Kristen Burke

Aug 17, 2021

IT FDN 110A

Assignment 06

GitHub: <https://github.com/kristenrb/IntroToProg-Python-Mod6>

The To Do List, All-New, All-Different

# Introduction

This assignment explored creating and using functions in a python script. The prompt asked for a program for building a “To Do” list, with tasks and their priority level. As before, a starting file was provided, separated out into each step, leaving spaces blank to fill in a function. The assignment was coded in PyCharm.

# Functions and Concepts Used

A custom grouping of commands is called a function and must be defined before it can be used in the body of the script. The format of a function is normally a name with parenthesis for potential parameters, such as **some\_function(parameter)**. When the function is used, arguments are passed into the parameters as the values for the function to act on. Defining functions gives an easy way to organize a script into processing and input/output sections, making the program easier to read and use elsewhere.

# Creating and editing the Program

I started by updating the change log in the Assignment 6 starter file (Figure 1). I also left the defined global variables at the top of the script.

The next section of the script contains the processing class of functions. The first function for reading data from a file was already filled in. The next three needed to be completed: **add data to list**, **remove data from list**, and **write data to file** (Figure 2). I worked on each part one-by-one, also adding a call to the function in the main body of the program under the relevant menu option.

But for the program to even do any processing, it would need some data to process. The next section was the presentation, or input/output section of the script. Here, the only functions left to create were the **input new task with priority** and **input task to remove** functions. I looked at the previous assignment for what code to use and changed some of the variable names (Figure 3).

Tying together the presentation and processing functions in the main body of the program was a bit trickier. I had to pay close attention to global versus local variables. The solution I landed on could probably still use some tweaking for consistency (Figure 4).

