Sherin Soman

August 07, 2021

Assignment 05

<https://github.com/SherinJoel/IntroToProg-Python>

**Create a script using a list of dictionaries and text files**

**Introduction**

In this Assignment, I explained the steps I have done to create a script using a list of dictionaries and text files. I start my program by loading the data in a text file called ToDoList.txt into a python list of dictionaries rows. The script continues to display a menu of choices to the user until the user ask to exit the program. I used a printed "menu" to guide the user through this process.

**Create a new Project in PyCharm**

To create a new Project in PyCharm, I created a sub-folder called Assignment 05 inside of the \_PythonClass folder and used \_PythonClass\Assignment05 as its location to create the new project. Within the project, I added the starter file, "Assignment05\_Starter.py.

**Adding code to the Python Script**

In this assignment, I modified a starter script that manages a "ToDo list." I started my script by updating the change log in the script's header.

***Step 1 - Load the any data in a text file called ToDoList.txt into a python list of dictionaries rows***

To load the data from the text file, I opened the text file ToDoList.txt in“r” mode using a built-in open() function.

# *How To Read File In Python*

* *Using Python inbuilt functions*
* **read()** method - for reading the whole content of the file.
* **readline()** method - return one line
* **readlines()** method **-** returns a list containing each line in the file as a list element.

* *Read data using for loop*

By looping through the lines of the file, we can read the whole file, line by line.

In this assignment, I used a for loop to read the string data from the file line by line and split it into a list using the string split() method. The variable strData is used to represent a row of text data from the file.

split() **method** - returns the list of strings after breaking the given string by a specified separator.

The string splits into list at the specified separator (comma) and is stored in the list variable lstData. Each element in the lstData is accessed using the index number and is stored as a dictionary value in the variable dicRow. Task and Priority are the keys used in the dictionary.

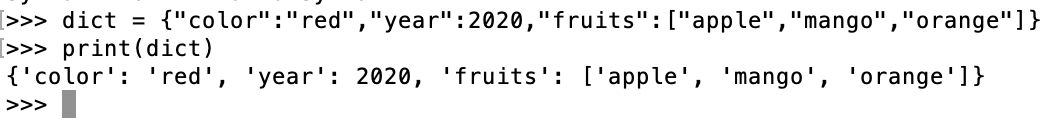
**strip() method** - removes any leading (spaces at the beginning) and trailing (spaces at the end) characters (space is the default leading character to remove)

***Python Dictionary***

***Dictionaries written with curly brackets are used to store data values in key:value pairs.***

A dictionary is a collection which is ordered, changeable and does not allow duplicates.

The values in dictionary items can be of any data type.

 Figure 1: Dictionary containing items of different data types

The **append** () method - appends an element to the end of the list.

Each time the for loop runs, the dictionary is appended using the append() method to the list variable lstTable that acts as a 'Table' of rows. Once the for loop ends, the file is closed using the close() function.

***Step 2 - Display a menu of choices to the user***

A menu of choices is displayed to the user using the print() function. The variable *strChoice* is used to capture the user option selection.While (True) is an infinite loop that executes the loop body indefinitely until the user enters the exit option.

***Step 3 - Show the current items in the table***

If the user opted strChoice = = 1, then the current items in the list table are displayed.

**len()** which returns the length of the list determines if the list is empty or not. If the length of the lstTable is not zero, the items in the table are printed using a for loop.

The **continue statement** forces the loop to jump back to the beginning and reevaluate the condition.

***Step 4 - Add a new item to the list/Table***

If the user opted strChoice = = 2, then the input() function asks the user to input a task and priority and stores it in variables todo\_task and todo\_priority. The values are stored as dictionary values for the keys 'Tasks' and Priority' and appended to the list variable lstTable.

***Step 5 - Remove a new item from the list/Table***

Python uses the following list methods to remove elements from the list

* clear() - Remove all items
* **pop()** - Remove an item by index and get its value
* **remove()** - Remove an item by value
* **del()** - Remove items by index or slice. The first index is 0 and last index is -1.

***Step 6 - Save tasks to the ToDoToDoList.txt file***

To write to the ToDoToDoList.txt file, the file is opened in the “w” mode.

