

Assignment -1
Python Programming

| | |
|---------------------|-------------------|
| Assignment Date | 19 September 2022 |
| Student Name | Tharanyaa |
| Student Roll Number | 2019115113 |
| Maximum Marks | 2 Marks |

Question-1:

Split this string

s = "Hi there Sam!"

Solution:

```
s1=s.split()
print(s1)
```

Output:

1. Split this string

```
s = "Hi there Sam!"

s1=s.split()
print(s1)
```

Python

Python

... ['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

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

Output:

```
[1] planet = "Earth"
    diameter = 12742
    print("The diameter of {} is {} kilometers.".format(planet,diameter))

The diameter of Earth is 12742 kilometers.
```

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']}]}]}
print(d["k1"][3]["tricky"][3]["target"][3])
```

Output:

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

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

Python

```
print(d["k1"][3]["tricky"][3]["target"][3])
```

Python

.. hello

Question-4:

4.1 Create an array of 10 zeros?

Solution:

```
import numpy as np
array=np.zeros(10)
print("An array of 10 zeros:")
print(array)
```

Output:

```
import numpy as np
array=np.zeros(10)
print("An array of 10 zeros:")
print(array)
```

Python

.. An array of 10 zeros:
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

4.2 Create an array of 10 fives?

Solution:

```
array=np.ones(10)*5
print("An array of 10 fives:")
print(array)
```

Output:

```
array=np.ones(10)*5
print("An array of 10 fives:")
print(array)
```

Python

```
An array of 10 fives:
[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:

```
import numpy as np
array=np.arange(20,36,2)
print("Array of all the even integers from 30 to 70")
print(array)
```

Output:

```
import numpy as np
array=np.arange(20,36,2)
print("Array of all the even integers from 30 to 70")
print(array)
```

Python

```
... Array of all the even integers from 30 to 70
[20 22 24 26 28 30 32 34]
```

Question-6:

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

Solution:

```
np.arange(9).reshape(3,3)
```

Output:

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

[+ Code](#) [+ Markdown](#)

```
np.arange(9).reshape(3,3)
```

Python

```
array([[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:

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

```
gfg = np.concatenate((a, b), axis = 0)
print (gfg)
```

Output:

7. Concatinate a and b

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

```
import numpy as np

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

gfg = np.concatenate((a, b), axis = 0)

print (gfg)
```

[1 2 3 4 5 6]

Question-8:

Create a dataframe with 3 rows and 2 columns

Solution:

```
import pandas as pd
data = [['tom', 10], ['nick', 15], ['juli', 14]]
# Create the pandas DataFrame df = pd.DataFrame(data, columns=['Name', 'Age'])
# print dataframe.print(df)
```

Output:

Pandas

8. Create a dataframe with 3 rows and 2 columns

```
import pandas as pd

data = [['tom', 10], ['nick', 15], ['juli', 14]]

# Create the pandas DataFrame
df = pd.DataFrame(data, columns=['Name', 'Age'])

# print dataframe.
print(df)
```

| | Name | Age |
|---|------|-----|
| 0 | tom | 10 |
| 1 | nick | 15 |
| 2 | juli | 14 |

Question-9:

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

Solution:

```
dates = pd.date_range(start="1-1-2023", end="10-2-2023")
for i in dates:
    print(i)
```

Output:

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