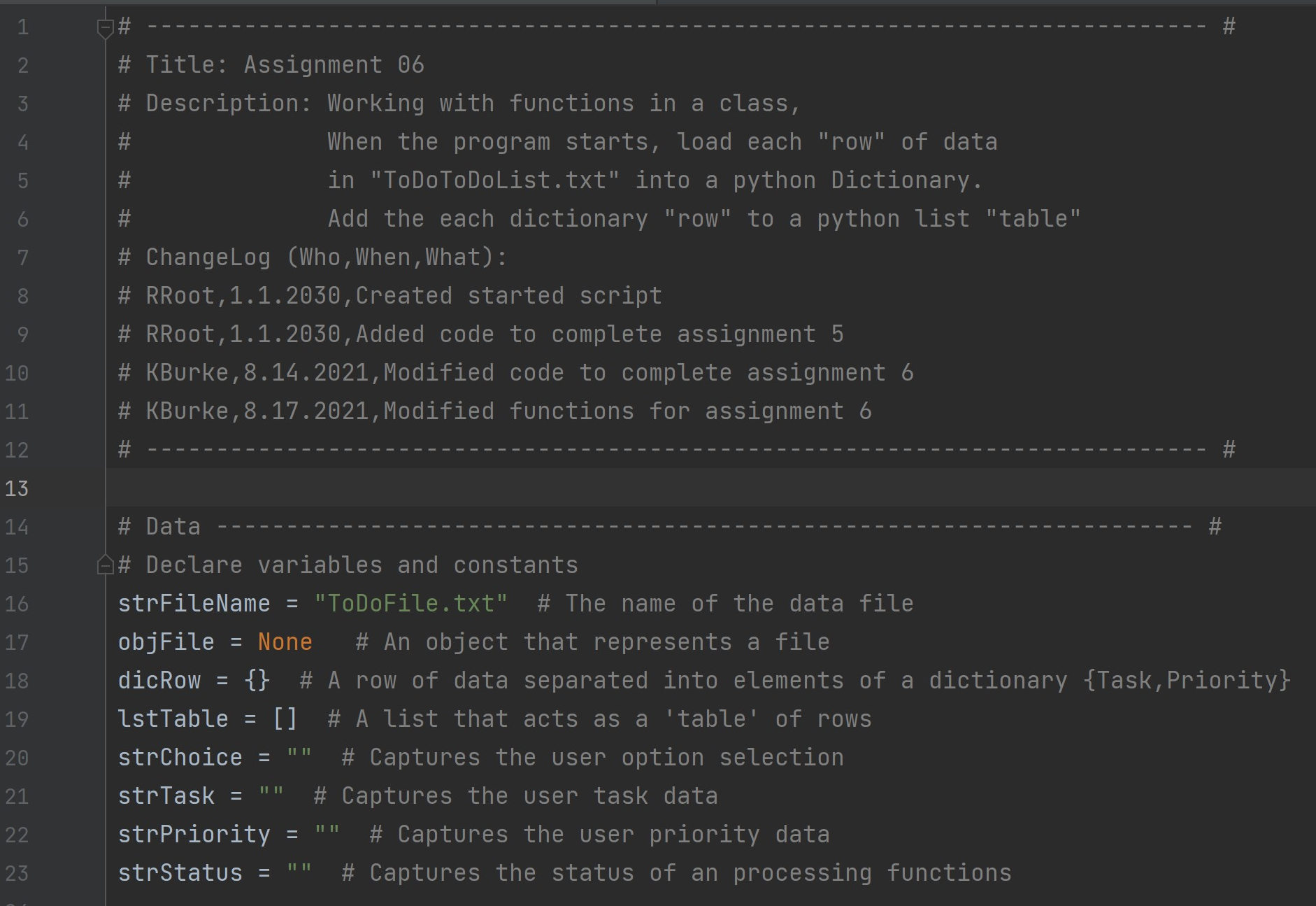


Figure 1: Script header and declared variables

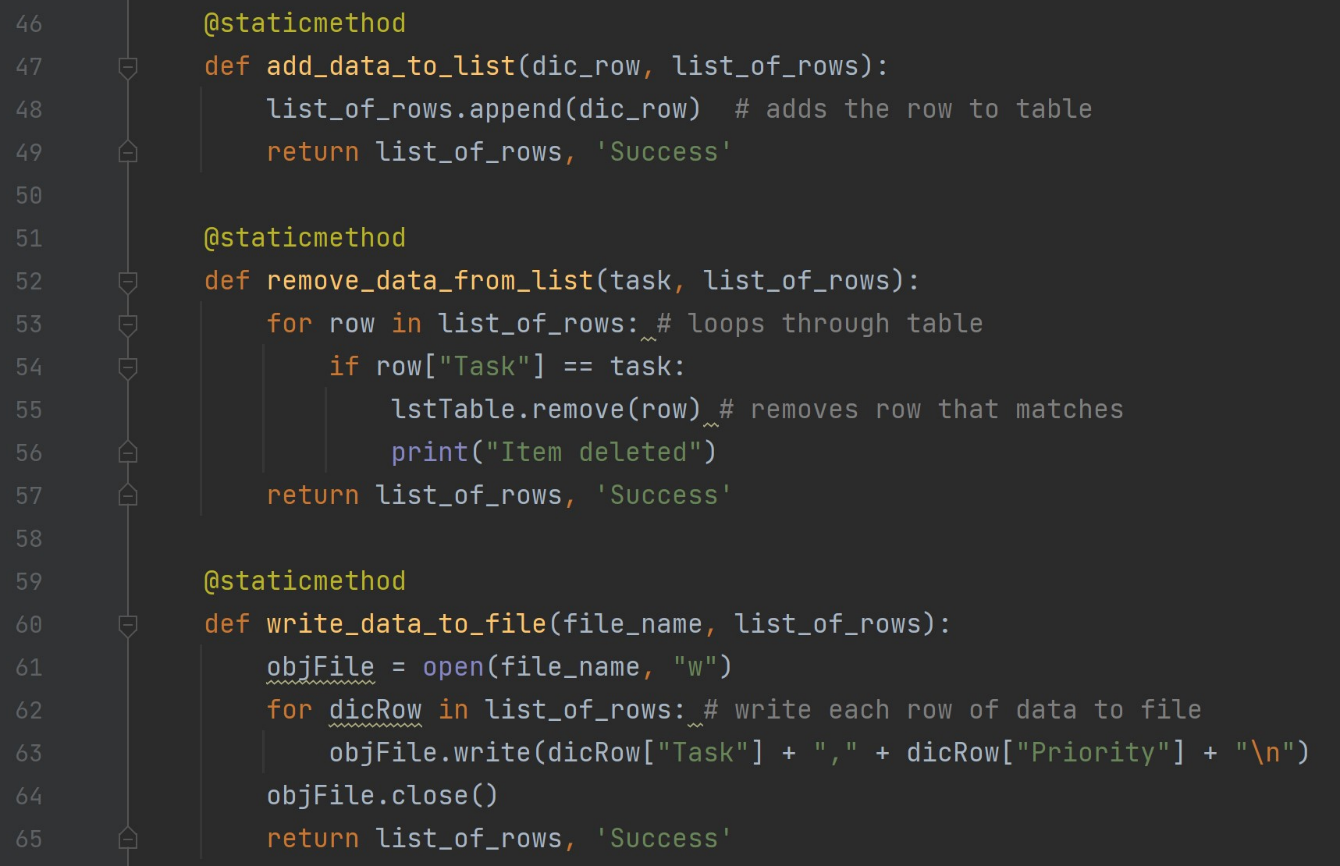


Figure 2: processing functions add data, remove data, and write to file.

