

Letha Dunn

February 17, 2019

Foundations of Programming: Python

Assignment 06

# Functions, Classes, Methods

---

## Contents

Introduction	2
Preparing the assignment	2
Function to read the text file	2
Function to display the menu	3
Function to show the current to-do list	4
Function to add a task to the list	5
Function to remove a task from the list	5
Function to save the list to the text file	6
Better separating processing from presentation	7
Better separating step four	7
Better separating step five	8
Creating a class	9
Summary	9
Appendix: Running in Terminal	10

---

# Introduction

For Assignment 06 we were instructed to:

- Modify Prof. Root's to-do list script to use processing functions
- Put all of the processing functions in a class
- Update the functions to be methods

I successfully completed the assignment, tested it and have included it with this document in a zip file. This document describes how I completed the assignment.

## Preparing the assignment

I created a new project named Assignment06. In it I created a new Python file, Assignment06.py and copied Prof. Root's script into the new file. I performed a few cleanup tasks on the code before I really got started. I changed header to show that I what I'd done to the file.

```
#-----#
# Title: Functions, Class, Methods
# Dev:   Letha
# Date:  Feb 17, 2019
# ChangeLog: (Who, When, What)
#   Letha, 02/17/2019, Created script based on RRoot's Assignment05 answer script
#   Letha, 02/17/2019, Changed name of strData to lstData
#   Letha, 02/17/2019, Changed prompt for menu number from 4 to 5
#   Letha, 02/17/2019, Removed path to file
#-----#
```

I created a text file containing the initial to-do list to load into my program, todo.txt. It contained:

```
Clean House,Low
Pay Bills,High
```

I tested the new script with the minor changes I had made. This was to make sure I was starting off the assignment with working code.

## Function to read the text file

I identified that the Step 1 portion of the script, reading data from the text file, was entirely processing. So I moved it to a new processing function named ReadFileData().

```
# Processing for step 1
# When the program starts, load the any data you have
# in a text file called Todo.txt into a python Dictionary.
# This step is entirely processing.
'''This function shows reads a file, collects data in a dictionary, and writes it to a list table'''
def ReadFileData(FileName)
    objFile = open(FileName, "r")
    for line in objFile:
        lstData = line.split(",") # readline() reads a line of the data into 2 elements
        dicRow = {"Task":lstData[0].strip(), "Priority":lstData[1].strip()}
        lstTable.append(dicRow)
    objFile.close()
    return lstTable
```

I called the function later in my script.  
lstTable = ReadFileData(objFileName)

I tested my program with this new function and was delighted to discover that it still ran fine. I didn't break it!

