**Experiment: 3** Date: 06-11-2019

1. **Creation of python script on the modules of NumPy**

**Program:**

import numpy as np

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

print(type(a))

print(a.shape)

print(a[0], a[1], a[2])

a[0] = 5

print(a)

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

print(b.shape)

a = np.zeros((1,2)) # Create an array of all zeros

print(a)

b = np.ones((1,2)) # Create an array of all ones

print(b)

c = np.full((2,2), 7) # Create a constant array

print(c)

e = np.random.random((2,2)) # Create an array filled with random values

print(e)

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

row\_r1 = a[1, :] # Rank 1 view of the second row of a

row\_r2 = a[1:2, :] # Rank 2 view of the second row of a

print(row\_r1, row\_r1.shape) # Prints "[5 6 7 8] (4,)"

print(row\_r2, row\_r2.shape)

col\_r1 = a[:, 1]

col\_r2 = a[:, 1:2]

print(col\_r1, col\_r1.shape) # Prints "[ 2 6 10] (3,)"

print(col\_r2, col\_r2.shape)

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

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

a[np.arange(4), b] += 10

x = np.array([[1,2],[3,4]], dtype=np.float64)

y = np.array([[5,6],[7,8]], dtype=np.float64)

print(x + y)

print(np.add(x, y))

x = np.array([[1,2],[3,4]])

y = np.array([[5,6],[7,8]])

v = np.array([9,10])

w = np.array([11, 12])

print(np.dot(v, w))

x = np.array([[1,2],[3,4]])

print(np.sum(x))

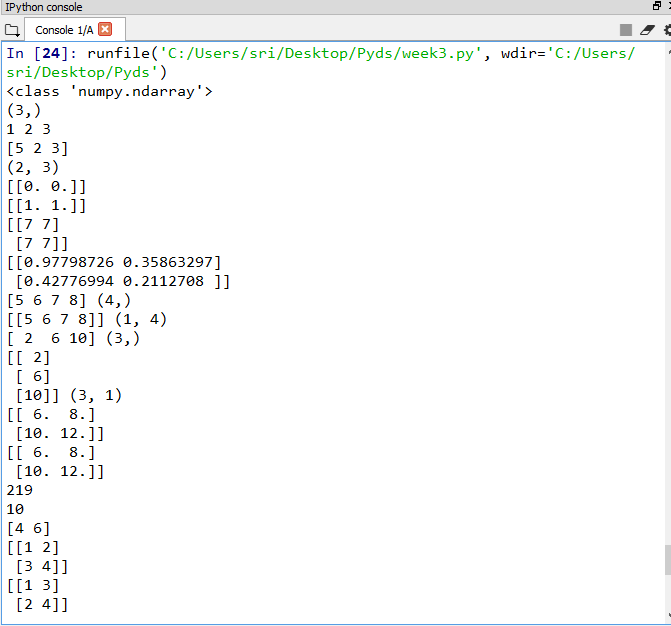
print(np.sum(x, axis=0))

x = np.array([[1,2], [3,4]])

print(x)

print(x.T)

**Result:**

****

**Experiment: 3** Date: 14-10-2019

1. **Analyse the given series of data using pandas**

**Program:**

import pandas as pd

import numpy as np

data = np.array(['a','b','c','d'])

s = pd.Series(data,index=[100,101,102,103])

print(s)

t = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])

print(t[:3])

print("DataFrame :")

import pandas as pd

df = pd.DataFrame()

print(df)

data = [1,2]

df = pd.DataFrame(data)

print(df)

data = [['Alex',10],['Bob',12],['Clarke',13]]

data = {'Name':['Tom', 'Jack', 'Steve'],'Age':[28,34,29]}

df = pd.DataFrame(data)

print(df)

data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}]

df1 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b'])

df2 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b1'])

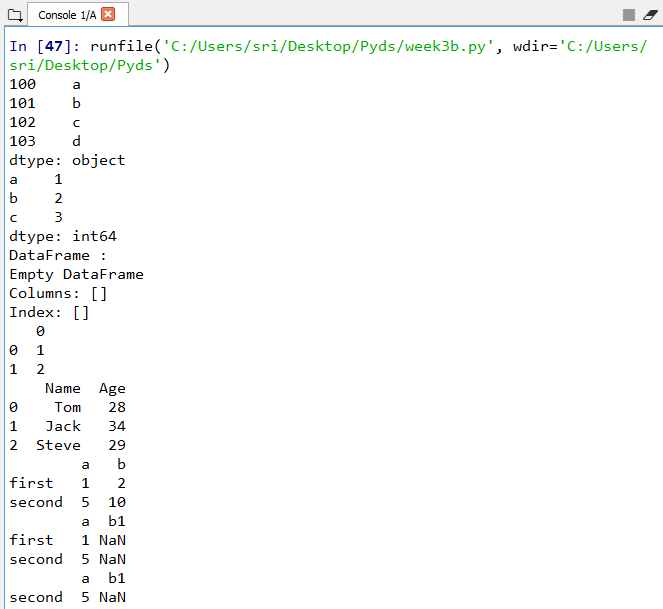
print(df1)

print(df2)

df2 = df2.drop('first')

print(df2)

**Result:**



**Experiment: 3** Date: 14-10-2019

1. **Python programs that uses the dictionaries, tuples and other data structures**

**Program :**

dict\_one = {'name':'Jack', 'age': 26}

print(dict\_one['name'])

print(dict\_one.get('age'))

dict\_one['address'] = 'Downtown'

print(dict\_one)

squares = {1:1, 2:4, 3:9, 4:16, 5:25}

print(squares.pop(4))

print(squares)

print(squares.popitem())

print(squares)

print(squares)

squares.clear()

print(squares)

del squares

marks = {}.fromkeys(['Math','English','Science'], 0)

print(marks)

for item in marks.items():

print(item)

list(sorted(marks.keys()))

squares = {x: x\*x for x in range(6)}

print(squares)

squares = {}

for x in range(6):

squares[x] = x\*x

squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}

print(1 in squares)

print(2 not in squares)

print(49 in squares)

squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}

for i in squares:

print(squares[i])

squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}

print(len(squares))

print(sorted(squares))

print("Tuples:")

a\_tuple = ()

print(a\_tuple) # Output: ()

a\_tuple = (1, 2, 3)

print(a\_tuple) # Output: (1, 2, 3)

my\_tuple = (1, "Welcome", 4.9)

print(my\_tuple) # Output: (1, "Hello", 3.4)

a\_tuple = ("Data", [8, 4, 6], (1, 2, 3))

print(a\_tuple)

a\_tuple = 5, 9.8, "science"

print(a\_tuple) # Output: 3, 4.6, "dog"

a, b, c = a\_tuple

print(a) # 3

print(b) # 4.6

print(c) # dog

a\_tuple = ("hello")

print(type(a\_tuple)) # <class 'str'>

a\_tuple = ("hello",)

print(type(a\_tuple)) # <class 'tuple'>

a\_tuple = ('T','e','c','h','n','o','l','o','g','y')

print(a\_tuple[0]) # 'p'

print(a\_tuple[5]) # 't'

n\_tuple = ("Data", [8, 4, 6], (1, 2, 3))

print(n\_tuple[0][3]) # 's'

print(n\_tuple[1][1]) # 4

print(a\_tuple[-1])

print(a\_tuple[1:4])

print(a\_tuple[:-7])

print(a\_tuple[7:])

print(a\_tuple[:])

m\_tuple = ('a','n','d','r','o','i','d')

print(m\_tuple.count('a')) # Output: 2

print(m\_tuple.index('d')) # Output: 3

print("List :")

b\_list = []

b\_list = ['n','u','t','e','r','a','l']

print(b\_list[0])

print(b\_list[2])

print(b\_list[4])

n\_list = ["Amazing", [2,0,1,5]]

print(n\_list[0][1])

print(n\_list[1][3])

my\_list = ['p','r','o','b','e']

print(my\_list[0])

print(my\_list[2])

print(my\_list[4])

n\_list = ["Happy", [2,0,1,5]]

print(n\_list[0][1])

print(n\_list[1][3])

print(b\_list[2:5])

c\_list = ['p','r','o','b','l','e','m']

del c\_list[2]

print(c\_list)

del c\_list[1:5]

print(c\_list)

k\_list = [3, 8, 1, 6, 0, 8, 4]

print(k\_list.index(8))

print(k\_list.count(8))

k\_list.sort()

print(k\_list)

k\_list.reverse()

print(k\_list)

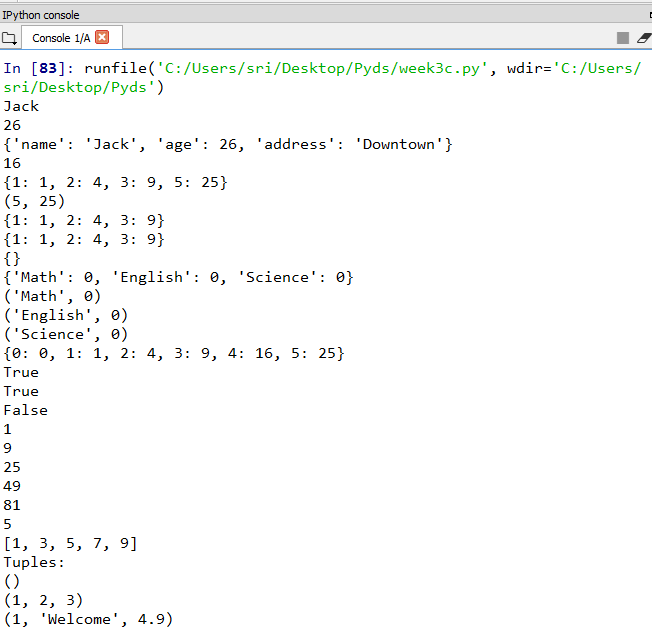
pow2 = [2 \*\* x for x in range(10)]

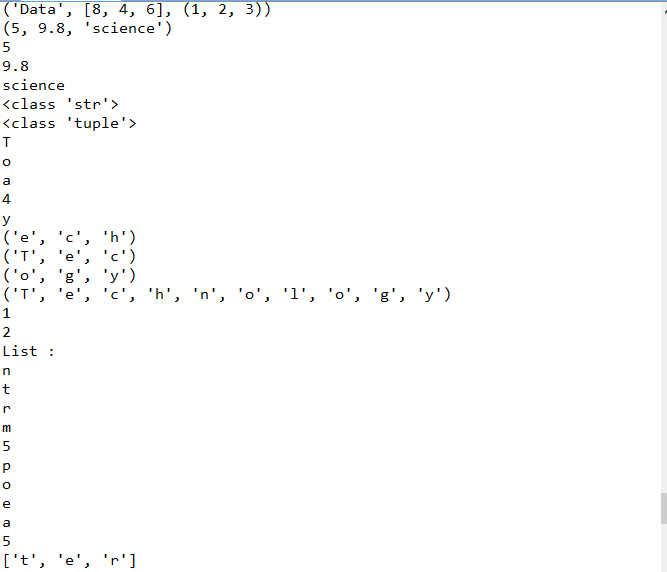
print(pow2)

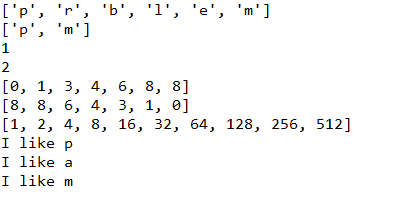
for c\_list in ['p','a','m']:

print("I like",c\_list)

**Result:**

****

****

****