Jonathan Kung

11/23/2022

IT FDN 110A

Assignment 6

https://github.com/jonkung/IntroToProg-Python-Mod06

Functions/Classes

Introduction

This week, the main topics were custom functions. Creating multiple custom functions, we can create a to do list similar to the previous assignment. There are two classes, one for inputs/outputs and the other for processing the data into the working list. Then there is the main script which uses the functions to run.

Assignment

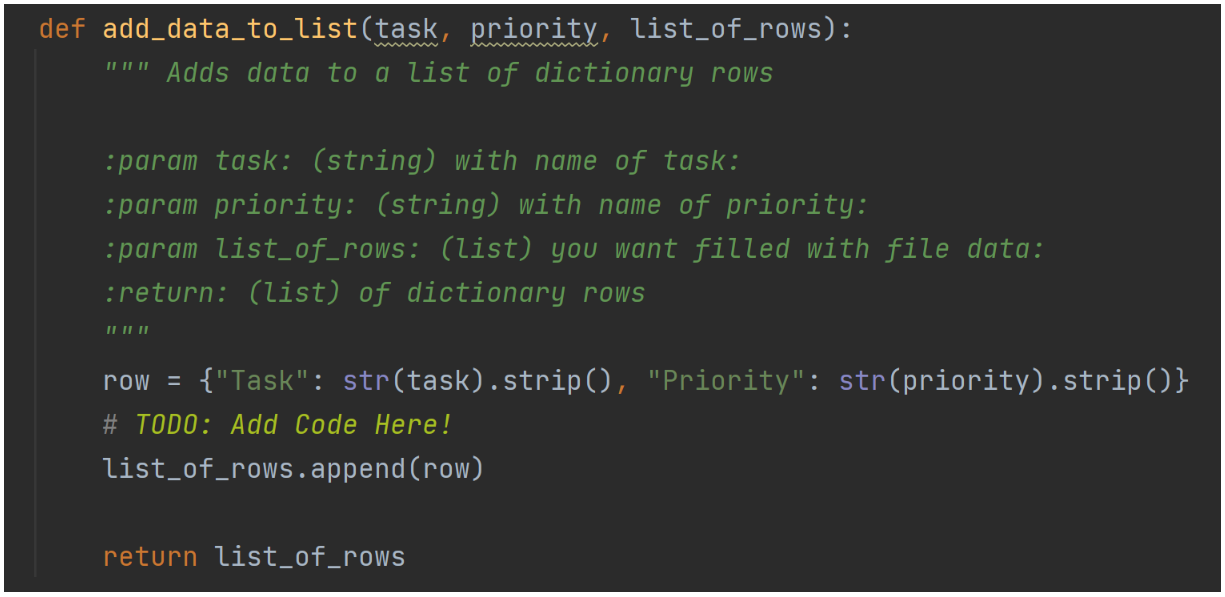
In this assignment, I needed to create a program that creates a to do lists. But in this assignment, the program will need to use multiple custom functions to store the data and then write the data into a txt file in a list object.

Processor

The processor class contains functions that processes the data and edit them into the list. The processor function can read data, add data, remove data, and write the data into a file.

Add data to list

Adding data to list function has three inputs: task, priority and the main list. The task and priority are entered by the user, and the list of rows is the working list that contains all the data. The goal of this function is to add new data into the working list. This is done using the .append function to add to the end of the list.