```
dates = pd.date_range(start="1-1-2023", end="10-2-2023")
for i in dates:
    print(i)
```

Python

... Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
2023-01-01 00:00:00
2023-01-02 00:00:00
2023-01-03 00:00:00
2023-01-04 00:00:00
2023-01-05 00:00:00
2023-01-06 00:00:00
2023-01-07 00:00:00
2023-01-08 00:00:00
2023-01-09 00:00:00
2023-01-10 00:00:00
2023-01-11 00:00:00
2023-01-12 00:00:00
2023-01-13 00:00:00
2023-01-14 00:00:00
2023-01-15 00:00:00
2023-01-16 00:00:00
2023-01-17 00:00:00
2023-01-18 00:00:00
2023-01-19 00:00:00
2023-01-20 00:00:00
2023-01-21 00:00:00
2023-01-22 00:00:00
2023-01-23 00:00:00
2023-01-24 00:00:00
2023-01-25 00:00:00
...
2023-02-09 00:00:00
2023-02-10 00:00:00
2023-02-11 00:00:00
2023-02-12 00:00:00
2023-02-13 00:00:00
2023-02-14 00:00:00
2023-02-15 00:00:00
2023-02-16 00:00:00
2023-02-17 00:00:00
2023-02-18 00:00:00
2023-02-19 00:00:00
2023-02-20 00:00:00
2023-02-21 00:00:00
2023-02-22 00:00:00
2023-02-23 00:00:00
2023-02-24 00:00:00
2023-02-25 00:00:00
2023-02-26 00:00:00
2023-02-27 00:00:00
2023-02-28 00:00:00
2023-03-01 00:00:00
2023-03-02 00:00:00
2023-03-03 00:00:00
2023-03-04 00:00:00
2023-03-05 00:00:00
2023-03-06 00:00:00
2023-03-07 00:00:00
2023-03-08 00:00:00
2023-03-09 00:00:00
2023-03-10 00:00:00
2023-03-11 00:00:00
2023-03-12 00:00:00
2023-03-13 00:00:00
2023-03-14 00:00:00
2023-03-15 00:00:00
2023-03-16 00:00:00
2023-03-17 00:00:00
2023-03-18 00:00:00
2023-03-19 00:00:00
2023-03-20 00:00:00
2023-03-21 00:00:00
2023-03-22 00:00:00
2023-03-23 00:00:00
2023-03-24 00:00:00
2023-03-25 00:00:00
2023-03-26 00:00:00
2023-03-27 00:00:00
2023-03-28 00:00:00
2023-03-29 00:00:00
2023-03-30 00:00:00
2023-03-31 00:00:00
2023-04-01 00:00:00
2023-04-02 00:00:00
2023-04-03 00:00:00
2023-04-04 00:00:00
2023-04-05 00:00:00
2023-04-06 00:00:00
2023-04-07 00:00:00
2023-04-08 00:00:00
2023-04-09 00:00:00
2023-04-10 00:00:00
2023-04-11 00:00:00
2023-04-12 00:00:00
2023-04-13 00:00:00
2023-04-14 00:00:00
2023-04-15 00:00:00
2023-04-16 00:00:00
2023-04-17 00:00:00
2023-04-18 00:00:00
2023-04-19 00:00:00
2023-04-20 00:00:00
2023-04-21 00:00:00
2023-04-22 00:00:00
2023-04-23 00:00:00
2023-04-24 00:00:00
2023-04-25 00:00:00
2023-04-26 00:00:00
2023-04-27 00:00:00
2023-04-28 00:00:00
2023-04-29 00:00:00
2023-04-30 00:00:00
2023-05-01 00:00:00
2023-05-02 00:00:00
2023-05-03 00:00:00
2023-05-04 00:00:00
2023-05-05 00:00:00
2023-05-06 00:00:00
2023-05-07 00:00:00
2023-05-08 00:00:00
2023-05-09 00:00:00
2023-05-10 00:00:00
2023-05-11 00:00:00
2023-05-12 00:00:00
2023-05-13 00:00:00
2023-05-14 00:00:00
2023-05-15 00:00:00
2023-05-16 00:00:00
2023-05-17 00:00:00
2023-05-18 00:00:00
2023-05-19 00:00:00
2023-05-20 00:00:00
2023-05-21 00:00:00
2023-05-22 00:00:00
2023-05-23 00:00:00
2023-05-24 00:00:00
2023-05-25 00:00:00
2023-05-26 00:00:00
2023-05-27 00:00:00
2023-05-28 00:00:00
2023-05-29 00:00:00
2023-05-30 00:00:00
2023-05-31 00:00:00
2023-06-01 00:00:00
2023-06-02 00:00:00
