**Theresa Ward**

November 17, 2020

IT FDN 110 A

Assignment05

<https://github.com/ScrappySnacks/IntroToProg-Python>

# **“To Do” List Assignment**

## Introduction

Given starter code, we are asked to create a menu-driven program that asks for tasks and priorities, captures them as pairs in dictionary objects, and then assembles them into a table list object. The user can choose to see the data, add a new item (task), remove an item (task), or save data to a text file. This assignment document will describe the steps taken to create the program and ensure it is working as expected. As with past assignments, PyCharm is used here for script development.

## Script Development

For this assignment, we are to structure the program in PyCharm using the starter code provided. Using the separation of concerns strategy presented in the starter code, I declare variables and constants first, followed by processing and presentation. I take some liberties with the script to improve functionality, including warning the user if a task is not present in the list table upon choosing option 3. The organization and functionality of the code is described as follows:

1. I first declare an object that represents the file location, strFile. I add several variables to the starter code, such as newTask and newPriority which are needed to capture user input later in the script.
2. In the processing step, the script loads any data that is currently stored in “ToDoList.txt.” I unpack this data for presentation to the user.
3. In the input / output step, a menu of options is presented within a *while* loop that is set to Boolean True. The user enters a selection, which is formatted to a string data type. An *if* statement is used to run the script based on the user input:
   1. If the user enters “1,” the script unpacks and displays data currently in the list table, lstTable.
   2. If the user enters “2,” the script prompts the user to input a new task and priority. These are converted to lower case, captured in a dictionary object, dicRow, and appended to the list table object, lstTable. The keys “task” and “priority” are the “column” names assigned. Figure 1 shows an example of output in PyCharm when “2” and “1” are chosen.
   3. If the user enters “3,” the script prompts the user to enter a task for deletion. However, added functionality here tells the user if a task is not contained in the list table. I declare a variable, *i*, which will be used as a flag to alert the user. If the task is contained in the list, the script iterates this variable to equal one. If the task is not found, the variable remains equal to zero and displays the message, “Task is not listed. Try again.” The deletion of the task is accomplished through the remove() method. Figure 2 shows an example of output in PyCharm when “3” is chosen.
   4. If the user enters “4,” the script will open “ToDoList.txt” and save the list table contents to the file. If the user enters “5,” a break will end the program. Figure 3 shows an example of output in PyCharm when “4” and “5” are selected. Figure 4 shows the contents of the list table object successfully saved to file.

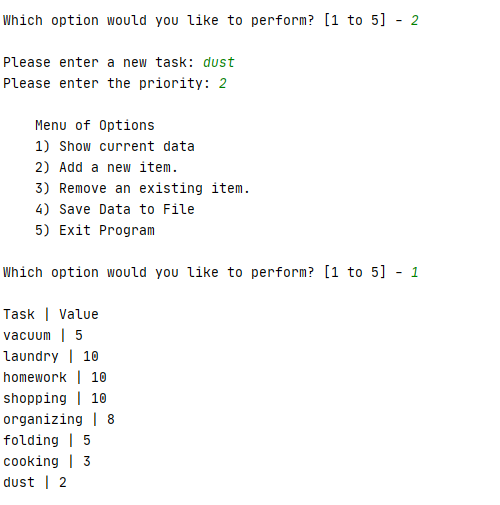


Figure 1. Output in PyCharm when “2” and “1” selected

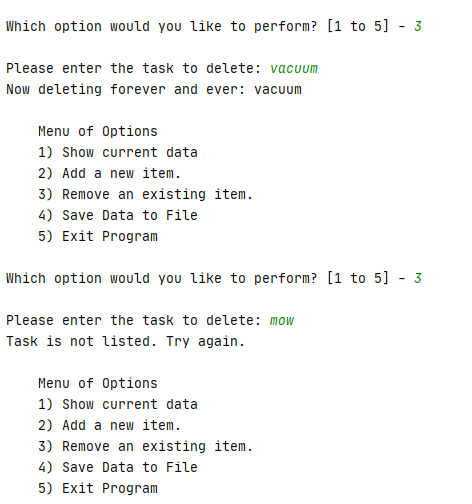


Figure 2. Output in PyCharm when “3” selected. The script will inform the user if the task does not exist.

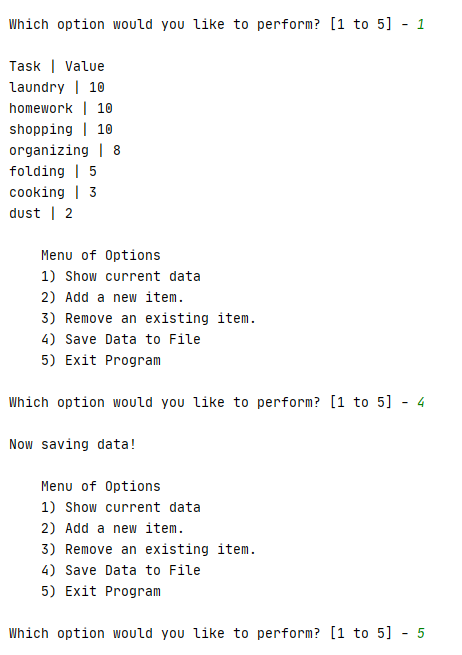


Figure 3. Output in PyCharm when “4” and “5” selected. “1” is selected here to verify   
that the contents of the text file are correct.