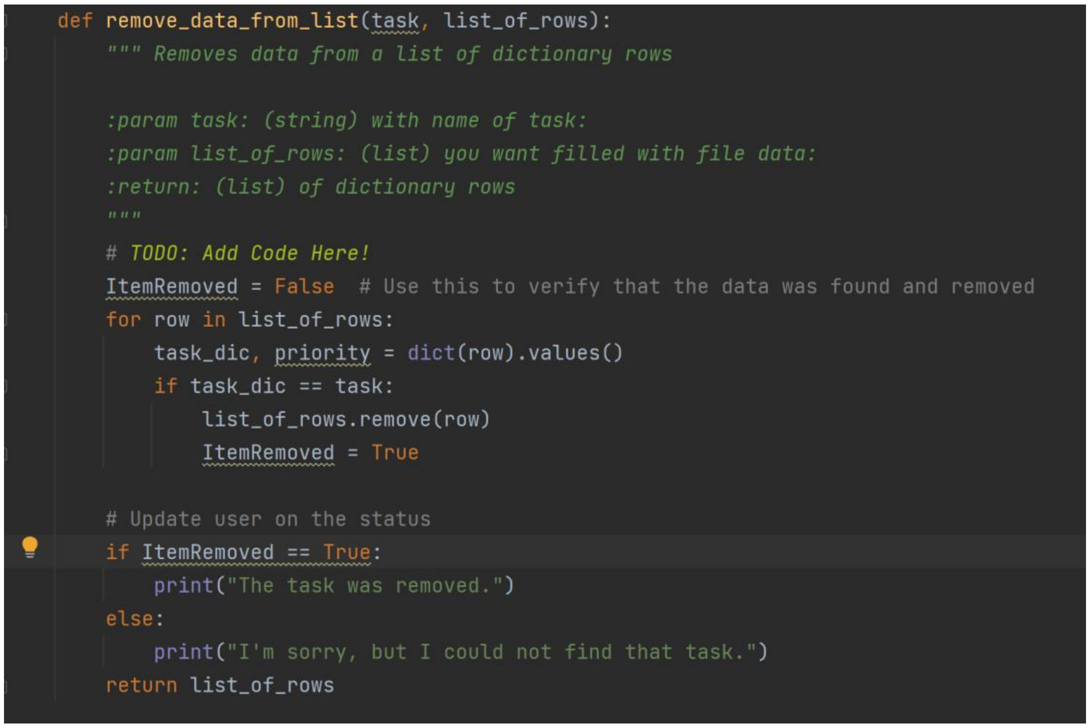


```python
def add_data_to_list(task, priority, list_of_rows):
    """ Adds data to a list of dictionary rows

    :param task: (string) with name of task:
    :param priority: (string) with name of priority:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    row = {"Task": str(task).strip(), "Priority": str(priority).strip()}
    # TODO: Add Code Here!
    list_of_rows.append(row)

    return list_of_rows
```

Figure 1: adding task using .append

Remove data to list

This function has two inputs: task and the main working list. To remove data, I used a for loop to go into every row of the working list to see if the task entered by the user equals to one of the task in the working list. If true a .remove function is used to remove that row. If the task entered does not match any task in the list, it will return an error and will not remove any rows.

