**DATA SCIENCE & MACHINE LEARNING-LAB CYCLE 2**

1. Create a three dimensional array specifying float data type and print it.

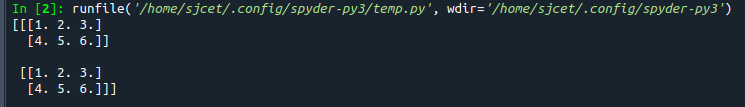
**CODE:**

import numpy as np

arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]] , dtype = float )

print (arr)

**OUTPUT**



1. Create a 2 dimensional array (2X3) with elements belonging to complex data type and print it. Also display
2. the no: of rows and columns
3. dimension of an array
4. reshape the same array to 3X2

**CODE**

import numpy as np

x = np.array([[2, 4, 6], [6, 8, 10]], dtype = complex )

print("original Array")

print(x)

print("No.of rows and columns=",x.shape)

print( "Dimension=" ,x.ndim)

newarr = x.reshape(3,2)

print(" Reshaped Array ")

print(newarr)

**OUTPUT**

****

1. Familiarize with the functions to create
2. an uninitialized array
3. array with all elements as 1,
4. all elements as 0

**CODE:**

import numpy as np

a=np.empty([2,2],dtype= int)

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

b=np.ones((2, 2),dtype=int)

print("All elements as 1: \n",b)

c=np.zeros((2,2),dtype=int)

print("All elements as 0: \n",c)

OR

import numpy as np

x=np.empty([2, 2])

print(x)

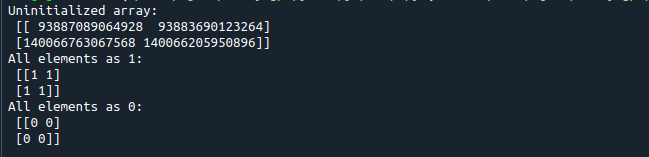
y=np.full((2, 2), 1)

print(y)

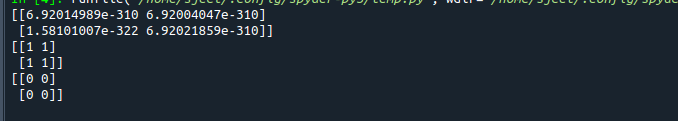
z=np.full((2, 2), 0)

print(z)

**OUTPUT**



**2nd output**



1. Create an one dimensional array using **arange** function containing 10 elements.

Display

1. First 4 elements
2. Last 6 elements
3. Elements from index 2 to 7

**CODE**

import numpy as np

a = np.arange(1, 11, 1)

print("The original array : ",a)

print("First 4 elements")

first\_element = a[:4]

print(first\_element)

print("Last 6 elements")

first\_element1 = a[5:]

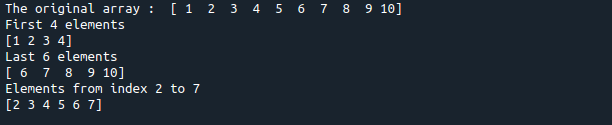
print(first\_element1)

print("Elements from index 2 to 7")

first\_element2 = a[1:7]

print(first\_element2)

**OUTPUT**

****

1. Create an 1D array with **arange** containing first 15 even numbers as elements
2. Elements from index 2 to 8 with step 2(also demonstrate the same using slice function)
3. Last 3 elements of the array using negative index
4. Alternate elements of the array
5. Display the last 3 alternate elements

**CODE**

import numpy as np

a=np.arange(1,16,1)

print("Arange:",a[2:8])

arr=np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])

print("Slicing:",arr[2:8])

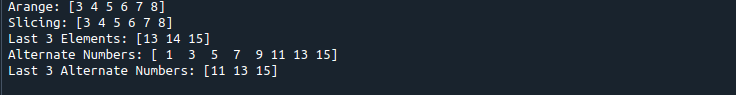
print("Last 3 Elements:",a[-3:])

a1=a=np.arange(1,16,2)

print("Alternate Numbers:",a1)

print("Last 3 Alternate Numbers:",a1[-3:])

**OUTPUT**

****

1. Create a 1 Dimensional array with 4 rows and 4 columns.
2. Display all elements excluding the first row
3. Display all elements excluding the last column
4. Display the elements of 1st and 2nd column in 2nd and 3rd row
5. Display the elements of 2nd and 3rd column
6. Display 2nd and 3rd element of 1st row

