DUW
May 24, 2022
Foundations of Programming, Python
Assignment 06
GitHub:

Working with Functions

Introduction

In this assignment I will explain step-by-step how I added code to an existing Python program to manage a to do list, allow the user to see the existing data, add tasks based on user input, remove data based on user's request, and then store the data in a text file called, ToDoList.txt. Assignment 06 is meant to prove understanding of functions. The last summary section includes my observations and key takeaways from applying Python module 6.

Creating the Program

The steps I followed to develop and run the program:

1. **Created a new script file with the .py extension**
    a. Opened PyCharm and created a new sub-folder called Assignment06 inside of the _PythonClass folder.
    b. Create a new project in PyCharm that uses the _PythonClass\Assignment06 folder as its location.
    c. Created a python script file within my project called Assignment06.py and saved with the Python file extension.
2. **Updated the script header including inline comments in the script file**
    a. Updated the change log (who, when, what).
3. **Added the code in the script file**
    Added the below statements as instructions for Python to execute. The code implements the separation of concerns approach and includes three main sections: processing data, input/output, and main body of the script.
    a. Step 1: Built the functions in the class "Processor"
        i. The function to read data from a text file was already defined.
        ii. Built the function to add data to the list/table of data.
            - Formatted the row of data as a dictionary
            - Appended the new row of data to the existing list/table
            - Returned the list/table.
        iii. Built the function to remove data from the list.
            - Added a for loop that runs through column "Task" in the table to identify the task the user wants removed.
            - Added code to remove the task.
            - Added a print statement to indicate to the user that the task has been removed.
            - Added another print statement to print the current list/table.
            - Returned the list/table.
        iv. Saving/writing the data to a text file when the user makes that choice
            - Requested that the ToDoList.txt file open and start writing to the file.

- Added a for loop that runs through columns “Task” and “Priority’.
- Closed writing to the text file.
- Returned the list/table.

Figure 1 includes my added code for this step.

```
@staticmethod
def add_data_to_list(task, priority, list_of_rows):
    # TODO: Added Code Here!
    row={"Task": str(task).strip(), "Priority": str(priority).strip()}
    list_of_rows.append(row) #({'Task': task, 'Priority': priority})
    return list_of_rows, 'Success'

@staticmethod
def remove_data_from_list(task, list_of_rows):
    # TODO: Added Code Here!
    for row in list_of_rows:
        if row['Task'].lower() == task.lower():
            list_of_rows.remove(row)
        print('Task removed!')
        print(list_of_rows)
    return list_of_rows, 'Success'

@staticmethod
def write_data_to_file(file_name, list_of_rows):
    # TODO: Added Code Here!
    objFile = open('ToDoFile.txt', 'w')
    for row in list_of_rows:
        objFile.write(str(row['Task']) + ',' + str(row['Priority']) + '\n')
    objFile.close()
    return list_of_rows, 'Success'
```

Figure 1: Functions in the class Processor

b. Step 2: Built the functions in the class Input/Output
   i. The menu of choices, the input request for options, and the function to print current tasks in the list were already defined from pre-existing code. The while loop for the menu of choices is meant to repeat the print statement (menu) until the user tells it to break (the last choice).
   ii. Built the function to request input from user to add a new task and priority:

- Added two input statements to collect information from the user
- Returned the current list of tasks and priorities

iii. Built the function to request input from user to remove a task:
- Added an input statements to ask the user what task they would like removed from the list.
- Returned the current list of tasks and priorities.

