

Jason Johanneck
2022-May-23
IT FDN Programming 110
Module05 Assignment06

FDN Programming 110 Assignment 06

Github Repository Link: https://github.com/jjohanne514/IntroToProg_PythonAssignment06.git

Introduction

Assignment06 asks us to include the same Menu choices as Assignment 05 but wrap the IO and data processing code into functions that will be invoked when handling user selected menu choices.

Building the script.

The script uses a pre-built python starter file where we are to complete the TODO sections. Here is my code listing from NotePad++.

Figure 1 - Assignment 06 Code Listing

```
1 # ----- #
2 # Title: Assignment 06
3 # Description: Working with functions in a class,
4 # When the program starts, load each "row" of data
5 # in "ToDoList.txt" into a python Dictionary.
6 # Add the each dictionary "row" to a python list "table"
7 # ChangeLog (Who,When,What):
8 # RRoot,1.1.2030,Created started script
9 # Jason Johansen, 5/23/2022, Modified code to complete assignment 06
10 # ----- #
11
12 # Data ----- #
13 # Declare variables and constants
14 strFileName = "ToDoList.txt" # The name of the data file
15 file_obj = None # An object that represents a file
16 row_dic = {} # A row of data separated into elements of a dictionary (Task,Rank)
17 lstTable = [] # A list that acts as a 'table' of rows
18 choice_str = "" # Captures the user option selection
19
20
21 # Processing ----- #
22 class Processor:
23     """ Processes the data in a list of dictionaries to and from a text file """
24
25     @staticmethod
26     def read_file_to_list_of_dictionaries(file_name, list_of_dictionary_rows):
27         list_of_dictionary_rows.clear() # clear any old data before loading
28         file = open(file_name, "r") # Causes ERROR if file does not exist!
29         for line in file:
30             item, value = line.split(",")
31             row = {"Task": item.strip(), "Rank": value.strip()}
32             list_of_dictionary_rows.append(row)
33         file.close()
34         return list_of_dictionary_rows, 'success'
35
36     @staticmethod
37     def remove_data_from_list_of_dictionaries(list_of_dictionary_rows, task_to_remove):
38         for row in list_of_dictionary_rows:
39             if row["Task"].lower() == task_to_remove.lower():
40                 lstTable.remove(row)
41                 # print("row removed")
42                 return list_of_dictionary_rows, 'success'
43
44     @staticmethod
45     def add_data_to_list_of_dictionaries(list_of_dictionary_rows, item, value):
46         row = {"Task": str(item).strip(), "Rank": str(value).strip()}
47         list_of_dictionary_rows.append(row)
48         return list_of_dictionary_rows, 'success'
49
50     @staticmethod
51     def write_file_from_list_of_dictionaries(file_name, list_of_dictionary_rows):
52         file = open(file_name, "w")
53         for row in list_of_dictionary_rows:
54             file.write(row["Task"] + "," + row["Rank"] + "\n")
55         file.close()
56         return list_of_dictionary_rows, 'success'
57
58 # Presentation (Input/Output) ----- #
59 class IO:
60     @staticmethod
61     def print_menu():
62         print("""
63         Menu of Options
64         1) Load Data from File
65         2) Add a new Task
66         3) Remove an existing Task
67         4) Save Data to File
68         5) Exit Program
69         """)
70
71     @staticmethod
72     def input_menu_choice():
73         choice = str(input("Which option would you like to perform? [1 to 4] - ")).strip()
74         print() # Add an extra line for looks
75         return choice
76
77     @staticmethod
78     def print_current_list_items(list_of_rows):
79         print("***** The Current Items Are: *****")
80         for row in list_of_rows:
81             print(row["Task"] + " (" + row["Rank"] + ")")
82         print("*****")
83         print() # Add an extra line for looks
84
85     @staticmethod
86     def input_task_and_rank():
87         task = str(input("What is the Task? - ")).strip()
88         rank = str(input("What is the Rank? - ")).strip()
89         print() # Add an extra line for looks
90         return task, rank
91
92     @staticmethod
93     def input_task_to_remove():
94         task = str(input("Remove which item? - ")).strip()
95         print() # Add an extra line for looks
96         return task
97
98 # Main Body of the Script ----- #
99 while(True):
100     IO.print_current_list_items(lstTable)
101     IO.print_menu() # Test menu
102     strChoice = IO.input_menu_choice() # Test user choice
103
104     if (strChoice == "1"): # 1) Load Data from File
105         lstTable, status = Processor.read_file_to_list_of_dictionaries(strFileName, lstTable)
106         if status == 'success':
107             print('Done!')
108
109     elif (strChoice == "2"): # 2) Add a new item
110         strTask, strRank = IO.input_task_and_rank()
111         lstTable, status = Processor.add_data_to_list_of_dictionaries(lstTable, strTask, strRank)
112         if status == 'success':
113             print('Done!')
114
115     elif (strChoice == "3"): # 3) Remove an existing item
116         strTask = IO.input_task_to_remove()
117         lstTable, status = Processor.remove_data_from_list_of_dictionaries(lstTable, strTask)
118         if status == 'success':
119             print('Done!')
120
121     elif (strChoice == "4"): # 4) Save Data to File And Exit Program
122         lstTable, status = Processor.write_file_from_list_of_dictionaries(strFileName, lstTable)
123         if status == 'success':
124             print('Done!')
125     elif (strChoice == "5"):
126         print("Goodbye!")
127         break
```

