August 16, 2023

Foundations of Programming: Python

Assignment 06

Functions

Introduction

For our sixth assignment we had to build upon a starter code which included the creation and use of functions and classes. This script would display a five-item menu with options to display the data we obtained from our txt file, add, or remove a task and priority, save to txt file, and close the application.

Creating the script

For this script a framework was provided to us. We used similar code as last time (assignment 5) to get user input, we displayed a menu of five options that let the user select to add and remove a task and its priority, display current added value, and finally have the option to exit the application and save or not save the collected data into a txt file. All within the context of functions and classes.

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

Data Saved!
******* The current tasks ToDo are: *******
Run (Low)
Fish (Low)
Grade (Medium)
Hike (Low)
Code (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!
```

Python script run successfully on the terminal.

Summary

I learned a lot this module. How as your code starts to grow you will need to implement design patterns. The concept of separating code into categories: code for data, code for processing data and code for input/output (presentation layer code). Much of the software world is organized and logically divided this way.

You typically define all you functions and classes at the top of your script and then at the bottom is the main body of the script where you call all the listed functions and classes.

I also saw the power of using debugging tools and testing small sections of code to make sure the script works when put together with mockup data.

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


    1 usage
    @staticmethod
    def output_menu_tasks():
```

When creating a function as part of a class the decorator @staticmethod needs to be added as part of the class structure.