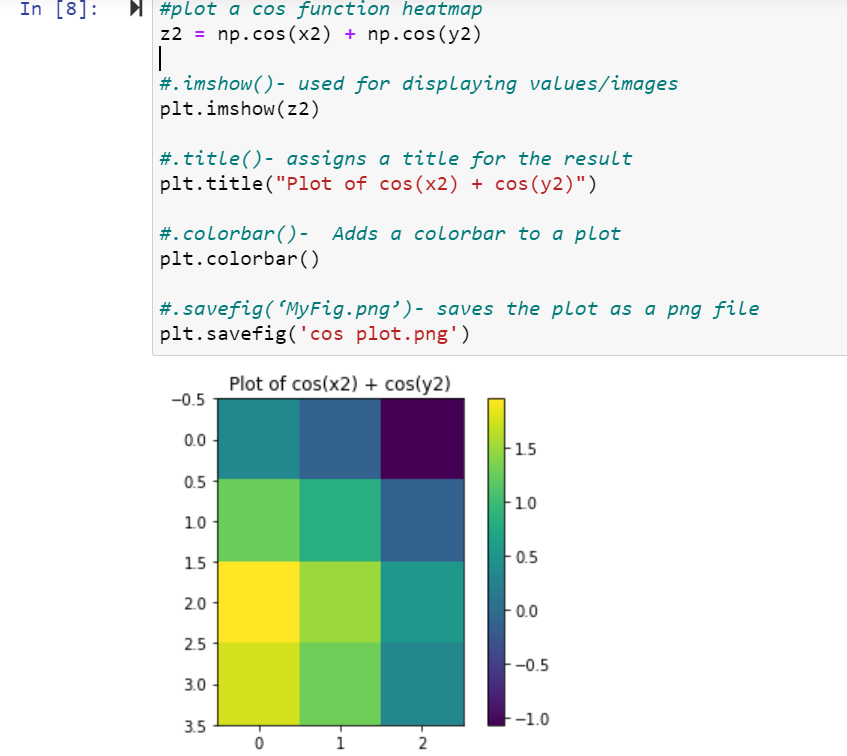
****

****

1. Statistical Processing and Sketching Graph: We perform some statistical and other mathematical operations on Array and present it in the form of graphs. We use a new library named ‘matplotlib’ for graphical uses and representations.

**Code & Output:**

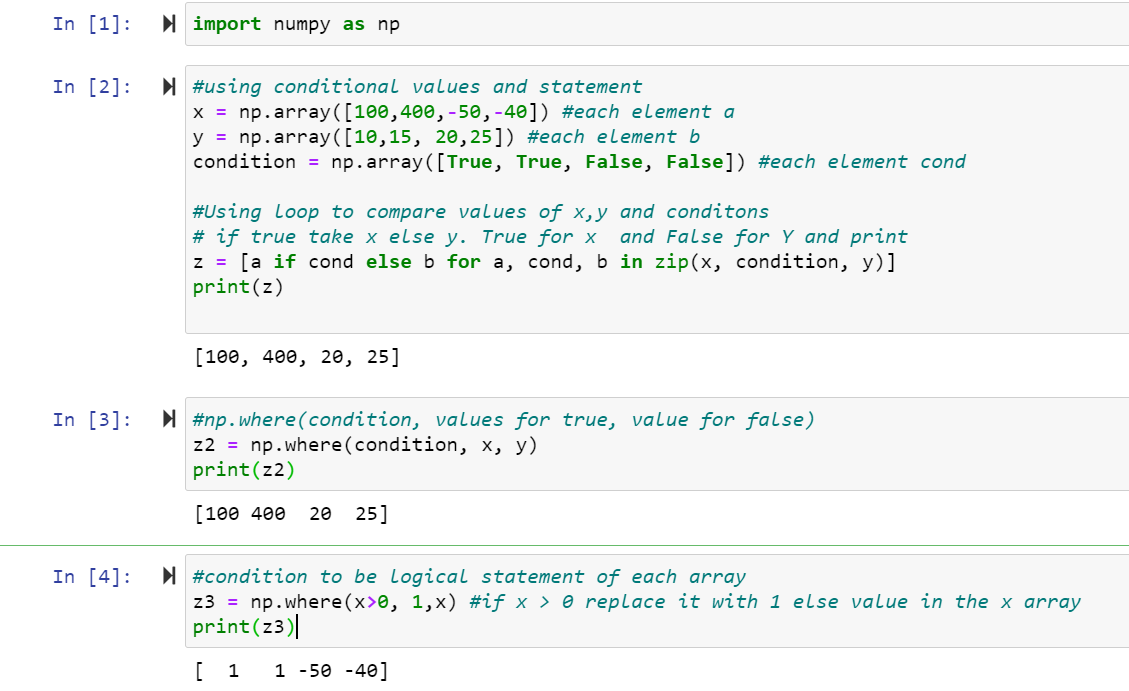
****

****

1. Conditional Clauses, Functions and Boolean Operation:

Conditional Clauses:

**Code & Output:**

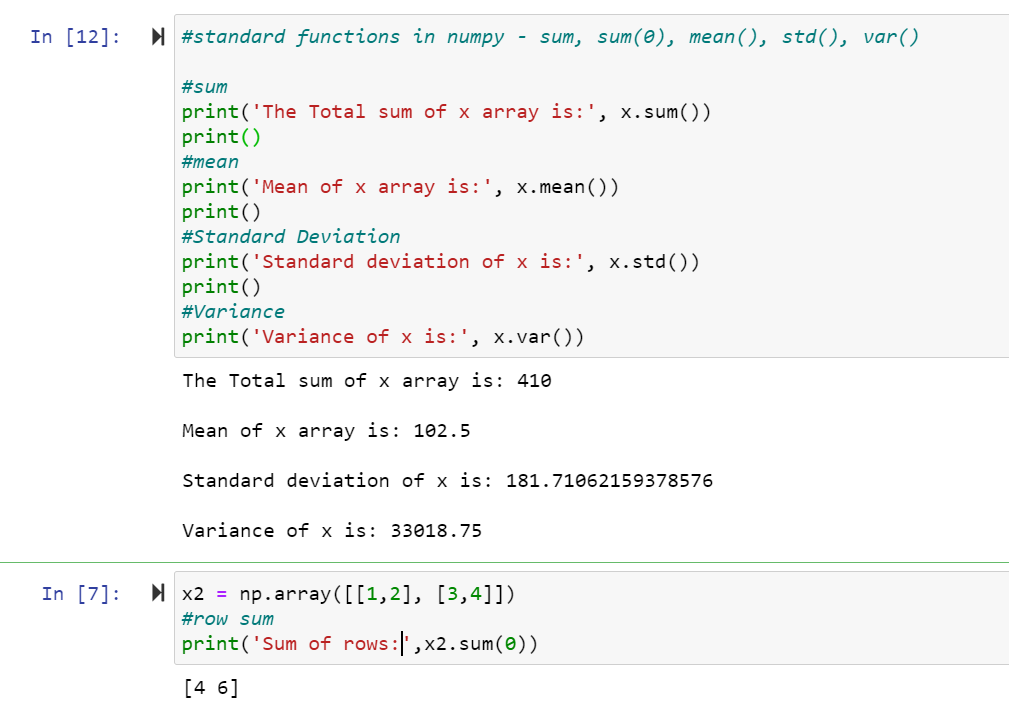
****

Disadvantages of using Loop to compare:

* It’s very confusing.
* It’s includes very long statement.
* It’s will be tedious and difficult to maintain once the start using more than 2 variables

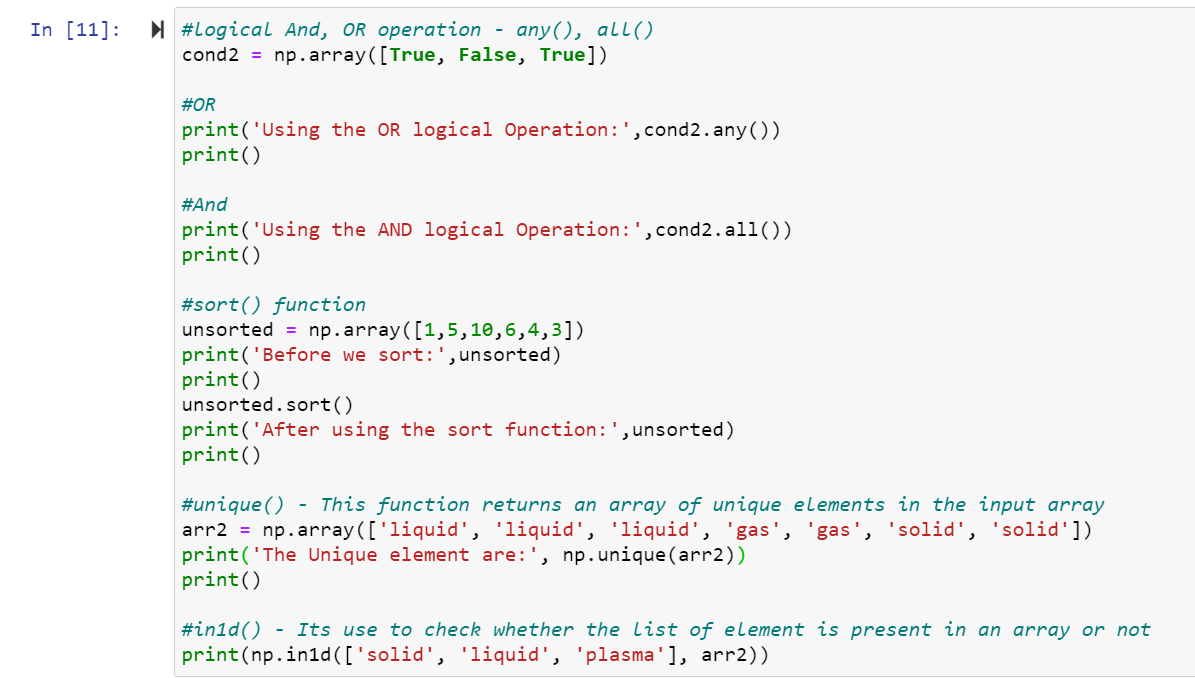
Functions:

**Code & Output:**

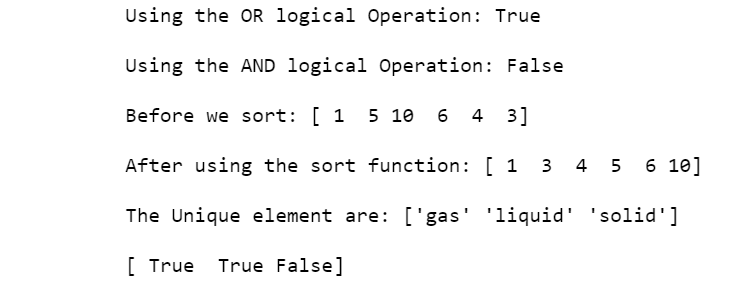


Boolean Operation:

**Code:**

****

**Output:**

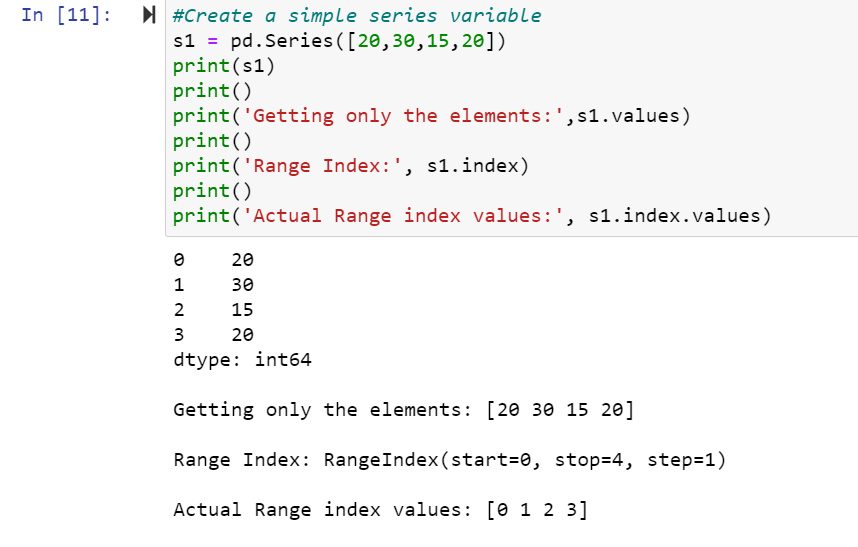


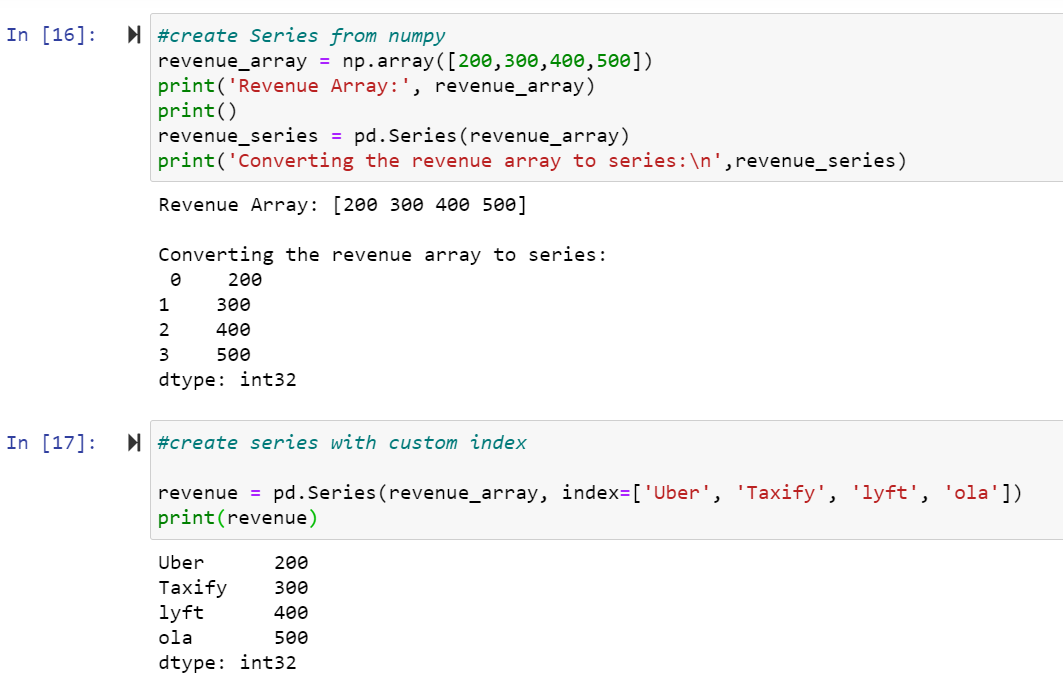
**PANDAS:**

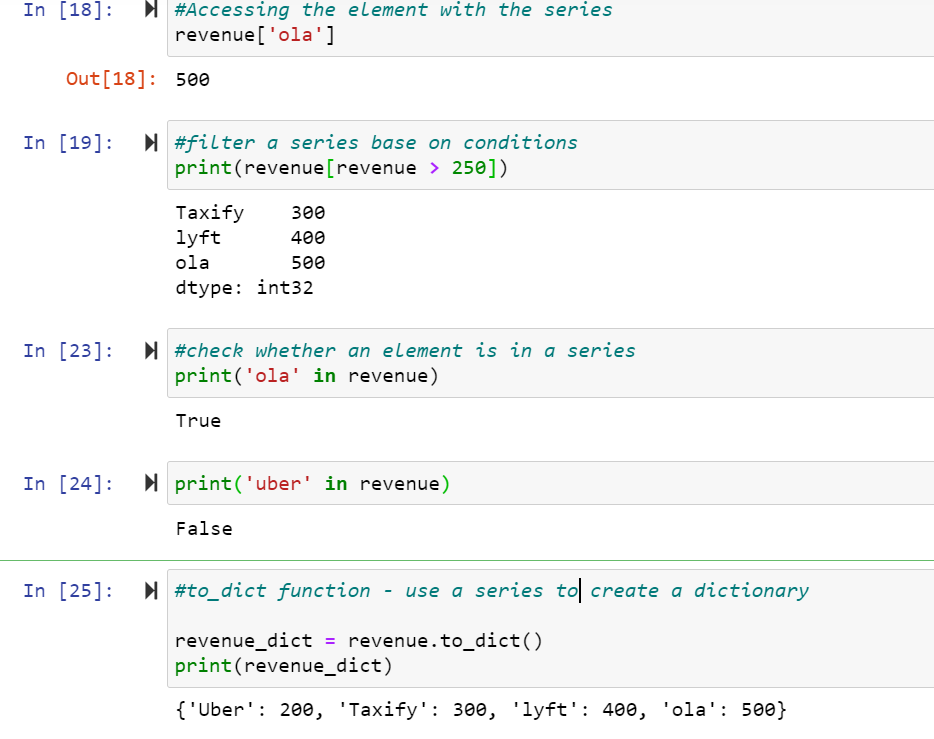
Pandas is a fast, powerful, flexible and easy to use open source data analysis tool and high level manipulation tool provided by the developers of python which is built on the numpy library and extends numpy functionality. It provides us with 2 key data structures which are called:

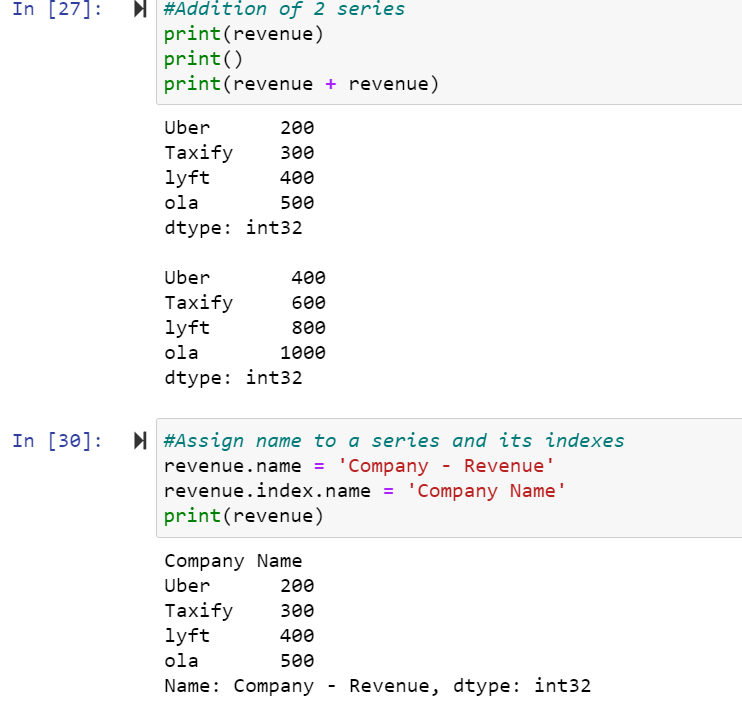
* Series
* DataFrame
* Working with Pandas Series: It’s a 1 Dimensional data array which is capable of holding any type of data.

**Code & Output:**

****

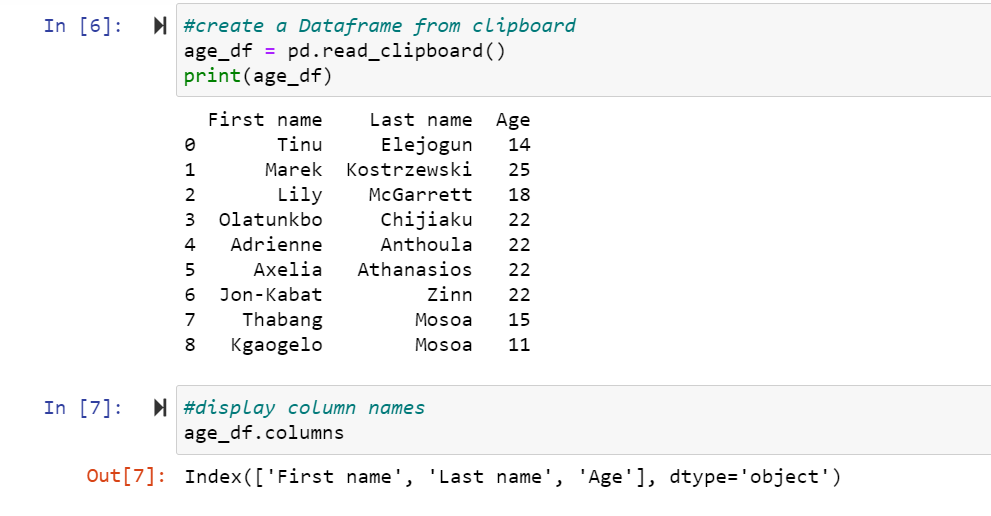






* Working with Pandas DataFrame: It’s a 2 Dimensional data structure with 3 principal components I.e Rows, Columns and Value. Data are aligned in a tabular fashion in rows and columns.