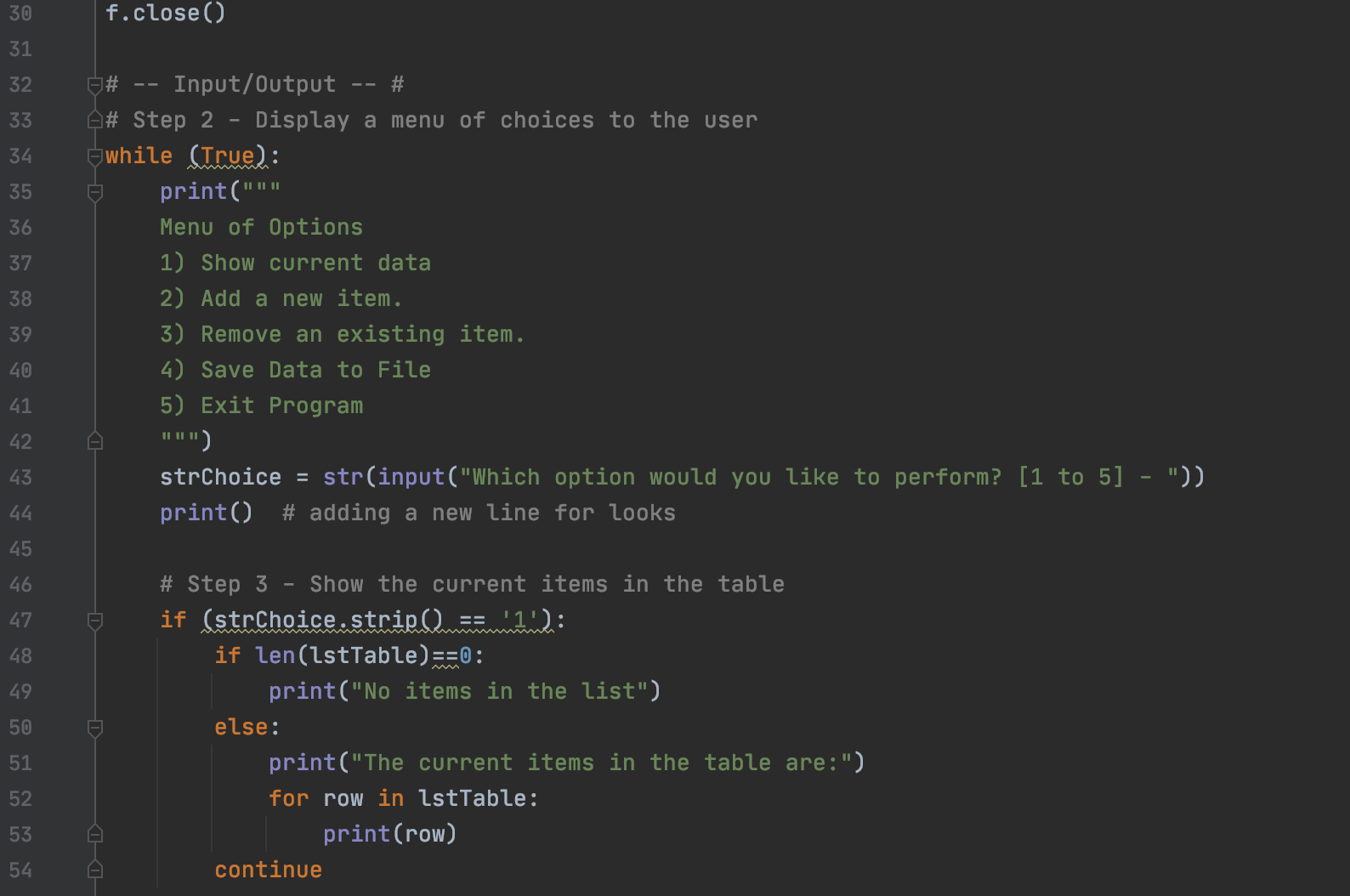
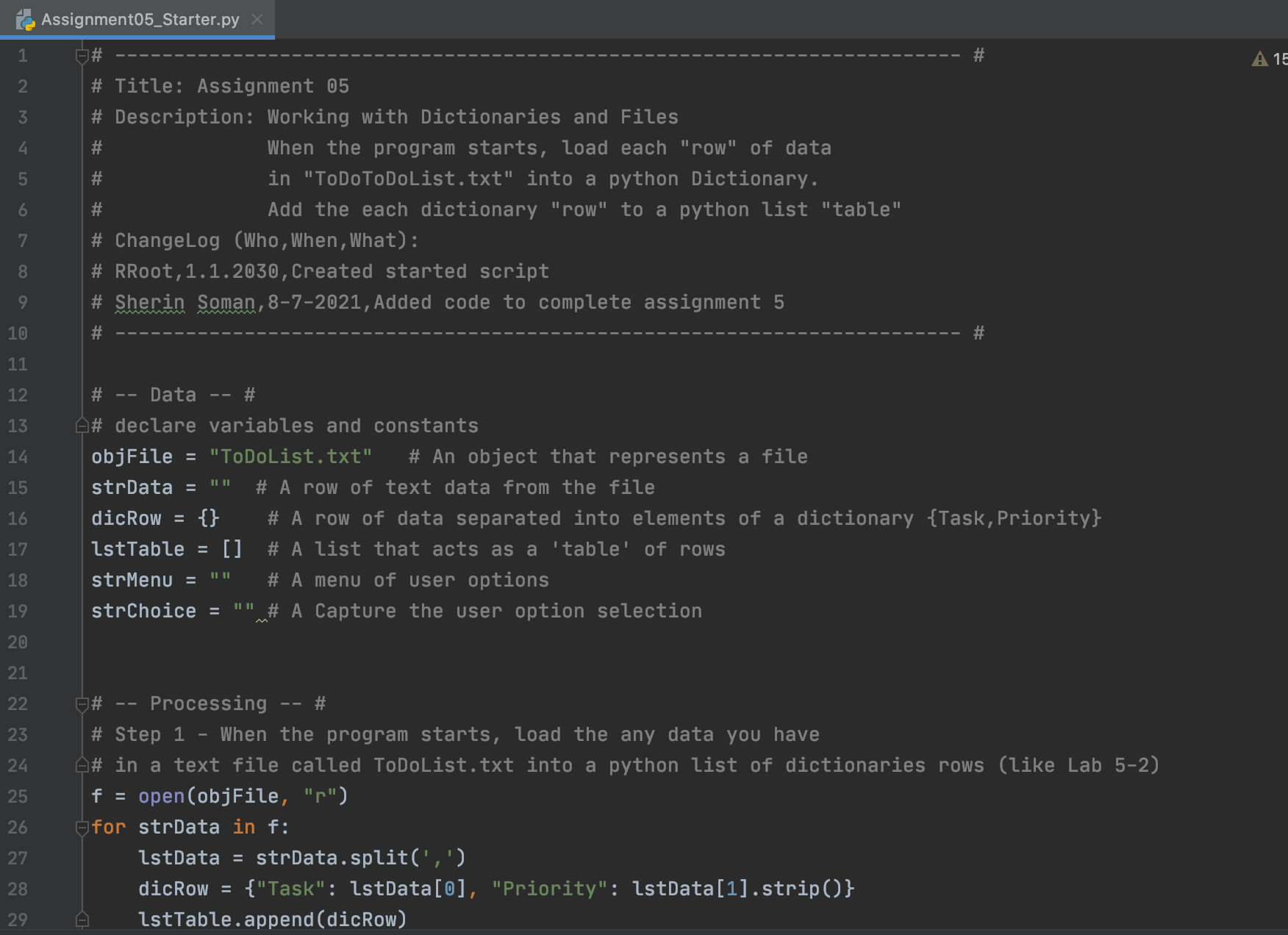
Python has a set of built-in methods that you can use on dictionaries.

