Claire Lynch

November 16, 2020

IT FDN 100 A

Assignment 05

GITHUB LINK

To-Do List with Dictionaries

# Introduction

This week I created a program that creates and organizes a To-Do list for the user, using Dictionaries, Tables, and Lists. I also relearned how to use GitHub and how to upload my files to repositories on there, and how script templates can be useful to provide a consistency in my code across multiple files. I also learned about the Try/Except error-handling construct, which helps make any errors that the user see more friendly to them.

# Dictionaries

Dictionaries are similar to Lists and Tuples, but they replace an Index with a Key subscript. To indicate that you want something to be a dictionary, you use brackets to do so.



***Figure 1.1: An example of a Dictionary from my program. Task is one of the keys, and priority is the other one.***

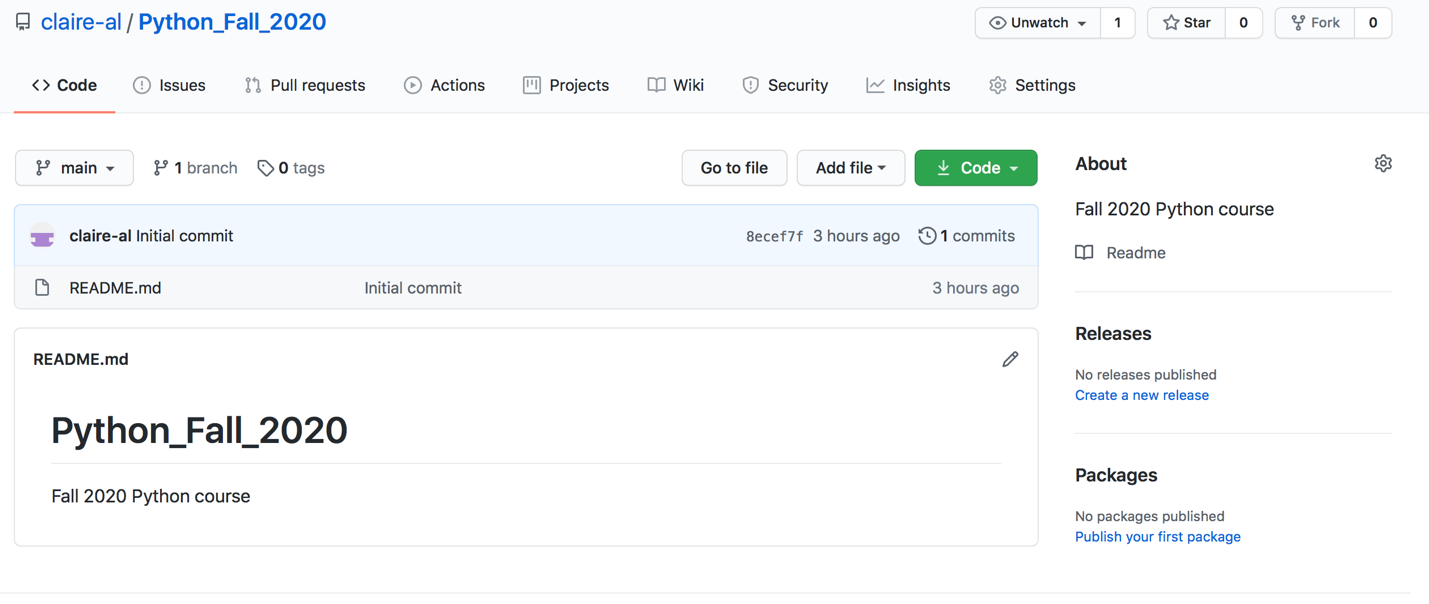
It can be helpful to think of a dictionary as a row of data, and each row can be added to a list to create a table of data. It can be helpful to think of it like an Excel spreadsheet.

# Writing Better Scripts

There are numerous things that I can do to make my scripts look better. I can follow the **Separation of Concerns** concept, where you separate each part of your program into different sections, with each section addressing a different concern. Another thing I can do is to use a **Script Template**, which is a template that I could consistently use with any Python program I’m writing, which can ultimately make my programs look cleaner and more consistent. **Handling errors** is also helpful, such as with the Try/Except error-handling construct, because it makes any errors that do occur more user-friendly, meaning that perhaps the user can fix it before having to get someone who better-understands the program involved. The last thing you can do is use **functions**, which reduces the amount of code you have to write numerous times.

