Laura Truong

April 14, 2020

Introduction to Programming

Assignment05

ToDo list using Dictionary in Python

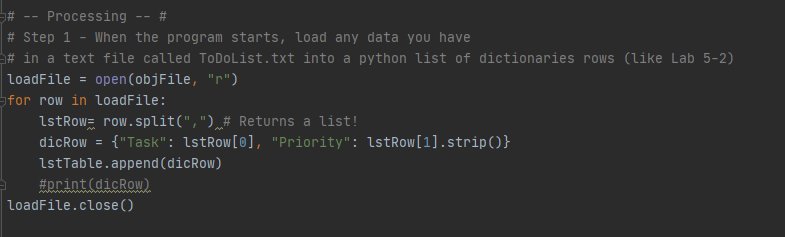
# Introduction

In this assignment, Students have been asked to modify a Python script file that manages a “ToDo list”. This “ToDo” file contains two columns of data “Task” and “Priority” that will be loaded into a Python Dictionary object. The rows of data will be added to a Python list object to create a table of data. This paper will outline the steps I took to create the code to add a new item to the list, remove an item from the list, save tasks to the “ToDoList.txt” file, show the current data, and exit the program. I will them run the script in a command shell from the Windows OS using Python version 3. We will begin by loading data in a text file when a program starts.

# Data

When the program starts, we will load any data we have in a text file called “ToDoList.txt” into a python list of dictionary rows. Since there is no data yet, I had to create a txt file in my folder location "C:\\_PythonClass\Assignment05\ToDoList.txt". In the txt file, each line has a Task and Priority separated by a comma. **Figure 1** shows the code reading the lines using lstRow and the split() function separating the items in the list based on the commas in each line. dicRow shows the keys “Task” and “Priority” being indexed with the strip() function to remove spaces at the beginning and end of each string. dicRow is then appended to a list table.

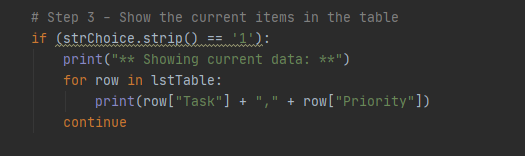
Figure 1



# Show the current items in the table

In this program, we are using a menu of options in a while loop to fulfill conditions that meet the criteria of the “==” operator. Showing the current items in the list is assigned to the value 1, so if strChoice which is the input of the user, equals 1, we will perform this menu item. To display the rows in the lstTable, we use a for loop as shown in **Figure 2** to run through all the rows in the table and print the dictionary item values in each row in the format of task, comma, then priority.

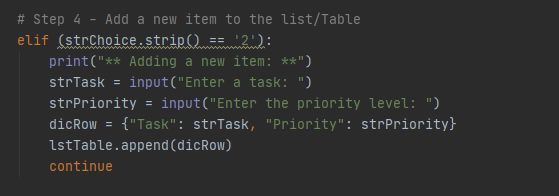
Figure 2



# Adding a new item to the list

Adding a new item to the list is assigned to the value 2, so if strChoice which is the input of the user equals 2, we will begin this task. To add a new item to the Table lstTable, we ask the user for two inputs, strTask which contains the input for a task, and strPriority, which contains the input for the priority level. dicRow will then create a dictionry with the the inputs of the user from strTask to “Task” and strPriority to “Priority”. dicRiow is then added to the end of the table lstTable using the append() method as seen in **Figure 3.**