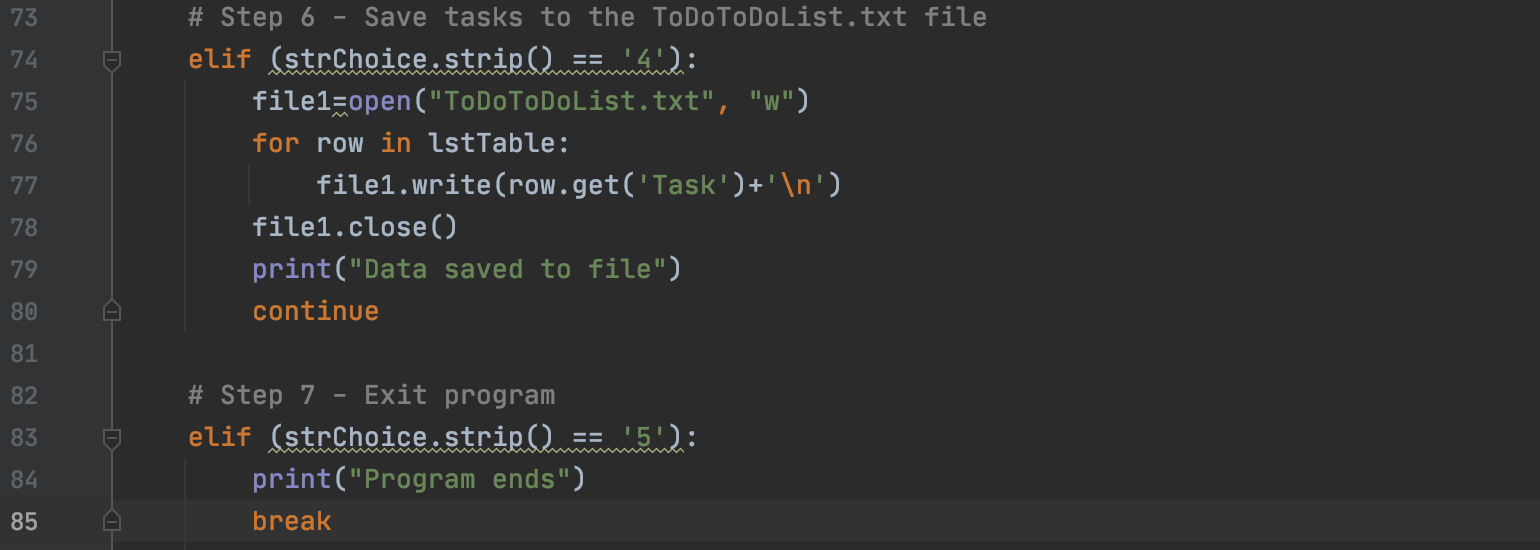
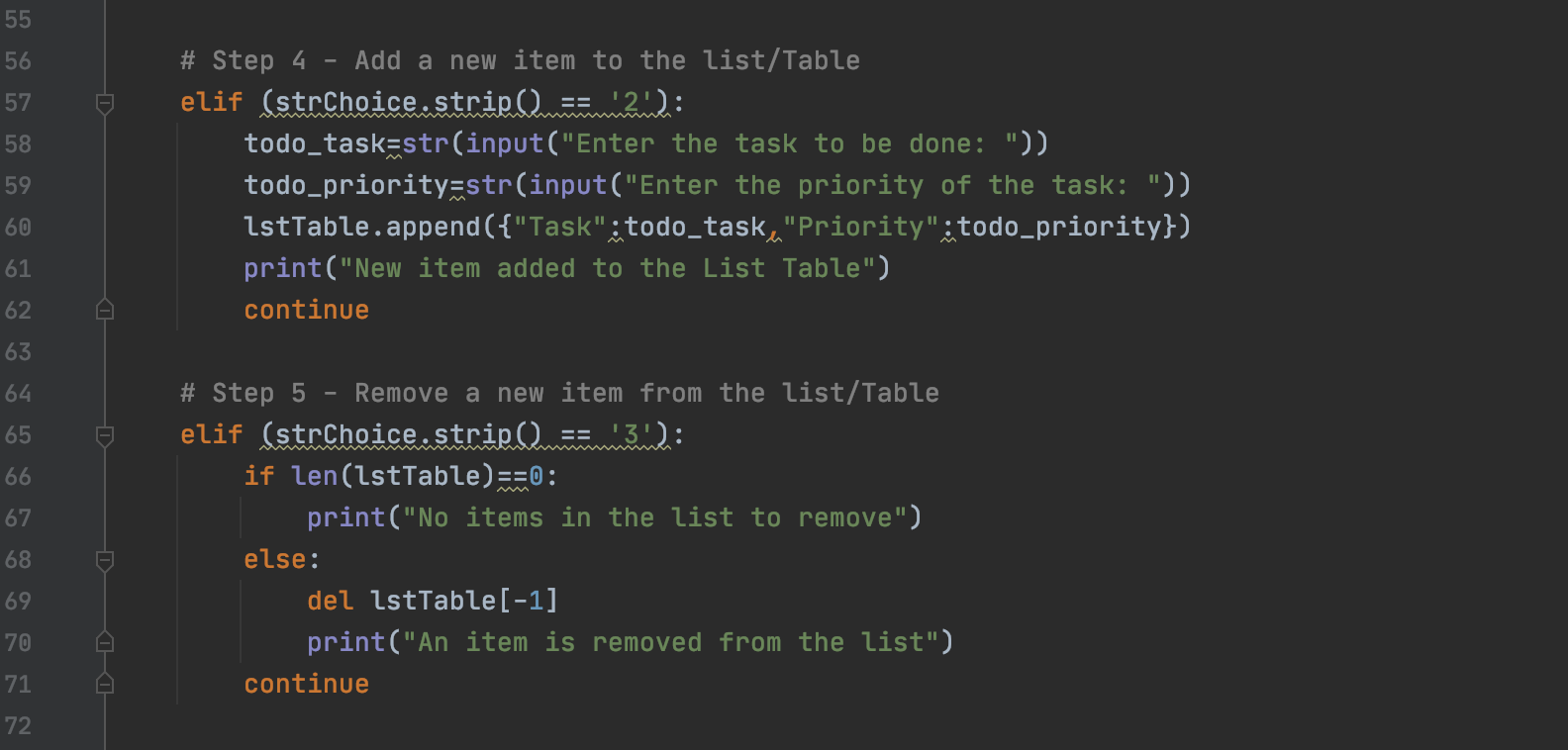
* **get()** - returns a value for the given key. If key is not available, then returns default value None.
* **items() -** Returns a list containing a tuple for each key value pair
* **keys() -** Returns a list containing the dictionary's keys
* **values() -** Returns a list of all the values in the dictionary

