

# Assignment -1

## Python Programming

Assignment Date	:	15 October 2022
Student Name	:	Rithika R
Student Roll Number	:	73771914158
Maximum Marks	:	2 Marks

### Basic Python

#### Question-1:

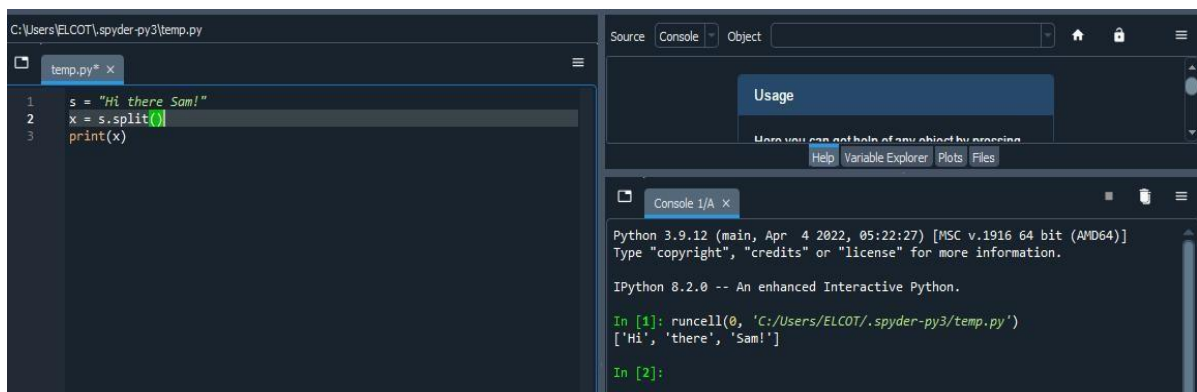
Split this string

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

Solution:

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

Output:



#### Question-2:

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

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

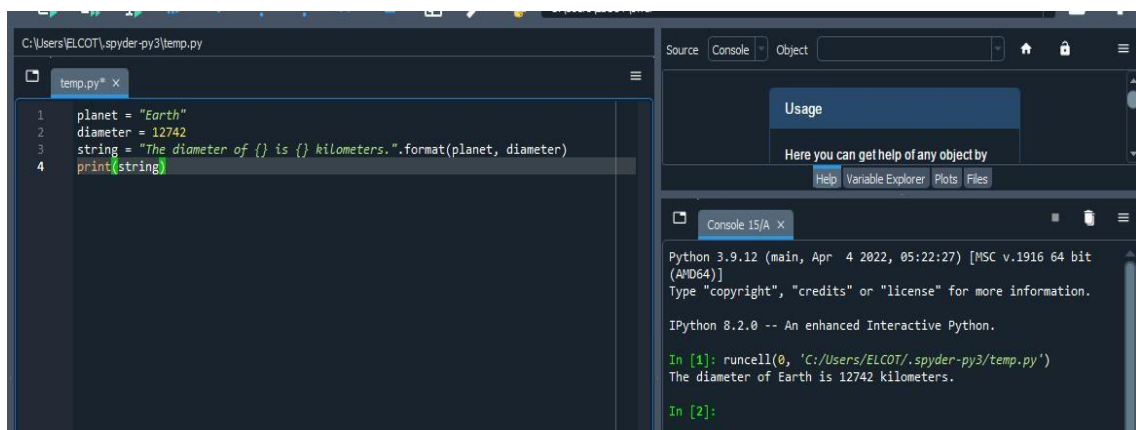
```
planet = "Earth"  
diameter = 12742
```

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

**Solution:**

The diameter of Earth is 12742 kilometers.

**Output:**



The screenshot shows the Spyder Python IDE interface. The editor on the left contains a script named 'temp.py' with the following code:

```
1 planet = "Earth"  
2 diameter = 12742  
3 string = "The diameter of {} is {} kilometers.".format(planet, diameter)  
4 print(string)
```

The right-hand pane shows the IPython console with the following output:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.  
IPython 8.2.0 -- An enhanced Interactive Python.  
  
In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')  
The diameter of Earth is 12742 kilometers.  
  
