Thivanka Samaranayake

May 17, 2020

Foundations of Programming (Python)

Assignment05

Using Dictionaries and Lists to Build Programs

# Introduction

This assignment outlines the steps for using lists and dictionaries to create a script that takes simple inputs from the user to store information in a text file. This module also explores the use of templates. The starting code outline for the script is given. The goal was to write in the specifics of the code based on the template. Finally, this module introduced GitHub. This data repository is useful in sharing and managing coding projects with colleagues and oneself.

# Lists and Dictionaries

Lists are a way to hold collections of data within Python. Lists are very similar to tuples. However lists are mutable. In another words, lists can be modified by adding or deleting the information within them. Lists are defined in Python using square brackets. Figure 1 demonstrates how data can be inputted into a list. Dictionaries are much like lists. However, unlike a list dictionary allow you to store information in pairs as opposed to in a sequence. Dictionaries hold a key and its value in the same way that a row of data is stored in Excel. Dictionaries replace the **index** subscripts with **key** subscripts. This can be powerful since an index has limitations. Index sequencing is numerical. Key subscripts are made out of strings. Having string keys provides more insight into the nature of data. Dictionaries are defined in Python using curly brackets.

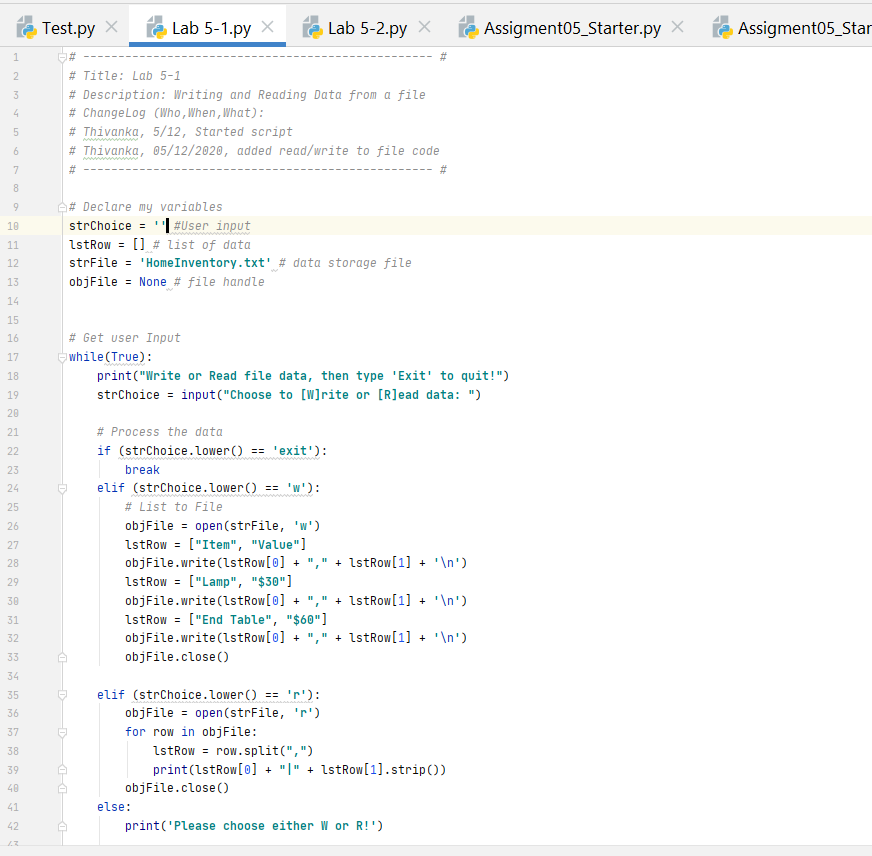
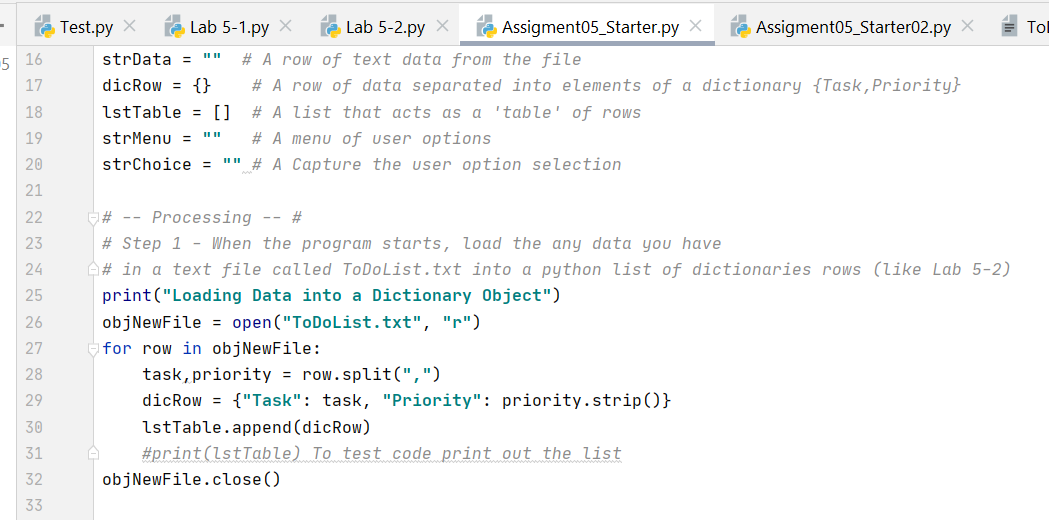


