Andrei Arevalo

January 31st, 2021

Foundations of Programming, Python

Assignment05

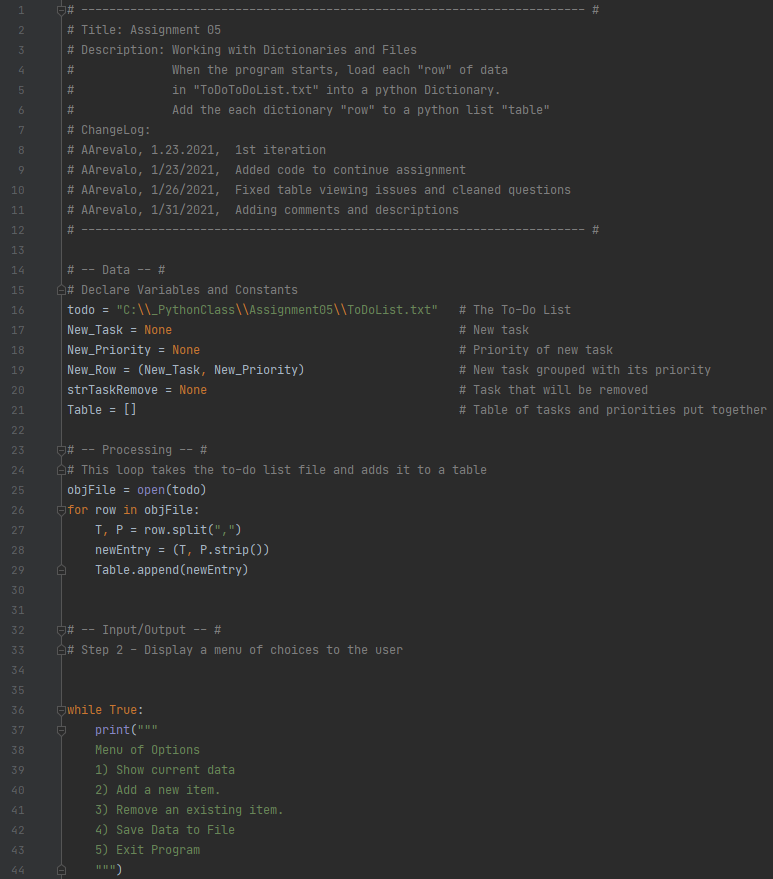
Modifying a To-Do List

# **Introduction**

In this assignment we are tasked with importing data from a to-do list and then manipulating the data several different ways including adding new tasks and deleting existing tasks. This is achieved using a while loop with 5 different options: show current data, add a new item, remove an existing item, save data to a file, and exiting the program. This code also builds on the ideas brought forth in the previous week with strings and tuples with now using dictionaries and rows and of data in tables. There were several opportunities to script this code using functions, especially in scenarios where I found myself reusing code from a different portion of the script. Functions were not allowed on this assignment but the use of them could have made this script shorter. Overall this assignment was a good example of manipulating different types of data and working with a text file.

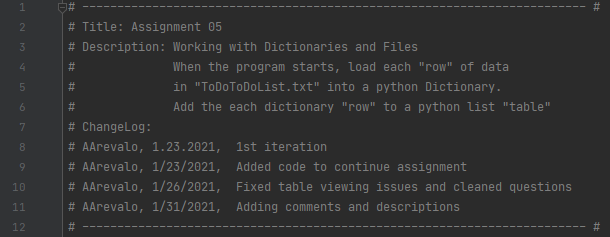
# **Creating the Script**

1. The purpose of this program is to capture and save tasks and priorities in a text file. This is building on previous weeks by adding additional features that allow the user to change the data from the text file and then resave it to a new file. The script allows for additions to the text file as well as showing the current data to the user. The main script is focused on a while loop that remains open until user exits the program. Before the while loop, the data is loaded into a table and variables are declared. (figure 1)



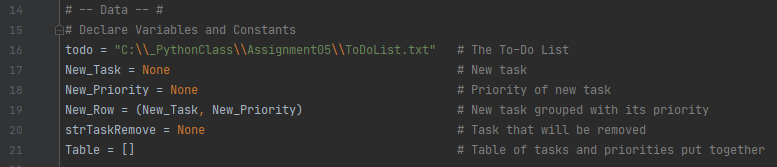
***Figure 1: The main script***

1. At the top of the script is the title block, which shows information about the script and keeps track of revision history in case the file needs to be revised in the future. The author and date are also included in this title block. In addition below the title block is a summary of the assignment with the task outlined. The main goal for this assignment was to import data from an existing to-do list and manipulate the data in several different ways. (Figure 2)



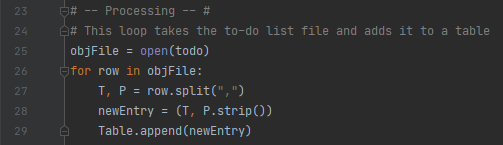
***Figure 2: Title block of the script***

