

Dishanth Iyer

Feb 25, 2022

Foundations of Programming: Python

Assignment 06

<https://github.com/dishiyer/IntroToProg-Python/>

# Assignment 06: Functions

The sixth module focused on creating scripts using Functions. As part of the module, I also learned about Classes, and debugging.

## Preparing for Assignment

In preparation for completing the assignment, I followed the Assignment04.docx step by step to gather an understanding of the following:

- a. Functions, Parameters, Arguments
- b. Return values
- c. Global and Local variables
- d. Functions to organize code
- e. Function vs. Class
- f. Separations of Concern coding pattern
- g. Debugging
- h. GitHub

In order to do so, I reviewed the module video, the references and websites. I also reviewed the textbook material.

## Performing the Assignment

I followed the instructions to create the Folders in the C: drive and creating the Assignment starter script. In addition, I installed PyCharm and followed the directions to setup the environment and directory. I tried to use comments in script to remind myself about the organization and purpose of the code. I also watched the videos for the class to get an understanding of what is expected. I finally finished the assignment by uploading to GitHub.

I updated my assignment after reviewing the Assignment module.

## Create Python script

I created the Python script using the PyCharm IDE. I tried to add comments to explain the code as well.

Assignment06\_Dlyer.py - C:\PythonClass\Assignment06\Assignment06\_Dlyer.py (3.7.7)

File Edit Format Run Options Window Help

```
# ----- #
# 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
# DIyer,2.25.2022,Modified code to complete assignment 06
# ----- #

# Data ----- #
# Declare variables and constants
file_name_str = "ToDoFile.txt" # The name of the data file
file_obj = None # An object that represents a file
row_dic = {} # A row of data separated into elements of a dictionary {Task,Prio
table_lst = [] # A list that acts as a 'table' of rows
choice_str = "" # Captures the user option selection

# Processing ----- #
class Processor:
    """ Performs Processing tasks """

    @staticmethod
    def read_data_from_file(file_name, list_of_rows):
        """ Reads data from a file into a list of dictionary rows

        :param file_name: (string) with name of file:
        :param list_of_rows: (list) you want filled with file data:
        :return: (list) of dictionary rows
        """
        list_of_rows.clear() # clear current data
        file = open(file_name, "r")
        for line in file:
            task, priority = line.split(",")
            row = {"Task": task.strip(), "Priority": priority.strip()}
            list_of_rows.append(row)
        file.close()
```

```

@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()}
    list_of_rows.append(row)
    return list_of_rows

@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
    """
    for row in list_of_rows:
        if row["Task"].lower() == task.lower():
            list_of_rows.remove(row)
            print("Row Removed")
    return list_of_rows

@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
    """
    file = open(file_name, "w")
    for row in list_of_rows:
        file.write(row["Task"]+", "+row["Priority"] + "\n")

```

```
Assigment06_Dlyer.py - C:\PythonClass\Assignment06\Assignment06_Dlyer.py (3.7.7)
File Edit Format Run Options Window Help

    return list_of_rows

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

class IO:
    """ Performs Input and Output tasks """

    @staticmethod
    def output_menu_tasks():
        """ Display a menu of choices to the user

        :return: nothing
        """
        print('''
Menu of Options
1) Add a new Task
2) Remove an existing Task
3) Save Data to File
4) Exit Program
''')
        print() # Add an extra line for looks

    @staticmethod
    def input_menu_choice():
        """ Gets the menu choice from a user

        :return: string
        """
        choice = str(input("Which option would you like to perform? [1 to 4] - "))
        print() # Add an extra line for looks
        return choice

    @staticmethod
    def output_current_tasks_in_list(list_of_rows):
        """ Shows the current Tasks in the list of dictionaries rows

        :param list_of_rows: (list) of rows you want to display
```

```
Assignment06_Dlyer.py - C:\PythonClass\Assignment06\Assignment06_Dlyer.py (3.7.7)
File Edit Format Run Options Window Help

    return task, priority

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

        :return: (string) with task
        """
        task = str(input("What is the name of task you wish to remove? - ")).strip()
        print() # Add an extra line for looks
        return task

# Main Body of Script ----- #

# Step 1 - When the program starts, Load data from ToDoFile.txt.
Processor.read_data_from_file( file_name=file_name_str, list_of_rows=table_lst)

# Step 2 - Display a menu of choices to the user
while (True):
    # Step 3 Show current data
    IO.output_current_tasks_in_list(list_of_rows=table_lst) # Show current data
    IO.output_menu_tasks() # Shows menu
    choice_str = IO.input_menu_choice() # Get menu option

    # Step 4 - Process user's menu choice
    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

    elif choice_str == '2': # Remove an existing Task
        task = IO.input_task_to_remove()
        table_lst = Processor.remove_data_from_list(task=task, list_of_rows=table_lst)
        continue # to show the menu

    elif choice_str == '3': # Save Data to File
        table_lst = Processor.write_data_to_file(file_name=file_name_str, list_of_rows=table_lst)
        print("Data Saved!")
```

