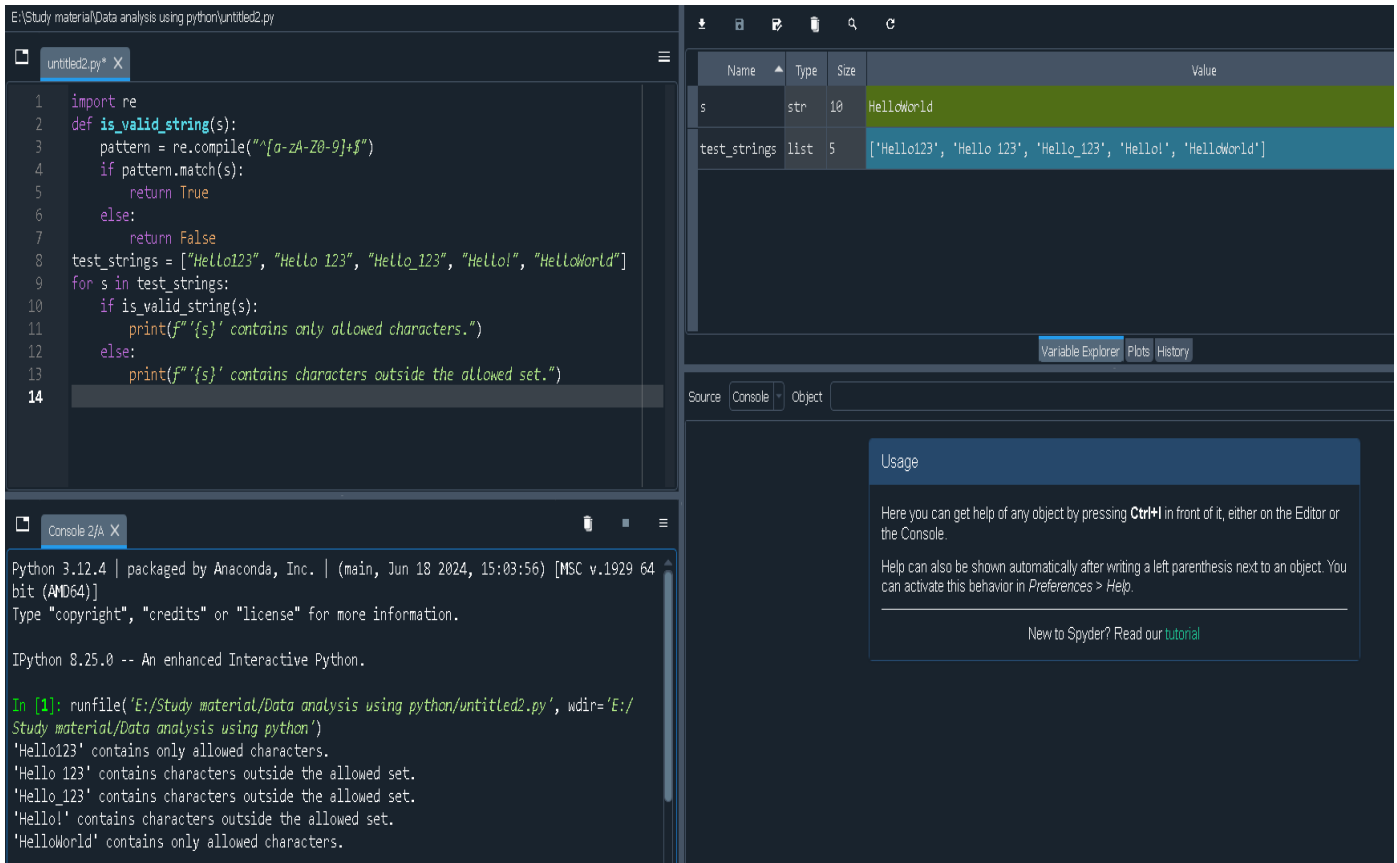


MODULE – 6 ASSIGNMENT

Regular Expression

1) Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9)



The screenshot displays the Anaconda Spyder IDE interface. The main editor window shows a Python script named `untitled2.py` with the following code:

```
1 import re
2 def is_valid_string(s):
3     pattern = re.compile("[a-zA-Z0-9]+$")
4     if pattern.match(s):
5         return True
6     else:
7         return False
8 test_strings = ["Hello123", "Hello 123", "Hello_123", "Hello!", "HelloWorld"]
9 for s in test_strings:
10     if is_valid_string(s):
11         print(f"{s} contains only allowed characters.")
12     else:
13         print(f"{s} contains characters outside the allowed set.")
14
```

The Variable Explorer on the right shows the following variables:

Name	Type	Size	Value
s	str	10	HelloWorld
test_strings	list	5	['Hello123', 'Hello 123', 'Hello_123', 'Hello!', 'HelloWorld']

The IPython console at the bottom shows the output of the script:

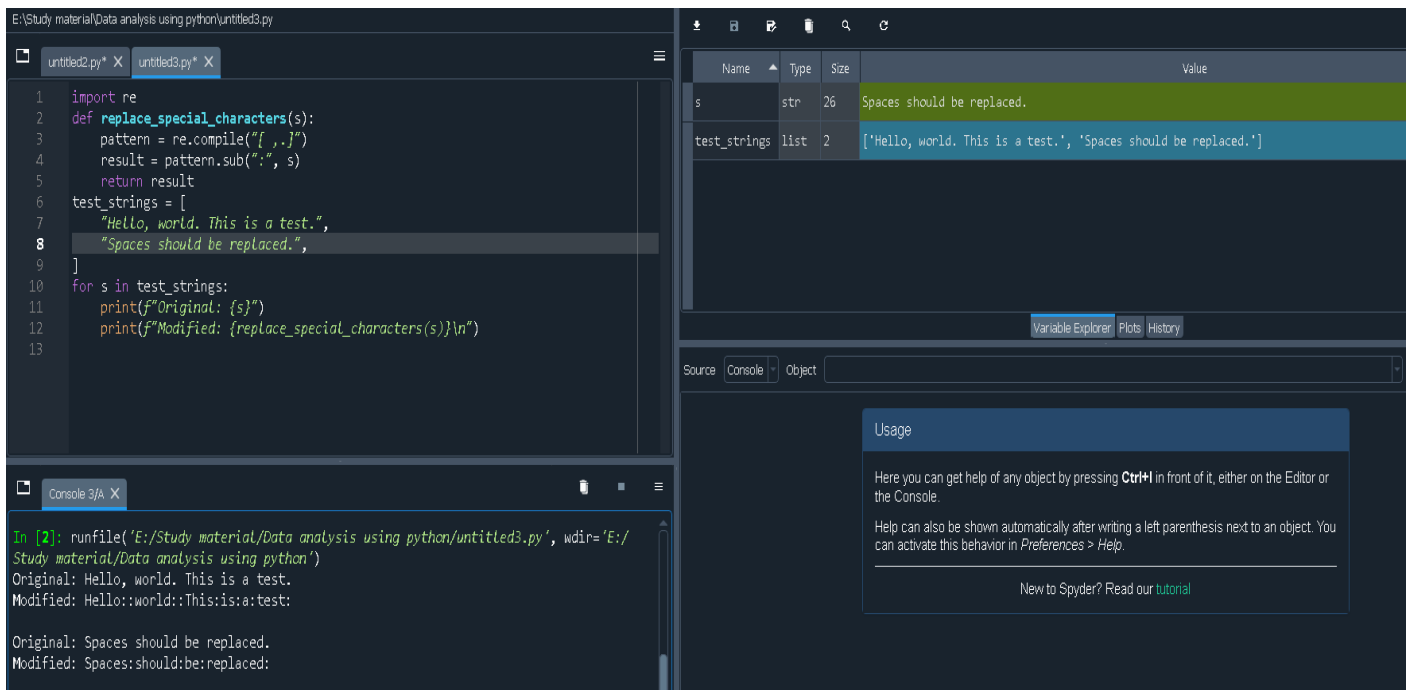
```
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 8.25.0 -- An enhanced Interactive Python.

In [1]: runfile('E:/Study material/Data analysis using python/untitled2.py', wdir='E:/Study material/Data analysis using python')
'Hello123' contains only allowed characters.
'Hello 123' contains characters outside the allowed set.
'Hello_123' contains characters outside the allowed set.
'Hello!' contains characters outside the allowed set.
'HelloWorld' contains only allowed characters.
```

The Variable Explorer also includes tabs for Variable Explorer, Plots, and History. The Console tab is currently selected, showing the output of the script. A help panel is visible on the right side of the console, titled "Usage", providing information on how to get help for any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console. It also mentions that help can be shown automatically after writing a left parenthesis next to an object, and that this behavior can be activated in **Preferences > Help**. A link to the tutorial is provided: [New to Spyder? Read our tutorial](#).

2) Write a Python program to replace all occurrences of space, comma, or dot with a colon.



The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script named `untitled3.py` with the following code:

```
1 import re
2 def replace_special_characters(s):
3     pattern = re.compile("[ ,.]")
4     result = pattern.sub(":", s)
5     return result
6 test_strings = [
7     "Hello, world. This is a test.",
8     "Spaces should be replaced.",
9 ]
10 for s in test_strings:
11     print(f"Original: {s}")
12     print(f"Modified: {replace_special_characters(s)}\n")
13
```

The Variable Explorer on the right shows the following variables:

Name	Type	Size	Value
s	str	26	Spaces should be replaced.
test_strings	list	2	['Hello, world. This is a test.', 'Spaces should be replaced.']

The Console window at the bottom shows the output of the script:

```
In [2]: runfile('E:/Study material/Data analysis using python/untitled3.py', wdir='E:/Study material/Data analysis using python')
Original: Hello, world. This is a test.
Modified: Hello:world:This:is:a:test:

Original: Spaces should be replaced.
Modified: Spaces:should:be:replaced:
```

A help dialog box is visible in the bottom right corner, titled "Usage", providing information on how to get help for any object by pressing **Ctrl+H**.