Jason Verberne

November 23, 2022

Foundations of Programming: Python

Assignment06

https://github.com/jasonverberne/IntroToProg-Python-Mod06 (external link)

https://jasonverberne.github.io/IntroToProg-Python-Mod06/ (external link)

TO DO LIST – TASK AND PRIORITY – EDITING EXISTING SCRIPTS

INTRODUCTION

This paper will discuss the steps taken in completing Assignment06 of the Foundations of Programming: Python Course. This assignment required the modification of / addition to a Python script that would allow the user to select from a menu to show task and priority data, add a new task and priority to the list, remove an existing dictionary from the list, save the information to a text file, and exit the program.

BACKGROUND

Prior projects and knowledge that contributed to the successful completion of this assignment include:

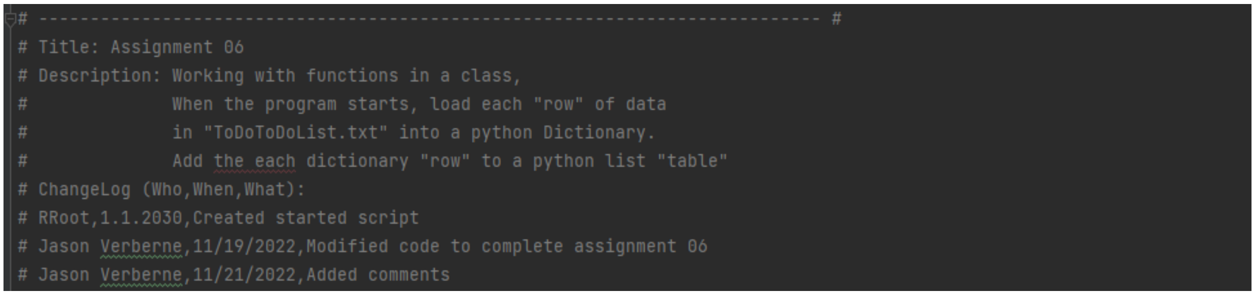
1. Assignment01: This assignment requested the creation of a Python script that would accept the first and last name of a user and, in return, would display the concatenated information provided as the full name.
2. Assignment02: This assignment requested the creation of a Python script that would independently request two numbers from the user and, in return, would display the sum, difference, product, and quotient of those two numbers.
3. Assignment03: This assignment requested the creation of a Python script that would gather a household item from the user, gather an estimated value of the item, and save that information in text file.
4. Assignment04: This assignment requested the creation of a Python script that would allow the user to select from a menu to 1) gather multiple household items and estimated costs, 2) display the information gathered from the user back to the user, and save that information in text file and exit the program.
5. Assignment05: This assignment required the modification of a Python script that would allow the user to select from a menu to show task and priority data, add a new task and priority to the list, remove an existing dictionary from the list, save the information to a text file, and exit the program.
6. Independent study prior to the course, including:
    a. YouTube: https://www.youtube.com/watch?v=rfscVS0vtbw, freeCodeCamp.org, Mike Dane (presenting), uploaded July 11, 2018 (External Site)

b. Pierce County Library Book: Begin to Code with Python, Miles, Robert S., Microsoft/Pearson Education, Inc., 2018, ISBN 9781509304523

I chose to perform my coding in PyCharm instead of using IDLE.

Header

Before writing code that will actually perform the task, it is important to provide supporting information to any programmer, including myself for later use, about the purpose and historical activities of the code. This information does not have to be lengthy, although it could be if needed. In this case, I provided a description of what the code would do, then provided the Who, What, and When of the changes.



```
# ---------------------------------------------------------------------------- #
# Title: Assignment 06
# Description: Working with functions in a class,
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# Jason Verberne,11/19/2022,Modified code to complete assignment 06
# Jason Verberne,11/21/2022,Added comments
```

Figure 1: Header of Python code

Because this information is not intended to be processed by Python, I started each line with a “#” symbol. In Python, any line that begins with a number sign (aka the pound sign) will not be processed by Python. Alternatively, I could have used two sets of three single quotes (‘’’) with the header information between the sets, which would have provided the same result.

