# **Ian Sanchez**

# **Nov 06, 2019**

# **IT FDN 100 A**

# **Assignment05**

Managing a To-Do List

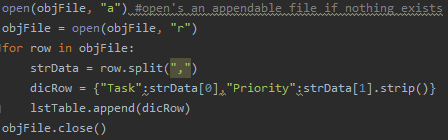
Introduction

For this week’s assignment, we were instructed to modify a “starter” script to complete the required task. The program was to be able to manage a to-do list with two columns of data, Task and Priority. The user would be able to view, add, delete and save data to the file.

Processing

The data for this assignment was already provided so we will start with writing the processing part of the script. We will tell the program to load data that is in ‘objFile’ into a python dictionary like so. Set strData to split the row with a “,”; set the dicRow with the keys “task” and “priority” and strip any unnecessary spaces (Fig. 1)

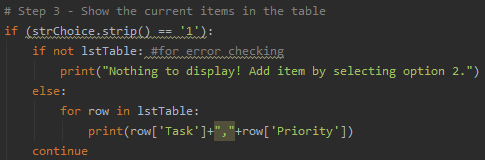
**Figure 1. Processing**



Show current data

Because the input and output was provided in the starter script, we will move on to writing the code for the first option which will be to Show Current Data. I started the code with ‘if not’ to see if there is an ‘lstTable’ to read data from. If none exists, it will print a message to the user that there is no data and to select another option to input data. Otherwise, it will print the row for ‘task’ and ‘priority’. (Fig. 2)

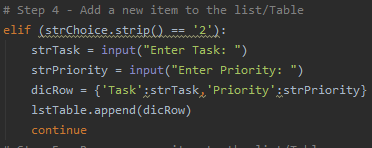
**Figure 2. Show current data**



Adding new items

To add new items we will create 2 new string values to equal the user’s input. In the dictionary we set those values to the proper key and append to lstTable. (Fig. 3)

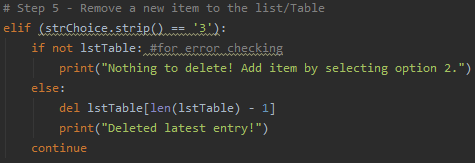
**Figure 3. Adding new items**



Removing data

For this part I start with writing ‘if not’ just like for the first option to check for lstTable and display a message to the user to add data. ‘Else’ it we have the program del an item from “lstTable[len(lstTable) – 1]” which translates to the last entry in the table. The program then prints to the user that the latest entry has been deleted.(Fig. 4)

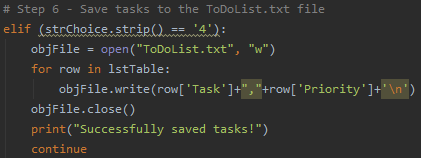
**Figure 4. Removing data**



Saving the data to a file

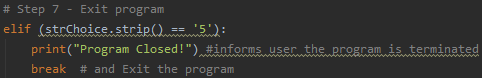
Option 4 in the program is to save the data into a “ToDoList” text file. We have the program open the “ToDoList.txt” and use ‘w’ to write the data from memory over anything that is currently in the text file. objFile.write will write the data under the Task and Priority keys and also perform a carriage return. The program also prints to the user that the data has been saved. (Fig 5)

**Figure 5. Displaying current data**



Exit program

The last option from the menu is to close the program. Pretty simple and straightforward. I added a print command just to show the user that the program has indeed closed.



Summary

In this assignment we learned how to work with someone else’s code and add to it to complete a given task. We also learned Separation of Concern’s and the importance of it in the code. In the module we learned about how dictionaries and lists worked together. I was able to apply that knowledge in how to create and modify a Python dictionary and write it onto a text file.