

Assignment -1

Team ID	PNT2022TMID03497
Maximum Marks	2 Marks

Question-1:

Split this string

Solution:

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

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

```
#-----#
```

```
#-----#
```

```
In [1]: s = "Hi there Sam!"
print(s.split())

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

Question-2:

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

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

Solution:

```
planet = "Earth"
```

```
diameter = 12742
```

```
txt="The diameter of {plt} is {dr} kilometres.".format(plt=planet,dr=diameter)
```

```
print(txt)
```

```
#-----#  
#-----#
```

```
In [8]: planet = "Earth"  
        diameter = 12742  
  
In [9]: txt="The diameter of {plt} is {dr} kilometres.".format(plt=planet,dr=diameter)  
        print(txt)  
The diameter of Earth is 12742 kilometres.
```

Question 3:

In this nest dictionary grab the word "hello"

Solution:

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

Question 4.1:

Create an array of 10 zeros?

Solution:

	import numpy as np
	array=np.zeros(10)
	print(array)

Question4.2:

Create an array of 10 fives?

Solution:

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

Numpy

```
In [4]: import numpy as np
```

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

```
In [5]: array=np.zeros(10)
print(array)
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
In [6]: array=np.ones(10)*5
print(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:

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

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

```
In [7]: array=np.arange(20,35,2)
print(array)
[20 22 24 26 28 30 32 34]
```

Question 6:

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

Solution:

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

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

```
In [20]: arr=np.arange(0,9).reshape(3,3)
print(arr)
[[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])
```

7. Concatenate a and b

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

```
In [22]: a=np.array([1,2,3])
         b=np.array([4,5,6])
         np.concatenate([a,b])

Out[22]: array([1, 2, 3, 4, 5, 6])
```

Question 8:

Create a dataframe with 3 rows and 2 columns

Solution:

```
import pandas as pd
data=pd.DataFrame(index=np.arange(3), columns=np.arange(2))
print(data)
```

8. Create a dataframe with 3 rows and 2 columns

```
In [9]: import pandas as pd

In [10]: data=pd.DataFrame(index=np.arange(3), columns=np.arange(2))
         print(data)

   0  1
0 NaN NaN
1 NaN NaN
2 NaN NaN
```

Question 9:

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

Solution:

```
data=pd. date_range (start="1/1/2023", end="10/2/2023") print (data)
```

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

```
In [11]: data=pd.date_range(start="1/1/2023",end="10/2/2023")
print(data)

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',
              ...,
              '2023-09-23', '2023-09-24', '2023-09-25', '2023-09-26',
              '2023-09-27', '2023-09-28', '2023-09-29', '2023-09-30',
              '2023-10-01', '2023-10-02'],
              dtype='datetime64[ns]', length=275, freq='D')
```

Question 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]]
data=pd.DataFrame(lists,columns=["s.no","pattern","number"])print(data)
```

10. Create 2D list to DataFrame

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

```
In [12]: lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

```
In [13]: data=pd.DataFrame(lists,columns=["s.no","pattern","number"])
print(data)
```

	s.no	pattern	number
0	1	aaa	22
1	2	bbb	25
2	3	ccc	24