Jason Killmer

2/15/2021

IT FDN 110

Assignment 5

**Assignment 5**

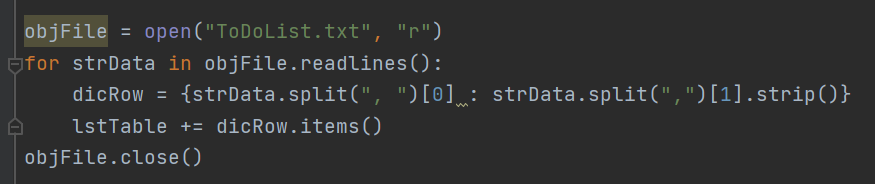
**Introduction**

This assignment combines elements of all previous assignments, and adds the additional use of the dictionary variable type. The dictionary functions much like the list, where items are stored in a sequence and can be moved, deleted, or changed. The new feature in a dictionary is that it uses strings as indices (called keys in this case) rather than integers.

The goal of the assignment is to both read and write to a file, all while incorporating elements of dictionaries and lists.

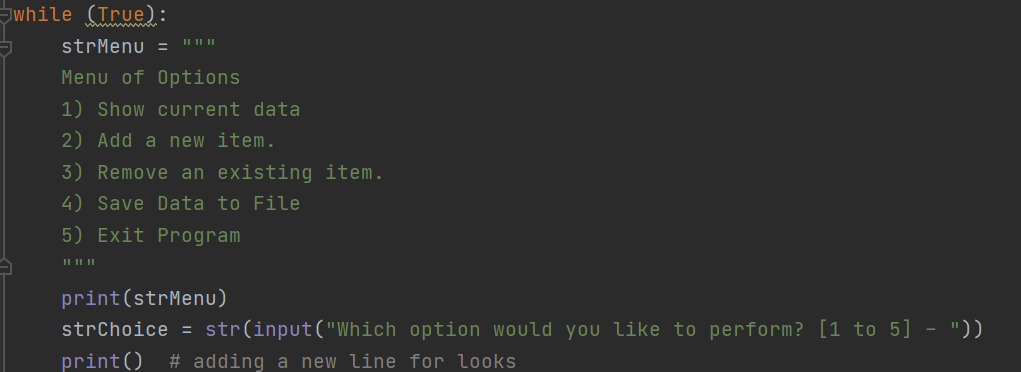
**Coding Steps**

Information from the text file is first read into the program, where the data is fed into a table that consists of rows of dictionaries. This is accomplished using the open command with the second passed string being “r” for “read”. It is important to use the close command after the data is read to ensure data can be written to the file in a later section.



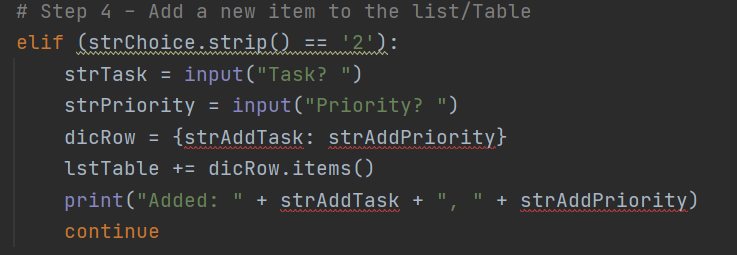
***Figure 1*** *Reading Data into the Program*

The main program structure is built around an initial “menu of options”, which operates under a while loop that will loop infinitely until a “break” command is used. The “break” command exits the loop when used, and is useful under an if/else statement to exit the program under a defined condition.



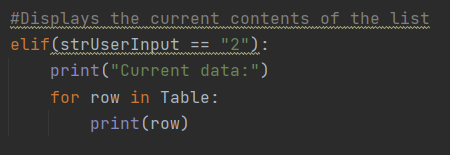
***Figure 2:*** *Infinite while loop with Menu of Options*

The user’s input is read and the program acts according to the if/else conditions, determining which course of action based on the user’s choice. The user’s data is stored in a two- dimensional list (a list of dictionaries of strings), which could be directly translated into a table. The “rows” (dictionaries of strings) can be added to the Table list either via the += operator or the append() function.