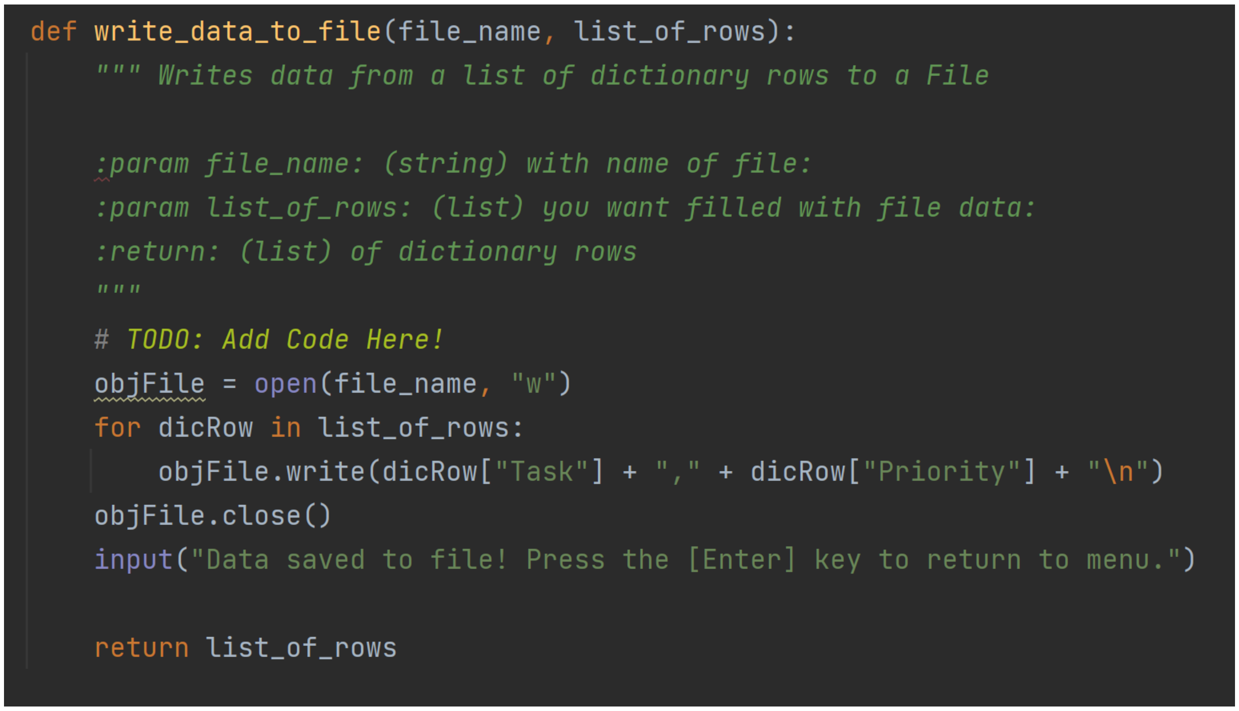


```python
def remove_data_from_list(task, list_of_rows):
    """ Removes data from a list of dictionary rows

    :param task: (string) with name of task:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    # TODO: Add Code Here!
    ItemRemoved = False  # Use this to verify that the data was found and removed
    for row in list_of_rows:
        task_dic, priority = dict(row).values()
        if task_dic == task:
            list_of_rows.remove(row)
            ItemRemoved = True

    # Update user on the status
    if ItemRemoved == True:
        print("The task was removed.")
    else:
        print("I'm sorry, but I could not find that task.")
    return list_of_rows
```

Figure 2: removing data from list

Writing data to file

This function has two inputs: the file and the working list. The purpose of this function is to write the list into a txt file. I used a for loop and for every row in the list, I used .write function to write data into the file.



```python
def write_data_to_file(file_name, list_of_rows):
    """ Writes data from a list of dictionary rows to a File

    :param file_name: (string) with name of file:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    # TODO: Add Code Here!
    objFile = open(file_name, "w")
    for dicRow in list_of_rows:
        objFile.write(dicRow["Task"] + "," + dicRow["Priority"] + "\n")
    objFile.close()
    input("Data saved to file! Press the [Enter] key to return to menu.")

    return list_of_rows
```

Figure 3: writing data to file

Input/Output (IO)

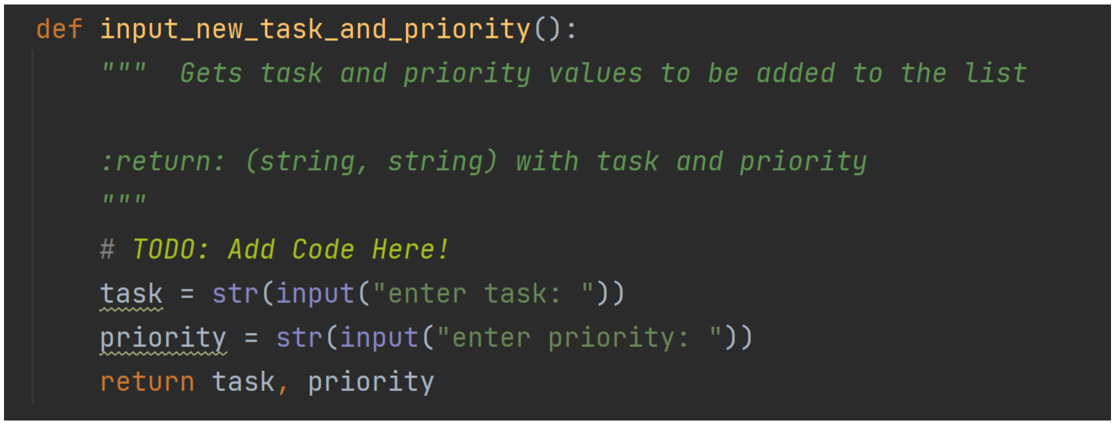
The IO class (input/output) performs input and outputs tasks that asks users what they want to do and gathers the data. There are multiple functions in this class: output menu, input menu choice from user, output current list, input new tasks, and input tasks to remove.

