Maggie Valencia
February 19, 2023
IT FDN 110 A
Assignment06

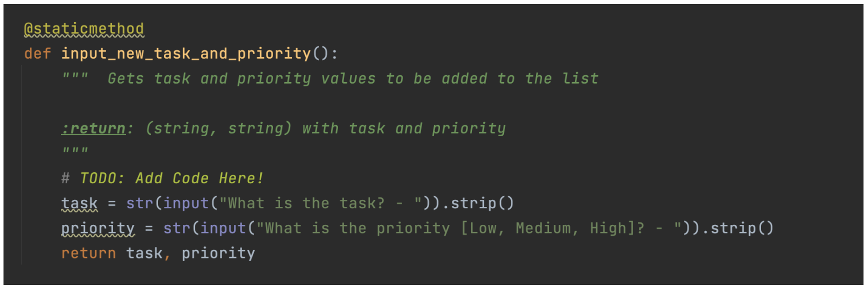
Assignment 06 - Function

Introduction

In Module 6, we learned that functions are used to perform a task and return control to the program. There are advantages to doing this in that it breaks up code into small size modules that are easier to create and work with. The idea with Functions is that they should focus on doing one job and doing that job well.

Step 1 - Add Data to List

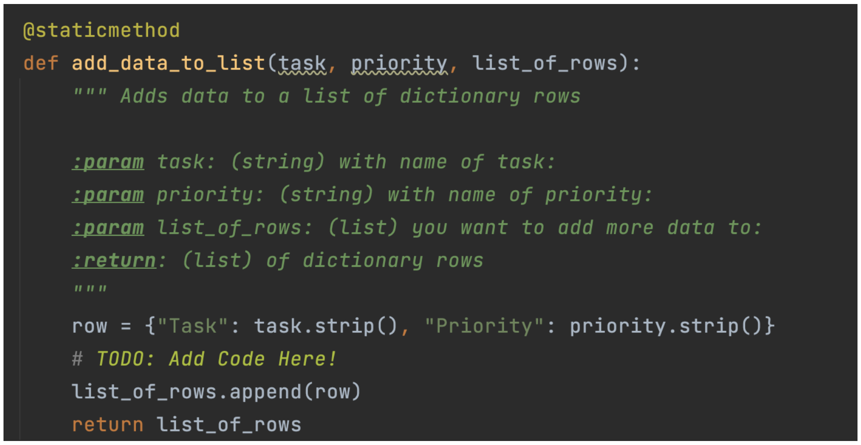
To add data to a list IO.input_new_task_and_priority() function (Figure 1) is modified with input request for task and priority to an user. We use the **return statement** so that the function can send objects back to the original code.



```
@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
    """
    # TODO: Add Code Here!
    task = str(input("What is the task? - ")).strip()
    priority = str(input("What is the priority [Low, Medium, High]? - ")).strip()
    return task, priority
```

Figure 1 - Input New Tasks

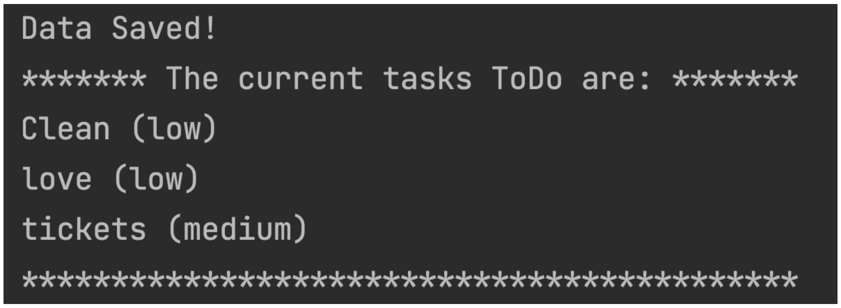
These objects are called function return value which can then be used to perform other processing by other functions as we can see in Figure 2 in the Processor.add_data_to_list function that is passed the task and priority from the prior function:



```
@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 to add more data to:
    :return: (list) of dictionary rows
    """
    row = {"Task": task.strip(), "Priority": priority.strip()}
    # TODO: Add Code Here!
    list_of_rows.append(row)
    return list_of_rows
```

Figure 2 - Add Data to a List

Figure 3 demonstrates that task and priority were appended to the table_lst variable:



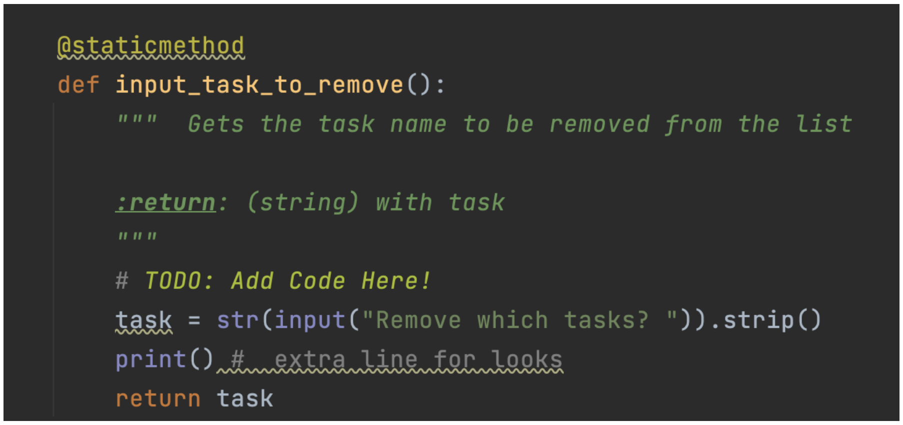
```
Data Saved!
******* The current tasks ToDo are: *******
Clean (low)
love (low)
tickets (medium)
*******************************************
```

Figure 3 - Add Tasks to the List

Key challenge with this work is wrapping your head around how functions actually work and how information is passed from the main body to the function. When a function does something simple like print, that is easy enough to grasp. Where there is struggle is with slightly more complex things, such as when a function has to process some data that passed to the function and then return some other data to be passed to another function., I am still struggling to wrap my head around this.

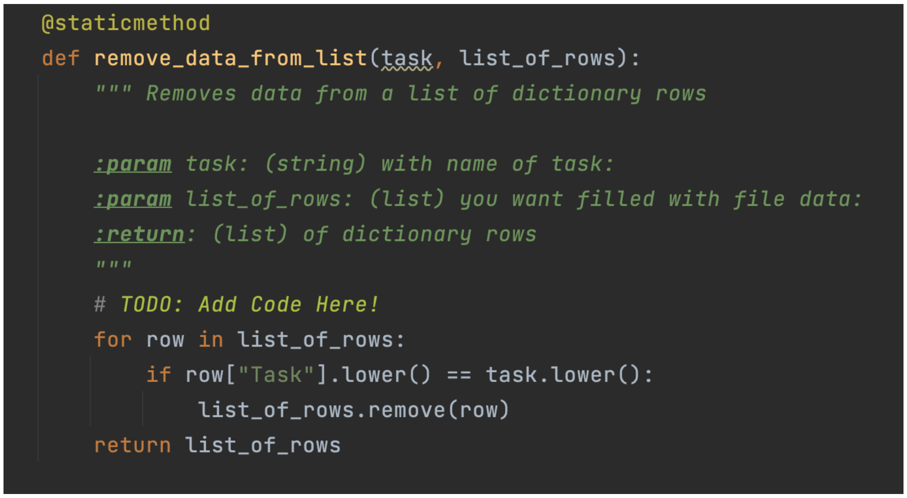
Step 2 - Remove data from list

Same as the previous example, we are using the IO.input_task_to_remove function in figure 4 to ask the user for a task to remove. These values are then returned to the original code for processing by the remove_data_from_list function in figure 5. In figure 6, we can see the task has been removed.



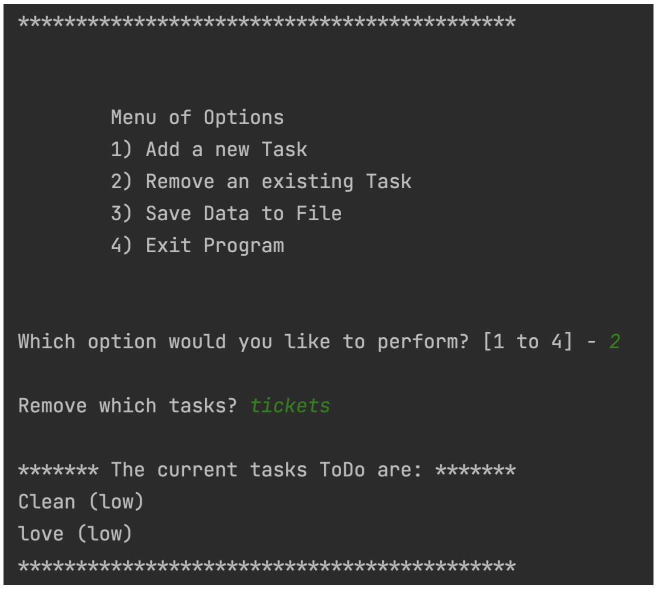
```
    @staticmethod
    def input_task_to_remove():
        """  Gets the task name to be removed from the list

        :return: (string) with task
        """
        # TODO: Add Code Here!
        task = str(input("Remove which tasks? ")).strip()
        print()  #  extra line for looks
        return task
```

Figure 4 - Ask for task to remove



```
@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!
    for row in list_of_rows:
        if row["Task"].lower() == task.lower():
            list_of_rows.remove(row)
    return list_of_rows
```

Figure 5 - Remove Task from the list

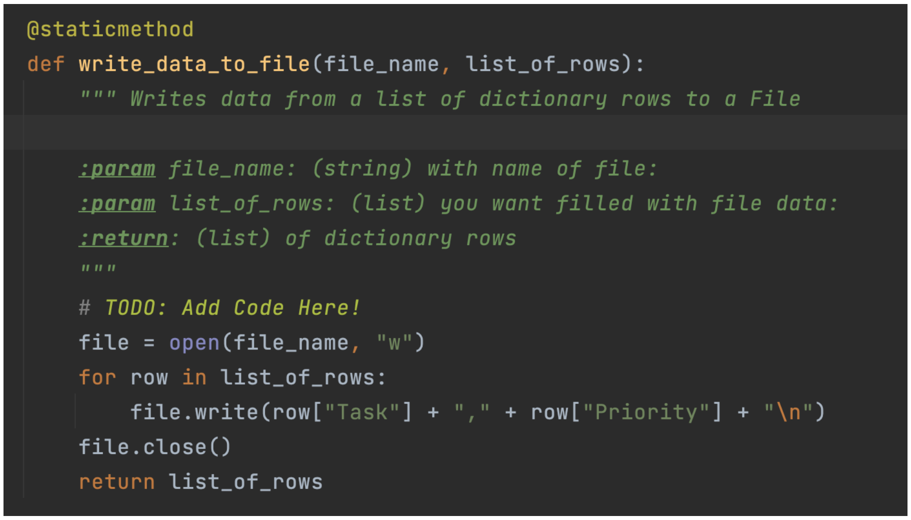


```
*******************************************


        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

Remove which tasks? tickets

******* The current tasks ToDo are: *******
Clean (low)
love (low)
*******************************************
```

Figure 6 - Success removing data from list

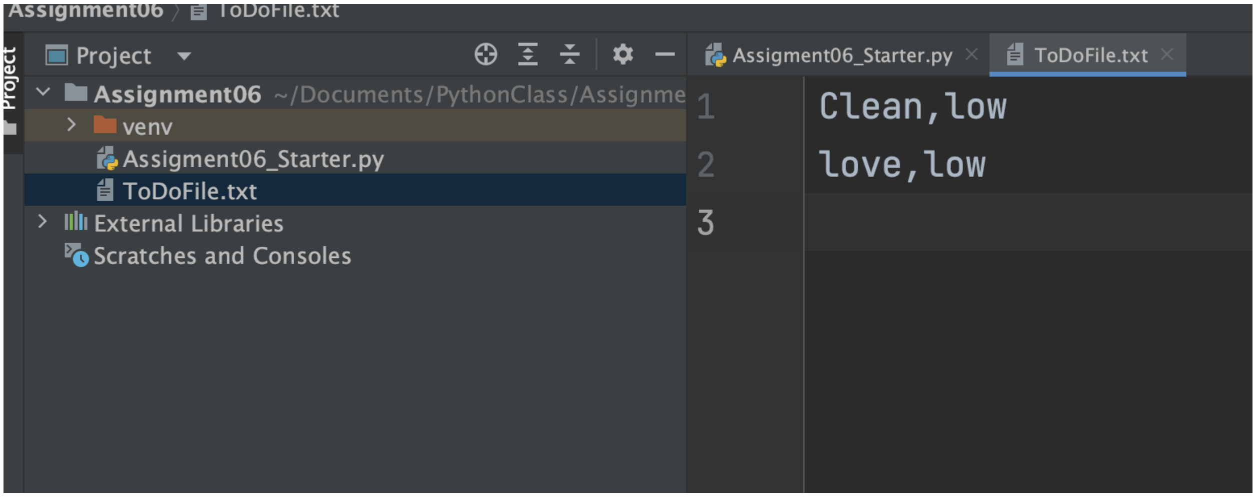
Step 3 - Write Data to File

Finally, we want to write data to a file in figure 7. I ensured that ToDoFile.txt was present in the same directory to avoid errors when running the program. Here we are opening a file and any data in the tablelst is saved to the existing file as can be seen in Figure 8.



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

Figure 7 - Writing Data to a file

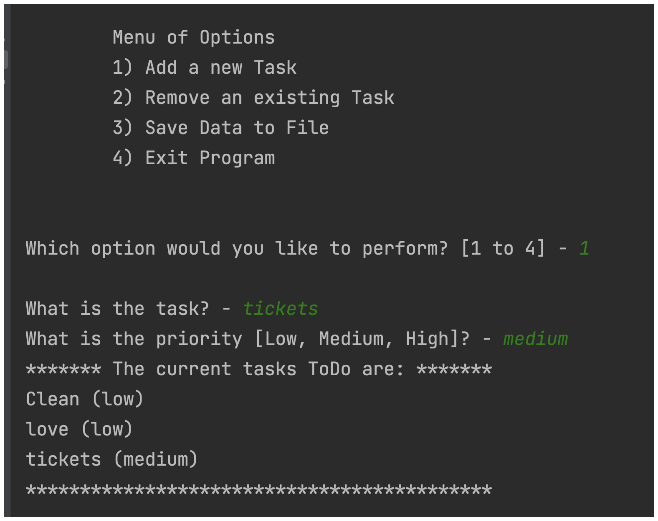


```
Clean,low
love,low
```

Figure 8 - Text file

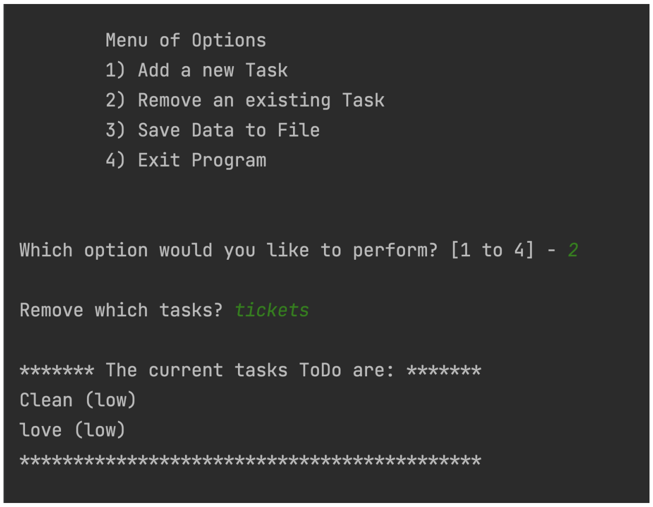
Step 5 - Code Working in PC

Below figures 8 through



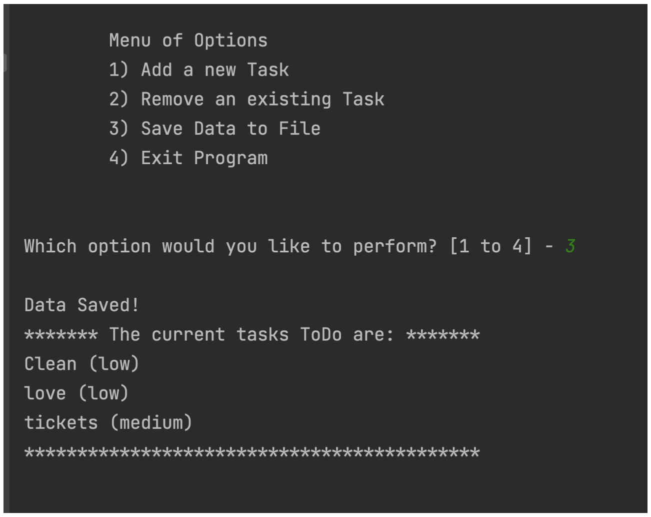
```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


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

What is the task? - tickets
What is the priority [Low, Medium, High]? - medium
******* The current tasks ToDo are: *******
Clean (low)
love (low)
tickets (medium)
*******************************************
```

Figure - 9 New Task



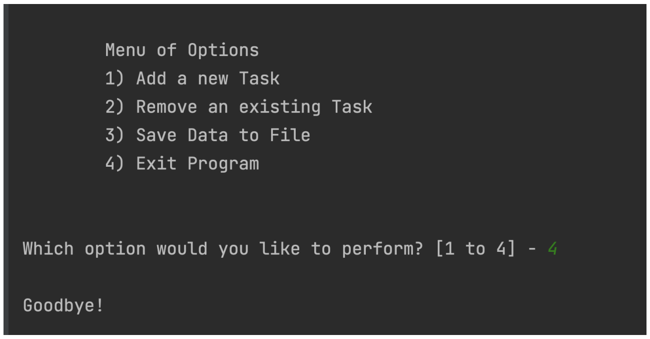
```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


Which option would you like to perform? [1 to 4] - 2

Remove which tasks? tickets

******* The current tasks ToDo are: *******
Clean (low)
love (low)
*******************************************
```

Figure - 10 Remove task



```
        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: *******
Clean (low)
love (low)
tickets (medium)
*******************************************
```

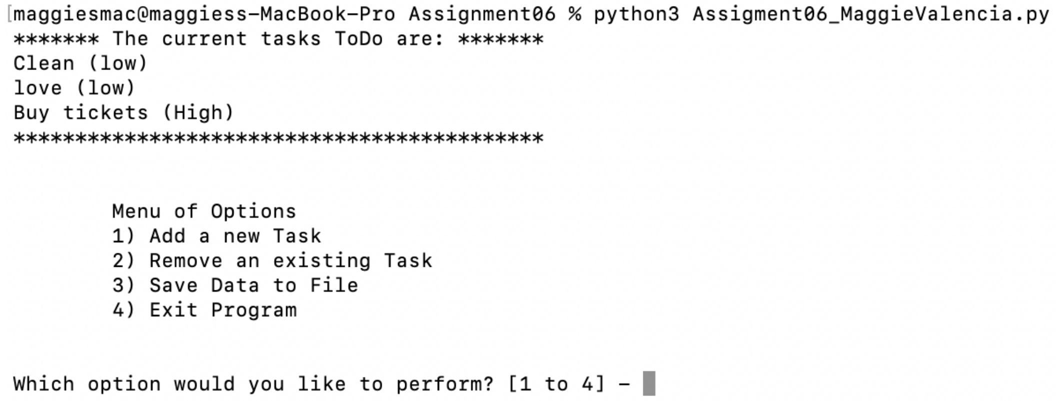
Figure 11 - Save task



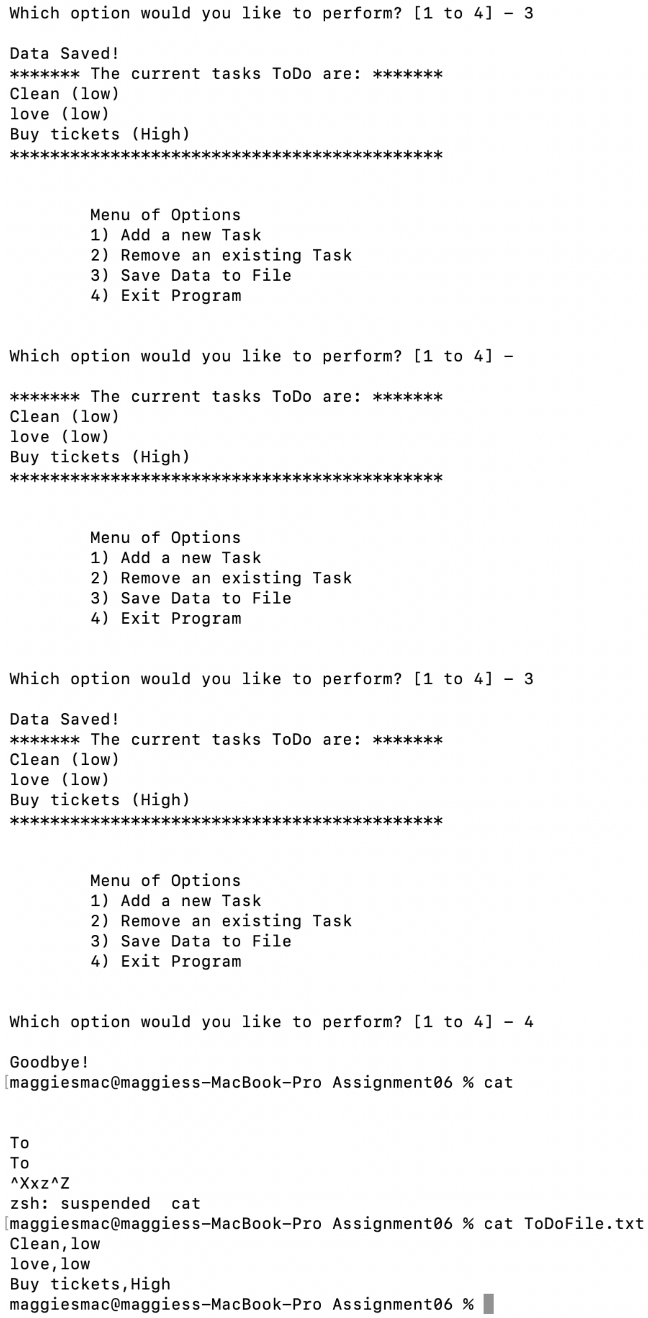
```
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
        3) Save Data to File
        4) Exit Program


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

Goodbye!
```

Figure 12 - End Program

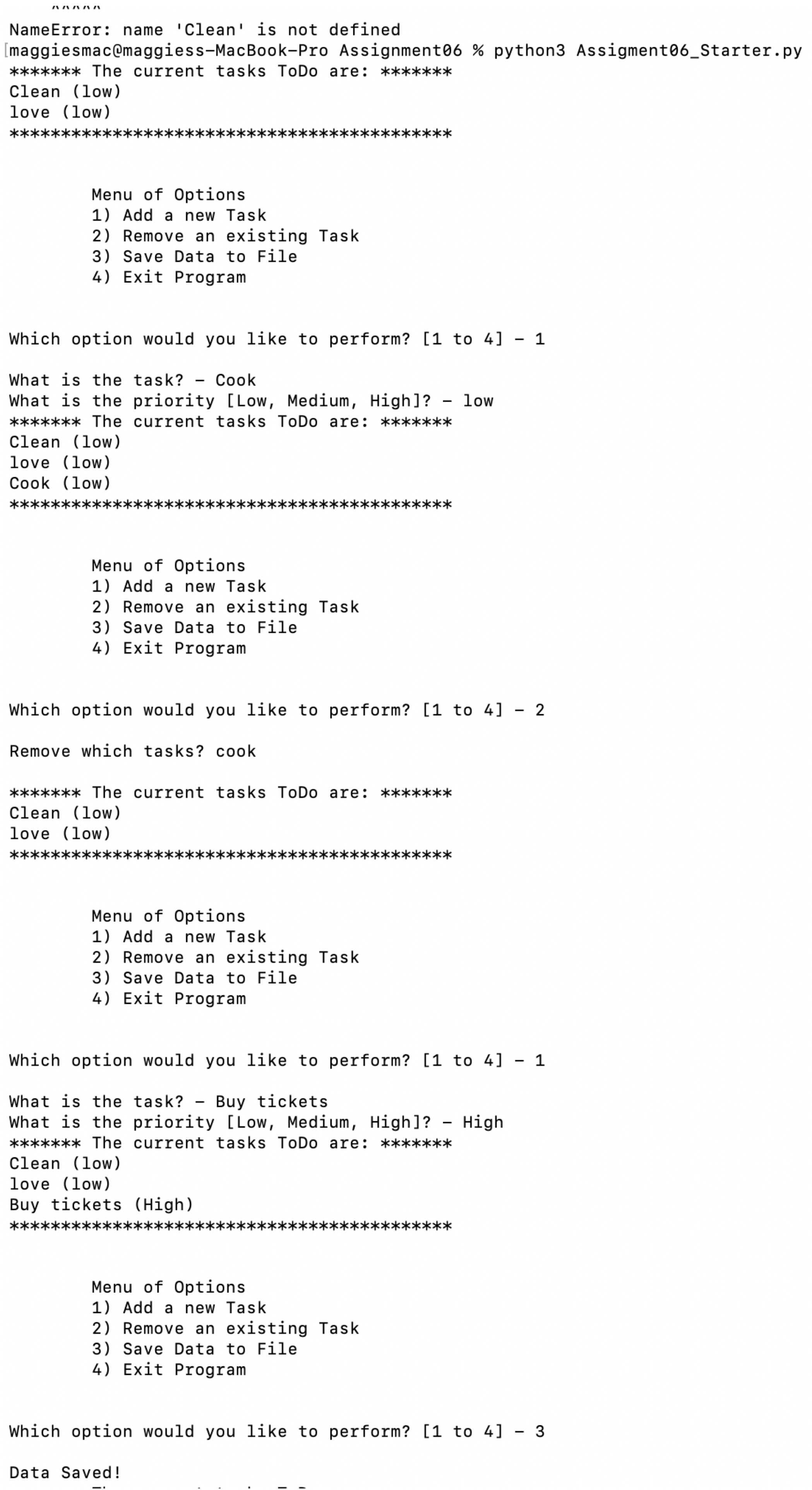
Step 6 - Code Working in Terminal



```
[maggiesmac@maggiess-MacBook-Pro Assignment06 % python3 Assigment06_MaggieValencia.py
******* The current tasks ToDo are: *******
Clean (low)
love (low)
Buy tickets (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] - 
```

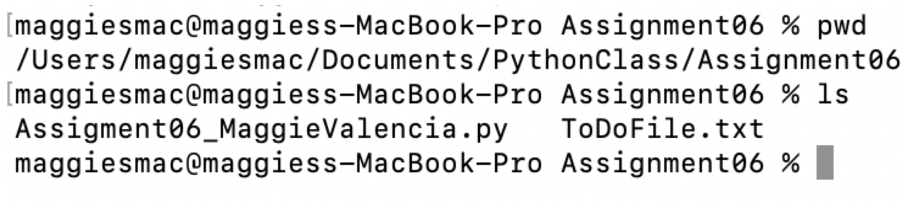


```
Which option would you like to perform? [1 to 4] - 3

Data Saved!
******* The current tasks ToDo are: *******
Clean (low)
love (low)
Buy tickets (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] -

******* The current tasks ToDo are: *******
Clean (low)
love (low)
Buy tickets (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: *******
Clean (low)
love (low)
Buy tickets (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!
[maggiesmac@maggiess-MacBook-Pro Assignment06 % cat


To
To
^Xxz^Z
zsh: suspended  cat
[maggiesmac@maggiess-MacBook-Pro Assignment06 % cat ToDoFile.txt
Clean,low
love,low
Buy tickets,High
maggiesmac@maggiess-MacBook-Pro Assignment06 % 
```



```
    ^^^^^
NameError: name 'Clean' is not defined
maggiesmac@maggiess-MacBook-Pro Assignment06 % python3 Assigment06_Starter.py
******* The current tasks ToDo are: *******
Clean (low)
love (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] - 1

What is the task? - Cook
What is the priority [Low, Medium, High]? - low
******* The current tasks ToDo are: *******
Clean (low)
love (low)
Cook (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] - 2

Remove which tasks? cook

******* The current tasks ToDo are: *******
Clean (low)
love (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] - 1

What is the task? - Buy tickets
What is the priority [Low, Medium, High]? - High
******* The current tasks ToDo are: *******
Clean (low)
love (low)
Buy tickets (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!
```

Text File and Code in same folder



```
[maggiesmac@maggiess-MacBook-Pro Assignment06 % pwd
 /Users/maggiesmac/Documents/PythonClass/Assignment06
[maggiesmac@maggiess-MacBook-Pro Assignment06 % ls
 Assigment06_MaggieValencia.py   ToDoFile.txt
 maggiesmac@maggiess-MacBook-Pro Assignment06 % 
```

Summary

In this module we learned how to work with Functions to create a more modular approach to running programs. Key learning is that working with many rows of code and on top of that working with someone else’s code is difficult. IIt is easy enough to get lost with what changes you made and how to trace-back. I found myself making a typo somewhere that was not obvious for a novice like myself to troubleshoot.

For this homework I changed my approach by doing the following:

1. Moving chunks of code to a sandbox and working there until things worked right and then moving back.
2. Tracking which functions where working correctly and collapsing those sections so I would not get confused
3. Heavily relied on the debugging tool to understand why things were not working right, the class video and Wednesday sessions were super helpful in this area and I wish we would have had that lesson much earlier - it would have made doing prior homework much easier.