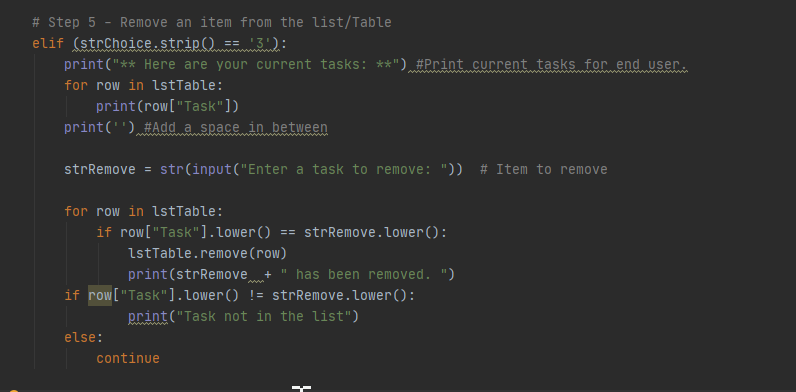
Figure 3



# Removing data from the list

Removing an item to the list is assigned to the value 3, so if strChoice which is the input of the user equals 3 we will begin this task. I copied the line of code from “Show the current Items in the table” to display the items in the list for the end user to quickly see and decide what they would like to remove. I begin by asking the user to enter a task assigned to strRemove and check if the task is in the list. I add the .lower() function so that even if the end user enters in capital letters it will revert to one case and remove the task the end user entered in. I used a for loop to search through the rows to see if the input that the user entered matches something in the rows. If it does, that row will be removed from the lstTable and print that the task has been removed. However, if what the end user enters does not match “!=” something that is in the row, the code will print that the task is not in the list. Else, the code will continue and take the end user back to the menu of options. **Figure 4**  shows the for loop being used to go through the items in each row to see if it is something that matches with the input of the end user or not.

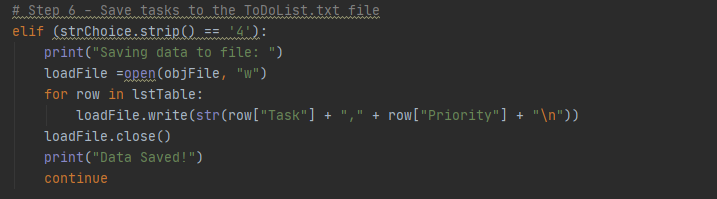
Figure 4



# Saving data to .txt file

Saving data to a .txt file is assigned to the value 4, so if strChoice which is the input of the user equals 4, we will begin this task. We begin by opening the file and using the letter “w” to write to it. To save the table to the “ToDoList.txt” only the values of each row are saved to the file as shown in **Figure 5**. Row[“Task”] will save the “Task” value and row[“Priority”] will save the “Priority” value. The data is written into the file with a comma to separate the two and a carriage return is added to the end of the line start the next data set on a new line.

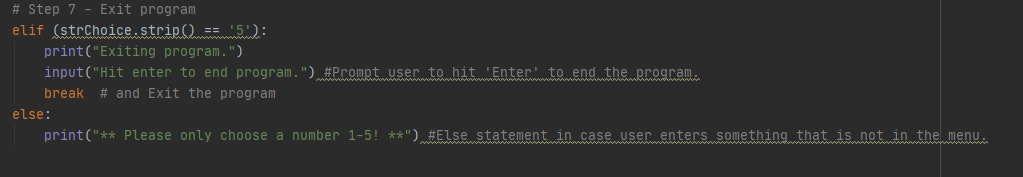
Figure 5

****

# Exiting the program

Exiting a program is assigned to the value 5, so if strChoice which is the input of the user equals 5, we will begin this task. Exiting a program is fairly simple, I add a break to the end of the program to exit. A break statement terminates the nearest enclosing loop.

Figure 6



# Running the script

Now that I have successfully created my script, I can hit the “run” tab located at the top of the file to execute it. Before running the script, the script will need to be saved. I saved my assignment as “Assignment05.py” to this location "C:\\_PythonClass\Assignment05\Assignment05.py" in the c drive in a folder called \_PythonClass and a subfolder called Assignment05. **Figure 7** shows the result of running the script from Pycharm Community edition 2020.1 version 201.6668.115 and adding a new item and displaying the data. **Figure 8** shows removing an existing item from the table. **Figure 9** shows running the save data to file and exiting program.

