Katie London
11/17/2021
Foundations of Programming: Python
Assignment06
https://github.com/KatieLondon/IntroToProg-Python-Mod06

Creating a To-Do List

Introduction

In assignment 6, we modify code to create a program to manage a to-do list. The program itself is similar to the one we worked on for homework in assignment 5 but in this one, we group code into functions and classes to better achieve separation of parameters. The result is separate classes for data, processing and presentation. This assignment also provides practice with the practical skill of working with someone else's code.

To-Do List Script

Data

Since we are starting from existing code, the data section of the to-do list script was already provided. It contains several string variables to capture user input, a dictionary object to hold each row of data and a list object to act as a table and store the rows of dictionary elements.

Processing Class

The processing steps are broken down into functions inside the processor class which are indicated by the @staticmethod keyword. The class has four functions: read_data_from_file, add_data_to_list, remove_data_from_list and write_data_to_file. The function names are descriptive and use snake casing with an underscore between words to improve readability. Another element that we added this week, to improve the readability of the code, is to provide a comment immediately after the function definition called a docstring. This comment has a description of what the function does along with the parameters the function takes and what values are returned.

The read_data_from_file function was provided while the other three functions, listed above, required additional code to implement their functionality. The add_data_to_list function has three parameters: task, priority and list_of_rows. Therefore, all that was left to do was to put the task and priority into a dictionary element and append the dictionary onto the end of the list and return the list and a string indicating the function was successfully executed.

The next function implemented was the remove_data_from_list function which has two parameters: task and list_of_rows. Inside this function we use a for loop to loop through each row in the table to find a match to the argument task provided by the user. When there's a match, we use the remove function to delete the row with the specified task. Within the function, it should be noted that the variables are missing data type prefix which is common practice in Python within a function. Finally, this function returns the table data and the status to the main body of the script where the function was called.

The last function within the processing class is the write_data_to_file function that has two parameters: file_name and list_of_rows. Within this function, we open a text file and use a for loop to cycle through each row of data in the table and write it to the text file, before closing the file. The text file is saved within the same folder as the program itself since no explicit path is provided. This function returns the table of data and the function execution status.

Presentation Class

The presentation class contains seven functions each handling either input needed from the user or output to the user as indicated by the function name. The functions include print_menu_Tasks, input_menu_choice, print_current_Tasks_in_list, input_yes_no_choice, input_press_to_coninue, input_new_task_and_priority and input_task_to_remove. Of these seven functions, only two remain to be implemented. The input_new_task_and_priority function takes no parameters and returns two strings which store the task and priority provided by the user. The second function, input_task_to_remove, also has no parameters. This function returns the input from the user as to which task they'd like to remove.

Main Body

The main body is the section of the code where the functions are called. After the program starts and loads the data, we enter a while loop where we continue presenting menu choices to the user until they choose to exit. Within the while loop there is an if-elif construct to match each of the five menu choices. Inside each menu choice, there are calls to the processor class functions and input-output class functions. To use functions stored in a class we have to call the name of the class followed by a period and then followed by the function name. Each function also needs to include arguments to be mapped to the parameters in the function definition. Finally, to capture the return data from the function, each function is assigned to a variable within the main body of the script. For functions with more than one return value, such as the processor functions, we can unpack the values just as we would unpack a tuple. The second

return value for the processor class function is a string to indicate that the code was successfully executed.

Running Python Script

I successfully ran Assignment06 script, to implement a to-do list, in PyCharm and in the Terminal. In PyCharm, I tested each of the 5 menu options as shown in Figure 1 - Figure 4 below. On the Terminal I only tested one menu option since I had previously demonstrated the other options were working correctly in PyCharm.