get() is used here to get the value for the specified key 'Task' and is written to the file using the file write().

***Step 7 – Exit program***

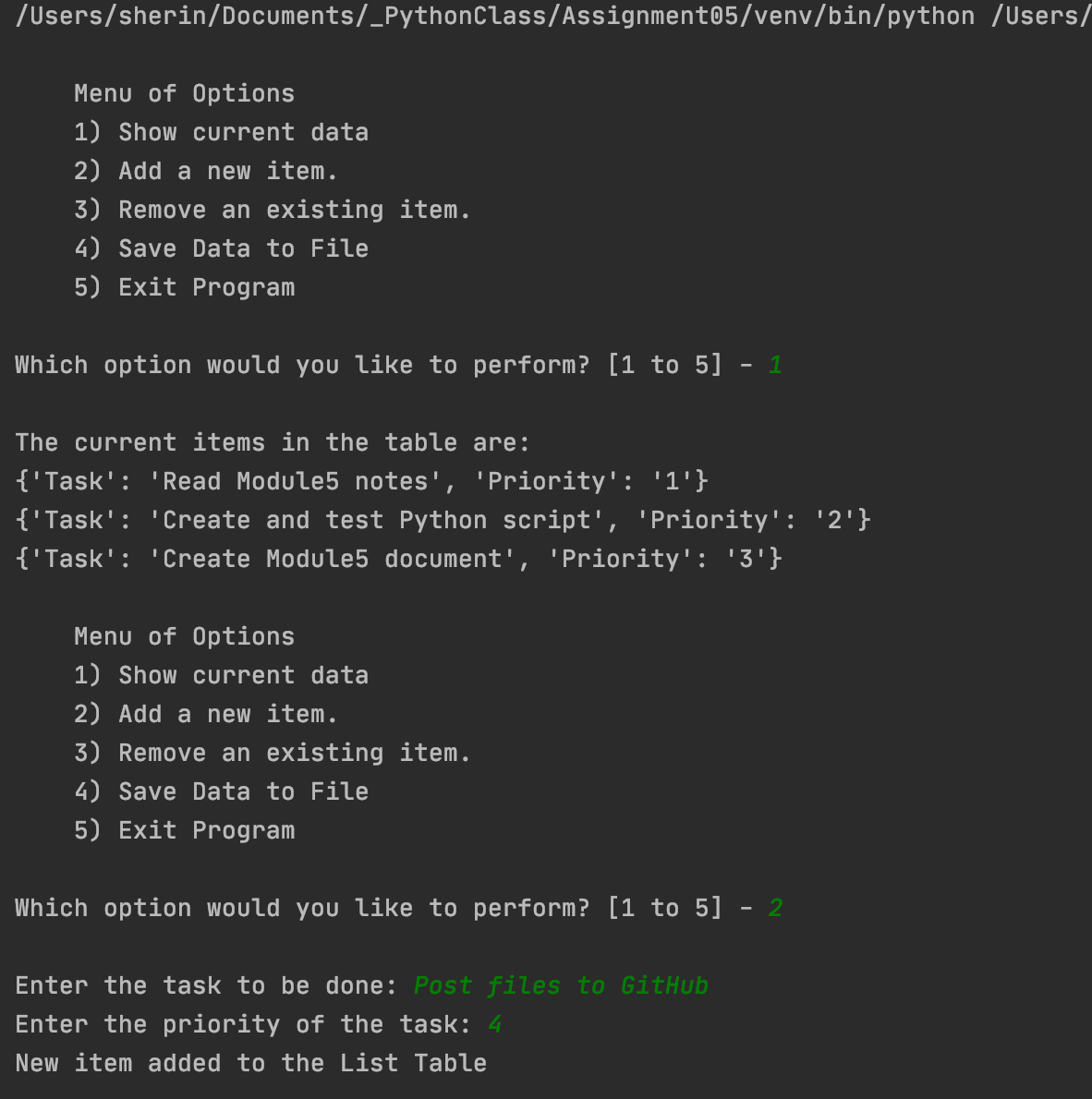
The break statement is used to exit from the while loop