Git and GitHub

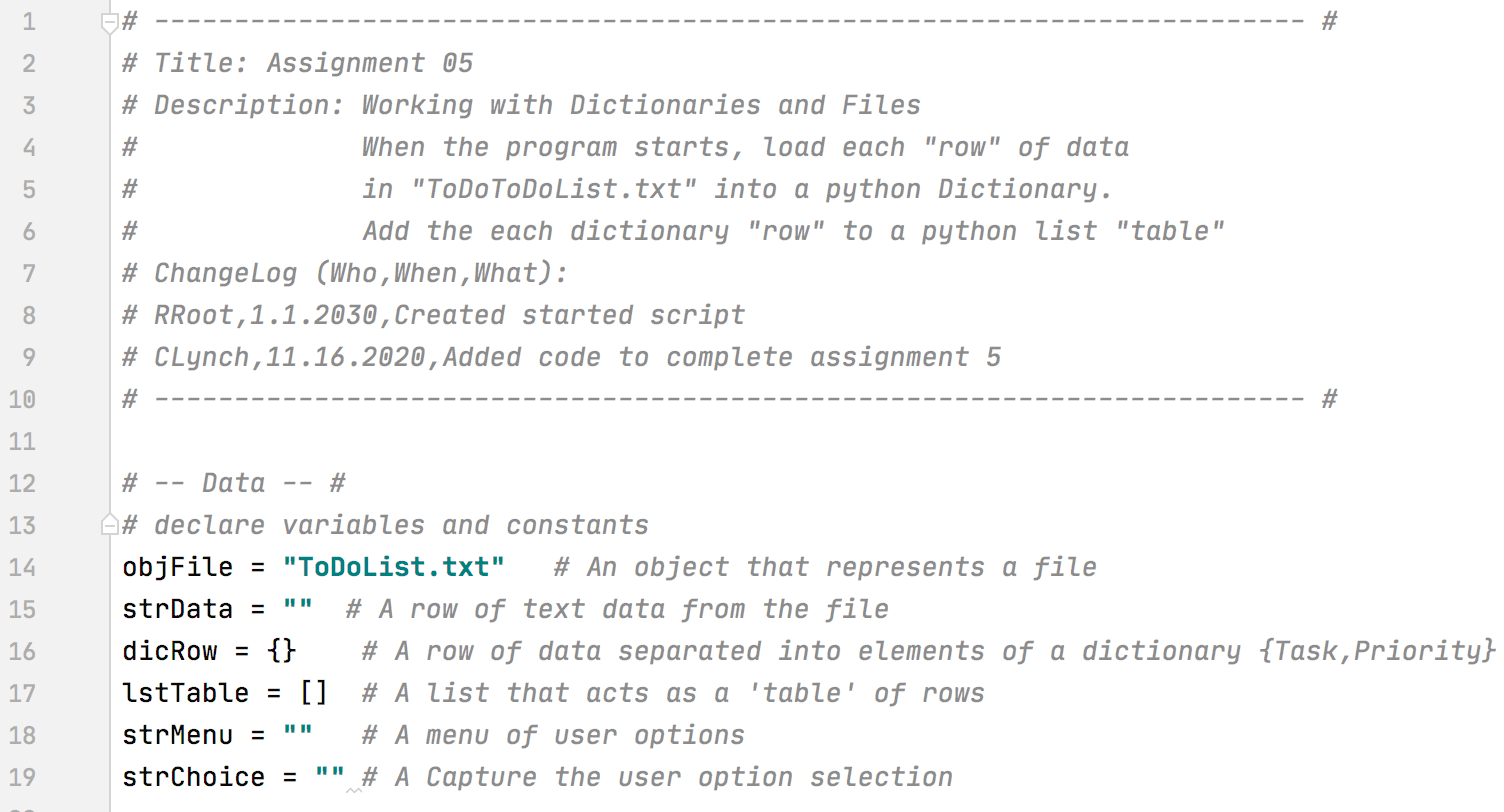
I’ve used GitHub in the past, but GitHub is a wonderful resource to return to. It can store backups of any programs you write in case anything were to be lost, and allows for better collaboration among programmers. It was easy enough for me to sign back into my account and create a new repository, where I can store all of my files for this class.