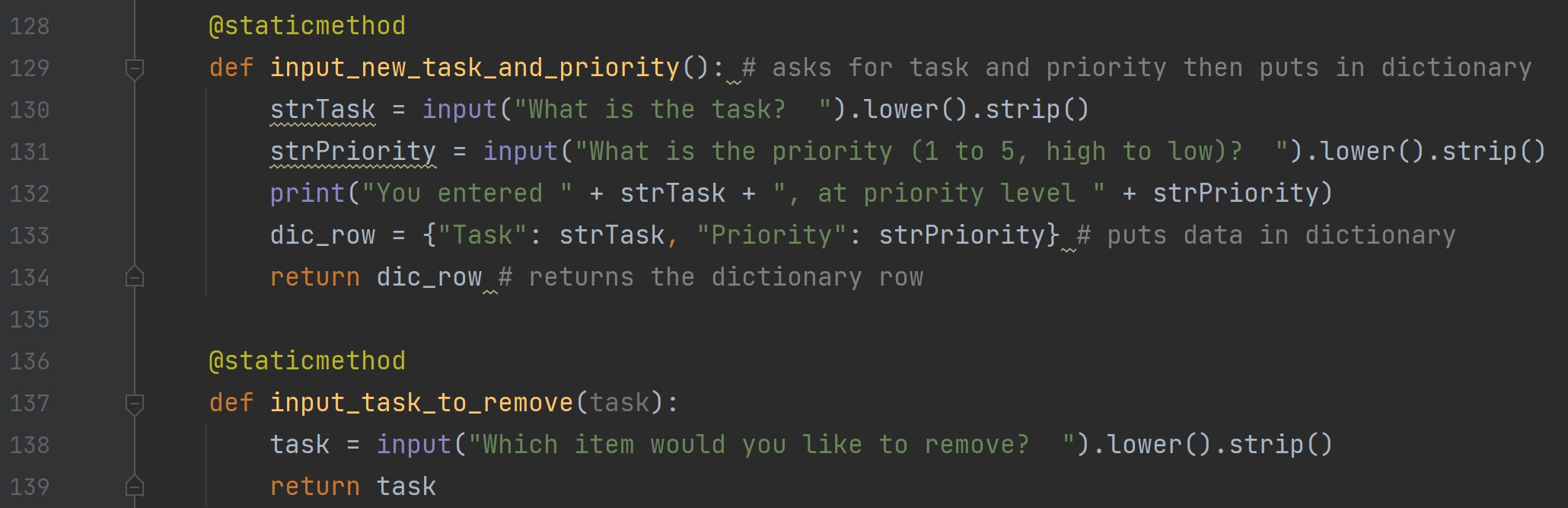


Figure 3: new task/priority and new task to remove functions

