

ASSIGNMENT -1

| | |
|-----------------|------------------|
| Name | RAGUL.U.M |
| Register Number | 210519205039 |
| Team size | 4 |
| Team ID | PNT2022TMID25119 |

Basic Python

1. Split this string

```
In[:]  
s = "Hi there Sam!"  
  
In[:]  
s=s.split(" ")  
s  
  
Out[:]  
['Hi', 'there', 'Sam!']
```

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

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

```
In[:]  
planet = "Earth"  
diameter = 12742  
  
In[:]  
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[:]  
d =  
{ 'k1': [1,2,3,{'tricky': ['oh', 'man', 'inception',{'target': [1,2,3,'hello']}]}]  
}  
  
In[:]  
d['k1'][3]['tricky'][3]  
  
Out[:]  
{'target': [1, 2, 3, 'hello']}
```

Numpy

```
In [2]:  
import numpy as np
```

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

```
In []:  
np.zeros(10,dtype=int)  
  
Out[ ]:  
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])  
  
In []:  
np.ones(10,dtype=int)*5  
  
Out[ ]:  
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 []:  
evenIntegers = np.arange(20,36,2)  
evenIntegers  
  
Out[ ]:  
array([20, 22, 24, 26, 28, 30, 32, 34])
```

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

```
In []:  
values=np.random.randint(9,size=(3,3))  
values  
  
Out[ ]:  
array([[4, 8, 2],  
       [1, 4, 0],  
       [8, 7, 0]])
```

7. Concatenate a and b

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

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

Pandas

8. Create a dataframe with 3 rows and 2 columns

In [1]:

```
import pandas as pd
```

In []:

```
s1 = pd.Series(np.random.rand(2))
s2 = pd.Series(np.random.rand(2))
s3 = pd.Series(np.random.rand(2))
df = pd.DataFrame([s1, s2, s3])
df
```

Out[]:

| | 0 | 1 |
|---|----------|----------|
| 0 | 0.238527 | 0.343665 |
| 1 | 0.306049 | 0.645804 |
| 2 | 0.908118 | 0.925606 |

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

In [3]:

```
date=pd.date_range(start='1st Jan,2023',end='10th Feb,2023')
dates = pd.Series(date)
dates
```

Out[3]:

| | |
|----|------------|
| 0 | 2023-01-01 |
| 1 | 2023-01-02 |
| 2 | 2023-01-03 |
| 3 | 2023-01-04 |
| 4 | 2023-01-05 |
| 5 | 2023-01-06 |
| 6 | 2023-01-07 |
| 7 | 2023-01-08 |
| 8 | 2023-01-09 |
| 9 | 2023-01-10 |
| 10 | 2023-01-11 |
| 11 | 2023-01-12 |
| 12 | 2023-01-13 |
| 13 | 2023-01-14 |
| 14 | 2023-01-15 |
| 15 | 2023-01-16 |
| 16 | 2023-01-17 |
| 17 | 2023-01-18 |

```
18    2023-01-19
19    2023-01-20
20    2023-01-21
21    2023-01-22
22    2023-01-23
23    2023-01-24
24    2023-01-25
25    2023-01-26
26    2023-01-27
27    2023-01-28
28    2023-01-29
29    2023-01-30
30    2023-01-31
31    2023-02-01
32    2023-02-02
33    2023-02-03
34    2023-02-04
35    2023-02-05
36    2023-02-06
37    2023-02-07
38    2023-02-08
39    2023-02-09
40    2023-02-10
dtype: datetime64[ns]
```

10. Create 2D list to DataFrame

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

```
In [4]:
```

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

```
In [ ]:
```

```
df = pd.DataFrame(lists, columns=['name', 'number'])
print(df)
```

```
name  number
0   aaa      22
1   bbb      25
2   ccc      24
```