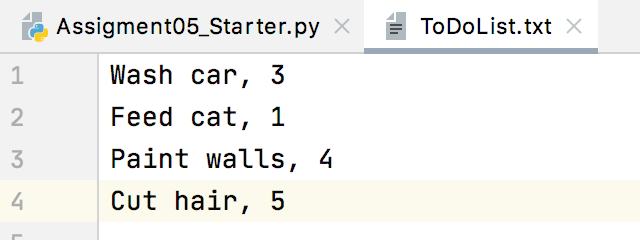
***Figure 2.1: My repository that I created for this course.***

Writing My Script

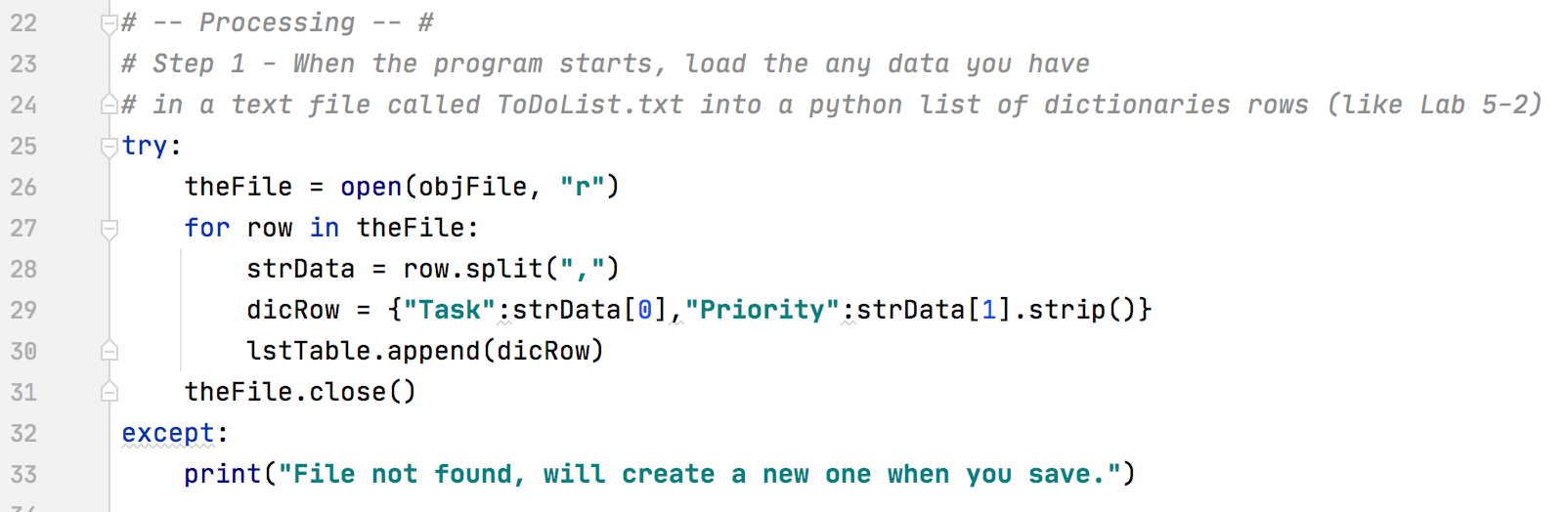
I at first found it challenging to write a program having been given the bones by Professor Root, but once I took the time to understand the variables and what the program needed to accomplish, I actually found it nice that I didn’t have to write all of the smaller details that were already defined. I found it helpful to have the defined variables separated at the top, with the processing section following, completed by the Input and Output. In the Input/Output section, each of the five choices in the Menu of Options was separated and I worked on getting each one to work individually before moving onto the next.