Input choice

This function asks the user what they want to do and it uses an input function. This function then returns the choice the user inputted.

Input new task

This function asks users to input a new task and priority. This is done using an input function. The function then returns the task and priority value.

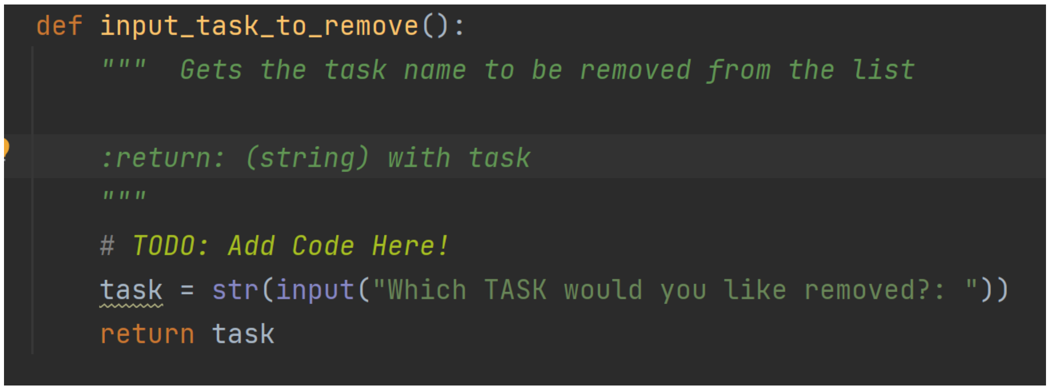


```
def input_new_task_and_priority():
    """  Gets task and priority values to be added to the list

    :return: (string, string) with task and priority
    """
    # TODO: Add Code Here!
    task = str(input("enter task: "))
    priority = str(input("enter priority: "))
    return task, priority
```

Figure 4: asking users to input new task

Input task to remove

This function asks the user what task they want to remove and this function is similar to the input new task function as it uses an input function and then returns that value.