The Python code was broken out into three sections:

Section one: A function class for Data Processing was defined that included four functions:

Figure 2- Processor Class with four functions

```
# Processing -----#
class Processor:
    """ Processes the data in a list of dictionaries to and from a text file """

    @staticmethod
    def read_file_to_list_of_dictionaries(file_name, list_of_dictionary_rows):
        list_of_dictionary_rows.clear() # clear any old data before loading
        file = open(file_name, "r") # Causes ERROR if file does not exist!
        for line in file:
            item, value = line.split(",")
            row = {"Task": item.strip(), "Rank": value.strip()}
            list_of_dictionary_rows.append(row)
        file.close()
        return list_of_dictionary_rows, 'success'

    @staticmethod
    def remove_data_from_list_of_dictionaries(list_of_dictionary_rows, task_to_remove):
        for row in list_of_dictionary_rows:
            if row["Task"].lower() == task_to_remove.lower():
                list_of_dictionary_rows.remove(row)
                # print("row removed")
        return list_of_dictionary_rows, 'success'

    @staticmethod
    def add_data_to_list_of_dictionaries(list_of_dictionary_rows, item, value):
        row = {"Task": str(item).strip(), "Rank": str(value).strip()}
        list_of_dictionary_rows.append(row)
        return list_of_dictionary_rows, 'success'

    @staticmethod
    def write_file_from_list_of_dictionaries(file_name, list_of_dictionary_rows):
        file = open(file_name, "w")
        for row in list_of_dictionary_rows:
            file.write(row["Task"] + "," + row["Rank"] + "\n")
        file.close()
        return list_of_dictionary_rows, 'success'

# Presentation (Input/Output) -----#
```

Section two: A function class for screen IO was created with five Input/Output functions:

Figure 3- IO Class with five functions defined

```
58 # Presentation (Input/Output) -----#
59 class IO:
60     @staticmethod
61     def print_menu():
62         print('''
63         Menu of Options
64         1) Load Data from File
65         2) Add a new Task
66         3) Remove an existing Task
67         4) Save Data to File
68         5) Exit Program
69         ''')
70
71     @staticmethod
72     def input_menu_choice():
73         choice = str(input("Which option would you like to perform? [1 to 4] - ")).strip()
74         print() # Add an extra line for looks
75         return choice
76
77     @staticmethod
78     def print_current_list_items(list_of_rows):
79         print("***** The Current Items Are: *****")
80         for row in list_of_rows:
81             print(row["Task"] + " (" + row["Rank"] + ")")
82         print("*****")
83         print() # Add an extra line for looks
84
85     @staticmethod
86     def input_task_and_rank():
87         task = str(input("What is the Task? - ")).strip()
88         rank = str(input("What is the Rank? - ")).strip()
89         print() # Add an extra line for looks
90         return task, rank
91
92     @staticmethod
93     def input_task_to_remove():
94         task = str(input("Remove which item? - ")).strip()
95         print() # Add an extra line for looks
96         return task
97
```

Section three: The main body of the script contains logic for processing menu selections:

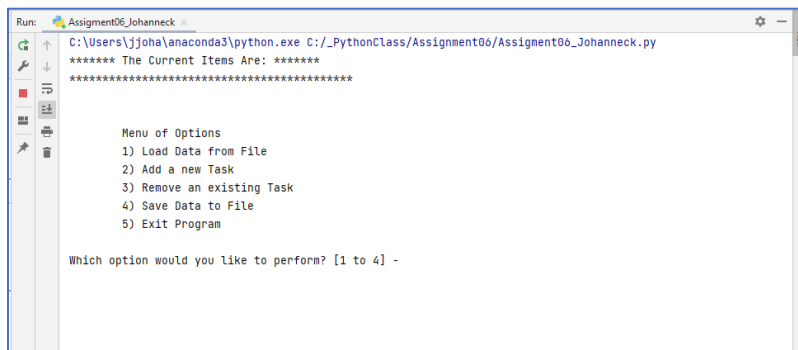
Figure 4 - Main Script section for Menu Processing Logic

```
99 # Main Body of the Script ----- #
100 while(True):
101     IO.print_current_list_items(lstTable)
102     IO.print_menu() # Test menu
103     strChoice = IO.input_menu_choice() # Test user choice
104
105     if (strChoice == "1"): # 1) Load Data from File
106         lstTable, status = Processor.read_file_to_list_of_dictionaries(strFileName, lstTable)
107         if status == 'success':
108             print('Done!')
109     elif (strChoice == "2"): # 2) Add a new item
110         strTask, strRank = IO.input_task_and_rank()
111         lstTable, status = Processor.add_data_to_list_of_dictionaries(lstTable, strTask, strRank)
112         if status == 'success':
113             print('Done!')
114     elif (strChoice == "3"): # 3) Remove an existing item
115         strTask = IO.input_task_to_remove()
116         lstTable, status = Processor.remove_data_from_list_of_dictionaries(lstTable, strTask)
117         if status == 'success':
118             print('Done!')
119     elif (strChoice == "4"): # 4) Save Data to File And Exit Program
120         lstTable, status = Processor.write_file_from_list_of_dictionaries(strFileName, lstTable)
121         if status == 'success':
122             print('Done!')
123     elif (strChoice == "5"):
124         print("Goodbye!")
125         break
126
```

Running the script.

From the PyCharm application:

Here is the Initial Display after pressing the run button:



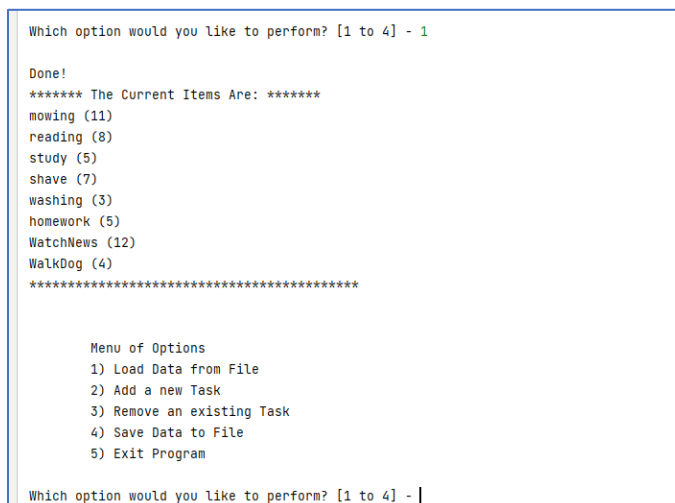
The screenshot shows the PyCharm Run window for the file 'Assignment06_Johanneck.py'. The output text is as follows:

```
Run: Assignment06_Johanneck.py
C:\Users\jjoha\anaconda3\python.exe C:/PythonClass/Assignment06/Assignment06_Johanneck.py
***** The Current Items Are: *****
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] -
```

Here is output after selecting the option to Load the ToDoList.txt data:



The screenshot shows the PyCharm Run window after selecting option 1. The output text is as follows:

```
Which option would you like to perform? [1 to 4] - 1

Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
shave (7)
washing (3)
homework (5)
WatchNews (12)
WalkDog (4)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - |
```

Here is the output after selecting the menu choice to remove a task:

```
*****
Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 3

Remove which item? - shave

Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
washing (3)
homework (5)
WatchNews (12)
WalkDog (4)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - |
```

Here I am selecting the menu option to add a new Task for Planting Trees:

```
*****
Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 2

What is the Task? - Plant Trees
What is the Rank? - 6

Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
washing (3)
homework (5)
WatchNews (12)
WalkDog (4)
Plant Trees (6)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - |
```

