Saglara Orellana
May 14, 2022
Foundation of Programming: Python
Assignment 06
https://github.com/saglarao/IntroToProg-Python-Mod06

Creating To Do File

Introduction

The program starts with displaying the data and menu of choices, then user is able to add more data, remove a task, save all data to the file and reload it. It is similar to the previous assignment's script, however this time, I need to define functions first and then call them in the main body of the script.

Processing functions

There are 4 functions in processing block of the script: reading data from the file, adding data, removing data and saving it to the file. Reading data has already been defined in the script. I proceed with defining adding data to the list.

Function *add_data_to_list* has three parameters: *task*, *priority* and *list_of_rows*. First, I will need to create a dictionary *row*, that contains *task* and *priority*. I assigned key value "Task" to *task* and "Priority" to *priority*, using method *title()* to make it look pretty and have same format, no matter how the user entered it. Then I added it to the main list of dictionaries (*list_of_rows*). Lastly, I updated the return string value from 'Success' to 'New task has been added!'.

```
def add_data_to_list(task, priority, list_of_rows):
    row = {"Task": task.title(), "Priority": priority.title()}
    list_of_rows.append(row)
    return list_of_rows, 'New task is added!'
```

Next I am going to add code to define function *remove_data_from_list*. To remove the task I need to loop through the table of data (list of dictionaries), comparing each value under "Task" key with the function's parameter *task*. If they are equal, that dictionary row is removed from the list.

Next, I want my program to display a message when the task is removed or it was not found. I added a block of code, that is looking, if the parameter *task* is in table of data, then it displays a message "Task has been removed", if not, it notifies the user, that "Task was not found". It was added above the loop to remove the task, otherwise it will always displays "Task was not found", since the task will be always removed, if it is below:

```
def remove_data_from_list(task, list_of_rows):
    if any(task.lower() in row["Task"].lower() for row in list_of_rows):
        print("The task has been removed!")
    else:
        print("Task was not found")
    for row in list_of_rows:
        if row["Task"].lower() == task.lower():
            list_of_rows.remove(row)
    return list_of_rows, 'Success'
```

The last function in Processor class is *write_data_to_file* with two parameters *file_name* and *list_of_rows*. First I need to *open* the file object to *write* to save the data into a file. Then I need to unpack the list of dictionaries *list_of_rows* and write it into a file with comma as a separator and adding a new line at the end. I closed the file and then updated the return string value from 'Success' to 'Data was saved to the file!'

```
def write_data_to_file(file_name, list_of_rows):
    file = open(file_name, "w")
    for line in list_of_rows:
        file.write(str(line["Task"]) + ',' + str(line["Priority"]) + "\n")
    file.close()
    return list_of_rows, 'Data was saved to the file!'
```

Input/output functions

There are 7 functions in IO class, that display data (output) and capture user input. Functions to display menu (*print_menu_Tasks*), to collect user choice of menu option (*input_menu_choice*), to display current tasks on the list (*print_current_Tasks_in_list*), to capture yes/no answer from the user (*input_yes_no_choice*) and to pause the program before continue (*input_press_to_continue*) were already defined in the script.

I need to define the function to let the user to enter a new task and priority called *input_new_task_and_priority*. I assigned variables *task* and *priority* the value of input function:

```
def input_new_task_and_priority():
    task = input('Enter a new task: ')
    priority = input('Enter a priority: ')
    return task, priority
```

The last IO function I need to define is *input_task_to_remove* to get the user to enter the task to remove:

```
def input_task_to_remove():
    task = input('Enter a task to remove: ')
    return task
```

Main Body of the Script

Now that all functions are defined, I can call them in the main body of the script. First step is to load the data from the file ToDoFile.txt. The code was already written for this step, calling for function *read_data_from_file* from Processor class with arguments *strFileName* and *lstTable*. I added *try-except* construct to catch errors in case the file doesn't exist:

```
# Step 1 - When the program starts, Load data from ToDoFile.txt.
try:
    Processor.read_data_from_file(strFileName, lstTable)  # read file data
except:
    print('No file is found. New file will be created.')
```

Next, the script starts the while loop with showing current data and menu, then asks the user to choose a menu option. This part of the script was also already written and I kept it without changes.

