T.J. Tharp

May 15th, 2023

IT FDN 110 A Sp 23: Foundations Of Programming: Python

Assignment 05

github repo: https://github.com/Tadlol/ToDoList

Using Python to Create a To Do List

Intro

The assignment this week was a challenge for me. We had to take code that another person stated and finish it to meet the requirements of the assignment. I had to determine what the original coder planning with the code they had so far, plan how I would finish out each section of code and then complete the code using techniques learned this week. Some of the techniques included lists, dictionaries and try, except statements to account for possible errors.

List and Dictionaries

The lists and dictionaries gave me a run for my money this week. I was able to import the .txt file into the dictionary rows and list table with little trouble but working with them to display to the user as well as adding and removing tasks caused me some headache. At first I thought I could get away with printing the list table directly but I found the results to be unsightly for the user. I decided to parse out the table information by dictionary row and print the key and value directly. This allowed me to remove curly braces and quotes from the printed text that the user would see. In order to accomplish this task I looped through the rows in the table and printed the rows one at a time in a key, value format. (Fig. 1)

```
# Step 3 - Show the current items in the table
if strChoice.strip() == '1':
    for dicRow in lstTable:
        print(dicRow["Task"] + "," + dicRow["Priority"])
    continue
```

Figure 1: Table loop

When it came to adding items to the list I was able to do so simply by taking two user inputs, turning them into a dictionary key, value pair and then appending the list table with the new dictionary row. I found writing the code for removing the table list rows by key was the most difficult task for me this week. After attempting a few methods that I thought would work like 'lstTable.remove(dicRow[strRemTask]', I just wasn't gaining any ground. I had to review the labs, and book a few more times before realizing that I could use similar syntax to how I displayed the list for the user. I used a for else loop to iterate though the list table's dictionary rows by task title then remove the row if the user's input matched the key. If the input did not match a key in the table then the user would be told that the task they tried to remove was not on the ToDo list. (Fig. 2)

```
# Step 5 - Remove a new item from the list/Table
elif strChoice.strip() == '3':
    strRemTask = input("What task would you like to remove? ")
    for dicRow in lstTable:
        if dicRow["Task"] == strRemTask:
            lstTable.remove(dicRow)
            print(strRemTask + " has been removed from the ToDo List.")
            break
        else:
            print(strRemTask + " is not on the ToDo List1.")
        continue
```

Figure 2: For else loop

T.J. Tharp

May 15th, 2023

IT FDN 110 A Sp 23: Foundations Of Programming: Python

Assignment 05

github repo: https://github.com/Tadlol/ToDoList

Try Except

After reviewing a classmate's github code, I realized that I might have an error with my code. I did all my testing with an existing .txt file and did not experience any errors with the final code, however if there was no existing text file the program would experience an error and fail to load or run. I chose to combat this with a try except clause. The program would try to load an existing file and if one was present, the program would state that the user's data was imported successfully. If the program was unable to locate an existing file, the user would be notified that they were starting from starting a new to do list. I tested the program with and without an existing .txt file and experienced no errors. (Fig. 3)

```python
# -- Processing -- #
# Step 1 - When the program starts, load any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)
try:
    objFile = open(strFile, "r")
    for row in objFile:
        strData = row.split(",")
        dicRow = {"Task": strData[0], "Priority": strData[1].strip()}
        lstTable.append(dicRow)
    objFile.close()
    print("ToDo List imported successfully!")
except:
    print("Starting a new Todo List!")
```

Figure 3: Try, Except clause

Summary

The assignment this week was tough for me but I learned a valuable lesson. Don't overthink it. I was able to achieve all the requirements of the assignment with skills that I have learned over the past few weeks. Loops and if statements can complete a significant amount of workload when used properly and syntax used for certain scenarios can most likely be used for other like my example about printing the list table and removing items from the list. Images of my code running in PyCharm (Fig.5) and CMD (Fig.4) are below and on separate pages due to their size.

T.J. Tharp

May 15th, 2023

IT FDN 110 A Sp 23: Foundations Of Programming: Python

Assignment 05

github repo: https://github.com/Tadlol/ToDoList

Figure 4: Code Running In CMD

T.J. Tharp

May 15th, 2023

IT FDN 110 A Sp 23: Foundations Of Programming: Python

Assignment 05

github repo: https://github.com/Tadlol/ToDoList

```
ToDo List imported successfully!

    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

Clean the Cat box,high

    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 new task: Wash Car
Enter Wash Car's priority:low

    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

What task would you like to remove? Clean the Cat box
Clean the Cat box has been removed from the ToDo List.
```

```
    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

Your ToDo List has been saved!

    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

Did you save your ToDo List already? (y/n)y
The program will now close.

Process finished with exit code 0
```

Figure 5: Code Running In PyCharm