Name: Charles Shin

Date: 5/17/2021

Course: IT FDN 110 A Sp21: Foundations of Programming: Python

Assignment06

GitHub URL:

# Assignment 06: ToDoList using Functions

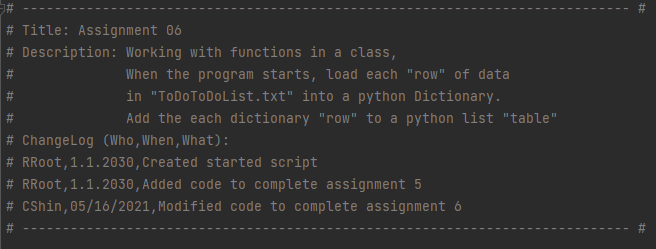
## Introduction

This assignment is to familiarize with adding/removing data using Function. Source code capture images can be found in the Appendix section, which is at the end of this document.

The structure and the function of the script is almost identical to what was constructed for the Assignment 05, except the Assignment 06 used the *functions* to construct and run the script.

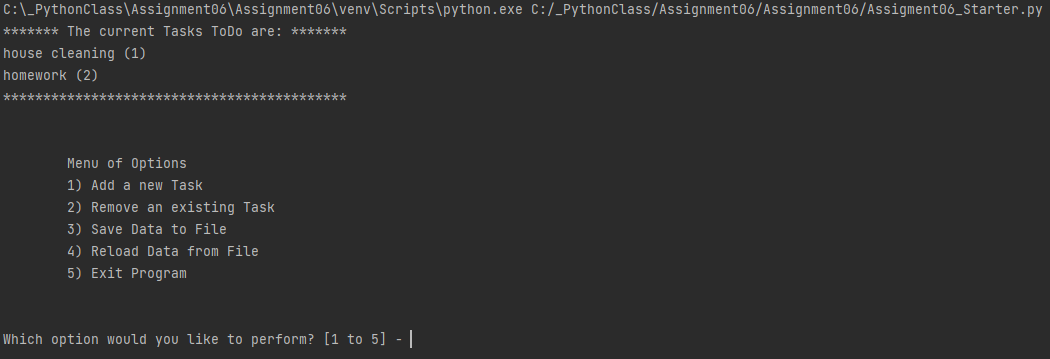
## Discussion

Figure 1 shows the header of the script. As shown, the instructor already provided with the basal script to work with. At the end of the header shows the information of the last person (i.e. me) who made changes.



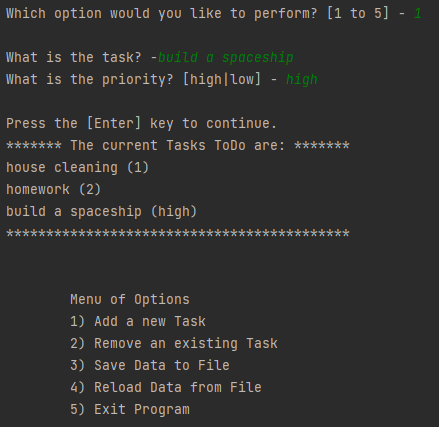
*Figure 1. A header showing who changed when, and what was changed.*

Figure 2 shows what is displayed when the script is first run. It shows current data in the txt file and shows the user the *Menu of Options* so that the user can choose the selection.



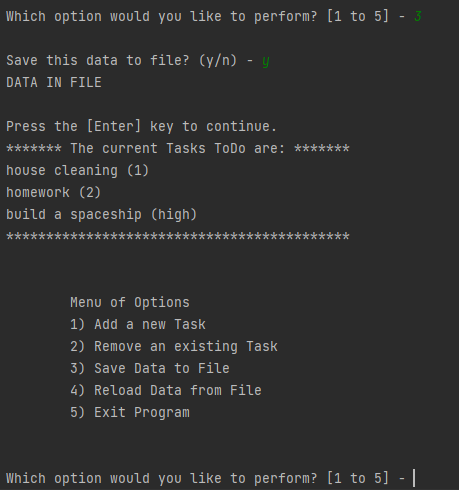
*Figure 2. When the script was started. It shows the current tasks and priorities loaded*

Figure 3 shows when option #1 was selected. When selected, the prompt asks what the task and the priority are. Once entered, the prompt asks the user to press enter to continue. When enter is entered, the current data is displayed, followed by the Menu of Options. As shows in the figure, ‘build a spaceship’ and ‘high’ were entered for the task and the priority, respectively.



