Craig Muth

11/19/2022

IT FDN 110 A Au 22: Foundations Of Programming: Python

Assignment 06

<https://github.com/cjmuth/IntroToProg-Python-Mod06>

TODO List - Read/Write using Functions, and the GitHub Repository

# Introduction

The goal of this project is to create a TODO list program, that will use functions to read data from an existing text file, allow the user to view the existing data, make changes, and save the data back to the text file.

A partial program has been provided to start from. All the necessary structure exists, but operational code has been omitted in several places so it will not run as it currently exists. So we will need to map the logic for the existing code, identify where it is lacking, and develop the logic and code to make it work.

# Designing the program

Examining the starter file, the logic flow looks like this - with areas where code is missing is indicated.

* Data
  + declare variables and constants
* Processing
  + class Processor
    - def read\_data\_from\_file
      * open the file in read mode
      * loop through rows in data file
        + read data from file, splitting into discrete values
        + assign the discrete values to a dictionary
        + add dictionary to list\_of\_rows
      * when end of rows, close file
      * return list\_of\_rows to main program
    - def add\_data\_to\_list
      * assign passed values to a dictionary
      * *MISSING CODE*
      * return list\_of\_rows to main program
    - def remove\_data\_from\_list
      * *MISSING CODE*
      * return list\_of\_rows to main program
    - def write\_data\_to\_file
      * *MISSING CODE*
      * return list\_of\_rows to main program
* Input/Output
  + class IO
    - def output\_menu\_tasks
      * print menu of options to screen
    - def input\_menu\_choice
      * get user option
      * return choice to main program
    - def output\_current\_tasks\_in\_list
      * loop through dictionaries in list\_of\_rows
        + print values to screen
    - def input\_new\_task\_and\_priority
      * *MISSING CODE*
    - def input\_task\_to\_remove
      * *MISSING CODE*
  + call read\_data\_from\_file
  + while true
    - call output\_current\_tasks
    - call output\_menu\_tasks
    - call input\_menu\_choice
    - if "1"
      * call input\_new\_task\_and\_priority
      * call add\_data\_to\_list
        + pass values task, priority, table\_lst
    - if "2"
      * call input\_task\_to\_remove
      * call remove\_data\_from\_list
        + pass values task, table\_lst
    - if "3"
      * call write\_data\_to\_file
        + pass values, file\_name\_str, table\_lst
    - if "4"
      * exit program

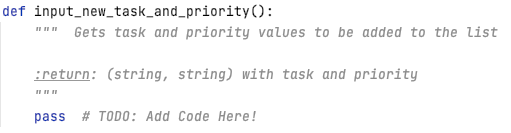
Since both the processing and input/output sections include functions exchanging arguments with main program, instead of addressing the code section by section - we’ll pull the related pieces from each section. This will make it easier to understand how the values are passed around for each operation.

### Add a new task

When the user selects option 1, the program runs the following code

***Figure 1: Menu option 1. Add a new task***

This instructs the computer to first run the function input\_new\_task\_and\_priority and assign the output to the variables task and priority. Looking at the pseudocode above, we can see this function is one of the areas that is missing code.

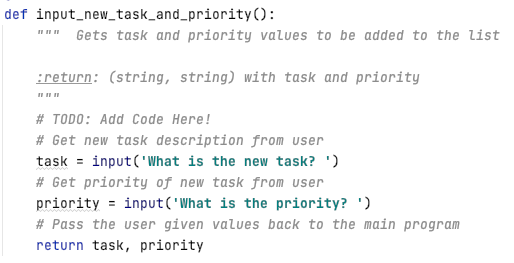


***Figure 2: Input new task***

Neither the call to the function, nor the function itself, show that values need to be passed to the function - but the call in the main program is assigning values to two variables. So the logic for this function should be:

* + Get new task description from user
  + Get priority of new task from user
  + Pass the user given values back to the main program

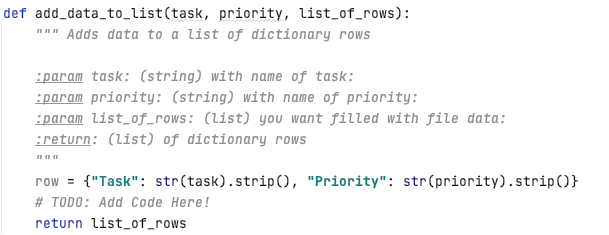
We use input statements to get values from the user, then the return statement is used to pass the values back to the main program (Ramos, Leodanis Pozo (n.d.). *The Python return Statement: Usage and Best Practices,* <https://realpython.com/python-return-statement/>) (External site).



***Figure 3: Updated Input new task***

Now that we have values for the new task, the code calls the function add\_data\_to\_list, this time

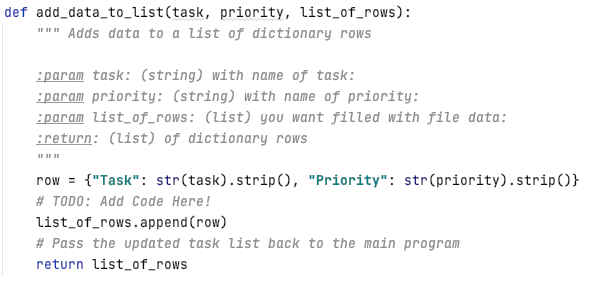
we are passing the values from the previous function, along with the existing value of variable table\_list, and assigns the results back into the table\_list variable.



***Figure 4: Add data to list***

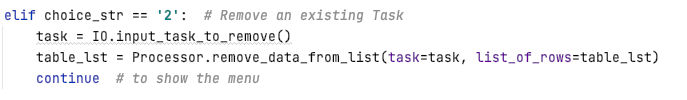
We see the function is designed to receive three parameters, and

* + Receive values from calling code
  + Assign task and priority to a dictionary
  + Add the dictionary to list\_of\_rows
  + Return list\_of\_rows to main program

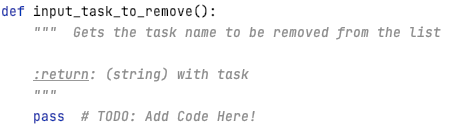


***Figure 5: Updated Add data to list***

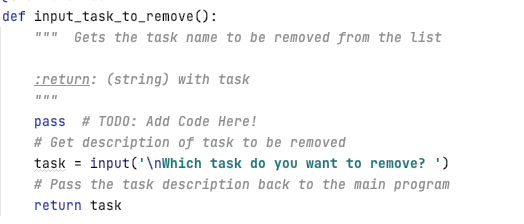
### Remove an existing task



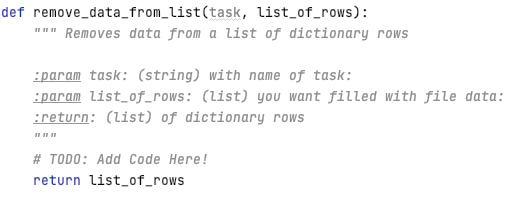
***Figure 6:***



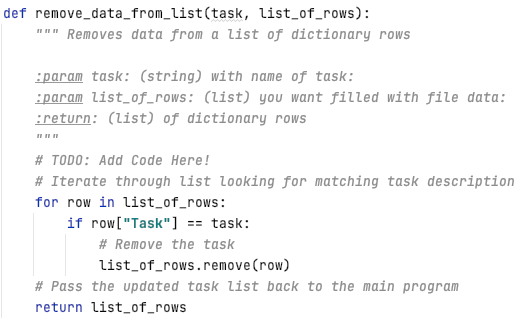
***Figure 7:***



***Figure 8:***

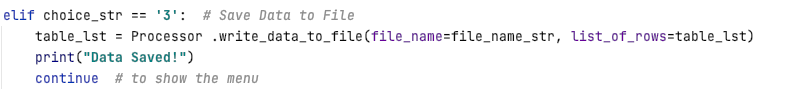


***Figure 9:***

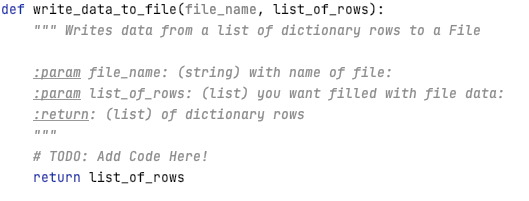


***Figure 10:***

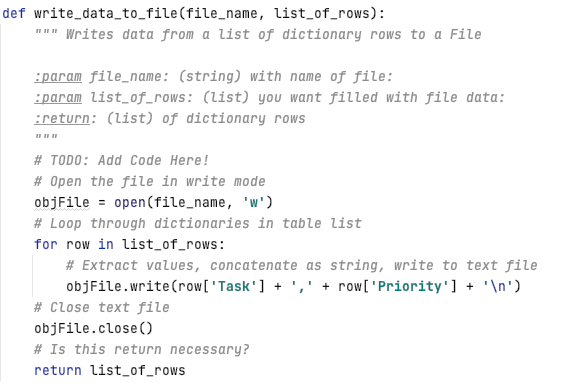
### Save data to file

******

***Figure 11:***

******

***Figure 12:***

******

***Figure 13:***

# Running the program

### Executing the program in Pycharm:

### Executing in a Terminal window

# GitHub page

GitHub is a hosting site using version control software original developed by Linus Torvalds for his Linux

# Summary

In this project we created a program to interact with text file - reading data into memory, modifying it,