This is the output after selecting menu option #4 to Save data to disk:

```
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 4

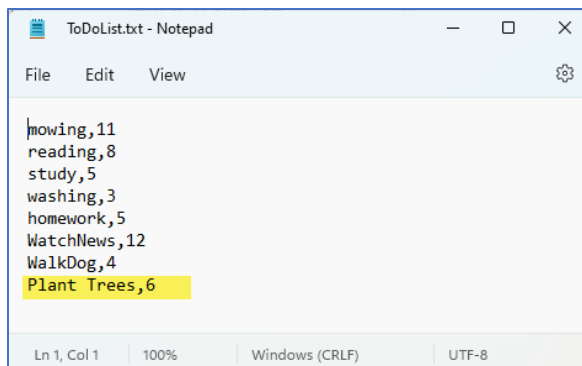
Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
washing (3)
homework (5)
WatchNews (12)
WalkDog (4)
Plant Trees (6)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - |
```

Here I am showing the contents of ToDoList.txt after saving to disk:

Figure 5- Contents of ToDoList.txt after Menu Option 4 selected



```
ToDoList.txt - Notepad
File Edit View
mowing,11
reading,8
study,5
washing,3
homework,5
WatchNews,12
WalkDog,4
Plant Trees,6
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Finally, here is PyCharm output after selecting menu option #5 to Exit:

```
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 5

Goodbye!

Process finished with exit code 0
|
```

Running the script from the Windows Command prompt:

Here I run the Assignment06_Johanneck.py and then selecting the menu option to load data from file:

```
Command Prompt - Python Assignment06_Johanneck.py

C:\_PythonClass\Assignment06>Python Assignment06_Johanneck.py
***** The Current Items Are: *****
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 1

Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
washing (3)
homework (5)
WatchNews (12)
WalkDog (4)
Plant Trees (6)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] -
```


The following screen shows the addition of a new task:

```
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 2

What is the Task? - GetREIitems
What is the Rank? - 18

Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
washing (3)
homework (5)
WatchNews (12)
WalkDog (4)
Plant Trees (6)
GetREIitems (18)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] -
```

The below screen print is from my choice to remove an item from the list:

```
*****
Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 3

Remove which item? - washing

Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
homework (5)
WatchNews (12)
WalkDog (4)
Plant Trees (6)
GetREIitems (18)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] -
```

In this next screen print, I chose to Save the current list table of dictionary rows to disk:

```
*****
Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] - 4

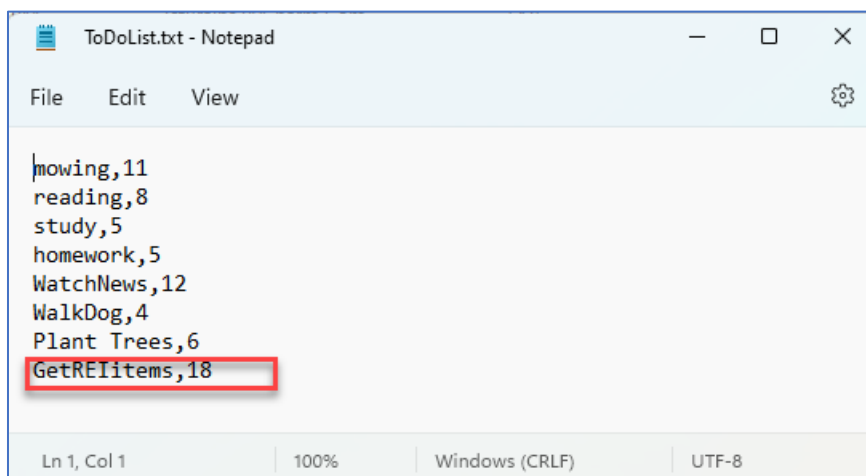
Done!
***** The Current Items Are: *****
mowing (11)
reading (8)
study (5)
homework (5)
WatchNews (12)
WalkDog (4)
Plant Trees (6)
GetREIitems (18)
*****

Menu of Options
1) Load Data from File
2) Add a new Task
3) Remove an existing Task
4) Save Data to File
5) Exit Program

Which option would you like to perform? [1 to 4] -
```

The contents of the ToDoList.txt file after removing the Washing Task and Saving the list of Tasks to Disk:

Figure 6 - contents of ToDoList.txt from Notepad



Finally, after selecting option #5 to Exit:

```
*****  
  
Menu of Options  
1) Load Data from File  
2) Add a new Task  
3) Remove an existing Task  
4) Save Data to File  
5) Exit Program  
  
Which option would you like to perform? [1 to 4] - 5  
Goodbye!  
C:\_PythonClass\Assignment06>
```

Summary

In this assignment we learned how to consolidate screen I/O and table processing into functions. We also learned how pass parameters into functions and evaluate function return arguments.