Junle Li

Aug 10, 2020

Fundamentals of Programming: Python

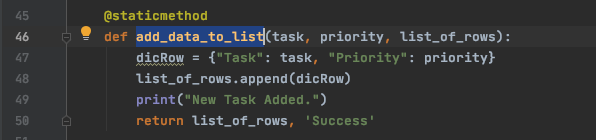
Assignment05

**Introduction**

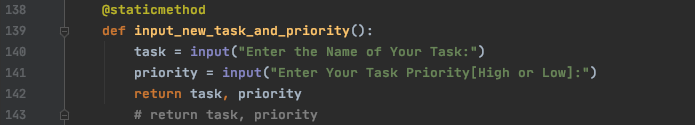
This week’s assignment is to add on codes by using Professor Root’s starter file then make the script work. This week’s script is similar to last week but now we use functions to perform tasks instead.

**Step 1: Finish add\_data\_to\_list function**

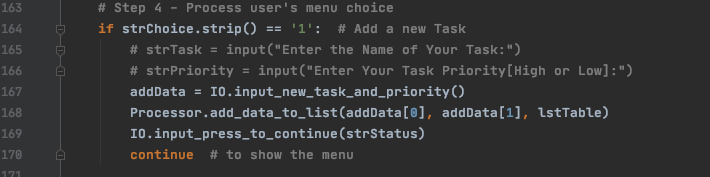
As most of the codes are filled by the starter, all we do is like converting last week’s code and apply them with the functions we learned this week. So the first step I took is to complete the add data function defined under the Processor class. So that we could call it later on.



Then under the input/output class, I see a function was defined to be used as input, so I added the code that would capture the user's input on the name of the task and its priority.

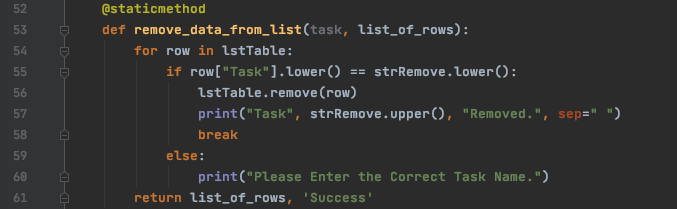


Now we can complete the remaining part of adding new tasks and priority by calling the two functions we defined previously. It is under the if statement when the user choose to perform add new task option. We first call the input\_new\_task\_and\_priority function, after we capture user input, then we call add\_data\_to\_list to have our data written into the list of rows. 0 for task, 1 for priority.

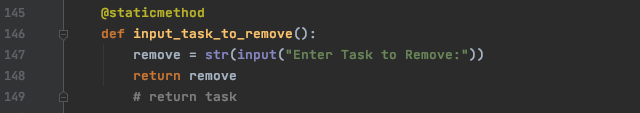


**Step 2: Remove task**

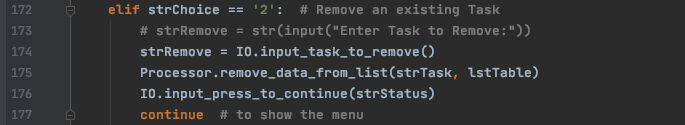
To finish the existing task, we will do similar things to above by completing the defin function first. Use an if statement to check if the user input matches our task data if row["Task"].lower() == strRemove.lower(): before we perform to the removing action, other wise it prints an error saying incorrect input.



Under the IO class, we simply add a code asking for the user’s input.

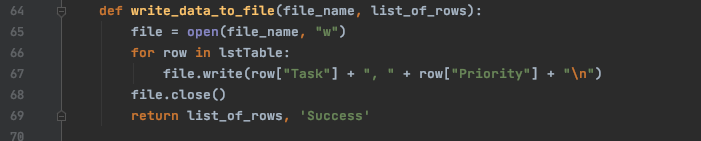


In the main body, we will be calling the input\_task\_to\_remove For user input, then have it process the removing action by calling remove\_data\_from\_list Function under the Processor class.

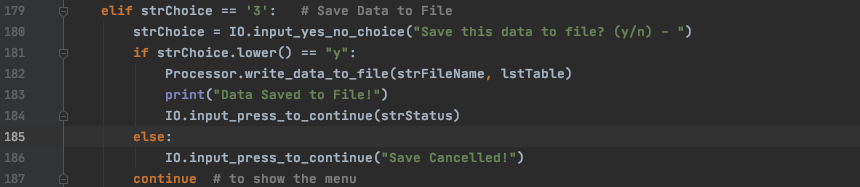


**Step 3: Save data to file**

Option 3 allow users to save data to file, as a result, the function needs to perform an action that could write the data into the file, it is the same as last week.

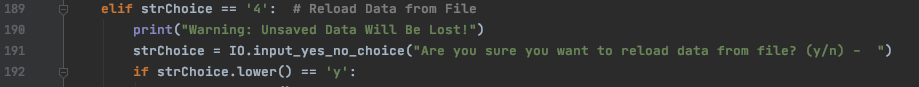


After we complete the function, we will need to call it in the while loop. It starts with a confirmation message that needs the user to choose yes or no. When the user chooses yes, we will call the function under the Processor class, and the data will be saved with a message. If the user chooses no, it will cancel the action and return to the menu.



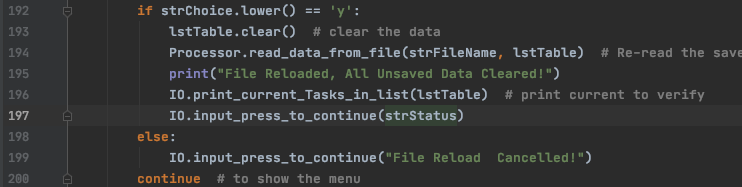
**Step 4: reload data**

No functions def for this option. Reload data means clear current unsaved data and re-read the file again, it is like a start over. It already starts with a print message and if statement.



So when the user chooses yes, I will first clear all the data by using .clear().

After everything is cleared, and since the read file function is already defined under the Processor class, I will just need to call it and have it read the file again. Then prints the tasks to users again to let the user know that the file is being reloaded.



**Conclusion**:

This is a similar code compared to last, but a bit confused on where to start, but it get a lot easier when I watch the video guide. During coding, I tried to use function to do the reload option, but it turned out that it took more steps and makes the code more complicated, so I changed it back as I saw the review video.