Daylen Haines

November 11, 2019

Foundations of Programming: Python

Assignment 06

**Python To Do File Modification**

**Introduction**

In this document, I will describe how to modify a pre-existing script that manages a “ToDoFile.” Focusing on organizing the pre-existing code through the use of functions. I will also provide an overview of functions, parameters, arguments, return values, global vs local variables, and how functions can help organize your code. This document provides the steps I took and can serve as a guidance for beginning programmers writing or modifying a script with functions.

**Function**

In Python a function is a way of grouping one or more statements (Root R., Mod6PythonProgrammingNotes. UW Foundations of Python Programming, 2019). Functions must be defined before they can be utilized in your Python script. Once the function is defined, you can execute the statement or statements in your function by simply calling the function (Figure 3.0)

**Parameter**

A parameter is an optional subcomponent of functions that allow the user to pass values into a function for processing. Values passed into parameters are better known as “arguments” (Root R., Mod6PythonProgrammingNotes. UW Foundations of Python Programming, 2019). A benefit of parameters is that there is no real limit to how many parameters you can include in a function. You can utilize them to pass arguments that calculate variables. Its standard practice in Python and most other programming languages to not use a prefix for your parameters (Figure 3.0)

**Return Values**

Functions have the ability to return one or more values that are captured via a variable. Return values make a function that act like an expression, which means you can use the results of a function immediately without placing the results in a variable. Capturing the results in a variable allow you to use the variable results multiple times without having to call the function again (Root R., Mod6PythonProgrammingNotes. UW Foundations of Python Programming, 2019). One benefit of return values is that there is no presentation code in the function, making it easier to divide into data, processing, and presentation. You can utilize return values as a single data point or multiple data points. When utilizing multiple data points, it’s important to capture them in a tuple (Figure 3.0).

**Global vs Local Variables**

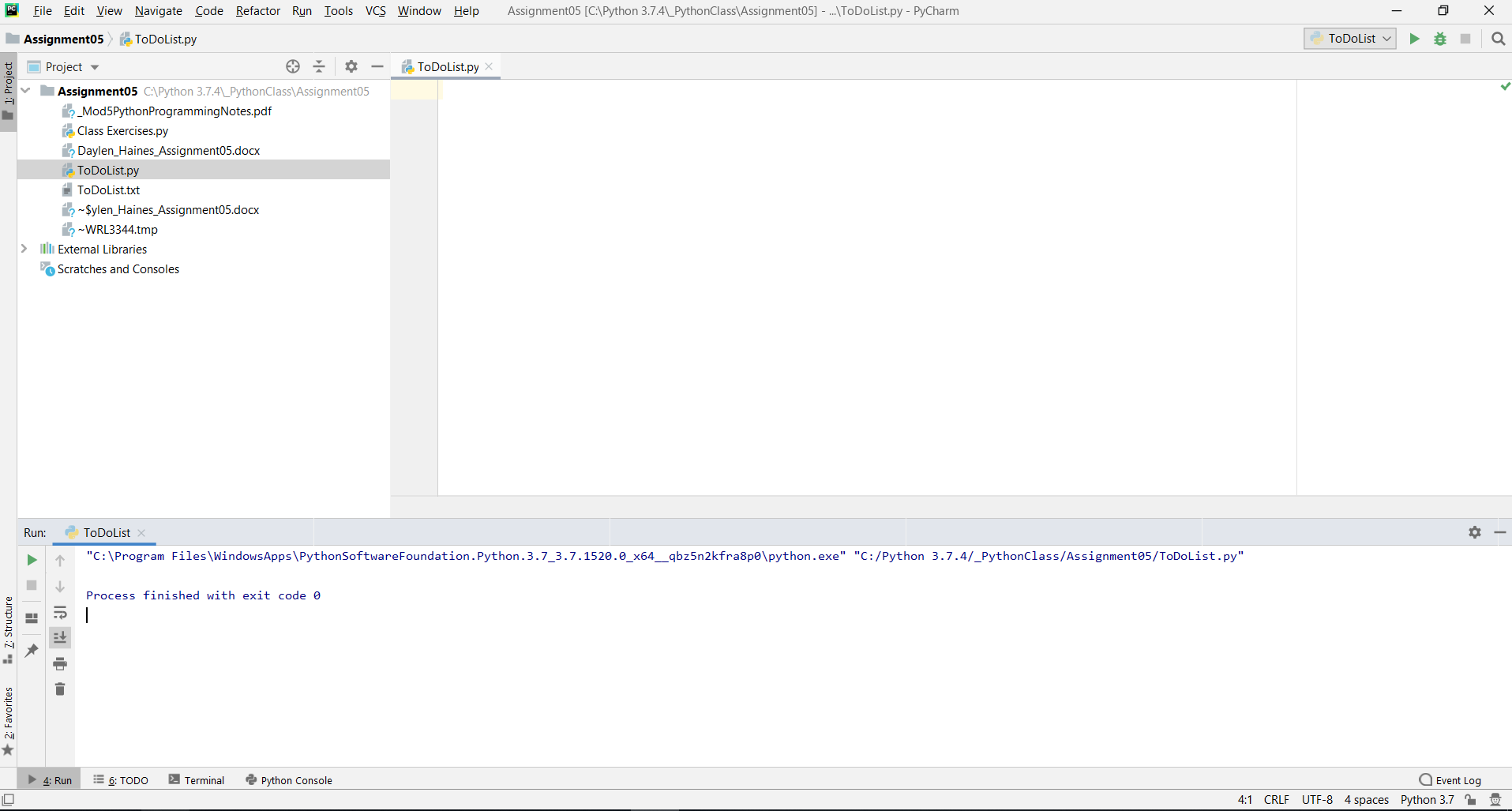
Variables in a script are defined as either local or global. Local functions are declared in a function and cannot be accessed outside of that function. Global functions are declared in the “body” of a script and can be used anywhere within the script (Root R., Mod6PythonProgrammingNotes. UW Foundations of Python Programming, 2019). While not recommended, if you decide to utilize global variables, be sure to use the keyword “global” in front of your variable or else you can over “shadow” a variable with the same name in your script.

**Separations of Concerns**

As we begin to develop more complex Phyton scripts/ programs, keeping our data clean and concise gets more and more important. Not only will this benefit the programmer, it will also benefit anyone trying to read your code. Separations of Concerns deals with separating your program into predefined area that address a different concern (Root R., Mod5PythonProgrammingNotes. UW Foundations of Python Programming, 2019). The main areas are Data (ex. Declaring variable and constants), Processing (Tasks you are going to perform on data within your code), and Presentation (input/ output, obtaining user input). By structuring your script/ program to separate these three areas will allow for much cleaner and easier programming. Functions greatly help structure your script into these predefined areas.

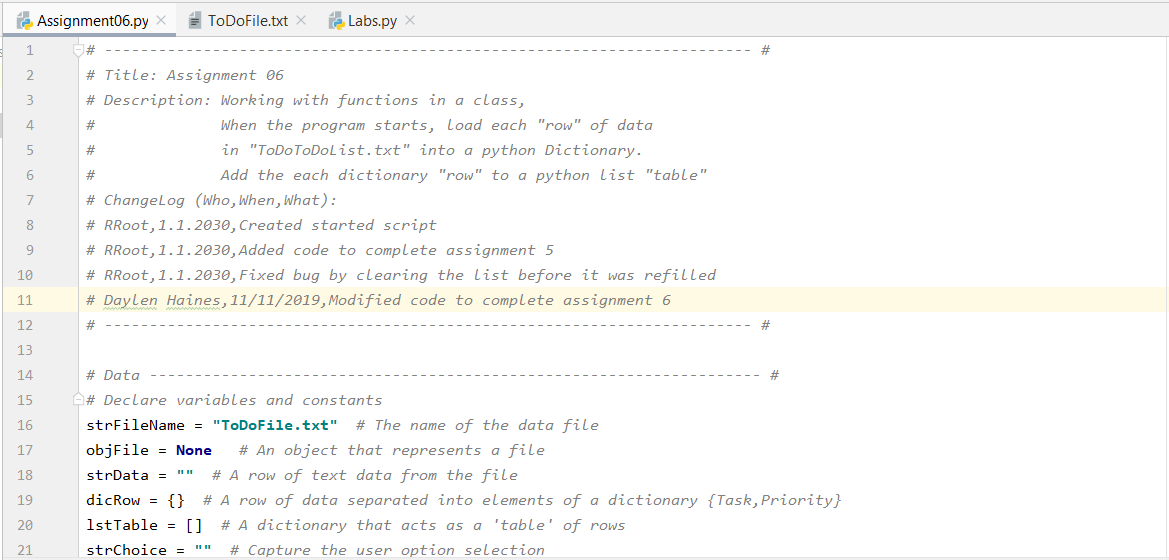
**Writing the Requested Python Program**

