**Name: K. Praveen kumar**

**Assignment-8(20-12-23)**

**JSON:**

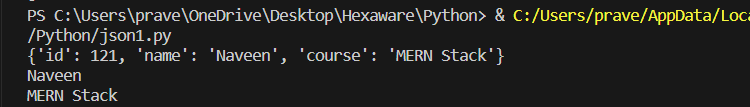
JSON stands for Java script object Notation. It means that a script file which is made of text in programming language, is used to store and transfer the data.

Python supports JSON through a built-in package called json.

**Convert json string to Dictionary:**



Output:

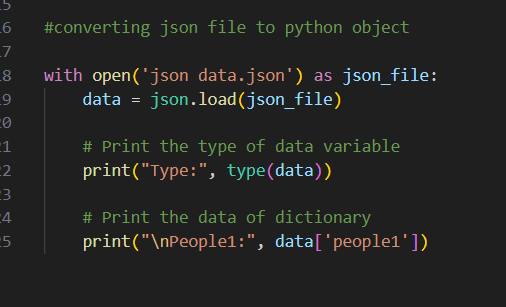


**Convert json file to Python object:**

Consider a json file which is having some data and the data can be in the format which is shown below.

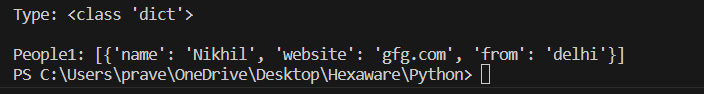


Now we load the data to convert it into a python object.



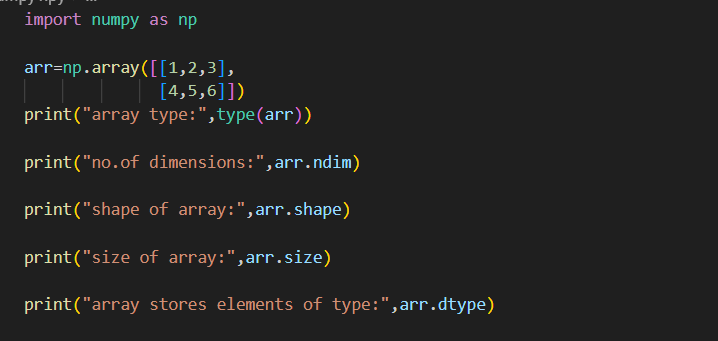
**The output will be printed in dictionary format.**

**Output:**

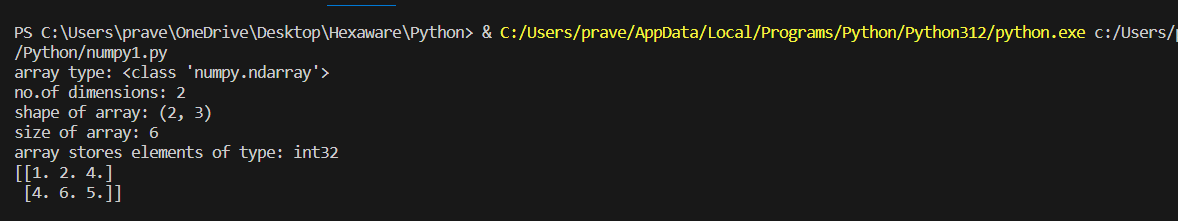


**Numpy:**

**Numpy** is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays.



**Output:**



**Numpy Array Creation:**

a=np.array([[1,2,4],[4,6,5,]],dtype='float')

print(a)

b=np.array((1,3,8))

print('\n array created using tuple:',b)

c = np.zeros((3, 4))

print ("An array initialized with all zeros:\n", c)

d = np.full((3, 3), 6, dtype = 'complex')

print ("An array initialized with all 6s."