Figure 7

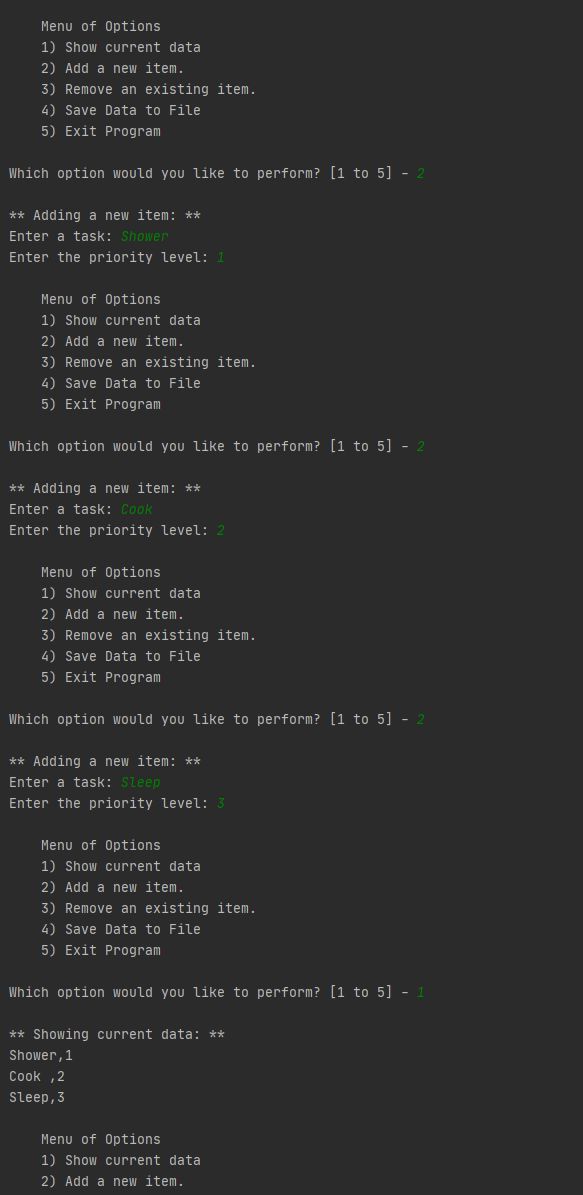


Figure 8

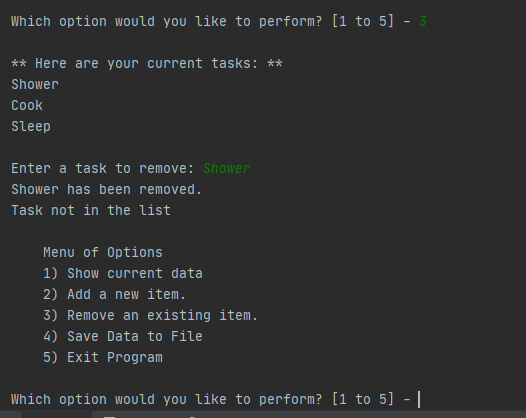
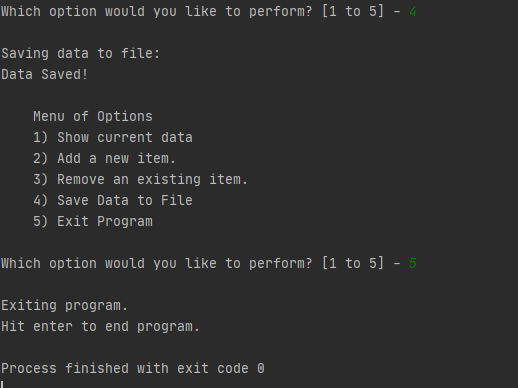


Figure 9

****

Meanwhile, **Figure 10** shows the saved text file in the path the file was opened at: "C:\\\_PythonClass\Assignment05\ToDoList.txt". **Figure 11** shows the data that was written into the text document. The data is formatted to have an Task name and Priority level on each line and start the next Task on a new line.