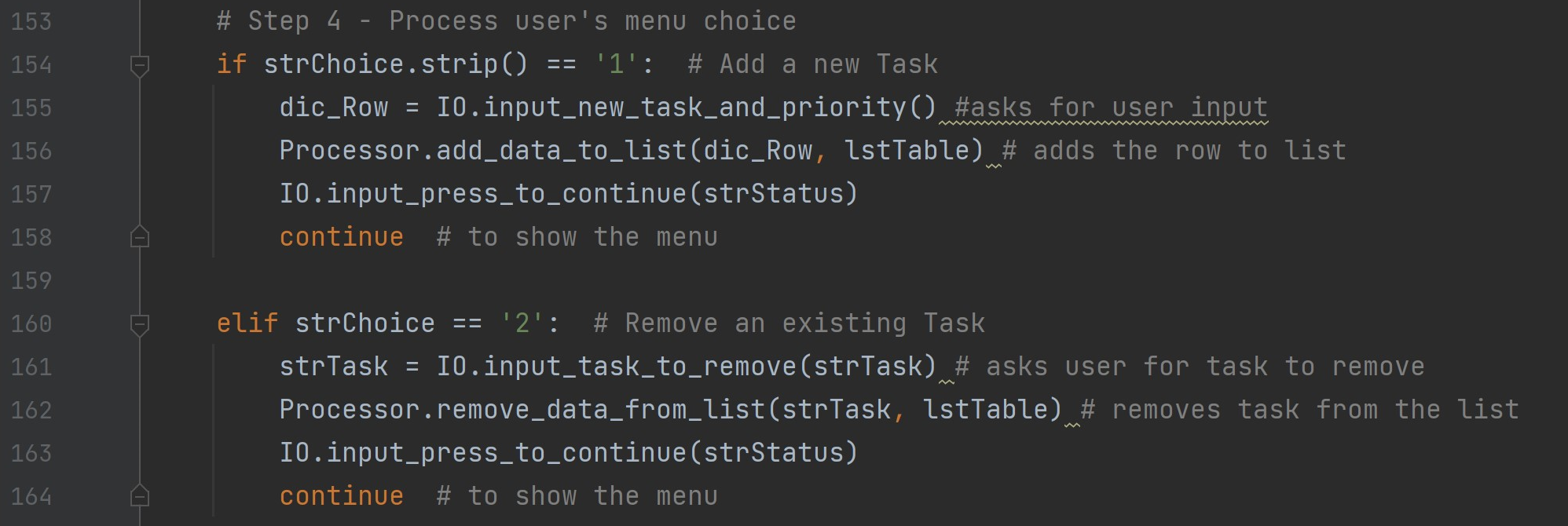
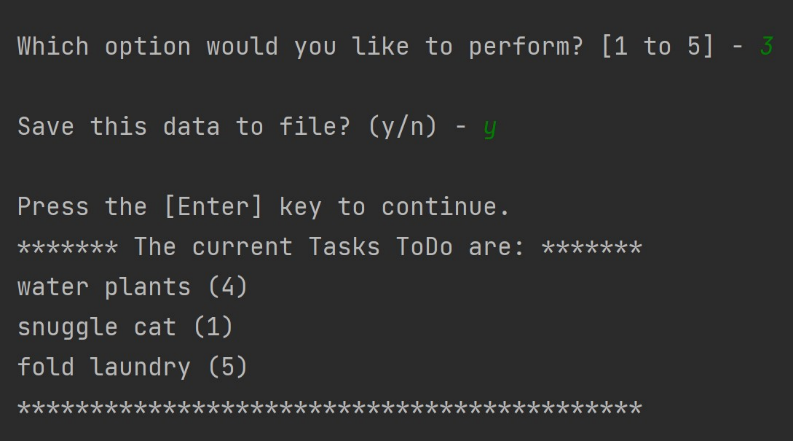
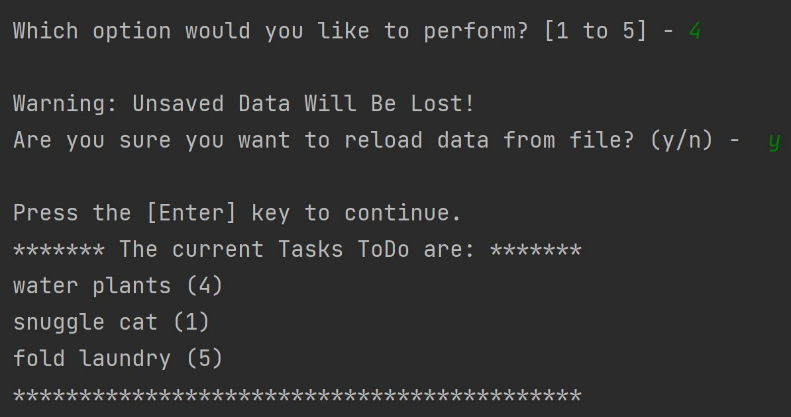
