Adrian Ronsse

August 10, 2021

Foundations Of Programming: Python

Assignment 05

**Working with Dictionaries and Files**

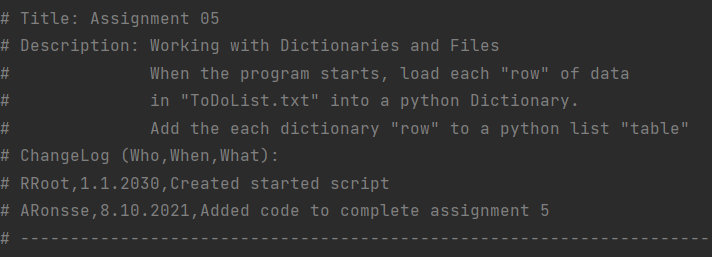
**Introduction:**

This paper details the steps to update an existing Python script with new code to perform functions described in the pseudo code, including displaying dictionary information in a list table, adding dictionary information to the list table, deleting dictionary information from the list table, and saving that information to a file.

**Writing the code:**

Step 1 – Update Change Log

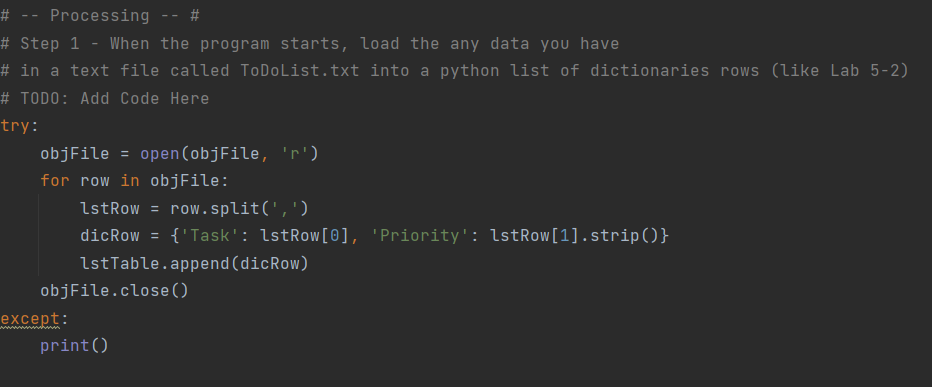
First I just needed to update the change log to show that I was interacting in the file.



*Figure 1 – Update change log*

Step 2—Read information from an existing file

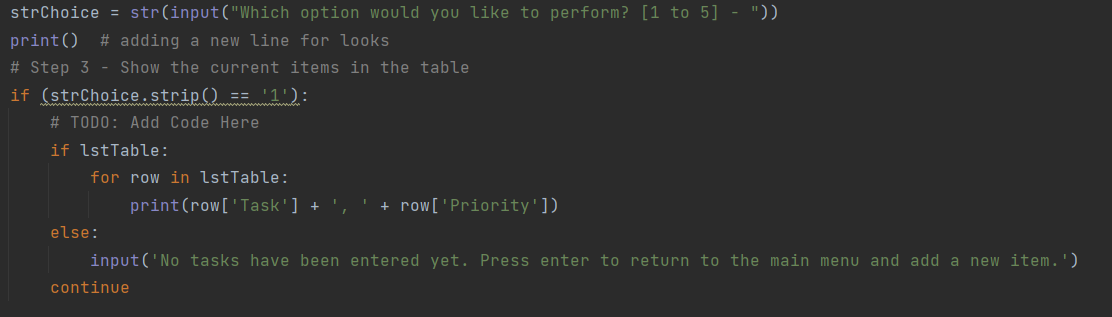
The first directive from Professor Root’s pseudo code was to load data from an existing text file. To do this I opened the objFile defined in the section above as “ToDoList.txt”, then I used a for loop to process that data into list rows, assign the data dictionary keys, and finally append the dictionary rows into a list table. Then I closed the file to prevent potential conflicts. In this section I also used the try/except feature introduced in this module so that the program would not be interrupted by an error code if no such file existed.



*Figure 2 – Processing data from an existing file*

Step 3—Display list table data

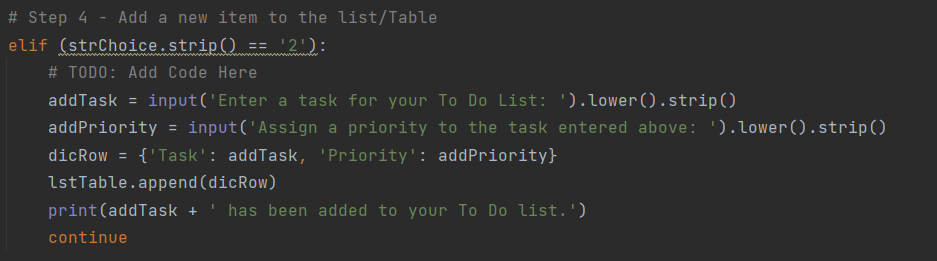
The next directive requested me to enter code to display a table of existing data to the customer. Again here I used a for loop to display the data in an agreeable format, and I used an if/else statement to capture the case where no data had yet been entered into the table.



