Katrina Taylor

August 10, 2020

Foundations of Programming: Python

Assignment05

**Lists and Dictionaries**

# **Introduction**

The goal of this assignment is to create a new program that gets user input, loads data from an existing text file into dictionaries, and adds/removes/reads dictionaries from a list.

# **Text File**

First, I had to create the text file my program would load titled ToDoList.txt (Figure 1).

A picture containing screenshot, drawing

Description automatically generated

***Figure 1: ToDoList.txt text file***

# **Python** **Script**

Next, I started working on the Python Script that would carry out the program. All Python Scripts should start with a comment known as a header. This header should include the title of the script, the description, and a change log of who implemented the change, when it happened, and what was the change (Figure 2).

***A screenshot of a cell phone

Description automatically generated***

***Figure 2: The header I used in my assignment***

This particular program then required use of a starter script (Figure 3).

# -- Data -- #  
# declare variables and constants  
objFile = "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  
  
  
# -- Processing -- #  
# Step 1 - When the program starts, load the any data you have  
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)  
# TODO: Add Code Here  
objOpenFile = open(objFile, "r")  
for row in objOpenFile:  
 lstRow = row.split(",")  
 dicRow = {"Task": lstRow[0]. "Priority": lstRow[1].strip()}  
 lstTable.append(dicRow)  
objOpenFile.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'):  
 # TODO: Add Code Here  
 continue  
 # Step 4 - Add a new item to the list/Table  
 elif (strChoice.strip() == '2'):  
 # TODO: Add Code Here  
 continue  
 # Step 5 - Remove a new item from the list/Table  
 elif (strChoice.strip() == '3'):  
 # TODO: Add Code Here  
 continue  
 # Step 6 - Save tasks to the ToDoToDoList.txt file  
 elif (strChoice.strip() == '4'):  
 # TODO: Add Code Here  
 continue  
 # Step 7 - Exit program  
 elif (strChoice.strip() == '5'):  
 # TODO: Add Code Here  
 break # and Exit the program

***Figure 3: The starter script***

I then added my code to the starter script (Figure 4). I used a for loop to iterate through the dictionaries in my list table. I used the input() function for the end user to indicate what tasks they wanted to add or remove. I used the remove() function to remove tasks, the append() function to add tasks, the print() function to read the text file, and the close() function to save any edits to the text file.

# -- Data -- #  
# declare variables and constants  
objFile = "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  
strTask = "" # A task that the user provides  
strPriority = "" # A priority that the user provides  
  
  
# -- Processing -- #  
# Step 1 - When the program starts, load the data you have  
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)  
# TODO: Add Code Here  
objOpenFile = open(objFile, "r")  
for row in objOpenFile:  
 lstRow = row.split(",")  
 dicRow = {"Task": lstRow[0], "Priority": lstRow[1].strip()}  
 lstTable.append(dicRow)  
objOpenFile.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'):  
 # TODO: Add Code Here  
 for dicRow in lstTable:  
 print(dicRow)  
 continue  
 # Step 4 - Add a new item to the list/Table  
 elif (strChoice.strip() == '2'):  
 strTask = input("Enter a task you'd like to add: ")  
 strPriority = input("Enter it's priority: ")  
 dicRow = {"Task": strTask, "Priority": strPriority}  
 lstTable.append(dicRow)  
 continue  
 # Step 5 - Remove a new item from the list/Table  
 elif (strChoice.strip() == '3'):  
 strTask = input("Enter a task you'd like to remove: ")  
 for dicRow in lstTable:  
 if strTask == dicRow["Task"]:  
 lstTable.remove(dicRow)  
 continue  
 # Step 6 - Save tasks to the ToDoToDoList.txt file  
 elif (strChoice.strip() == '4'):  
 objOpenFile = open(objFile, "w")  
 for dicRow in lstTable:  
 objOpenFile.write(dicRow["Task"] + "," + dicRow["Priority"] + "\n")  
 objOpenFile.close()  
 continue  
 # Step 7 - Exit program  
 elif (strChoice.strip() == '5'):  
 print("Exiting program")  
 break # and Exit the program

***Figure 4: My code entered in the areas that showed “Add code here” in the starter script***

# **Script Output**

When the script is ran, the end user will not see the header comments. The end user will only see the statements printed from the print() and input() functions, in addition to whatever they type when they interact with the program. The script was successful at accomplishing the goal of this assignment (Figure 5 and Figure 6).

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

Which option would you like to perform? [1 to 5] - 1

{'Task': 'Do homework', 'Priority': 'high'}

{'Task': 'Shop online', 'Priority': 'low'}

{'Task': 'Call my parents', 'Priority': 'medium'}

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

Which option would you like to perform? [1 to 5] - 2

Enter a task you'd like to add: Vacuum

Enter it's priority: medium

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

Which option would you like to perform? [1 to 5] - 1

{'Task': 'Do homework', 'Priority': 'high'}

{'Task': 'Shop online', 'Priority': 'low'}

{'Task': 'Call my parents', 'Priority': 'medium'}

{'Task': 'Vacuum', 'Priority': 'medium'}

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

Which option would you like to perform? [1 to 5] - 3

Enter a task you'd like to remove: Vacuum

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

Which option would you like to perform? [1 to 5] - 1

{'Task': 'Do homework', 'Priority': 'high'}

{'Task': 'Shop online', 'Priority': 'low'}

{'Task': 'Call my parents', 'Priority': 'medium'}

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

Which option would you like to perform? [1 to 5] - 4

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

Which option would you like to perform? [1 to 5] - 5

Exiting program

***Figure 5: The output in PyCharm after running and interacting with my program***

***A screenshot of a cell phone

Description automatically generated***

***A screenshot of a cell phone

Description automatically generated***

***Figure 6: The output in Terminal after running and interacting with my program***

The saved file now contains the input the user provided (Figure 7).

A picture containing screenshot, drawing

Description automatically generated

***Figure 7: The text file shows the saved data***

# **Summary**

In this assignment I created a program that gets user input, loads data into dictionaries, and adds/removes/reads dictionaries from a list.