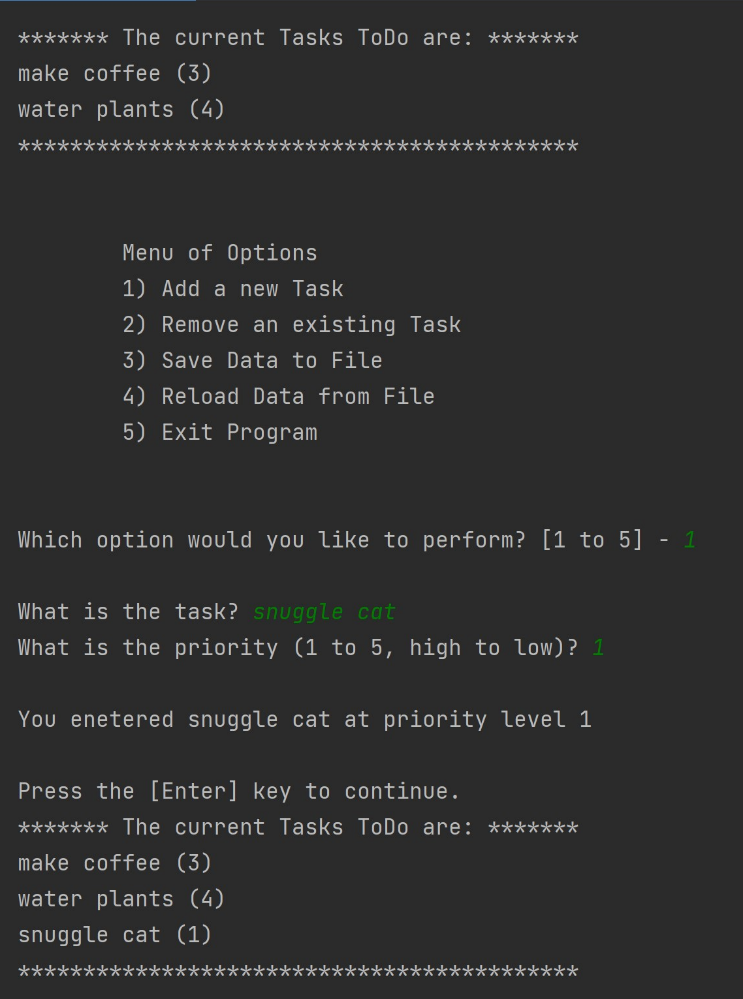
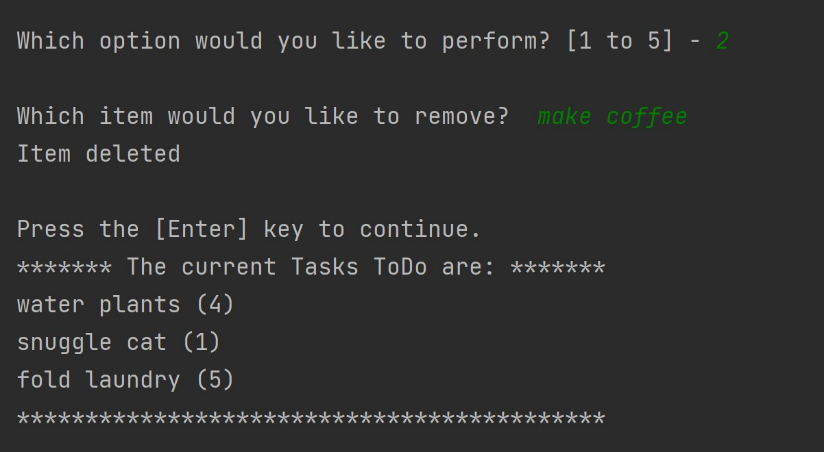
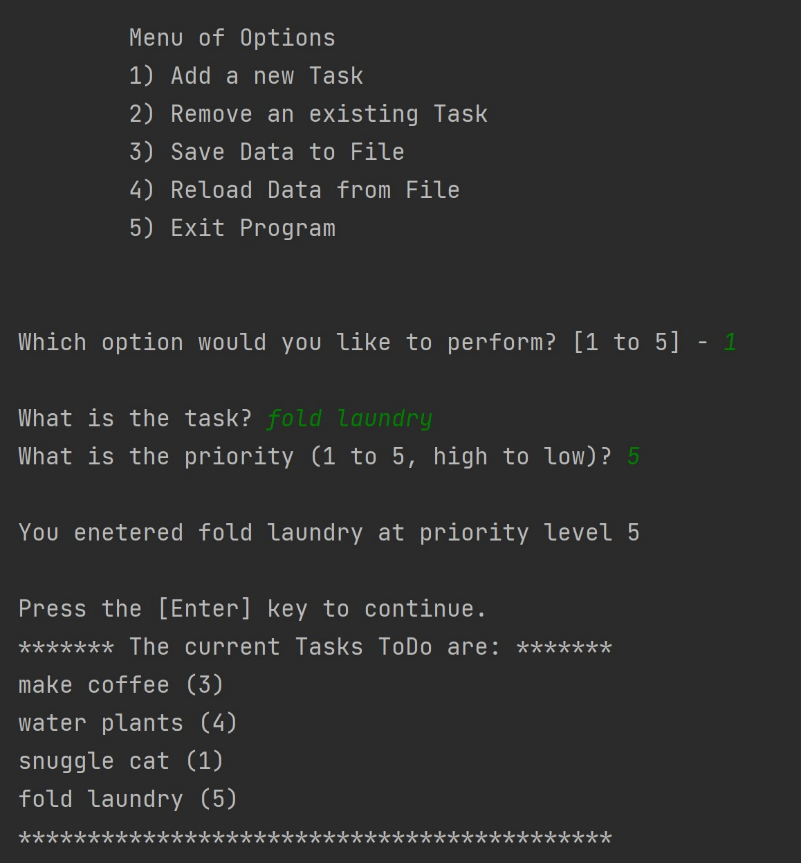


