Mercedes Gonzalez Gonzalez

August 9nd, 2020

IT FDN 110 A

Assignment 05

<https://github.com/mglezglez/IntroToProg-Python>

WORKING WITH LISTS & DICTIONARIES:

# Introduction

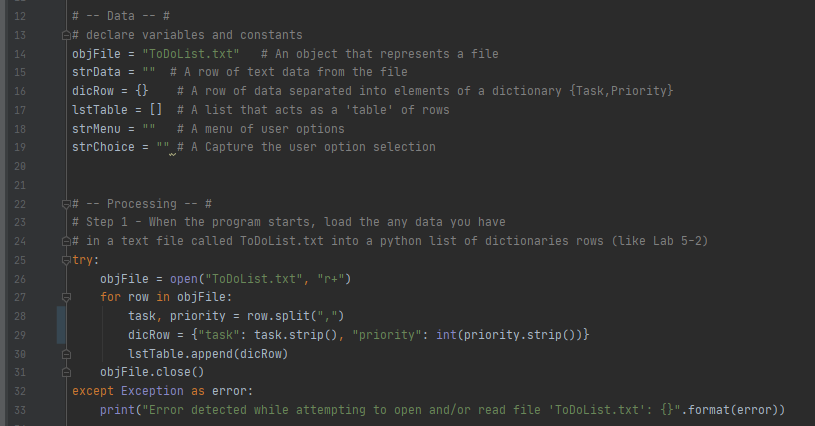
This week, as part of Module 5 in Foundation of Programming (Python) course, I learned a new data structure in Python, that is, dictionaries and I continue to expand my knowledge about lists, tuples, loops and conditional statements. This document contains a detailed description on how I modified a script template that contained a menu with options and implemented each option in the menu that would allow the user to retrieve tasks from a To-Do list file, display the tasks in the list, add new tasks, remove tasks and save the final modified list to the initial file.

# Modifying the script template

In order to implement this program, I re-used a script template provided for this assignment and I proceeded to write the code in each section that would allow the user to accomplish the option selected.

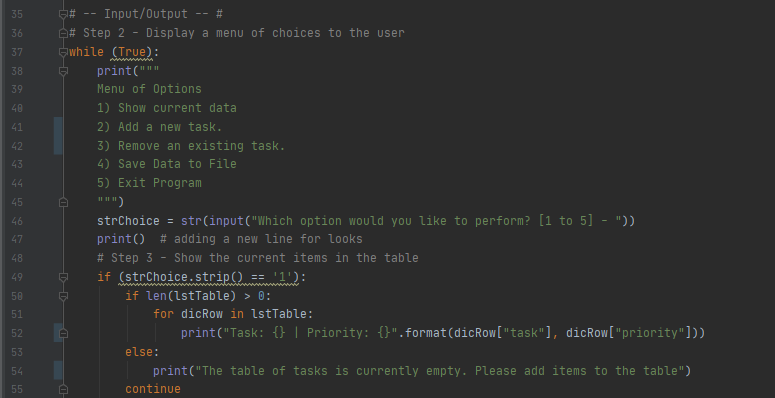
The initial section of the program defines all the variables and reads all the data from the ToDoList.txt file. I decided to include a ***try/except block*** to catch any errors that might occur while attempting to open or read the file. For example, the file might not exist in the first place.

The data is saved as a list of dictionaries in lstTable variable.



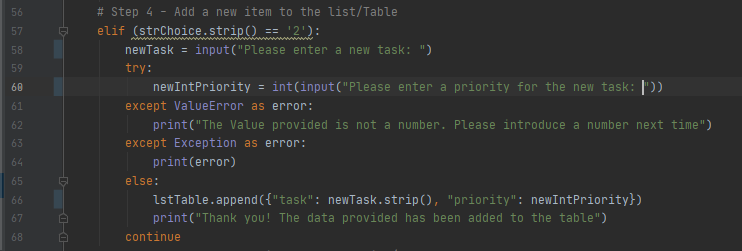
The next section contains the menu with the list of option.