            "Array type is complex:\n", d)

e=np.random.random((2,3))

print("random array:\n",e)

f=np.arange(0,30,5)

print('sequential array:\n',f)

g=np.linspace(0,5,10)

print('sequential array with 10 numbers between 0 and 5:\n',g)

#reshaping

arr=np.array([[1,3,4,5],[3,4,5,6],[7,8,9,0]])

newarr=arr.reshape(2,2,3)

print('originl array:\n',arr)

print('reshaped array:\n',newarr)

#flattended array

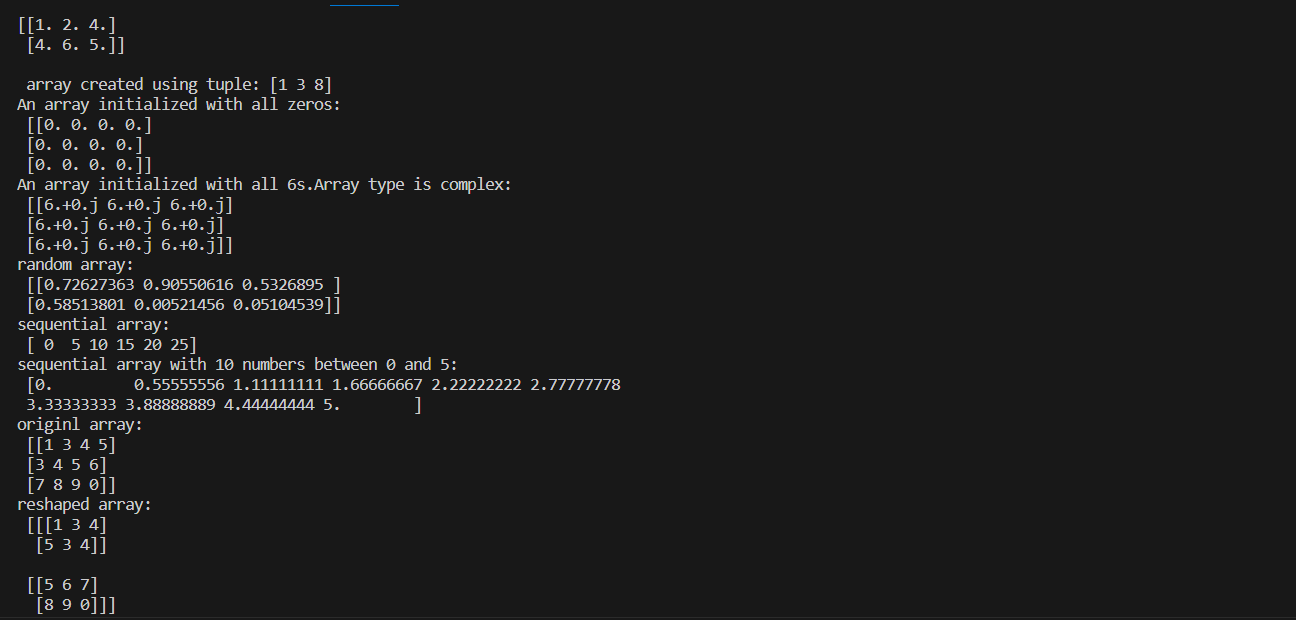
arr=np.array([[1,2,4],[4,5,6]])

newarr=arr.flatten()

print("\noriginal array:\n",arr)

print("flattened array:\n",newarr)

**Output:**

****

****

**Numpy Array Indexing:**

arr=np.array([[-1,2,0,4],

              [4, -0.5, 6, 0],

              [2.6, 0, 7, 8],

              [3, -7, 4, 2.0]])

temp=arr[:2, ::2]

print("\narray with first 2 rows and alternate columns:\n",temp)

#integer array indexing

temp = arr[[0, 1, 2, 3], [3, 2, 1, 0]]

print('\nelements at indices (0, 3), (1, 2), (2, 1),(3, 0):\n', temp)

