Nathan Shelby
23MAY2023
Foundations of Programming: Python
Assignment 06
https://github.com/Nebulis01/IntroToProg-Python-Mod06

Assignment 06: To Do List, now with functions

Introduction:

This is the 6th assignment, part of Module 6 performed as part of the Foundations of Programming: Python course for the Spring 2023 session at University of Washington. The task list includes reading chapter 6 of the coursebook and several web videos covering how functions operate, what data they will accept and return and how to use them.

The primary coursework this week is to modify a starting program, or your functional program from Assignment05 to accomplish several tasks including reading data from a file, echoing it back to the user, manipulating data in memory and writing it to disk – a core requirement of the ask was take your existing program and wrap functions around code where you deem necessary.

Creating your script:

In this script we were asked to use the provided starter and work inside to accommodate the asks within – read data from a file (if present)? Use the menu and manipulate the data in several ways – display, edit, write.

Running through main sets up sample data by executing the function *filework()*. We look at the disk to see if *ToDoList.*txt is present, importing data (skipping the header row) if it is. Otherwise, the *else* statement creates both the sample data in *lstTable* and writes data to disk for a subsequent run. [Figure 1]

```
def filework():
    global lstTable
    if os.path.isfile(objFile):
        importfile = open(objFile, "r")
        skipfileline = importfile.readlines()[1:]
        for row in skipfileline:
            lstRow = row.split(",")
            dicRow = {"Task": lstRow[0], "Priority": lstRow[1].strip()}
            lstTable.append(dicRow)
            importfile.close()
        return lstTable

    else:
```

```
        importfile = open(objFile, "w")
        lstRow = ["Task", "Priority"]
        importfile.write(lstRow[0] + "," + lstRow[1] + "\n")
        lstRow = ["Vacuum Carpet", "1"]
        importfile.write(lstRow[0]+","+lstRow[1]+"\n")
        lstRow = ["Wash Windows", "2"]
        importfile.write(lstRow[0] + "," + lstRow[1] + "\n")
        lstRow = ["Clean Dishes", "3"]
        importfile.write(lstRow[0] + "," + lstRow[1] + "\n")
        importfile.close()
        importfile = open(objFile, "r")
        skipfileline = importfile.readlines()[1:]
        for row in skipfileline:
            lstRow = row.split(",")
            dicRow = {"Task": lstRow[0], "Priority": lstRow[1].strip()}
            lstTable.append(dicRow)
        importfile.close()
        return lstTable
```

Figure 1 – # -- filework() -- #

We make sure to globalize the *lstTable* variable so that other items can utilize it throughout the script, otherwise python will not return expected data due to how it handles local and global variables.

The menu makes use of a while() loop and once input is entered an if() statement evaluates the provided input for 1,2,3,4,5 and processes accordingly.

For a press of 1, we execute *selection1()* -- a *for* loop is used to iterate through the *lstTable* list and display data to the user.

For a press of 2, we execute *selection2(),* (see Figure 2) ask the user to provide input in string form, add those strings to a dictionary and append the dictionary to the in memory *lstTable* list, echoing the content back after the addition.

```
def selection2():
    strTask = input("Enter Task Name:")
    strPriority = input("Enter Priority:")
    dicRow = {"Task": strTask, "Priority": strPriority}
    lstTable.append(dicRow)
    for objrow in lstTable:
        print(objrow)
```

Figure 2 – 'selection2()'

For a press of 3, we execute *selection3(),* (see Figure 3) we instruct the user to review a printed list and provide a value to remove. We loop through the contents of *lstTable* and provide the location of items within the list to remove. We prompt the user to enter that integer input and pass that input to a *del* statement removing the item and kicking them back to the menu home.

```
def selection3():
    print("You will need to review the list and enter the index value for
removal:")
    print("The index values are:")
    for objrow in range(len(lstTable)):
        print(objrow, end=" ")
        print(lstTable[objrow])
```

```
    strIDRemove = int(input("Enter an ID to Remove:"))
    del lstTable[strIDRemove]
```
Figure 3 – 'selection3()'