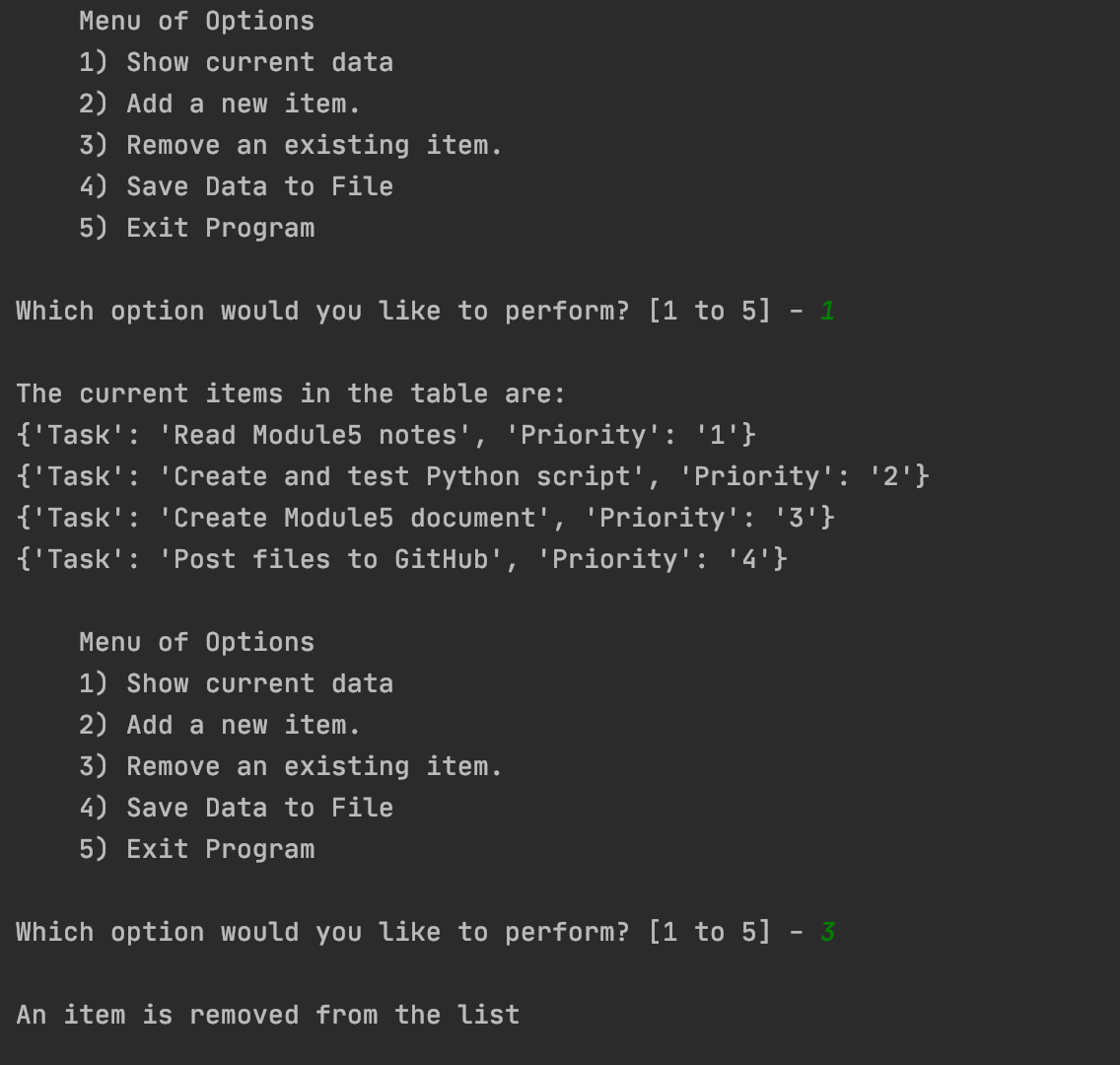
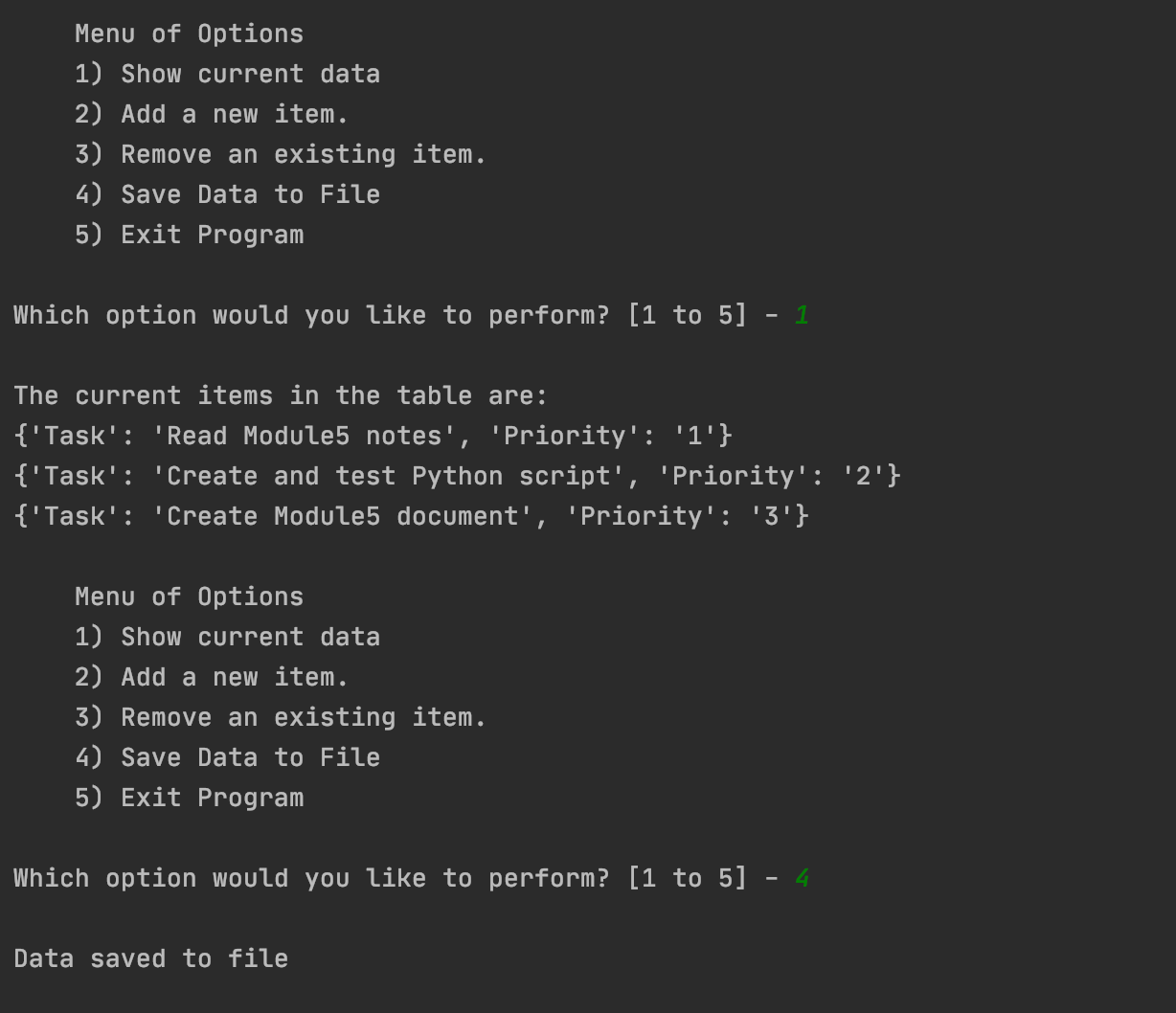