In [2]:
```

**Question-3:**

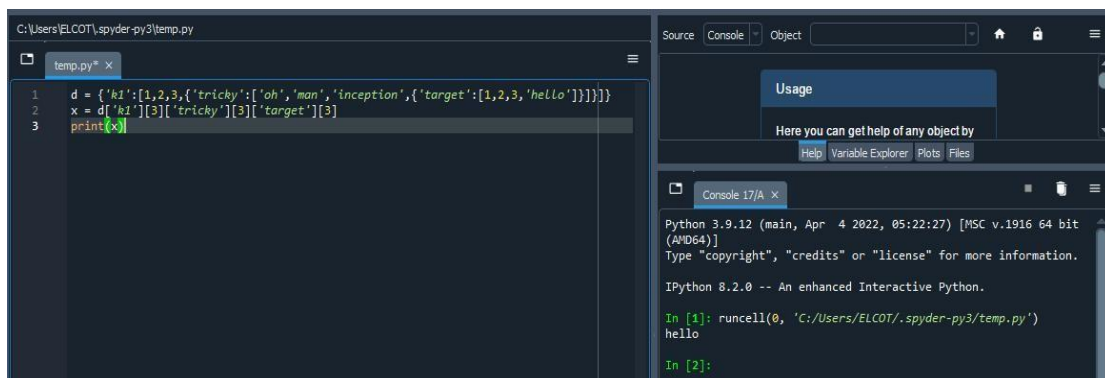
In this nest dictionary grab the word "hello"

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

**Solution:**

Hello

**Output:**



The screenshot shows the Spyder Python IDE interface. The editor on the left contains a script named 'temp.py' with the following code:

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

The right-hand pane shows the IPython console with the following output:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.  
IPython 8.2.0 -- An enhanced Interactive Python.  
  
In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')  
hello  
  
In [2]:
```

# Numpy

## Question-4:

```
import numpy as np
```

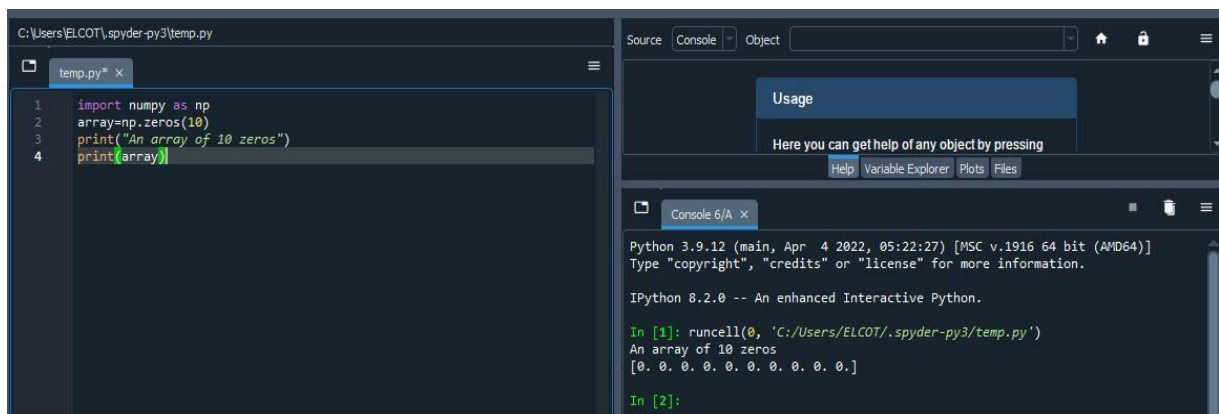
### 4.1 Create an array of 10 zeros?

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

Solution:

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

Output:



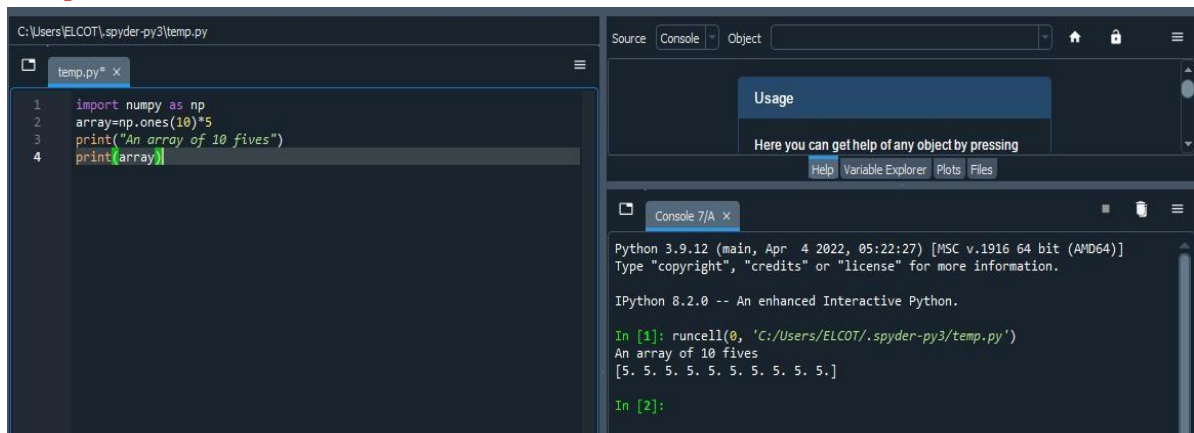
### 4.2 Create an array of 10 fives?

```
import numpy as np
Array=np.ones(10)*5
print("An array of 10 fives")
print(array)
```