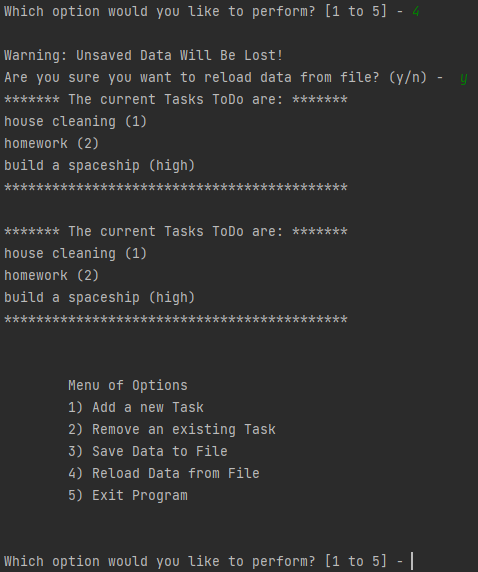
*Figure 3. Option #1 selected. A new task and priority had been typed in, but they had not been saved.*

Figure 4 shows when option #3 was selected. When selected, the prompt asking if the user is sure to save the data into the file. When ‘y’ is entered, it notifies that the data is now in file, and it shows the current data in the file, followed by the Menu of Options.



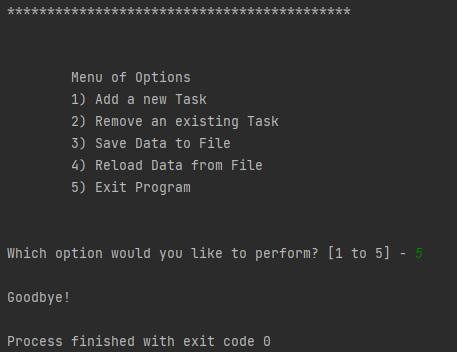
*Figure 4. Prompt when option #3 was selected*

Figure 5 shows the result when option #4 was selected. When selected, it shows the warning that any unsaved data will be lost. When the user concurs, the current data in the file is shown, followed by the Menu of Options.



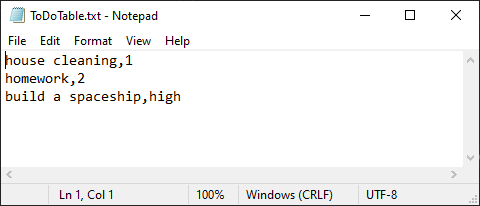
*Figure 5. Prompt when option #4 was selected*

Figure 6 shows when option #5 was selected. As expected, selecting option #5 exits the program, followed by “Goodbye!”.



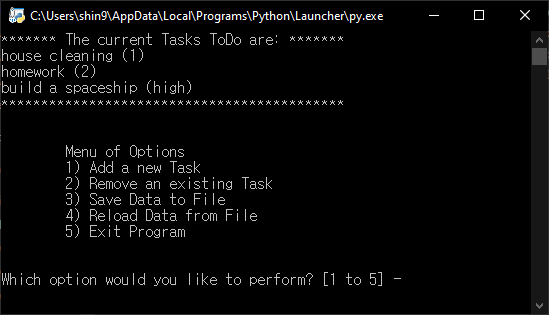
*Figure 6. Selecting option #5 exits the program*

Once exit the program, the text file was checked. As shown in Figure 7, a new task entered while running the program was saved.



