

Assignment -1

Python Programming

Assignment Date	9 September 2022
Student Name	MUKESHKUMAR
Student Roll Number	211419104169
Maximum Marks	2 Marks

BASIC PYTHON

Question-1:

Split this string

```
s = "Hi there Sam!"
```

Solution:

```
s.split()
```

```
['Hi', 'there', 'Sam!']
```

Basic Python

1. Split this string

```
In [1]: s = "Hi there Sam!"
```

```
In [2]: s.split()
```

```
Out[2]: ['Hi', 'there', 'Sam!']
```

Question-2:

Use .format() to print the following string.

Output should be: The diameter of Earth is 12742 kilometers.

```
planet = "Earth"
```

```
diameter = 12742
```

Solution:

```
a="The diameter of {} is {} kilometers".format(planet,diameter)
```

```
print(a)
```

```
The diameter of Earth is 12742 kilometers
```

2. Use .format() to print the following string.

Output should be: The diameter of Earth is 12742 kilometers.

```
In [4]: planet = "Earth"
        diameter = 12742

In [5]: a="The diameter of {} is {} kilometers".format(planet,diameter)
        print(a)

The diameter of Earth is 12742 kilometers
```

Question-3:

In this nest dictionary grab the word "hello"

```
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]]]}
```

Solution:

```
print(d['k1'][3]['tricky'][3]['target'][3])
```

```
hello
```

3. In this nest dictionary grab the word "hello"

```
In [6]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]]}]

In [ ]: print(d['k1'][3]['tricky'][3]['target'][3])
```

NUMPY

import numpy as np **Question-4:**

1 Create an array of 10 zeros?

Solution:

```
np.zeros(10)
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

 2 Create an array of 10 fives?

Solution:

```
np.ones(10)*5
```

```
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

```
In [8]: np.zeros(10)
Out[8]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])

In [19]: np.ones(10)*5
Out[19]: array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

Question-5:

Create an array of all the even integers from 20 to 35

Solution:

```
np.arange(20,35,2)
```

```
array([20, 22, 24, 26, 28, 30, 32, 34])
```

5. Create an array of all the even integers from 20 to 35

```
In [10]: np.arange(20,35,2)
Out[10]: array([20, 22, 24, 26, 28, 30, 32, 34])
```

Question-6:

Create a 3x3 matrix with values ranging from 0 to 8

Solution:

```
np.array([[0,1,2],[3,4,5],[6,7,8]])
```

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

6. Create a 3x3 matrix with values ranging from 0 to 8

```
In [11]: np.array([[0,1,2],[3,4,5],[6,7,8]])
Out[11]: array([[0, 1, 2],
               [3, 4, 5],
               [6, 7, 8]])
```

Question-7:

Concatenate a and b

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

Solution:

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

```
array([1, 2, 3, 4, 5, 6])
```

7. Concatenate a and b

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

```
In [12]: a=np.array([1,2,3])  
          b=np.array([4,5,6])  
          np.concatenate((a,b))  
  
Out[12]: array([1, 2, 3, 4, 5, 6])
```

PANDAS

Question-8:

Create a dataframe with 3 rows and 2 columns **import**
pandas **as** pd

Solution:

```
data = {  
    "calories": [420, 380, 390],  
    "duration": [50, 40, 45]  
}
```

```
#load data into a DataFrame object:  
df = pd.DataFrame(data)
```

```
print(df)
```

	calories	duration
0	420	50
1	380	40
2	390	45

Pandas

8. Create a dataframe with 3 rows and 2 columns

```
In [13]: import pandas as pd

In [14]: data = [['a',10],['b',20],['c',30]]
df = pd.DataFrame(data, columns=['Alpabet','Values'])
print(df)
```

	Alpabet	Values
0	a	10
1	b	20
2	c	30

Question-9:

Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023 **Solution:**

```
pd.date_range(start='1/1/2023',end='2/10/2023')
```

```
DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
               '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',
               '2023-01-09', '2023-01-10', '2023-01-11', '2023-01-12',
               '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',
               '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',
               '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',
               '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',
               '2023-01-29', '2023-01-30', '2023-01-31', '2023-02-01',
               '2023-02-02', '2023-02-03', '2023-02-04', '2023-02-05',
               '2023-02-06', '2023-02-07', '2023-02-08', '2023-02-09',
               '2023-02-10'],
              dtype='datetime64[ns]', freq='D')
```

9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023

```
In [15]: pd.date_range(start='1/1/2023',end='2/10/2023')

Out[15]: DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',
                        '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',
                        '2023-01-09', '2023-01-10', '2023-01-11', '2023-01-12',
                        '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',
                        '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',
                        '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',
                        '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',
                        '2023-01-29', '2023-01-30', '2023-01-31', '2023-02-01',
                        '2023-02-02', '2023-02-03', '2023-02-04', '2023-02-05',
                        '2023-02-06', '2023-02-07', '2023-02-08', '2023-02-09',
                        '2023-02-10'],
                        dtype='datetime64[ns]', freq='D')
```

Question-10:

Create 2D list to DataFrame

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

Solution:

```
pd.DataFrame(lists)
```

	0	1	2
0	1		aaa 22
1	2		bbb 25
2	3		ccc 24

10. Create 2D list to DataFrame

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
In [16]: lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
In [17]: pd.DataFrame(lists)
```

```
Out[17]:
```

	0	1	2
0	1	aaa	22
1	2	bbb	25
2	3	ccc	24