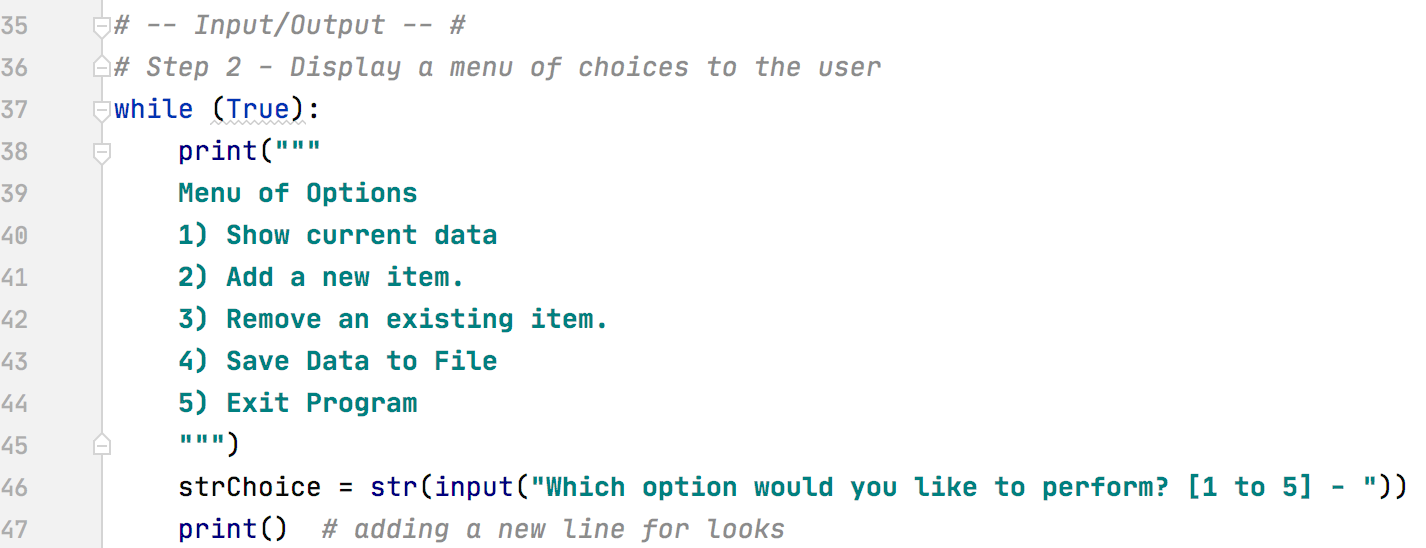
***Figure 3.1: The comment and the Data section in my program.***



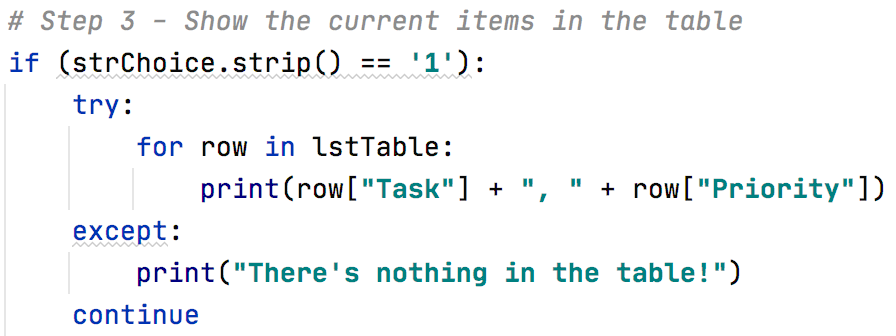
***Figure 3.2: The ToDoList.txt text file at the beginning of the program.***



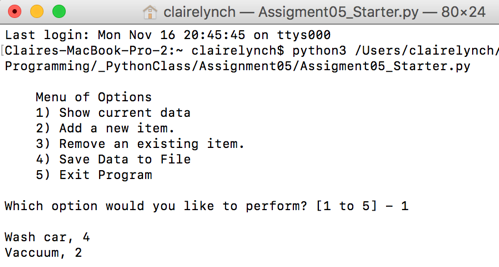
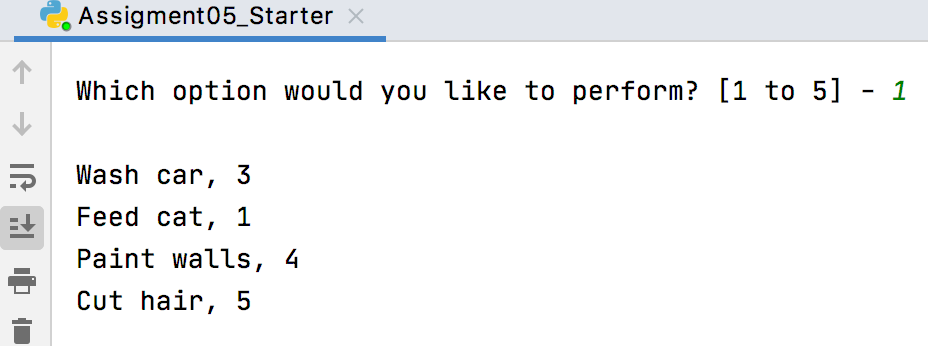
***Figure 3.3: The processing section of my program.***

