Gabriela Tedeschi

August 8, 2021

IT FDN 110

Assignment 05

# Finishing The To Do List Program

## Introduction

In this document, I will explain the steps I took to add onto the code we were given for Assignment 05 and create a program that allows users to pull up an existing to do list from a file, add and remove items, and save the updated list to the file. This program required me to move data between a file, variables, dictionaries, and lists.

## Processing

Since some of the code was already written for this assignment, I began by reviewing the existing code and comments so I would understand how the program was supposed to work and where the to-do sections were. Then, I created a text file called ToDoList and added some starter data to it.

For step 1 of the assignment, the processing section, I started by creating a variable for the file connection and using an open function to open ToDoList.txt and read from it. I used a for loop to take each row of data in the text file and add it as a row in the list, lstTable. To do this, I put the row into a list, lstRow, using the commas in the file to identify the individual elements. Then, I created a dictionary, dicRow, by assigning elements from lstRow to keys. To finish the process, I appended dicRow to lstTable. This process would repeat as many times as needed to store each line in the text file in lstTable.

Because I was using a new variable, lstRow, I went up to the data section and declared this variable.

Text

Description automatically generated

**Figure 1:** Processing section – read data from the file and store it in a list.

## Input/Output Section

The code for the menu of options was provided, so the next step was to work on the first option in the while loop, which displays current data. For this option, I used a for loop to go through each element (dictionary) of lstTable and print each task and its priority with a comma separator on a new line.

For the second option in the while loop, which allows users to add data, I started with two input functions that allow the user to enter a task and its priority level. I decided to store these inputs in the variables, addtask and addpriority, so I declared these variables in the data section. I then created a dictionary using these variables and appended the dictionary to lstTable.

Text

Description automatically generated

**Figure 2:** Beginning of Input/Output section – allow users to display current data or add new data

I struggled with the code for the next option, step 5, which allows users to remove data. While I wanted to allow users to specify which row of data they wanted to delete by entering the task name, I couldn’t figure out how to delete a specific dictionary within lstTable by referencing a value. My limited workaround was to use a simple remove function that would remove the last row of data entered. It works by using the variables addtask and addpriority, which reference whatever values were last entered in these variables by the user in the add new data option.

For the next option, step 6, which saves the new to-do list to the file, I began by reopening the text file with an open function and indicating that I wanted to overwrite the contents. Then, using a for loop and a write function, I separated the elements in each dictionary within lstTable and wrote them to the file with each task priority pair on a new line.

For step 7, I allow the users to end the loop and the program by pressing any key.

Text

Description automatically generated

**Figure 3:** Ending of Input/Output section – allow users to delete data, save data to file, and exit program

## Running The Program

Please see the screenshots below of the program running in PyCharm and the command shell.

Text

Description automatically generated

Text

Description automatically generated

**Figure 4a and 4b:** Program running in PyCharm

Text

Description automatically generated

Text

Description automatically generated

**Figure 5a and 5b**: Program running in command prompt

## Summary

In this document, I walked through the steps I took in working with the starter script to create a program that lets users pull a to do list from a file, add and remove tasks, and save the updated list to the file. This program gave me good practice working with another programmer’s code, reading from a file, and using dictionaries, though I definitely need more practice with dictionaries.