## Function to display the menu

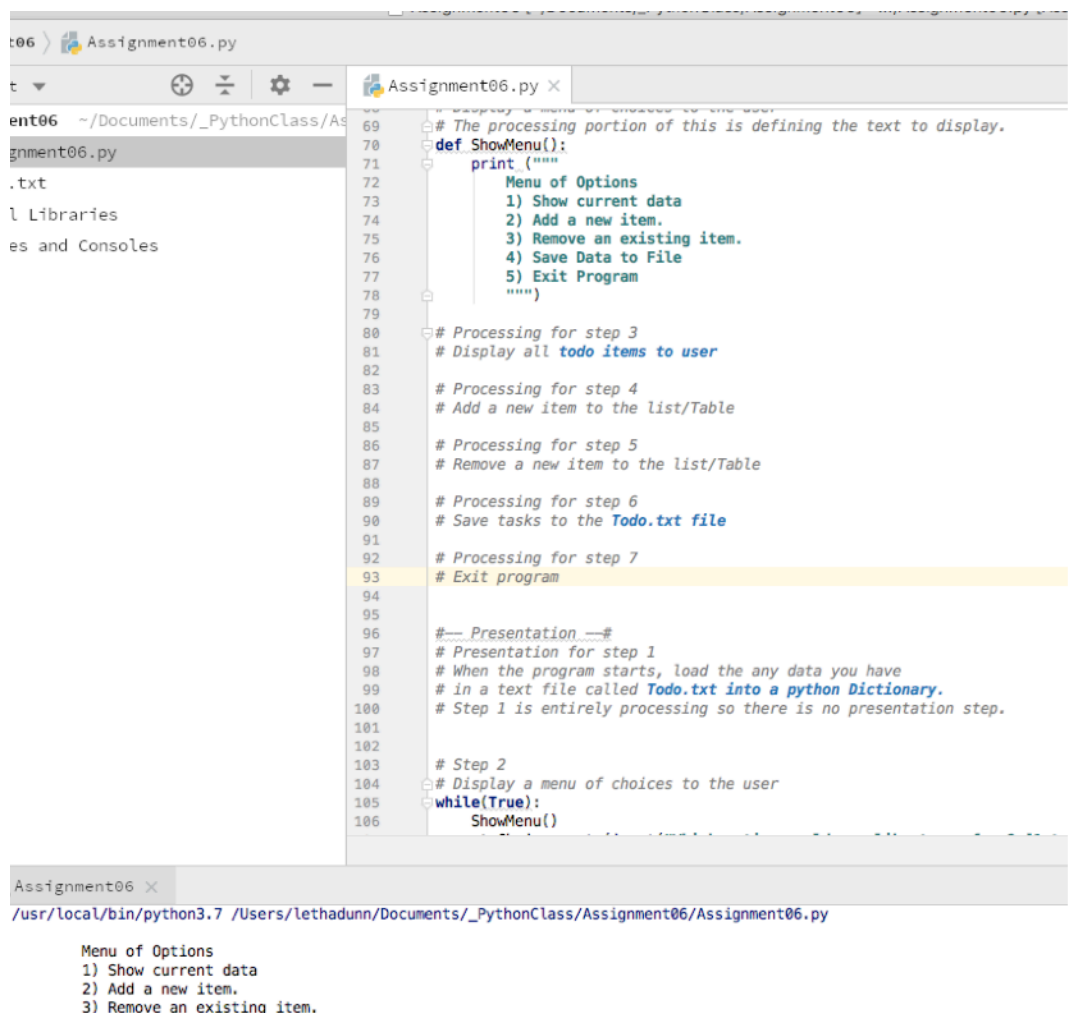
I created a ShowMenu() function to accomplish the repeated task of displaying a menu of options to the user:

```
# Processing for step 2
# Display a menu of choices to the user
# The processing portion of this is defining the text to display.
'''This function shows the menu of options'''
def ShowMenu():
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
    """)
```

I replaced the code to display the menu in the while loop.

ShowMenu()

I verified that it worked.



The screenshot shows a Python IDE with a file named 'Assignment06.py'. The code is as follows:

```
69 # The processing portion of this is defining the text to display.
70 def ShowMenu():
71     print("""
72     Menu of Options
73     1) Show current data
74     2) Add a new item.
75     3) Remove an existing item.
76     4) Save Data to File
77     5) Exit Program
78     """)
79
80 # Processing for step 3
81 # Display all todo items to user
82
83 # Processing for step 4
84 # Add a new item to the list/Table
85
86 # Processing for step 5
87 # Remove a new item to the list/Table
88
89 # Processing for step 6
90 # Save tasks to the Todo.txt file
91
92 # Processing for step 7
93 # Exit program
94
95
96 #--- Presentation ---#
97 # Presentation for step 1
98 # When the program starts, load the any data you have
99 # in a text file called Todo.txt into a python Dictionary.
100 # Step 1 is entirely processing so there is no presentation step.
101
102
103 # Step 2
104 # Display a menu of choices to the user
105 while(True):
106     ShowMenu()
```