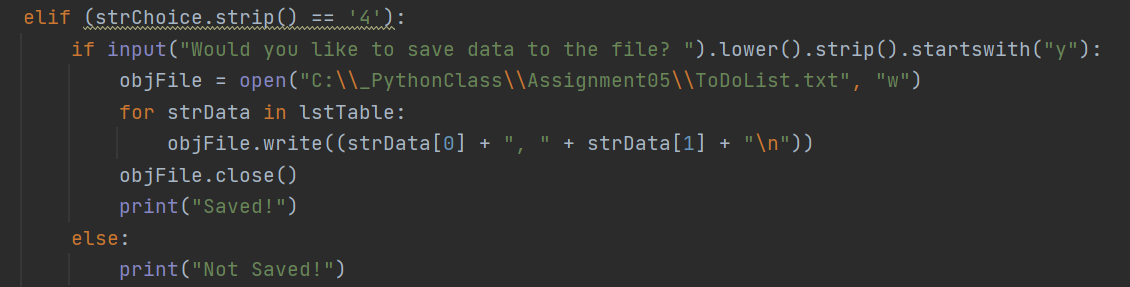
***Figure 3:*** *Adding Dictionaries to Lists*

In Python, the indexing does not need to be defined when creating a for loop, and the syntax is very simple. You can loop through a string, range, list, or many other variable types simply by using an arbitrary variable and the “in” command. The loop then iterates one step through whatever range variable you specify. However, if you require an integer value for the index within the for loop, you can use the enumerate command, which converts the index variable into a two element tuple (index, item in sequence at current index). This was not used in the final version of this program.

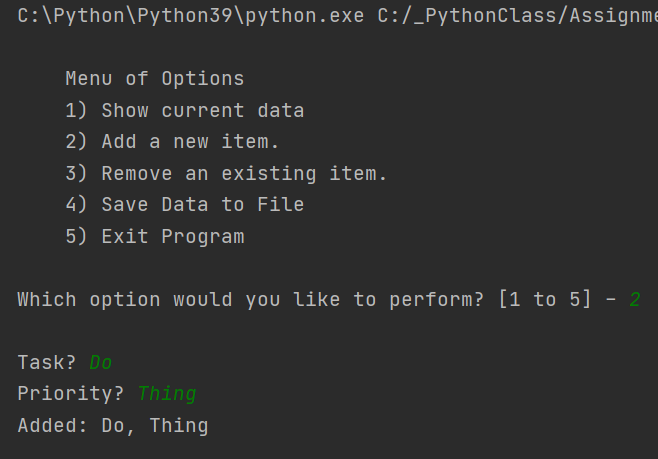


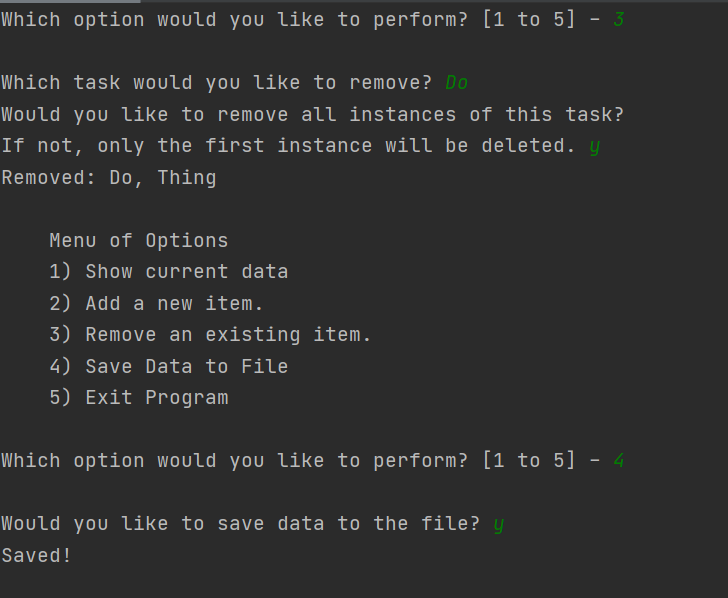
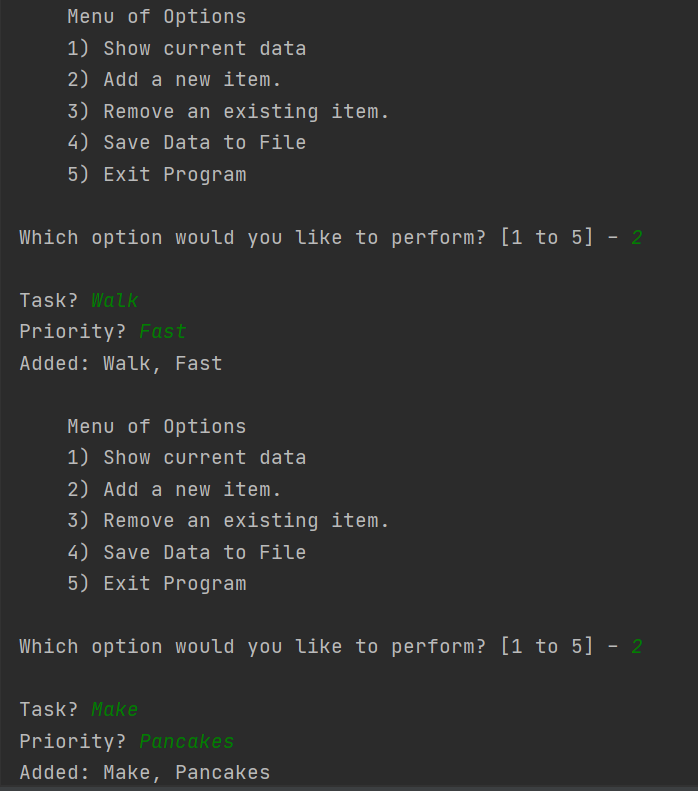
***Figure 4:*** *For Loops*

