

# Assignment -1

## Basic python

|                     |                     |
|---------------------|---------------------|
| Assignment Date     | 08 September 2022   |
| Student Name        | Sivagama Sundari. V |
| Student Roll Number | 211419104255        |
| Maximum Marks       | 2 Marks             |

## Basic Python ¶

### 1. Split this string

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

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

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

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

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

```
In [3]: planet = "Earth"  
diameter = 12742
```

```
In [4]: 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"

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

```
In [6]: print (d['k1'][3]['tricky'][3]['target'][3])
```

hello

# Numpy

```
In [7]: import numpy as np
```

## 4.1 Create an array of 10 zeros?

## 4.2 Create an array of 10 fives?

```
In [8]: array = np.zeros(10)
print(array)
```

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

```
In [9]: array=np.ones(10)*5
print(array)
```

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

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

```
In [10]: array=np.arange(20,35,2)
print(array)
```

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

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

```
In [11]: array = np.arange(0,9).reshape(3,3)
print(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])**

```
In [12]: a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
conc_arr = np.concatenate((a, b), axis = None)
print(conc_arr)
```

```
[1 2 3 4 5 6]
```

## Pandas

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

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

```
In [14]: s=[[1,2],[3,4],[5,6]]
df = pd.DataFrame(s,columns=['First column', 'Second column'])
print(df)
```

|   | First column | Second column |
|---|--------------|---------------|
| 0 | 1            | 2             |
| 1 | 3            | 4             |
| 2 | 5            | 6             |

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

```
In [15]: first2023 = pd.date_range(start='2023-01-01', end='2023-10-02', freq='MS')
list2023 = []
for i in first2023:
    list2023.append(i.strftime('%Y-%m-%d'))
list2023
```

```
Out[15]: ['2023-01-01',
'2023-02-01',
'2023-03-01',
'2023-04-01',
'2023-05-01',
'2023-06-01',
'2023-07-01',
'2023-08-01',
'2023-09-01',
'2023-10-01']
```

### 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]: df = pd.DataFrame(lists,columns=['First column', 'Second column', 'Third column'])
print(df)
```

|   | First column | Second column | Third column |
|---|--------------|---------------|--------------|
| 0 | 1            | aaa           | 22           |
| 1 | 2            | bbb           | 25           |
| 2 | 3            | ccc           | 24           |