Utilizing the notes from the TA session and understanding how use Classes and Functions, I was able to complete the code.

## Test Script

I tested the script within the IDE using the 'Run' function. The script executed successfully. The test also output the text file.

Below is a PDF output of the "Run" window results from PyCharm.

```
File - Assignment06_DiYer
1 C:\_PythonClass\Assignment06\venv\Scripts\python.exe
  C:/_PythonClass/Assignment06/Assignment06_DiYer.py
2 ***** The current tasks ToDo are: *****
3 *****
4
5
6      Menu of Options
7      1) Add a new Task
8      2) Remove an existing Task
9      3) Save Data to File
10     4) Exit Program
11
12
13 Which option would you like to perform? [1 to 4] - 1
14
15 What is the task? - C
16 What is the priority? - low
17 ***** The current tasks ToDo are: *****
18 C (low)
19 *****
20
21      Menu of Options
22      1) Add a new Task
23      2) Remove an existing Task
24      3) Save Data to File
25      4) Exit Program
26
27
28
29 Which option would you like to perform? [1 to 4] - 2
30
31 What is the name of task you wish to remove? - c
32
33 Row Removed
34 ***** The current tasks ToDo are: *****
35 *****
36
37      Menu of Options
38      1) Add a new Task
39      2) Remove an existing Task
40      3) Save Data to File
41      4) Exit Program
42
43
```

Page 1 of 2

```
File - Assignment06_DiYer
44
45 Which option would you like to perform? [1 to 4] - 1
46
47 What is the task? - jo
48 What is the priority? - low
49 ***** The current tasks ToDo are: *****
50 jo (low)
51 *****
52
53      Menu of Options
54      1) Add a new Task
55      2) Remove an existing Task
56      3) Save Data to File
57      4) Exit Program
58
59
60
61 Which option would you like to perform? [1 to 4] - 3
62
63 Data Saved!
64 ***** The current tasks ToDo are: *****
65 jo (low)
66 *****
67
68      Menu of Options
69      1) Add a new Task
70      2) Remove an existing Task
71      3) Save Data to File
72      4) Exit Program
73
74
75
76 Which option would you like to perform? [1 to 4] - 4
77
78 Goodbye!
79
80 Process finished with exit code 0
81
```

Page 2 of 2

## Run Script

Once I was confident enough, I opened the Command Prompt in Windows and ran the script.

```
Command Prompt
4) Exit Program

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

What is the task? - add
What is the priority? - high
***** The current tasks ToDo are: *****
jo (low)
chore (low)
add (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] - 2

What is the name of task you wish to remove? - add

Row Removed
***** The current tasks ToDo are: *****
jo (low)
chore (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 ToDo are: *****
jo (low)
chore (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

Goodbye!

C:\_PythonClass\Assignment06>
```

```
ToDoFile.txt - Notepad
File Edit Format View H
jo,low
chore,low

Ln 1, Col 1
```

My code works as intended.