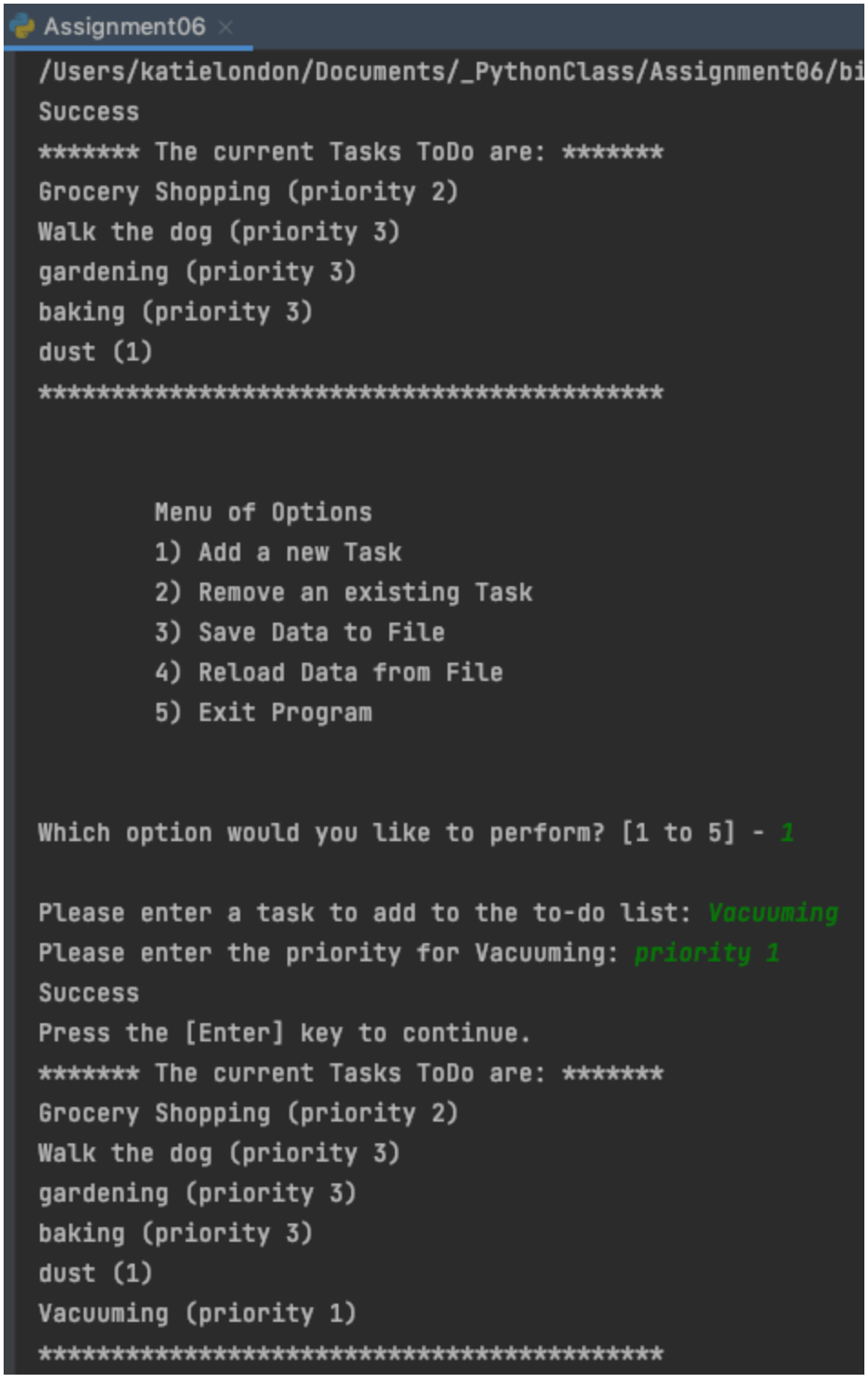

Figure 1: Screenshot of Menu option choice 1 in PyCharm

```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Reload Data from File
        5) Exit Program


Which option would you like to perform? [1 to 5] - 2

Please enter the name of the task you would like to remove: walk the dog
Success
Press the [Enter] key to continue.
******* The current Tasks ToDo are: *******
Grocery Shopping (priority 2)
gardening (priority 3)
baking (priority 3)
dust (1)
Vacuuming (priority 1)
*****************************************
```