Document Focus

Assignment 06 had several elements, including:

1. This assignment required me to review existing code and write missing code so the program would perform the intended task. In this document, I will focus on the code that I wrote as opposed to code that was provided.
2. Update github.com
3. Publish a website on github.com

Code

The basic elements of this code include:

1. variables - the variables I used to store the data are:
    a. count – Used to store integer 0 that would be increased as it went through a for loop

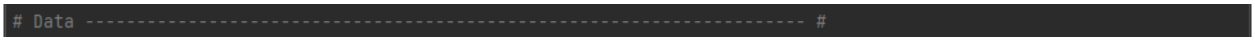
    b. file – used to represent the opened text file
    c. task – used to represent the user provided task. This variable is used for both adding a new task and removing a task from the list, depending on the user selection from the menu.
    d. priority – used to represent the priority of the newly provided task

2. .append – this method appends the identified object to the list
3. for/in – This loop iterates through an object, in this case a list, and performs the programmer defined action on each object in the list.
4. open/close file – this opens and closes files allowing potential updates to occur or the program to read from the opened file. In this case, it was a text file.
5. input() - this function allows the user to provide information to be stored in Python
6. return – used as the end of a function, this statement defines what is returned after the function is complete

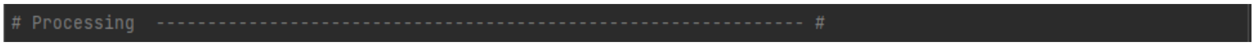
----

Separation of Concern

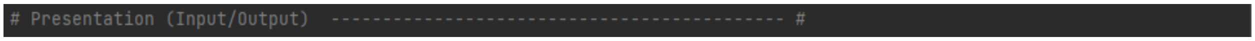
"Separation of Concern," is defined as, *"In computer science, separation of concerns (SoC) is a design principle for separating a computer program into distinct sections, so that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program."* https://en.wikipedia.org/wiki/Separation_of_concerns (external link), 2019. In our Assignment06 template, we had several notes identifying the separate areas.



```
# Data -------------------------------------------------------------------------- #
```

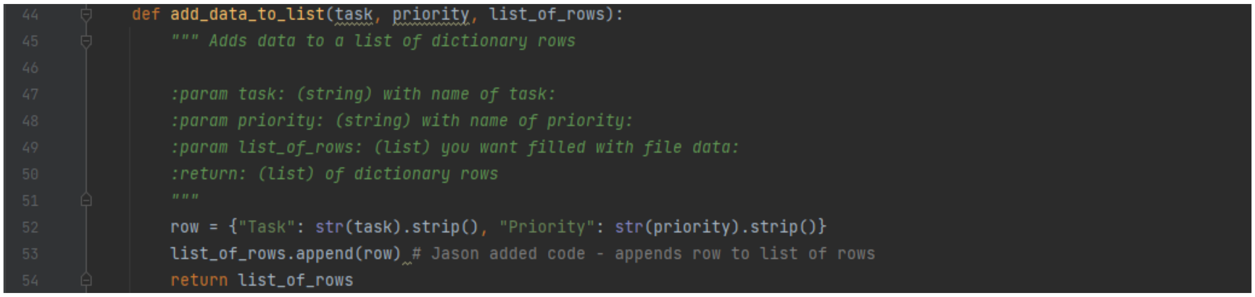


```
# Processing  ------------------------------------------------------------------- #
```

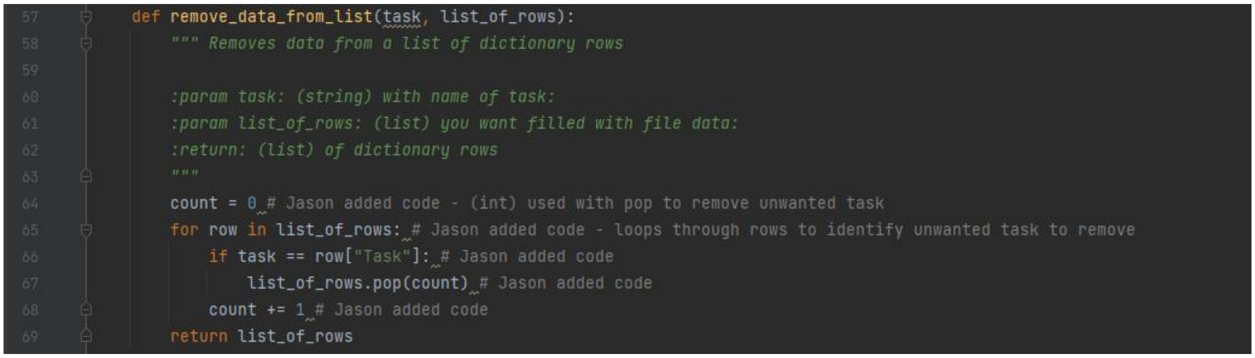


```
# Presentation (Input/Output)  ------------------------------------------------- #
```