*Figure 3 – Formatting table data for display*

Step 4—Add new dictionary item to a list table

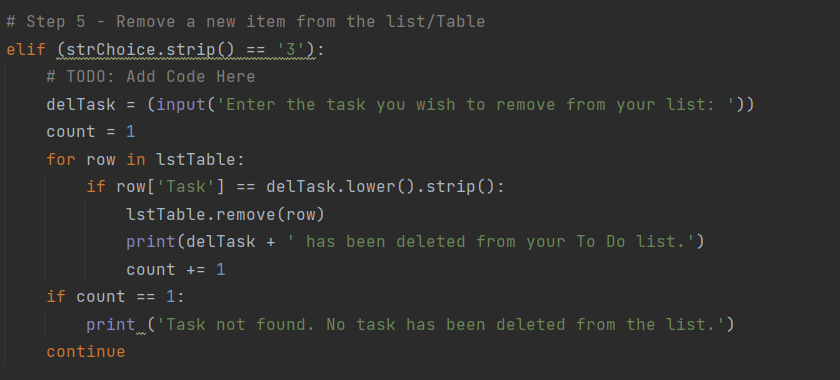
The next step required code for collecting new data from the user. Here I used the dictionary type to collect the data and assign keys for future reference. Then the data is appended to the list table with any existing data and displays a message to the user that the action has been completed successfully.



*Figure 4 – Collecting user data and assigning dictionary keys*

Step 5—Remove item from list table

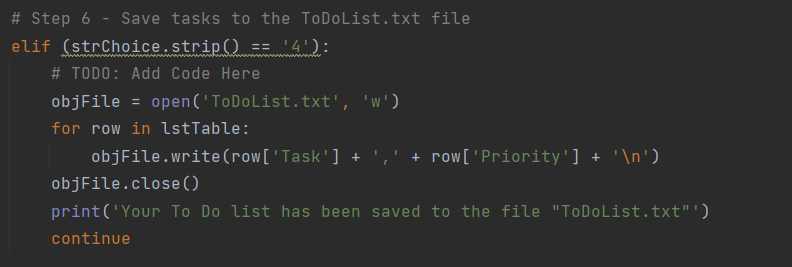
This portion of the code uses a for loop to search the list table for the data that the user wishes to delete, and then displays a message back to the user once the data is deleted. I got stuck on this part trying to use an else statement within the for loop to display a message to the user if their input was not present, but when I tried that, it created too many messages for the user since it was generating a message for each row in the table. Ultimately I referenced classmate Lauren Michelle Turner’s work posted in our class discussion board and used her strategy of the counter to prevent duplicate or conflicting messages from generating, but still accomplishing the goal of informing the user if their input had not achieved the intended result.



*Figure 5 – Using a for loop to delete data and a counter to display the appropriate message to the user*

Step 6—Save data to file

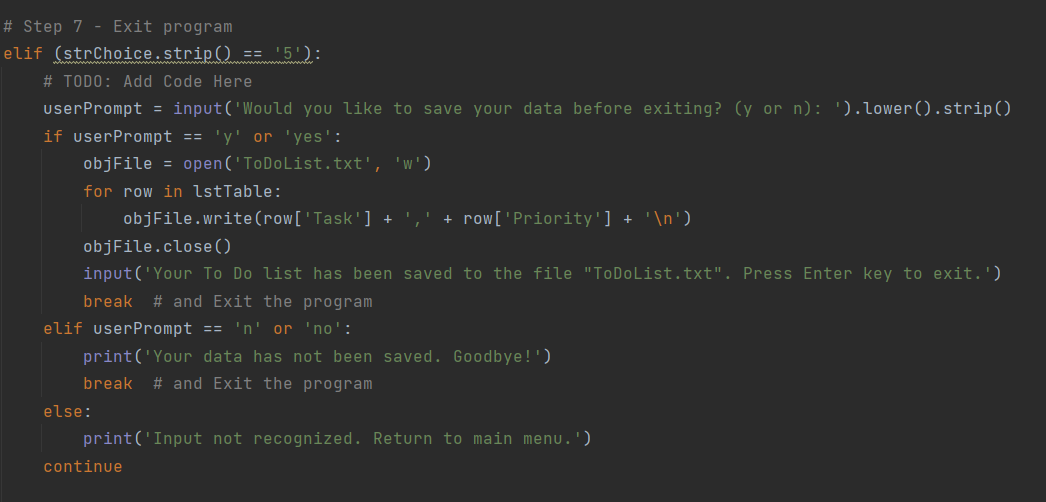
Next I input code to save the file as needed. This uses a for loop to write the components of each row and uses the keys assigned in steps 1 and 4 to pull out the user input for proper formatting in the text file. This concatenates the items with a comma in between and a new line at the end of each row so that the file will be in CSV format. I used ‘w’ instead of ‘a’ to write the file because I did not want to duplicate any existing data that was in the file to begin with, since I pulled that data into the list table in step 2.

