Dania Mahmood

Feb. 16, 2021

Foundations of Programming, Python

Assignment 5: Create script that demonstrates using lists and dictionaries to read, write, remove

and save to file.

GitHub Link:

Modifying a text file of Tasks and Priorities

# Introduction

Lists are used to hold collection of data and include a lot of built-in functions. You can use a list to load data from file. This is very important if you need to work with data in memory before saving it on to the disk. In this assignment, I have used a list that I named “lst” to load data from ToDo.txt file. I use the list to then display it contents, write to it and remove items from it. When the user selects to save the file, I then save the content of the list to ToDo.txt.

I have used dictionaries in my assignment to help me easily access elements of the List. Dictionaries, are very similar to List, Tuple or Strings but instead of using numeric indexes to access elements they use character keys. You can define a dictionary by using {}, example below.

DataDic = {}

Like lists, dictionaries support a lot of built in functions like items(), values() and keys(). I have used the values() function to compare user input string to the value stored in the list/.

# Script

As usual, I started my script by adding the script header. You can see in the changelog what and when I added my changes.

I defined my variable in the beginning and added their usage as in-line comments. I used the while loop to keep giving users the option to choose the next step until they choose to exit. But before I run the while loop, I copied the content of the ToDo.txt file to a list that I can modify in the loop.

Based on their choice I used the IF statement to make their input to the option and then execute only that section of the code.

In this script, users can pick options 1 to 5. If they pick 1, I run the section of the code that display the current data from the list.

**if** strUsrOpt == 1:  
 **for** row **in** lst:  
 print(row[**"Task"**] + **","** + row[**"Priority"**])

If they picked 2, I run the section of the code that writes the data and appends it to the list.

**elif** strUsrOpt == 2:  
 strTask = input(**"Enter the task: "**)  
 strPri = input(**"Enter task priority: "**)  
 lstRow = {**"Task"**: strTask, **"Priority"**: strPri}  
 lst.append(lstRow)

If they pick 3, then I execute the section of the script that lets user enter the Task they want to remove from the list.

**elif** strUsrOpt == 3:  
 strTaskRm = str(input(**"Enter task name you want to remove? "**))  
 lst\_len = len(lst)  
  
 i = 0  
 found = **False  
 while** (i < lst\_len):  
 **if** strTaskRm **in** lst[i].values():  
 **del** lst[i]  
 print(strTaskRm + **" has been removed!"**)  
 found = **True  
 break** i = i + 1  
  
 **if** found == **False**:  
 print(**"Task not found!"**)

If they pick 4, then I execute the section of the script that lets user save their modification to ToDo.txt file.

**elif** strUsrOpt == 4:  
 objFile = open(**"ToDo.txt"**, **"w"**)  
 **for** dicRow **in** lst:  
 objFile.write(dicRow[**"Task"**] + **","** + dicRow[**"Priority"**] + **"\n"**)  
 objFile.close()  
 print(**"Your data has been saved!"**)

If they pick 5, I simply exit the script.

**elif** strUsrOpt == 5:  
 exit()

Below is a snapshot of my script,

*############################################################################################################  
# Title: Assignment 05  
# Description: Working with Dictionaries and Files  
# To display user a menu of options to choose from. Their choices are  
# Reading a file, writing to a file, removing from file,  
# saving to file and lastly exit the program.  
#  
# ChangeLog (Who,When,What)  
# DaniaM,02.14.2021,Started writing script, defined menu and executed read, write and exit options  
# DaniaM,02.15.2021,Executed the save and remove  
#############################################################################################################  
  
# Declare variables*dicRow = {} *# Dictionary definition*strTaskRm = **""** *# Enter to be removed*strUsrOpt = **""** *# To store option from the menu picked by user*lstRow = [] *# List of items in a row from file*lst = [] *# List of task and priorities*strMenu = **"Menu of Options" "\n" "1) Display current data" "\n" "2) Add Data to List" "\n" "3) Remove an item" "\n" "4) "** \  
 **"Save data to file" "\n" "5) Exit Program "***# Copy content of file to a List*objFile = open(**"ToDo.txt"**, **"r"**)  
**for** row **in** objFile:  
 lstRow = row.split(**","**)  
 dicRow = {**"Task"**: lstRow[0], **"Priority"**: lstRow[1]}  
 lst.append(dicRow)  
objFile.close()  
  
*# Execute script depending on user input***while** (**True**):  
 print(strMenu)  
 strUsrOpt = int(input(**"Which option would you like to perform? [1 to 5] "**))  
 **if** strUsrOpt == 1:  
 **for** row **in** lst:  
 print(row[**"Task"**] + **","** + row[**"Priority"**])  
  
 **elif** strUsrOpt == 2:  
 strTask = input(**"Enter the task: "**)  
 strPri = input(**"Enter task priority: "**)  
 lstRow = {**"Task"**: strTask, **"Priority"**: strPri}  
 lst.append(lstRow)  
  
  
 **elif** strUsrOpt == 3:  
 strTaskRm = str(input(**"Enter task name you want to remove? "**))  
 lst\_len = len(lst)  
  
 i = 0  
 found = **False  
 while** (i < lst\_len):  
 **if** strTaskRm **in** lst[i].values():  
 **del** lst[i]  
 print(strTaskRm + **" has been removed!"**)  
 found = **True  
 break** i = i + 1  
  
 **if** found == **False**:  
 print(**"Task not found!"**)  
  
 **elif** strUsrOpt == 4:  
 objFile = open(**"ToDo.txt"**, **"w"**)  
 **for** dicRow **in** lst:  
 objFile.write(dicRow[**"Task"**] + **","** + dicRow[**"Priority"**] + **"\n"**)  
 objFile.close()  
 print(**"Your data has been saved!"**)  
  
 **elif** strUsrOpt == 5:  
 exit()

# Summary

Lists and Dictionaries are same in regard to storing collection of data that can vary in data type. They both have a lot of useful built-in functions. Users can access elements of a list by using numeric indexes while elements of dictionaries can be accessed via character keys that makes it easier to understand what elements are being accessed.

Both List and Dictionaries let users load and modify data in memory before saving it to the hard drive.