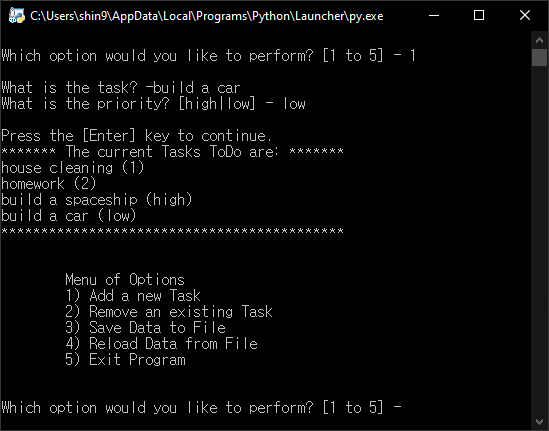
*Figure 7. The text file that shows updated (i.e. new data saved) file after the script was run.*

Once the script was run by the PyCharm, it was again run by the OS. Figure 8 shows the start page when the script was run by the OS. The script file was simply double-clicked to initiate the program.



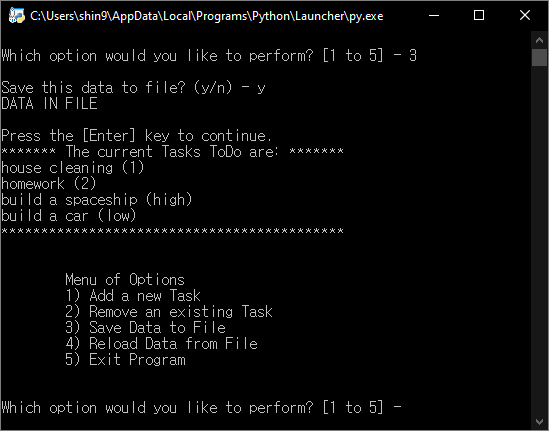
*Figure 8. When the program was started by the OS*

Figure 9 shows when option #1 was selected, and the reaction of the program was similar to when the program was run by the PyCharm.



*Figure 9. When Option #1 was selected in the OS*

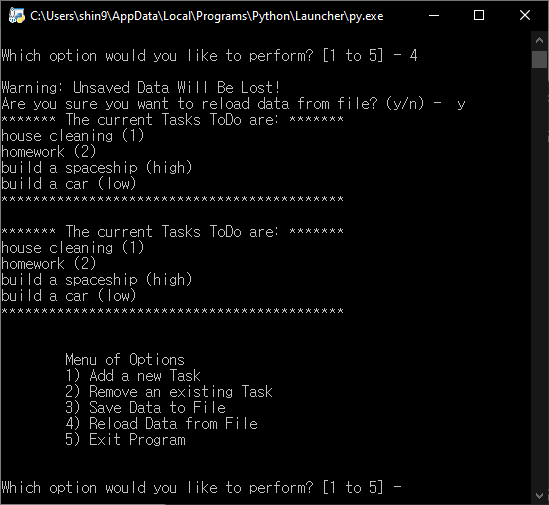
Figure 10 shows when the user selected option #3. The sequence and reaction of the program were almost identical to when the program was run by the PyCharm. This time, “Build a car” and “low” were entered for the Task and the Priority, respectively.



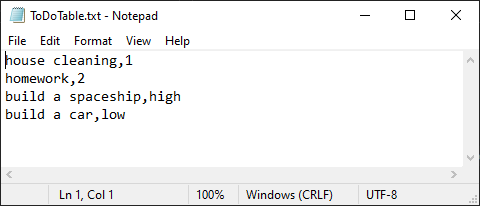
*Figure 10. When Option #3 was selected in the OS*

Figure 11 shows the result when option #4 was selected. Again, the result and reaction of the program were very similar to when the program was run by the PyCharm.

Selecting option #5 in the OS environment immediately closed the running program. Once the program running was exited, the text file was once again checked. As shown in Figure 12, the new Task and Priority entered while running the program in the OS are shown in the last line of the text file.