Once user enters a menu option, the program starts processing it. If the user entered '1', I need to call a function to add a new task. Before I can add a new task, I need to collect the data from the user. So, first, I call for IO class function *input_new_task_and_priority*. This function returns two values *task* and *priority,* so I assigned variables *strTask* and *strPriority* to the values of this function. Now that I have the data from the user, I can add it to the list by calling Processor class function *add_data_to_list*, this function also returns two values *list_of_rows* and a string value of process completion. I assigned variable *lstTable* and *strStatus* to function *add_data_to_list* and passed it values of *strTask*, *strPriority* and *lstTable*. Then this block of code ends with *IO.input_press_to_continue* with *strStatus* as its argument.

```
if strChoice.strip() == '1':  # Add a new Task
    strTask, strPriority = IO.input_new_task_and_priority()
    lstTable, strStatus = Processor.add_data_to_list(strTask, strPriority,
lstTable)
    IO.input_press_to_continue(strStatus)
    continue  # to show the menu
```

When user chooses option 2 of the menu the program will ask the user to enter a task and then removes it from the list, if it exists. First, I assign *strRemove* to the value of function *IO.input_task_to_remove* to have the user enter what task they want to remove. Then I decided to ask the user to confirm, like almost all programs do before deletion, that they want that task to be removed, by assigning *strChoice* to *IO.input_yes_no_choice*. If the user entered 'y', strChoice has value of 'y', then lstTable and strStatus are assigned values of function *Processor.remove_data_from_list* with arguments *strRemove* and *lstTable.* Next program will pause for the user to press enter with function input_press_to_continue. I have removed the optional *strStatus*, as status message is displayed as part of the function *remove_data_from_list*. If the user entered something else from 'y', the program pauses and shows the message that the task has remained on the list.

```
elif strChoice == '2':  # Remove an existing Task
    strRemove = IO.input_task_to_remove()
    strChoice = IO.input_yes_no_choice("Proceed to remove " +
strRemove.title() + " task from the list (y/n)?: ")
    if strChoice.lower() == "y":
        lstTable, strStatus = Processor.remove_data_from_list(strRemove,
lstTable)
```

```
        IO.input_press_to_continue()   # status is displayed in the function
above
    else:
        IO.input_press_to_continue(strRemove.title() + " remained on the
list.")
    continue  # to show the menu
```

Option 3 of the menu lets the user to save the data back to the file ToDoFile.txt. Again, as all good programs do, it asks the user to confirm, if they want to save the data to file. The variable *strChoice* is assigned to the value of IO class function *input_yes_no_choice*. If strChoise is 'y', then lstTable and strStatus are assigned to the values of processing function *write_data_to_file* with arguments *strFileName* and *lstTable*. Then program pauses for the user to press "Enter" and shows the action status *strStatus*. If the value of strChoise is something else, "Save Cancelled!" is displayed with pause for the user to press Enter.

```
elif strChoice == '3':  # Save Data to File
    strChoice = IO.input_yes_no_choice("Save this data to file? (y/n) - ")
    if strChoice.lower() == "y":
        lstTable, strStatus = Processor.write_data_to_file(strFileName,
lstTable)
        IO.input_press_to_continue(strStatus)
    else:
        IO.input_press_to_continue("Save Cancelled!")
    continue  # to show the menu
```

If the user chose option 4 of the menu, the program will reload the data from the file again, like at the beginning of the script. Warning message is displayed first, following by yes/no input function. If the user proceeds with yes and enters 'y', then the program needs to read the data from the file. I incorporated *try-except* construct again in case there is no file created yet:

```
elif strChoice == '4':  # Reload Data from File
    print("Warning: Unsaved Data Will Be Lost!")
    strChoice = IO.input_yes_no_choice("Are you sure you want to reload data
from file? (y/n) -  ")
    if strChoice.lower() == 'y':
        try:
```

```
            lstTable, strStatus = Processor.read_data_from_file(strFileName,
lstTable)
            IO.input_press_to_continue(strStatus)
        except:
            print("No file was found. Please save the data first.")
    else:
        IO.input_press_to_continue("File Reload  Cancelled!")
    continue  # to show the menu
```

Lastly, the program ends, when the user chooses option 5 of the menu. I don't need to add any code here, however at the very end I added the else clause to catch any erroneous entries for menu choices:

```
elif strChoice == '5':  # Exit Program
    print("Goodbye!")
    break  # and Exit
else:
    print('Please only choose 1 - 5!')
```

Testing the Script

Figure 01 shows, how the program starts with displaying current list of tasks, that it uploaded from the ToDoFile.txt, that was created during prior tests of the script. The very first run gave me an error, because I did not have a file with that name, which prompted to use try-except in my code. Then it displays menu of options.

