Michael Clark

February 20, 2021

IT FDN 110 B

Assignment 06

https://github.com/MClark89/IntroToProg-Python

Creating a Python Script that Gathers User Data

Introduction

As we delve deeper into this course, we can learn more advanced ways of gathering and processing user input. One way to process user input in a more efficient way is through functions. Functions allow for the person writing script to condense large script files into single phrases. This week we used functions to simplify the output of user generated data.

Writing the Script in Steps

This week’s assignment was almost the same as last weeks, except we incorporated functions, so we can become familiar with how they work. The main body of the script was copied from a file given to us by Professor Root. It was split into three sections Processing, Presentation, and Main body of script.

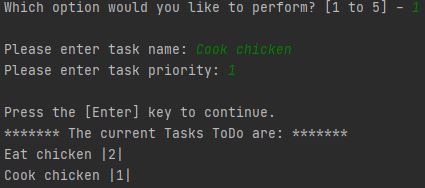
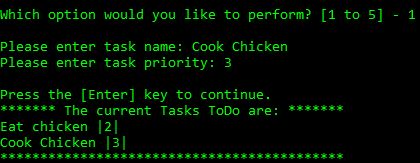
In the processing section, we were tasked with defining three functions; add data to a list, remove data from said list and write the data to a text file. These were like the homework from last week, so I copied them, but had to make a few edits for the code to work. One of the main differences was using the list\_of\_rows function which allowed us to easily define what data set was being edited by adding either a .append to add or a .remove to subtract an item from the list.

For the Presentation section, the code was mostly written out for us. All that needed to be updated was defining two functions. The first function was defining how the data was stored in the list by defining the Task and Priority to specific rows and allowing this data to be appended. The other code that needed to be entered was for a function that allowed for data to be removed from the list based specifically on the task name.

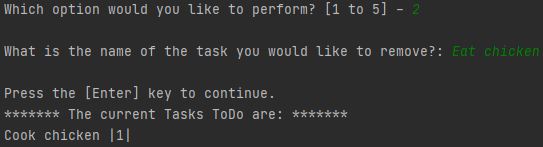
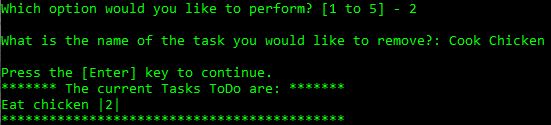
The final section where the main body of the script was written was relatively straight forward. All major functions had already been defined in the previous two sections. All that was needed was to polish up the end user experience. This was accomplished by using the correct functions for the correct task. If you wanted to remove a task you would use the specific function that defined how to remove a task, while making sure it created an easy experience for the end user. While the two sections defining the functions used more lines of code than the main body of the script, what is more important is the implication of the first two sections. When creating a large code file, functions can allow a programmer to be more efficient with their time and space, by using functions that condense large statements into simple variables that can be used repeatedly.

## Checking the Script

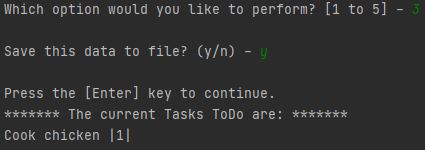
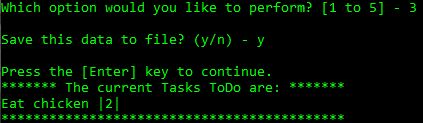
In order to make sure that the script performed the functions properly, we needed to test it out in both PyCharm and the command window. In Figures 1-5 it shows how the script allowed the end user to enter data onto a list, remove other data from the same list, and save the data to a text document.



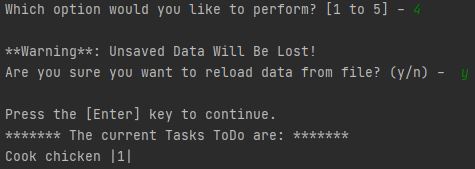
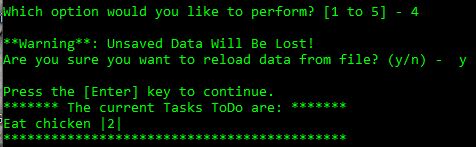
***Figure 1. Entering in Data***



***Figure 2. Removing Data***



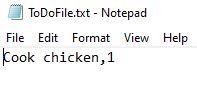
***Figure 3. Saving Data***



***Figure 4. Reloading Data***



***Figure 5. Exiting Script***



***Figure 6. Checking Text File***

Summary

As we get further into the course, the scripts we are using are becoming more complex and larger. In order to save space and time, functions are implemented to allow the programmer to condense large scripts into simple phrases that then can be copied and pasted throughout the script.