The output window at the bottom shows the menu being displayed:

```
Assignment06 x
/usr/local/bin/python3.7 /Users/lethadunn/Documents/_PythonClass/Assignment06/Assignment06.py

Menu of Options
1) Show current data
2) Add a new item.
3) Remove an existing item.
```

## Function to show the current to-do list

Next I made a function named `ShowTaskList()` for showing the current to-do list, something that this program does often.

```
# Processing for step 3
# Display all todo items to user
# Processing portion is collecting the information to show to the user
'''This function collects the todo list to show to the user'''
def ShowTaskList():
    print("***** The current items ToDo are: *****")
    for row in lstTable:
        print(row["Task"] + "(" + row["Priority"] + ")")
    print("*****")
```

And I called it in the display section of the code.

```
if (strChoice.strip() == '1'):
    ShowTaskList()
```

I tested to confirm that I didn't break anything.

```
1) Show current data
2) Add a new item.
3) Remove an existing item.
4) Save Data to File
5) Exit Program
```

Which option would you like to perform? [1 to 5] - 1

```
***** The current items ToDo are: *****
Clean House(High)
Shovel Drive(High)
Weed Garden(Low)
*****
```

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

Once I confirmed that the function worked, I replaced the redundant to-do list print code with `ShowTaskList()` everywhere. I tested the entire program to confirm that the multiple changes still worked.

## Function to add a task to the list

My next step was to make a function named `AddTask()` that let the user add a new task to the list.

```
# Processing for step 4
# Add a new item to the list/Table
'''This function lets the user add a new task to the list'''
def AddTask():
    strTask = str(input("What is the task? - ")).strip()
    strPriority = str(input("What is the priority? [high|low] - ")).strip()
    dicRow = {"Task": strTask, "Priority": strPriority}
    lstTable.append(dicRow)
```

I called that function in the presentation portion of my code:

```
# Step 4
# Add a new item to the list/Table
elif(strChoice.strip() == '2'):
    AddTask()
    ShowTaskList()
```

I tested it. It worked.

## Function to remove a task from the list

Then I made a function named `RemoveTask()` that let the user remove a task from the list.

```
# Processing for step 5
# Remove a new item to the list/Table
'''This function lets the user remove a task from the list'''
def RemoveTask():
    # 5a-Allow user to indicate which row to delete
    strKeyToRemove = input("Which TASK would you like removed? - ")
    blnItemRemoved = False # Creating a boolean Flag
    intRowNumber = 0
    while (intRowNumber < len(lstTable)):
        if (strKeyToRemove == str(list(dict(lstTable[intRowNumber]).values())[0])): #
the values function creates a list!
            del lstTable[intRowNumber]
            blnItemRemoved = True
            # end if
            intRowNumber += 1
    # end for loop
    # 5b-Update user on the status
    if (blnItemRemoved == True):
        print("The task was removed.")
    else:
        print("I'm sorry, but I could not find that task.")
```

I called that function in the presentation portion of my code.

```
# Step 5
# Remove a new item to the list/Table
elif(strChoice == '3'):
    RemoveTask()
    ShowTaskList()
```

I tested it. It worked.