1. At the beginning of the script, several variables are declared to show any future developers the naming convention I used for the different variables in the program. This allows for much easier reading in the future if someone else needs to change the script. (Figure 3)



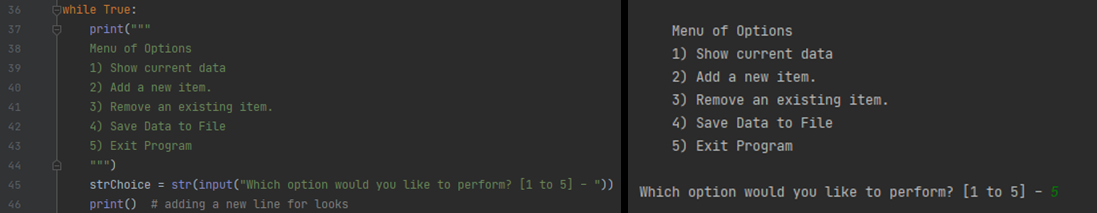
***Figure 3: Declaring variables to be used in the script***

1. The next part of the script uses the “for” loop to import the rows of data from the to-do list. The rows of data are then split by finding the comma in the row, and the left side is saved as the task and the right side is saved as the priority. (Figure 4)



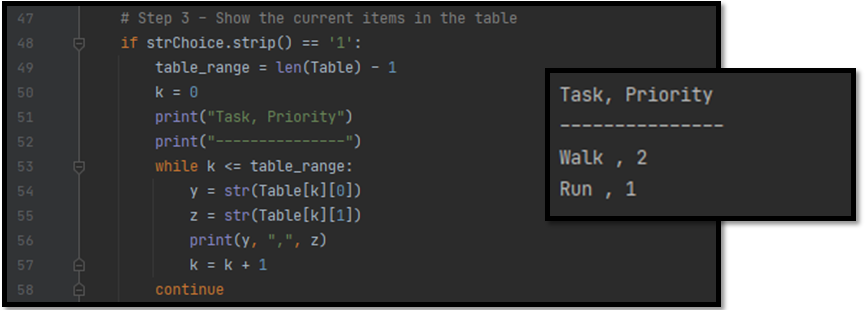
***Figure 4: Uploading the current tasks and priorities from the .txt file.***

1. The next part of the script asks the user for the input on which feature to use. This is based on the print out from the main menu. (Figure 5)



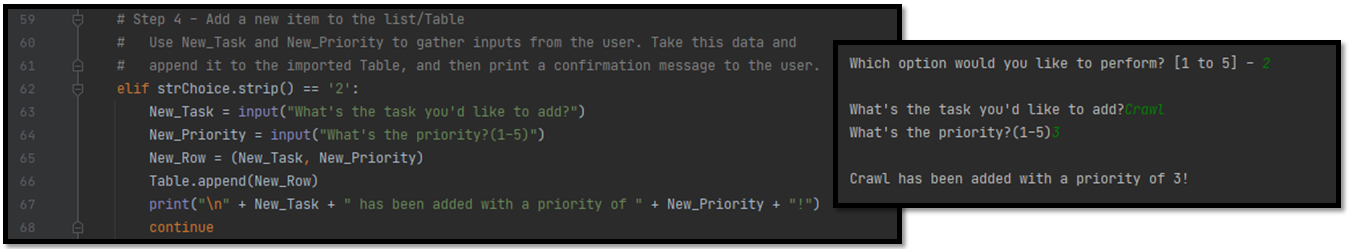
***Figure 5: Menu Display Code (left) The menu in Pycharm (Right)***

1. If the user selects 1 (show the current items in the table), the script runs through the table and for each row of the table, the task and priority are printed back to the user. The task and priority are printed back as strings to make it easier to read for the user.



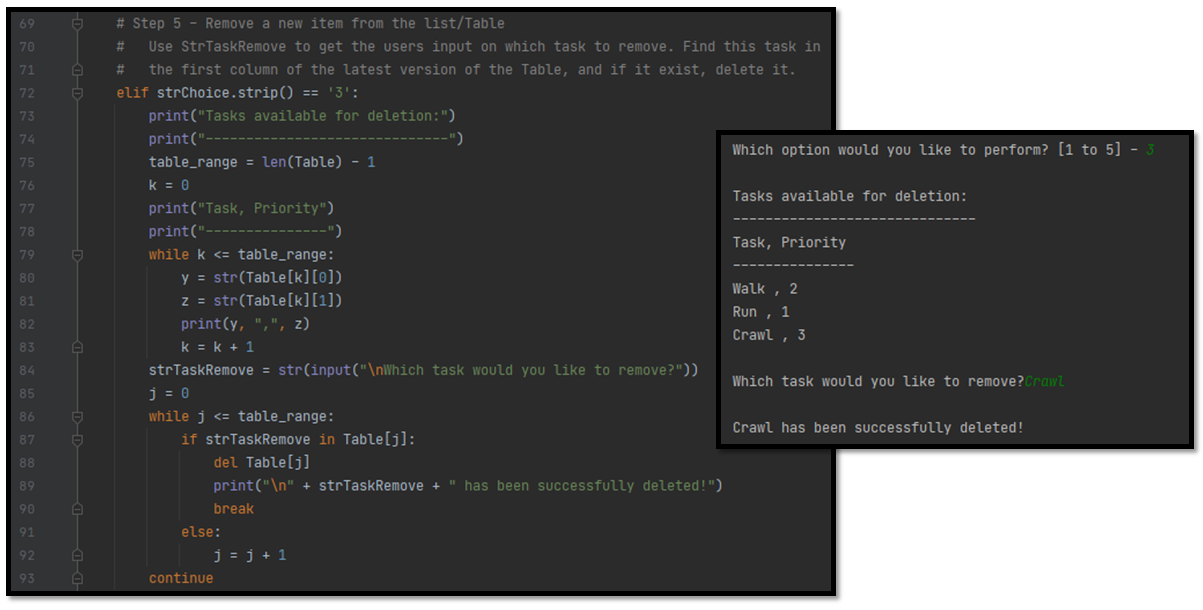
***Figure 6: Displaying the current code table (left) output in Pycharm (right)***