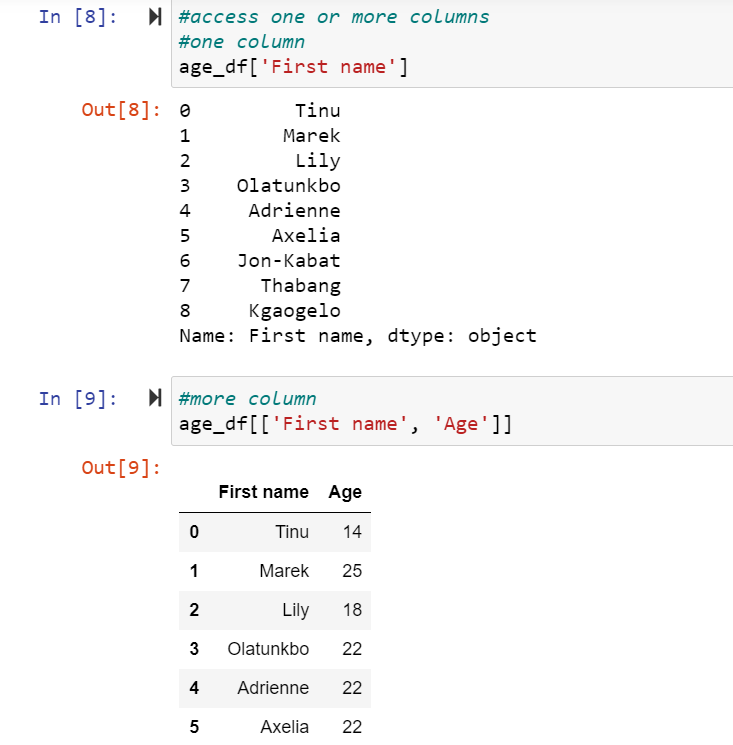
**Code & Output:**

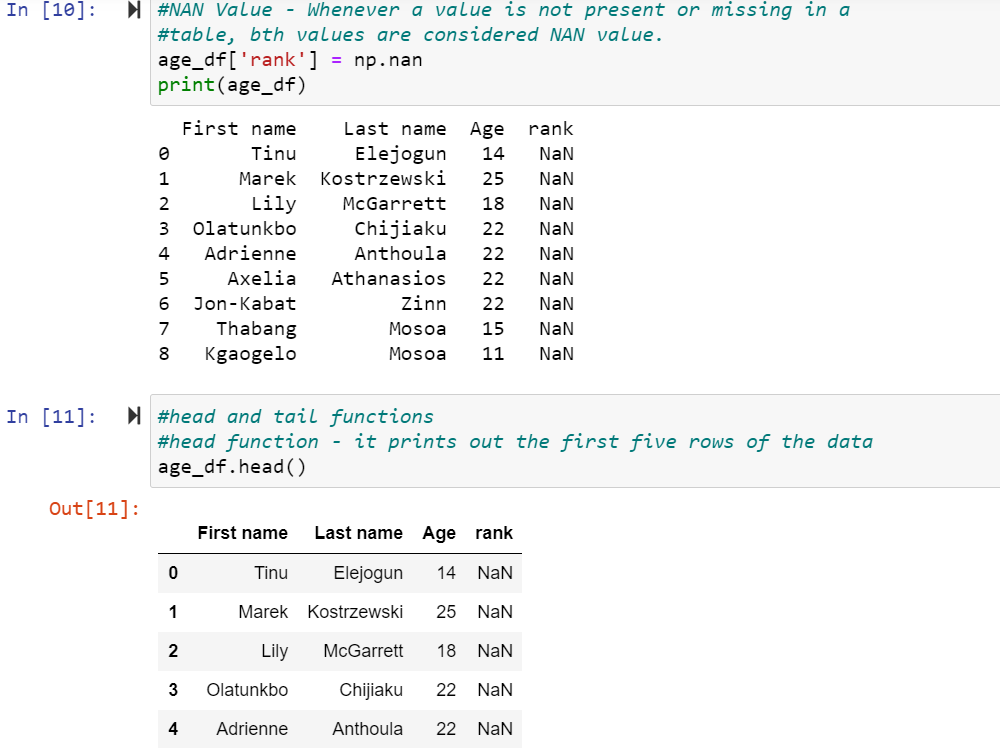


read\_clipboard()- read the value for the data frame copied on the clipboard.

df.columns- displays all the columns in the dataframe.

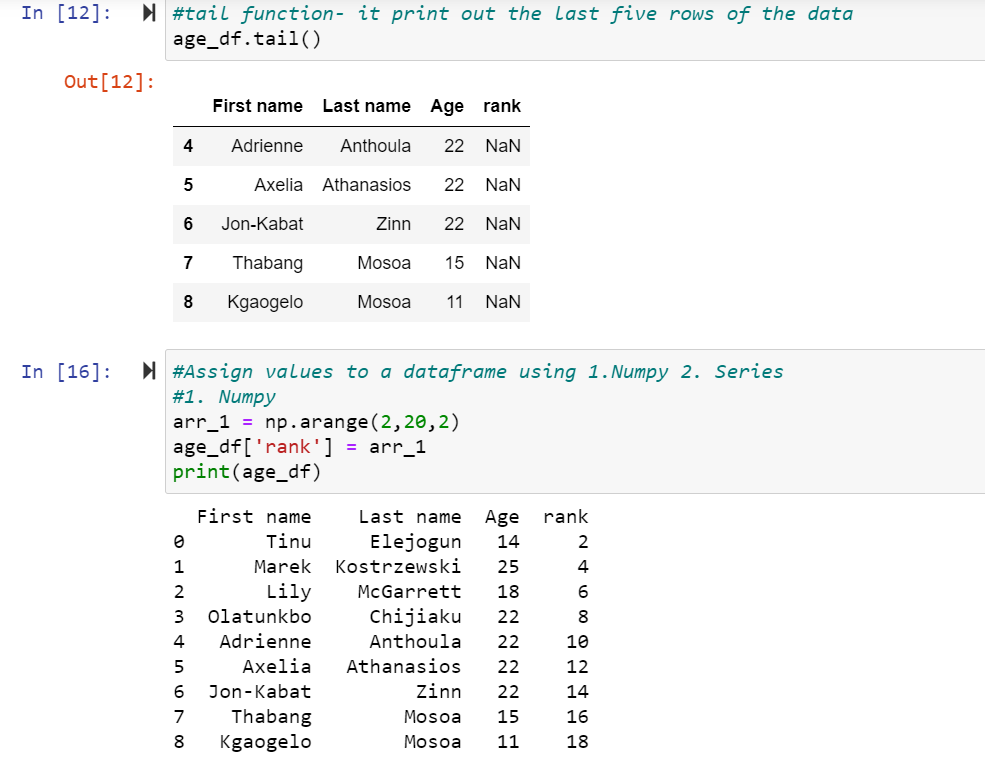
Individual column or multiple columns can be accessed by specifying the name of the columns.





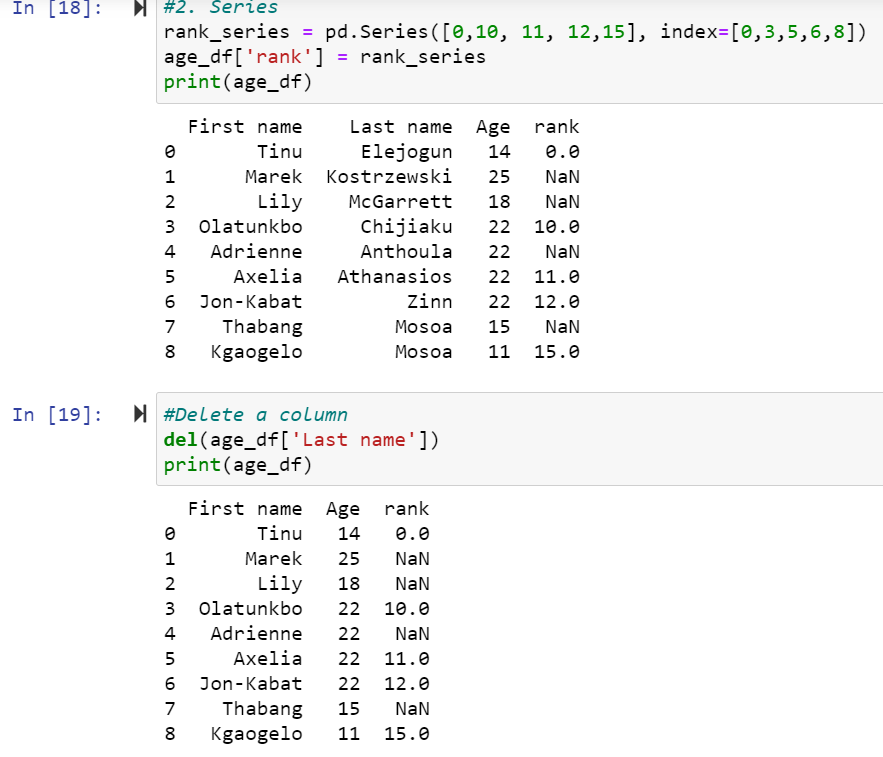
df.head() - It prints out the first five rows of the data.

df.tail() - It print out the last five rows of the data.

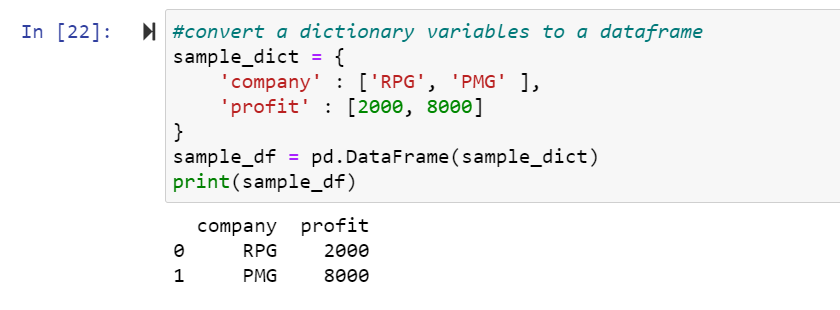


Assigning values to a dataframe using Numpy: The above code assigns the values of the array arr\_1 to the rank column.

Assigning values to a dataframe using Series: The code below assigns value 0,10,11,12,15 to index 0,3,5,6,8.



The code below converts a dictionary sample into a dataframe. This can be done by passing dict in the DataFrame().

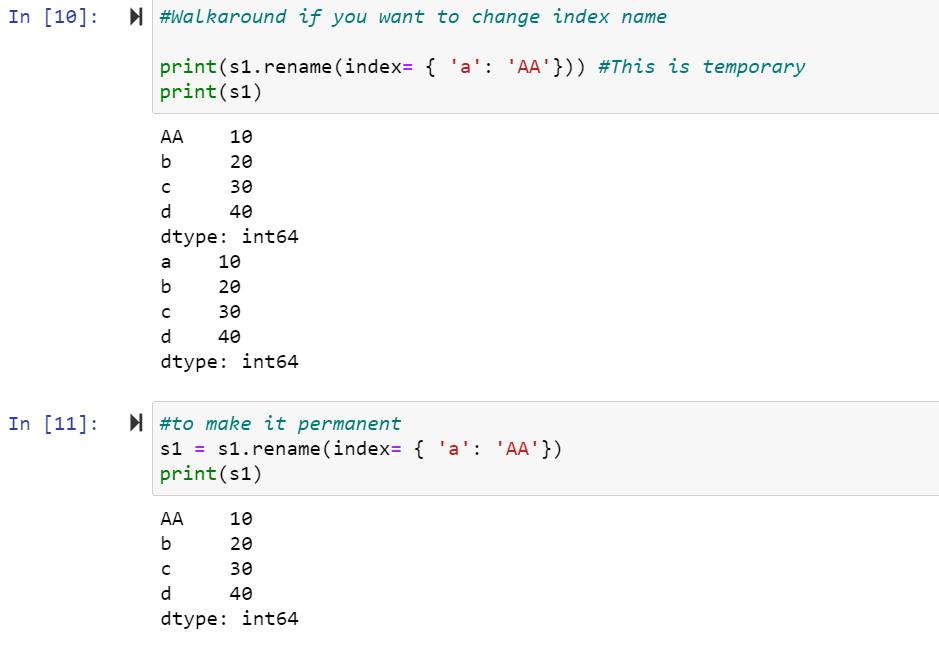


* Indexes in Pandas: In both Series & DataFrames structure of pandas we use index to refer to the row&column. Indexes are immutable.