Figure 4: main body of code for adding new task and removing an existing task

# Running the Program

I ran the program in PyCharm and found it to be working as I hoped (Figure 5). The menu appears after every option and does not change so that part was cropped out in the screenshots for more space to show the rest of the program working. I also verified that the data saved does show up in the text file.



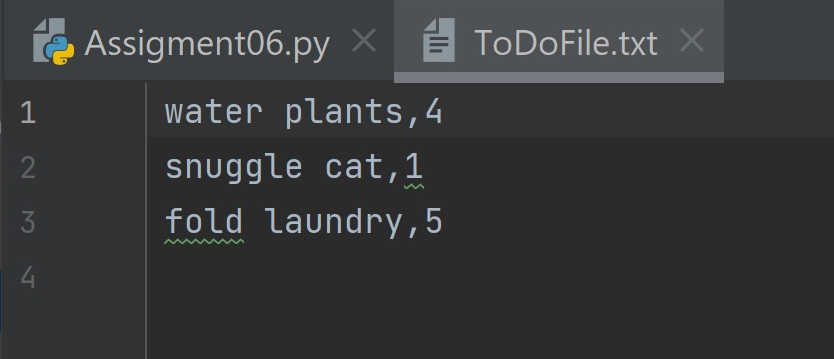


Figure 5: Running the program in PyCharm.

Figure 6: Verifying data was saved to text file.

After changing the file directory, I was able to make the program run in the command window as well (Figure 7).