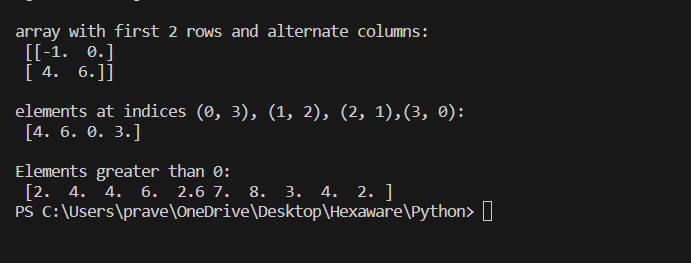
#boolean array indexing

cond=arr>0

temp=arr[cond]

print ("\nElements greater than 0:\n", temp)

**Output:**

****

**Numpy Basic operations:**

import numpy as np

a = np.array([1, 2, 5, 3])

# add 1 to every element

print ("Adding 1 to every element:", a+1)

# subtract 3 from each element

print ("Subtracting 3 from each element:", a-3)

# multiply each element by 10

print ("Multiplying each element by 10:", a\*10)

# square each element

print ("Squaring each element:", a\*\*2)

# modify existing array

a \*= 2

print ("Doubled each element of original array:", a)

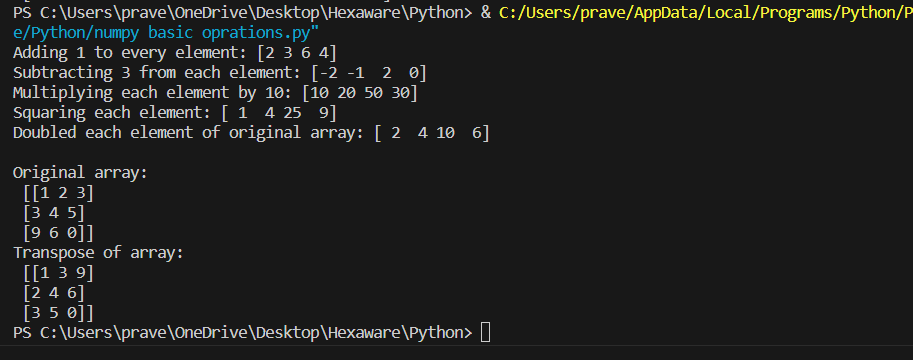
# transpose of array

a = np.array([[1, 2, 3], [3, 4, 5], [9, 6, 0]])

print ("\nOriginal array:\n", a)

print ("Transpose of array:\n", a.T)

**Output:**

****

**Numpy – Unary operators:**

import numpy as np

arr = np.array([[1, 5, 6],

                [4, 7, 2],

                [3, 1, 9]])

# maximum element of array

print ("Largest element is:", arr.max())

print ("Row-wise maximum elements:",

                    arr.max(axis = 1))

# minimum element of array

print ("Column-wise minimum elements:",

                        arr.min(axis = 0))

# sum of array elements

print ("Sum of all array elements:",

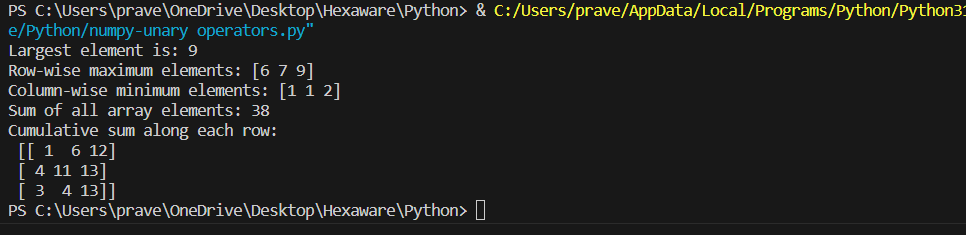
                            arr.sum())

# cumulative sum along each row

print ("Cumulative sum along each row:\n",

                        arr.cumsum(axis = 1))

**output:**



**Numpy – Binary operators:**

import numpy as np

a = np.array([[1, 2],

            [3, 4]])

b = np.array([[4, 3],

            [2, 1]])

