

Assignment – 1

Basic Python

Assignment Date	10 September 2022
Student Name	Amuthayazhini K
Student Roll Number	2019504005
Maximum Marks	2 marks

1. Split this string

s = "Hi there Sam!"

Solution:

```
s.split()
```

Output:

```
✓ [1] 1 s = "Hi there Sam!"  
0s
```

```
✓ [2] 1 s.split()  
0s  
['Hi', 'there', 'Sam!']
```

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

```
planet = "Earth"  
diameter = 12742
```

Solution:

```
print('The diameter of {} is {} kilometers.'.format(planet,diameter))
```

Output:

```
[3] 1 planet = "Earth"  
2 diameter = 12742
```

```
[4] 1 print('The diameter of {} is {} kilometers.'.format(planet,diameter))
```

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

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

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

Solution:

```
d['k1'][-1]['tricky'][-1]['target'][-1]
```

Output:

```
[5] 1 d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

```
[6] 1 d['k1'][-1]['tricky'][-1]['target'][-1]
```

'hello'

4.Numpy

4.1 Create an array of 10 zeros?

Solution:

```
import numpy as np
np.zeros(10)
```

Output:

```
[10] 1 np.zeros(10)
```

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

4.2 Create an array of 10 fives?

Solution:

```
import numpy as np
np.ones(10)*5
```

Output:

```
[11] 1 np.ones(10)*5
```

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

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

Solution:

```
evenarray = np.arange(20,36,2)
evenarray
```

Output:

```
[13]  1  evenarray = np.arange(20,36,2)
      2  evenarray

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

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

Solution:

```
mat = np.arange(0,9).reshape(3,3)
```

Output:

```
[14]  1  mat = np.arange(0,9).reshape(3,3)
      2  mat

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

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])
con1 = np.concatenate((a,b),axis=0)
con1
```

Output:

```
[15]  1  a = np.array([1,2,3])
      2  b = np.array([4,5,6])
      3  con1 = np.concatenate((a,b),axis=0)
      4  con1

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

Pandas

8. Create a dataframe with 3 rows and 2 columns:

Solution:

```
import pandas as pd
d = [['a','b'],['c','d'],['e','f']]
DF = pd.DataFrame(d)
DF
```

Output:

```
1 import pandas as pd
2

[18] 1 d = [['a','b'],['c','d'],['e','f']]
    2 DF = pd.DataFrame(d)
    3 DF
    4
```

	0	1
0	a	b
1	c	d
2	e	f

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

Solution:

```
date_series = pd.date_range(start='01-01-2023',end='02-10-2023')
date_series
```

Output:

```
[19] 1 date_series = pd.date_range(start='01-01-2023',end='02-10-2023')
    2 date_series

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')
```

10. Create 2D list to DataFrame

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

Solution:

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]  
dataframe2D = pd.DataFrame(lists)  
dataframe2D
```

Output:

```
[20] 1 lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
[21] 1 dataframe2D = pd.DataFrame(lists)  
    2 dataframe2D
```

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