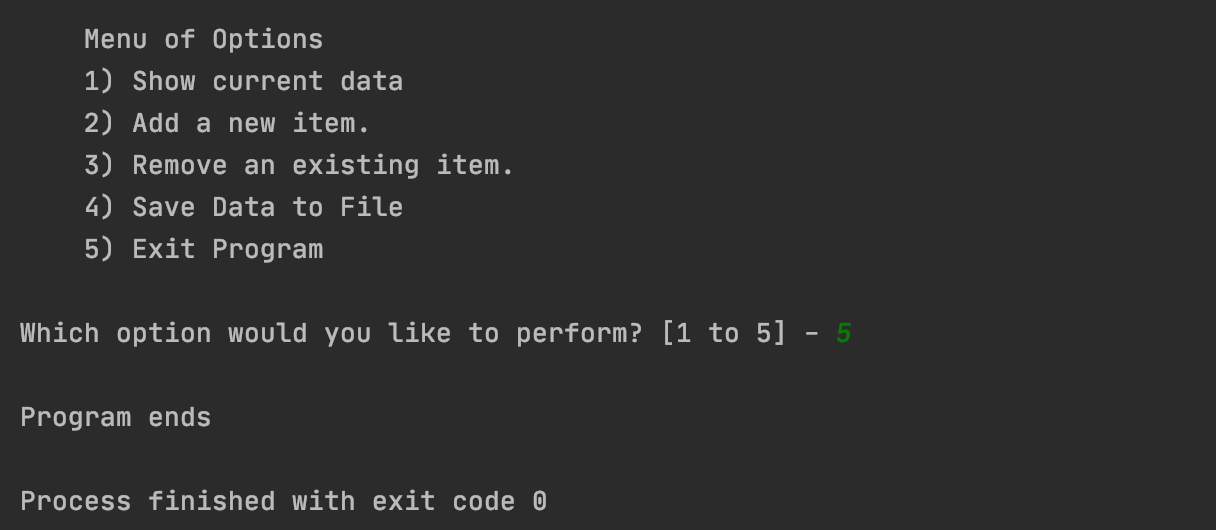
******