Figure 2: Screenshot of Menu option choice 2 working in PyCharm

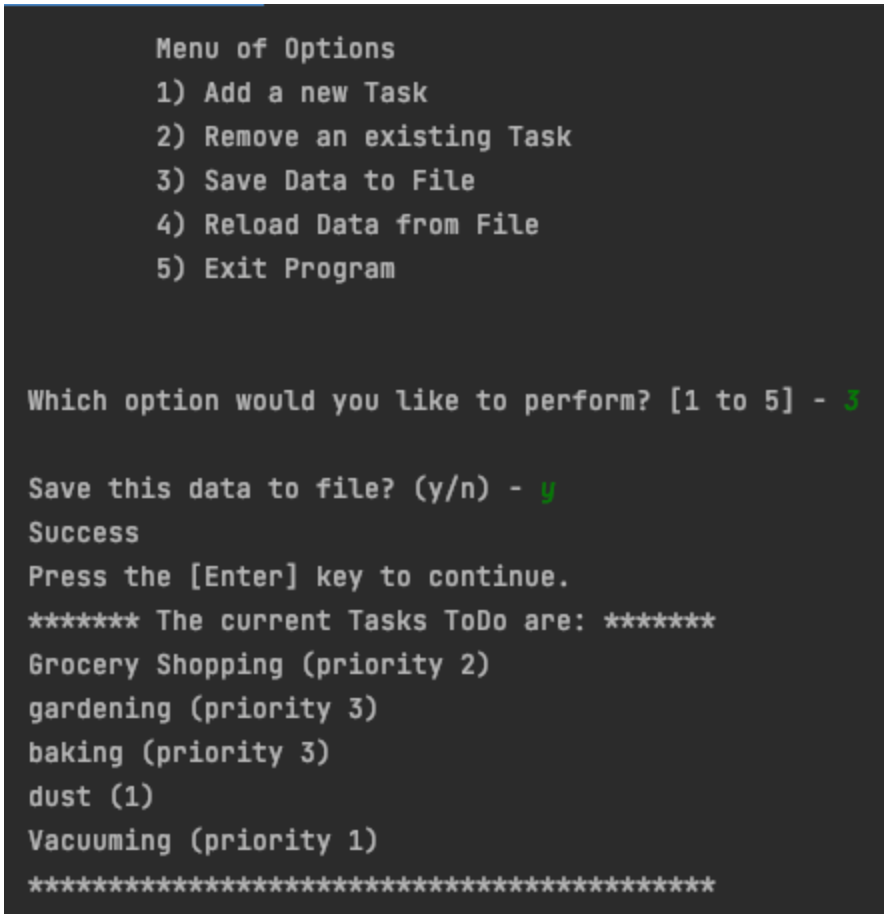

Figure 3: Screenshot of Menu option choice 3 working in PyCharm

```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Reload Data from File
        5) Exit Program

Which option would you like to perform? [1 to 5] - 4

Warning: Unsaved Data Will Be Lost!
Are you sure you want to reload data from file? (y/n) -  y
Success
Press the [Enter] key to continue.
******* The current Tasks ToDo are: *******
Grocery Shopping (priority 2)
gardening (priority 3)
baking (priority 3)
dust (1)
Vacuuming (priority 1)
*********************************************


        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Reload Data from File
        5) Exit Program


Which option would you like to perform? [1 to 5] - 5

Goodbye!
```

Figure 4: Screenshot of menu option choice 4 and 5 working in PyCharm

Figure 5: Screenshot of the Assignment06 script successfully running in Terminal

Summary

To complete assignment 6, we modified a script to manage a to-do list. The script was organized into three sections: data, processing and presentation. The processing steps and presentation steps were inside functions within their respective classes. The functions were called within the main body of the script and passed arguments to be used inside the function. The function output had to be captured and assigned to a variable to be used within the main body of the script.