Figure 1—Storing Data in a List

# Reading Data into Lists and Dictionaries

Both lists and dictionaries have a suite of methods that accompany them. In order to fully utilize these methods, it is useful to read data into lists and dictionaries. Both can read data from text files. Common way to read data from a file into a list or dictionary is to use a for loop. Using a for loop allows the information to be passed into the list or dictionary row by row. This can be specially powerful when the data entered can be differentiated using keys. For example in the homework exercise for this module, we are entering data in the form of a “Task” and a “Priority.” This type of data is specially powerful when read into a dictionary. See Figure 2 for how data from a text file was read using a dictionary.



**Figure 2**—Step 1 is reading data from text file into a dictionary and storing it in a list

# Separation of Concerns

As code gets more and more complex, the need to organize it becomes important. Separation of concerns is one method for improving scripts. According to Wikipedia, separation of concerns is the design principle of diving up a code into major distinct sections. This allows for greater degrees of freedom for the program design, deployment, or usage [1]. For example one method of dividing up a script is Data, Processing and Input/Output. This organizes the data and makes it easy to follow along, further refinement, and adding follow on code.

# Functions

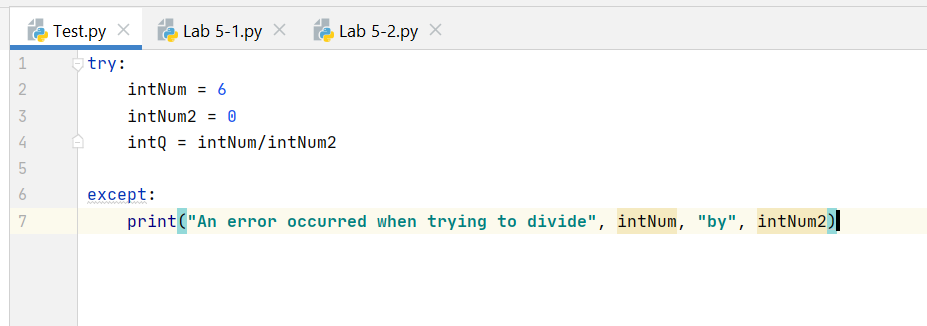
Functions allow you to group and organize a set of programming statements under one command. Then by invoking the command, the function allows the set of programming statements to be called. In Python the function must be defined first before the script can be called. When Python gets to the line that calls the function, it jumps to the area of the program where the function is defined and runs through the functions’ programming statements. After doing so, it typically returns a value that is then used to run the rest of the code in the program. Functions can be a powerful tool to organize and structure data. Functions go hand in hand with the concept of Separation of Concerns.

# Script Templates

Script templates are aides used in coding to organize data and provide consistency. They also serve the dual role of making code look more professional and easier to read. PyCharm allows for easy creation of templates. Script templates can be as simple as saving the comments at the beginning of a program to be used for future programs. Or script template can include the pseudo code and main sections of the code that can be divided up in a program.

# Error Handling

Error handling is an important aspect of coding. It provides useful information to the user of the program on how to resolve errors to continue with the program. Most times humans cannot decipher the error messages given out by a computer. Thus it is important to program in user-friendly error messages. It also allows an organized way of grouping statements to be processed. Try except is a error handling technique where the try allows the user to test the code for errors and the except allows the user to handle the error. This allows the programmer to trap errors. Final advantage in error handling is the ability to move statements into another area of the program to specifically handle the error. That way the program doesn’t get bogged down and stuck on the error. Figure 3 shows an example of error handling in the program.



