

Tim Randall
May 16th, 2023
Intro to Programming (Python) – IT FDN 110A Spring 2023
Assignment 05
<https://github.com/trandallUW/IntroToProg-Python>

To Do List with Task & Priority

Introduction

The purpose of this assignment was to develop an understanding of inputting data from a file, letting a user modify that data, and saving the data to an external file. The process taught me how to store data into dictionaries and use the standard functions of a dictionary to process or display that data efficiently. This assignment was also a lesson on how to take someone else's code and modify it to meet my end requirements. Last but not least, the assignment taught me how to use GitHub and collaborate with others using that platform.

Overall Process & Learning

My first step was to take the starter code that was given to me and get an understanding of the overall requirements for the task based on the comments provided by Randall Root. In general, the starter code gave a good general outline, with several variables declared, but the code was very incomplete.

I used code from my previous assignments and approached the code one section at a time, using Pycharm to run the code and print() functions as placeholders to show that my code was working. The concept of “keys” and “values” within a dictionary was difficult for me to grasp. The first step of reading an external file and formatting the data into a dictionary took a long time and a lot of trial and error (and watching videos / researching the internet). Figure 1 shows my final code for processing the data from an external file into a dictionary.

```
28 # -- Processing -- #
29 # Step 1 - When the program starts, load any data that is
30 # in a text file called ToDoList.txt into a python list of dictionaries rows
31 objFile = open(strFile, "r")
32 for row in objFile:
33     strData = row.split(",")
34     dicRow = {strTask: strData[0], strPriority: strData[1].strip()}
35     lstTable.append(dicRow)
36 objFile.close()
```

Figure 1 - Processing the data into a dictionary

Displaying the menu options to the user was fairly straightforward and was given as part of the starter code. I decided to declare a variable for the user menu. This was not required, but I chose to do this in case I needed to use the “user menu” in other parts of the code (typing a string vs. typing the menu every time). Figure 2 shows the user menu section.

```

38 # -- Input/Output -- #
39 # Step 2 - Display a menu of choices to the user
40 strMenu = ("""
41     Menu of Options
42     1) Show current data
43     2) Add a new item.
44     3) Remove an existing item.
45     4) Save Data to File
46     5) Exit Program
47     """)

```

Figure 2 - User Menu as a Variable

Displaying the user data and extracting the information from a dictionary took a little research. It took me a while to realize how the keys and values worked. I also researched examples online and used some code for formatting indentations to make the task and priority list more readable. Figure 3 shows my final code:

```

55 # Step 3 - Show the current items in the table
56 if (strChoice.strip() == '1'):
57     print("{:<20} {:<20}".format('TASK', 'PRIORITY')) # Header
58     print("{:<20} {:<20}".format('____', '_____')) # Separator
59     for row in lstTable:
60         print("{:<20} {:<20}".format(row[strTask], row[strPriority]))
61     continue
62

```

Figure 3 - Learning how to format and display dictionary data

Developing the code to add a new set of data was not too complex. I decided to declare additional variables for user inputs (strTask and strUserPriority). I would probably pick different names for these variables if I were to start over from scratch – the names were too similar to other variables (lesson learned – it's important to have unique and simple naming conventions). Figure 4 shows the code for adding data.

```

63     # Step 4 - Add a new item to the list/Table
64     elif (strChoice.strip() == '2'):
65         strUserTask = input("Enter a task: ")
66         strUserPriority = input("Enter a priority: ")
67         dicRow = {strTask:strUserTask, strPriority:strUserPriority}
68         lstTable.append(dicRow)
69         print("Task added to list! ") # User confirmation message
70         continue
71

```

Figure 4 - Adding Data

The code that I used for removing an item was very simple and I decided not to over complicate it. I wasn't able to figure out the for / if / else statement logic to not have my code print "Task not found" for every row that was FALSE. I would prefer to have my code only display the message "Task not found" once. That will be a lesson for the next assignment. Figure 5 shows the final code.

```

72     # Step 5 - Remove a new item from the list/Table
73     elif (strChoice.strip() == '3'):
74         strRemove = input("What task would you like to remove?: ")
75         for row in lstTable:
76             if row[strTask].lower() == strRemove.lower():
77                 lstTable.remove(row)
78                 print(strRemove + " - Task removed!") # User confirmation message
79             else:
80                 print("Task not found") # User message that item was not found
81         continue

```

Figure 5 - Remove an item

The code to save and exit the program was used from previous assignments and the class lectures. By this point, I was able to see the benefit of using string variables and embedding them in the print messages – to have better feedback to the user on what my code was doing. Figure 6 shows the code.

```

83     # Step 6 - Save tasks to the ToDoToDoList.txt file
84     elif (strChoice.strip() == '4'):
85         objFile = open(strFile, "w")
86         for row in lstTable:
87             objFile.write(str(row[strTask])+","+str(row[strPriority])+"\n")
88         objFile.close()
89         print("Data saved to " + strFile + "!") # User confirmation message
90         continue
91
92     # Step 7 - Exit program
93     elif (strChoice.strip() == '5'):
94         print("Exit program is complete") # Display to user that program has ended
95         break # and Exit the program

```

Figure 6 - Save to File and Exit

GitHub

The assignment also taught me how to sign up for GitHub and use it to post my code. It will be interesting to use Git and GitHub for getting open source code from others and using the functionality created by other coders. This was great to finally learn about this website and how it is used.

My user name is trandallUW and my repository for the class is located at:

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

Final Code

My final code worked as required. Figure 7, 8, and 9 show my code working in Pycharm. Figure 10 shows my code working in the command shell (for the display menu option – all other code works the same).

```
C:\_PythonClass\venv\Scripts\python.exe C:\_PythonClass\Assignment05\ToDoList.py
```

Menu of Options

- 1) Show current data
- 2) Add a new item.
- 3) Remove an existing item.
- 4) Save Data to File
- 5) Exit Program

Which option would you like to perform? [1 to 5] - 1

TASK	PRIORITY
----	-----
Clean	Low
HW	High
sleep	low

Figure 7 - Display Current Data

Which option would you like to perform? [1 to 5] - 2

Enter a task: Laundry

Enter a priority: Low

Task added to list!

Figure 8 - Add a task

What task would you like to remove?: *Laundry*

Task not found

Task not found

Task not found

Laundry - Task removed!

Menu of Options

- 1) Show current data
- 2) Add a new item.
- 3) Remove an existing item.
- 4) Save Data to File
- 5) Exit Program

Which option would you like to perform? [1 to 5] - *4*

Data saved to ToDoList.txt!

Menu of Options

- 1) Show current data
- 2) Add a new item.
- 3) Remove an existing item.
- 4) Save Data to File
- 5) Exit Program

Which option would you like to perform? [1 to 5] - *5*

Exit program is complete

Figure 9 - Remove a task, save to file, exit program

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\12064>cd c:\_PythonClass\Assignment05

c:\_PythonClass\Assignment05>python ToDoList.py

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

TASK                PRIORITY
----                -
Clean               Low
HW                  High
sleep               low
```

Figure 10 - Command Shell

Summary

This assignment taught me how to take external data, process it into dictionaries and tables, and then modify / save the data to an external file. It also was a great lesson in how to use someone else's starter code, and share / post comments on other user's code on GitHub. I look forward to learning more about the technical details behind dictionaries and the way to organize my code into functions.