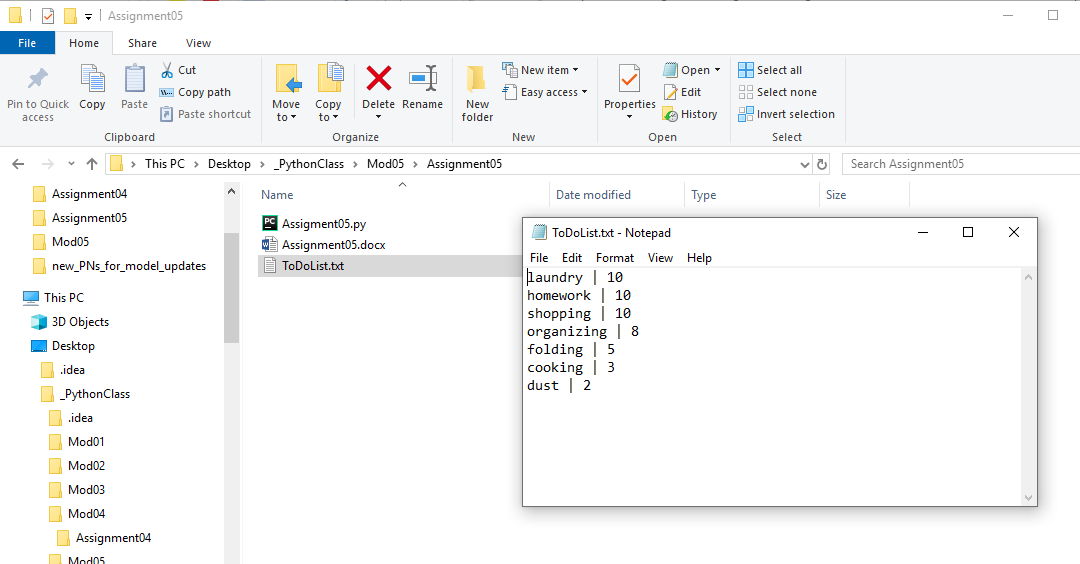


Figure 4. Example of data written to ToDoList.txt.

The script can also be run successfully from the command window as presented in Figure 5.

Although starter code was provided, this assignment was challenging in two primary ways:

* Understanding the intent of the variables declared: I both added and deleted variables. For instance, I removed the strMenu variable, as it didn’t appear to have a role in the script.
* Employing both dictionary and list methods to organize the data. “Key” and “value” concepts while thinking of a table having columns and rows can be at times confusing for a new programmer. However after this assignment, I can see the advantages of constructing the data table in this way.

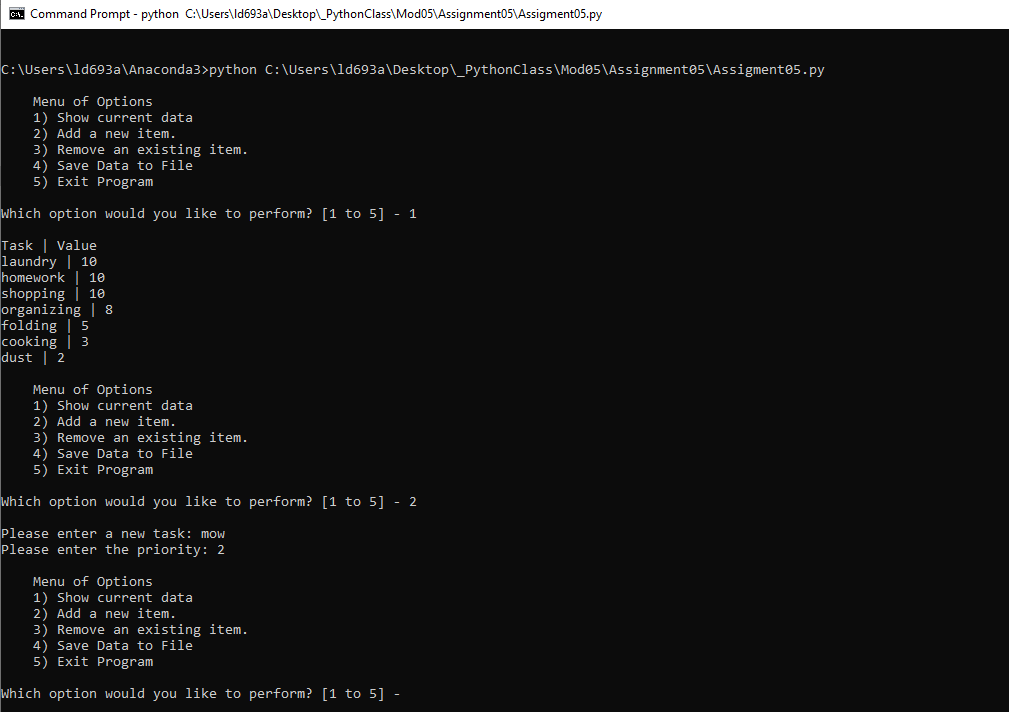


Figure 5. Result of running the Python script via the command window

## Summary

In summary, I was successful in building and executing a script that meets the requirements of Module 05. The script employs a menu so that the user can select one of several options: show current data, add a new task, delete a task, save data to file or exit the script.