**NUMPY:**

Import numpy as np

* Basically imports the numpy module and we can call it with the np initials
* Performs all mathematical calculations using arrays

Using 1-D array

* Means that we will be using a single column values
* Eg. var1=[1,2,3]
* var1\_array=np.array(var1)
* this code will convert the given var1 values into 1-D array

Using 2-D array

* Means that we will be using columns and rows both in the form of a matric which is in array form
* Eg. var2=[2,5,3]
* Var2\_array=np.array([var1,var2])
* this code will convert the given var1,var2 values into 2-D array

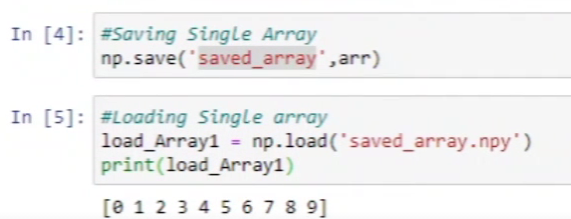
Using np.zeros, np.ones, np.empty and np.eye

* zeros will fill array matrix with zeros, rows and column depend upon the value entered with ()
* ones will fill array matrix with ones, rows and column depend upon the value entered with ()
* empty will fill array matrix with last used values or will be empty, rows and column depend upon the value entered with ()
* eye will print identical arrays

np.arange(0,50,2)

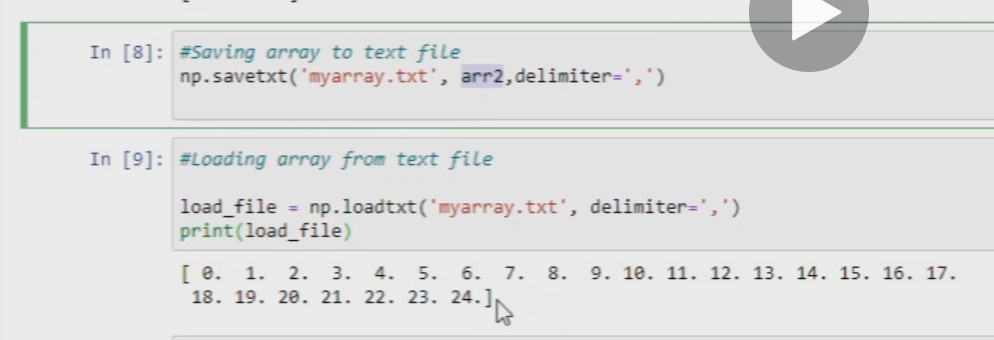
* it defines range from 0 to 50 while only printing elements that are alternated as 2 is the 3rd parameter

dtypes

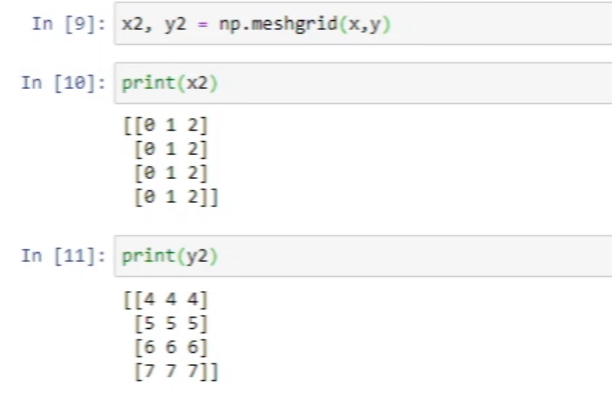
* using to show datatype of the values within a variable
* Worked on scalar operation that is nothing but the normal arithmetic operations performed using arrays like multiply, squaring, -, +, /, etc.
* Array indexing start with 0 in python
* We are using different slicing techniques to manage or data within [:,:] these. Also, slicing doesn’t assign another location memory so it will affect the previous data variable as well.
* We used numpy operation like np.sqrt, exp for square root and exponential value.
* We used np.save() to save the values of arrays in HDD so that runtime values won’t be lost.
* Savez is used to save for more that one array at a time, it saves it in zip form instead of .npy



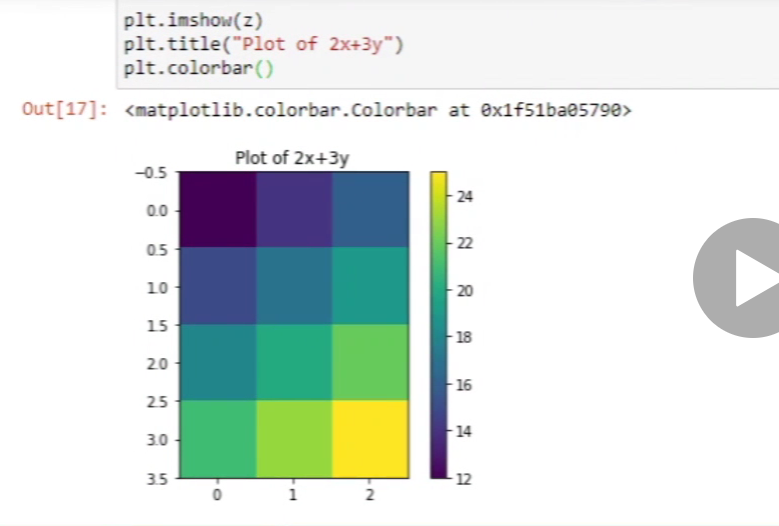
* Savetxt is used to save arr values in txt file