**Code & Output:**

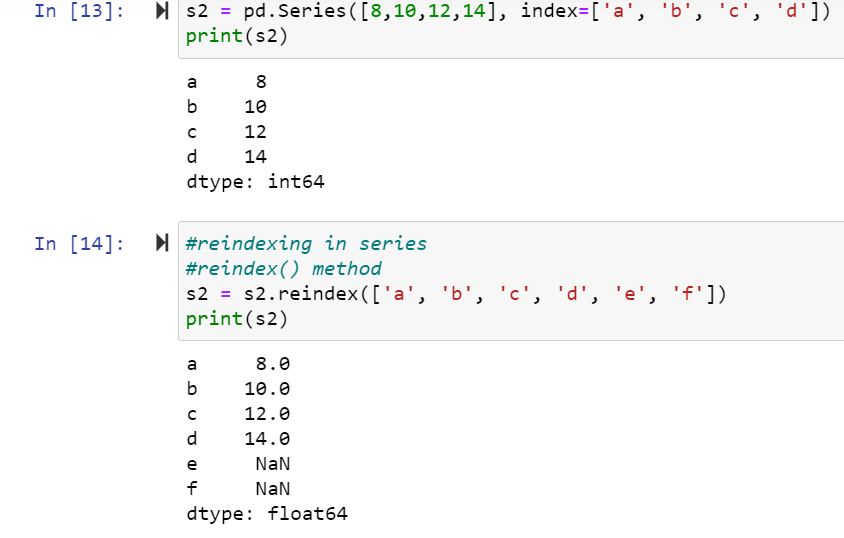


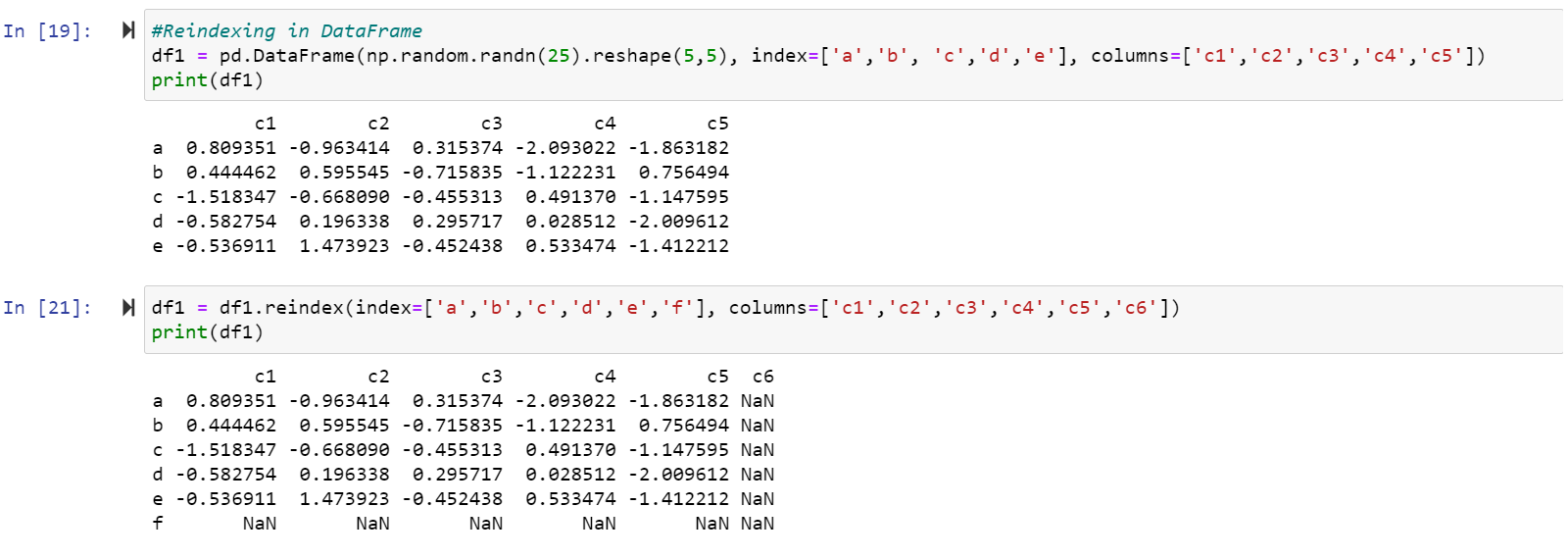




* Reindexing in Pandas Series & DataFrames: Reindexing allows to adjust row column labels and dataset data without changing the internal structure.

**Code & Output:**

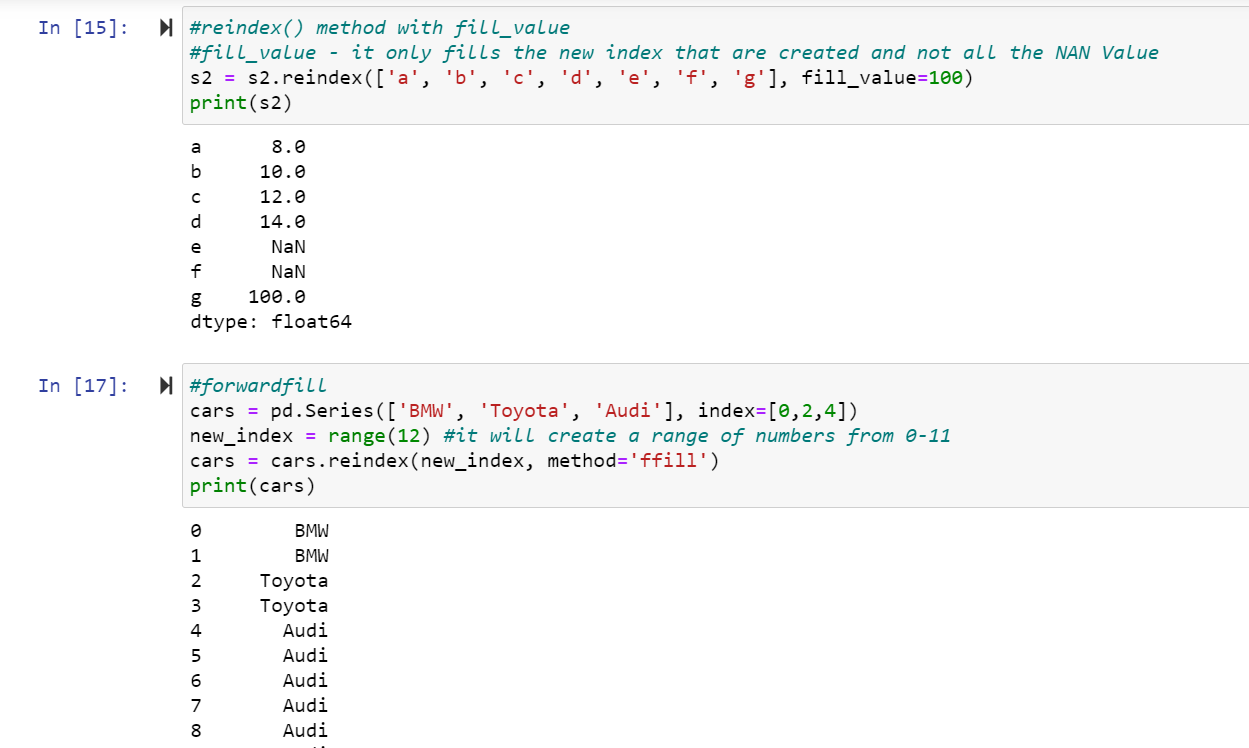




np.randn(n) - It provides n random numbers.

np.reshape()- It reshapes the given value into rows and columns.

The above code creates a new dataframe using random numbers. We can also re-index the column values using reindex().



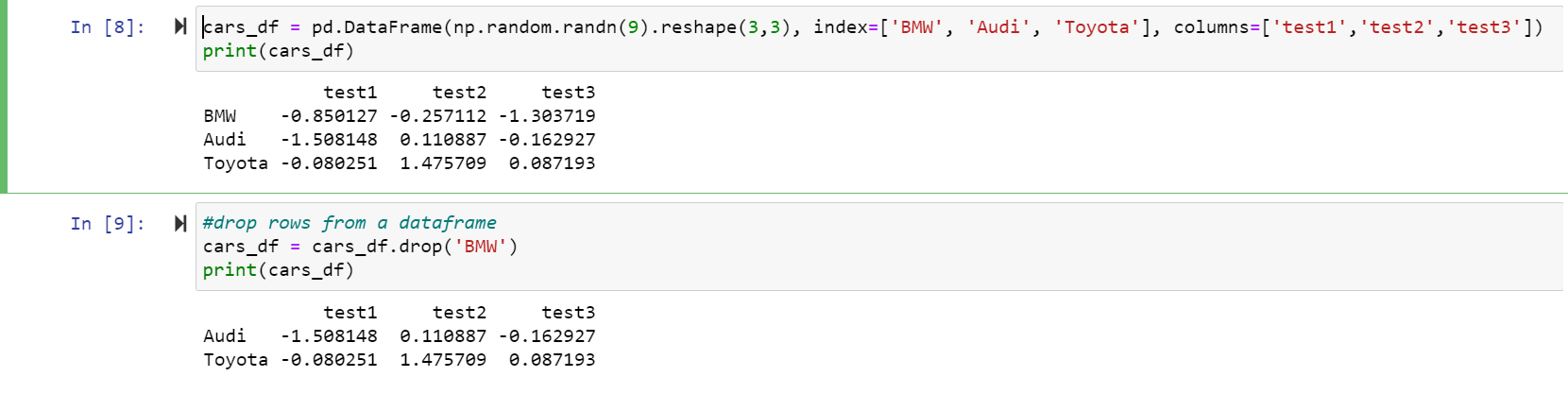
fill()- fills in the given value to the new index that are created and not all the NAN value.

ffill()-Forward fill(ffill) is used to fill in the missing values.

