Emmanuel Twum

Foundations of Programming Python

Randal Root

Assignment 05

10/29/2017

Reading and Writing Dictionary Items To/From a Text File

The purpose of this paper is to describe how to create a python program which will read data from a text file and display it as a dictionary, add new items to the dictionary, remove existing items from the dictionary, and save data back to the text file. The program code is broken up into a section for data, processing, input/output.

The first section of the code is the data section (See Figure 1). The purpose of the data section is to define the variables and/or constants used in the program. The first variable defined is the variable to open the text file in read mode (See Figure 1). The next variables are for reading the existing two lines in the text file (See Figure 1). The last variables defined in the code are for taking the rows in the text file then converting them to dictionary items (See Figure 1). The dictionary items are then made into rows in a table.

**Figure 1:** Data

#\*\*\*\*Data\*\*\*\*\*

#open the text file in read mode

file = open("C:\\_PythonClass\ToDo.txt","r+")

#read the first two lines

strData = file.readline()

strData1 = file.readline()

#convert the existing two lines in the text file to two dictionary rows in a table

dicRow1 = {"Task":strData.split(",")[0],"Priority":strData.split(",")[1]}

dicRow2 = {"Task":strData1.split(",")[0],"Priority":strData1.split(",")[1]}

tblData = [dicRow1,dicRow2]

The next part of the code is the processing data. This code is broken out into four different functions. The first function, as shown in Figure 2, takes the tblData variable and shows what data is currently in it.

**Figure 2:** Show Current Table Data

#show current table data

def ShowCurrentData():

print(tblData)

The next function is for adding a new task/priority to the table tblData (see Figure 3). The function gets user input for the task and the priority which are then converted to a dictionary row. The new row is then appended to the existing data in the table.

**Figure 3:** Add Task to Table

#add task to the table

def AddItem():

while(True):

#get user input for task and priority

strTask = input("Please enter a task: ")

strPriority = input("Please enter the task's priority (low, medium, high): ")

#convert users inputs into a dictionary row

dicNewRow = {"Task":strTask.capitalize(), "Priority":strPriority.lower()+"\n"}

#append current table with new row

tblData.append(dicNewRow)

userinput = input("Would you like to continue inputting items? (Y/N) ")

if(userinput.lower() == 'y'):

continue

else:

break

The third function of the program is used to remove items from the table tblData based on user input (see Figure 4). The user inputs the task he or she would like to remove then the program lookups up the value in the task column of the dictionary rows. The task is then removed if it’s found in a row.

**Figure 4:** Remove Item from Table

#Remove item from the table

def RemoveItem():

#get user input for removing a task

strRemoveTask = input("Input the task you would like to remove: ")

#look up user input in each row in the task column; delete row if found

for x in tblData:

if(strRemoveTask.capitalize() == x ["Task"]):

x.clear()

print("Task Deleted")

The last function is for saving the latest task information back to the text file. The file is opened in write mode which overwrites the current text file with the new data. The values of the dictionary rows are then pulled from the table, converted to a list, and then saved to the text file as ‘task, priority’.

**Figure 5:** Save Data to File

#save latest data to the text file

def SaveData():

#overwrite current data in the file with new data

file = open("C:\\_PythonClass\ToDo.txt","w")

#pull values from the dictionary and turn them into a list; save to file

for x in tblData:

x.values()

this\_list = list(x.values())

file.write(this\_list[0] + "," + this\_list[1])

file.close()

The last part of the program is the input/output code (see Figure 6). The user inputs a number from the menu of options what he or she would like to do. The program then runs the function defined in the processing section of the code. This will continue to loop until the user inputs the number five to end the program.

**Figure 6:** Input/Output

#\*\*\*\*Input/Output\*\*\*\*

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? Input [1 to 5] - "))

print()#adding a new line

# Step 3 -Show the current items in the table

if (strChoice == '1'):

ShowCurrentData()

continue

# Step 4 - Add a new item to the list/Table

elif(strChoice == '2'):

AddItem()

continue

**Figure 6:** Input/Output (Continued)

# Step 5 - Remove a new item to the list/Table

elif(strChoice == '3'):

RemoveItem()

continue

# Step 6 - Save tasks to the ToDo.txt file

elif(strChoice == '4'):

SaveData()

continue

elif (strChoice == '5'):

break #and Exit the program