

Name: Natalie Pritchett

Date: 8/29/2023

Course: Foundations of Programming: Python

Assignment06 – Working with Functions

<https://github.com/natpritch/IntroToProg-Python/tree/Assignment06>

## Introduction and TODO #1

To get started I opened the Assignment06\_Starter.py file using the Visual Studio Code editor. Then, I edited the script header to update the change log. For the first TODO I appended the new row items that had been created in the code provided for line 52 to the list of rows. I also added a return so that the user could view the updated list of To Do items.

```
42     @staticmethod
43     def add_data_to_list(task, priority, list_of_rows):
44         """ Adds data to a list of dictionary rows
45
46         :param task: (string) with name of task:
47         :param priority: (string) with name of priority:
48         :param list_of_rows: (list) you want to add more data to:
49         :return: (list) of dictionary rows
50         """
51         #list_of_rows = {"Task": str(task).strip(), "Priority": str(priority).strip()}
52         row = {"Task": str(task).strip(), "Priority": str(priority).strip()}
53         # TODO: Add Code Here!
54         list_of_rows.append(row) # Add the new row to the list/table
55
56         return list_of_rows
57
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + -

```
Which option would you like to perform? [1 to 4] - 1
What is the task? dust
What is the priority? (high/med/low) med
***** The current tasks ToDo are: *****
mop (low)
sweep (med)
mow (high)
dust (med)
*****
```

## TODO #2

The next to do item was to remove the selected item from the list. I created a for loop that would check to see if the task entered by the user matched a task in the list of tasks. If there is a match, then the task will be removed. If there is no match for the user provided task input then nothing happens. Either way,

the current list of tasks is returned for the user to review to make sure that task removal was performed correctly. If it was not done correctly, they can review the task list and try again.

```
58     @staticmethod
59     def remove_data_from_list(task, list_of_rows):
60         """ Removes data from a list of dictionary rows
61
62         :param task: (string) with name of task:
63         :param list_of_rows: (list) you want filled with file data:
64         :return: (list) of dictionary rows
65         """
66         # TODO: Add Code Here!
67         for row in list_of_rows:
68             if row["Task"] == task:
69                 list_of_rows.remove(row)
70                 break
71         return list_of_rows
72
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

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

What is the name of task you wish to remove? : dust

\*\*\*\*\* The current tasks ToDo are: \*\*\*\*\*

mop (low)

sweep (med)

mow (high)

\*\*\*\*\*

### TODO #3

The third To Do writes the user entered information to the text file 'ToDoFile.txt'. The return statement lets the user review what information in the To Do list was saved. The data entered can also be verified by reviewing the saved text file.

```
73     @staticmethod
74     def write_data_to_file(file_name, list_of_rows):
75         """ Writes data from a list of dictionary rows to a File for option 3
76
77         :param file_name: (string) with name of file:
78         :param list_of_rows: (list) you want filled with file data:
79         :return: (list) of dictionary rows
80         """
81         write_file = open("ToDoFile.txt", "w")
82         for item in list_of_rows:
83             write_file.write(item['Task'] + ',' + item['Priority'] + '\n')
84         write_file.close()
85         return list_of_rows
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python

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

Data Saved!  
\*\*\*\*\* The current tasks ToDo are: \*\*\*\*\*  
mop (low)  
sweep (med)  
mow (high)  
\*\*\*\*\*

ToDoFile.txt - Notepad  
File Edit Format View Help  
mop,low  
sweep,med  
mow,high

#### TODO #4

The fourth To Do provides the presentation code so that the user knows what to do. First, the user is prompted to enter a task. Then, they are asked to select a priority level of low, medium, or high. I also added a return statement so that the user could view the updated list of To Do items.

```

132     @staticmethod
133     def input_new_task_and_priority():
134         """ Gets task and priority values to be added to the list
135
136         :return: (string, string) with task and priority
137         """
138         task = str(input("What is the task? ")) #ask user to enter task
139         priority = str(input("What is the priority? (high/med/low) ")) #
140         return task, priority
141

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

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

What is the task? dust

What is the priority? (high/med/low) low

\*\*\*\*\* The current tasks ToDo are: \*\*\*\*\*

mop (low)

sweep (med)

mow (high)

dust (low)

\*\*\*\*\*

## TODO #5

The final To Do item provides the information that will be presented if the option to remove an item is selected. The user is asked what is the name of the task that they would like to remove. This prompt helps the user know what they should enter after making this selection.

```

142     @staticmethod
143     def input_task_to_remove():
144         """ Gets the task name to be removed from the list
145
146         :return: (string) with task
147         """
148         removeTask = str(input("What is the name of task you wish to remove? : ")).strip()
149         return removeTask
150

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Python + -

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

What is the name of task you wish to remove? : dust

\*\*\*\*\* The current tasks ToDo are: \*\*\*\*\*

mop (low)

sweep (med)

mow (high)

\*\*\*\*\*