

Brandon Raupe

5/22/2023

Foundations of Programming: Python

Assignment 06

<https://braupe1.github.io/IntroToProg-Python-Mod06/>

Python Script: Task List Functions

The Problem

The request of Assignment 6 is to build a script that will accept an entry for a task and a priority of the task. Since we have been provided the skeleton script framework, there will be four options the user may choose, 1) Add a new Task, 2) Remove an existing Task, 3) Save Data to a File, or 4) Exit the program.

The Script

The following critical areas of the script may be found in the Github repository linked in the header of this document. Line number ranges will be discussed where “TODO” requests were made. The figures mentioned in this document may be found within the docs folder of the previously mentioned repository.

- **Def read_data_from_file (lines 33-46):**
 - Added some try-except functionality to handle errors reading the file or when the file was not found.
- **Def add_data_to_list (lines 60):**
 - The script reads `list_of_rows.append(row)`, which appends the row data to the `list_of_rows` parameter.
- **Def remove_data_from_list (lines 72-74):**
 - I added a for loop against the `list_of_rows` list. An if statement looks to see if the provided row task matches a task in the list then the remove function is used to remove the row.
- **Def write_data_to_file (lines 86-89):**
 - I called the `file_obj` to open and write. I then created a for loop to write row task and priority, separated by a comma to the `file_obj` file before closing the `file_obj` to save the data.
- **Def input_new_task_and_priority (lines 144-146):**
 - The user is requested to enter a task followed by the priority input. Then the function finally returns the task and priority.
- **Def input_task_to_remove (lines 156-176):**
 - The `strKeyToRemove` variable captures the task to be removed.
 - The `blnItemRemoved` sets a Boolean to false in order to verify at what point the requested task is found.
 - A for loop is run against the `table_lst` and identifies the task and priority in the dictionary row values. An if statement then changes the input task and the `table_list` task to

lowercase to evaluate if found. When a task is found, the remove function is run for that row in the `table_lst`, and the Boolean is changed to True.

- Once the Boolean detects a True statement, a statement is printed to indicate that the task was found and removed. If the task was not found, a print statement indicates such feedback.
- The full function ends by running a for loop to print the current tasks and priorities in the `table_lst`.

The Results in PyCharm

Figure 1 displays the selections made during testing within PyCharm. During this testing, I entered two tasks and could save, remove and exit without issue. Figures 2 and 3 in the docs repository display the results of those actions.

The Results in the Command Prompt

Figure 4 in the docs repository displays those selections made during testing in which somewhat mirrored actions from the PyCharm testing were taken to add, remove, save, and exit the program without issue. Figure 2 also applies to how the `ToDoList.txt` file appeared following the Command Prompt portion of the exercise.

Summary

Overall this exercise stretched my skills to near maximum capability. Although much of what we discussed and worked on in the prior week was directly related to this exercise, it seemed significantly more intense as we were asked to analyze the existing code and determine what was missing. These efforts were quite challenging, with much debugging and trial and error.

I have certainly progressed in the first six weeks of the course, but I was humbled this week and hoped I could build on what we worked on this week with additional time both in and outside the classroom.