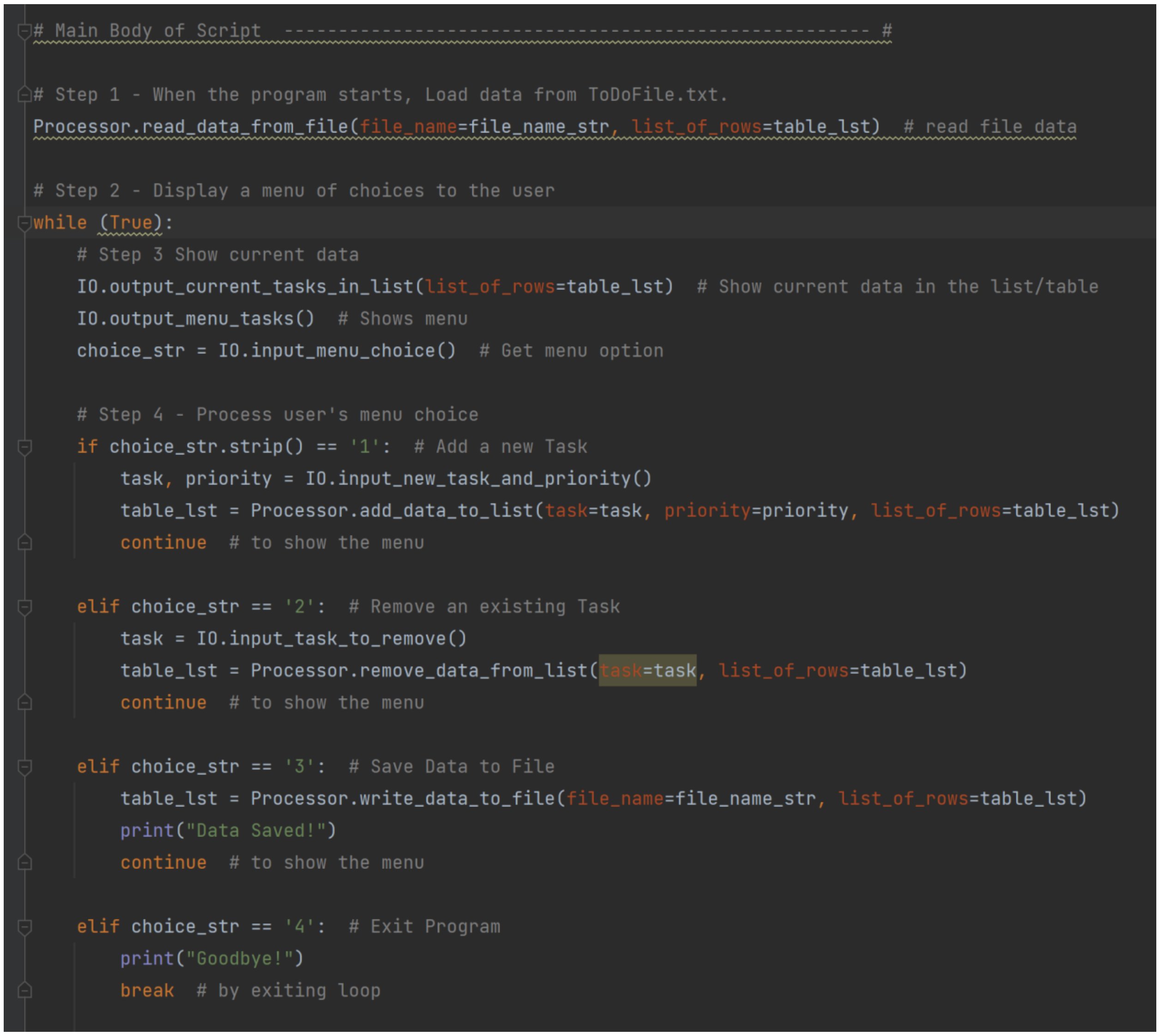


```
def input_task_to_remove():
    """  Gets the task name to be removed from the list

    :return: (string) with task
    """
    # TODO: Add Code Here!
    task = str(input("Which TASK would you like removed?: "))
    return task
```

Figure 5: asking user what task they want to remove

Main Body

In the main body, the code first reads the txt file and see if any data is in it. Next the code asks the user to input what they want to do from the menu choices. The code uses if/then statements to perform the task based on what the user inputs. For example, if the user inputs “1” (add new task), then the code will go to the IO input new task function which will return the string of the new task and priority. Then going to the processor adding new data to list function, the returned value of the new task and priority is inputted into this function and added to the end of the working list.