# add arrays

print ("Array sum:\n", a + b)

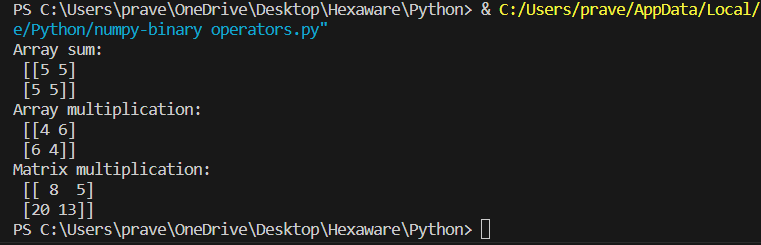
# multiply arrays (elementwise multiplication)

print ("Array multiplication:\n", a\*b)

# matrix multiplication

print ("Matrix multiplication:\n", a.dot(b))

**Output:**

****

**Numpy – ufuncs:**

import numpy as np

a=np.array([0,np.pi/2,np.pi])

print('sine values:',np.sin(a))

#exponential values

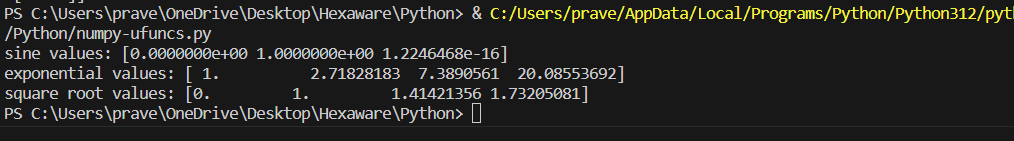
a=np.array([0,1,2,3])

print("exponential values:",np.exp(a))

#square root values

print('square root values:',np.sqrt(a))

**output:**

****

**Numpy Sorting Arrays:**

import numpy as np

a=np.array([[1, 4, 2],

            [3, 4, 6],

            [0, -1, 5]])

print('\nsorted order:\n',np.sort(a,axis=None))

# sort array row-wise

print ("Row-wise sorted array:\n",np.sort(a, axis = 1))

# specify sort algorithm

print ("Column wise sort by applying merge-sort:\n",np.sort(a,axis = 0,kind ='mergesort'))

# Example to show sorting of structured array

# set alias names for dtypes

dtypes = [('name', 'S10'), ('grad\_year', int), ('cgpa', float)]

# Values to be put in array

values = [('Hrithik', 2009, 8.5), ('Ajay', 2008, 8.7),

           ('Pankaj', 2008, 7.9), ('Aakash', 2009, 9.0)]

# Creating array

arr = np.array(values, dtype = dtypes)

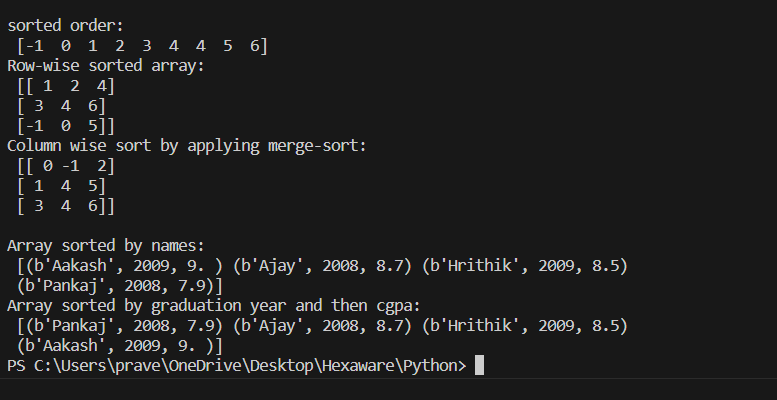
print ("\nArray sorted by names:\n",

            np.sort(arr, order = 'name'))

print ("Array sorted by graduation year and then cgpa:\n",

                np.sort(arr, order = ['grad\_year', 'cgpa']))

**Output:**

****