**CODE**

import numpy as np

x = np.array([[2, 4, 6,1], [6, 8, 10,1],[1, 2, 1,1], [1, 1, 1,1]])

print(x)

print("Display all elements excluding the first row")

print(x[1:])

print("Display all elements excluding the last column")

print(x[:, :3])

print("Display the elements of 2 nd and 3 rd column")

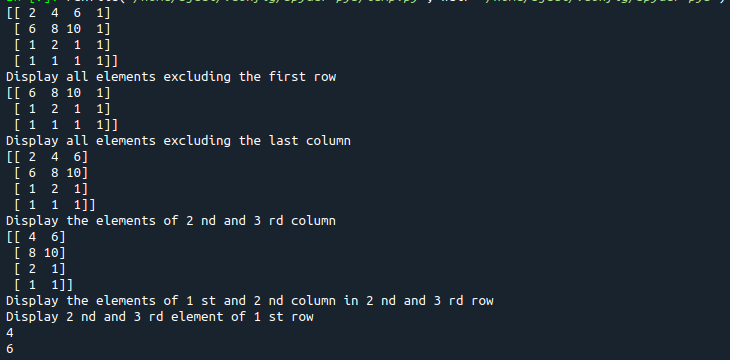
print(x[:, 1:3])

print("Display 2 nd and 3 rd element of 1 st row")

print(x[0,1])

print(x[0,2])

**OUTPUT**

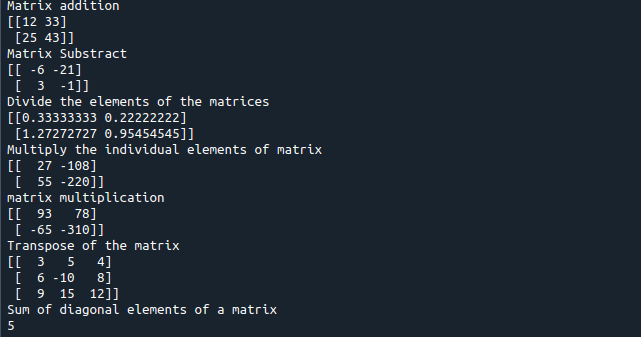


1. Create two 2D arrays using array object and
2. Add the 2 matrices and print it
3. Subtract 2 matrices
4. Multiply the individual elements of matrix
5. Divide the elements of the matrices
6. Perform matrix multiplication
7. Display transpose of the matrix
8. Sum of diagonal elements of a matrix

**CODE**

import numpy as np  
  
 M1 = np.array([[3, 6], [14, 21]])  
 M2 = np.array([[9, 27], [11, 22]])  
 M3 = M1 + M2    
 print("Matrix addition")  
 print(M3)  
 M1 = np.array([[3, 6], [14, 21]])  
 M2 = np.array([[9, 27], [11, 22]])  
 M3 = M1 - M2    
 print("Matrix Substract")  
 print(M3)  
  
 M1 = np.array([[3, 6], [14, 21]])  
 M2 = np.array([[9, 27], [11, 22]])  
 M3 = M1 / M2    
 print("Divide the elements of the matrices")  
 print(M3)  
  
  
 M1 = np.array([[3, 6], [5, -10]])  
 M2 = np.array([[9, -18], [11, 22]])  
 M3 = M1 \* M2  
 print("Multiply the individual elements of matrix")   
 print(M3)  
  
  
 M1 = np.array([[3, 6], [5, -10]])  
 M2 = np.array([[9, -18], [11, 22]])  
 M3 = M1.dot(M2)   
 print("matrix multiplication")   
 print(M3)  
  
 M1 = np.array([[3, 6, 9], [5, -10, 15], [4,8,12]])  
 M2 = M1.transpose()  
 print("Transpose of the matrix")   
 print(M2)  
  
 M1 = np.array([[3, 6, 9], [5, -10, 15], [4,8,12]])  
 print("Sum of diagonal elements of a matrix")   
 print(np.trace(M1))

