Chanel M. Leung

May 18, 2020

Foundations of Programming: Python

Assignment 05

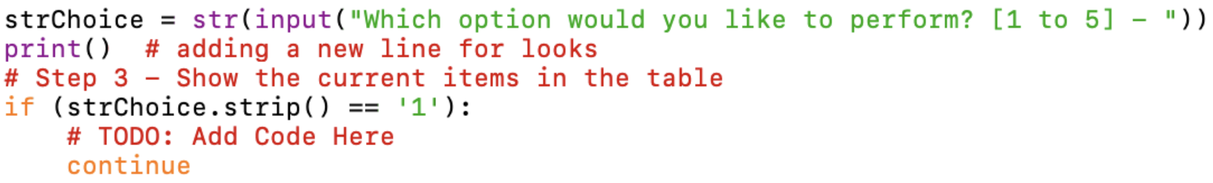
# Tuples, Lists & Dictionaries

## Introduction

Module 5 worked with creating scripts using lists and dictionaries, then processing the data from the scripts to be put onto text files to be saved. This module also gave us starters or outlines to help lead us in the direction of coding which I learned can be both easier as well as more difficult. Using these topics of lists, dictionaries and text files, I was able to create a script that would allow a user to create a to do list with the task that needed to be done as well as the priority. This to do list can add tasks, view the tasks, remove tasks and save the tasks to a text file which I will be explaining the steps to reach the goal of Assignment 5.

## Script Templates

Through this module, I was able to learn more about script templates, which before we would use pseudo code to help try and organize the script. This helps create a more professional look as well as make it easier for many to read the script and find which part of the script could be not working together. For Assignment 5, we were able to use a template to help create a script (Figure 1) which was really helpful since this was a longer script than I have in this class. The template script provided us a notation of where to add code to but basically outlined roughly what the code should do using pseudocode.

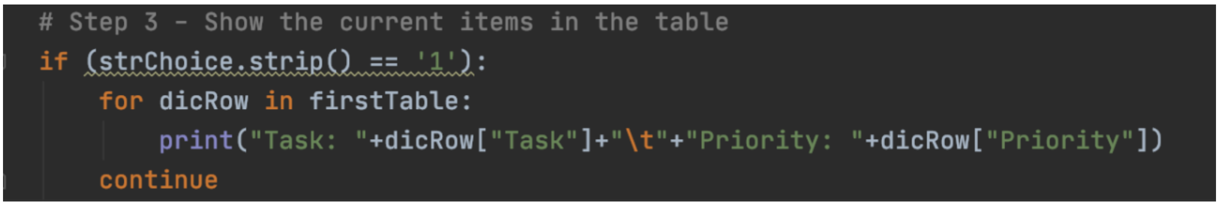


```
strChoice = str(input("Which option would you like to perform? [1 to 5] - "))
print()  # adding a new line for looks
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    # TODO: Add Code Here
    continue
```

*Figure 1*

## Lists

Previous modules have worked with lists but working with them is versatile in scripts. Lists hold a collection of objects that can be used to take those objects out to use them when presenting data. The lists can be used to store, load and save data to a file which is helpful in the long run of collecting and saving users data input. In the script for Assignment 5, I used lists to store, load and save the users inputted “Task” and “Priority” (Figure 2). This way I was able to just get the information of the task and the priority if I wanted to print that out alone or manipulate/process the data in any way. Through creating this script, for each choice of user input I was able to take certain objects for printing out the current data option, when removing items and saving tasks from the to do list.

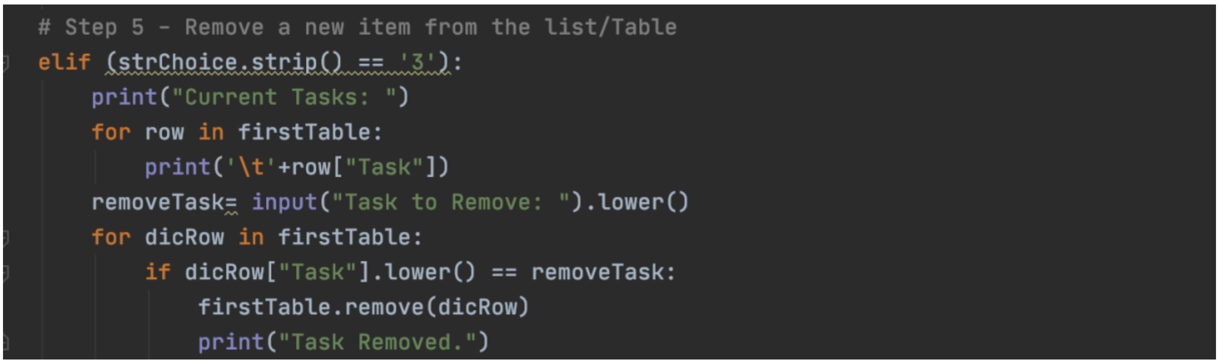


```
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    for dicRow in firstTable:
        print("Task: "+dicRow["Task"]+"\t"+"Priority: "+dicRow["Priority"])
    continue
```

*Figure 2*

## Dictionaries

Module 5 taught us how to utilize dictionaries which can create something like a spreadsheet where information or data can be stored. These dictionary tables can help organize data and take rows of the table to display data to the user. Using dictionaries, you are able to add or read rows from the dictionary tables. In the script for Assignment 5, I was not only able to use dictionaries for adding (Figure 2), but also removing and saving tasks to text files. For adding, I used dictionaries in order to append the “table” to add new data that the user had inputted. Removing tasks from the lists was more difficult for me to code, but I was able to remove a row from the tasks dictionary to reach the goal or removing it off the list. When saving tasks, I was able to use dictionaries to get the certain task and priority information to save to a text file.



```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    print("Current Tasks: ")
    for row in firstTable:
        print('\t'+row["Task"])
    removeTask= input("Task to Remove: ").lower()
    for dicRow in firstTable:
        if dicRow["Task"].lower() == removeTask:
            firstTable.remove(dicRow)
            print("Task Removed.")
```

*Figure 3*

## Summary

Assignment 5 was more time consuming that I would think even though I had a script template to base the code off of. This is why I thought it was easy, but difficult to use a template since knowing what you want to input is outlined, but difficult because we would want to create our own way of how the script would lay out. Through the different labs we went through, I was able to utilize the different topics at least once through my script to complete the objective of creating a to do list that is able to be read, added to, remove and save user input data.