Option # 1:



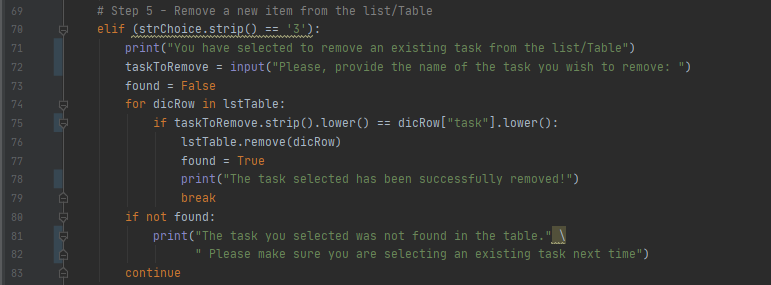
In option 1, the program iterates over the lstTable and prints each task, priority pair stored as a dictionary by using a combination of a ***for loop*** and the ***print*** statement. I decided to use an ***if/else conditional*** statement to validate whether there are tasks loaded to the lstTable, otherwise, a message is displayed to the user to indicate that the table is empty and does not contain any tasks.

Option # 2:



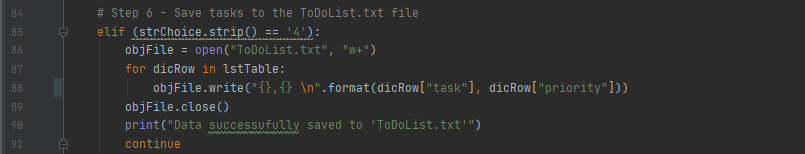
In option 2, the program asks the user for a new task, priority pair to add to the list. As can be seen, I used a **try/except block** to catch any instance when the user does not provide a number as priority. For example, he/she could provide “five” instead of 5 as priority. In those cases, an Exception would be raised while Python tries to cast a string into a float and this is catch by the try/except block and the user is notified with a message. Otherwise, the new task, priority pair is added to the table as a dictionary.

Option # 3:



In option 3, the program asks the user for a task that she/he would like to remove. After the user provides the task name, the program iterates over the table looking for such task by using a ***for loop***. If the task is not found, the user is notified of this via a message printed on the screen. If the task is found, it is removed from the table and the for loop stops.

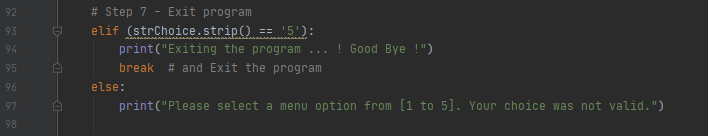
Option # 4:



In option 4, all the data currently stored as a list of dictionary is saved to the file. The “w+” option will override any data that was previously stored in the file.

Option # 5:

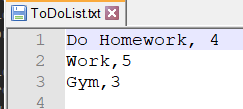
Option 5 is the simplest of all since it only prints a good-bye message to the user, followed by the else block to catch any other option above 5 which is not valid.