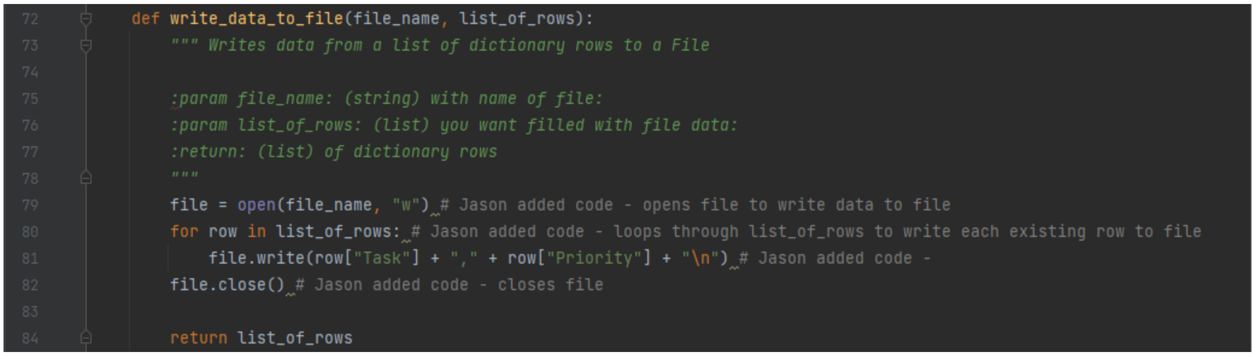
Figure 2, Separations of Concern from the provided script



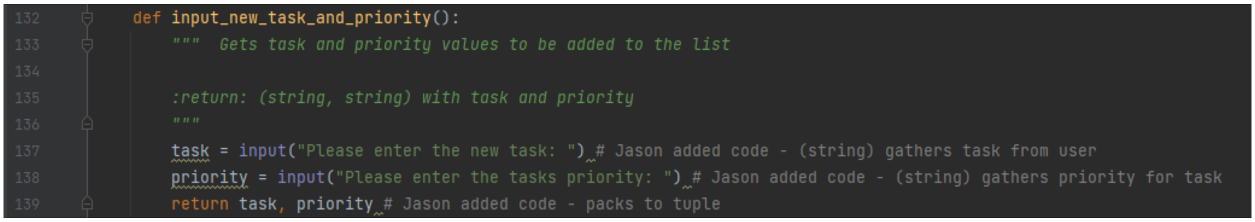
```
def add_data_to_list(task, priority, list_of_rows):
    """ Adds data to a list of dictionary rows

    :param task: (string) with name of task:
    :param priority: (string) with name of priority:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    row = {"Task": str(task).strip(), "Priority": str(priority).strip()}
    list_of_rows.append(row) # Jason added code - appends row to list of rows
    return list_of_rows
```



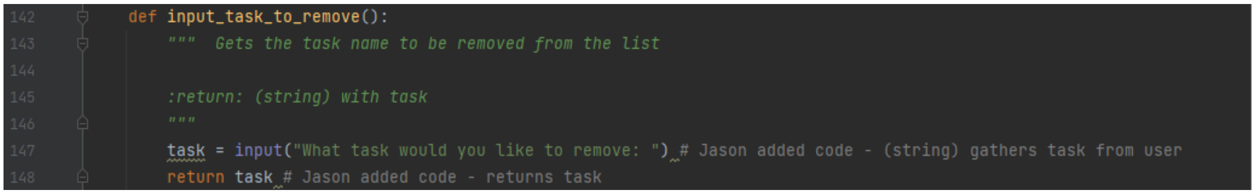
```
def remove_data_from_list(task, list_of_rows):
    """ Removes data from a list of dictionary rows

    :param task: (string) with name of task:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    count = 0 # Jason added code - (int) used with pop to remove unwanted task
    for row in list_of_rows: # Jason added code - loops through rows to identify unwanted task to remove
        if task == row["Task"]: # Jason added code
            list_of_rows.pop(count) # Jason added code
        count += 1 # Jason added code
    return list_of_rows
```



```
def write_data_to_file(file_name, list_of_rows):
    """ Writes data from a list of dictionary rows to a File

    :param file_name: (string) with name of file:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    file = open(file_name, "w") # Jason added code - opens file to write data to file
    for row in list_of_rows: # Jason added code - loops through list_of_rows to write each existing row to file
        file.write(row["Task"] + "," + row["Priority"] + "\n") # Jason added code -
    file.close() # Jason added code - closes file

    return list_of_rows
```



```
def input_new_task_and_priority():
    """  Gets task and priority values to be added to the list

    :return: (string, string) with task and priority
    """
    task = input("Please enter the new task: ") # Jason added code - (string) gathers task from user
    priority = input("Please enter the tasks priority: ") # Jason added code - (string) gathers priority for task
    return task, priority # Jason added code - packs to tuple
```

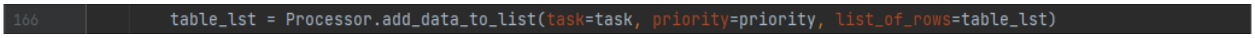


```
def input_task_to_remove():
    """  Gets the task name to be removed from the list

    :return: (string) with task
    """
    task = input("What task would you like to remove: ") # Jason added code - (string) gathers task from user
    return task # Jason added code - returns task
```

Figure3: Python functions that contain added code. Added code denoted by "*# Jason added code*"

Lines 23 - 94

1. Addition of Line 53: The add_data_to_list function is called when the user chooses option 1 from the menu list to add a 'task' and 'priority.' This function defined the row dictionary in line 52, but failed to append the new dictionary as a list item to the table_lst variable. Line 53 appends the additional object to the list.



```
        table_lst = Processor.add_data_to_list(task=task, priority=priority, list_of_rows=table_lst)
```

Figure4: Python code that calls the add_data_to_list function.

2. Addition of Lines 64-68: The function remove_data_from_list function is called when the user chooses option 2 from the menu list to remove data from the list. The function was missing the necessary code to review each dictionary/object in the list to search for the user provided task to remove. The for/in loop and the if statement were written to accomplish this review, but once the task in the dictionary is equal to the provided task, the pop method is used to remove the list item. Because the pop method requires input on the list position item to remove, the count += 1 iteration is used to provide the needed pop data.
3. Addition of Lines 79-82: The write_data_to_file function is called when the user chooses option 3 to save data to the file. The original file failed to open, write to, and close the file. The for loop iterates through each list item, which is a dictionary. Line 81 is used to pull the task and priority from the dictionary and write to the text file with a comma separator.
4. Addition of Lines 137-139: The input_new_task_and_priority function is called when the user chooses option 1 from the menu list to add a 'task' and 'priority.' In line 165, I could see that the script wanted to unpack a tuple with the 'task' and 'priority' which would come from the return statement. The function, therefore, needed a task and priority variable, each with an input() function to gather the data from the user to be processed into the return statement.
5. Addition of Line 147: The input_task_to_remove function is called when the user chooses option 2 from the menu list. Line 148's return statement needed a task to return. Line 147 uses the input function to gather the data from the user to pass to the return statement.

Results

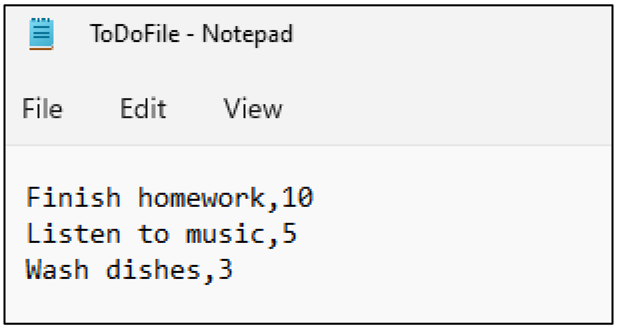
The results of the program were as expected, with the programming loading a list of tasks and priorities from an existing text file and the user selecting from a menu to 1) add new tasks to a list, 2) remove items from an existing list, 3) saving the updated information to a text file, and/or 4) Exiting the program.

```
--------------------------------------------------------------------------------

C:\_PythonClass\Assignment06>python Assigment06_Starter.py
******* The current tasks ToDo are: *******
Finish homework (10)
Listen to music (5)
Make dinner (8)
******************************************


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


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

