**Name: Stephanie Yong**

**Date: August 9, 2022**

**Course: Foundations of Programming: Python**

**Assignment: Document Your Knowledge – Module05 – Lists and Dictionaries**

**Document Your Knowledge – Module05 – Lists and Dictionaries**

**Introduction**

In Module05 – Lists and Dictionaries, utilizing an assignment that is similar to Assignment04, the main focus for Assignmen05 is to build a script with dictionary.

**Getting Started**

Assignment05\_Starter.py has been provided as a base. Script was then built on top of the starter.py as shown in Figure #1:

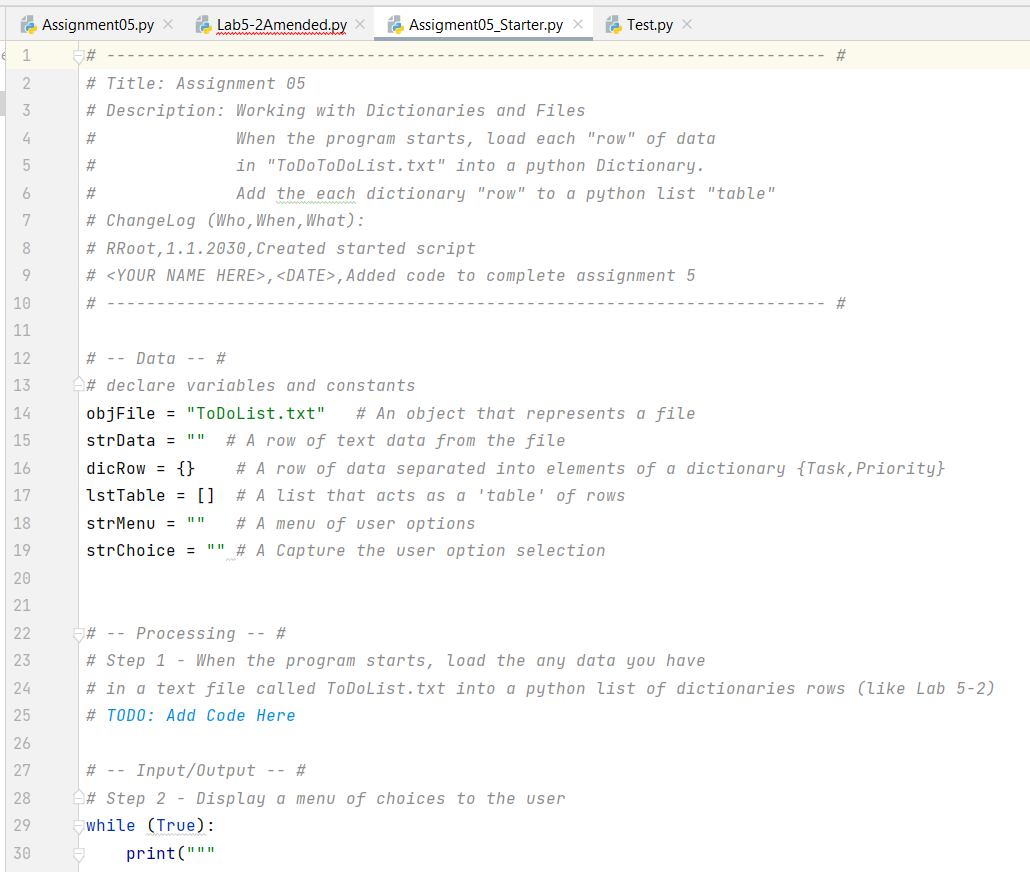


Figure #1

**Getting Started (continued)**

The starter.py reinforces a standard format and serves as a template for Assignment05. Within the starter.py, headers, pseudo codes, etc. have already been included which has significantly reduced duplication of work as well as providing structure to the script.