*Figure 11. When Option #4 was selected in the OS*



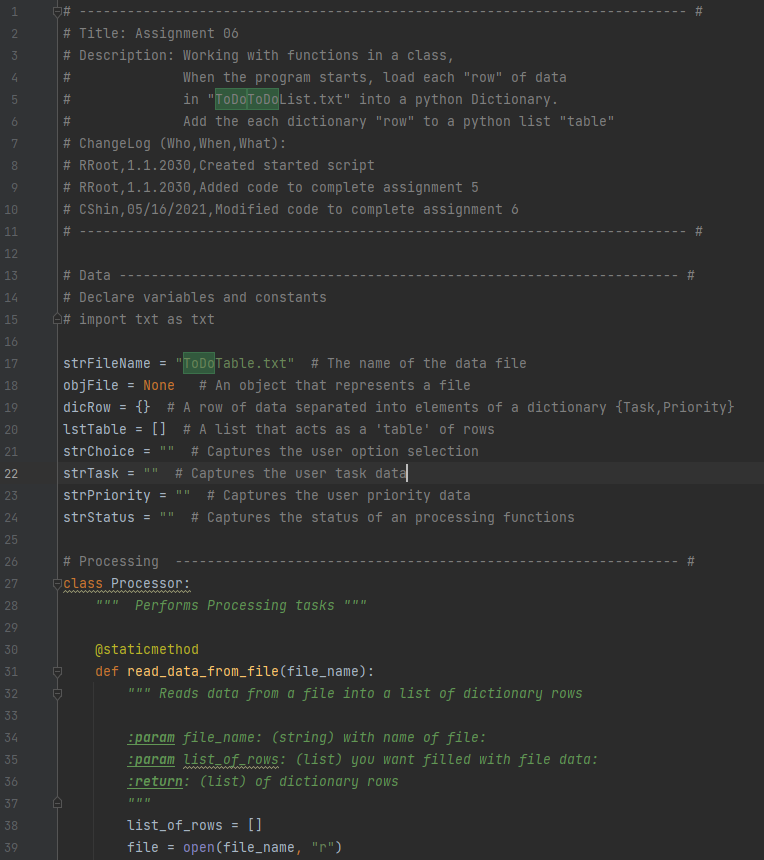
*Figure 12. Updated text file after running the program*

## Summary

Like Assignment 05, a backbone source code to work with was provided for this assignment. It was insightful, and I was learning a lot by checking out what was done and the flow of the script. On the other hand, just like the human language, how the script was structured was a bit different than how I would approach, so it was somewhat hard at first to fully understand what the original author intended to do.

## Appendix

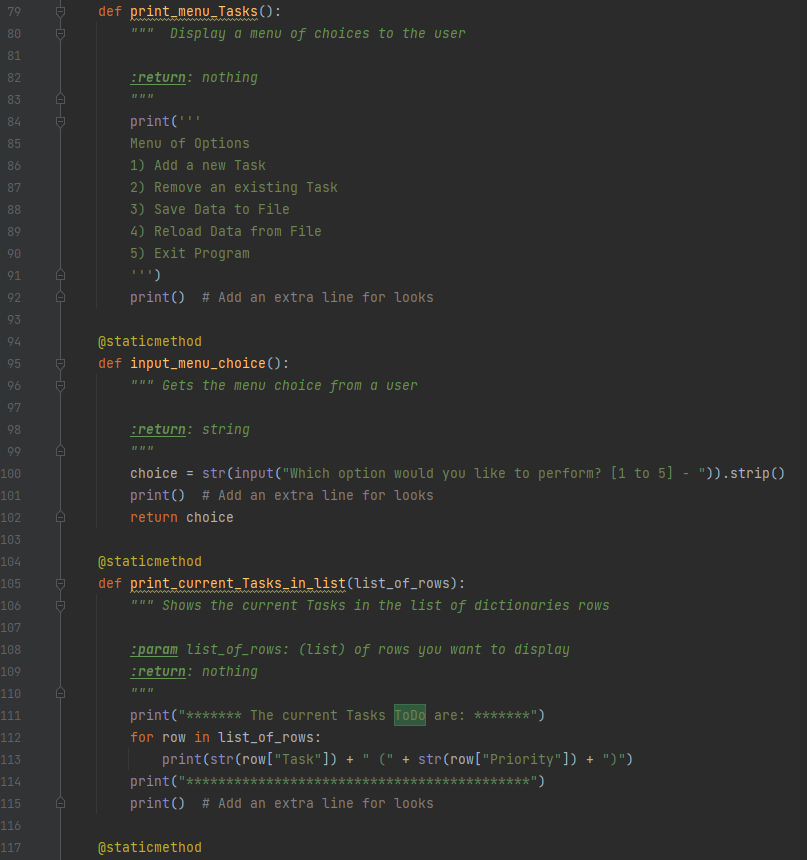
Below shows captured images of the source code for this assignment. Source code will also be available in the GitHub (URL found on p.1 of this document) and in the zip file that will be submitted.