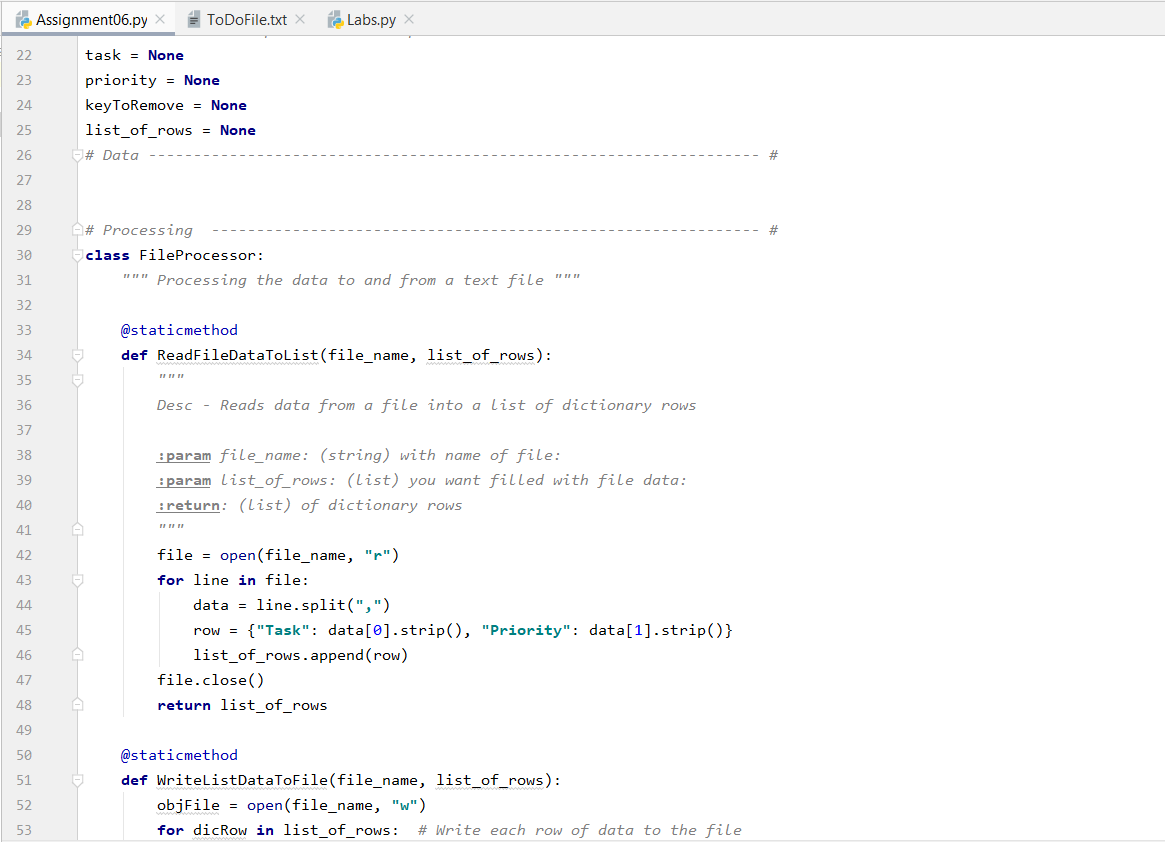
I first created a new sub folder for Assignment06 and opened up the PyCharm program. Once the program was opened, I followed the steps that professor Root provided in the “Intro to Python Mod03” (<https://www.youtube.com/watch?v=EoCXrbkdI0c&feature=youtu.be>) (External Site) video, but made some minor modifications to create Assignement06.py file in my Assignment06 folder. Once I had PyCharm up and ready to go, I utilized the “Intro to Python Mod06” (<https://www.youtube.com/watch?v=jiXmXhwgHp8&feature=youtu.be>) (External Site) video Professor Root provided and Python Programming for the Absolute Beginner book (Dawson M., Python Programming For The Absolute Beginner. Course Technology, 2010) to complete the requested task of modifying a pre-existing script through the use of additional functions to help organize the code.

