

Assignment -1

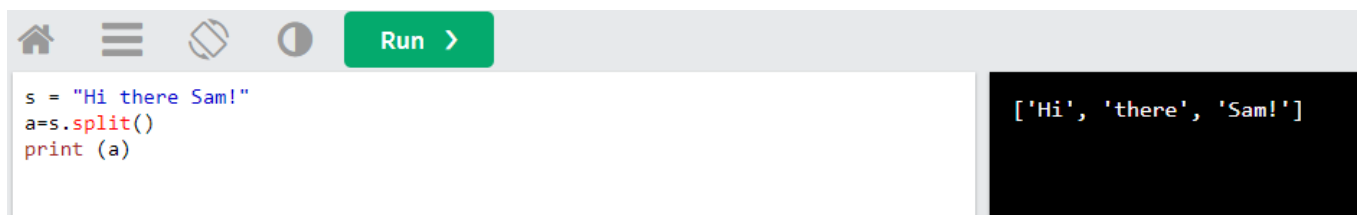
Assignment Date	16 September 2022
Student Name	Tamilselvan D
Student Roll Number	621519104078
Maximum Marks	2 Marks

Question-1:

Split this string

Solution:

```
s = "Hi there Sam!"  
a=s.split()  
print (a)
```



```
s = "Hi there Sam!"  
a=s.split()  
print (a)
```

```
['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))
```



Run >

Result Size: 66B

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

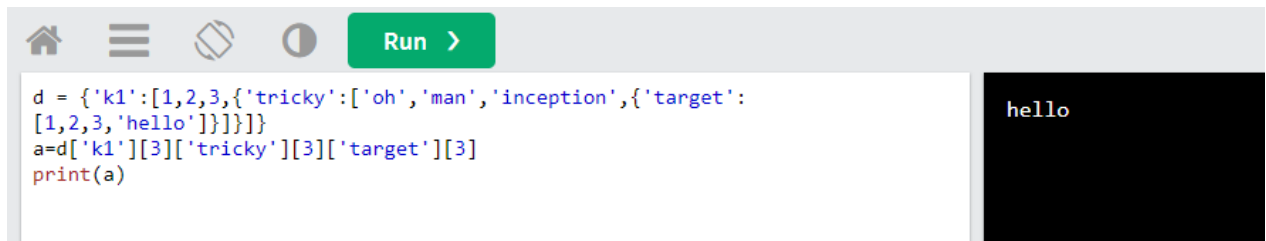
The diameter of Earth is 12742 kilometers.

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



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

hello

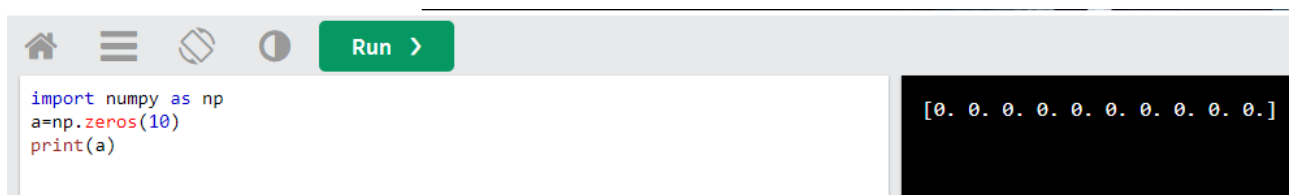
Question-4:

NUMPY

4.1- Create an array of 10 zeros?

Solution:

```
import numpy as np
a=np.zeros(10)
print(a)
```



```
import numpy as np
a=np.zeros(10)
print(a)
```

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

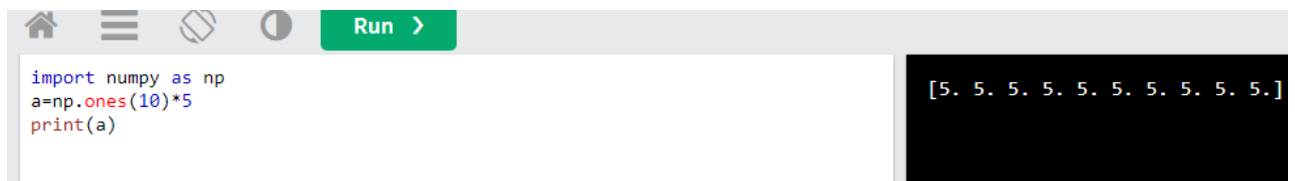
4.2 Create an array of 10 fives?