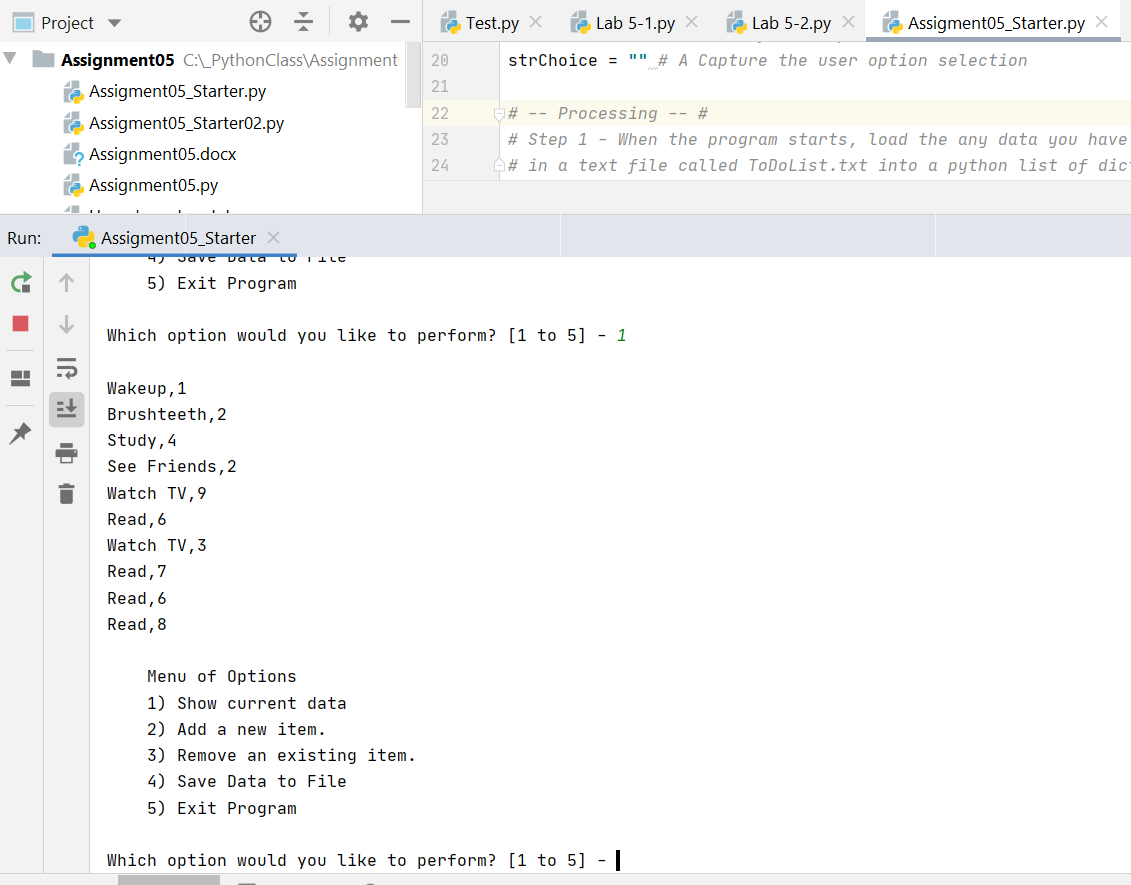
**Figure 3**—Divide by 0 Error Handling

# GitHub

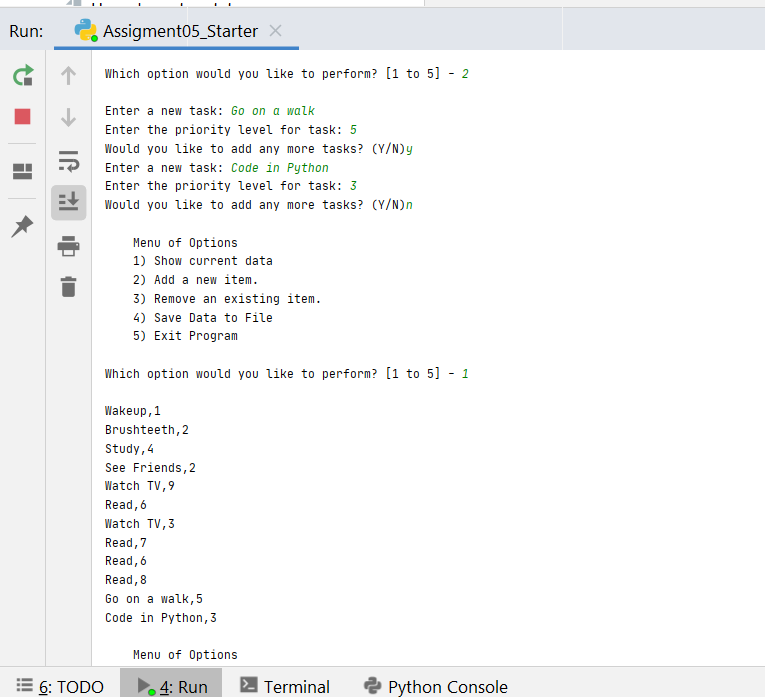
GitHub is a source control software that allows users to backup, revision control, and share coding data. GitHub keeps the revisions of code so that users can incrementally develop a program safely without fear of loss of data. Furthermore, it makes collaboration easy where coding data is stored in a central location for all team members to access it. GitHub also allows other contributors to add, download, or make changes to existing code. These features of GitHub make collaboration much easier.