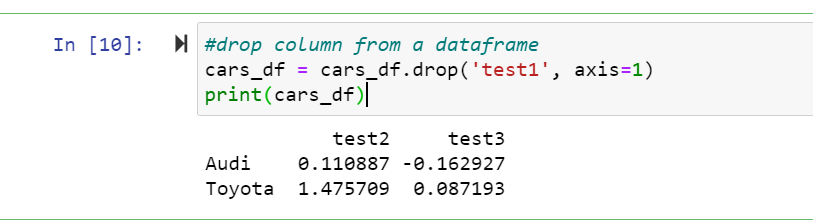
* Dropping entries from Data types: Dropping missing values in pandas or NAN values can be done under different situation, it is important because it generates complexity and incomplete data to work on.

**Code & Output:**



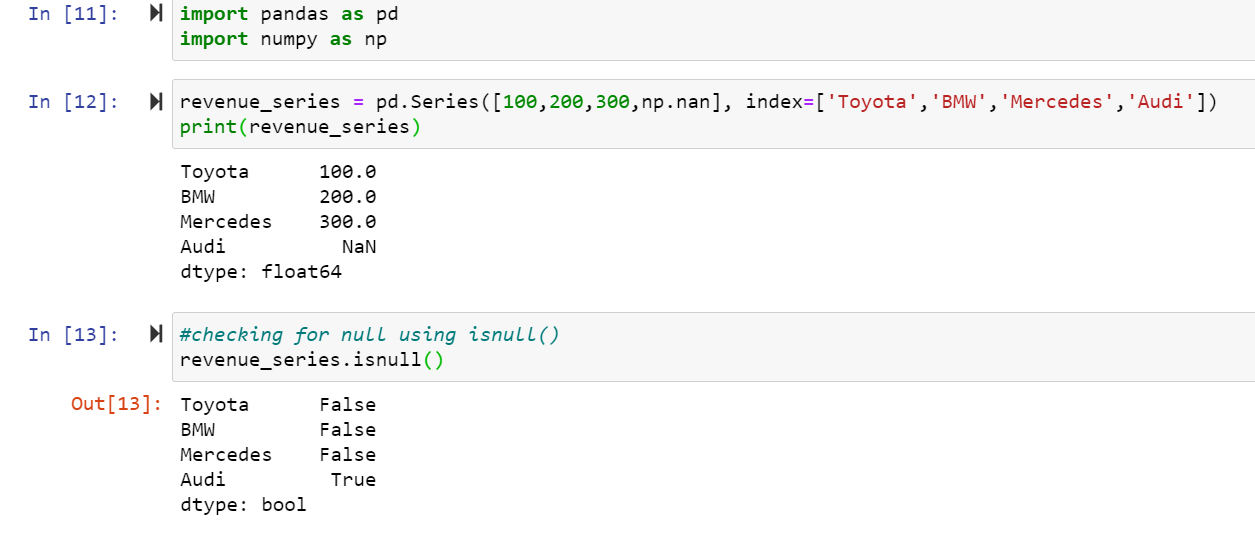


If you want to use the drop() function you need to indicate the axis and the default is axis equals to zero (axis = 0) which you don’t need when you want to drop a row from a dataframe, but to drop a column from a dataframe you need to set your axis equals to one (axis = 1) which is displayed in the code below.

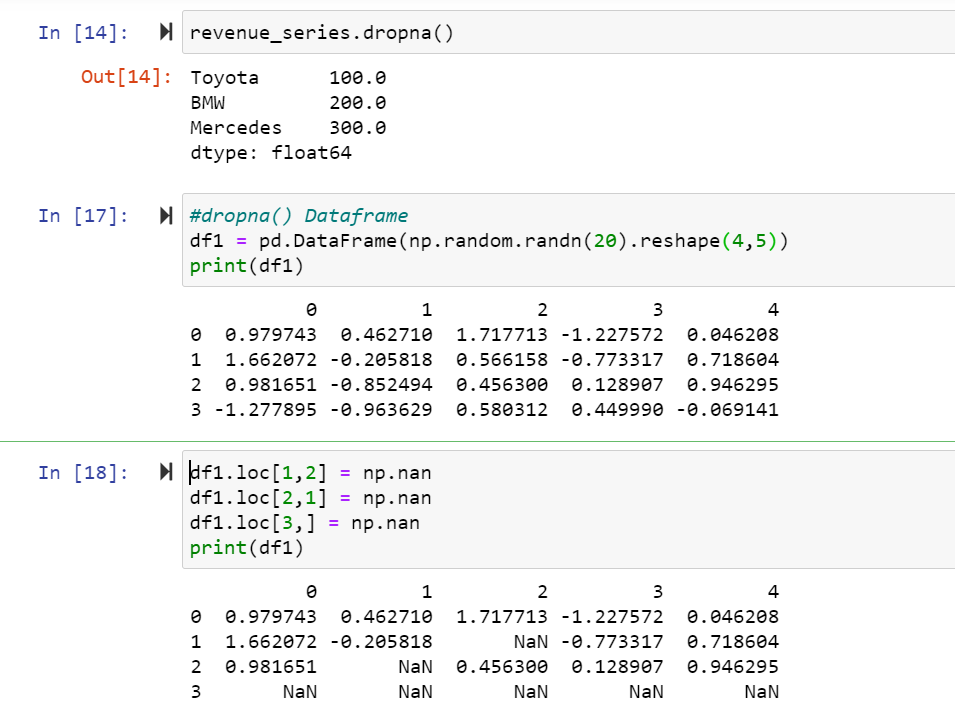


* Handling Null Data in Pandas:

**Code & Output:**



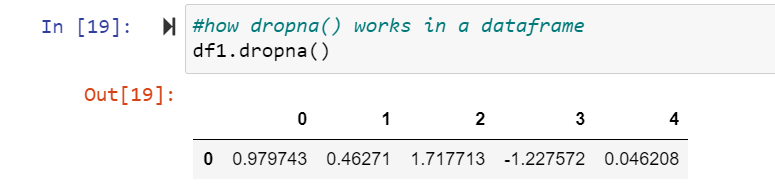
.isnull() - It’s used to check for NAN/Null values in a series or dataframes.



Series.dropna() - It’s used to drop a NAN/Null value.

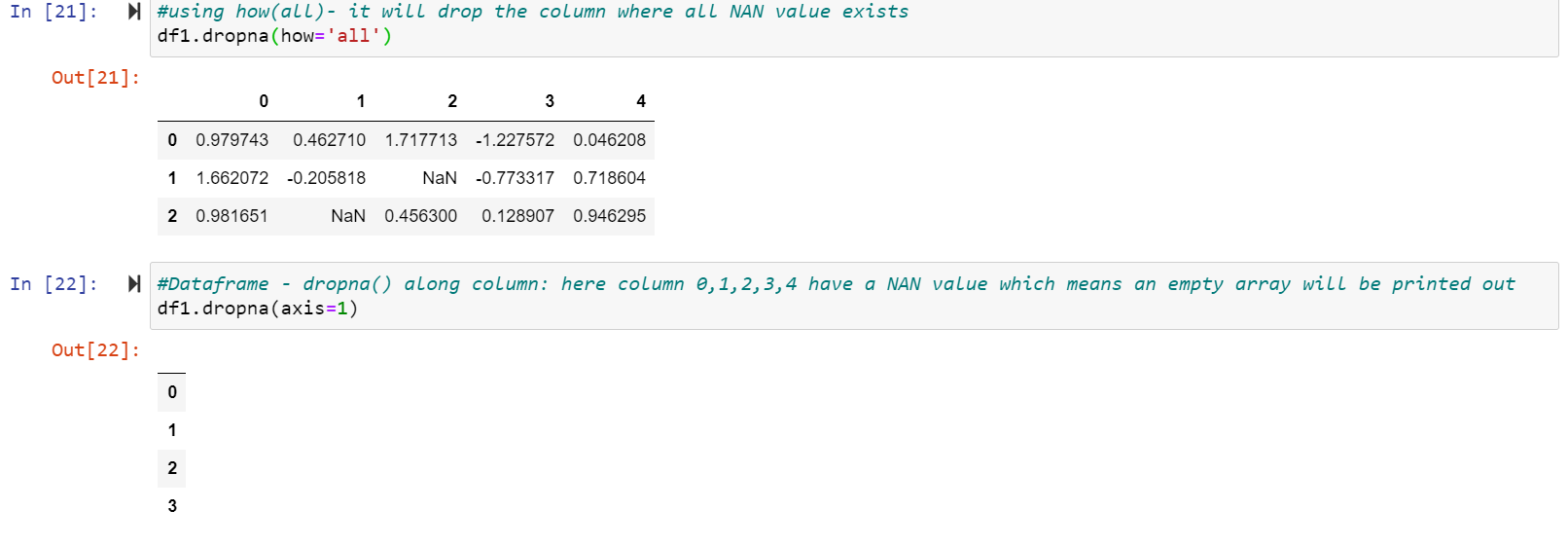
.loc[] - It’s used to select specific element of a dataframe and assign values to them.

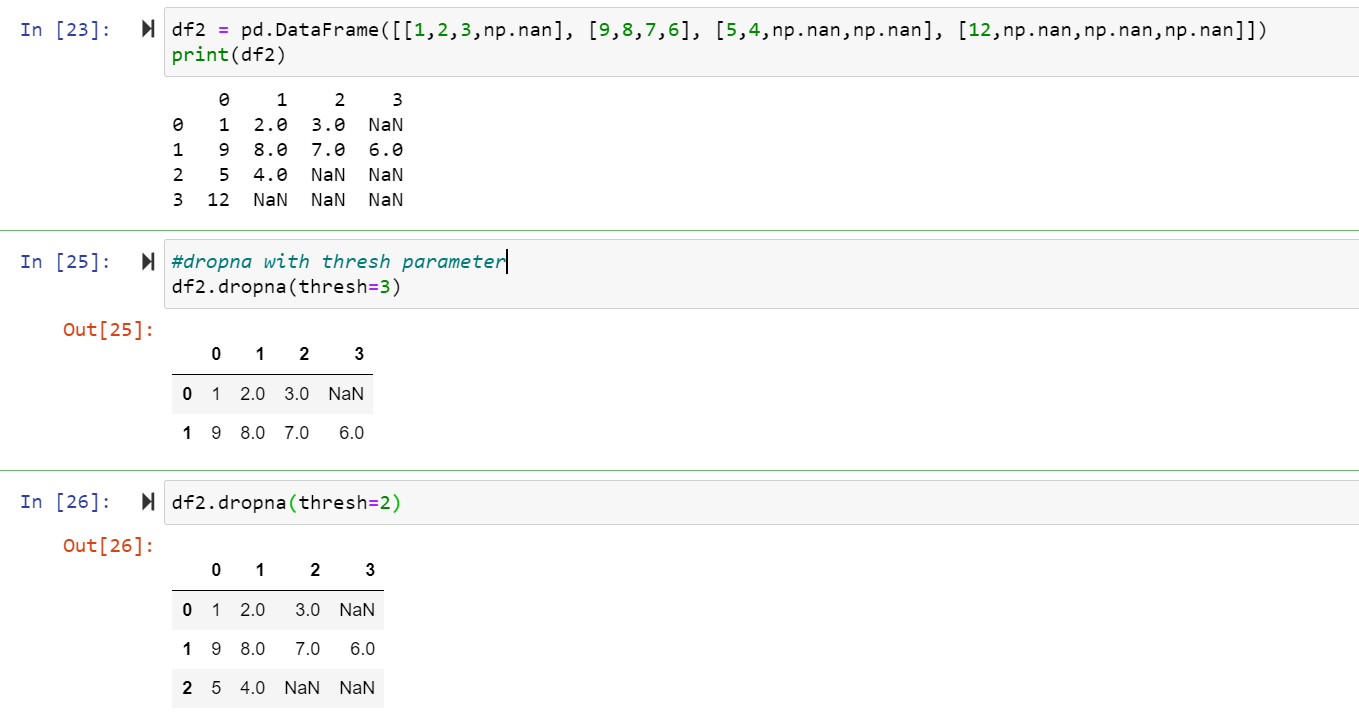
From the code above, a dataframe was created and we added some NAN values, so the code below shows us implementing a dropna() function which removed all the rows containing an NAN value.