**Defining Variables**

Variables in the starter.py have already predefined as shown in Figure #2:

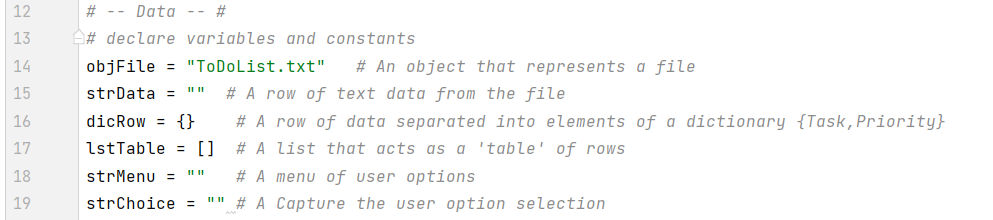
****

Figure #2

One of the difficulties working with someone else’s template is that codes and variables provided could create confusion. Some of the variables defined in the starter.py have not been used, like strData and strMenu.

**Separation of Concerns**

The idea of separation of concerns is introduced during Module05. Most programmer would divide the script into Data, Processing, and Input-Output as shown in the Figure #3 below:

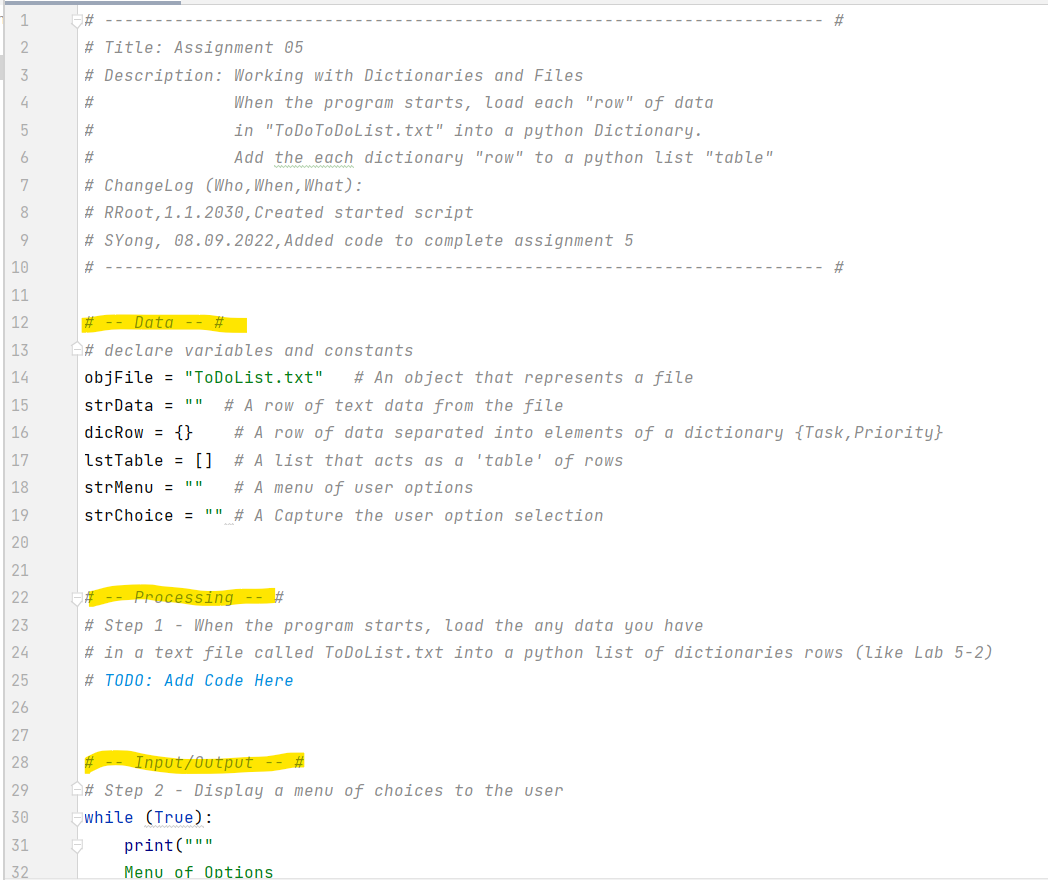
****

Figure #3

**Separation of Concern (continued)**

Since for Assignment05 would not be using any functions, the processing section is yet to be further discussed in Assignment06.