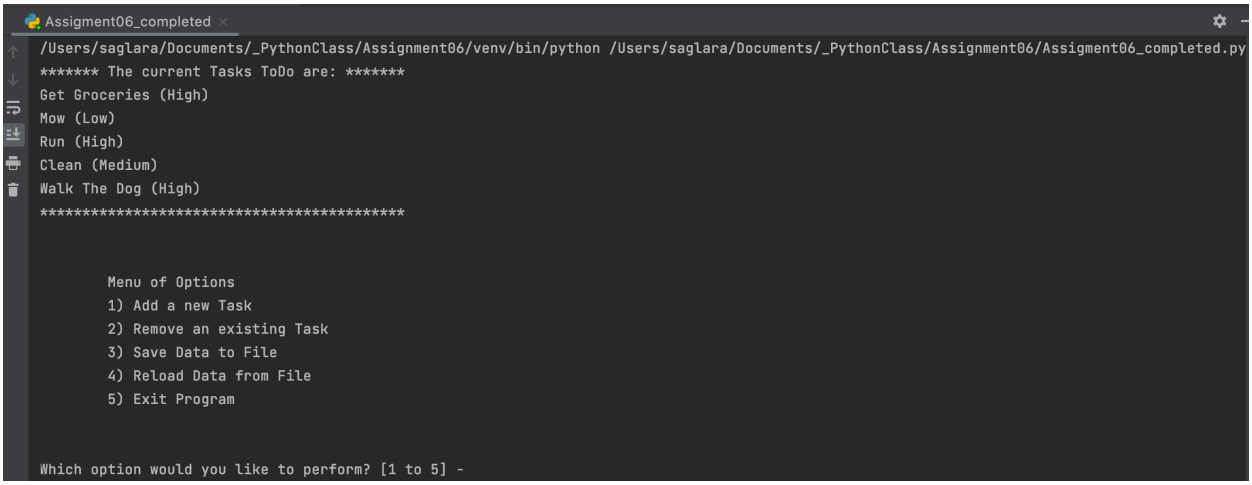

Figure 01. Starting the program.

I choose option 1 to add a new task “Homework” to the list. The program asked to enter a task and its priority and then let me know, that new task has been added. After pressing Enter key to continue, I can see, that new task “Homework” has been added to the list.

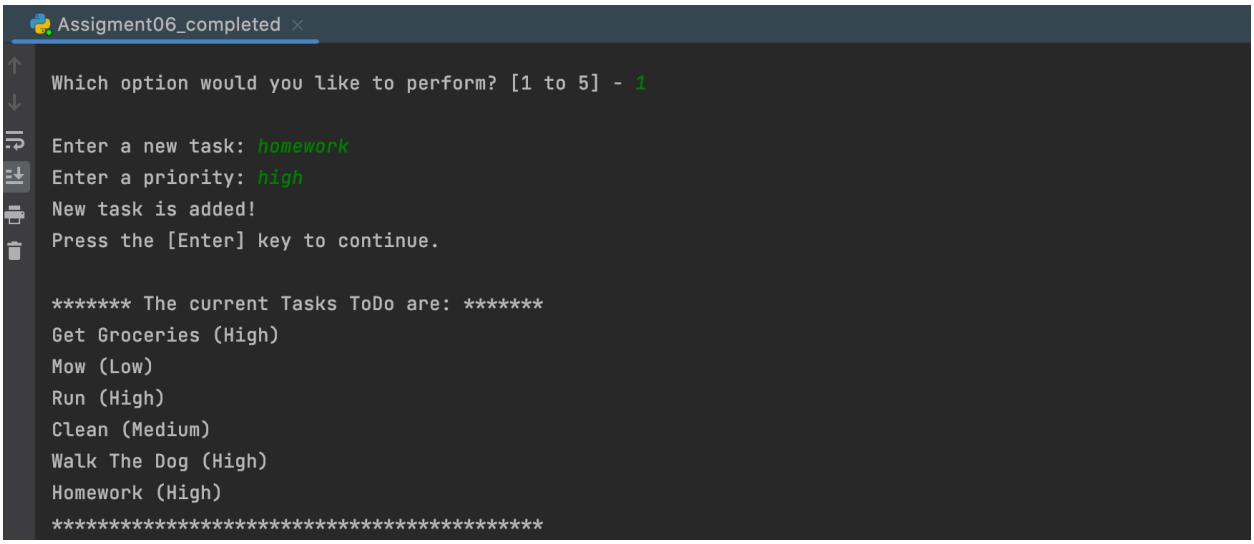

Figure 02. Adding a new task.