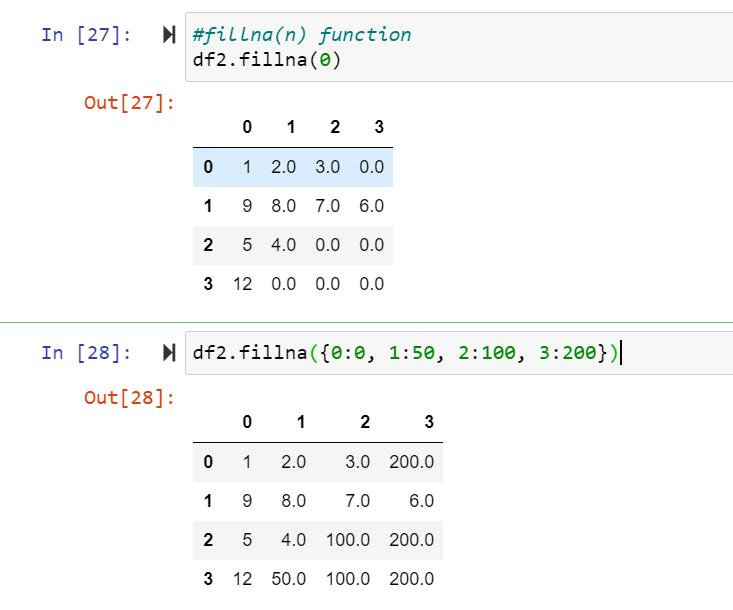
Disadvantages of dropping all NAN values in a dataframe

1. The row will be deleted even thou it has only one NAN value not considering other data which might be useful later.
2. It’s distorts the original distribution; it can produce skewed data.





.thresh(n) - It’s checks for the condition if n actual values or more exists than the NAN values then the column should exist, but if it’s less than the NAN values then the column should be deleted.



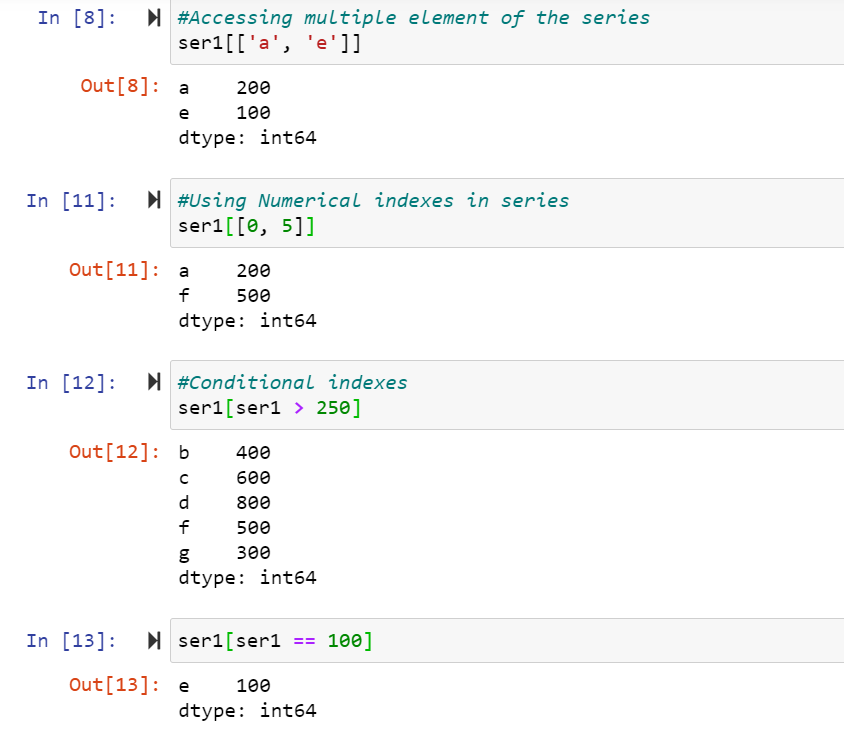
.fillna(n) function - It replaces an NAN value with the n value.

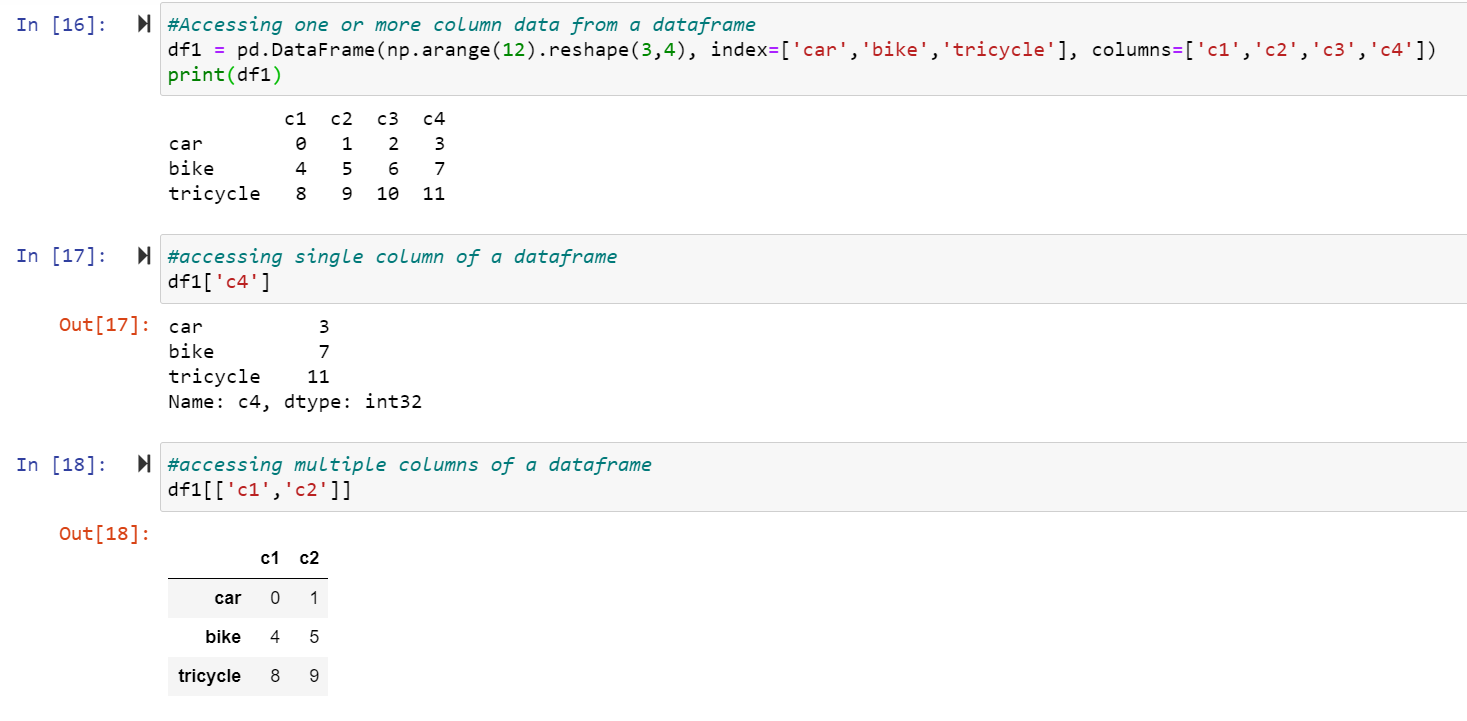
In the code above(In [28]), we filled the NAN values using the fillna method and a dictionary method. Basically, the code means that for

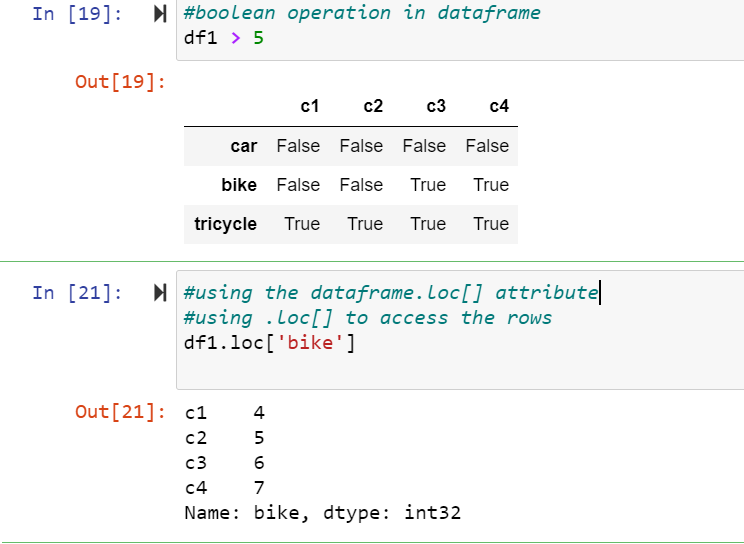
* 0 Index, the NAN value should be replaced with 0.
* Index 1, the NAN value should be replaced with 50.
* Index 2, the NAN value should be replaced with 100.
* Index 3, the NAN value should be replaced with 200.
* Selecting and Modifying data in Pandas: Pandas enables us to alter and select specific data and use certain methods and functionality to make changes to them as required.

