Shruthi Srivatsan

November 17, 2020

Foundations of Programming- Python

Assignment 05

Creating a Task List Script with Multiple Options for User in Python

Introduction:

In this paper I will be discussing how to write a script that will import data from an existing text file and loading it into GitHub. This paper will also go over how to view the current data in the file, add more data, delete repeated data or save and exit. The new input will be programmed as dictionaries.

Declaring Variables and Starting Script:

First I used the python assignment 5 starter.py file to use the lay out and notes that was provided for this assignment. Using that we were asked to fill out the portions that were labeled “#TODO-add code here. Before creating the table of options, I was asked to write code that takes a file named “todolist.txt” and load the data from a text file and place it into a table.

First I had to declare the variables that would be used throughout this program. This would allow us to follow through with the script, everything is defined and program will be able to simply do the functions that are asked of it. I defined the following variables, shown in Figure 1.

Text

Description automatically generated

Figure 1: Variables Defined

Then I started to write the code. I started with “try” which asks the computer to load the file and then run through a loop. So because strFile is defined, my next line of code was strFile=open(objFile, “r”) which says that the text file that was defined as objFile will be read-“r”. Then the next code was for row in strFile which serves as a loop is entered. Then I entered the strData=row.split(“,”) which should return a list for you. The next line of dicRow={“Task”:strData[0], “Priority”strData[1].strip()} is used to process the data just like the exercise in module 5- listing 8. For each line the data will be read as a string and converted to dictionary to: “lsttable”. If the text is not found then it will mention that it was not found. This script is showing in Figure 2.

Text

Description automatically generated

Figure 2. Loading file from computer

Step 1

This section starts off with the while(True) statement which will start the loop. The user will be presented a menu that asks them to enter a number between 1 and 5 (Figure 3) as defined as strChoice.

Text

Description automatically generated

Figure 3. Code that prompts user to enter option from Menu

If user selects 1- it needs to show the current data. For that I wrote an if statement that says if strchoice.strip() == 1 then to print (lsttable). This will print the data pulled from the try statement in the first part of the code (Figure 4). Continue is added at the end of this code to allow this code to continue another loop and prompt user for next entry



Figure 4. Code to display current data

Step 2

User can enter “2” and it will have the user enter a new task and priority that will append the lsttable and store into the todolist.txt file that has been imported. If there is no txt file then this would be empty but because you are adding on to the data now, it should have created in the variable declaration step. After this I included a “continue” to continue the loop and prompt menu options to user again (Figure 5).

Text

Description automatically generated

Figure 5. Adding new item to list/table

Step 3

In this step if the user enters 3, it will allow the user to delete a row of data if the data matches with another entry. So looking at Figure 6, you can see that this code will ask the user if they would like to remove. Then it has a for row in lsttable and continued by if row [“Task”] == strItem ; lstTable.remove(row). So what this will do is when the user enters the name of the Task that they want to delete exactly the way it is entered in the txt.file, it will go in and remove that from the table (Figure 6).

Text

Description automatically generated

Figure 6. Removing item from list/table

Step 4

In this step, If the user selects option4, it will allow them to save the data. It will first ask “would you like to save your data?” Then it will require the user to input y for yes or n for no. If user enters y for yes then it will open the objfile and write whatever has been entered for Task and Priority and copy the table to the txt file and save it. Once saved it will print “Data is saved to file”(Figure 7).

Text

Description automatically generated

Figure 7. Save tasks to todolist.txt

Step 5:

This step allows user to exit the program when user enters 5. It will then print Thank you for using this program and break which will end the program. This is shown in Figure 8.

Graphical user interface

Description automatically generated

You can run this code in two ways: on Pycharm (Figure 9) and on Terminal (Figure 10). Once you run this your output file should look like Figure 11. Text

Description automatically generated

Figure 9 Running script in pycharm

Text

Description automatically generated

Figure 10 Running script in Terminal

Graphical user interface, text

Description automatically generated

Figure 11. Text output file

Conclusion: In this paper I learned how to open a file and discussed how to write a script that will import data from an existing text file. I also was able to provide the user with options to choose from to edit the list or remove items from the list and then save the file. Overall I learned all this doing the chapter readings and watching the module videos.