******

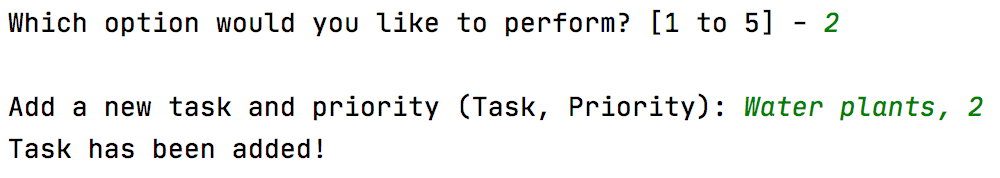
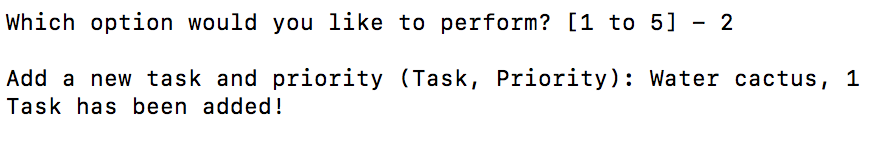
***Figure 3.4: The beginning of my Input/Output section.***

******

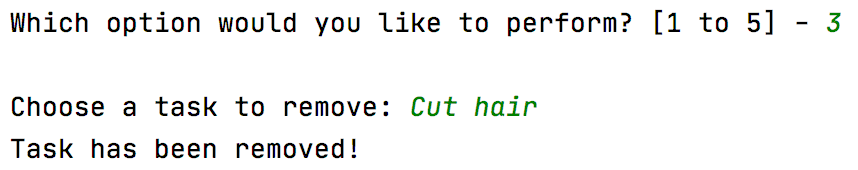
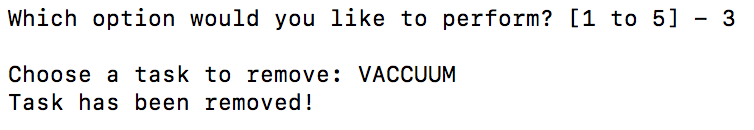
***Figure 3.5: The script to accomplish option 1, showing the current items in the table.***

******

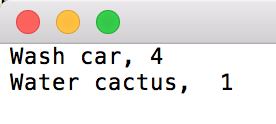
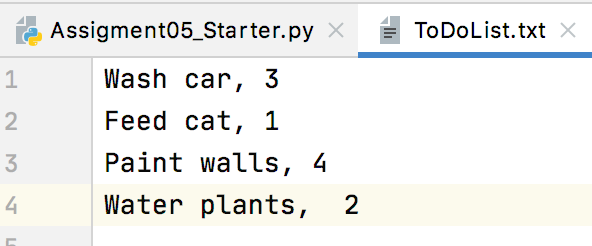
***Figures 3.6 and 3.7: The script running from PyCharm (left) and Terminal (right).***

*** ***

***Figures 3.8 and 3.9: Option 2, adding a new task running from PyCharm (left) and Terminal (right).***

*** ***

***Figures 3.10 and 3.11: Option 3, removing a task running from PyCharm (left) and Terminal (right).***

******

***Figures 3.12 and 3.12: The final TODO list after being saved, from the PyCharm inputs (left), and the terminal inputs (right).***

# Summary

I actually really enjoying writing this script. After initially thinking I wouldn’t like having parts of the script already written by Professor Root, I ended up liking it since it forced me to think slightly differently than I’m used to, and it challenged me to write a script in a more “collaborative” way. I’m excited to do more with Dictionaries, and found that I actually grasped it fairly quickly since I spent the time thinking about its structure before jumping into any programming. I’m also excited to be working with Git and GitHub again, and am excited to see how it’ll allow us to be more collaborative.