Next I am trying to remove a task from the list and I chose to remove “Run”, because I hate running and it shouldn’t have a high priority. First, the program asks me to confirm if I want to remove Run task from the list, after I confirm by entering ‘y’, I get the message that the task has been removed. After pressing Enter key to continue and looking at the current list, I can no longer see “Run” in there.

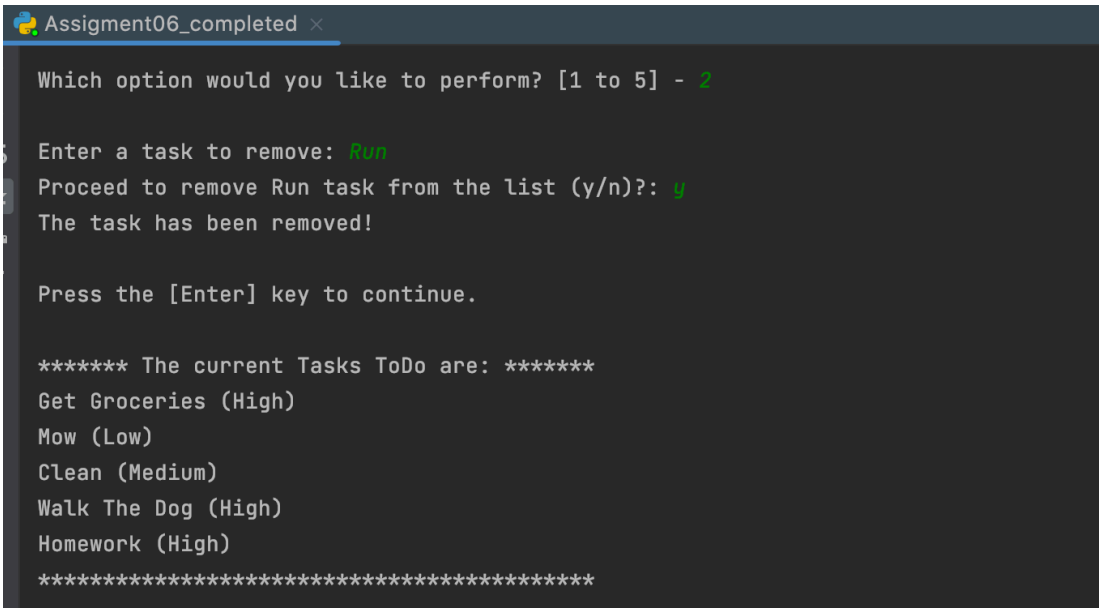

Figure 03. Removing task from the list

I am ready to save my updated list to the file, when I choose option 3 of the menu, the program asks, if I want to save it, and after I enter 'y' as my confirmation, I get the message that the data was saved.

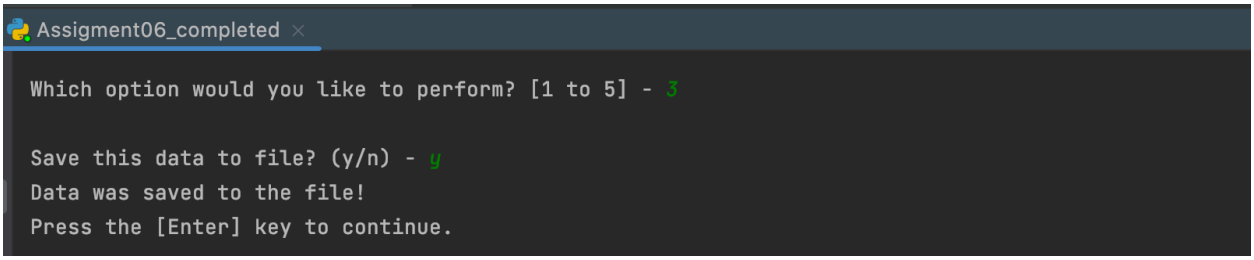

Figure 04. Saving data to file.

Figure 05 shows my updated list was saved into a file ToDoFile.txt.

