Jason Johanneck

2022-May-17

IT FDN Programming 110

Module05 Assignment05

FDN Programming 110 Assignment 045

# Introduction

Assignment05 asks us to write a python script that will load data from a text file and the provide the user with a list of options to either view the data, update the data or remove a data element. The data input from the user is to be stored as a list where the individual rows are dictionary items. When saved out to disk, the data is converted to string/text and appended to the contents of the file.

# Building the script.

The script uses a prebuilt python starter file where we are to complete the TODO sections. Here is my code listing from NotePad++.

Figure1 - Assignment 05 Python script Listing

Text

Description automatically generated

# Step1: Loading data from the text file:

In this section of code, I load the data from the ToDoList.txt file into a list table, then loop through the table row by row to display the task list. The code snippet for this is in Figure 2 below.

Figure - Loading and Displaying the ToDoList.txt file.

Text

Description automatically generated

# Step2: Display a Menu of Choices:

This was prebuilt for us:

Figure - Prebuilt code to display the Menu.

Text

Description automatically generated

# Step3: Display data the user has entered.

Here I am looping through the list table which stores the user entered input. For each row, the task and rank/priority are displayed.

Figure - Displaying User entered text

Text

Description automatically generated

# Step4: Code block to display data the user has entered.

As noted in Figure 5, I am prompting the users for the task and priority/rank, using one input statement. The idea is the user should enter the task and rank separated by a comma. The data is then loaded into a dictionary and added as a row into a list table held in memory.

Figure - Prompting the User for the Task and Priority values.

Text

Description automatically generated

# Step5: Removing a task from the current user entered list.

As listed in the below code snippet, here I prompt the user for the task to be removed, then search the list table for that task. A message is then displayed on the success/failure of this operation.

Figure - Removing a Task

A screenshot of a computer

Description automatically generated with medium confidence

# Step6: Saving user entered data to disk:

As listed in the below, in this code block the data stored by the user is written to disk. The data is appended to the end of the file. After the data is written to disk, the list table which holds user entered tasks is cleared to ensure it is not appended multiple times during subsequent requests to save the data.

Figure - Writing the current task list to file.

Text

Description automatically generated

# Running the script.

To run the code, I chose to use the Spyder application rather than Pycharm as I was having issues with the UI display in Pycharm.

# From the Spyder interface:

# Here is the Initial Display after pressing the run button:

Text

Description automatically generated

Here I select the option to add several tasks/priorities, then I display my current list:

Text

Description automatically generated

Here I remove the cleaning task I just entered, and the result of that action is displayed to the user:  
  
Text

Description automatically generated

Finally, I write the contents to disk:  
Text

Description automatically generated

The contents of the file after this operation is listed in Figure 8. The TV repair task was appended to the end of the file as expected.

Figure - Final contents of ToDoList.txt

Chart

Description automatically generated

# Runing the script from the Windows Command prompt:

The following screen print shows the execution of Assignment 05. The initial load and display of the file contents is shown here as well.

Text

Description automatically generated

The below screen print shows the addition of two tasks:

Text

Description automatically generated

Here I am displaying the two tasks I just entered:  
  
Text

Description automatically generated

Here I am removing the task to Read Python book:

Text

Description automatically generated

I display the current task list once more to show that the item was indeed removed:  
  
Text

Description automatically generated

Finally, I save my current task list to disk using menu option 4:

Text

Description automatically generated

The contents of the file now has the task to Watch News which was appended to the end of the file as shown in Figure 9:

Figure - Final Contents of ToDoList.txt

Graphical user interface, text, application

Description automatically generated

# Summary

For Assignment 5 we learned the following:

* How to build dictionary’s and store them as a row in a list.
* How to find a dictionary item in a list row.
* Further file I/O processing using read and write operations.
* We gained additional exposure to displaying and process a user menu of options.