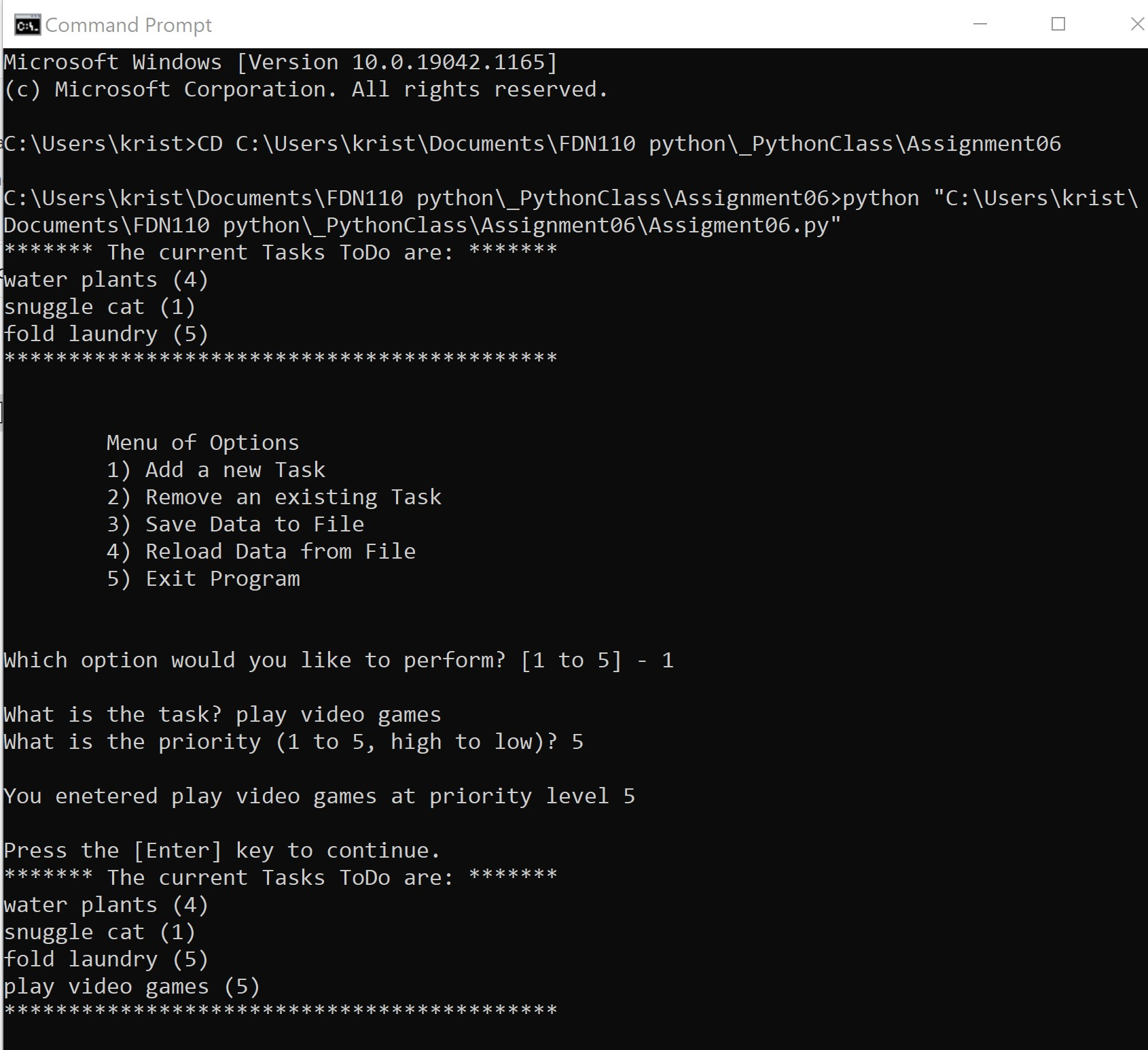


Figure 7: Running the program in the command window.

# Conclusion

Overall, I was able to edit the starting code into something functional, avoiding some of the issues from previous assignments. I can appreciate the reasoning behind separating out the different types of functions as it does make the script feel more organized.