The last step in the program is to write to the text file, in much the same way this was done in Assignment 3: open(“file”), .write(“string”), and .close(). A file should always be closed when not needed to avoid unintended writing to it, as this command prevents any future write commands during the current program execution.

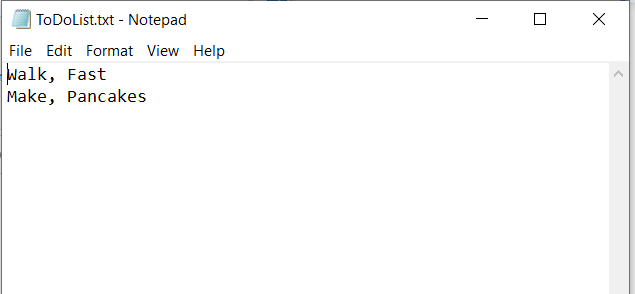
**

***Figure 5:*** *Writing to File*

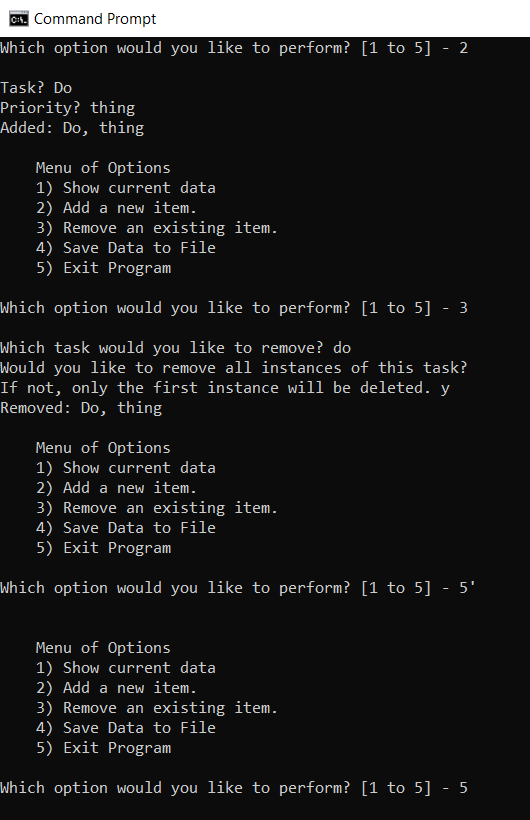




***Figure 6:*** *PyCharm Output*

**

***Figure 7:*** *PyCharm Output to Text*



***Figure 8:*** *Console Output*

**Conclusion**

This program was an exercise in the functions of writing to a file, lists, dictionaries, loops, and if statements. It combines different tools that have been learned on previous assignments into one program and ties them together to show how they can be used in conjunction with one another.