****

*Figure 6 – Using a for loop to write rows of data to a file and using keys to pull the data into CSV format*

Step 7—Exit the program

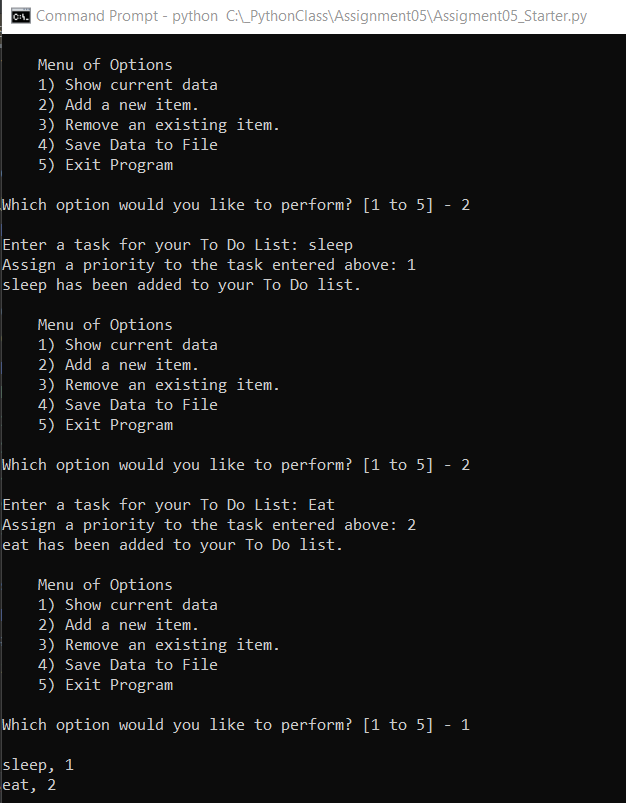
The directive here only required that the code close the program, but I know that as a user I appreciate reminders to save my work when I attempt to exit programs, so I included that feature here, giving the user an option to save once they had selected to exit the program. This portion recycles code from step 6 if the user elects to save, otherwise it displays a message to the user and ends the program with a break. It also includes a redirect to the main menu if the user does not enter one of the expected acknowledgements: “y”, “yes”, “n”, or “no”.

****

*Figure 7 – Option to save before exit*

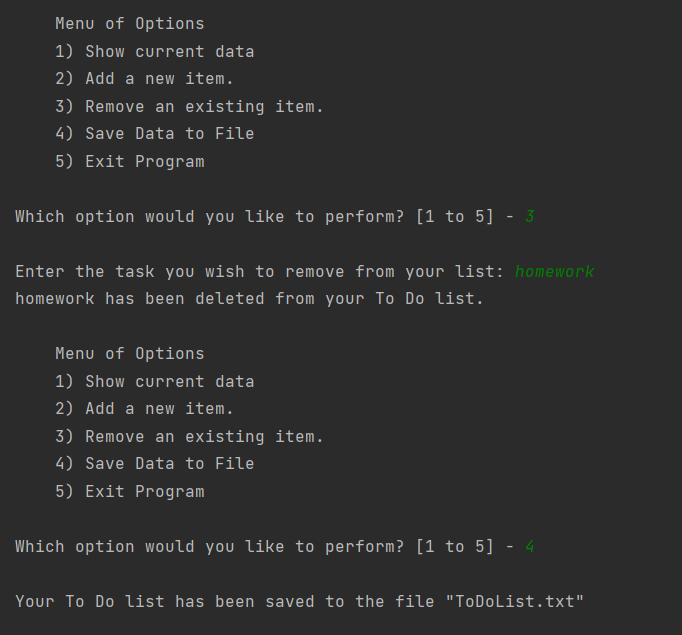
**Testing the code:**

After preparing the code, I needed to test it to make sure it was working as expected. First I used the command console. Here I see that I am able to add new items to the list and to review them:



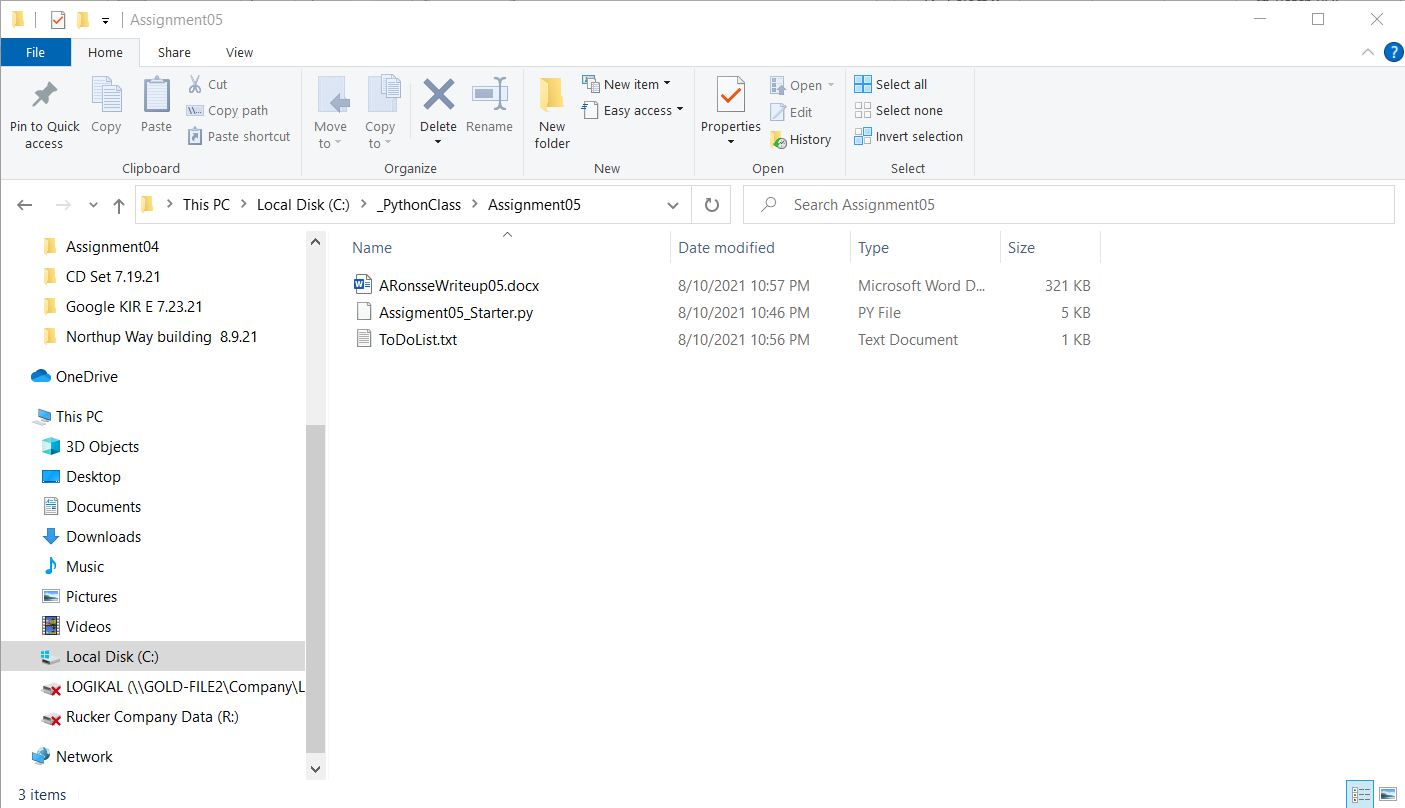
*Figure 8 – Command console test*

Then I tested the code in PyCharm as well. As shown below, I am able to delete items from the list and save to file.



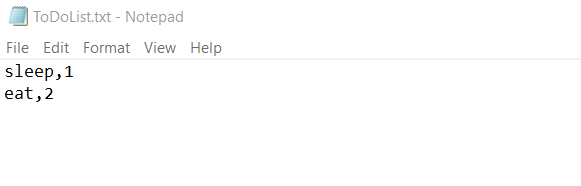
*Figure 9 – PyCharm test*

Finally I needed to check that the file had been created and that it stored the data as intended. Here I see that the .txt file has been created in my assignment folder:



*Figure 10 – Checking new file location*

Then I opened the file to make sure the data was contained:



*Figure 11 – File contains data*

**Summary:**

In this assignment I was able to update a program to include code that collected information from the user into a dictionary and used keys to update and process that data according to user selections.