Mellony Hoskinson

May 11, 2022

Foundations of Programming: Python

Assignment 05

Lists and Dictionaries

# Introduction

In this module, we learn about modifying a script that uses lists and dictionaries. This document will act as a standard operating procedure for modifying a new script that manages a "ToDo list." The "ToDo" file will contain two columns of data, "Task" and "Priority." The user will enter a selection from a menu, add a new item to the list, remove it, save the text file and exit the program. In addition, we are also now able to answer a number of questions related to lists and dictionaries, separation of concerns, script templates, and GitHub.

# Methods

1. Declare variables

# -- 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  
strFile = "ToDoFile.txt"

1. Load data to a text file called ToDoList.txt

# -- 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)  
dicRow1 = {"Task": "1", "Priority": "High"}  
dicRow2 = {"Task": "2", "Priority": "Low"}  
lstTable = [dicRow1, dicRow2]  
objFile = open(strFile, "r")  
for row in objFile:  
 dicRow = {"Task": "1", "Priority": "1"}  
 print(dicRow)  
 print(dicRow["Task"] + ',' + dicRow["Priority"])  
 lstTable.append(dicRow)  
 objFile.close()

1. Display a menu of options and ask the user to enter a choice.
2. # -- 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(lstTable)  
    continue

4. Add a new item to the table.

# Step 4 - Add a new item to the list/Table  
elif (strChoice.strip() == '2'):  
 strTask = input("Enter a Task: ")  
 strPriority = input("Enter a Priority of High, Medium, or Low: ")  
 dicRow = {"Task": "1", "Priority": "1"}  
 lstTable.append(dicRow)  
 for lstRow in lstTable:  
 print(lstRow)  
 continue

1. Remove the item.

# Step 5 - Remove a new item from the list/Table  
elif (strChoice.strip() == '3'):  
 del dicRow[1]  
 continue

1. Save the tasks to the ToDoList.txt text file.
2. # Step 6 - Save tasks to the ToDoToDoList.txt file  
   elif (strChoice.strip() == '4'):  
    # *TODO: Add Code Here* continue

7. Exit the program

Questio # Step 7 - Exit program  
elif (strChoice.strip() == '5'):  
 break # and Exit the program

# Questions

1. What is the difference between a List and a Dictionary?
   1. A list holds a collection of objects and include built in functions that you can use to work with those objects. For lists, you use square brackets.
2. What is the between an Index and a Key?
   1. The subscript of a key is character data. The subscript of an index is numeric data.
3. How do you read data from a file into a List?
   1. Data in a file must be loaded from the hard drive into memory if the user wants to continue using stored data using the read option. objFile = open(strFile, “r”)
4. How do you read data from a file into a Dictionary?
   1. Declare variables, open file and read the data, split into a list, put in the keys and extract out the values from the list. You need to pull the data from the file into a list then create a dictionary from the list.
5. What is the programming pattern called “Separations of Concerns?”
   1. Separation of Concerns is a design principle for separating a computer program into distinct sections, so that each section addresses a separate concern. Three primary sections are data, processing, and presentation (or input and output).
6. How would you use a function to organize your code?
   1. A function is a named set of one or more statements and you call that function to run. Functions allow you to group a set of programming statements and later reference them by name. A function must be defined in a script before they can be called.

# -- Data -- #  
fltN1 = 0.0 # 1  
fltN2 = 0.0 # 2

# -- Processing -- #  
**def DivideValues**(): # 6, this code loads but does not run yet!  
 **return** (fltN1 / fltN2) # 7  
  
# -- Presentation (I/O) -- #  
fltN1 = float(input(**"Enter the first number: "**)) # 3  
fltN2 = float(input(**"Enter the second number: "**)) # 4  
print(**DivideValues**()) # 5  
print(**"Done"**) # 8

1. Why is a script template useful?
   1. A script template is useful for consistency which makes it easier for you and others reading your code.

# ------------------------------------------------- #

# Title: <Type the name of the script here>

# Description: <Type a description of the script>

# ChangeLog: (Who, When, What)

# <Example Dev,01/01/2030,Created Script>

# ------------------------------------------------- #

# -- Data -- #

# -- Processing -- #

# -- Presentation (I/O) -- #

1. Why is error handling using Try-Except recommend?
   1. You want to be able to control what the user sees. Error messages for users sometimes don’t make sense because the code was written for the developer, not the end user. It is nice for error messages for the end user be easier to interpret and understand.
2. What is GitHub, and why is it used?
   1. GitHub is a code share website where you can create folders to post your files and code to GitHub for others to download and use. You can also give and receive feedback from others on your code.

# Summary

In this module, we learned about modifying a script that uses lists and dictionaries. We outlined the steps and code for modifying a script that manages a "ToDo list." We then had the user enter a selection from a menu, add a new item to the list, remove it, save the text file, and then exit the program. In addition, we are also now able to answer a number of questions related to lists and dictionaries, separation of concerns, script templates, and GitHub.