Tiya Farah
5/24/2022
Foundations of Programming: Python
https://github.com/Tfarah22/IntroToProg-Python-Mod06

Assignment 6:

Introduction

For this assignment, I will demonstrate how to work with add/write data, save and display data as well as how to remove data from the text file.

Drafting the Code

I started by using the code provided by the instructor in module 06 files. I began by copying the data provided. My task consisted of opening and writing data into the file and filling in the required sections.

Step by Step Guide

Step #1: Copy the assignment06 starter code in PyCharm to begin. In the template provided there were notes that told you to #TODO: ADD Code Here. The first code that needed to be added were ones that added the data to the list of dictionaries. (see Fig. 1 and Fig.2)

Fig. 1

```
    @staticmethod
    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()}
        # TODO: Add Code Here!
        return list_of_rows

    @staticmethod
```

Fig.2

```python
@staticmethod
def add_data_to_list_of_dictionaries(list_of_dictionary_rows, task, priority):
    """ Adds data to a list of dictionary rows

    :param list_of_dictionary_rows: (string) with name of list your adding data to
    :param task: (string) with name of task
    :param priority: (string) with name of priority
    """
    row = {"Task": str(task).strip(), "Priority": str(priority).strip()}
    list_of_dictionary_rows.append(row)
```

Step #2: I then added the following code to remove data from the list of dictionary rows. (See Fig. 3 & Fig. 4)

Fig.3

```python
@staticmethod
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!
    return list_of_rows
```

Fig. 4

```python
def remove_data_from_list_of_dictionaries(list_of_dictionary_rows, task_to_remove):
    """ Removes a row of data from a list of dictionary rows

    :param list_of_dictionary_rows: (list) of dictionary data to remove a row from
    :param task_to_remove: (string) with name of the task in the dictionary's 'Task' key
    :return: (list) list_of_dictionary_rows, (bool) with status of success status
    """
    success_status = False  # Create a boolean Flag for loop
    row_number = 0  # Create a counter to identify the current dictionary row in the loop
    # Search though the table or rows for a 'Task' key match
    for row in list_of_dictionary_rows:
        if row["Task"].lower() == task_to_remove.lower():
            list_of_dictionary_rows.remove(row)
            # print("row removed")
            success_status = True
        row_number += 1
    return list_of_dictionary_rows, success_status
```

Step #3 For this step I added the script that would write data from a list to a file. The file was created in PyCharm titled ToDoList.txt. (Fig. 5 & Fig.6)

Fig.5

```python
@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
    """
    # TODO: Add Code Here!
    return list_of_rows
```

Fig.6

```python
@staticmethod
def write_file_from_list_of_dictionaries(file_name, list_of_dictionary_rows):
    """ Write data to a file from a list of dictionary rows

    :param file_name: (string) with name of file
    :param list_of_dictionary_rows: (list) of dictionary data saved to file
    :return: (bool) with status of success status
    """
    success_status = False
    file = open(file_name, "w")
    for row in lstTable:
        file.write(row["Task"] + "," + row["Priority"] + "\n")
    file.close()
    success_status = True
    return success_status
```

Step #4 was to create an input to add the task and priority of the task (See Fig. 7 & Fig.8)

Fig. 7

```
@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
    """
    pass  # TODO: Add Code Here!
```

Fig. 8

```
def input_task_and_priority():
    """ Gets data for a dictionary row

    :return: (tuple) of strings with task and priority
    """
    task = str(input("What is the task? - ")).strip()
    priority = str(input("What is the priority? [high|low] - ")).strip()
    print()  # Add an extra line for looks
    return task, priority
```

Step #5 For this step I added code that will get a task name removed from the list. (See Fig.9 & Fig. 10)

Fig. 9

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

    :return: (string) with task
    """
    pass  # TODO: Add Code Here!
```

Fig.10

```
@staticmethod
def print_data_removed_status(success_status):
    """ Print the status of the task removal process

    :param success_status: (bool) status you want to display
    """
    if success_status:
        print("The task was removed.")
    else:
        print("I'm sorry, but I could not find that task.")
    print()  # Add an extra line for looks
```

Step #6 I tested the script by running it and it returned the following result and it saved it to the TODOLIST.txt file. (See Fig. 11, Fig.12 & Fig.13

Fig. 11

```
/usr/local/bin/python3.10 "/Users/tiyafarah/Documents/_PythonClass/Assignment 6/main.py"

        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Remove an existing item.
        4) Save Data to File
        5) Reload Data from File
        6) Exit Program


Which option would you like to perform? [1 to 6] - 2

What is the task? - Hike
What is the priority? [high|low] - Low
```

Fig. 12

```
Which option would you like to perform? [1 to 6] - 2

What is the task? - Hike
What is the priority? [high|low] - Low

******* The current items ToDo are: *******
q (low)
w (high)
h (low)
Hike (Low)
*****************************************
```

Fig.13

Assignment 6 〉 ToDoFile.txt

Project

ToDoFile.txt | Main.py

```
1  q,low
2  w,high
3  h,low
4  Hike,Low
5
```

Summary

I was able to verify that the script worked because it returned the value when I ran the script. It also allowed me to do all of the commands I worked on adding, saving, viewing and removing input.