Figure 2 includes my added code for this step.

```
@staticmethod
def input_new_task_and_priority():
    pass  # TODO: Added Code Here!
    strTask = str(input('Enter a new task:'))
    strPriority = str(input('Assign a priority:'))
    return strTask, strPriority
    # return task, priority

@staticmethod
def input_task_to_remove():
    pass  # TODO: Added Code Here!
    task = str(input("Item to Remove: ")).strip()
    print()
    return task
```

Figure 2: Functions in the class Input/Output

c. Step 3: Built conditionals for the five different user input options and used the functions defined above in the main body of the script:
   i. Conditional to add a new task when the user makes that choice and to print the current list/table
   ii. Conditional to remove an existing task
   iii. Conditional to save data to file
   iv. Conditional to reload data from file
   v. Conditional to exit the program

Figure 3 includes my added code for this step.

```
# Step 4 - Process user's menu choice
    if strChoice.strip() == '1':  # Add a new Task
        # TODO: Added Code Here
        strTask, strPriority=IO.input_new_task_and_priority()
        TaskList, status = Processor.add_data_to_list(strTask, strPriority, TaskList)
        print("Status was ", status)
        IO.print_current_Tasks_in_list(TaskList)
        IO.input_press_to_continue(strStatus)
        continue  # to show the menu

    elif strChoice == '2':  # Remove an existing Task
        # TODO: Added Code Here
        task=IO.input_task_to_remove()
        Processor.remove_data_from_list(task, TaskList)
        IO.input_press_to_continue(strStatus)
        continue  # to show the menu

    elif strChoice == '3':  # Save Data to File
        strChoice = IO.input_yes_no_choice("Save this data to file? (y/n) - ")
        if strChoice.lower() == "y":
            # TODO: Added Code Here!
            Processor.write_data_to_file(objFile, TaskList)
            IO.print_current_Tasks_in_list(strStatus)
        else:
            IO.input_press_to_continue("Save Cancelled!")
        continue  # to show the menu

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':
        # TODO: Added Code Here!
        IO.print_current_Tasks_in_list
    else:
        IO.input_press_to_continue("File Reload  Cancelled!")
    continue  # to show the menu

elif strChoice == '5':  # Exit Program
    print("Goodbye!")
    break  # and Exit
```

Figure 3: Functions in the class Main Body of the Script

4. **Ran the program through CMD**
   a. Opened CMD and added the path file for “Assignment05”
   b. ***Figure 4*** shows the output in CMD

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

Enter a new task:Run
Assign a priority:1
Status was  Success
******* The current Tasks ToDo are: *******
Walk (2)
Run (1)
*****************************************


Press the [Enter] key to continue.
******* The current Tasks ToDo are: *******
Walk (2)
Run (1)
*****************************************
```

```
Which option would you like to perform? [1 to 5] - 3

Save this data to file? (y/n) - y
******* The current Tasks ToDo are: *******
*****************************************

******* The current Tasks ToDo are: *******
Walk (2)
Run (1)
*****************************************
```

```
Which option would you like to perform? [1 to 5] - 2

Item to Remove: Walk

Task removed!
[{'Task': 'Run', 'Priority': '1'}]

Press the [Enter] key to continue.
******* The current Tasks ToDo are: *******
Run (1)
*****************************************
```

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

Warning: Unsaved Data Will Be Lost!
Are you sure you want to reload data from file? (y/n) -  y
******* The current Tasks ToDo are: *******
Run (1)
*****************************************
```

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

Goodbye!
```

Figure 4: Output in the CMD

5. **Step 7: Opened the file in a text editor**

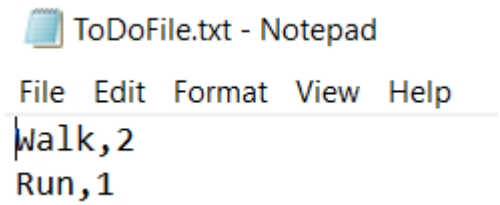

Figure 5: Output in text editor – Verifying that the file has data

   a. Located the text file and opened it to verify that it displays the user input
   b. ***Figure 5*** shows the output in the text editor

Summary

Creating and running Assignment 06 allowed me to build on the previous assignment and work with functions, classes, dictionaries, and lists. As more concepts are being introduced specifically with regards to data collections, I feel that I need more practice to better understand what code and syntax needs to be used when.