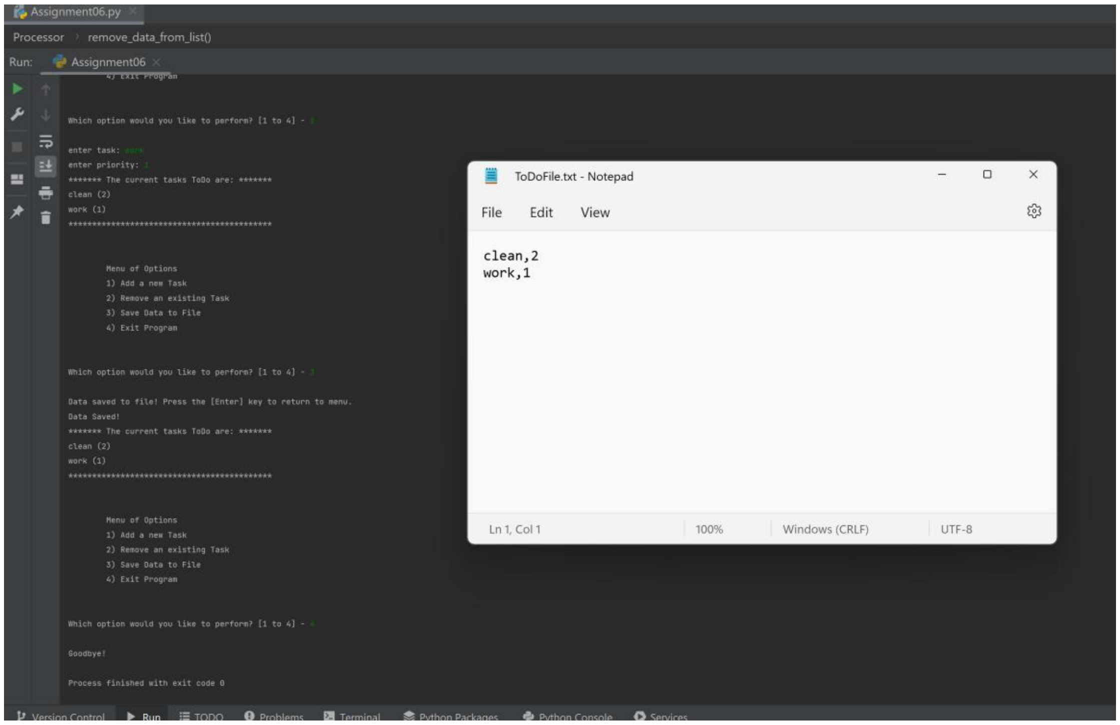


```
# Main Body of Script  ------------------------------------------------------ #

# Step 1 - When the program starts, Load data from ToDoFile.txt.
Processor.read_data_from_file(file_name=file_name_str, list_of_rows=table_lst)  # read file data

# Step 2 - Display a menu of choices to the user
while (True):
    # Step 3 Show current data
    IO.output_current_tasks_in_list(list_of_rows=table_lst)  # Show current data in the list/table
    IO.output_menu_tasks()  # Shows menu
    choice_str = IO.input_menu_choice()  # Get menu option

    # Step 4 - Process user's menu choice
    if choice_str.strip() == '1':  # Add a new Task
        task, priority = IO.input_new_task_and_priority()
        table_lst = Processor.add_data_to_list(task=task, priority=priority, list_of_rows=table_lst)
        continue  # to show the menu

    elif choice_str == '2':  # Remove an existing Task
        task = IO.input_task_to_remove()
        table_lst = Processor.remove_data_from_list(task=task, list_of_rows=table_lst)
        continue  # to show the menu

    elif choice_str == '3':  # Save Data to File
        table_lst = Processor.write_data_to_file(file_name=file_name_str, list_of_rows=table_lst)
        print("Data Saved!")
        continue  # to show the menu

    elif choice_str == '4':  # Exit Program
        print("Goodbye!")
        break  # by exiting loop
```

Figure 6: main body of script

Completed Script

The finished code and its text file is shown in figure 7.



```
        4) Exit Program


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

enter task: work
enter priority: 1
******* The current tasks ToDo are: *******
clean (2)
work (1)
*******************************************


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


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

Data saved to file! Press the [Enter] key to return to menu.
Data Saved!
******* The current tasks ToDo are: *******
clean (2)
work (1)
*******************************************


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


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

Goodbye!

Process finished with exit code 0
```

```
clean,2
work,1
```

Figure 7

Summary

In this assignment, I got to use multiple functions we learned from past weeks. Some functions used in the program are while loops, for loop, and if loops. This assignment uses list as the primary way to store the user inputs. By using lists, the user can easily input data and easily view the data. In addition, the user can save the data into a txt file similar to the previous assignment.