

## Assignment – 1

### Basic Python

Assignment Date	10 September 2022
Student Name	Prabakaran V
Student Roll Number	111619106102
Maximum Marks	2 Marks

#### 1. Split this string

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

Code:

```
print(s.split())
```

Output:

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

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

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

Code:

```
print("The diameter of {0} is {1} kilometers.".format(planet,diameter))
```

Output:

```
In [6]: planet = "Earth"  
       diameter = 12742
```

```
In [11]: print("The diameter of {0} is {1} 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']}]}]}
```

Code:

```
d['k1'][3]['tricky'][3]['target'][3]
```

Output:

```
In [12]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

```
In [49]: d['k1'][3]['tricky'][3]['target'][3]
```

```
Out[49]: 'hello'
```

## Numpy

### 4.1 Create an array of 10 zeros?

### 4.2 Create an array of 10 fives?

Code:

```
import numpy as np  
  
arr=np.zeros(10)  
print(arr)  
  
arr=np.ones(10)*5  
print(arr)
```

Output:

```
In [15]: import numpy as np
```

### 4.1 Create an array of 10 zeros?

### 4.2 Create an array of 10 fives?

```
In [21]: arr=np.zeros(10)  
print(arr)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
In [22]: arr=np.ones(10)*5  
print(arr)
```

```
[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]
```

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

Code:

```
arr=np.arange(20,35,2)
print(arr)
```

Output:

```
In [24]: arr=np.arange(20,35,2)
print(arr)

[20 22 24 26 28 30 32 34]
```

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

Code:

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

Output:

```
In [25]: arr=np.arange(0,9).reshape(3,3)
print(arr)

[[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])**

Code:

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

Output:

```
In [27]: a=np.array([1,2,3])
b=np.array([4,5,6])
np.concatenate((a,b),axis=0)

Out[27]: array([1, 2, 3, 4, 5, 6])
```

## Pandas

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

Code:

```
import pandas as pd
arr=[1,2,3]
dataframe=pd.DataFrame(arr,columns=["data"])
print(dataframe)
```

Output:

```
In [28]: import pandas as pd
```

```
In [36]: arr=[1,2,3]
dataframe=pd.DataFrame(arr,columns=["data"])
print(dataframe)
```

```
   data
0     1
1     2
2     3
```

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

Code:

```
date=pd.date_range(start='1-01-2023',end='2-10-2023')
print(date)
```

Output:

```
In [33]: date=pd.date_range(start='1-01-2023',end='2-10-2023')
print(date)
```

```
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]]
```

Code:

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

```
dataframe=pd.DataFrame(lists,columns=["number","letter","digit"])
print(dataframe)
```

### Output:

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

```
In [38]: lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
In [40]: dataframe=pd.DataFrame(lists,columns=["number","letter","digit"])
print(dataframe)
```

	number	letter	digit
0	1	aaa	22
1	2	bbb	25
2	3	ccc	24