```
Please enter the new task: Wash dishes
Please enter the tasks priority: 3
******* The current tasks ToDo are: *******
Finish homework (10)
Listen to music (5)
Make dinner (8)
Wash dishes (3)
********************************************


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


Which option would you like to perform? [1 to 4] - 2

What task would you like to remove: Make dinner
******* The current tasks ToDo are: *******
Finish homework (10)
Listen to music (5)
Wash dishes (3)
********************************************


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


Which option would you like to perform? [1 to 4] - 3

Data Saved!
******* The current tasks ToDo are: *******
Finish homework (10)
Listen to music (5)
Wash dishes (3)
********************************************
```

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


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

Goodbye!

C:\_PythonClass\Assignment06>
--------------------------------------------------------------------------------
```

Figure 5: Results after running the Python code and user providing input.



```
Finish homework,10
Listen to music,5
Wash dishes,3
```

Figure 6: Saved Text File of the User Input

Posting files to GitHub and creating a Webpage

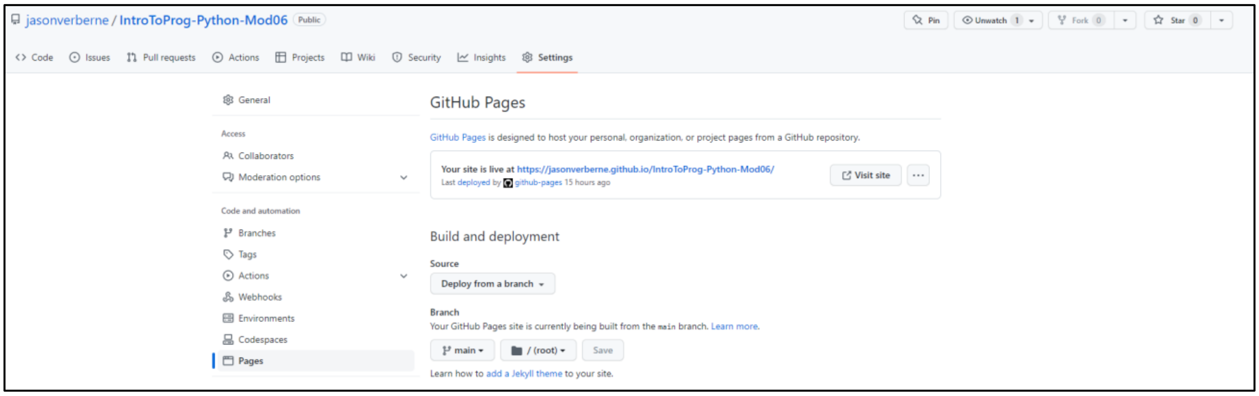
Assignment 06 required the addition of files to my personal GitHub page, linked above. Creating a new repository and posting the files online did not seem to be a difficult task since it was similar to what was performed for Assignment 05. We were also asked to create a webpage, link above, utilizing the GitHub website.

Challenges and learning

There were a few challenges and points of learning from this week's assignment.

- Coding – When reviewing other's code, for me, it was best to walk through the code and process it in my own mind, similar to how the debugging feature would operate. As I walked through each option the user could/would make, I found this caused me to 'bounce around' the coding, but the separation of concern's organization did make this easier. I also found the search feature in PyCharm to be useful when reviewing functions.

- GitHub – The only difficulty I faced when creating the website from GitHub was that it wasn't clear to me where I would find the link to the website. There was no "your website link will be xyz when created." Instead, it just appeared after it was processed. The other issue was that I wasn't clear on the Branch to choose. At first, I chose "none" from the dropdown menu, but correcting that to "main" seems to have resolved the issue.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at https://jasonverberne.github.io/IntroToProg-Python-Mod06/
Last deployed by github-pages 15 hours ago

Build and deployment

Source: Deploy from a branch

Branch
Your GitHub Pages site is currently being built from the main branch. Learn more.

main / (root)

Learn how to add a Jekyll theme to your site.

Figure 7: Image from the GitHub-Settings-Pages page

SUMMARY

This document described the steps taken and approach to completing Assignment06 of the Foundations of Programming: Python Course. It discussed the prior knowledge and background information considered when writing this script, the major sections of the script, and the returned results the user would expect. Finally, it discusses the experiences when completing the GitHub activities