Hayden Gaylord
IT FDN 110B
https://github.com/enjoyyourorange/IntroToProg-Python-Mod06
8/10/22

Assignment #6

Introduction

This portion of the class is focusing on creating our own functions to recall later in our code.

Creating our To-Do List Menu

Since Randal had given us a starter file, we first begin on creating a function that will populate our file, to do this, we need to define our rows with our tasks and priorities. Once populated, we have our program append the new rows to any existing table:

```
@staticmethod
def add_data_to_list(add_task, add_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(add_task).strip(), "Priority": str(add_priority).strip()}
    list_of_rows.append(row)
    return list_of_rows
```

fig1

Then, we need to create a function that lets the user remove certain tasks from their list. We do this by simply telling the program if the entry is equal to an existing task, we use the remove function to delete that entire row and then return what is existing:

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

    :param del_task: (string) with name of task:
    :param list_of_rows: (list) you want filled with file data:
    :return: (list) of dictionary rows
    """
    for row in list_of_rows:
        if row["Task"].lower()==del_task.lower():
            list_of_rows.remove(row)
    return list_of_rows
```

fig 2

Next, we are going to allow the user to save their entered tasks and priorities to a text file. We tell the program to open our file in write mode, and loop through and add to it any entered tasks and priorities, close the file, and let the user know the file has been saved:

```
    @staticmethod
    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
        """
        myfile=open(file_name,"w+")
        for row in list_of_rows:
            myfile.write(row["Task"] + "," + row["Priority"] + "\n")
        myfile.close()
        print("Thanks, your file has been saved.")
        return list_of_rows
"""
```

fig 3

Then, we move to our section of code that interacts with the user. The first IO function we create is to add new tasks to our to-do list. We do this by having our tasks and priorities decided from our user's input. Then we return the entered tasks and priorities to our caller code:

```
    @staticmethod
    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(str(("Please add a task" "\n")))
        priority=input(str(("Please give this task a priority" "\n")))
        return task, priority
```

:

Fig4

Next, we will create the function that allows a user to specify a task they'd like to remove. The user will enter a string and it will feed into our caller code and if there is a match, that line will be removed:

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

    :return: (string) with task
    """
    remove=str(input("Please enter the task you'd like to remove:")).strip()
    return remove
```

:

Fig5

Hayden Gaylord
IT FDN 110B
https://github.com/enjoyyourorange/IntroToProg-Python-Mod06
8/10/22

Summary

When I first opened the starter file, I got pretty anxious and unsure if I was up to speed. But after taking time and reviewing material and coming back, it made a lot more sense. Functions are a really nice tool in Python, it helps me separate my train of thoughts and put the process into a much more logical flow. I'm still a bit confused on how our functions created for processing interact with our IO functions, though.