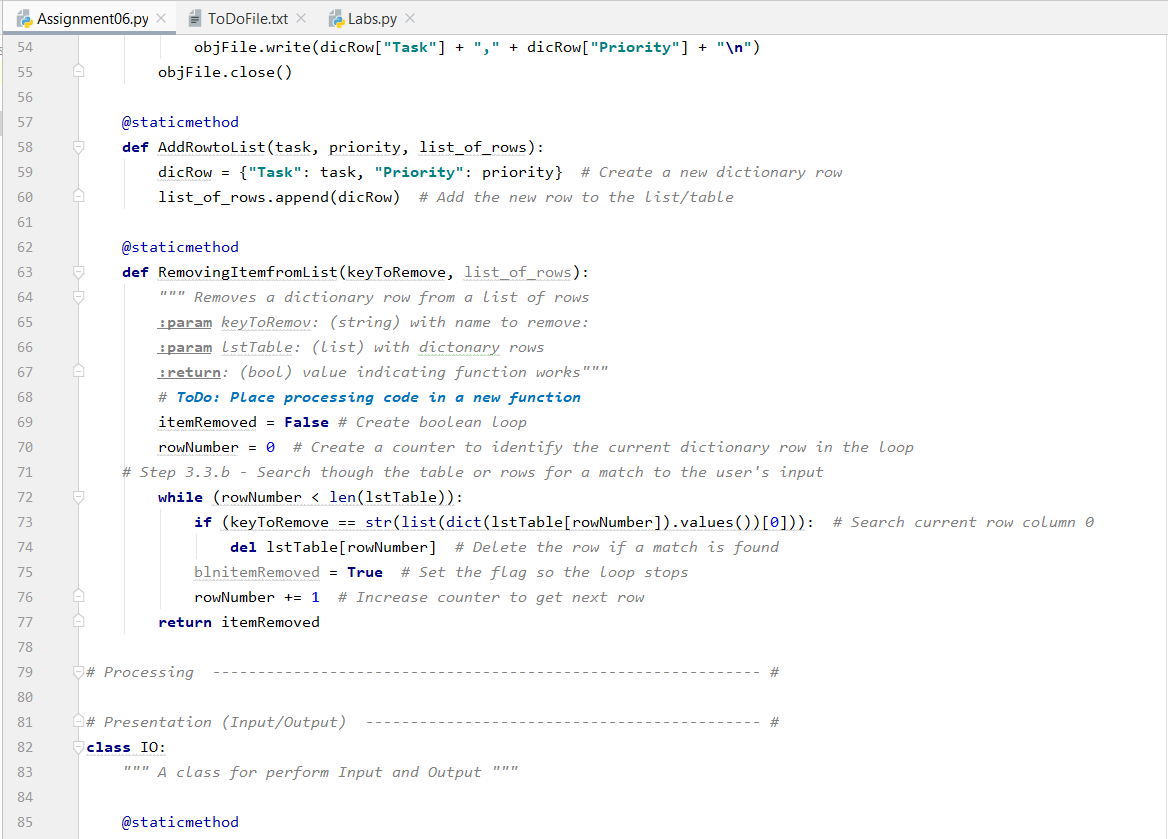


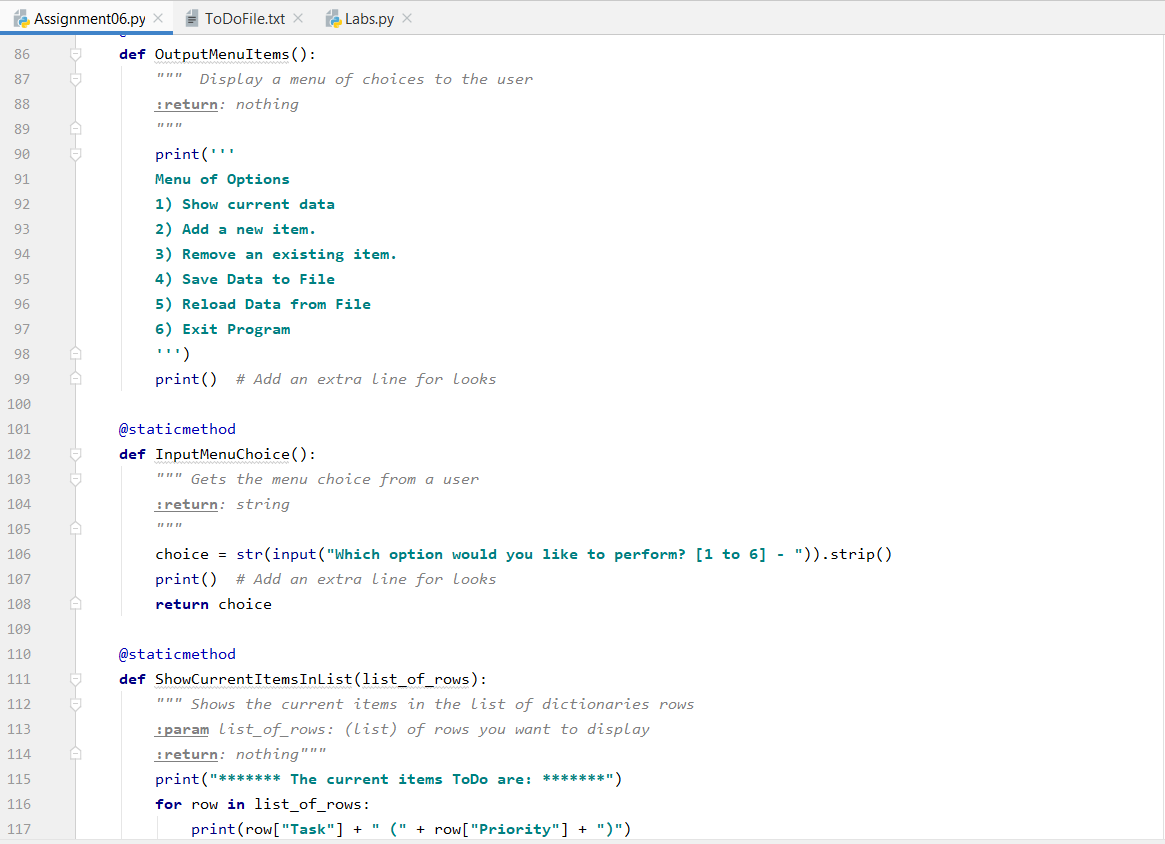
**Figure 2.0 – PyCharm Upon Launching**

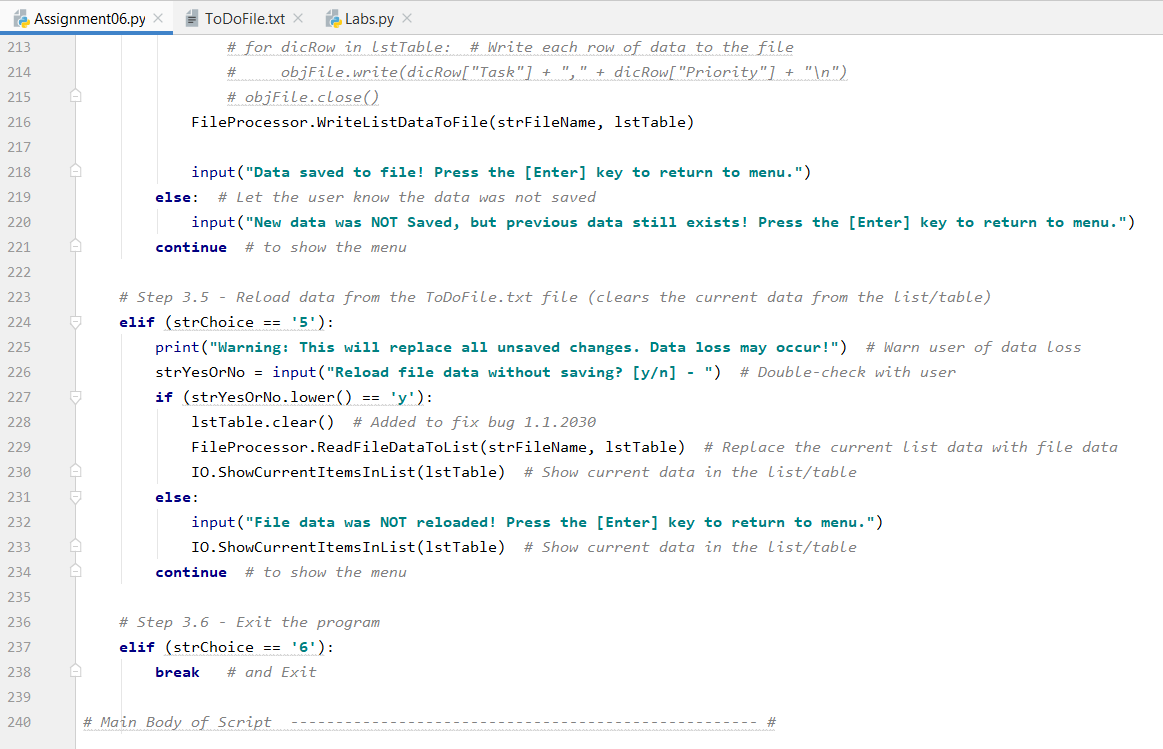
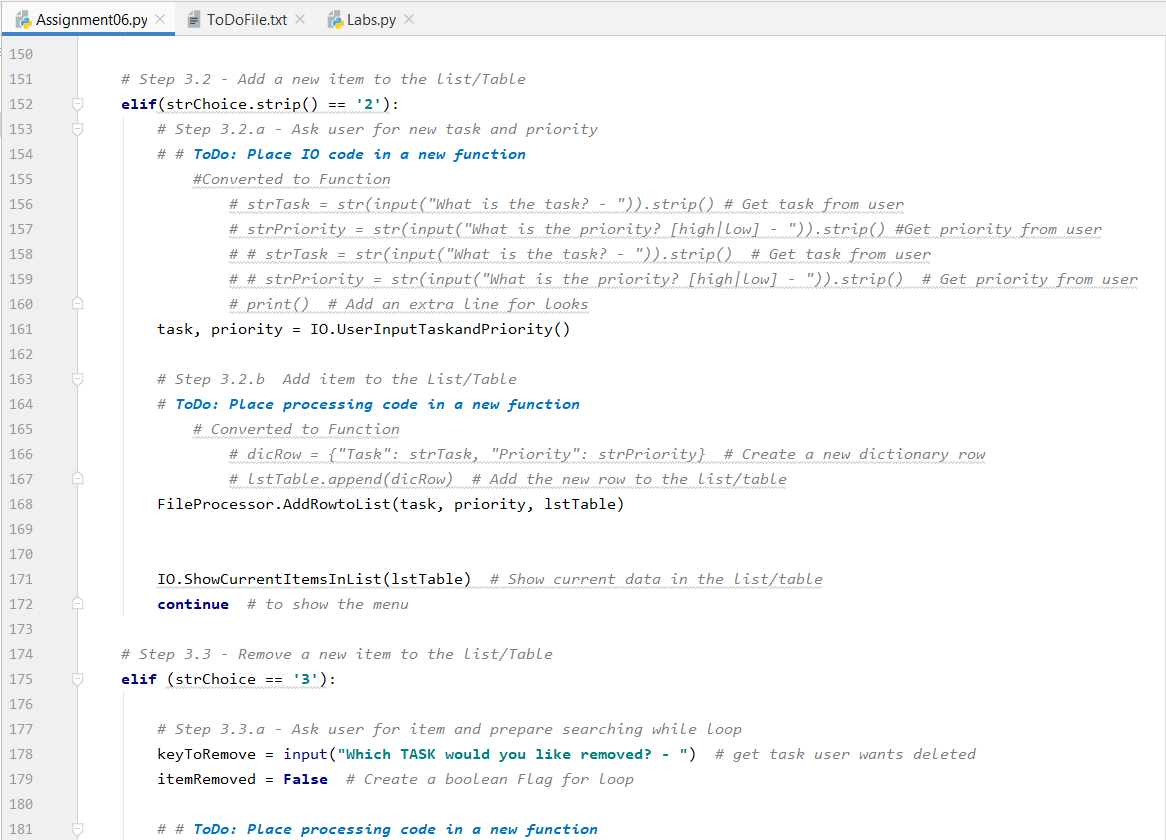
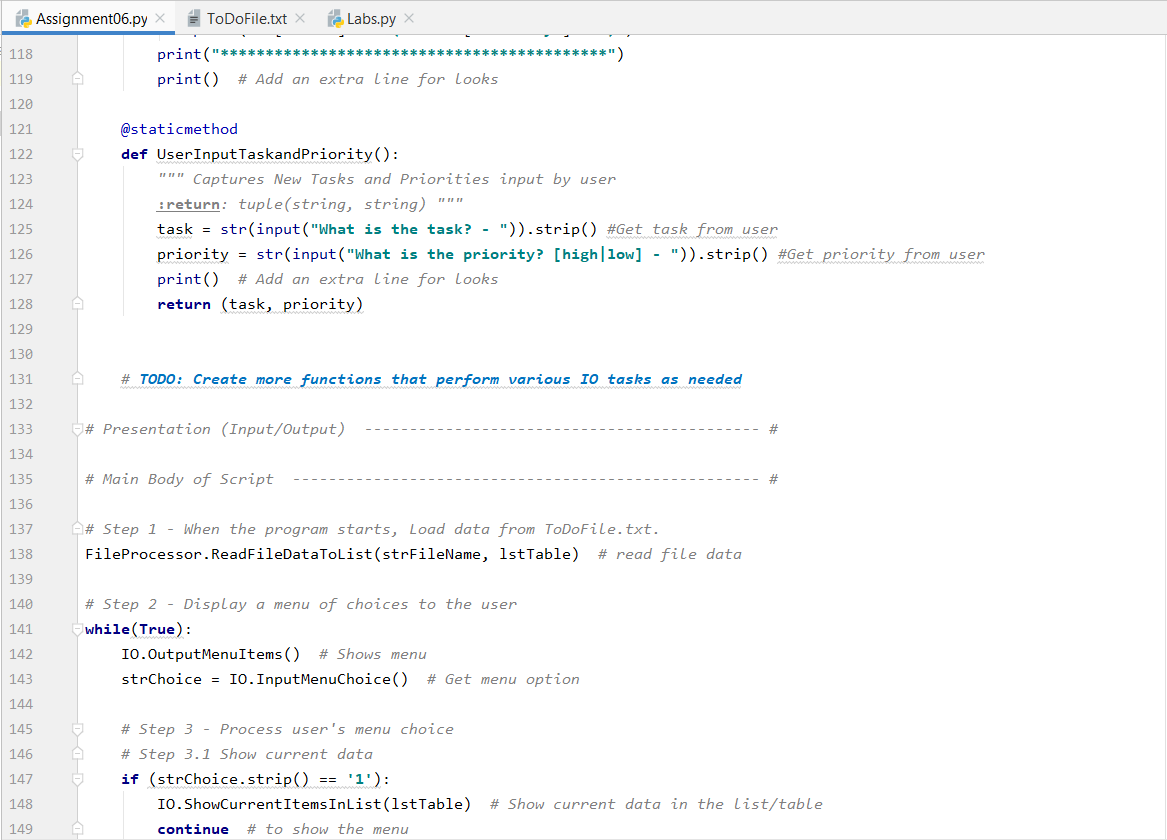
Figures 3.0, 3.1, 3.2 depict the program script I modified for the To Do List activity and its output.