## Function to save the list to the text file

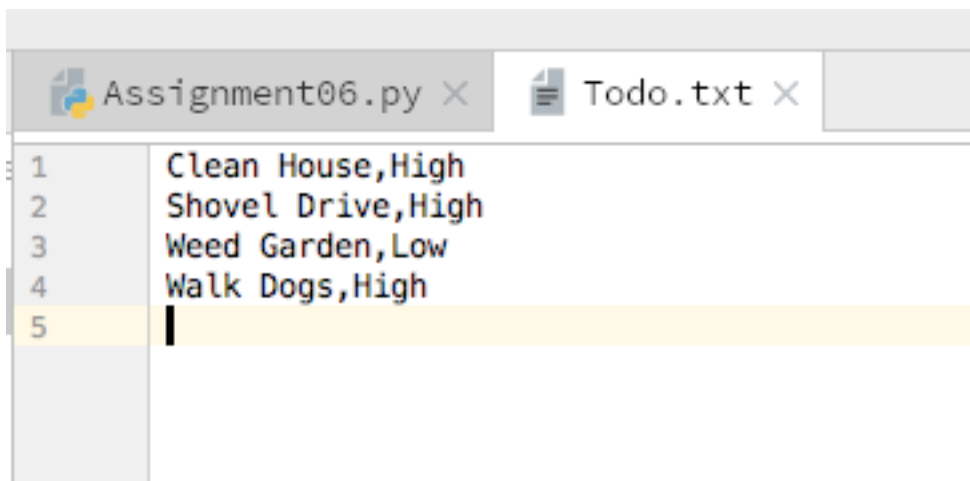
My next step was to make a function named `SaveToFile()` that saves the current list to a text file.

```
# Processing for step 6
# Save tasks to the Todo.txt file
'''This function saves the current list data to the file.'''
def SaveToFile(FileName):
    objFile = open(FileName, "w")
    for dicRow in lstTable:
        objFile.write(dicRow["Task"] + "," + dicRow["Priority"] + "\n")
    objFile.close()
```

I called that function in the presentation portion of my code.

```
# Step 6
# Save tasks to the ToDo.txt file
elif(strChoice == '4'):
    #5a Show the current items in the table
    ShowTaskList()
    #5b Ask if they want save that data
    if("y" == str(input("Save this data to file? (y/n) - ")).strip().lower()):
        SaveToFile(objFileName)
        input("Data saved to file! Press the [Enter] key to return to menu.")
    else:
        input("New data was NOT Saved, but previous data still exists! Press the [Enter] key to return to menu.")
    continue #to show the menu
```

I tested it by verifying that my text file contained the new task I added, Walk Dogs.



## Better separating processing from presentation

I read through my script and reviewed the assignment to determine if I was ready to move on to the next part. But when I reviewed steps 4 and 5 (adding and removing tasks) I decided I hadn't separated the processing from the presentation well enough.

To be safe, I made a backup of the Assignment06.py file that was working properly and moved forward to better separate processing from presentation in steps 4 and 5 of my script.

## Better separating step four

I made the AddTask() function accept two strings. It no longer matters how the function gets those strings. It simply adds the strings to the to-do list. This makes it easy to re-use. For example, in the future I could read strings from a different source and add them to the to-do list.

```
# Processing for step 4
# Add a new item to the list/Table
'''This function takes the user's new task and adds it to the list'''
def AddTask(strNewTask, strNewPriority):
    strTask = strNewTask
    strPriority = strNewPriority
    dicRow = {"Task": strTask, "Priority": strPriority}
    lstTable.append(dicRow)
```

The presentation portion of the script asks for the two strings. Then it calls the AddTask() function, feeding it those strings for processing.

```
# Step 4
# Add a new item to the list/Table
elif(strChoice.strip() == '2'):
    strNewTask = str(input("What is the task? - ")).strip()
    strNewPriority = str(input("What is the priority? [high|low] - ")).strip()
    AddTask(strNewTask, strNewPriority)
    ShowTaskList()
```

I tested it and it worked.

## Better separating step five

Similarly, I took the prompt for input out of the processing function `RemoveTask()`.

```
# Processing for step 5
# Remove a new item to the list/Table
'''This function tries to remove a task from the list'''
def RemoveTask(strTaskToRemove):
    # 5a-Allow user to indicate which row to delete
    strKeyToRemove = strTaskToRemove
    blnItemRemoved = False # Creating a boolean Flag
    intRowNumber = 0
    while (intRowNumber < len(lstTable)):
        if (strKeyToRemove == str(list(dict(lstTable[intRowNumber]).values())[0])): #
the values function creates a list!
            del lstTable[intRowNumber]
            blnItemRemoved = True
            # end if
            intRowNumber += 1
    # end for loop
    # 5b-Update user on the status
    if (blnItemRemoved == True):
        print("The task was removed.")
    else:
        print("I'm sorry, but I could not find that task.")
```

I added that prompt for input back in the presentation portion of the script. This part of the script still calls the `RemoveTask()` function, but this time it feeds it the string that the user input.

```
# Step 5
# Remove a new item to the list/Table
elif(strChoice == '3'):
    strTaskToRemove = input("Which TASK would you like removed? - ")
    RemoveTask(strTaskToRemove)
    ShowTaskList()
```

I tested it. It worked.