Figure 10

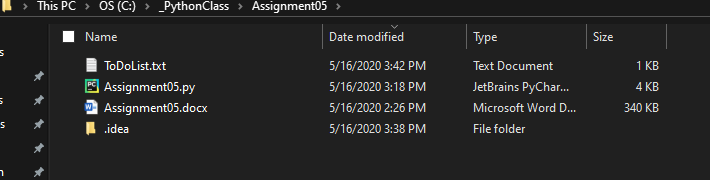
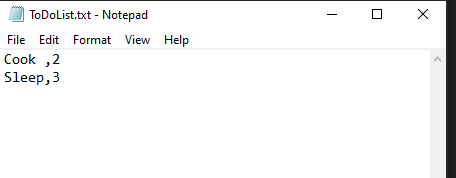
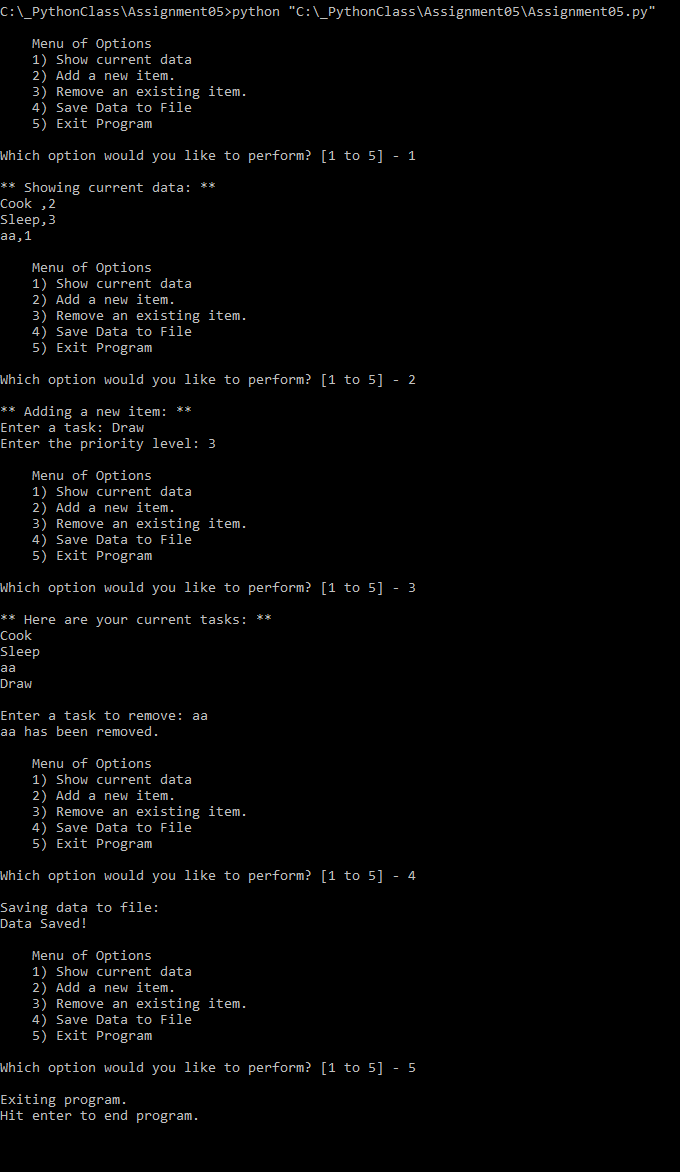


Figure 11



To run the script in an OS command, I opened a command prompt by typing “cmd” into the search bar of my desktop. I typed “Python "C:\\_PythonClass\Assignment05\Assignment05.py" into the command prompt providing the file location to open up the file. **Figure 12** shows the result of my program being run in the cmd line. I ran all 5 items in the menu of options.

Figure 12



# Summary

Students were asked to modify a Python script file that manages a “ToDo list”. This “ToDo” file contains two columns of data “Task” and “Priority” that are loaded into a Python Dictionary object. The rows of data are added to a Python list object to create a table of data. This program taught us how to add code to a Python script template and use dictionary objects to edit data in memory and in a file. This paper outlined the steps I took to create the code to add a new item to the list, remove an item from the list, save tasks to the “ToDoList.txt” file, show the current data, exit the program and test the results in a command shell from the Windows OS using Python version 3.