# Assignment05

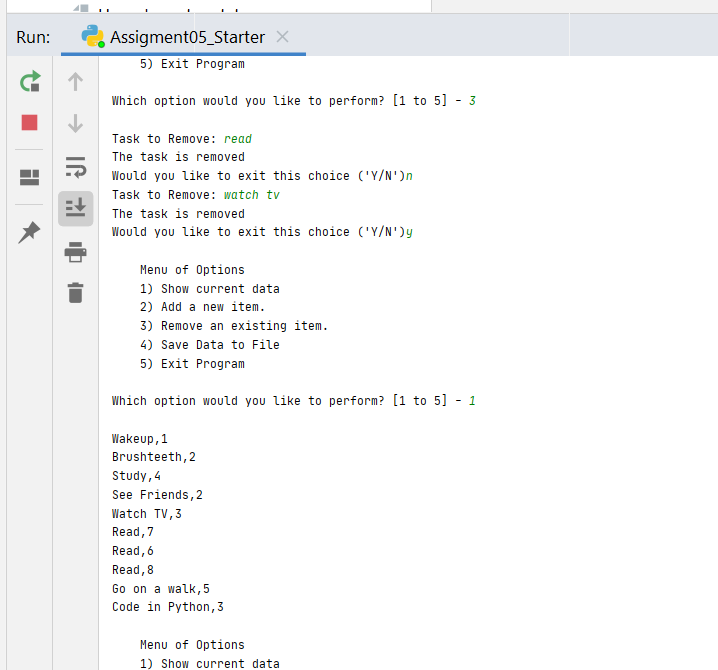
The program shows a menu of items that the user can use to read, add, delete, or store information. Each of the functions are demonstrated in the following figures.



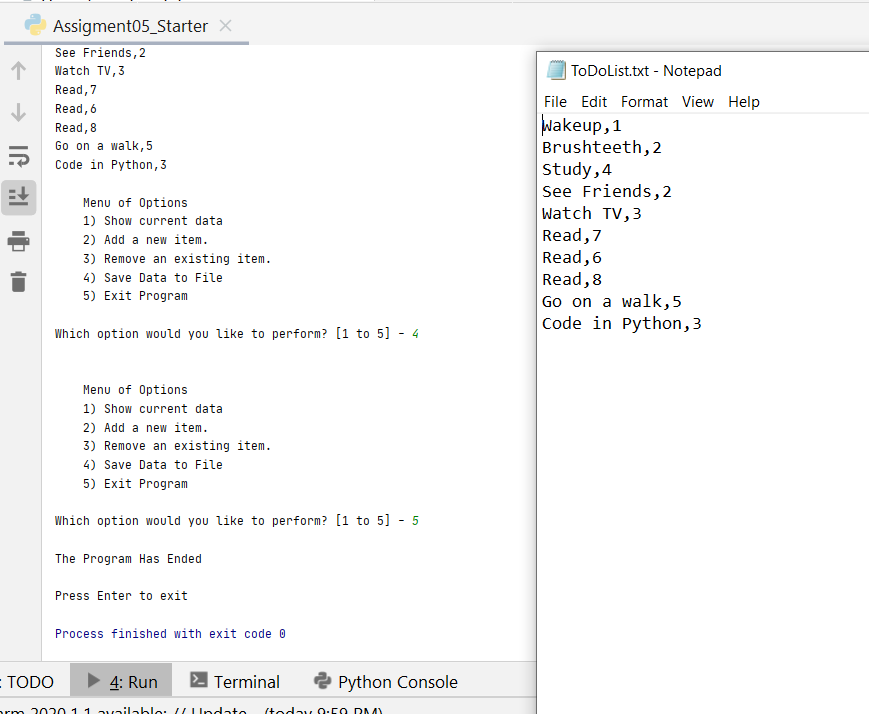
**Figure 4**—Reading Data



**Figure 5**—Add Data



**Figure 6**—Deleting Data



**Figure 7**—Saving Data

# Conclusion

This module was about organizing data whether from user input or program developed using lists and dictionaries. While each method had their advantages and disadvantages, using both in combination was a powerful method of storing data into a “table format.” Additionally, templates were used to organize scripts. This coincided with the concept of separation of concerns where code was divided into major parts to make it easier to modify and read. Functions were introduced as a way to implement code in a more efficient way. Error handling techniques were addressed to make code more user friendly. Finally, GitHub was introduced as a collaboration and source control tool for scripts.