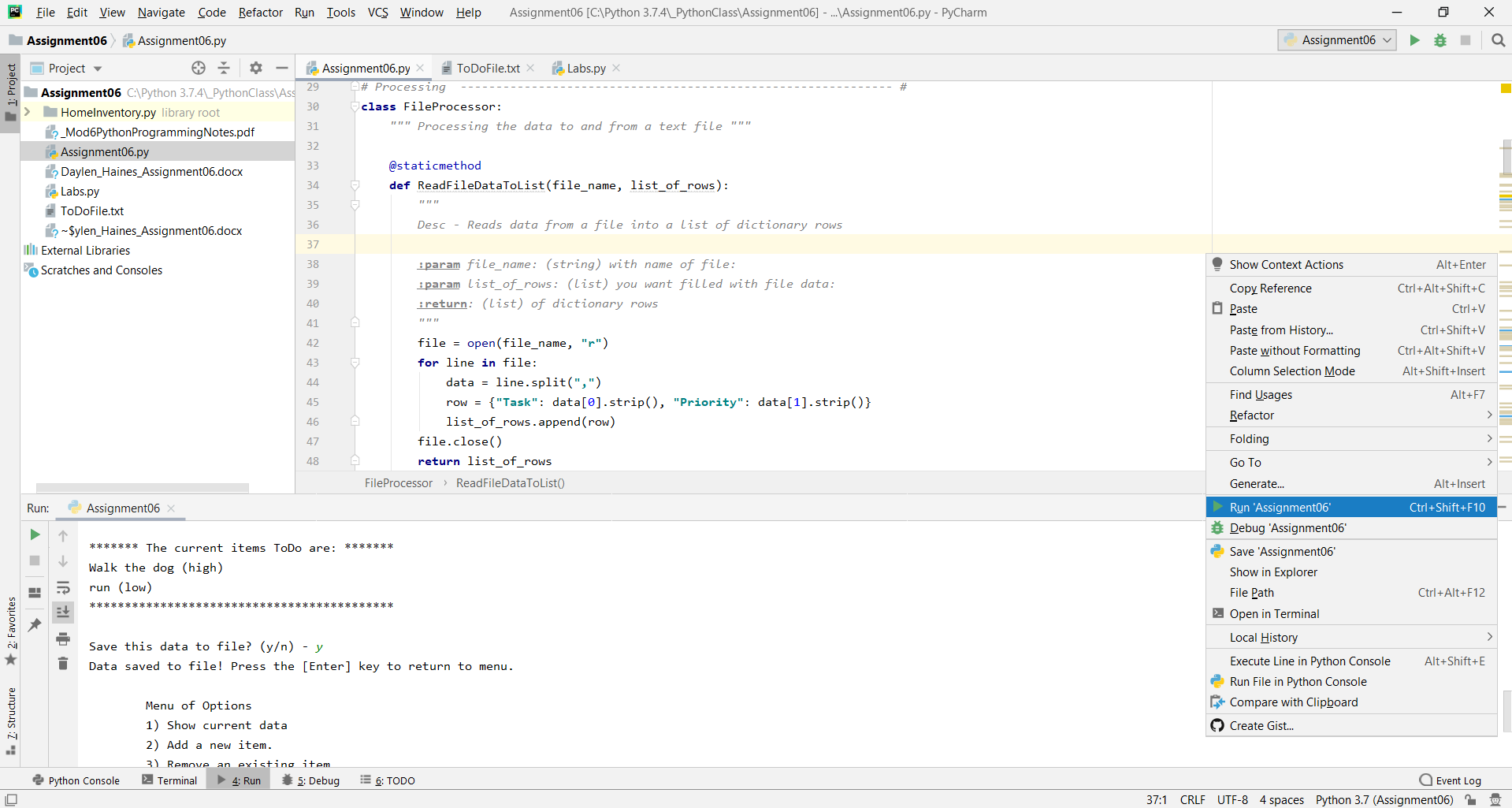




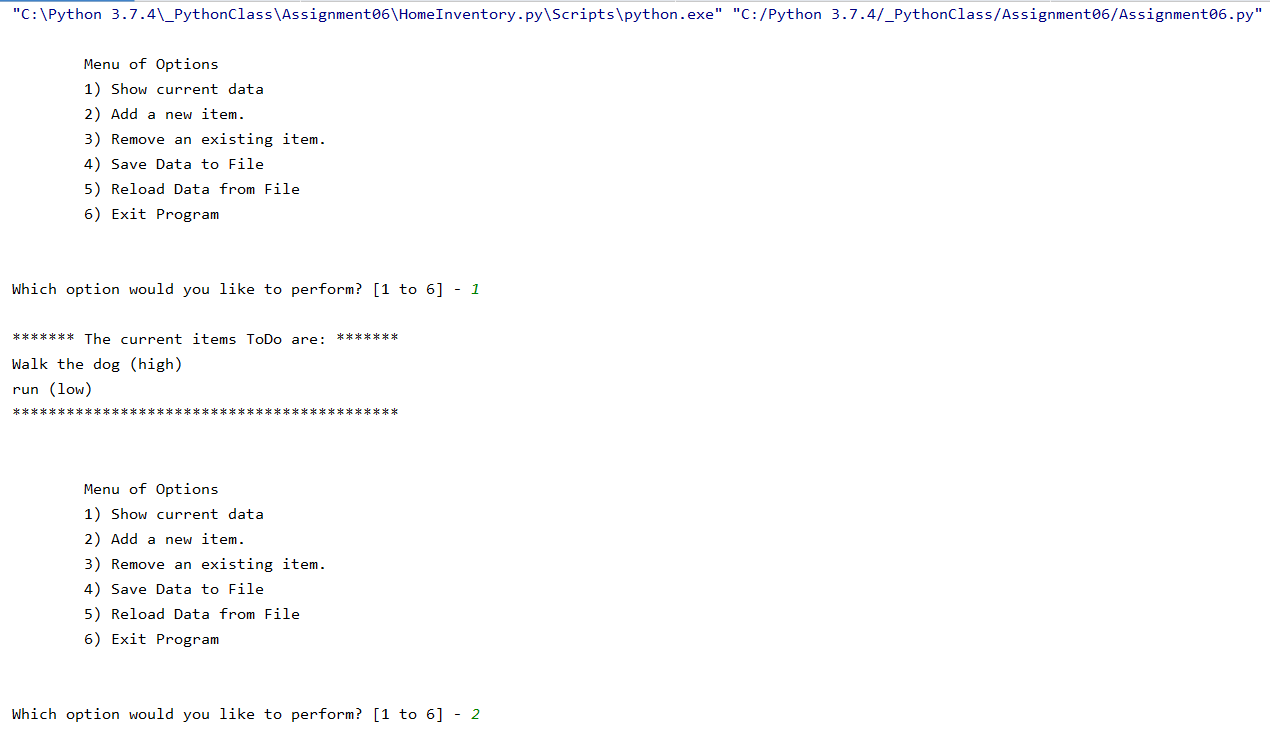


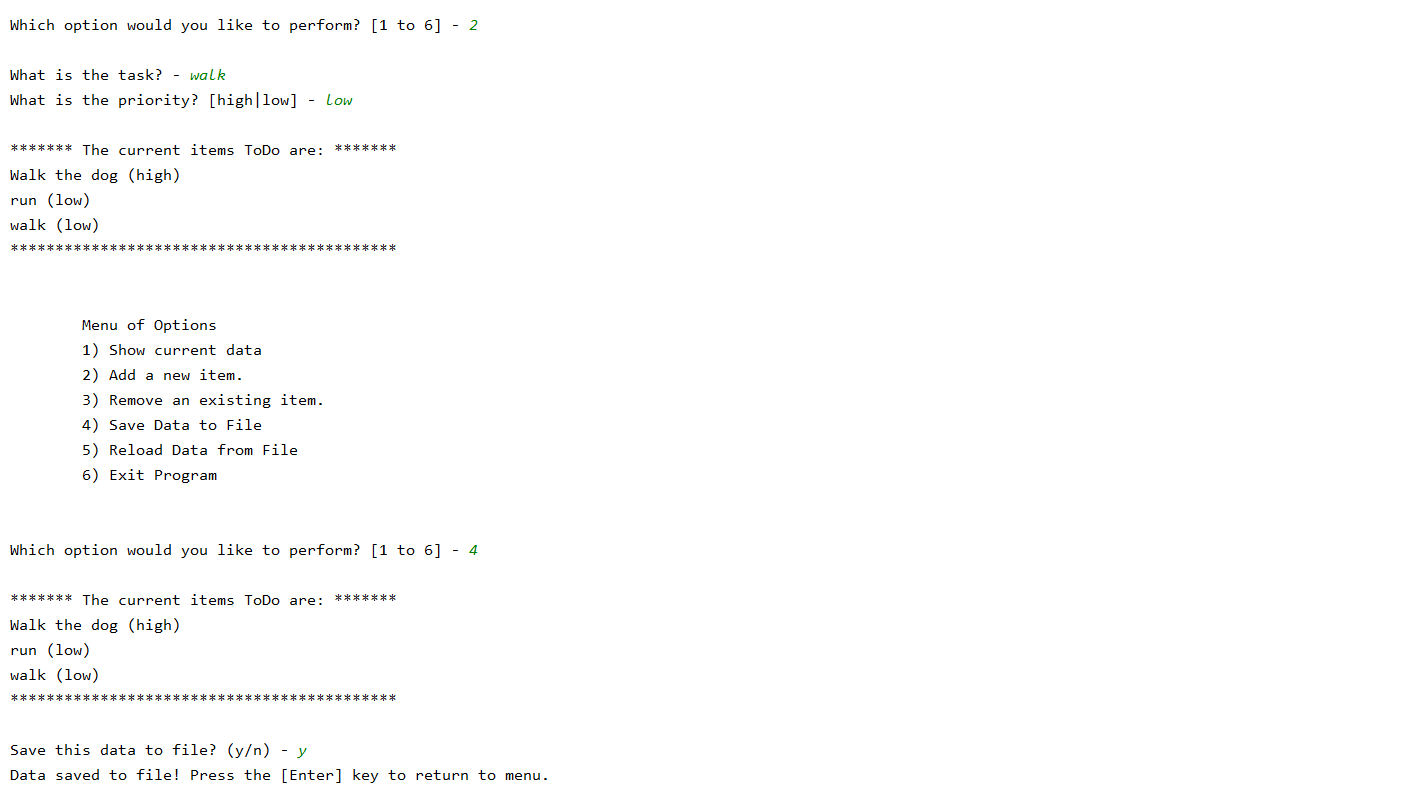


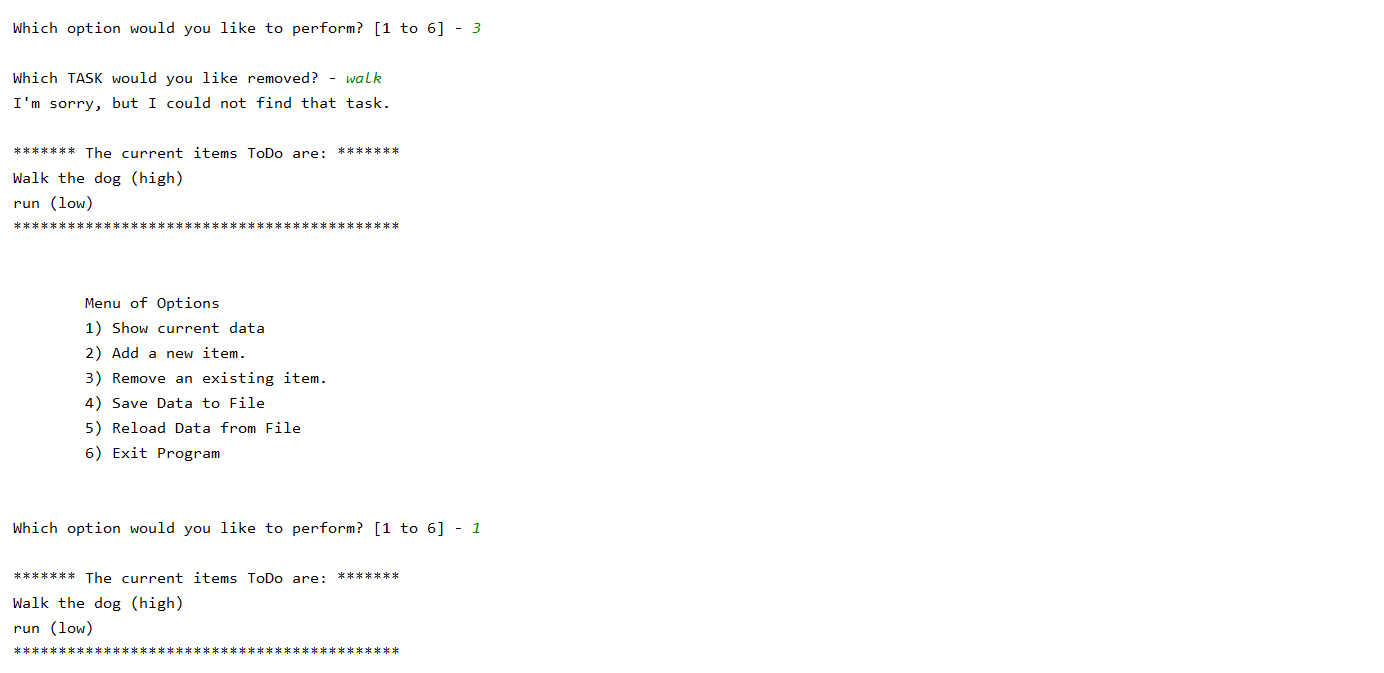
