Mike Steinauer

8/07/2022

Foundations of Programming: Python

Module 5

Here is my GitHub link for Assignment05

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

**Python Script Creation**

## **Intro**

This will show the steps I took to create a python script which adds to a to do list and saves or removes those additions.

## Step 1 – When the program starts, load any date you have.

Loading data As seen in (Figure1.1)

Text

Description automatically generated

Text

Description automatically generated

Figure 1.1 Loading data from the file into memory

## Step 2 – Display a menu of choices to the user.

Following along with the instructions we are displaying the menu as seen in Figure 1.2

Text

Description automatically generated

Text

Description automatically generated

Figure 1.2 Menu of choices

## Step 3 – Show the current Items in the Table

With data in memory, we can show the current data from out file. As seen in (Figure 1.3)

Text

Description automatically generated

Text

Description automatically generated

Figure 1.3 Display the current data to the user

## Step 4 – Add a new task to the List/Table

We can add a new task and it’s priority as seen in (Figure 1.4)

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Figure 1.4 Add a new Task / Priority

## Step 5 – Remove the last task added.

If the user would like, he can remove the newly added item from the list as seen if Figure 1.5

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Figure 1.5 Remove newly added item.

## Step 6 – Save Data to the file.

The next option – 4 – is the ability to save to the text file as shown in Figure 1.6

Text

Description automatically generated

Text

Description automatically generated

Figure 1.6 Saving to your file.

## Step 7 – Exit the Program

Final option will be to exit the program as seen in Figure 1.7.

Text

Description automatically generated

Text

Description automatically generated

Figure 1.7 Exit the program

## Summary

In conclusion we were able to build a program that takes a file, displays it, adds to it or removes data from it, saves data to it and exits the program. Windows CLI output can be seen in Figure 1.8

Text

Description automatically generated

Figure 1.8 Windows CLI output.

# ------------------------------------------------------------------------ #  
# Title: Assignment 05  
# Description: Working with Dictionaries and Files  
# When the program starts, load each "row" of data  
# in "ToDoToDoList.txt" into a python Dictionary.  
# Add each dictionary "row" to a python list "table"  
# ChangeLog (Who,When,What):  
# RRoot,1.1.2030,Created started script  
# MSteinauer,8.11.2022,Added code to complete assignment 5  
# ------------------------------------------------------------------------ #  
  
# -- Data -- #  
# declare variables and constants  
objFileName = "ToDoList.txt" # An object that represents a file  
strData = "" # A row of text data from the file  
dicRow = {} # A row of data separated into elements of a dictionary {Task,Priority}  
lstTable = [] # A list that acts as a 'table' of rows  
strMenu = "" # A menu of user options  
strChoice = "" # A Capture the user option selection  
objFile = None # An object that represents a file  
  
# -- Processing -- #  
# Step 1 - When the program starts, load any data you have  
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)  
  
# File to List  
  
objFile = open(objFileName, "r")  
for row in objFile:  
 lstRow = row.split(",") # Returns a list!  
 dicRow = {"Priority": lstRow[0], "Task": lstRow[1].strip()}  
 lstTable.append(dicRow)  
 print(dicRow)  
objFile.close()  
  
# -- Input/Output -- #  
# Step 2 - Display a menu of choices to the user  
  
while (True):  
 print("""   
 Menu of Options   
 1) Show current data   
 2) Add a new Task.   
 3) Remove the last task added.   
 4) Save Data to File   
 5) Exit Program   
 """)  
 strChoice = str(input("Which option would you like to perform? [1 to 5]: "))  
 print() # adding a new line for looks  
  
 # Step 3 - Show the current items in the table  
  
 if (strChoice.strip() == '1'):  
 print("\*\*\*\*\*\*\* The current items ToDo are: \*\*\*\*\*\*\*")  
 for row in lstTable:  
 print(row["Priority"] + " " + row["Task"])  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") # Step 4 - Add a new item to the list/Table  
 continue  
  
 # Step 4 - Add a new task and it's priority to the list/Table  
  
 elif (strChoice.strip() == '2'):  
 print("The priority of the task and the task you would like to add to your list")  
 strTask = input("Task to do: ")  
 strPriority = input("Priority of the task: ")  
 dicRow = {"Task": strTask, "Priority": strPriority}  
 lstTable += [dicRow]  
 continue  
  
 # Step 5 - Remove a new item from the list/Table  
  
 elif (strChoice.strip() == '3'):  
 lstTable.remove(dicRow)  
 continue  
  
 # Step 6 - Save tasks to the ToDoList.txt file  
  
 elif (strChoice.strip() == '4'):  
 objFile = open(objFileName, "w")  
 for dicRow in lstTable:  
 objFile.write(dicRow["Priority"] + "," + dicRow["Task"] + "\n")  
 objFile.close()  
 continue  
   
 # Step 7 - Exit program  
  
 elif (strChoice.strip() == '5'):  
 print("Exit the program.")  
 break #Exit the program