```
**Project title** : Intelligent Vehicle Damage Assessment and Cos
Estimator for Insurance Companies
<br>
**Name**: Gowtham P <br>
**Roll Number**:720719104063
```

Project title : Intelligent Vehicle Damage Assessment and Cost Estimator for Insurance Companies
Name: Gowtham P
Roll Number:720719104063

Basic Python

1. Split this string

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

```
s = "Hi there Sam"
s.split()
```

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

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

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

```
planet = "Earth"
diameter = 12742
```

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

```
'The diameter of Earth is 12742 kilometers'
```

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

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

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

```
'hello'
```

Numpy

```
import numpy as np
```

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

```
np.zeros(shape=10)
```

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

```
arr = np.ones(10)*5
print(arr)
```

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

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

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

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

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

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

```
[[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])

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

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

Pandas

8. Create a dataframe with 3 rows and 2 columns

```
import pandas as pd
```

```
records = {
    'Name' : ['user1', 'user2', 'user3'],
    'Age' : [18,19,20]
}

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

	Name	Age
0	user1	18
1	user2	19
2	user3	20

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 date in dates:
    print(date)
```

```
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
2023-01-26 00:00:00
2023-01-27 00:00:00
2023-01-28 00:00:00
2023-01-29 00:00:00
2023-01-30 00:00:00
2023-01-31 00:00:00
2023-02-01 00:00:00
2023-02-02 00:00:00
2023-02-03 00:00:00
2023-02-04 00:00:00
2023-02-05 00:00:00
2023-02-06 00:00:00
2023-02-07 00:00:00
2023-02-08 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
```

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]]
pd.DataFrame(lists)
```

	0	1	2
0	1	aaa	22
1	2	bbb	25
2	3	ccc	24

Double-click (or enter) to edit

Colab paid products - Cancel contracts here

✓ 0s completed at 1:18 AM