

Dmitriy Penzin

February 21, 2022

Foundations Of Programming: Python

<https://github.com/Taccdimas/IntroToProg-Python-Mod06>

Assignment 06 – Functions

The purpose of this assignment is to work with dictionaries, lists and two-dimensional data.

Functions in Python:

1. Block of statements
2. Allows to reuse the code
3. Can be well tested piece of code from previous projects
4. Call the function – call the block of code to run
5. Functions can have parameters (unlimited number)
6. Values passed into parameters are arguments
7. Help to divide code into 3 layers of concern.

Steps I took in performing Assignment 06:

1. I watched the module video
2. I read Chapter 6 of the book
3. I revied additional web pages and videos
4. I modified the script provided. Filled the missing steps. I found working with someone else's code quite challenging in terms of variable selection and overall organization.

Notes:

- I used PyCharm for this assignment as Integrated Development Environment.
- Every new input from the user was added to the end of the list variable with "append".
- Variable "File_obj" represents the file "ToDoList.txt". I don't provide any path to the file, so Python creates it in the same directory as script.
- I use access mode "w" which means that data in the file will be overwritten.
- I use access mode "r" to read from the file
- Closing the file with function "close" is not mandatory but a good practice.
- Assignment06 was also checked from the command prompt.

Running script:

```
Run: Assignment06_updated x
C:\_PythonClass\Assignment06\venv\Scripts\python.exe
***** The current tasks ToDo are: *****
study (med)
wash (high)
clean (med)
*****

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

Input TASK: cook
Input PRIORITY: high
***** The current tasks ToDo are: *****
study (med)
wash (high)
clean (med)
cook (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

Input TASK you want to remove: wash
***** The current tasks ToDo are: *****
study (med)
clean (med)
cook (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] - 3

Data Saved!

***** The current tasks ToDo are: *****
study (med)
clean (med)
cook (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] - 4

Goodbye!

Process finished with exit code 0
```

Step 1:

In order to add new task, I modified the “input_new_task_and_priority” function from IO class to capture user’s input. Processing function “add_data_to_list” added captured data to the table.

```
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!  
    task = input("Input TASK: ")  
    priority = input("Input PRIORITY: ")  
    return task, priority
```

```
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!  
    row_dic = {"Task": task, "Priority": priority}  
    table_lst.append(row_dic)  
    return list_of_rows
```

Step 2:

Removal of the data works similar to adding. IO function “input_task_to_remove” and processing function “remove_data_from_list” were modified. I also changed the variable “task” to “remove_task” in order to avoid shadowing. Processing function is looping through the rows of data and deletes a row if match is found.

```
def input_task_to_remove():  
    """ Gets the task name to be removed from the list  
  
    :return: (string) with task  
    """  
  
    # pass # TODO: Add Code Here!  
    remove_task = str(input("Input TASK you want to remove: "))  
    return remove_task
```

```
def remove_data_from_list(remove_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 i in list_of_rows:
        if i["Task"] == remove_task:
            list_of_rows.remove(i)
    return list_of_rows
```

Step 3:

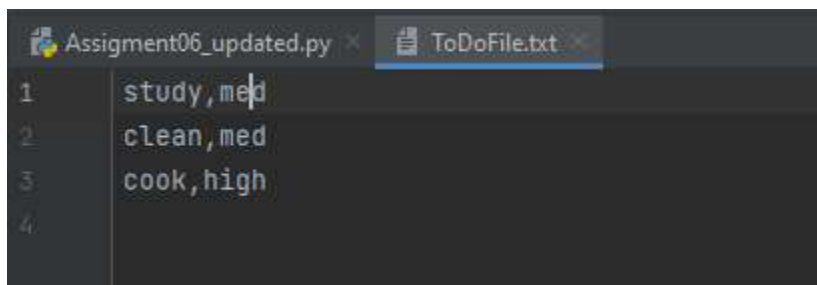
Saving data to the file. Access mode "w" is used. Only processing function "write_data_to_file" was modified. Saving was done by looping through the rows of data writing keys and values with the commas between them.

```
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!
    file_obj = open(file_name, "w")
    for i in list_of_rows:
        file_obj.write(str(i["Task"]) + "," + str(i["Priority"]) + "\n"))
    file_obj.close()
    return list_of_rows
```

TXT File:



```
1 study, med
2 clean, med
3 cook, high
4
```

Running from CMD:

```
Command Prompt - Python Assignment06_updated.py
C:\_PythonClass\Assignment06>Python Assignment06_updated.py
***** The current tasks ToDo are: *****
study (med)
clean (med)
cook (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] -
```

Summary:

During the work on this assignment, I learned how to work with two-dimensional data processed in functions. Functions were organized in classes. Module06 video and notes were very helpful.