Chloe Wei

May 14, 2020

IT FDN 100 A

Module 06 Assignment 06

<https://cvarw.github.io/IntroToProg-Python-Mod06/>

Adding Functions to Legacy Code

Introduction

This week we will practice an important skill for programmers: working with legacy code. Working with Randal’s script for the ‘ToDoList’ from Module 05’s assignment, I will create functions in their respective classes then have them called based on the user’s selection from the User Interface, UI.

A close up of a map

Description automatically generated

Figure : Structure of ToDoList

The menu options will be reorganized as functions in either the input-output class, “IO”, or the processing class, “Processor”, and will respond according to the inputs from the UI.

Formulating a Plan and Solution

When the script is run, the ‘ToDoList.txt’ will be read and the data stored into the list, lstTable. The data will be transposed into dictionaries acting as rows of data containing the keys, “Task” and “Priority”. Functions that will read the data from the text file, add or remove data from the lstTable, search through lstTable for a “Task”, and write the data from lstTable into the associated text file will be under the class “Processor”. The Menu, Input variables, and display of data will be under the class “IO”.

First, I ran Randal’s script and test all the options making note of the systems output based on the inputs. After making a run of the script, I read through the code and wrote down a list of the functions needed and where they were to be stored.

Adding New Data

Using the new function to assign data to global variables, strTask and strPriority, I then pass them into the “add\_data\_to\_list()” function, which will then add a new dictionary into the list, lstTable.

A close up of a sign

Description automatically generated

Figure : Replace code with call to IO.InputNewItem()

In the IO class, I relocated the input strings and assigned the new task & priorities to variables, task & priority, respectively.

A picture containing clock

Description automatically generated

Figure : New function in Class IO

After testing the code, I relocated the portion that appends the lstTable into the class, Processor, see figure 4 below.

A close up of a sign

Description automatically generated

Figure 4: 'add\_data\_to\_list' function

Deleting an Item

As with adding a new task, the assigning value to a variable was relocated to the IO as the function, “input\_task\_to\_remove()”. The value was parsed in the function remove\_data\_from\_list(), if it made a match, then it would be removed and UI would display it was found and deleted. If it was not found in the list, then the UI would display that a match was not found.

A screen shot of a smart phone

Description automatically generated

Figure 5: Removing task from lstTable

This script will stop searching through the remaining values upon its initial match and remove that. In other words, if there are more than one entry of the same name, only the first matching value will be removed.

Saving Data to a test file

Writing the data back into the file will only occur after the ‘Double-Check’ was made and user confirms saving.

A picture containing black, table, large, holding

Description automatically generated

Figure 6: when ready to save data

The write\_data\_to\_file() function will then write the data into the text file, ‘ToDoList.txt’.

A picture containing phone, meter

Description automatically generated

Figure : write data back to file

Now that all the functions are created, I did a few more tests of the script.

A screenshot of a cell phone

Description automatically generated

Figure 8: Removing a task in PyCharm

As I have been using the PyCharm compiler for tests & debugging, I was certain the code would run the same as the starter code.

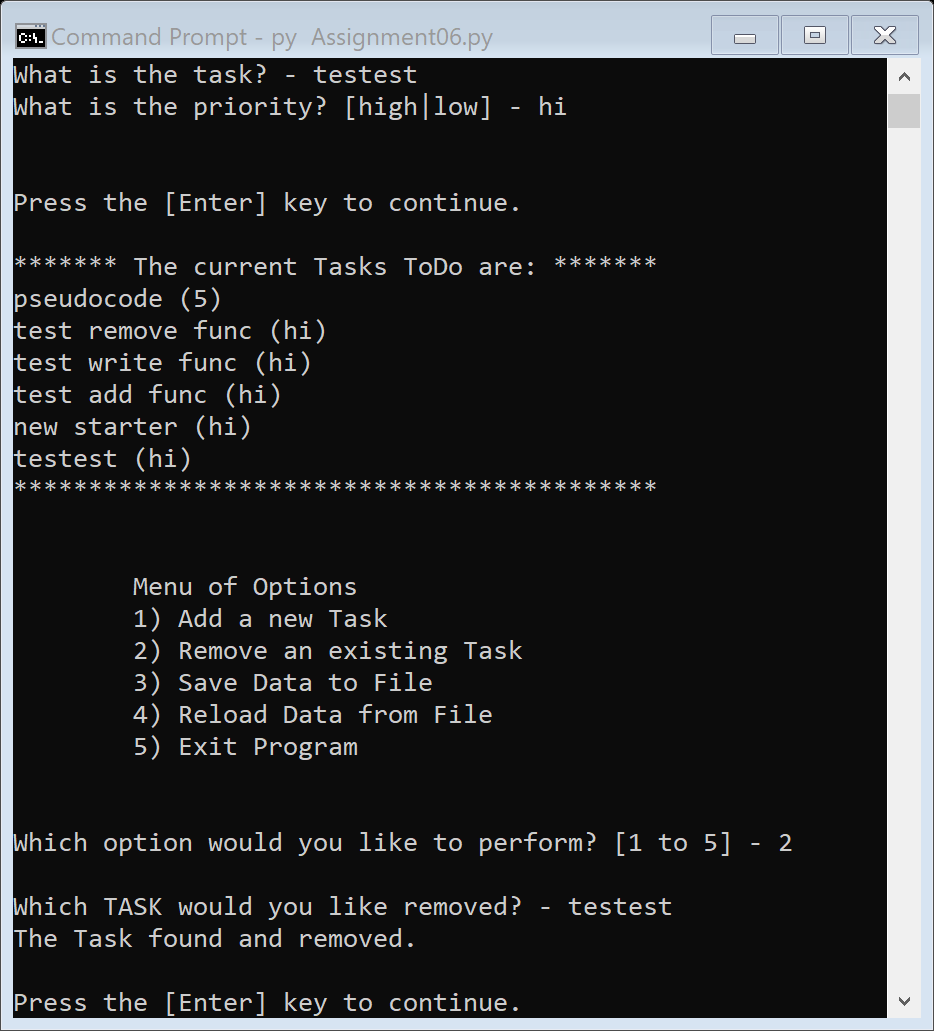


Figure : Script running in Command prompt to add data

Testing the script in the Command prompt gave the same results as well.

Now that I have made the required changes to the code, I proceeded to create a landing page for this and our remaining python assignments in GitHub. Took a little searching only to find the right help but was worth it to use their app then coding from scratch onto another server.

Summary

This time around, we were to put into practice creating functions out of existing code without effecting the front end of the application. By writing the processing statements in functions makes reading through the main code easier and just looks better. Also, having the processes separated into different functions allows for easier debugging and, as learned in class, adjustments/editing. Overall, another opportunity to grow my understanding of python, its similarities to other coding languages and learning more about GitHub’s plethora of uses.