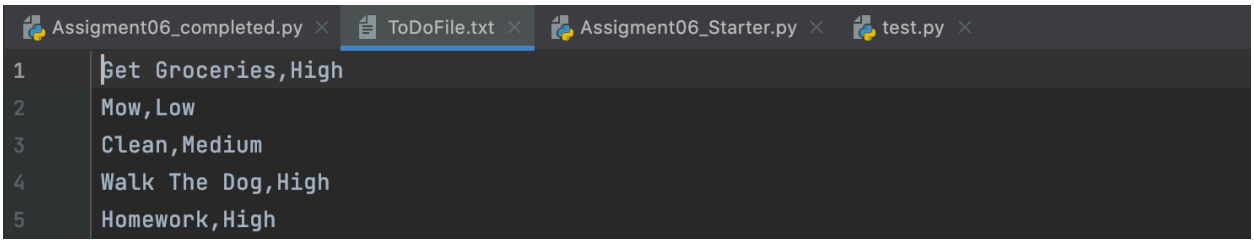

Figure 05. ToDoFile.txt.

To test if reloading data from the file works fine, I am going to add a new row of data in ToDoFile.txt, containing new task Read with medium priority. Then I choose option 4 from the menu, the program gives a warning and asks me to confirm. Once I enter 'y' to confirm, I receive 'Success' message and after pressing Enter key, I see the new data displayed with the task Read I added to the text file.

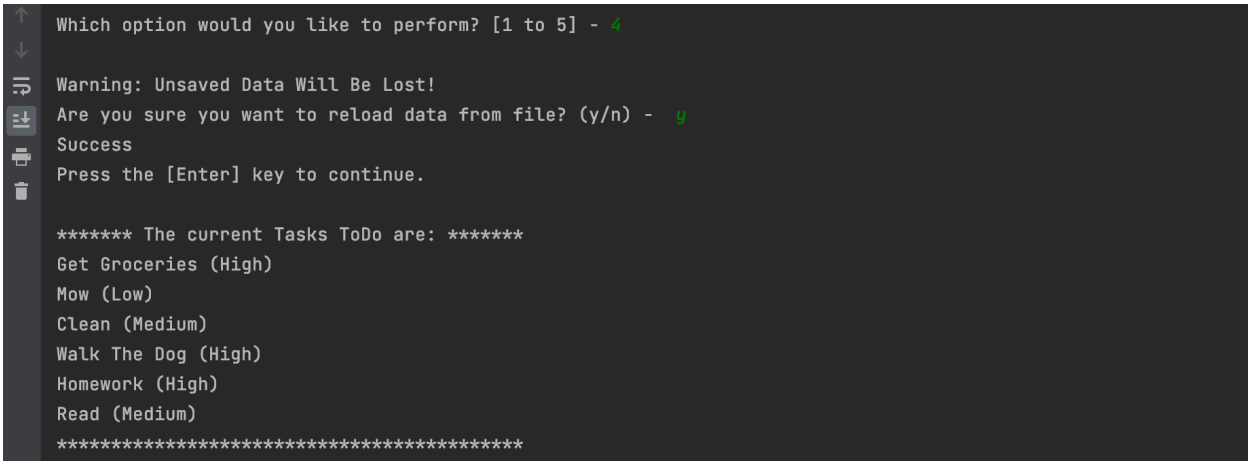

Figure 06. Reloading data from the file.

My next menu choice is 5 to exit the program. I receive “Goodbye!” message and program ends.

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

Goodbye!

Process finished with exit code 0
```

Figure 07. Exiting the program.

When I run the script from Command shell, it cannot locate the file, since it is looking in the main user folder and I did not specify the location in my script. I am able to add new tasks, that I can save into a ToDoFile.txt in the user folder on my computer.

```
Saglaras-MBP:~ saglara$ python3 /Users/saglara/Documents/_PythonClass/Assignment06/Assigment06_completed.py
No file is found. New file will be created.

******* The current Tasks ToDo are: *******
*******************************************


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


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

Enter a new task: Mow
Enter a priority: Low
New task is added!
Press the [Enter] key to continue.

******* The current Tasks ToDo are: *******
Mow (Low)
*******************************************


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


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

Enter a new task: Clean
Enter a priority: High
New task is added!
Press the [Enter] key to continue.

******* The current Tasks ToDo are: *******
Mow (Low)
Clean (High)
*******************************************
```

Figure 08. Running the script in Terminal.

I was able to perform all menu options in Terminal window and create a new ToDoFile.txt.

Summary

This script performs almost same as last week's assignment, allowing user to enter the data, remove it, save it into a file, load it from the file. Except this time, I got to define functions and then write the main body of the script by calling them.