******

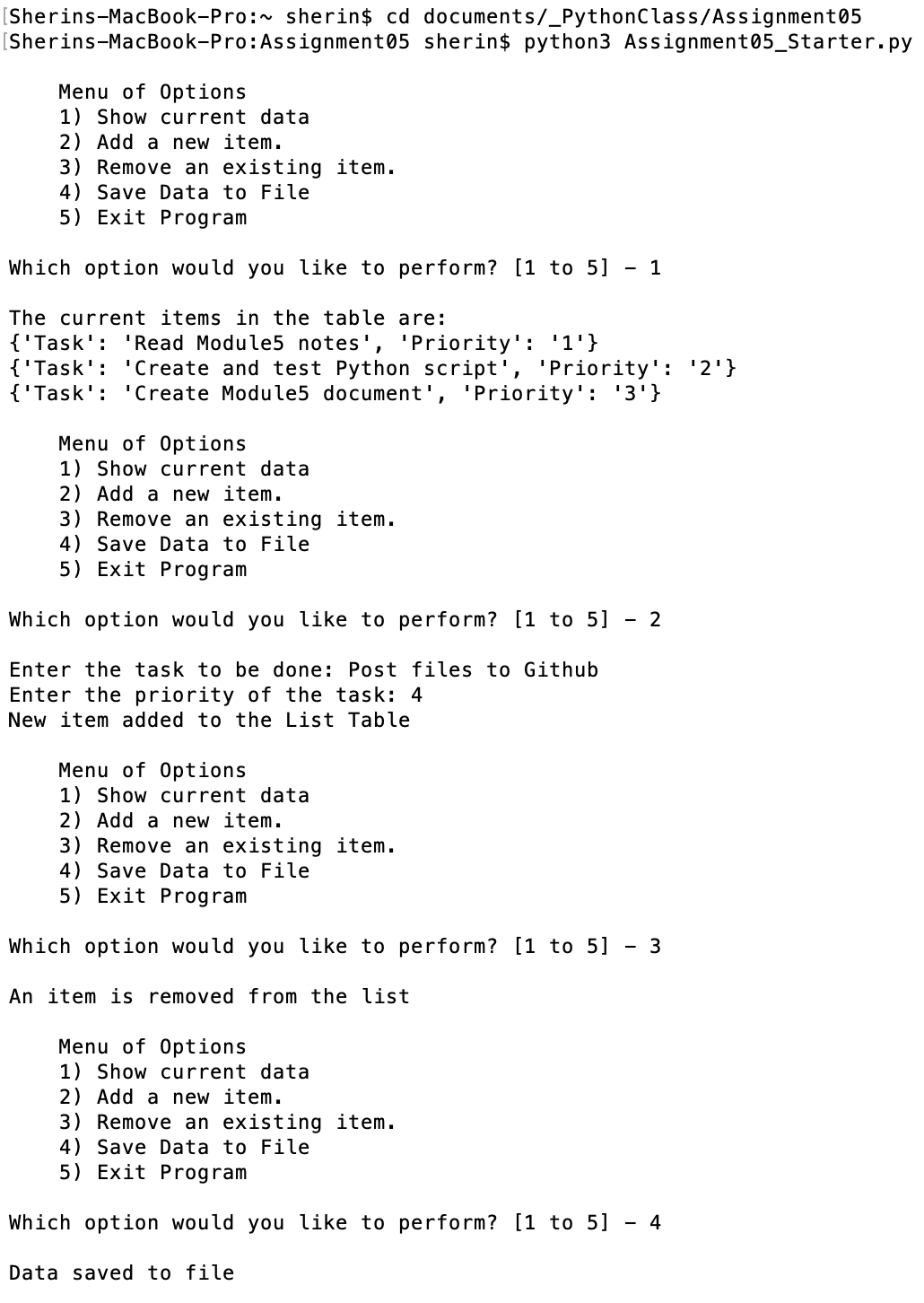
***Figure 2: Code in Python***

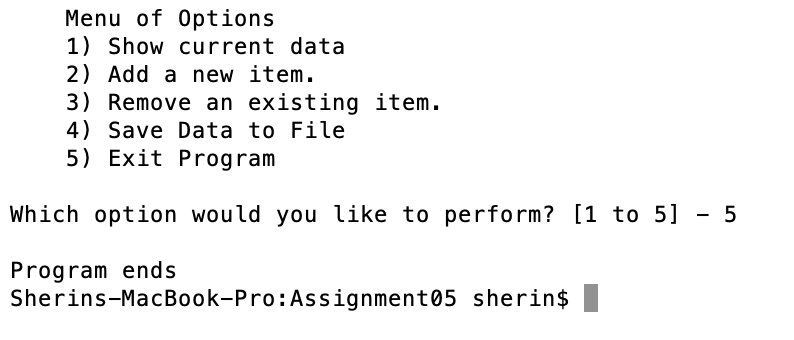
***Running the Script***

With the script created in its proper location, I run the script in both PyCharm (Figure 3) and an OS command/shell window (Figure 4).



***Figure 3: A screenshot of the script running in PyCharm.***

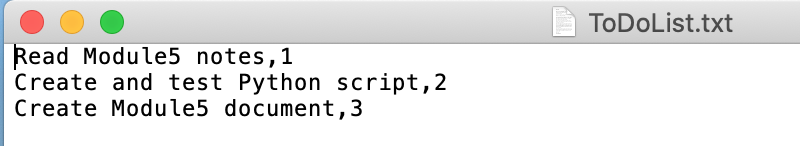
****

****

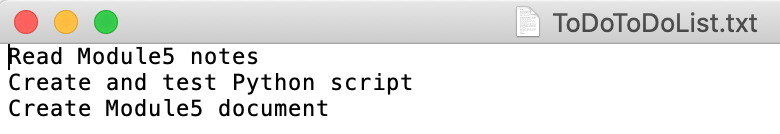
***Figure 4: A screenshot of the script running in Command Window.***

**Verifying the result**

Locate the text file and open it in a text editor. (Figure 8 and 9 )



***Figure 8: Verifying that the file has data***



***Figure 9: Verifying that the file has data***

**Summary**

In this assignment, I was able to write a python script using list of dictionaries and text files. The script is executed both in PyCharm and in command window to verify the results and the script ran as expected.