**While Loop and If…elif Statement**

Again, in Assignment05, while look and if elif statements are utilized for the logical flow of the script as shown in Figure #4:

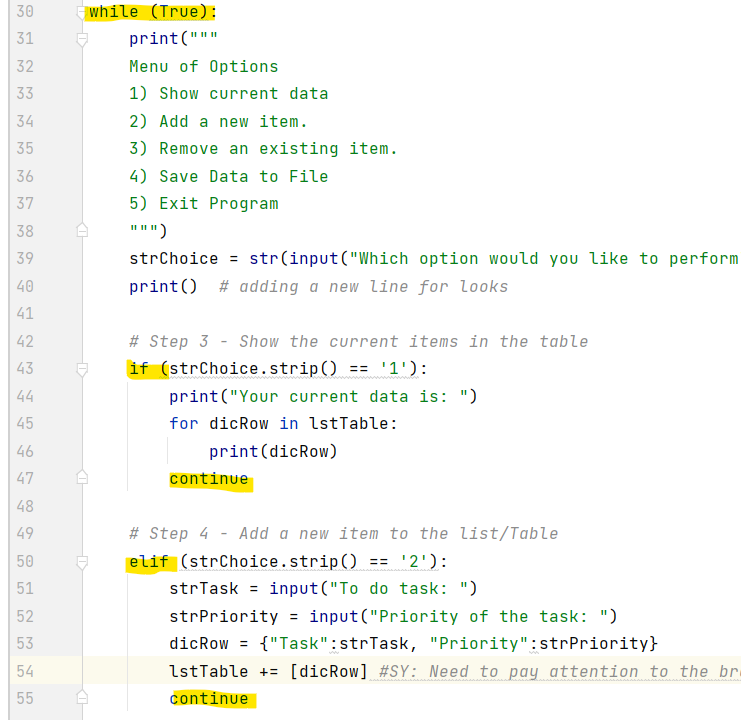


Figure #4

**While Loop and If…elif Statement (continued)**

The loop is ended with a “break” command at the very bottom of the script as shown in Figure #5:

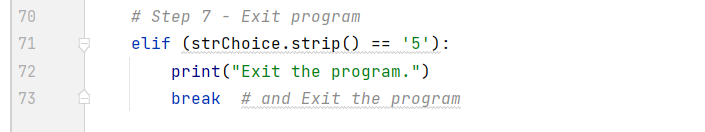


Figure #5

**Writing Data to a File**

In referencing to Figure #6 below, in row 64, when ToDoList.txt is used, in CMD environment, no text file would be created when script is run (though it works in PyCharm). So, to fix the problem, for the script to generate the text file in both the CMD environment as well as PyCharm, location is specified as shown in Figure #7.

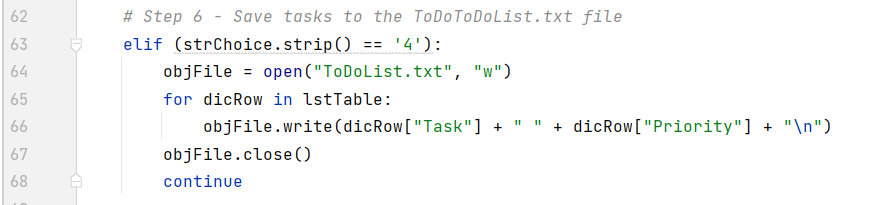


Figure #6

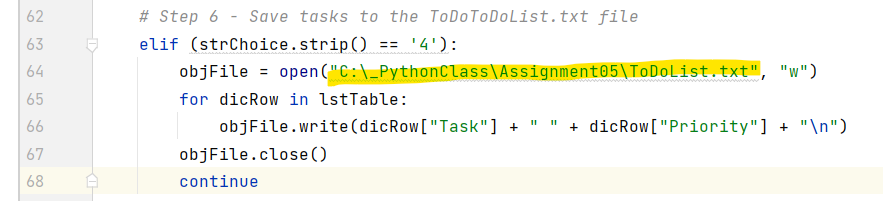
****

Figure #7

After “open” and “write” the data onto the designated file, make sure to “close” the file accordingly.