*Figure 13. Source code capture 1 of 5*

**

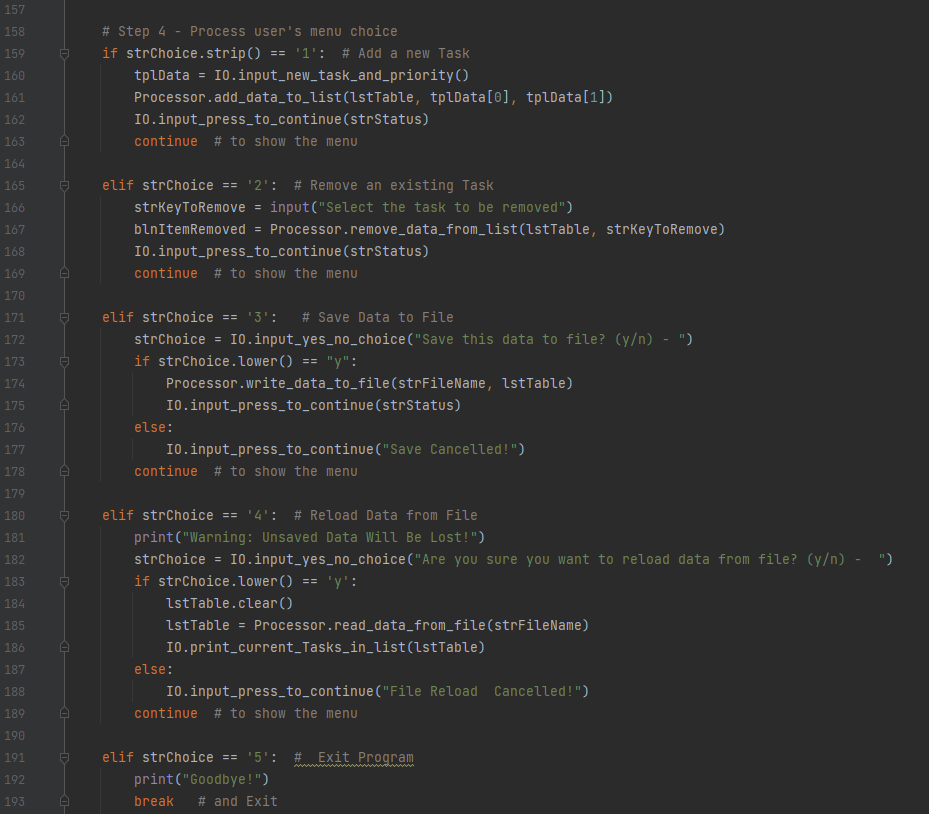
*Figure 14. Source code capture 2 of 5*

**

*Figure 15. Source code capture 3 of 5*

**

*Figure 16. Source code capture 4 of 5*

**

*Figure 17. Source code capture 5 of 5*