

Working with Functions

Introduction

This week we learned about functions and will use them to conduct operations we have learned in the past weeks. Functions are used to organize and conduct operations when called upon by the program/user. We'll be writing data to a file, saving data in memory, removing data and ending the program all with functions built into classes. The functions have been prewritten and we will fill in the function tasks.

Functions/Classes

This week we needed to fill in some functions within a mostly written program. We found these functions needing work defined with the #TODO note. There were two classes within the program, one for processing functions and the other for IO operations. These were setup to go hand in hand as we will get input from the user within the IO functions and then take that input to the processor function to then proceed with storing, removing, adding data.

Filling in the IO Functions

With the above knowledge that we'll take the input and then begin processing it, I decided to begin with the IO functions. Scrolling through the code our first task is to get a task with a priority from the user. This is all information from previous weeks.

```
def input_new_task_and_priority():  
    """ Gets task and priority values to be added to the list  
    :return: (string, string) with task and priority  
    """  
  
    # TODO: Add Code Here!  
    task = input("What task would you like to add?\n").strip()  
    priority = input("What is the priority of this task? (Low | High)\n").strip()  
    return task, priority
```

Figure 1.1: Function to input new task with priority.

As seen in Figure 1.1 we just assign prompted user input to a variable and use the strip function to remove any blank space the user may accidentally add. The new thing we will add from this week is the return command. This is used to return the values executed within the function to the caller. This is shown within the prewritten code below:

```
if choice_str.strip() == '1': # Add a new Task  
    task, priority = IO.input_new_task_and_priority()  
    table_lst = Processor.add_data_to_list(task=task, priority=priority, list_of_rows=table_lst)  
    continue # to show the menu
```

Figure 1.2: If statement executing functions based off of user menu choice.

Gregory Booth

02-22-2023

Assignment06

Foundations of Programming (Python)

<https://boothga.github.io/ITFnd100---Mod06/>

Figure 1.2 shows our returned values set as task and priority. These values are then set to equal task and priority within the next processing function. This allows them to then be executed within that processing function.

```
def input_task_to_remove():  
    """ Gets the task name to be removed from the list  
    :return: (string) with task  
    """  
  
    # TODO: Add Code Here!  
    task = str(input("What task would you like to remove?\n")).strip()  
    return task
```

Figure 2.1: User input deciding which task to remove.

Our next function to finish is the input for which task we want to remove. This is shown in Figure 2.1. To set it up we simply set user input to the task variable and then return this value. Again, this code will return the value, that value will be set to equal “task”. Then we can add this task variable to the next processing function by setting “task” within the function call to equal that variable. That is the last IO function we had to finish and then we move to the processing functions.

Filling in the Processing Functions

Our first function to complete is adding data to the list. All we do here is add to the table with the append function.

```
row = {"Task": str(task).strip(), "Priority": str(priority).strip()}  
# TODO: Add Code Here!  
list_of_rows.append(row)  
return list_of_rows
```

Figure 2.2: Adding input data to the table.

We have gathered the user input from the previous IO function which we use that data and then add it to the dictionary and we add it to the list. This is shown in Figure 2.2.

```
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  
    """  
  
    # TODO: Add Code Here!  
    for row in list_of_rows:  
        if row["Task"].lower() == task.lower():  
            list_of_rows.remove(row)  
    return list_of_rows
```

Figure 2.3: Removing data from our list.

Gregory Booth

02-22-2023

Assignment06

Foundations of Programming (Python)

<https://boothga.github.io/ITFnd100---Mod06/>

For the removal of the task we use the same setup as last week again. We build a for loop using a row for reference within the `list_of_rows` we have stored in memory. If any of those rows match the task the user input from the IO function that task will be taken from the list. This is done by using the `.remove` function on the table and within the parameters adding the row found.

```
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  
    """  
    # TODO: Add Code Here!  
    objFile = open("ToDoFile.txt", "w")  
    for row in list_of_rows:  
        objFile.write(str(row["Task"]) + "," + str(row["Priority"]) + "\n")  
    objFile.close()  
    return list_of_rows
```

Figure 3.1: Writing data to the `ToDoFile.txt`.

Last function we had to address was the writing the data to a file. Again we have done this process in previous weeks. As shown in Figure 3.1 we open the file using the file name "w" for write. Then we'll begin a for loop to gather all rows within our table and write them to the file. We set it up with the `objFile.write` function and within its parameters we place the rows "Task" and "Priority". Added in is a comma to make it look nicer along with a newline to place the next row below again for looks. Lastly we'll close the file.

Summary

We got introduced to functions and classes and how they can organize a code and provide a lot more flexibility with how we can write our code. The assignment took us through steps to complete already called functions by using information we should have been able to recall from previous weeks. After writing my code I ran it with PyCharm and Terminal and below will be screenshots of the code running in both.

Gregory Booth
02-22-2023
Assignment06
Foundations of Programming (Python)
<https://boothga.github.io/ITFnd100---Mod06/>

```
Project ▾ | main.py | ToDoFile.txt
Assianment06 ~\PvcharmProjects\ITFnd100\Mod06\1 | chores,low
2

Run: main
***** The Current Tasks To Do Are: *****
*****

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

What task would you like to add?
chores
What is the priority of this task? (Low | High)
low

***** The Current Tasks To Do Are: *****
chores (low)
*****

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 To Do Are: *****
chores (low)
*****

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?
chores

***** The Current Tasks To Do Are: *****
*****

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

Are you sure you want to end program? (Y or N)y
```

```
***** The Current Tasks To Do Are: *****
*****

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

What task would you like to add?
homework
What is the priority of this task? (Low | High)
high

***** The Current Tasks To Do Are: *****
homework (high)
*****

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

What task would you like to add?
chores
What is the priority of this task? (Low | High)
low

***** The Current Tasks To Do Are: *****
homework (high)
chores (low)
*****

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 To Do Are: *****
homework (high)
chores (low)
*****

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?
homework

***** The Current Tasks To Do Are: *****
chores (low)
*****

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

Are you sure you want to end program? (Y or N)y

Goodbye!
```