The script will keep adding data to the txt file with letter “a”. However, by using letter “w”, the script would overwrite the data instead.

**Dictionary**

One of the biggest distinguishment for dictionary is that it uses { }. Also, for dictionary, it uses key (character) subscripts instead of index subscripts. But other than that, at this point, personally cannot tell the differences between dictionary, list, tuple, or string. They all seems like being used for similar purposes and not sure why needing that many different types.

Building on top of what have been learned from Assignment04, in Assignment05:

1. Define the two variables, item and value
2. Then concatenate the two into a Row/Data
3. Creating a table with the rows/data
4. Add rows to table accordingly

But in Assignment05, dicRow is created which is specified with { }. Then, dicRow adds to lstTable which is a list specified by [ ] as shown in Figure #8 below:

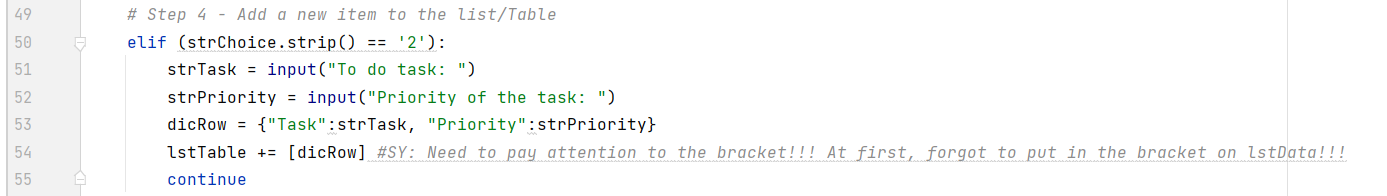
****

Figure #8

One of the challenges, is that when creating the lstTable, [ ] needs to be added around the dicRow. Forgetting to add the [ ] around the dicRow, the script is not running or adding items as planned.

**Run/Test/Debut the Script**

When the script is ready, it is being tested in PyCharm by running the script.

A new item added as shown in Figure #9 below:

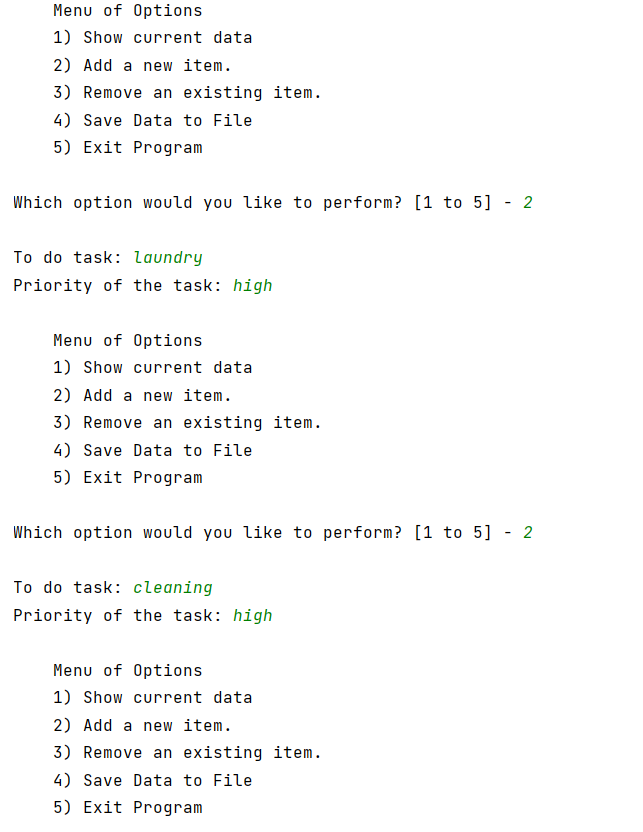
****

Figure #9

Inputting “3” to delete the latest item and input “1” would print out the current items in dictionary format Figure #10:

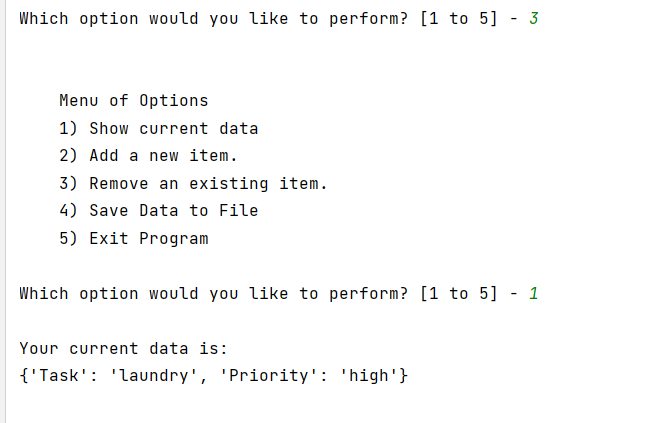
****

Figure #10

Inputting “4” as shown inFigure #11:

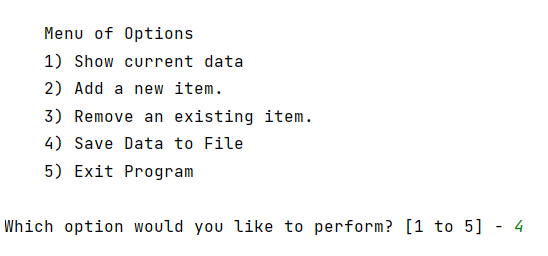
****

Figure #11

Would then generate a ToDoList.txt as shown in Figure #12:

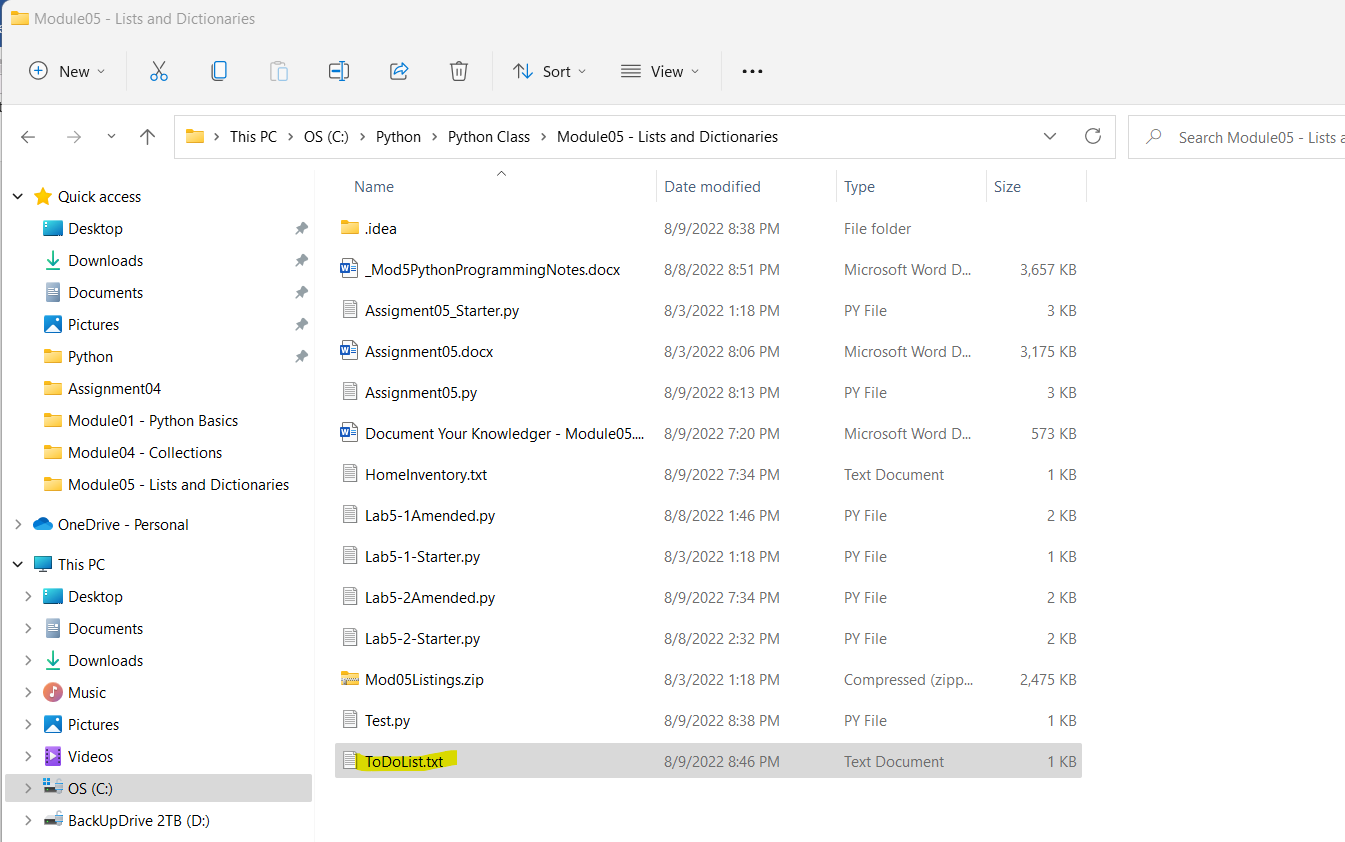


Figure #12

Within the ToDoList.txt, it would print out the items currently havingas shown in Figure #13:

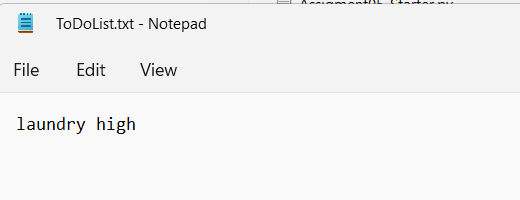


Figure #13

Figure #14 to Figure #18 show the script running and executing in CMD environment and corresponding text file is also found:

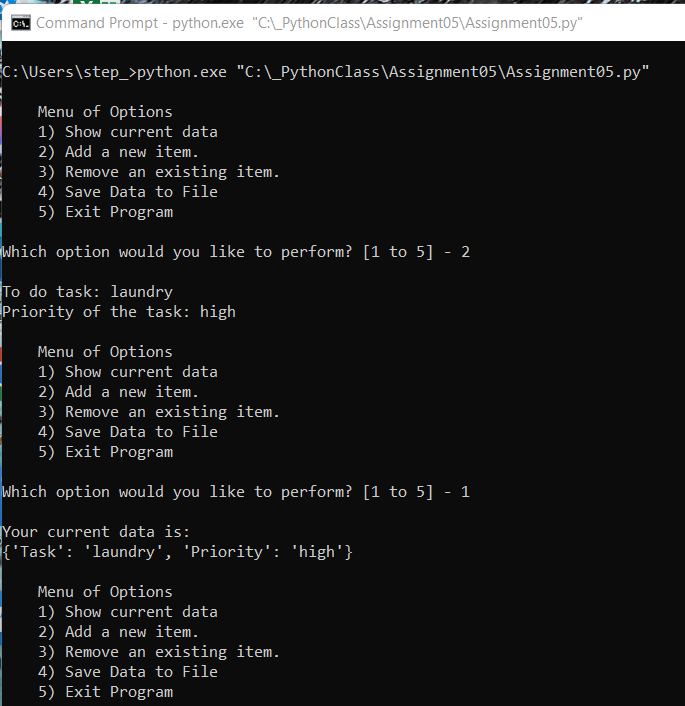
****

Figure #14

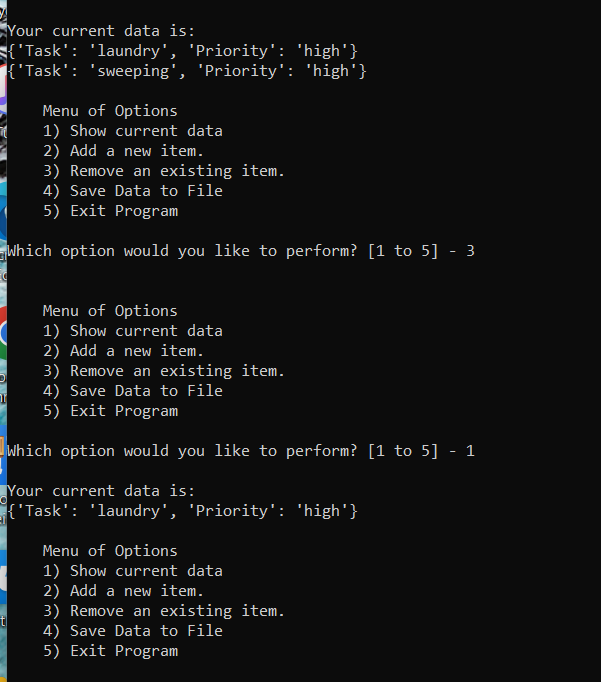


Figure #15

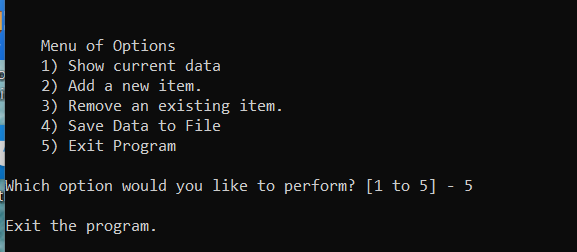


Figure #16

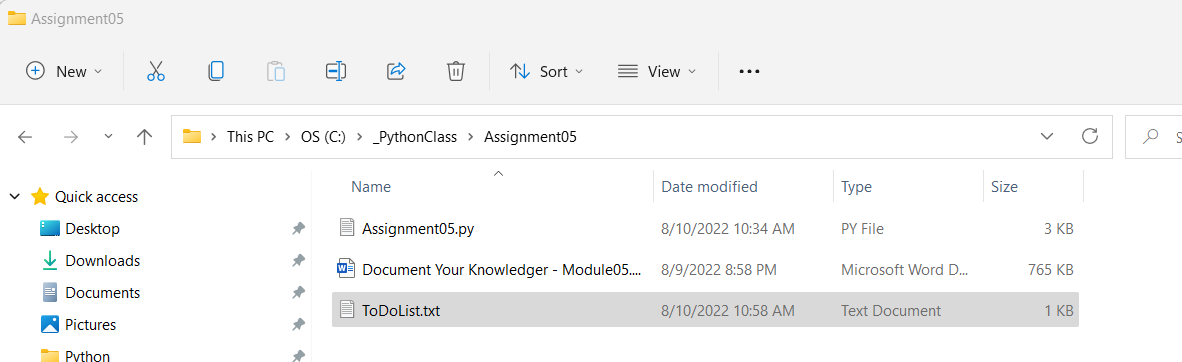


Figure #17

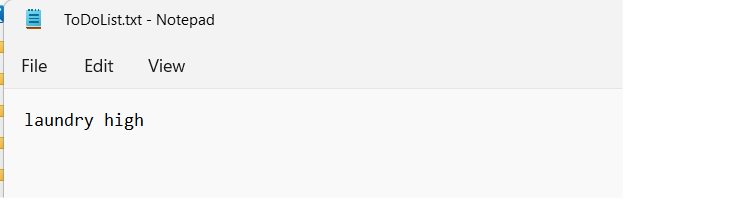


Figure #18

**Conclusion**

During Module05, dictionary is being incorporated into the script. And then rows of dictionary would then combine into a table coupling with all other functions like while loop, if-elif statement, on top of the most primitive command like “print” and “input”.