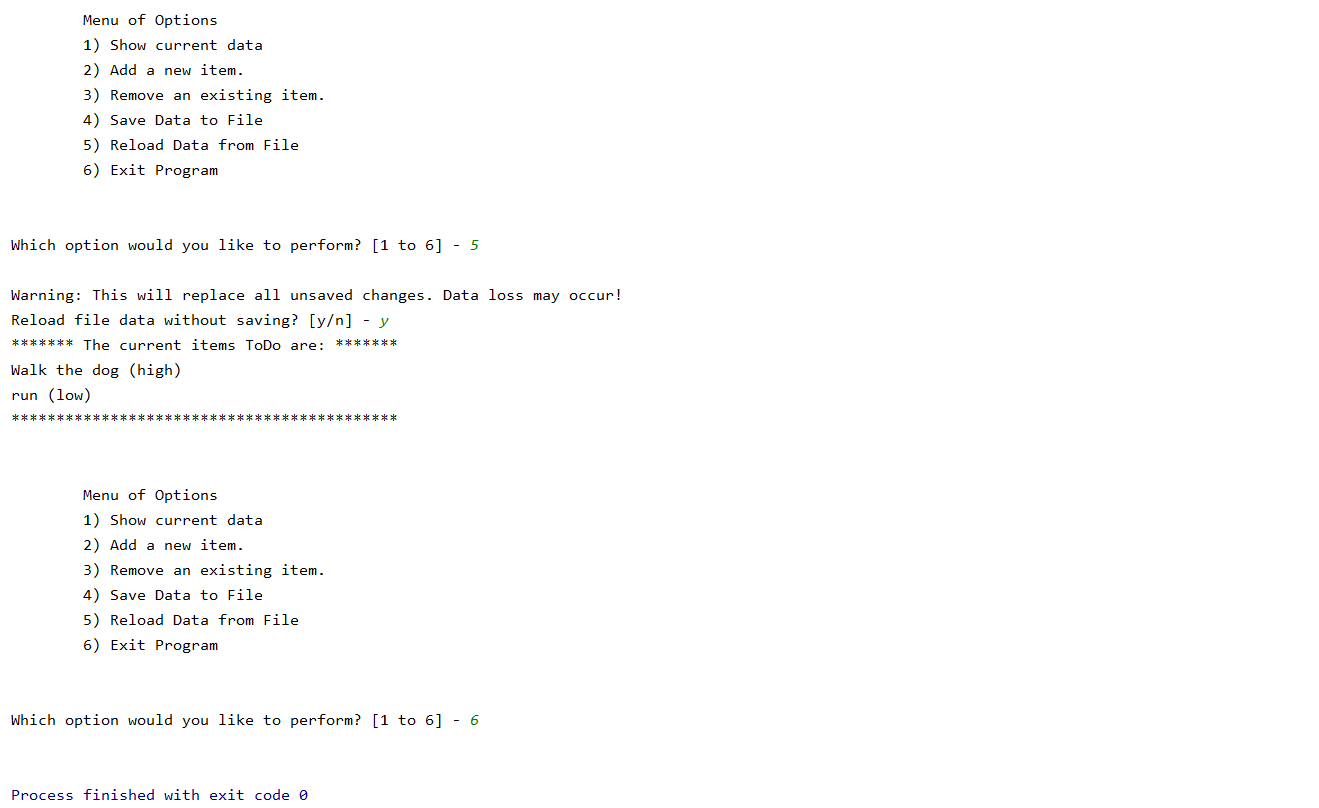
**Figures 3.0 – Showing the Test Python File I Modified in PyCharm for the To Do List Script**