For a press of 4, we execute selection4(), (see figure 4) we will save the data in *lstTable* to disk. We provide an echo of the contents of *lstTable* to the user and inform them that we will write to disk. As we've been using the *lstTable* list in memory we need to break this apart in to the *string* datatype for the *write()* or *writelines()* methods to handle. We do this by iterating through the contents of *lstTable* and creating a string variable with the dictionary items for *'Task'* and *'Priority'* that we have been visualizing throughout the program. We then pass that string variable to a file handler and when we have done this for all rows in *lstTable* we close the file so that python will commit the write to disk. Reporting back to the user that the write to disk was successful – this could be wrapped in a try/catch to improve resilience.

```
def selection4():
    for objrow in lstTable:
        print(objrow)
    print("Saving above list to file")
    writefile = open(objFile, "w")
    lstRow = ["Task", "Priority"]
    writefile.write(lstRow[0] + "," + lstRow[1] + "\n")
    for objrow in lstTable:
        strtowrite = objrow['Task'] + "," + objrow['Priority'] + "\n"
        writefile.writelines(strtowrite)
    writefile.close()
    print("Successfully saved to file.")
```
Figure 4 – 'selection4()'

For a 5, we execute *selection5()*, which is a change from last week – we can no longer *break* as the function is not within the *while* loop so we process *exit()* instead, printing a message because programs should also have manners 😊.

To provide proof of execution on the local client, ensuring that the script is functional a screenshot is provided. [See Figure 5]

Summary

This week we learned more about lists, working with dictionaries and basic file operations. I look forward to Module 7.

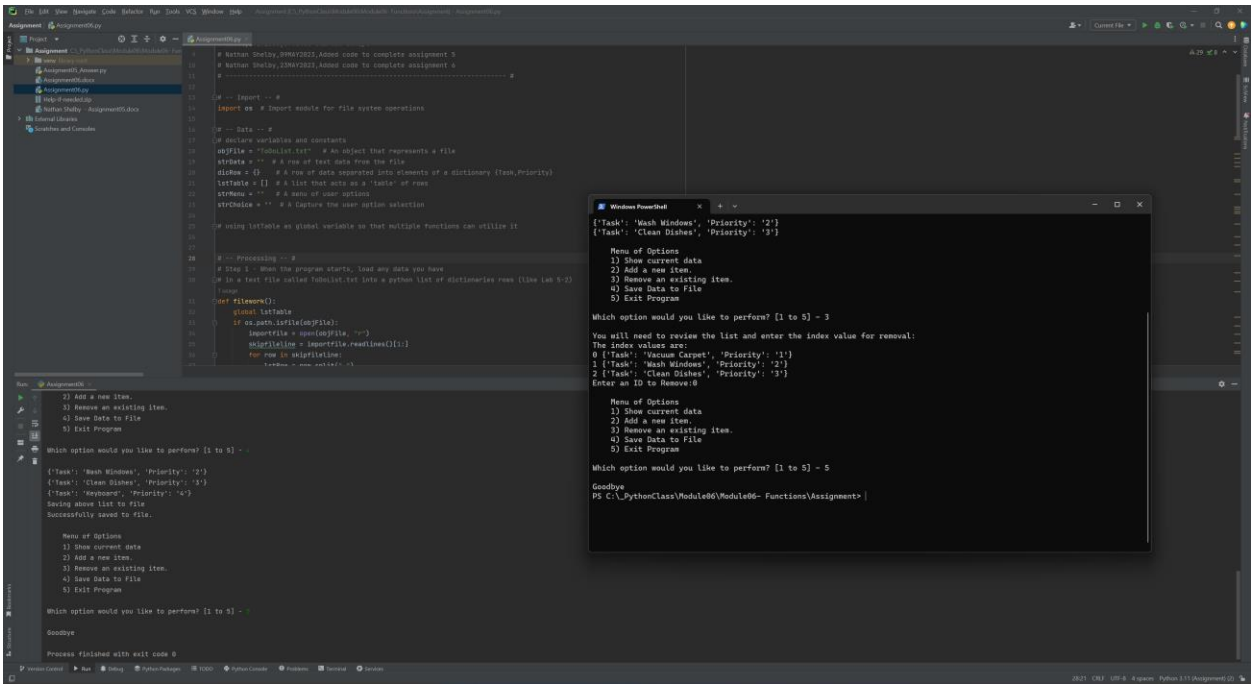

Figure 5 – Proof of Execution