**Code & Output:**

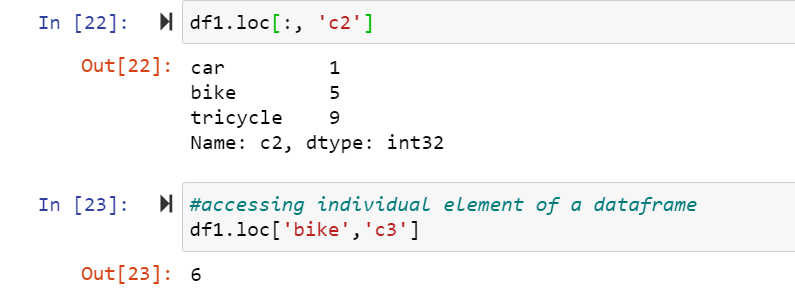






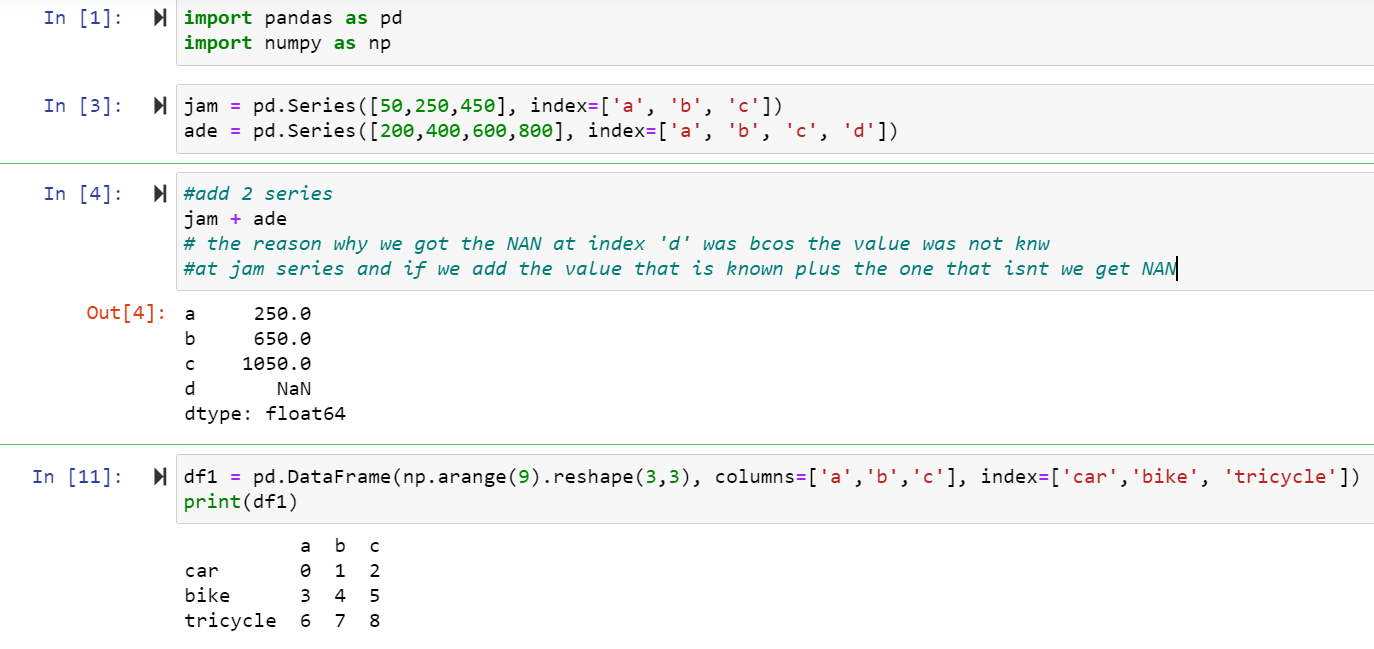


.loc[] attribute is frequently use to access the elements of the array. We can access a complete row,complete column or the individual element. It gives us a lot of flexibility.

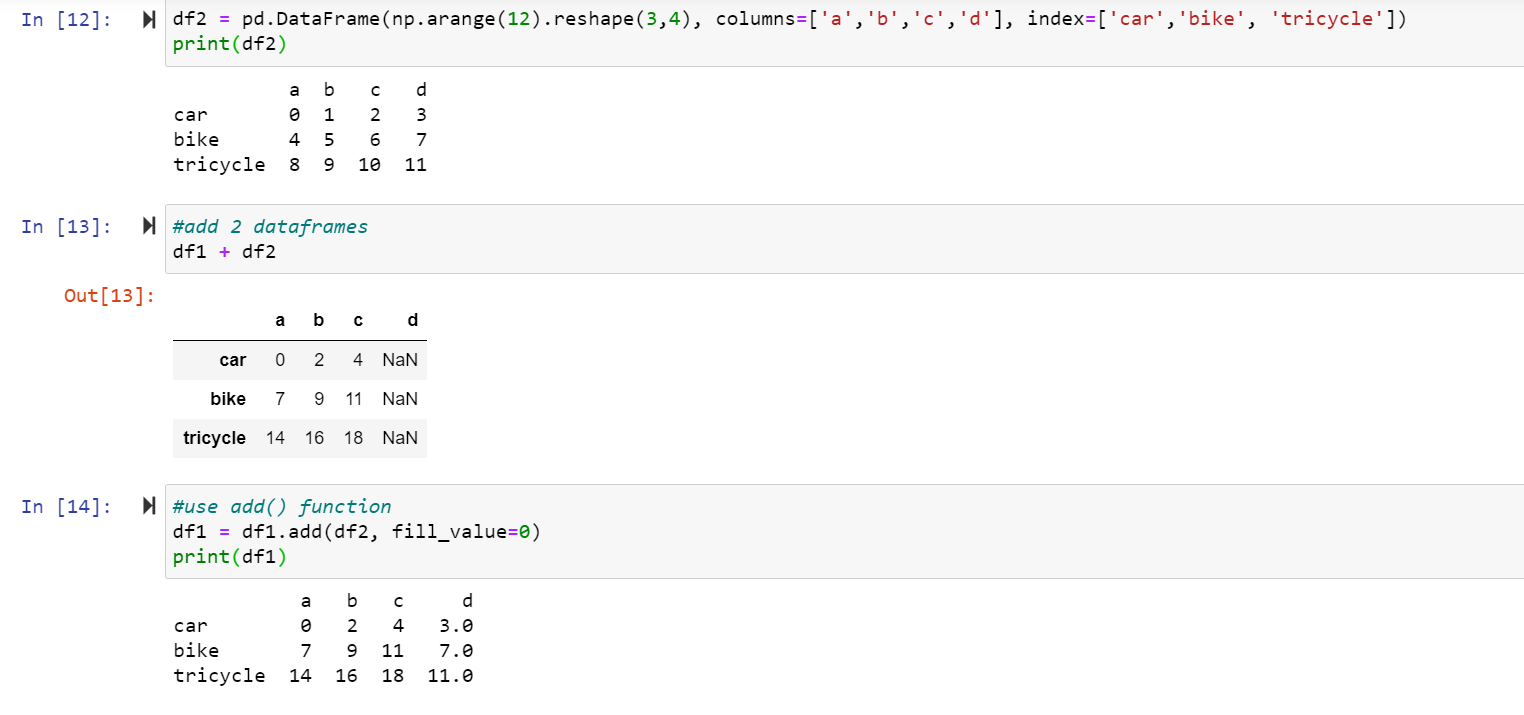


* Coordinate and Regulate data in Series and DataFrames: When we work on chunks of data, organizing data is crucial so that we can use it efficiently when required and effectively, coordination and regulation plays an important role in data arrangement.

**Code & Output:**

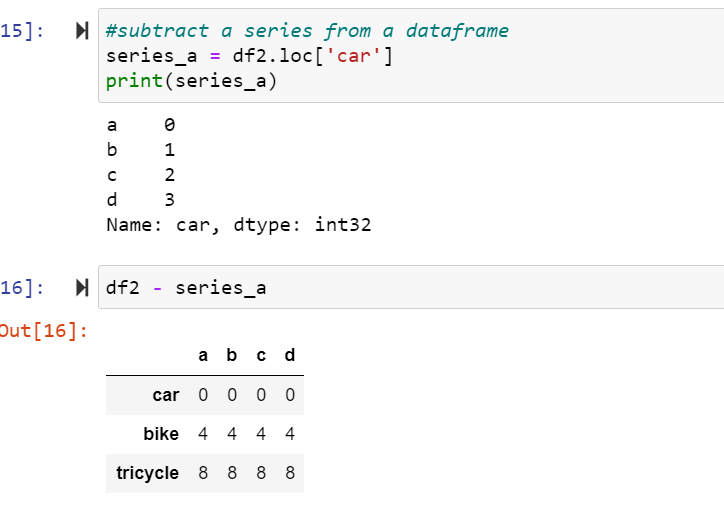


‘+’ operator when used adds the corresponding indexes of the series for both dataframe and series.



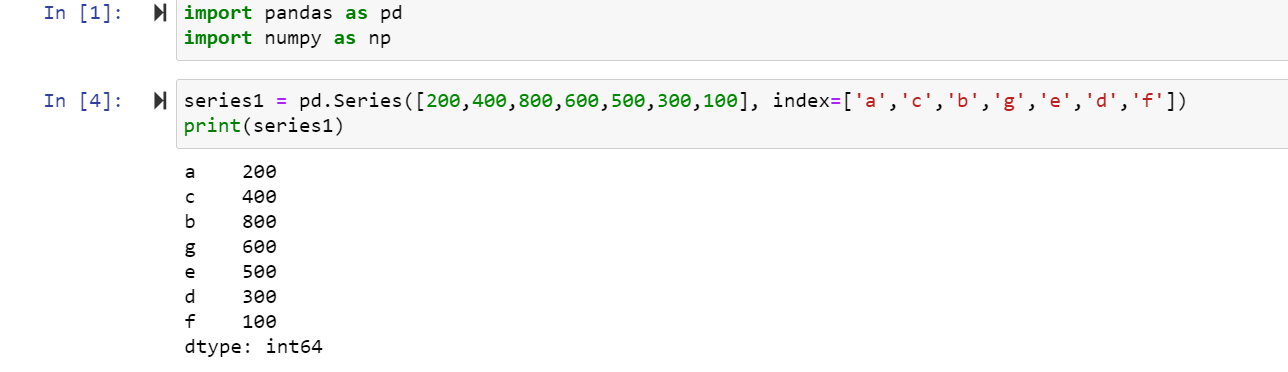
add() method is use to add dataframes and other elements(binary operator add). Equivalent to dataframe + other, but with support to replace missing data in one of the inputs with a fill value.