****

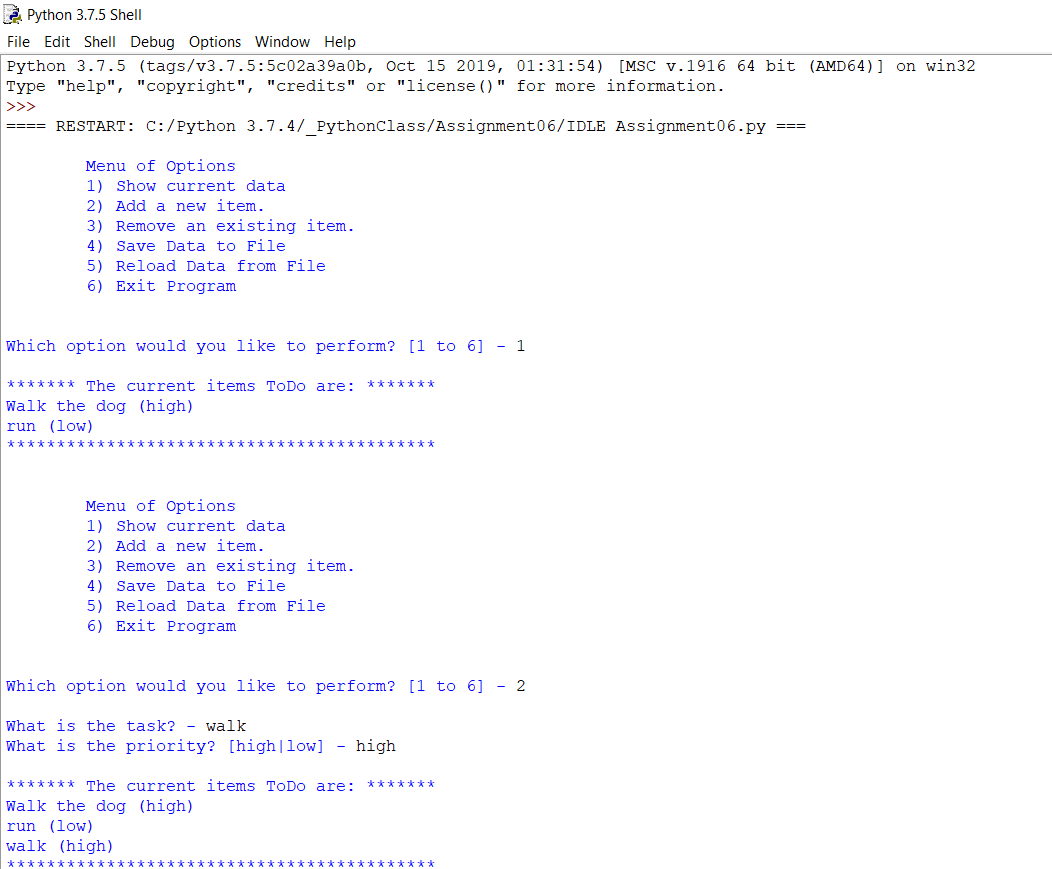
**Figure 3.1 – Showing the Test, Run, and Interactive PyCharm Functions for the To Do List Script**



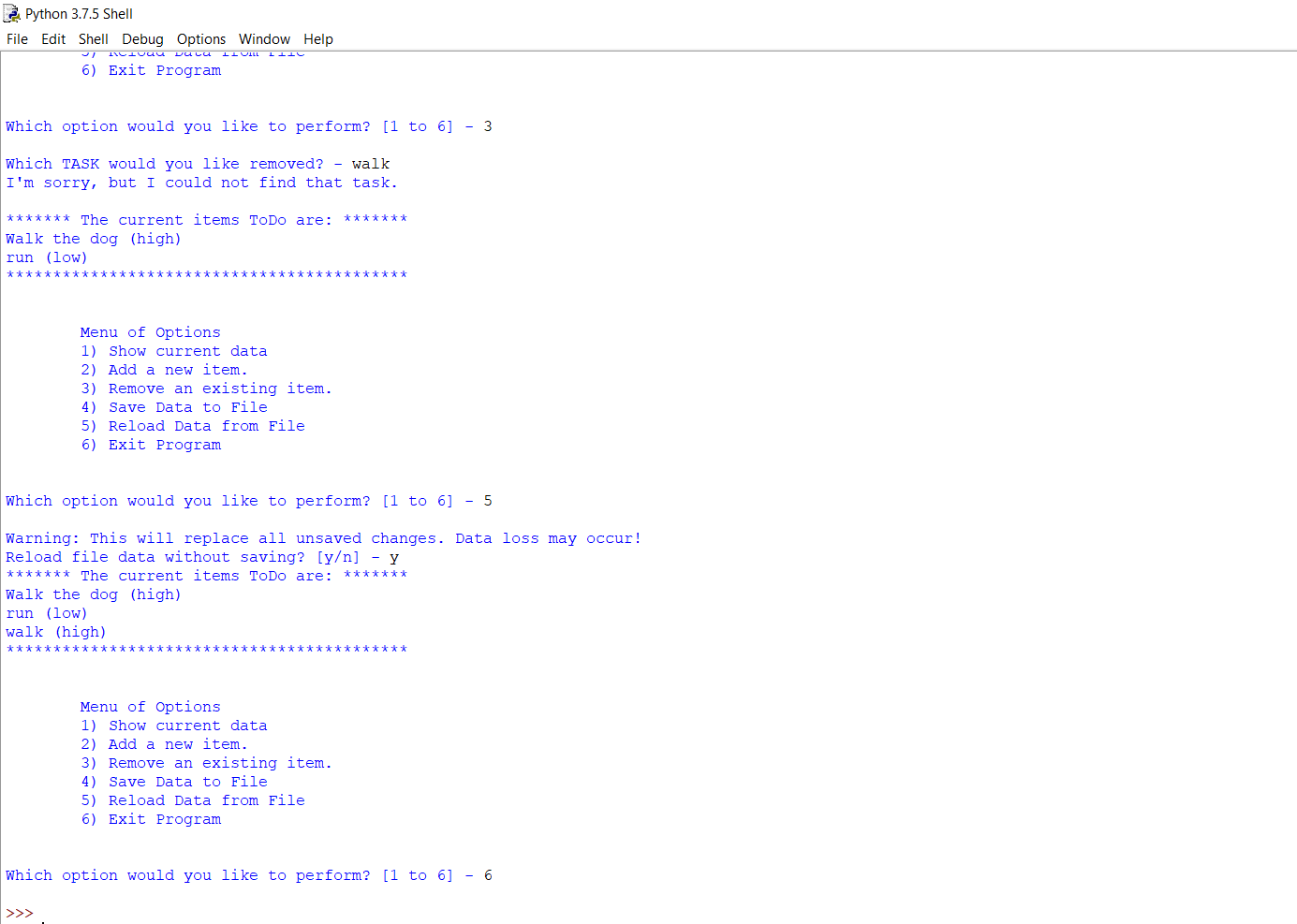


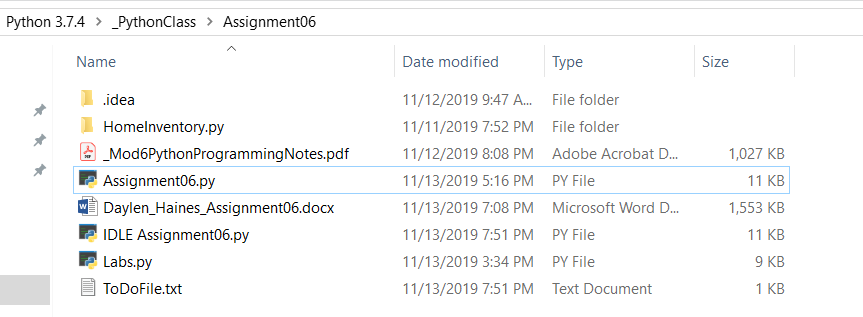
**Figure 3.2 – Showing the PyCharm Interactive Version of the Assignment06 Script**



