

This screenshot shows a Google Colab notebook with three exercises. Exercise 1 involves splitting a string. Exercise 2 involves using the .format() method to create a formatted string. Exercise 3 involves navigating a nested dictionary to retrieve a specific value. The notebook interface includes a file explorer on the left, a code editor in the center, and a taskbar at the bottom.

```
1. Split this string
```

```
[2] s = "Hi there sujil"
```

```
[3] print(s.split())
```

```
['Hi', 'there', 'sujil']
```

2. Use .format() to print the following string.  
Output should be: The diameter of Earth is 12742 kilometers.

```
[10] planet = "Earth"
      diameter = 12742
```

```
[11] print("The diameter of {0} is {1} kilometers.".format(planet,diameter))
```

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

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

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

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

This screenshot shows the continuation of the Google Colab notebook. It includes the completion of exercise 3, followed by a section on Numpy. Two sub-exercises are provided: creating an array of 10 zeros and creating an array of 10 fives. The notebook interface and taskbar are consistent with the previous screenshot.

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

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

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

```
hello
```

Numpy

```
[ ] import numpy as np
```

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

```
[15] a=np.zeros(10)
      print(a)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

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

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

IBM 1 - ravisujitha02@gmail.com x Assignment\_1.ipynb - Colaborat... Assignment\_1.ipynb - Colaborat... IBM-EPBL/IBM-Project-23793-10 x +

colab.research.google.com/drive/1FcMKsMxGrAeCSZp\_5RQUaytr5R9VkbQ#scrollTo=xNpl\_XXoYhs0

+ Code + Text

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

```
[17] a=np.arange(20,35,2)
      print(a)

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

Double-click (or enter) to edit

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

```
[18] m=np.arange(9,9).reshape(3,3)
      print(m)

[[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])

```
[19] a = np.array([1, 2, 3])
      b = np.array([4, 5, 6])
      print(np.concatenate((a,b),axis=0))

[1 2 3 4 5 6]
```

0s completed at 13:54

33°C Very high UV

IBM 1 - ravisujitha02@gmail.com x Assignment\_1.ipynb - Colaborat... Assignment\_1.ipynb - Colaborat... IBM-EPBL/IBM-Project-23793-10 x +

colab.research.google.com/drive/1FcMKsMxGrAeCSZp\_5RQUaytr5R9VkbQ#scrollTo=xNpl\_XXoYhs0

+ Code + Text

8. Create a dataframe with 3 rows and 2 columns

```
[8] import pandas as pd
     import matplotlib.pyplot as plt
     import numpy as np

data=[("Arunshanmugam","Mentor"),("sujitha","leader"),("lakshmi devi","Member")]
print(pd.DataFrame(data,columns=["Name","Role"]))
```

	Name	Role
0	Arunshanmugam	Mentor
1	sujitha	leader
2	lakshmi devi	Member

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

```
[6] from datetime import datetime

date=pd.date_range(start="2023-01-01",end="2023-02-10")
print(pd.Series(date))
```

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

0s completed at 13:54

33°C Partly sunny

IBM 1 - raviujitha02@gmail.com x Assignment\_1.ipynb - Colaborat... Assignment\_1.ipynb - Colaborat... IBM-EPBL/IBM-Project-23793-10 x +

colab.research.google.com/drive/1FcMKaMxGrAeCSZp\_5RQUaytr5R9VkbQ#scrollTo=xNpl\_XXoYhs0

+ Code + Text

✓ [6]

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  
dtype: datetime64[ns]

### 10. Create 2D list to DataFrame

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

```
✓ lists = [[1, 'xxx', 22], [2, 'yyy', 25], [3, 'zzz', 24]]  
pd.DataFrame(lists, columns=["S.No.", "Name", "Quantity"])
```

	S.No.	Name	Quantity
0	1	xxx	22
1	2	yyy	25
2	3	zzz	24

0s completed at 13:54

33°C  
Partly sunny

ENG  
IN 14:09  
18-09-2022