2023-06-03 00:00:00
2023-06-04 00:00:00
2023-06-05 00:00:00
2023-06-06 00:00:00
2023-06-07 00:00:00
2023-06-08 00:00:00
2023-06-09 00:00:00
2023-06-10 00:00:00
2023-06-11 00:00:00
2023-06-12 00:00:00
2023-06-13 00:00:00
2023-06-14 00:00:00
2023-06-15 00:00:00
2023-06-16 00:00:00
2023-06-17 00:00:00
2023-06-18 00:00:00
2023-06-19 00:00:00
2023-06-20 00:00:00
2023-06-21 00:00:00
2023-06-22 00:00:00
2023-06-23 00:00:00
2023-06-24 00:00:00
2023-06-25 00:00:00
2023-06-26 00:00:00
2023-06-27 00:00:00
2023-06-28 00:00:00
2023-06-29 00:00:00
2023-06-30 00:00:00
2023-07-01 00:00:00
2023-07-02 00:00:00
2023-07-03 00:00:00
2023-07-04 00:00:00
2023-07-05 00:00:00
2023-07-06 00:00:00
2023-07-07 00:00:00
2023-07-08 00:00:00
2023-07-09 00:00:00
2023-07-10 00:00:00
2023-07-11 00:00:00
2023-07-12 00:00:00
2023-07-13 00:00:00
2023-07-14 00:00:00
2023-07-15 00:00:00
2023-07-16 00:00:00
2023-07-17 00:00:00
2023-07-18 00:00:00
2023-07-19 00:00:00
2023-07-20 00:00:00
2023-07-21 00:00:00
2023-07-22 00:00:00
2023-07-23 00:00:00
2023-07-24 00:00:00
2023-07-25 00:00:00
2023-07-26 00:00:00
2023-07-27 00:00:00
2023-07-28 00:00:00
2023-07-29 00:00:00
2023-07-30 00:00:00
2023-07-31 00:00:00
2023-08-01 00:00:00
2023-08-02 00:00:00
2023-08-03 00:00:00
2023-08-04 00:00:00
2023-08-05 00:00:00
2023-08-06 00:00:00
2023-08-07 00:00:00
2023-08-08 00:00:00
2023-08-09 00:00:00
2023-08-10 00:00:00
2023-08-11 00:00:00
2023-08-12 00:00:00
2023-08-13 00:00:00
2023-08-14 00:00:00
2023-08-15 00:00:00
2023-08-16 00:00:00
2023-08-17 00:00:00
2023-08-18 00:00:00
2023-08-19 00:00:00
2023-08-20 00:00:00
2023-08-21 00:00:00
2023-08-22 00:00:00
2023-08-23 00:00:00
2023-08-24 00:00:00
2023-08-25 00:00:00
2023-08-26 00:00:00
2023-08-27 00:00:00
2023-08-28 00:00:00
2023-08-29 00:00:00
2023-08-30 00:00:00
2023-08-31 00:00:00
2023-09-01 00:00:00
2023-09-02 00:00:00
2023-09-03 00:00:00
2023-09-04 00:00:00
2023-09-05 00:00:00
2023-09-06 00:00:00
2023-09-07 00:00:00
2023-09-08 00:00:00
2023-09-09 00:00:00
2023-09-10 00:00:00
2023-09-11 00:00:00
2023-09-12 00:00:00
2023-09-13 00:00:00
2023-09-14 00:00:00
2023-09-15 00:00:00
2023-09-16 00:00:00
2023-09-17 00:00:00
2023-09-18 00:00:00
2023-09-19 00:00:00
2023-09-20 00:00:00
2023-09-21 00:00:00
2023-09-22 00:00:00
2023-09-23 00:00:00
2023-09-24 00:00:00
2023-09-25 00:00:00
2023-09-26 00:00:00
2023-09-27 00:00:00
2023-09-28 00:00:00
2023-09-29 00:00:00
2023-09-30 00:00:00
2023-10-01 00:00:00
2023-10-02 00:00:00
```

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]]
df1=pd.DataFrame(lists, columns=['col1', 'col2', 'col3'])
print(df1)
```

Output:

10. Create 2D list to DataFrame

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

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

Python

```
df1=pd.DataFrame(lists, columns=['col1', 'col2', 'col3'])  
print(df1)
```

Python

```
..      col1 col2 col3  
0         1  aaa   22  
1         2  bbb   25  
2         3  ccc   24
```