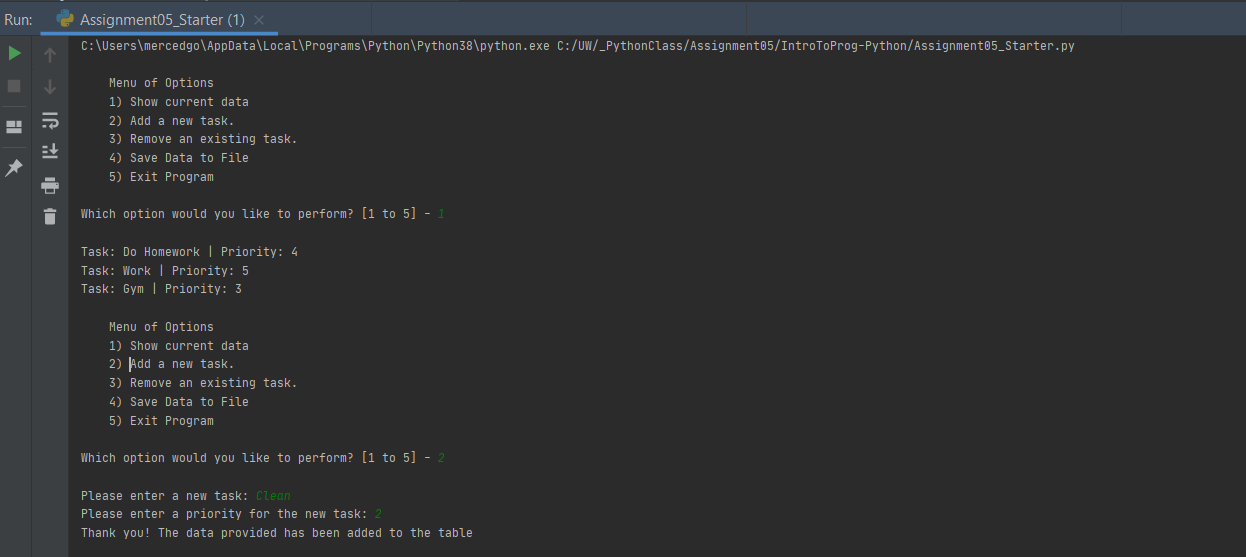
# Testing the script

The following section contains screenshots from the program while running.

The initial content of the ToDoList.txt file is shown below, before the program starts.



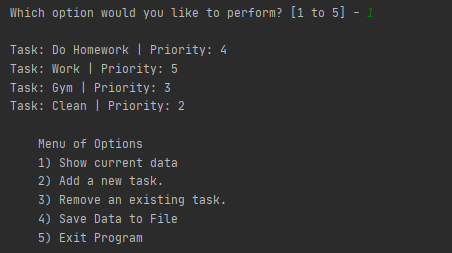
The following screenshot shows the initial of the list with option 1 and adds a new task,priority pair to the list with option 2.

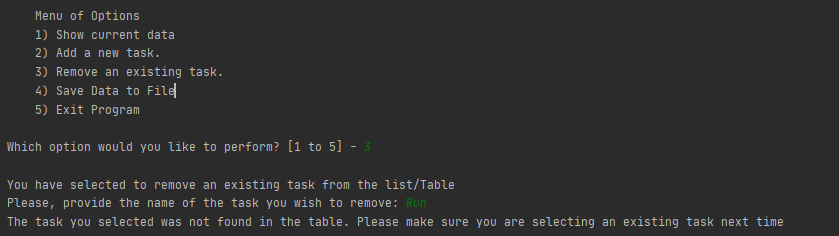


In the event that the priority for the new task is not a number, the following message is displayed on the screen.

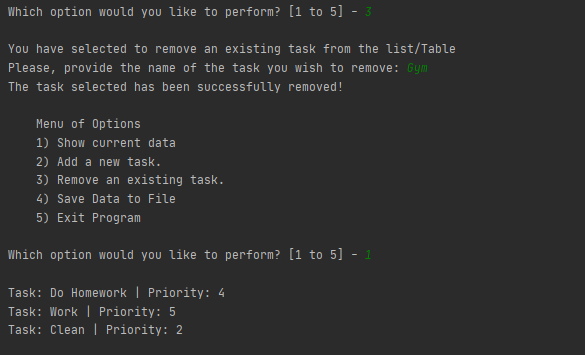
# 

Option 1 is used again to show the updated content of the table after adding a new task. Then Option 3 was selected to delete/remove a task from the table, but because the task does not exist, a message is displayed to the user indicating this and asking him/her to attempt to delete an existing task next time.

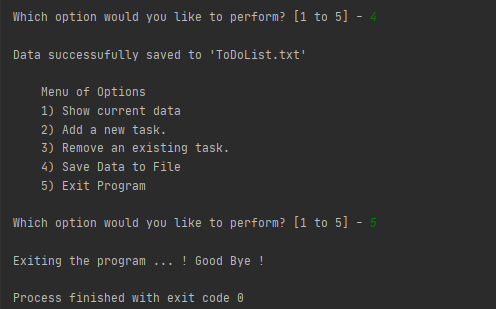


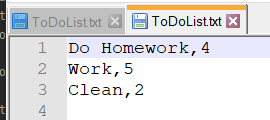


After this, the user selects to remove the “Gym” task using Option 3, and then Option 1 is used again to refresh the latest content in the table, showing that “Gym” was indeed removed.



Finally, option 4 is chosen to save all the data to the ToDoList.txt file and the program exists with option 5.





# Summary

Dictionaries and lists are two types of data structures in Python that allow solving multiple tasks regardless of their complexity. This exercise combined these data structures with conditional statements and loops to implement a shopping list of items that can be consulted and updated by using a menu of options.