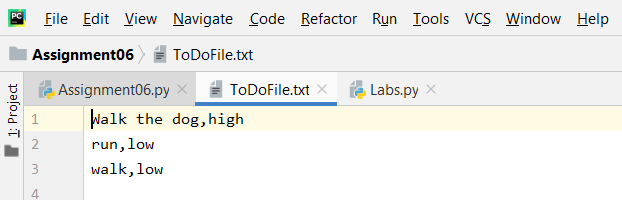


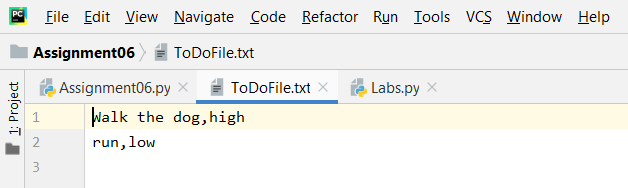


**Figure 3.3 – Showing the IDLE Command Shell To Assignment06**

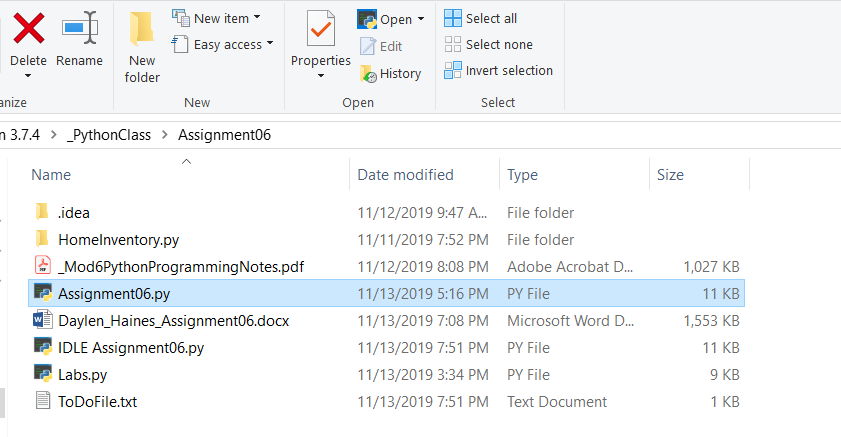


**Figure 3.4 – Showing the Assignment06 Folder**





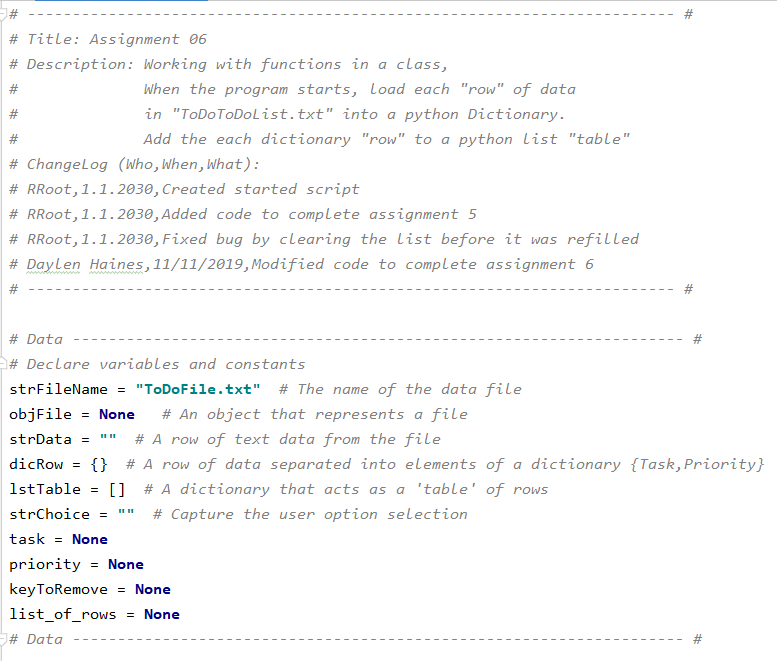
**Figure 3.5 – Showing the Text Input from the Assignment06 Script Adding and Removing (walk, low)**



**Figure 3.6 – Showing Accessing Command Shell for Assignment06 – Didn’t Run for Some Reason**

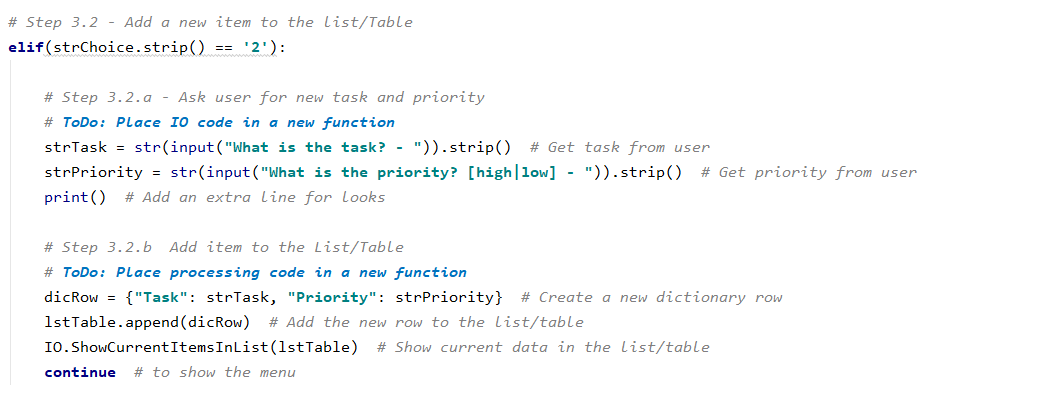
**Script Details**

I first created a new project in PyCharm located in my PythonClass Assignment06 folder. Next, I created a new Python file named Assignment06.py in PyCharm. Now that I was up, running, and saved, I began writing my script. I began by copying and pasting the Assignment06\_Stater.py script that professor Root provided into my ToDoList PyCharm project (https://canvas.uw.edu/courses/1342958/files/59683877?module\_item\_id=9971459 (External Site). Once the starter text was pasted into PyCharm, I began updating the key information, pseudo comments, and declaring the variables and constants that will be utilized throughout the rest of the script (Figure 4.0).



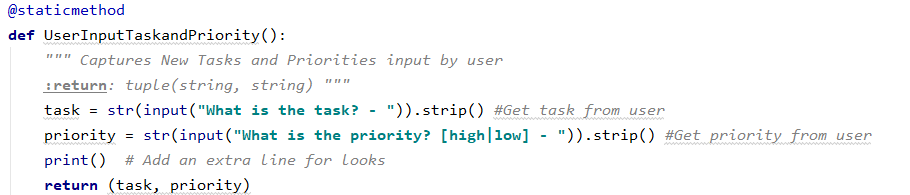
**Figure 4.0 – ChangLog updates and Declaring Variables**

Once all of the variables and constants were declared for the script, I was ready to test run the script and ensure that the script had full functionality before I began adding functions. When running the script, I received an error because the strFileName = ToDoFile.txt had not been created. I created the ToDoFile.txt in my Assignment06 folder and then re-ran the script. This time the script ran and functioned correctly. I was able to perform every Menu Option as defined in the script. Now that I knew the script functioned as intended, I was ready to further review the Assingment06 Starter script and see where I could make modifications to include additional functions. In going through the script, I noticed that there were some highlighted areas for us to focus on. The first area identified in the script was under Step 3.2 - Add a new item to the list/ table. Where we were to place an IO code in a new function and place a processing code in a new function



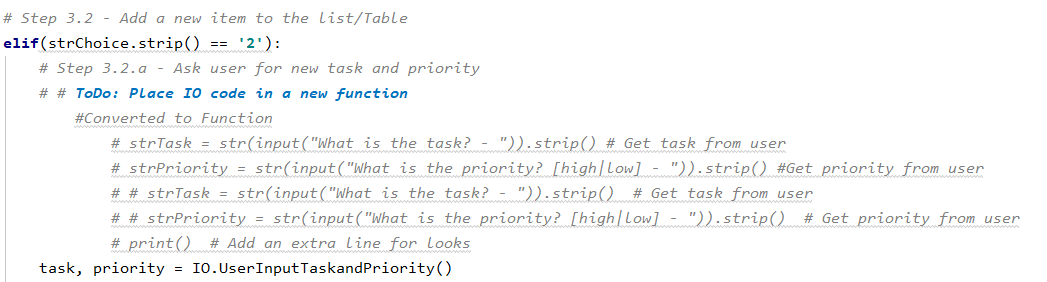
**Figure 4.1 – Add New Item to List/Table Original Script**

I first started with placing an IO code in a new function. To do this, I copied the script under Step 3.2 a, navigated up to the Presentation Input/ Output section of the script, located the last current function, created a new line below, and pasted the script. I then navigated to the top of the newly pasted script and started to define the function. Once the function was properly defined, I added what I would like the function to return which is the newly added task and priority. Now that the function was created, I needed to go back down to the main body of script and modify the script to reference the new function.



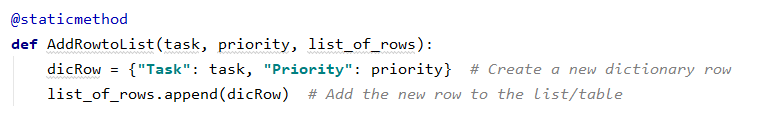
**Figure 4.2 – Add New Item to List/Table def and Function**

Once I navigated to step 3.2.a, I noted that I converted the script to a function, then commented out the original script. I then replaced it with the reference to the newly created UserInputTaskandPriority function. Next, I ran the script to ensure that the function worked appropriately and moved on to the next function task of placing a processing code in a new function.



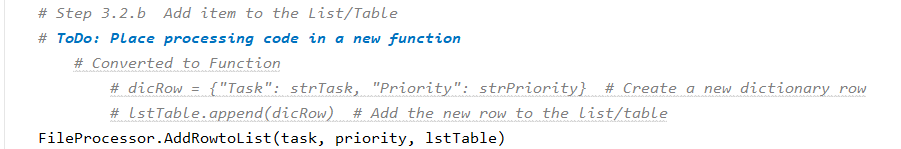
**Figure 4.3 – Add New Item to List/Table Function Inserted in Script**

To create a processing code for Adding items to List/ Table, I copied the script under 3.2 b, navigated to the Processing section of the script, located the last current function, created a new line below, and pasted the script. I then navigated to the top of the newly pasted script and started to define the function. Once the function was properly defined appending the newly added tasks and priorities to the list of rows, I needed to go back down to the main body of the script and modified the script to reference the new function.