I tested the entire program in PyCharm and Terminal. It worked.



## Creating a class

I defined a new class to hold all the processing functions that I created.

```
# Create a Class to hold a list of functions
class ProcessToDoList(object):
    """ This class contains methods for managing a to-do list """
```

I indented everything in the processing section of the script to put it in the class. Because all of the functions were now in the class, I had to turn them into methods.

I made every one of my custom functions a method by adding this line before it:

```
@staticmethod
```

I'm not sure why yet. But I'll learn why soon.

Then because all my functions were now methods in the class `ProcessToDoList`, I had to change every call to them. Here is an example:

```
# Step 4
# Add a new item to the list/Table
elif(strChoice.strip() == '2'):
    strNewTask = str(input("What is the task? - ")).strip()
    strNewPriority = str(input("What is the priority? [high|low] - ")).strip()
    ProcessToDoList.AddTask(strNewTask, strNewPriority)
    ProcessToDoList.ShowTaskList()
```

I tested the changes in PyCharm and everything worked fine.

Everything worked in Terminal too. See the appendix for screen shots of the program running in Terminal.

## Summary

In Assignment 06, I:

- Modified Prof. Root's to-do list script to use processing functions
- Put all of the processing functions in a class
- Updated the functions to be methods

I successfully completed the assignment, tested it and have included it with this document in a zip file.

## Appendix: Running in Terminal

```
Assignment06 — -bash — 98x36
[Lethas-MacBook-Air:Assignment06 lethadunn$ Python3 Assignment06.py

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

Which option would you like to perform? [1 to 5] - 1

***** The current items ToDo are: *****
Clean House(High)
Shovel Drive(High)
Weed Garden(Low)
Walk Dogs(High)
*****

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

Which option would you like to perform? [1 to 5] - 2

What is the task? - Repair Fence
What is the priority? [high|low] - Low
***** The current items ToDo are: *****
Clean House(High)
Shovel Drive(High)
Weed Garden(Low)
Walk Dogs(High)
Repair Fence(Low)
*****
```

```
Assignment06 — -bash — 98x36

Which option would you like to perform? [1 to 5] - 3

Which TASK would you like removed? - Clean House
The task was removed.
***** The current items ToDo are: *****
Shovel Drive(High)
Weed Garden(Low)
Walk Dogs(High)
Repair Fence(Low)
*****

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

Which option would you like to perform? [1 to 5] - 3

Which TASK would you like removed? - dfdsg
I'm sorry, but I could not find that task.
***** The current items ToDo are: *****
Shovel Drive(High)
Weed Garden(Low)
Walk Dogs(High)
Repair Fence(Low)
*****

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

```
I'm sorry, but I could not find that task.
***** The current items ToDo are: *****
Shovel Drive(High)
Weed Garden(Low)
Walk Dogs(High)
Repair Fence(Low)
*****
```

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

Which option would you like to perform? [1 to 5] - 4

```
***** The current items ToDo are: *****
Shovel Drive(High)
Weed Garden(Low)
Walk Dogs(High)
Repair Fence(Low)
*****
```

```
Save this data to file? (y/n) - y
Data saved to file! Press the [Enter] key to return to menu.
```

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

Which option would you like to perform? [1 to 5] - 5

```
Lethas-MacBook-Air:Assignment06 lethadunn$
```