Solution:

```
import numpy as np
```

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

```
print(a)
```



The image shows a Jupyter Notebook interface. The code editor on the left contains the following code:

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

The output area on the right displays the result of the code execution:

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

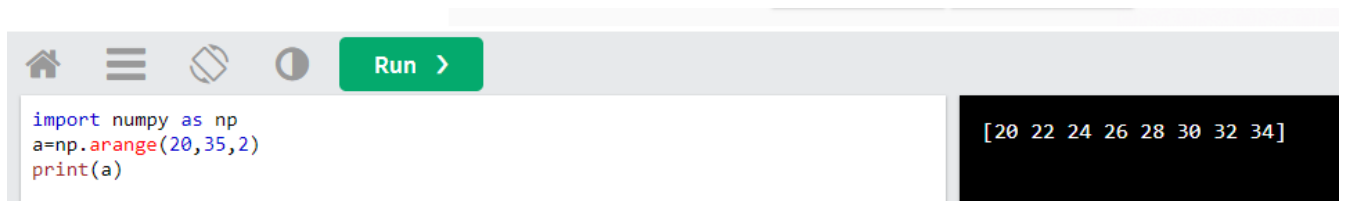
5. Create an array of all the even integers from 20 to 35

Solution:

```
import numpy as np
```

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

```
print(a)
```



The image shows a Jupyter Notebook interface. The code editor on the left contains the following code:

```
import numpy as np
a=np.arange(20,35,2)
print(a)
```

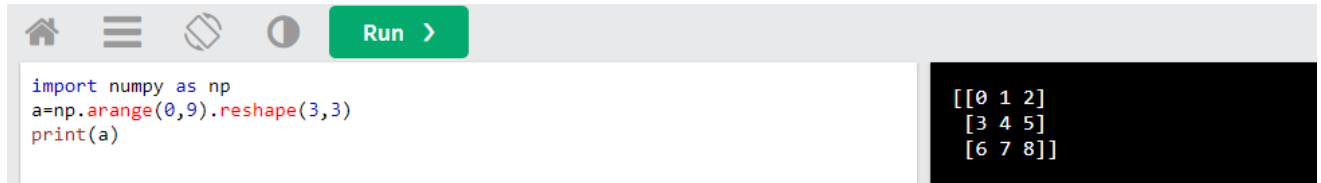
The output area on the right displays the result of the code execution:

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

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

Solution:

```
import numpy as np
a=np.arange(0,9).reshape(3,3)
print(a)
```

A screenshot of a Jupyter Notebook interface. The top bar contains icons for home, menu, undo, and redo, followed by a green 'Run' button with a right arrow. Below the bar, the code cell contains the following Python code:

```
import numpy as np
a=np.arange(0,9).reshape(3,3)
print(a)
```

 The output cell on the right shows the resulting 3x3 array:

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

```
import numpy as np
a=np.arange(0,9).reshape(3,3)
print(a)
```

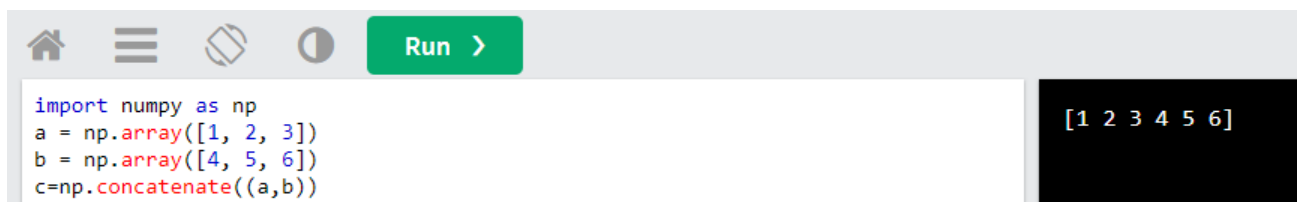
```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

7. Concatenate a and b

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

Solution:

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

A screenshot of a Jupyter Notebook interface. The top bar contains icons for home, menu, undo, and redo, followed by a green 'Run' button with a right arrow. Below the bar, the code cell contains the following Python code:

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

 The output cell on the right shows the resulting 1D array:

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

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

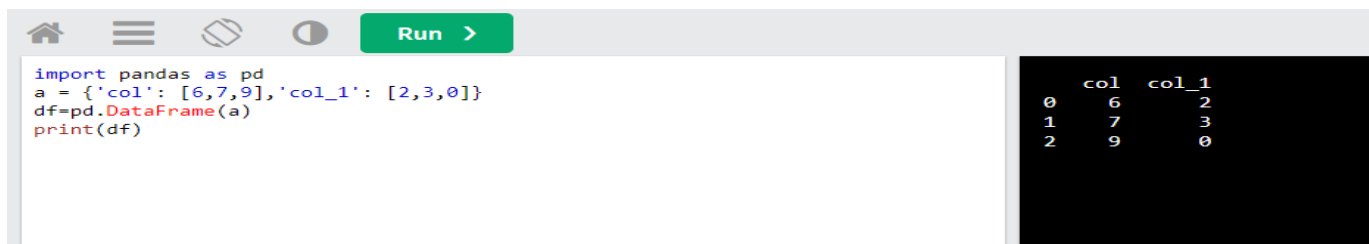
```
[1 2 3 4 5 6]
```

Pandas

8. Create a dataframe with 3 rows and 2 columns

Solution:

```
import pandas as pd
a = {'col': [6,7,9], 'col_1': [2,3,0]}
df=pd.DataFrame(a)
print(df)
```



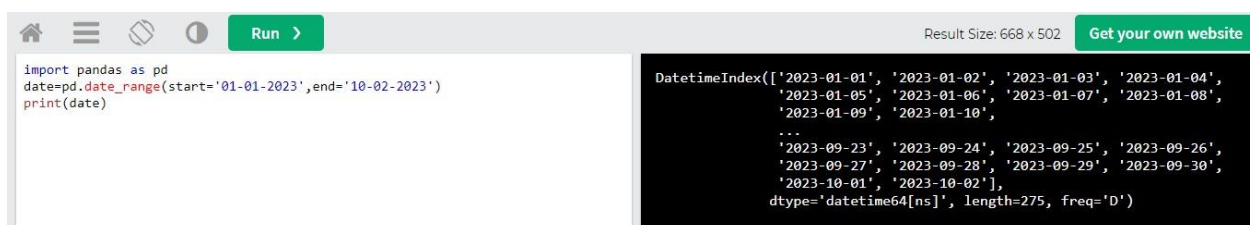
The screenshot shows a Jupyter Notebook interface. On the left, the code cell contains the following Python code: `import pandas as pd`, `a = {'col': [6,7,9], 'col_1': [2,3,0]}`, `df=pd.DataFrame(a)`, and `print(df)`. On the right, the output cell displays the resulting DataFrame as a table with two columns, 'col' and 'col_1', and three rows of data.

	col	col_1
0	6	2
1	7	3
2	9	0

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

Solution:

```
import pandas as pd
date=pd.date_range(start='01-01-2023',end='10-02-2023')
print(date)
```



The screenshot shows a Jupyter Notebook interface. On the left, the code cell contains the following Python code: `import pandas as pd`, `date=pd.date_range(start='01-01-2023',end='10-02-2023')`, and `print(date)`. On the right, the output cell displays the resulting DatetimeIndex as a list of dates from '2023-01-01' to '2023-10-02'. The output also includes metadata: `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')`. The interface also shows a 'Result Size: 668 x 502' and a 'Get your own website' button.

10. Create 2D list to DataFrame

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

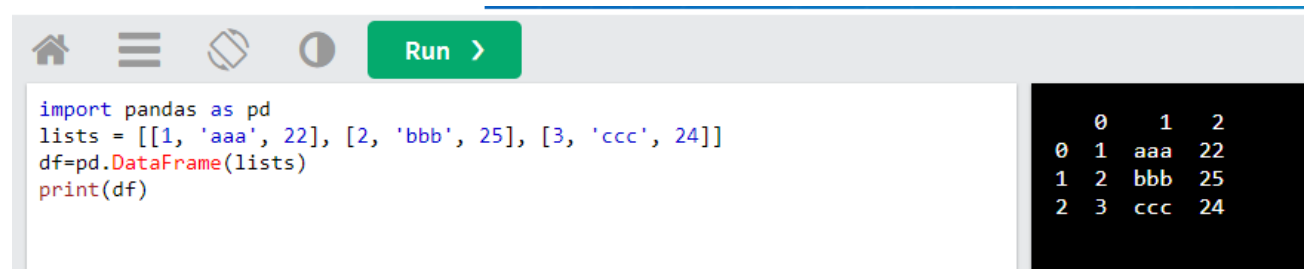
Solution:

```
import pandas as pd
```

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

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

```
print(df)
```



The screenshot shows a Jupyter Notebook interface. The top bar contains icons for home, menu, undo, and redo, along with a green 'Run' button. The code cell contains the following Python code:

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

The output cell displays the resulting DataFrame as a table:

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

