Gao Yujia

18 May 2021

Foundations of Programming: Python

Assignment06

Github link: [yujiagao/IntroToProg-Python-Mod06 (github.com)](https://github.com/yujiagao/IntroToProg-Python-Mod06/tree/main)

**Working with Functions**

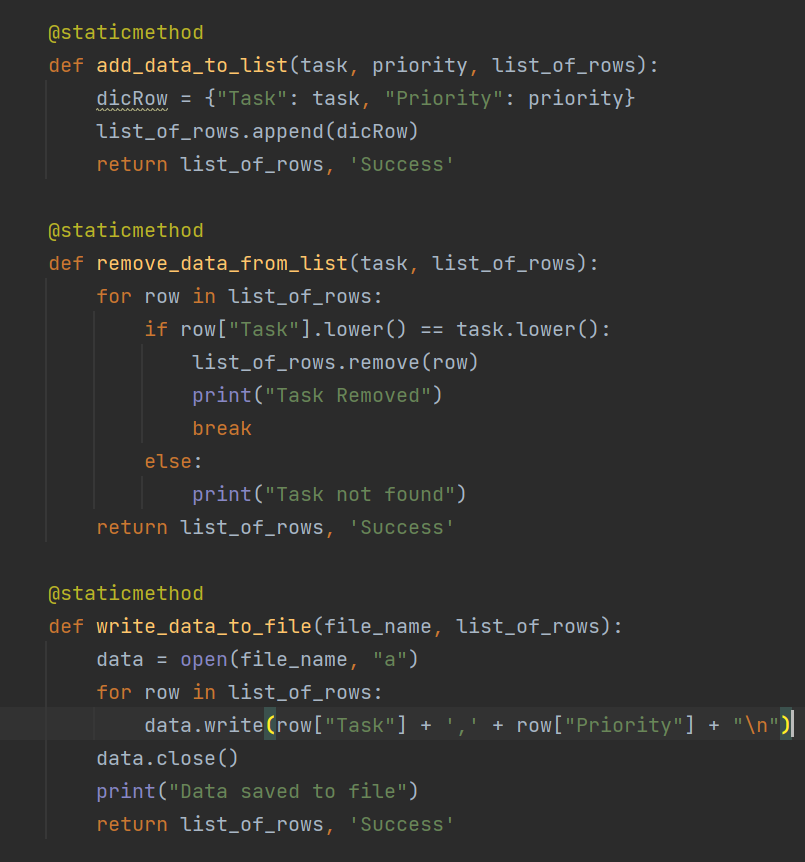
**Introduction:**

**Functions and classes are a way for us to organise our codes so that they do not appear too messy to ourselves, as well as others who use it. The difference between a function and a class is that a function defines a set of code, while a class defines a set of functions. To me, a class is like a folder, and a function is a subfolder within that folder, and the individual lines of codes that reside in a function are the files within that subfolder. It is not always easy creating functions and classes, and sometimes it might even be easier to just write the lines of code directly. But once we are familiar with the use of functions and classes, it makes things a lot easier and faster as we can keep the main presentation part of the scrip neat and short.**

**Creating the Python File**

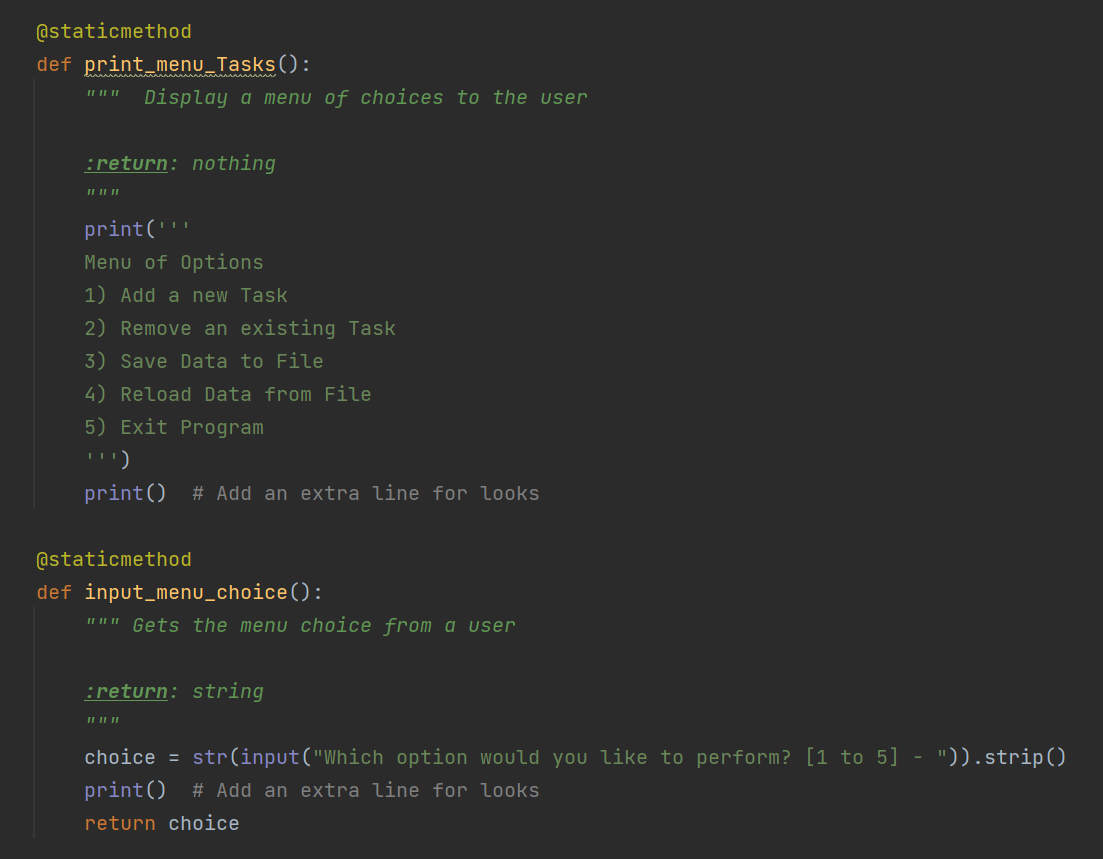
**There are multiple steps required in the creation of this ‘ToDoList’ file.**

**The first step is opening the Assignment06\_Starter.py file. Using this started file, we are to create the functions that will enable the script to work. The first things to do will be the ‘add\_data\_to\_list’, ‘remove\_data\_from\_list’ and ‘write\_data\_to\_file’ functions. These are done are shown in Fig. 1.**



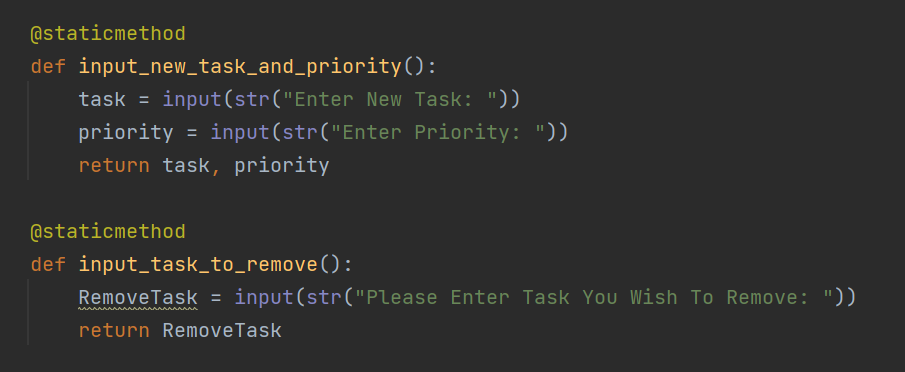
**Fig. 1**

**Once this is completed, we move on to the Input/Output portion of the script. The functions for this portion are placed in the ‘IO’ class to group them together. The first 2 steps of printing the menu of tasks and asking for an input choice are given in Fig. 2**



**Fig. 2**

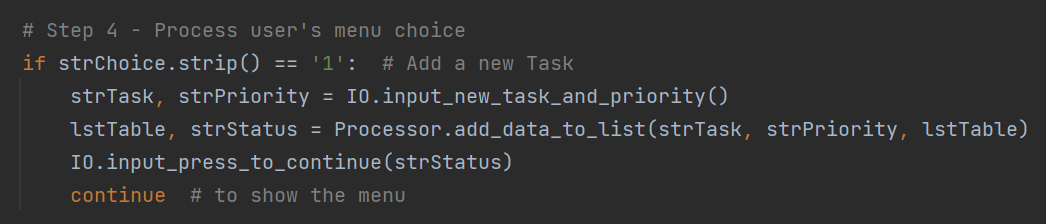
**What we are required to do instead is to write the codes for the functions ‘input\_new\_task\_and\_priority), and ‘input\_task\_to\_remove’. These codes will form the basis of interpreting the user’s input in the main body of the script, and are as follows in Fig. 3:**



**Fig. 3**

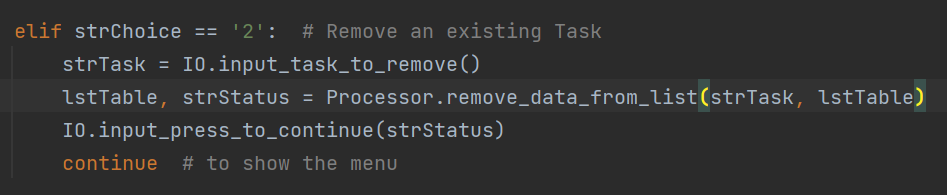
**Once this is completed, we move on to the main body of the script, where we organise these functions so that they perform the tasks that they are meant to.**

**For the first function, when the user enters ‘1’, this will signal the input of new data to the table. The first line of the subsequent code will rely on the IO class and the ‘input\_new\_task\_and\_priority’ function in that class. After that, the function ‘add\_data\_to\_list’ from the Processor class will save the user’s input into the table Fig. 4.**



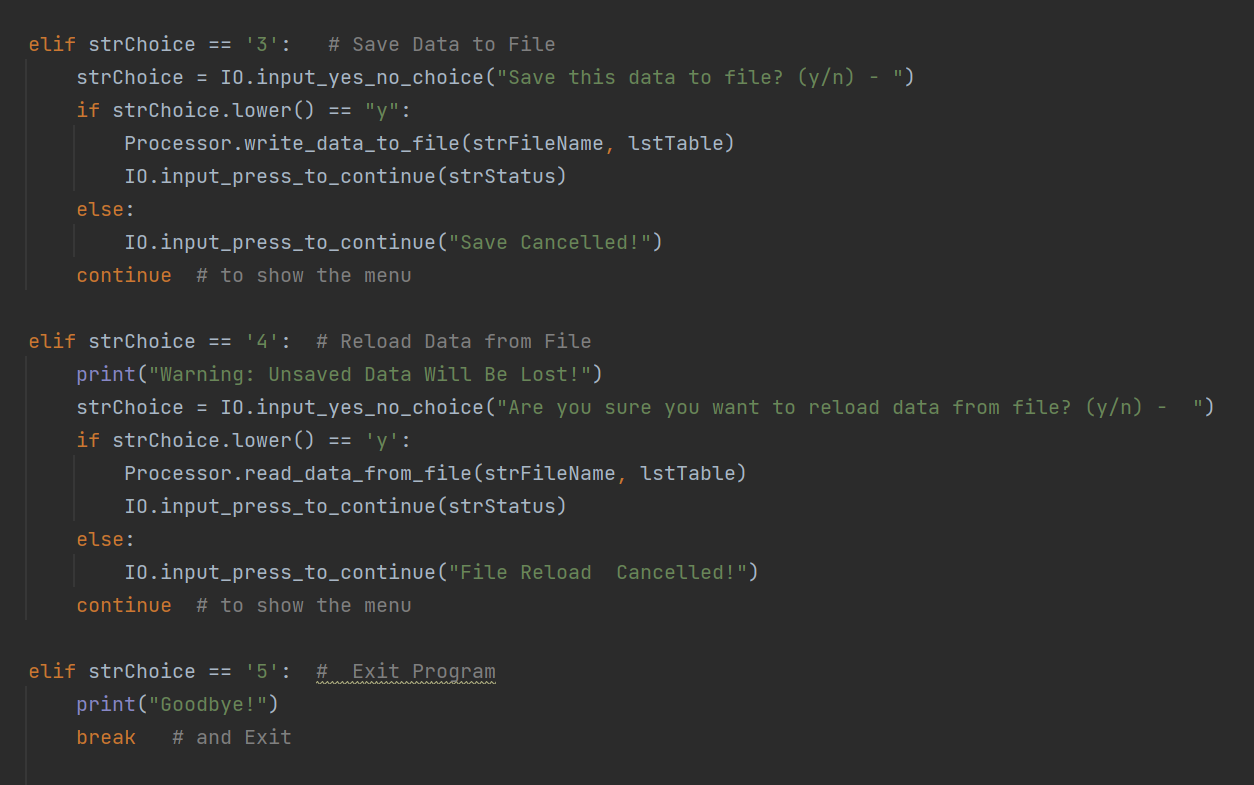
**Fig. 4**

**Option ‘2’ will remove the selected task within the table. Again, it will leverage on the respective functions within the IO and Processor class to perform this task. And as the codes for the functions were already pre-defined, it is simply calling up the necessary function to perform the task Fig. 5.**



**Fig. 5**

**The principle for options 3,4,5 are the same, except that we are not writing any new data into the table, but either saving the data into the .txt file or recalling the data from the .txt file Fig.6.**



**Fig. 6**

**Conclusion:**

**It is not easy working with functions and classes. It requires us to remember the functions that we created, and the functions also need to be absolutely correct, else we will have problems when running those functions in the main body of the script. However I believe that once I am more familiar and comfortable with functions and classes, it will make coding much faster and easier, compared to having to enter individual codes every time even for repeated tasks.**