Patrick Chuoy

August 9, 2023

Foundations of Programming: Python

Assignment 05

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

To Do List

# Introduction

In Module 05, I learned about loading list data from a text file, dictionaries, and GitHub. I applied this knowledge to create the To Do List script. Assignment 05 provided a starter file that I modified to make work. The script can read, add, and delete task data from a text file. The data is stored in a two-dimensional list with dictionaries that have a task and priority value. After finishing the script, I made a public GitHub repository and posted my files there.

## Loading Data from a Text File

When the program starts, it needs to load any data in the ToDoList.txt file. The data in the file should be rows with a string that is two words separated by a comma (task,priority). The program makes the rows into dictionaries and then adds it to a list table of dictionary rows.

To do this, I made a for loop that looped through each row in the file. Then, I needed to extract the task and priority from each row. I learned that you can use the split() function to separate elements. A row in the text file has “task,priority” so row.split(“,”) would separate the task and priority because of the comma in the middle. Then the program makes a dictionary with task and priority for keys, and the strings from the row for values. After that, the dictionary is added to the two-dimensional list table.

# Dictionaries

This was my first time working with dictionaries. They are similar to lists. Having keys instead of indexes of numbers was easy to understand. Adding, removing, and saving dictionaries only differed slightly from using lists.

# Future

Module 05 also introduced functions and error handling (try-except). Functions let you group statements that can be called later in the program. Error handling lets you manage potential errors and provide a more user-friendly error message. Assignment 05 said to not use functions because they will be used in the next assignment. Module 05 also stated that try-except will be covered in Module 06. I could have implemented error handling in this assignment but didn’t have time to fully understand and utilize it.

# Difficulties

If I had more time, I would have liked to do some of the options differently. Seeing examples in the textbook, I wanted to make it so the program would check for duplicates before adding an item. I could have used a flag like I did for removing items but decided to allow the user to be able to add tasks with the same name.

Instead of having the user type the task they wanted to remove, I thought about listing and numbering the existing items and then having the user select a number.

# Summary

The To Do List script utilized loading data from a text file and using dictionaries. It was an introduction to Separation of Concerns and organizing your code. I was able to get the script to work, but it’s not as refined as I would like. I want to learn more and hopefully be able to do better on the next assignment.

# ------------------------------------------------------------------------ #  
# 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  
# PChuoy,8.2.2023,Added code to complete assignment 5  
# ------------------------------------------------------------------------ #  
  
# -- Data -- #  
# declare variables and constants  
strFile = "ToDoList.txt" # An object that represents a file  
objFile = None # An object for text 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  
strTask = "" # A string for user task  
strPriority = "" # A string for priority of task  
  
  
# -- 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)  
objFile = open(strFile, "r")  
# Loop through strings in text file  
for row in objFile:  
 # Split task and priority by comma and make into list with 2 elements  
 lstRow = row.split(",")  
 # Add list elements into a dictionary with values of task and priority  
 dicRow = {"Task": lstRow[0].strip(), "Priority": lstRow[1].strip()}  
 lstTable.append(dicRow) # Add dictionary into current list table  
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 item.  
 3) Remove an existing item.  
 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("Task | Priority\n")  
 # Loop through each dictionary in current list table  
 for item in lstTable:  
 # Get the task and priority of the dictionary and print them  
 print(item["Task"] + " | " + item["Priority"])  
  
 continue  
  
 # Step 4 - Add a new item to the list/Table  
 elif (strChoice.strip() == '2'):  
 # While loop for if user wants to add multiple items  
 while (True):  
 strTask = str(input("What task do you want to add? ")).strip()  
 strPriority = str(input("What is the priority of the task? (High|Medium|Low) ")).strip()  
 lstTable.append({"Task": strTask, "Priority": strPriority})  
  
 # Stop adding items if user does not enter 'y'  
 strChoice = input("Continue adding tasks? ('y/n') ")  
 if strChoice.lower() != 'y':  
 break  
  
 continue  
  
 # Step 5 - Remove a new item from the list/Table  
 elif (strChoice.strip() == '3'):  
 # While loop for if user wants to delete multiple items  
 while (True):  
 blnFoundFlag = False # Flag for dealing with duplicates  
 strTask = input("What task do you want to delete? ")  
 # Loop through each dictionary in current list table  
 for item in lstTable:  
 # If task is in list table, delete  
 if item["Task"].lower() == strTask.lower():  
 lstTable.remove(item)  
 print("Task deleted")  
 blnFoundFlag = True  
  
 # Task is not found in list table  
 if not blnFoundFlag:  
 print("Task does not exist in the list")  
  
 # Stop deleting items if user does not enter 'y'  
 strChoice = input("Continue deleting tasks? ('y/n') ")  
 if strChoice.lower() != 'y':  
 break  
  
 continue  
  
 # Step 6 - Save tasks to the ToDoToDoList.txt file  
 elif (strChoice.strip() == '4'):  
 objFile = open(strFile, "w")  
 # Loop through each dictionary in current list  
 for row in lstTable:  
 # Write to file (task,priority)  
 objFile.write(str(row["Task"]) + "," + str(row["Priority"]) + "\n")  
 objFile.close()  
 print("Tasks saved to file.")  
  
 continue  
 # Step 7 - Exit program  
 elif (strChoice.strip() == '5'):  
 break # and Exit the program  
  
input("\nPress Enter to exit the program ")

Figure 1: My To Do List script

A screenshot of a computer test

Description automatically generatedFigure 2: Output in PyCharm exported to PDF

A screenshot of a computer screen

Description automatically generatedFigure 3: Output in CMDA screenshot of a computer

Description automatically generated

Figure 4: Text file after running script