* Meshgrid is used for x and y(rows and column) equal to form matrix



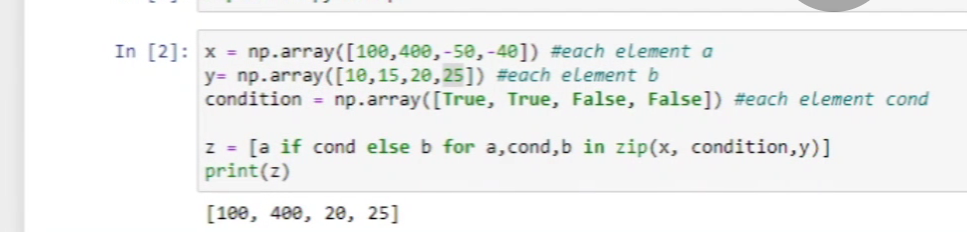
* Linear func heatmap using **matplotlib.pyplot**



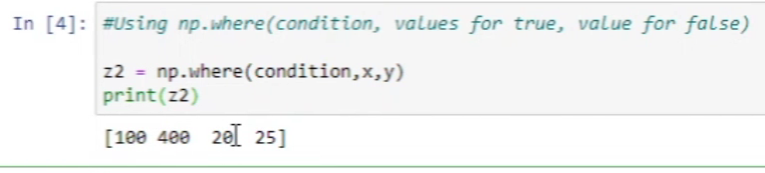
* Heatmap for cos value



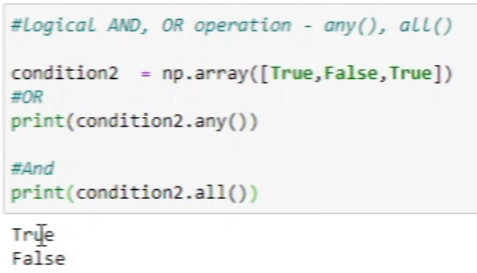
* Use plt.savefig(\_\_\_.png”) to save the map as png image
* If cond is true print value from x else y



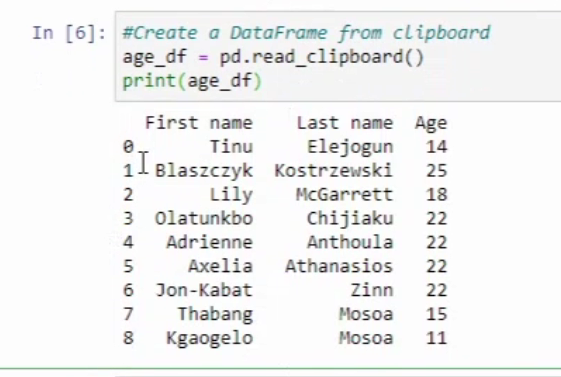
* 2nd method for same with where



* For “any “ any one value should be true to print true
* For “all “ all values should be true to print true

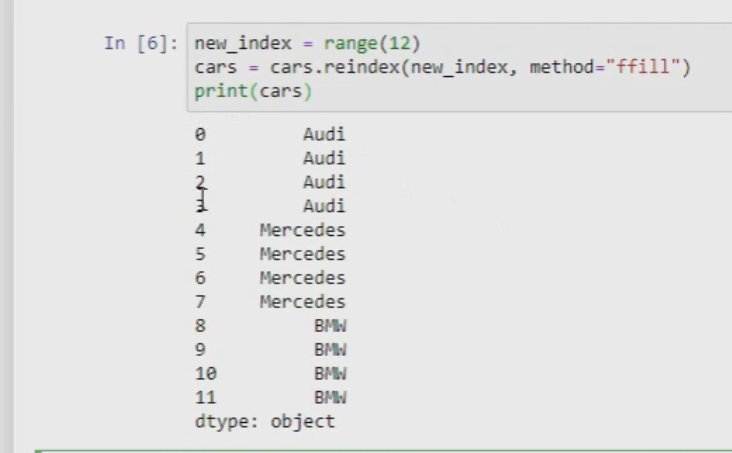


**Pandas:**

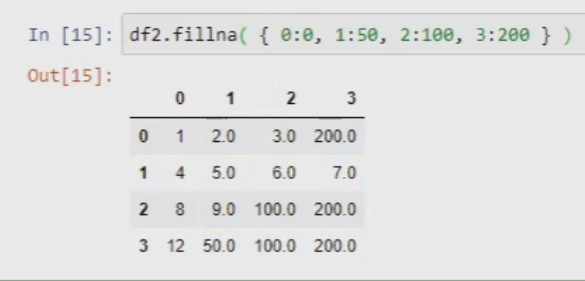
* We use import pandas to use it
* Created series and converted array into series and converted series of array into dictionaries and gave series a name and gave index a name.(var.name,var.index.name)
* To copy a table into the jupyter
* To rename a value of index



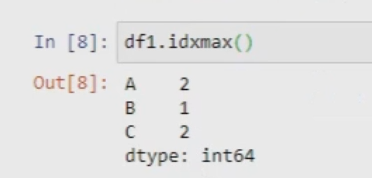
* Also, reindexing is is used to add more index elements
* arange is for array and range is for list
* ffill (forward fill) is to fill indexes till next index with same values as the last value



* We use drop function to drop rows or columns, also need to mention the axis 0 for row and 1 for column
* We can also use del keyword to delete 1 value at a time
* We can check for Nan values with isnull
* We can use fillna to fill NaN values per column bases by passing key as column name and its values for that column within a dictionary



* We use sort\_index to sort value by index and sort\_values to sort value by ascending values within data.
* Prints the index value for all columns with max values



* We can use unique to see all the unique columns