**OUTPUT**

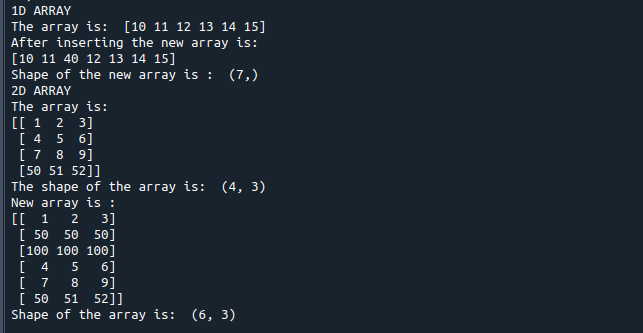
****

1. Demonstrate the use of insert() function in 1D and 2D array

**CODE**

import numpy as np  
  
arr1 = np.arange(10, 16)  
print("1D ARRAY ")  
print("The array is: ", arr1)  
  
obj = 2  
value = 40  
  
arr = np.insert(arr1, obj, value, axis=None)  
  
print("After inserting the new array is: ")  
print(arr)  
print("Shape of the new array is : ", np.shape(arr))  
  
print("2D ARRAY ")  
arr1 = np.array([(1, 2, 3), (4, 5, 6), (7, 8, 9), (50, 51, 52)])  
  
print("The array is: ")  
print(arr1)  
print("The shape of the array is: ", np.shape(arr1))  
  
a = np.insert(arr1, 1, [[50], [100], ], axis=0)  
  
print("New array is : ")  
print(a)  
print("Shape of the array is: ", np.shape(a))

**OUTPUT**



1. Demonstrate the use of diag() function in 1D and 2D array.

**CODE**

import numpy as np

a= np.array([[3, 6,7,8]])

print("1D Array \n",a)

b=np.array([[3, 6,8,7], [4, 2,1,0],[3,1,3,3],[1,1,2,2]])

print("2D Array \n",b)

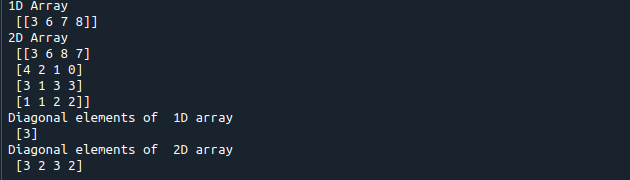
x=np.diag(a)

print("Diagonal elements of 1D array \n",x)

y=np.diag(b)

print("Diagonal elements of 2D array \n",y)

**OUTPUT**

****

1. Demonstarte the use of append() function in 1D and 2D   
    array.

**CODE**

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7])

a = np.append(arr, 88)

print("1D Array Diagonal values : \n", a)

arr = np.array( [ [1, 2, 3],[ 4, 5, 6] ])

print("2D arr : \n", arr)

a = np.append(arr, [22, 23, 24])

print("2D Array diagonal values : \n", a)

**OUTPUT**

****

**OR**

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7])

print("Original Array \n",arr)

newArr = np.append(arr, 88)

print("1D Appended Array \n",newArr)

arr1= np.array([[36,45],[50,52]])

print("Array 1 \n",arr1)

arr2= np.array([[56,60],[65,73]])

print("Array 2 \n",arr2)

res=np.append(arr1,arr2,axis=1)

print("2D Appended Array \n",res)

**OUTPUT**

****

1. Demonstarte the use of sum() function in 1D and 2D array.

**CODE**

import numpy as np

a = np.arange(5)

print("1D ARRAY \n")

print("The array is: \n", a)

print("\nSum of array : ", np.sum(a))

print("\n2D ARRAY ")

ar = np.array([(1, 2, 3), (4, 5, 6), (7, 8, 9)])

print("\nThe array is: ")

print(ar)

print("\nSum of array : ", np.sum(ar))

**OUTPUT**

****

**OR**

import numpy as np   
a = [20, 2, .2, 10, 4]    
b = ([[3, 6,8,7], [4, 2,1,0],[3,1,3,3],[1,1,2,2]])     
print("\nSum of 1D array : ", np.sum(a))   
print("\nSum of 2D array : ", np.sum(b))

**OUTPUT**

****

1. Create a 1D array and display the elements from indices 4 to 10 in descending order(use –values)

**CODE**

import numpy as np

#create numpy array

a = np.array([1,2,8,9,3,4,5,6,7])

print(a)

array\_copy = np.sort(a)[::-1]

print(array\_copy[4:10])

**OUTPUT**