1. If the user selects 2 (add a new item to the list/table) the new task and new priority are requested from the user using input questions. The new data is stored in a new row that is amended to the current table of data. (Figure 7)



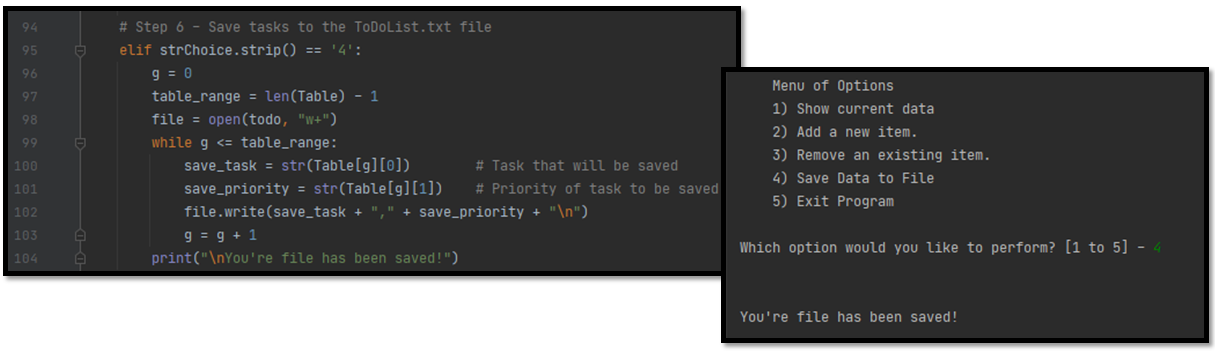
***Figure 7: Code for adding a new task/priority (left) output in Pycharm (right)***

1. If the user selects 3 (remove a new item from the list/table), the user is asked if they would like to remove a task from the list. First the current table is printed for the user so they can see which task they want to remove, then the user provides the name of the task that needs to be deleted. The script searches the table for this task and if it is found the task and priority are deleted from the table and user is notified that the task has been successfully deleted. (Figure 8)



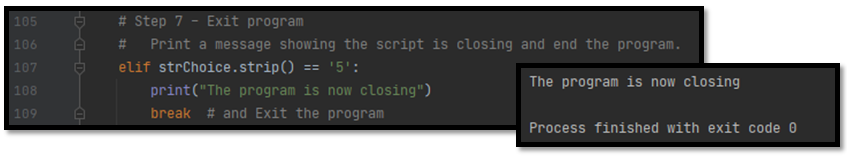
***Figure 8: Code for showing the current table and deleting an existing task (left) Output in Pycharm (right)***

1. If the user selects 4 (save tasks to the ToDoList.txt. file) the text file is opened and overwritten with the latest data from the table. A “while” loop is used to cycle through the list of tasks and priorities and the data is stored in the text file. (Figure 9)



***Figure 9: Code to overwrite the .txt file. (left) Output in Pycharm (right)***

1. If the user selects 5 (exit the program) the script is ended with a printout telling the user that the program is closing. (Figure 10)



***Figure 10: Code showing how to close the program (left) Output in Pycharm (right)***

# **Summary**

In this assignment we are tasked with importing data from a to-do list and then manipulating the data several different ways including adding new tasks and deleting existing tasks. In writing this script, I found an opportunity to write a function for printing the existing table. There were several times where I wanted to see what the latest data was on the table, and it was nice having a function to call that made the overall code shorter since I didn’t have to repeat the same line of code each time I wanted a printout of the table. In the same function, I added a feature that would save the data, and that was nice because I had two variables I could pass into the function and it gave my function multiple uses. For this assignment however, I strictly copy/pasted the display table code each time I wanted to display it. This code was a good example of continuing the use of conditional statements and adding onto what was achieved in assignment 4.