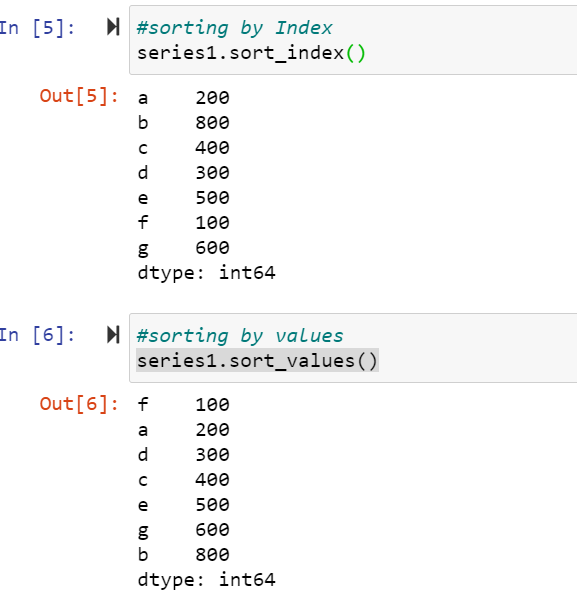
fill\_value:[None or float value, default is None] It fills missing NAN values with this values. If both positions of the dataframe are missing, the result is missing



* Ranking and Sorting in Pandas series:

**Code & Output:**

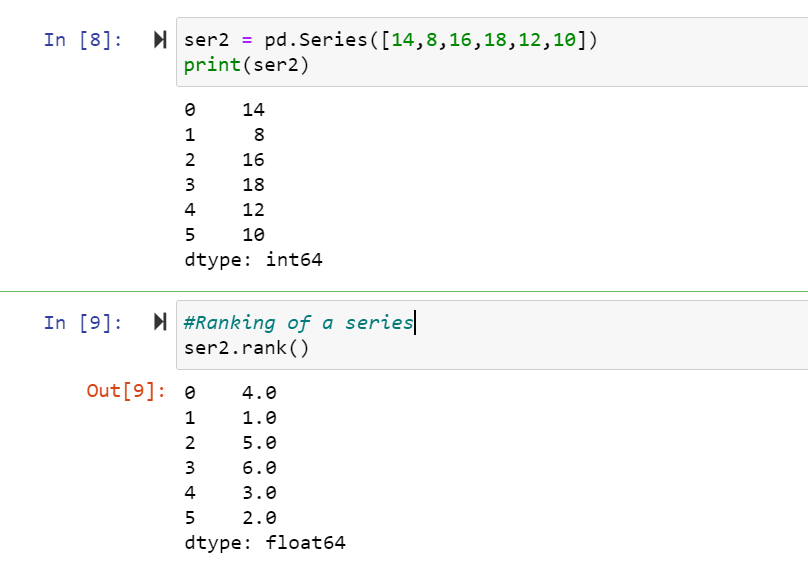




.sort\_index()- Its sort the series by the index values.

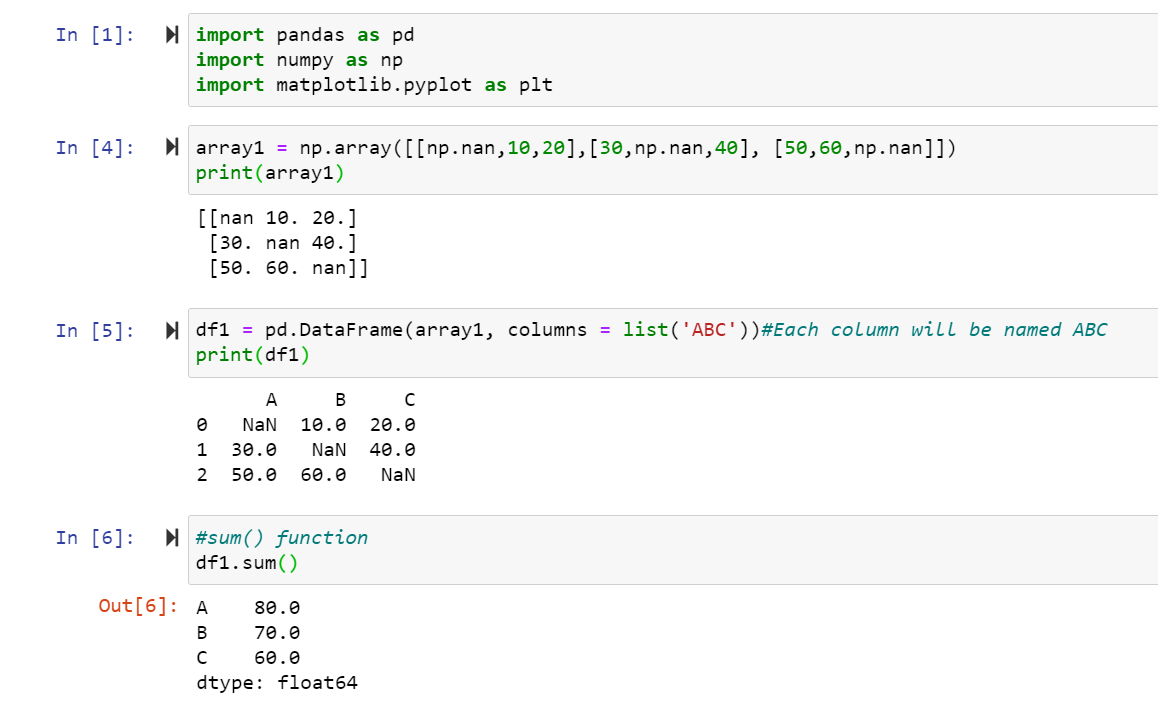
.sort\_values()- Its sort the series by the value of elements.

.rank()- First sort the series by its value(elements) and then assigns rank accordingly.



* Statistical Data Analysis and Graphs in Pandas:

**Code & Output:**

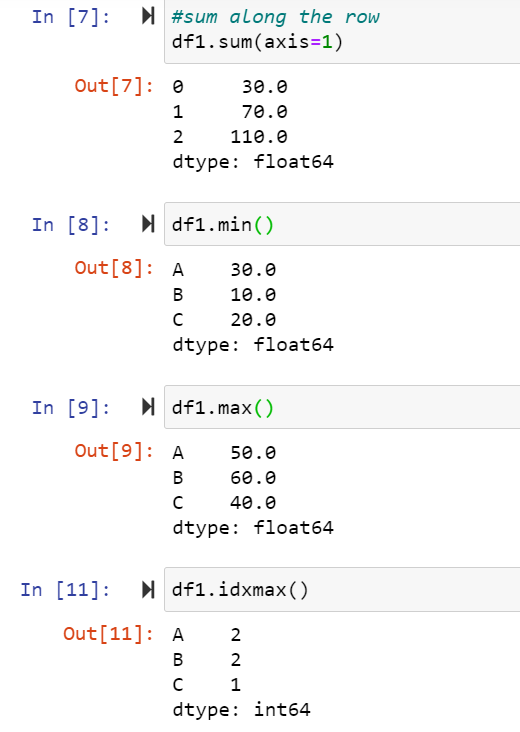
****

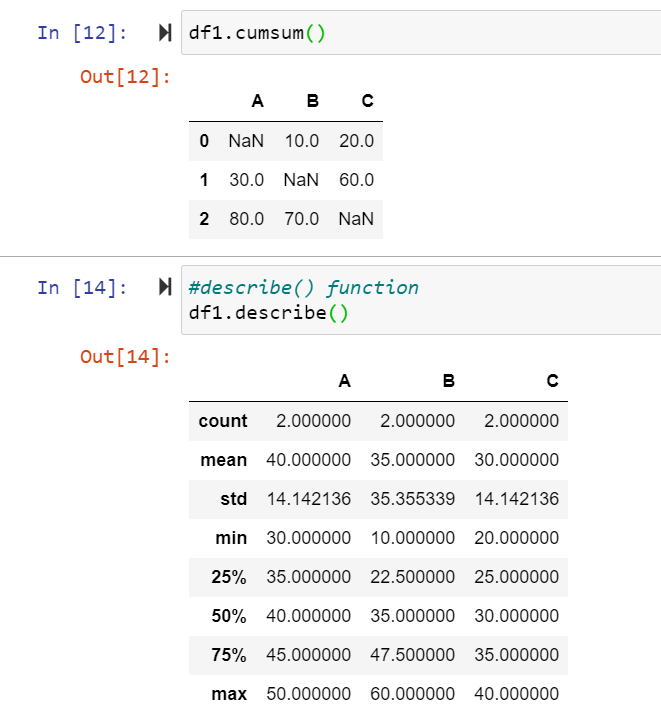
The sum() function adds the iterable element and returns the sum.

.min() function - It returns the minimum values in each column of the dataframe.

.max() function - It returns the maximum values in each column of the dataframe.

.idxmax() function - It returns the index containing highest value in the dataframe.

****



.cumsum() function - It returns the cumulative sum of the columns of the dataframe.

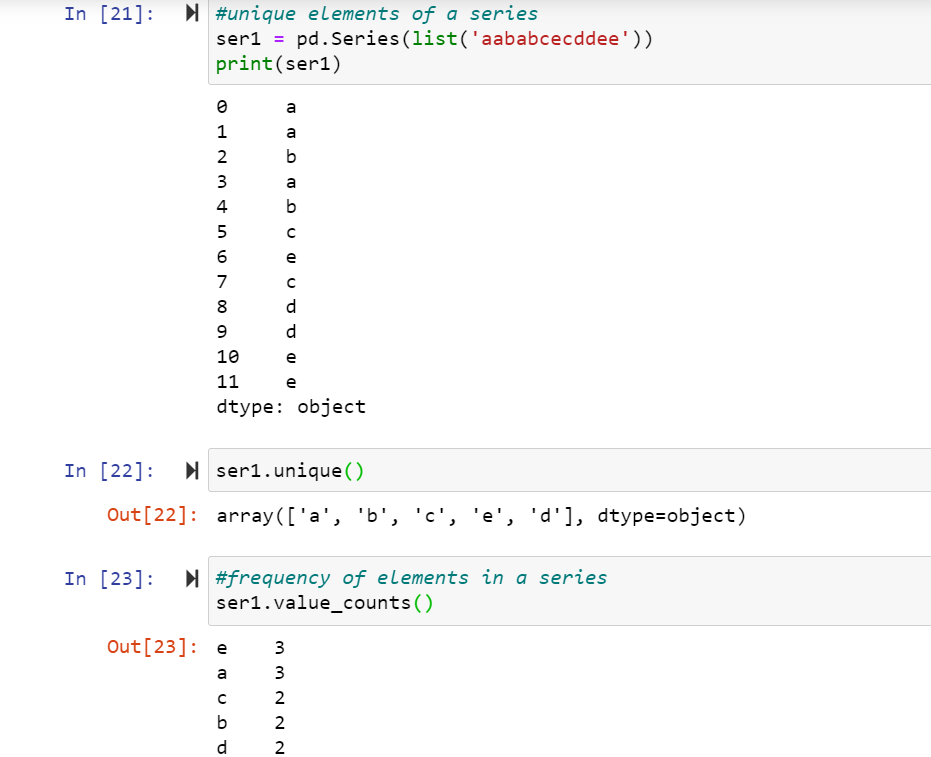
.describe() function - It returns the count,mean,standard deviation,minimum, maximum, percentile etc of each column.



.plot() function - It’s used to plot the dataframe values.

.legend() function - It’s used to identify each data element uniquely.

loc function: displays the legend object in the any specified(upper right,upper left,lower right,lower left) area in the plotted diagram.



.unique() method - It’s used to know all type of unique values in a series column.

counts() method - It print the count of each element in the series.