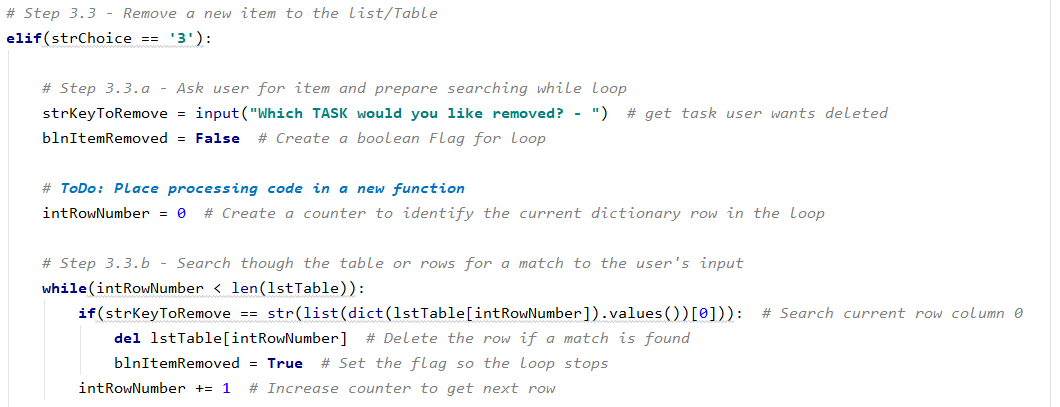


**Figure 4.4 – Add Item to List/Table def and Function**

Once I navigated to step 3.2 b, I noted that I converted the script to a function, then commented out the original script. I then replaced it with the reference to the newly created AddRowtoList function. Next, I ran the script to ensure that the function worked appropriately and moved on to the next function task of placing a processing code in a new function for Removing a new item from the List/ Table, Step 3.3b

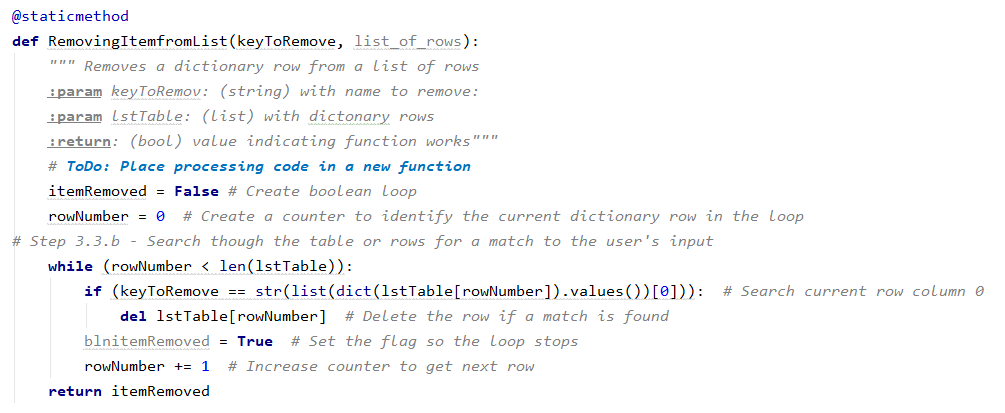


**Figure 4.5 – Add Item to List/Table Function Inserted in Script**

To create this processing code for removing a new item to the List/ Table, I copied the script under 3.3a, 3.3b, and navigated to the Processing section of the script. I located the last current function, created a new line below, and pasted the scripts. I then navigated to the top of the newly pasted script and started to define the function. 

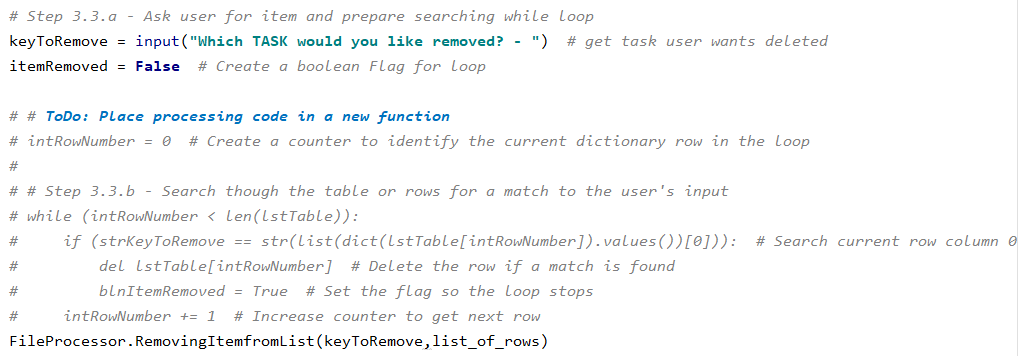
**Figure 4.6 – Remove New Item to List/Table Original Script**

Once the function was defined, (RemovingItemfromList) to remove a given task and priority from the table, I went back down to the main body of the script and modified the script to reference the new function.



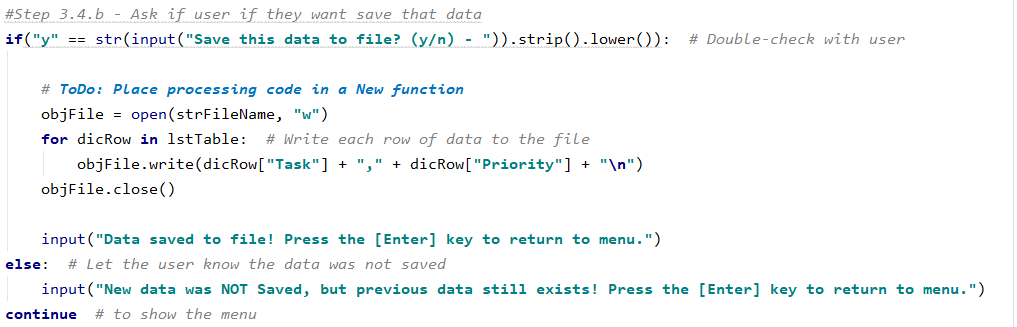
**Figure 4.7 – Remove Item to List/Table def and Function**

Once I navigated to step 3.3, I noted that I converted the script to a function, then commented out the original script. I then replaced it with the reference to the newly created RemovingItemfromList function. Next, I ran the script to ensure that all the functions worked correctly.



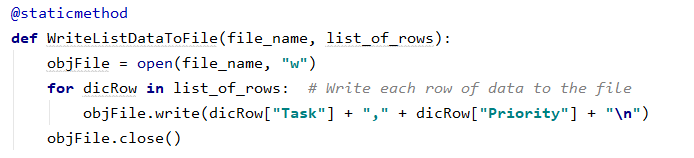
**Figure 4.8 – Add Item to List/Table Function Inserted in Script**

Once the script ran correctly, I then went on to the next area identified in the script for adding a function which was section 3.4 – Save Tasks to the ToDoFile.txt. Where we are to place a processing code in a new function, asking the user if they want to save their data?



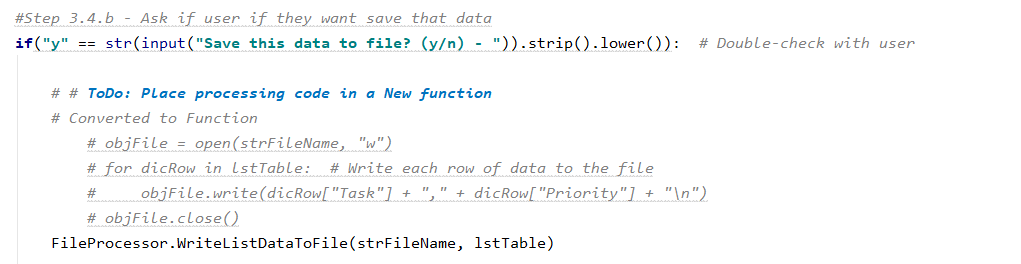
**Figure 4.9 – Ask User if They Want to Save Data Original Script**

To do this I copied the script under Step 3.4b, navigated to the Processing section of the script, located an open space within the current functions, created a new line below, and pasted the script. I then navigated to the top of the newly pasted script and started to define the function. Once the function was properly defined (WriteListDatatoFile) to write and save the tasks and priorities to the list of rows, I went back down to the main body of the script and modified the script to reference the new function.



**Figure 5.0 – Ask User if They Want to Save Data def and Function**

Once I navigated to step 3.4b, I noted that I converted the script to a function, then commented out the original script. I then replaced it with the reference to the newly created WriteListDataToFile function. Next, I ran the script to ensure that all the functions worked correctly.



**Figure 5.1 – Ask User if They Want to Save Data Function Inserted in Script**

As I wrote each function I would utilize the debugging tools available in PyCharm to ensure that my scripts functioned correctly as I moved through each function. After I created all of the functions and placed them in the script for all the identified areas, I was ready to run the Assignment06 script in PyCharm. I right clicked and went to Run ‘Assignment06.py’ which put it in an interactive window within PyCharm (Figure 3.1). Here I could try out my script to make sure it functioned as intended (Figure 3.2). Once I was happy with my script and my ToDoFile.txt file was being written to as I desired (Figure 3.5), I ensured my file was saved so I could run it in Command Shell. To run my script in Command Shell, I accessed the folder location on my computer, highlight the Assignment06.py (Figure 3.6) file selected the Python Open drop down, selected Python 3.7 and this launched my script in the Command Shell. For some reason, my script would not run in Command Shell. It would run perfectly fine in PyCharm without errors but would automatically close in Command Shell without prompting any inputs. To see if there was something wrong with my code, I ran the file in IDLE python Shell (Figure 3.3) where again the script function as intended. After hours of reviewing class notes, searching the web, and trying to rewrite the file, I was unsuccessful with getting the script to run in the Command Shell. However, I was able to ensure the user inputs were captured as desired through IDLE Shell and PyCharm, by opening the ToDoFile.txt, file and making sure the inputs were captured correctly.

**Summary**

Python has proven to be powerful, and user friendly programming language for creating interactive programs where you can access txt files, capture, display, save, and write user inputs. Utilizing what we learned in Modules01, Modules02, Modules03, Modules04, Module 05 and expanding our knowledge base to include functions. That being said, there were again issues with my files running in Command Shell. Due to two weeks in a row of this, I am going to re-install Python to see if that corrects the issue. Going forward with the guidance of Professor Root and Author Dawson, I am confident that I will be able to continue to expand my Python programming capabilities and successfully write more complex scripts and programs.

**GitHub**

GitHub - <https://dhaines44.github.io/IntroToProg-Python-Mod6/>

