Key Murray

05/18/2022

**IT FDN 110 A Sp 22: Foundations of Scripting: Python**

**Assignment 05**

Creating a To Do List Script

**Introduction**

In this document, I will explain the steps I took to modify the To Do List script that we were asked to add code to for assignment 5. I will go over the resources I used to help me complete the to do sections of the script. It was a challenging assignment, as it required me to modify someone else’s script which seems easier than it was since there are many ways to produce the same outcome.

**Creating a To Do List Script**

It was very helpful that Randal provided pseudo code along with the script so we knew which parts of the script still needed work and what each section of code would need to do in each step of the script. I reviewed all the variables that were already declared as that information would be the most useful to understand before starting to write any part of the script.

**Step 1: Processing the Data**

I copied the listing 8 example in the programming notes of Assignment 5 to get me started with the processing data step in the script. The first line of code uses the variable “objFile” to open the “ToDoList.txt” in read mode. A for-loop, which will loop though each sequence in the text file is used on the following line. Next, the split method is used to split the list once the text file is read and then returns data as a dictionary with comma separators. The next line is defining the keys in the dictionary. The line after appends data from the dictionary row to the table. The last line closes the text file (Figure 1).

***Text

Description automatically generated***

***Figure 1: Screen shot of the data used for processing.***

**Step 2: Display a menu of choices to the user**

The code to display a menu of choices to the user did not need any modifications as Randal provided the code. The while true loop was used since that would be the portion of code used to continue running while the condition was still true. The print function was used along with triple quotes to indicate that a multi-line string would be printed. And then the strChoice variable is used to capture the user’s input for the option they would like to select to continue going through or exiting the program. The print function with an empty value is used to print a new line for looks (Figure2).

***Text

Description automatically generated***

***Figure 2: Screen shot of the code to display a menu of choices to the user.***

**Step 3: Show the current items in the table**

To show the current list of items in the table, the “if” statement is used along with the variable strChoice.strip() which captures the user’s input from the strChoice variable used in Step 2. The for-loop is used to execute the print function for the dictionaries in the table. If the user selects “1” then the “dicRow” in the “lstTable” will print the “dicRow” and then follow the continue statement to loop back to the while-loop and print the menu of options again.

**Text, application

Description automatically generated**

**Figure 3: Screen shot of the code for showing the current items in the table.**

**Step 4: Add a new item to the List/Table**

The next option allows the user to add new tasks to the text file. The “elif” statement allows for multiple expressions to be used throughout the program. It works exactly like the “if” statement but moves on to the next condition in the script if the user’s input data does not meet its assigned condition. Next the print function is used to print “Please type a ‘Task’ and ‘Priority’” and then prints to a new line. Two new variables are introduced to gather the task and priority input from the user to then add it to the table as dictionaries. The strip method is used for the input data from the user to ensure that there are no leading or trailing characters when appending the data to the table. The lower method is used to return the input data to the dictionary in lower case characters. The append method is used to append the data to the existing dictionary rows in the table. Once the data has been appended to the table the print function is used to print to the user “New task has been added”.

**Text

Description automatically generated**

***Figure 4: Screen shot of the code for adding a new item to the List/Table.***

**Step 5:** **Remove an item from the List/Table**

The user also has the option to remove tasks from the table. This is presented when the user inputs “3” as this meets the condition of the “elif “statement in this block via the “strChoice” variable. A new variable “strTask\_remove” is created for this block of code to get input from the user of what task they would like to remove from the list. The next line uses the for-loop statement to execute the following “if” statement from the list of elements of the dictionaries in the table. If there is an object in memory that matches the user’s input, then it is removed from the table by using the remove method. It then prints to the user “Task removed!”.

**Text

Description automatically generated*Figure 5: Screen shot of the code to remove tasks from the table.***

**Step 6:** **Save tasks to the ToDoList.txt file**

If the user has added data to the table, it has only been stored in memory and not saved to the text file. If the user wanted to save data to the text file, they would select option “4” and the “elif “ statement would execute for variable “strChoice” with value “4”. The variable “objFile” is used to open the “ToDoList.txt” in write mode. The for-loop is used to write the data to the dictionary in the table. The close method is used in the following lines to close the file when done. Then “Tasks have been saved!” is printed to the user. It then returns to the beginning of the while-loop because of the continue statement at the end.

**Text

Description automatically generated**

***Figure 6: Screen shot of the code to save the data to the text file.***

**Step 7:** **Exit program**

The last menu option the user can choose is to exit the program. The “elif” method is used for the last time to represent option 5 from the menu by passing the value to the “strChoice” variable. The next line prints “Exiting the program.” And then breaks from the while loop.

**Text

Description automatically generated**

**Figure 7: Screen shot of the code to exit the program.**

**Summary**

This was the hardest script I have created so far, and I learned a lot from running into issues with lists and dictionaries. Working with lists was easier to understand for me then having to work with dictionaries. Overall, I feel good about what I was able to accomplish now and know that the next assignment will call for me to fix the things I was not able to figure out in this assignment.

**Script running in PyCharm:**

**Option 1:**

**Text

Description automatically generated**

**Option 2:**

**Text

Description automatically generated**

**Option 3:**

**Text

Description automatically generated**

**Option 4:**

**Text

Description automatically generated**

**A picture containing text

Description automatically generated**

**Option 5:**

**Text

Description automatically generated**

**Script running in Terminal:**

**Option 1:**

**Text

Description automatically generated**

**Option 2:**

**Text

Description automatically generated**

**Option 3:**

**Text

Description automatically generated**

**Option 4:**

**Text

Description automatically generated**

**A picture containing rectangle

Description automatically generated**

**Option 5:**

****