Dmitriy Alexandrov

November 12th, 2019

IT FDN 100 A

Assignment06

Coding a To Do Task List Script with Functions

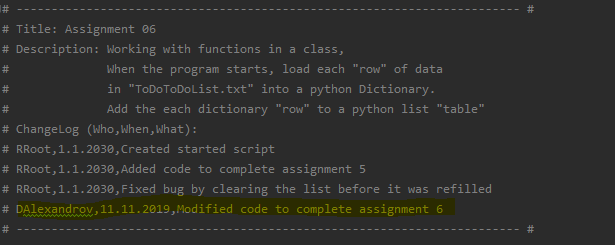
## Introduction

In this paper I am going to go through the steps that I took in adding code to a starter script to create a fully functional To Do Task List script, which opens up a ToDoList.txt file (or creates it, if the files doesn’t already exist) and is able to read, write and update tasks to that file using input that the user provides but this time using functions.

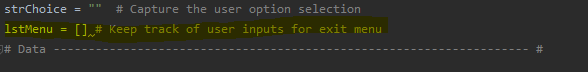
## Steps Taken in Adding Code to the To Do Task List Script

I started out the process by launching PyCharm and creating a new project called Assignment06 in my \_PythonClass folder on my C: drive using a virtual Python interpreter. After the project was created, I imported the starting code for the assignment called “Assignment06\_Starter.py”.

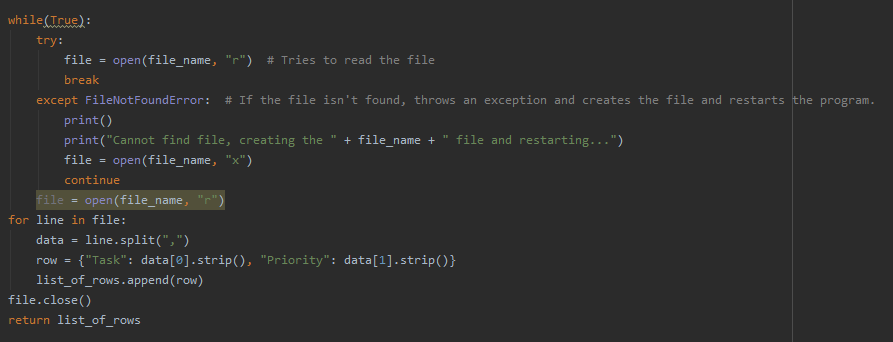
With the file in PyCharm, the first thing I added to the document was a change log entry in the header of the code to identify who was working on the code after the initial creation of it.



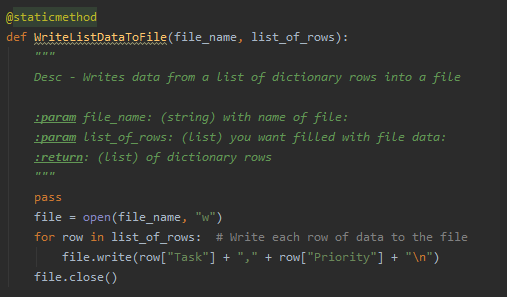
After taking a look at the declared variables that were already in the code, I identified what I already had to work with. The only change that I made to the already declared variables was adding a lstMenu variable to keep track of user inputs as an added feature that I also had in Assignment05.



The first change that I made was to the ReadFileDataToList function, because when I initially ran the script it gave me an error that the file that the file didn’t exist. So I added a try except statement to create the file if it doesn’t exist before trying to open it. This code tries to open the file, if an error is generated then it creates the file first then tries again.

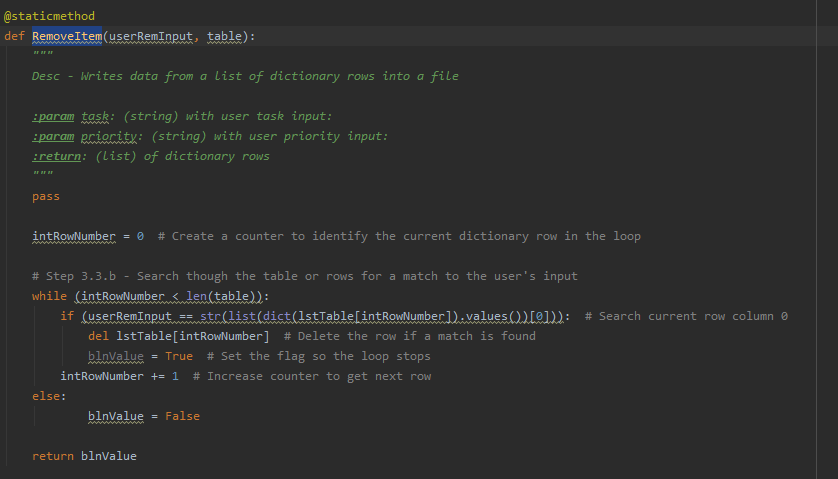


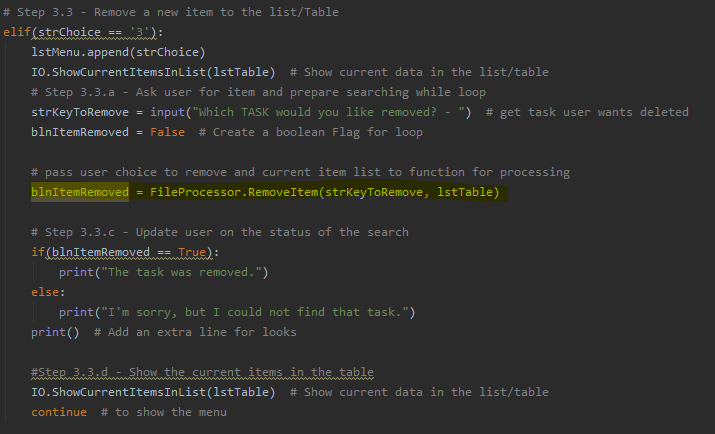
After fixing the ReadFileDataToList function, I moved to the WriteListDataToFile function and added the following code to it:



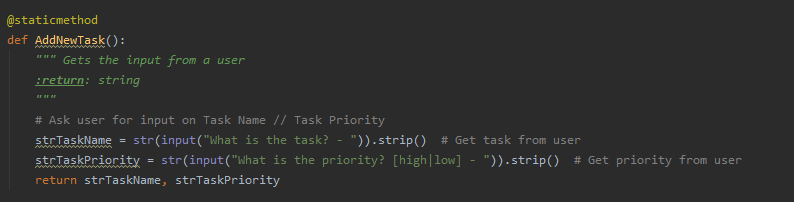
First, I added the Description and parameters to show the tooltip in PyCharm if someone wanted to see what this function does. After that I took the code from the Step #3.4 in the starter code and placed it into this function so that it can be reused in the future. This function took in two parameter (file\_name and list\_of\_rows) and writes the data to a file.

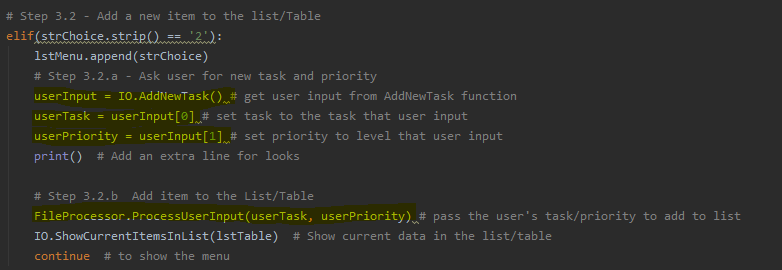
Continuing looking through the starter code, the remove task code needed to be a function so that it can be reused rather than being used once in the main code. For this I created a new function called RemoveItem which received two parameters userRemInput and table. userRemInput was what the user input for which task they wanted removed and “table” was the list of current tasks. When the function runs and receives these inputs it looks through the list of tasks and if it finds the task that the user wants to delete, it deletes it and returns a Boolean value to let the user know if the item was deleted or not.

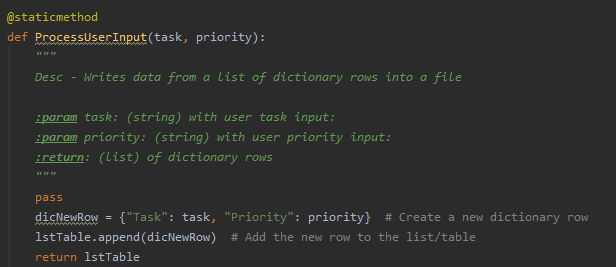




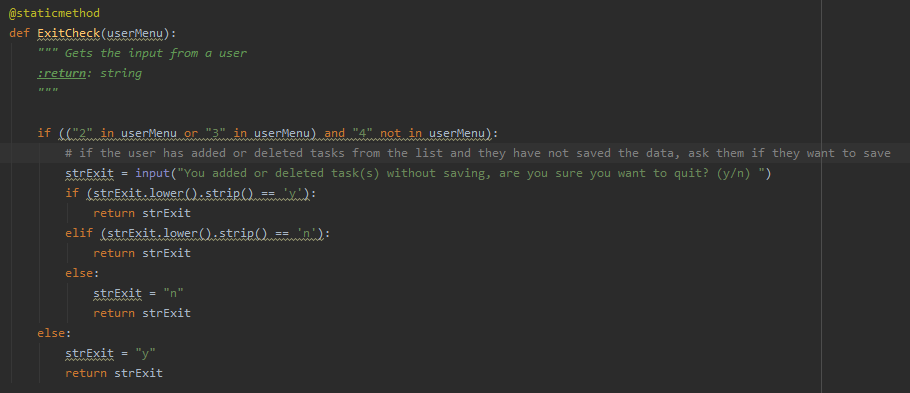
Jumping back above the delete task function, I added a function that would handle the adding of a new task by taking user input and after the function is executed return what the user put in, into a new local variable (strInput). After that I created two more local variables (strTask, strPriorityt) which took the 1st and 2nd values of strInput which were the task and priority that the user input. After the user input is received and stored, I created a ProcessUserInput function which added the user input to the task list.

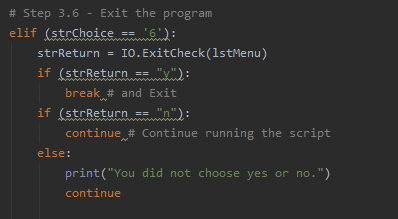






Lastly, I added the ExitCheck function that would run if the user chose to exit the script (option 6). This check would see what values were in the lstMenu variable that I added at the beginning and appended to whenever the user chose an option. With this list if the user ever added a task (option 2) or deleted a task from the list (option 3) and did not save their progress (option 4) then it would ask the user if they want to save their progress before exiting. The function then returned the user’s response and depending on if they chose to exit or not would exit the program or continue running until a save had been completed or the user just chose to exit.

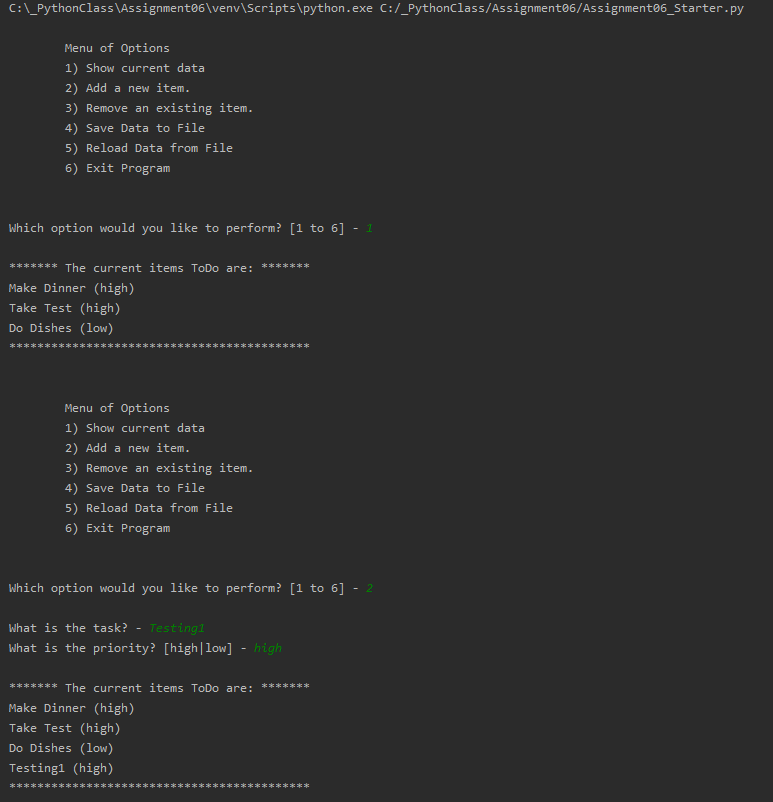


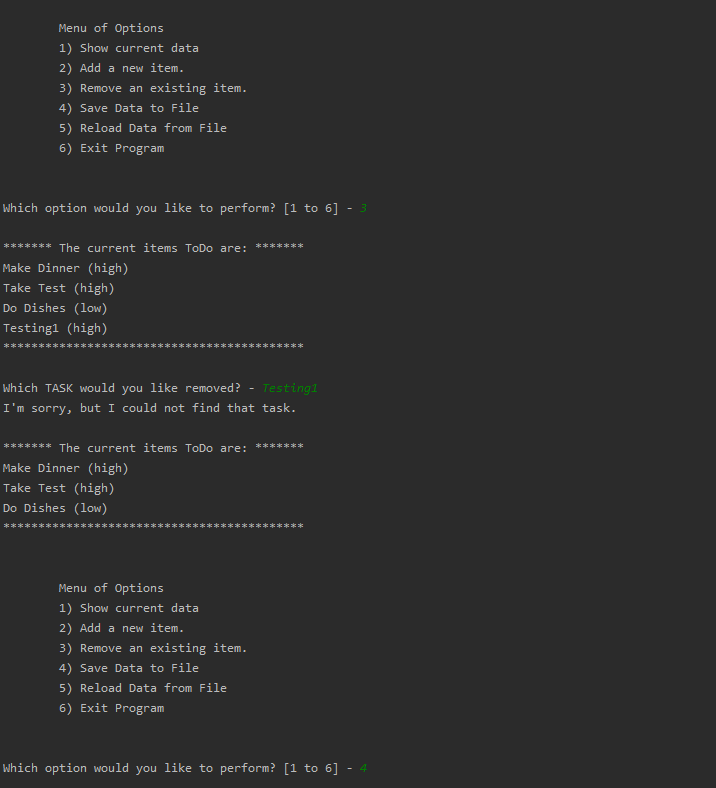


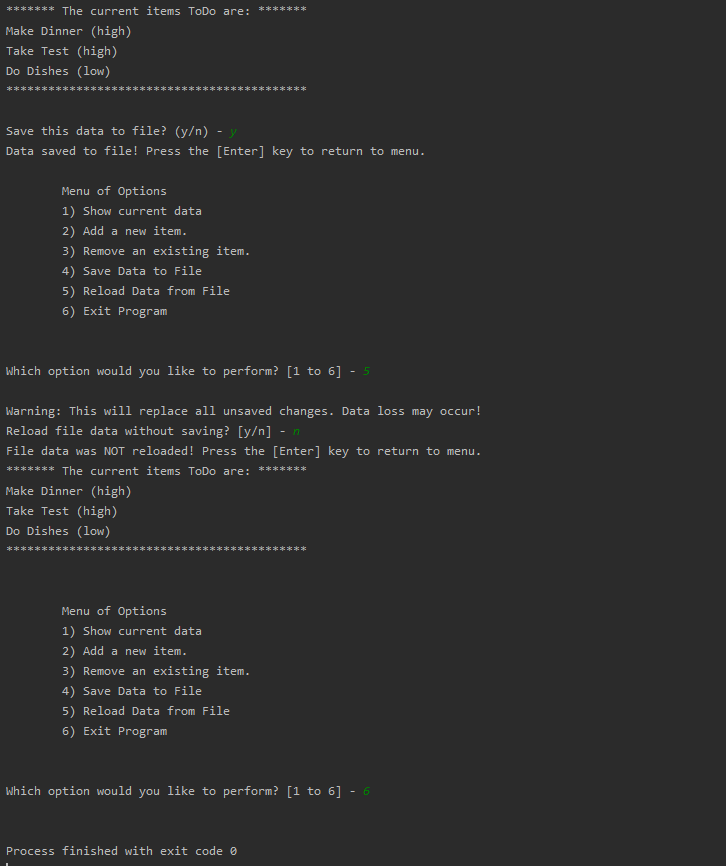
## Summary

Overall, the script worked as intended and reflected what was being asked for in the assignment. I added extra functionality to the script outside of the original assignment to help add some more functionality to the program as stated in the steps above.

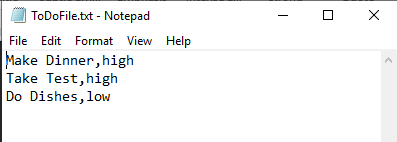
The results from running the script from PyCharm were:



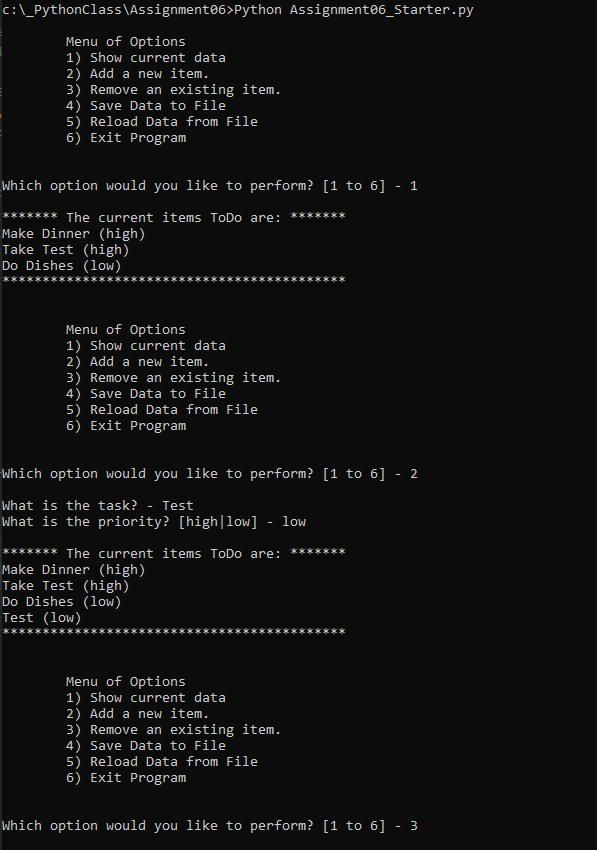


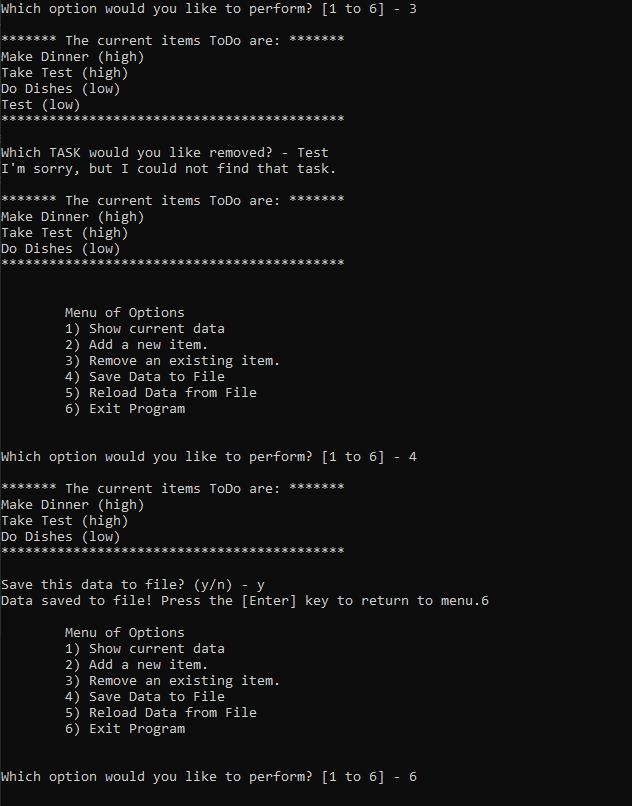


The output in the ToDoList.txt file was:



And running the script from the OS shell looked like this:





The output in the ToDoList.txt file was:

