

Assignment -1
Python Programming

| | |
|---------------------|-------------------|
| Assignment Date | 19 September 2022 |
| Student Name | Suja Marshalin.P |
| Student Roll Number | 813819104101 |
| Maximum Marks | 2 Marks |

1. Split this string

Solution:

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

```
s.split()
```

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

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

1. Split this string

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

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

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

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

Solution:

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

```
planet = "Earth"
```

```
diameter = 12742
```

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

The diameter of Earth is 12742 kilometers



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

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

```
In [5]: planet = "Earth"
        diameter = 12742

In [34]: 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"

Solution:

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

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

```
In [7]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

In [11]: d['k1'][3]['tricky'][3]['target'][3]

Out[11]: 'hello'
```

Numpy

import numpy as np

▼ Numpy

✓  import numpy as np

▼ 4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

✓  [8] np.zeros(10)
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])

✓  [9] np.ones(10)*5
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?



Solution:

```
np.zeros(10)
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
np.ones(10)*5
```

```
array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

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

```
np.arange(20,35,2)
```

```
array([20, 22, 24, 26, 28, 30, 32, 34])
```

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

```
✓ [10] np.arange(20,35,2)  
0s  
array([20, 22, 24, 26, 28, 30, 32, 34])
```

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

Solution:

```
np.arange(0,9).reshape(3,3)
```

```
array([[0, 1, 2],
```

```
       [3, 4, 5],
```

```
       [6, 7, 8]])
```

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

```
✓ [10] np.arange(0,9).reshape(3,3)  
0s  
array([[0, 1, 2],  
       [3, 4, 5],  
       [6, 7, 8]])
```

7. Concatenate a and b

Solution:

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



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

Pandas

▼ 7. Concatenate a and b

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

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

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

8. Create a dataframe with 3 rows and 2 columns

Solution:

```
import pandas as pd
data={'row1':[],'row2':[],'row3':[]}
df=pd.DataFrame(data,columns=['col1','col2'])
print(df)
Empty DataFrame
Columns: [col1, col2]
Index: []
```

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

```
[ ] import pandas as pd

✓ [13] import pandas as pd
Js
      data={'row1':[], 'row2':[], 'row3':[]}
      df=pd.DataFrame(data,columns=['col1','col2'])
      print(df)

      Empty DataFrame
      Columns: [col1, col2]
      Index: []
```

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

Solution:



```
import datetime

day_delta=datetime.timedelta(days=1)

start_date=datetime.date(2023,1,1)
end_date=datetime.date(2023,2,10+1)

for i in range((end_date - start_date).days):
    print(start_date + i*day_delta)
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
```



The screenshot shows a Jupyter Notebook interface in Google Colab. The notebook is titled "Copy of Assignment_1.ipynb". The code cell contains the following Python code:

```
import datetime
day_delta=datetime.timedelta(days=1)
start_date=datetime.date(2023,1,1)
end_date=datetime.date(2023,2,10)
for i in range((end_date - start_date).days):
    print(start_date + i*day_delta)
```

The output of the code is a list of dates from 2023-01-01 to 2023-01-11, displayed one per line. The status bar at the bottom indicates "0s completed at 3:11 PM".

The screenshot shows the same Jupyter Notebook interface. The code cell now displays a list of dates from 2023-01-11 to 2023-02-06, displayed one per line. The status bar at the bottom indicates "0s completed at 3:11 PM".

10. Create 2D list to DataFrame

Solution:

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

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

```
df = pd.DataFrame(lists)
```



Edit with WPS Office

▼ 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]]
```



```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]  
  
df = pd.DataFrame(lists)
```