Solution:

```
An array of 10 fives
[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]
```

Output:



The screenshot shows the Spyder IDE interface. The left pane displays a file named 'temp.py' with the following code:

```
1 import numpy as np
2 array=np.ones(10)*5
3 print("An array of 10 fives")
4 print(array)
```

The right pane shows the IPython console with the following output:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')
An array of 10 fives
[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]

In [2]:
```

**Question-5:**

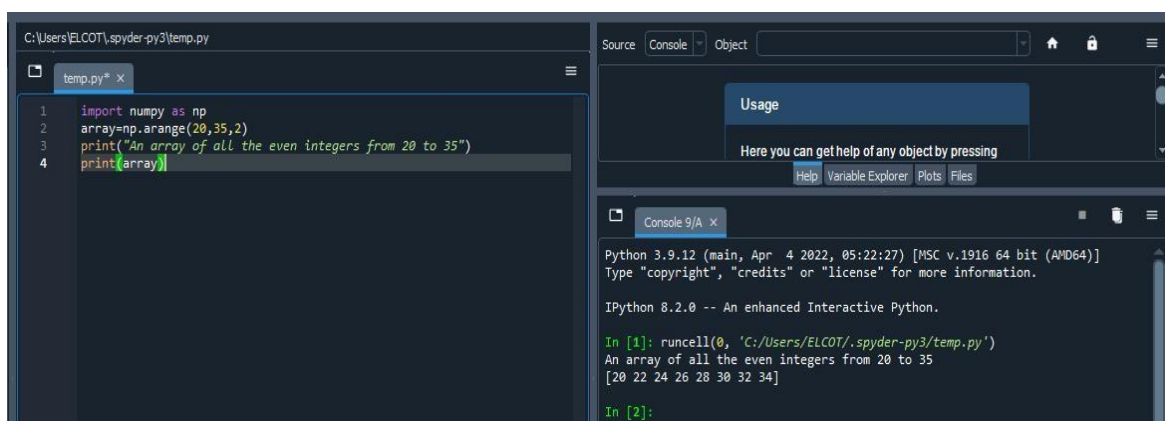
Create an array of all the even integers from 20 to 35

```
import numpy as np
array=np.arange(20,35,2)
print("An array of all the even integers from 20 to 35")
print(array)\
```

Solution:

An array of all the even integers from 20 to 35  
[20 22 24 26 28 30 32 34]

Output:



The screenshot shows the Spyder IDE interface. The left pane displays a file named 'temp.py' with the following code:

```
1 import numpy as np
2 array=np.arange(20,35,2)
3 print("An array of all the even integers from 20 to 35")
4 print(array)
```

The right pane shows the IPython console with the following output:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')
An array of all the even integers from 20 to 35
[20 22 24 26 28 30 32 34]

In [2]:
```

### Question-6:

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

```
import numpy as np
matrix = np.arange(0, 9).reshape(3,3)
print("A 3X3 matrix with values ranging from 0 to 8\n")
print(matrix)
```

Solution:

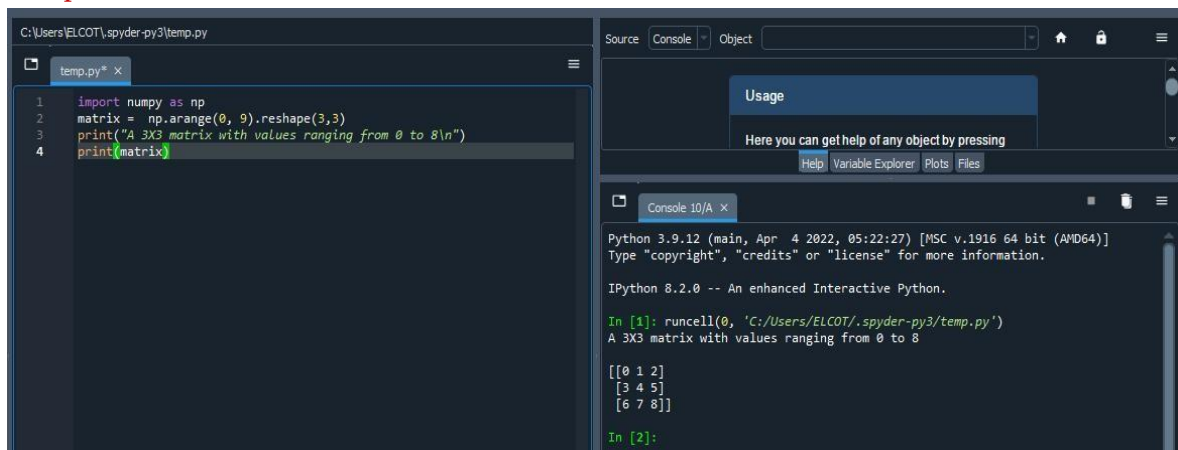
A 3X3 matrix with values ranging from 0 to 8

[[0 1 2]

[3 4 5]

[6 7 8]]

Output:

The screenshot shows the Spyder Python IDE interface. On the left, a code editor window titled 'temp.py' contains the following Python code:

```
1 import numpy as np
2 matrix = np.arange(0, 9).reshape(3,3)
3 print("A 3X3 matrix with values ranging from 0 to 8\n")
4 print(matrix)
```

The right side of the IDE is split into two panels. The top panel, labeled 'Console', shows the output of the script:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')
A 3X3 matrix with values ranging from 0 to 8

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

In [2]:
```

The bottom panel, labeled 'Object Explorer', is currently empty.

### Question-7:

Concatenate a and b

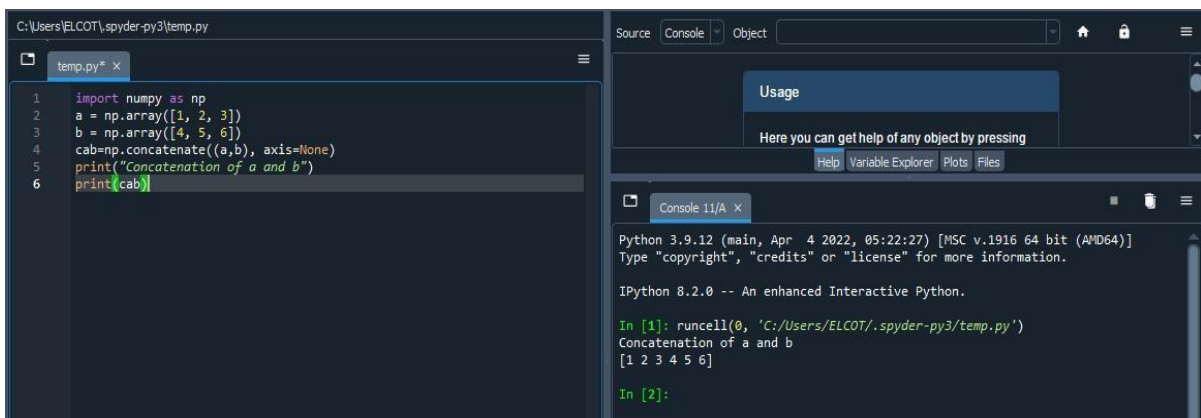
```
a = np.array([1, 2, 3]),
b = np.array([4, 5, 6])
```

```
import numpy as np
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
cab=np.concatenate((a,b), axis=None)
print("Concatenation of a and b")
print(cab)
```

**Solution:**

```
Concatenation of a and b
[1 2 3 4 5 6]
```

**Output:**



## Pandas

Question-8:

```
import pandas as pd
```

Create a dataframe with 3 rows and 2 columns

```
import pandas as pd
data = {'student_name': ['muthamizhan', 'karthik', 'Ravi'],
        'cgpa_marks': [9.4, 8.9, 8.7]}
df = pd.DataFrame(data)
df=df.to_string(index=False)
print("A dataframe with 3 rows and 2 columns\n")
print (df)
```

Solution:

A dataframe with 3 rows and 2 columns

student_name	cgpa_marks
muthamizhan	9.4
karthik	8.9
Ravi	8.7

Output:

The screenshot shows the Spyder Python IDE interface. On the left, a script named 'temp.py' is open, containing the following code:

```
1 import pandas as pd
2 data = {'student_name': ['muthamizhan', 'karthik', 'Ravi'],
3         'cgpa_marks': [9.4, 8.9, 8.7]}
4
5 df = pd.DataFrame(data)
6 df=df.to_string(index=False)
7
8 print("A dataframe with 3 rows and 2 columns\n")
9 print(df)
```

On the right, the IPython console shows the execution of the script. The output is:

```
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')
A dataframe with 3 rows and 2 columns

student_name cgpa_marks
muthamizhan    9.4
karthik        8.9
Ravi           8.7

In [2]:
```

## Question-9

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

```
import pandas as pd
import datetime
start = datetime.datetime.strptime("01-01-2023", "%d-%m-%Y")
date_generated = pd.date_range(start, periods=41)
print(date_generated.strftime("%d-%m-%Y"))
```

Solution:

```
Index(['01-01-2023', '02-01-2023', '03-01-2023', '04-01-2023',
      '05-01-2023', '06-01-2023', '07-01-2023', '08-01-2023',
      '09-01-23', '10-01-2023', '11-01-2023', '12-01-2023',
      '13-01-2023', '14-01-2023', '15-01-2023', '16-01-2023',
```

```

'17-01-2023', '18-01-2023', '19-01-2023', '20-01-2023',
'21-01-2023', '22-01-2023', '23-01-2023', '24-01-2023',
'25-01-2023', '26-01-2023', '27-01-2023', '28-01-2023',
'29-01-2023', '30-01-2023', '31-01-2023', '01-02-2023',
'02-02-2023', '03-02-2023', '04-02-2023', '05-02-2023',
'06-02-2023', '07-02-2023', '08-02-2023', '09-02-2023',
'10-02-2023'],
dtype='object')

```

Output:

The screenshot shows the Spyder Python IDE interface. On the left, a file named 'temp.py' is open, containing the following code:

```

1 import pandas as pd
2 import datetime
3 start = datetime.datetime.strptime("01-01-2023", "%d-%m-%Y")
4 date_generated = pd.date_range(start, periods=41)
5 print(date_generated.strftime("%d-%m-%Y"))

```

On the right, the IPython console shows the output of the script:

```

Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')
Index(['01-01-2023', '02-01-2023', '03-01-2023', '04-01-2023', '05-01-2023',
      '06-01-2023', '07-01-2023', '08-01-2023', '09-01-2023', '10-01-2023',
      '11-01-2023', '12-01-2023', '13-01-2023', '14-01-2023', '15-01-2023',
      '16-01-2023', '17-01-2023', '18-01-2023', '19-01-2023', '20-01-2023',
      '21-01-2023', '22-01-2023', '23-01-2023', '24-01-2023', '25-01-2023',
      '26-01-2023', '27-01-2023', '28-01-2023', '29-01-2023', '30-01-2023',
      '31-01-2023', '01-02-2023', '02-02-2023', '03-02-2023', '04-02-2023',
      '05-02-2023', '06-02-2023', '07-02-2023', '08-02-2023', '09-02-2023',
      '10-02-2023'],
      dtype='object')

In [2]:

```

Question-10:

Create 2D list to DataFrame

```

import pandas as pd

lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
df = pd.DataFrame(lists, columns=['no', 'name', 'd_no'])
df=df.to_string(index=False)
print("Given 2D list")
print(lists)
print("\n2D list to dataframe")
print(df)

```



Solution:

Given 2D list

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

2D list to dataframe

no name    d\_no

1    aaa        22

2    bbb        25

3    ccc        24

Output:

The screenshot shows a Jupyter Notebook interface with a code editor on the left and a console on the right. The code in the editor defines a 2D list, converts it to a DataFrame, and prints the results. The console output shows the execution of the code, including the 2D list and the resulting DataFrame.

```
1 import pandas as pd
2 lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
3 lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
4 df = pd.DataFrame(lists, columns=['no', 'name', 'd_no'])
5 df=df.to_string(index=False)
6 print("Given 2D List")
7 print(lists)
8 print("\n2D List to dataframe")
9 print(df)
```

Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [4]: runcell(0, 'C:/Users/ELCOT/.spyder-py3/temp.py')

Given 2D list

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

2D list to dataframe

```
no name d_no
1 aaa 22
2 bbb 25
3 ccc 24
```

In [2]: