Jennifer D. Smith

February 28, 2020

Foundations of Programming: Python

Assignment 06

The To-Do List with Classes and Functions

# Introduction

This week we expanded on the concept of Separation of Concerns with the introduction of classes and functions. We also increased use of GitHub and introduced GitHub Pages.

# Rebuilding a To Do List based on a someone else's code

This script is based on a supplied starter, which is essentially a refactoring of Assignment 5. The goal is to fully separate concerns and demonstrate the use of classes and functions. The classes and the required functions were ready to go, along with most of the main body. The missing parts were helpfully marked TODO. I borrowed heavily from my Assignment 5 code to complete this assignment, as we were expected to do.

I started by looking over the main body of the script and noting what each function was supposed to do. Then I started making each menu option work, starting with number 1. Option 1 required passing arguments based on user input and writing a new row (Figure 1).

A screenshot of a cell phone

Description automatically generated

Figure 1 Creating a new row with functions

For option 2, I needed to allow the user to remove a row and fill out the required functions to make it work (Figure 2).

A screenshot of a cell phone

Description automatically generated

Figure 2 Remove a row based on user input

First, get the key of the task to remove (Figure 3).

A screenshot of a cell phone

Description automatically generated

Figure 3 Get input

Then, remove the row from the file (Figure 4).

A screenshot of a cell phone

Description automatically generated

Figure 4 Remove a row

For option 3, I the requirement was to save the file with a confirmation message. I updated the write\_data\_to\_file function (Figure 5).

A screenshot of a cell phone

Description automatically generated

Figure 5 Save to file

Then I added the missing Processor.write\_data\_to\_file to Option 3 (Figure 6).

A screenshot of a cell phone

Description automatically generated

Figure 6 Write data to file

Option 4 was something we hadn't done in the previous assignment. This option would reload data from ToDoFile.txt and replace whatever was in memory. In this case, the line I needed was at the top of the main section under Step 1. I just had to modify the arguments (Figure 7).

A screenshot of a cell phone

Description automatically generated

Figure 7 Reloading data from a file

After I got the basic functionality working, I went back and added docstrings for all the functions and cleaned up the text.

When I went to do my Terminal run, I had managed to remember the shebang to make the file executable but failed to spell the file name correctly. Once I fixed that, everything worked as expected (Figures 8 - 11).

A close up of text on a black background

Description automatically generated

Figure 8 Finding a typo, running Option 1 and 2

A close up of text on a black background

Description automatically generated

Figure 9 Running Option 3 and adding some data to blow away with Option 4

A close up of text on a black background

Description automatically generated

Figure 10 Running Option 4 (and proof that Option 3 worked)

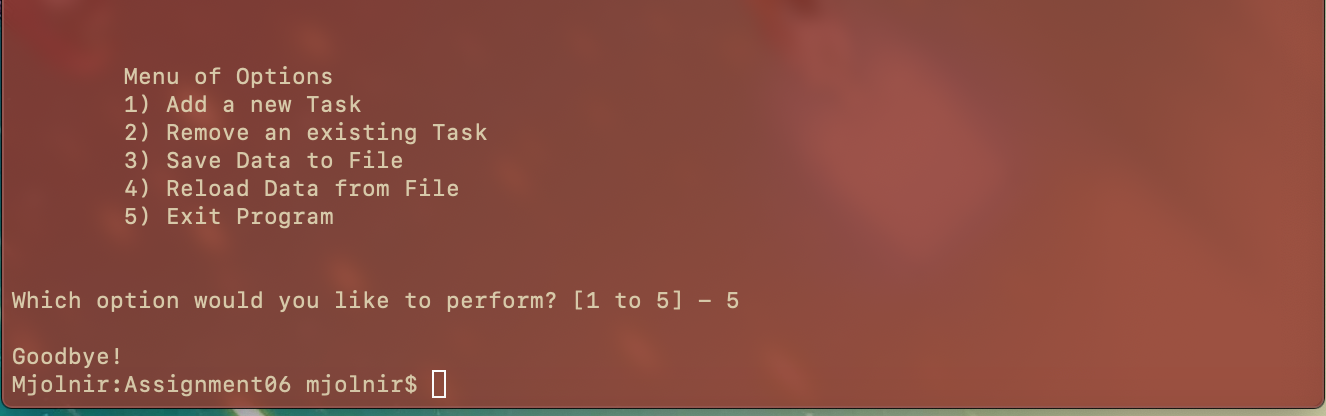


Figure 11 Option 5, the exit

# Conclusion

Using classes and functions makes code more modular and reusable. It does take some time to get used to moving back and forth between the function definitions and the main body of the code––this must be why professional developers often have a monitor set to portrait mode.

GitHub: <https://github.com/greeneyedsoandso/IntroToProg-Python-Mod06/tree/master/Assignment06>

GitHub Page: <https://greeneyedsoandso.github.io/IntroToProg-Python-Mod06/>