Miles Devine

8/7/2024

IT FDN 110

Assignment 06- Functions

Introduction

This week's module introduced us to creating our own functions and showed us how to group them using classes. We also looked into the difference between global and local variables and how they interact with functions. The module wrapped up by introducing us to the separation of concerns pattern.

The Assignment

The task we were given for this assignment was to create a Python program that uses constants, variables, print statements, flow control, conditional logic, dictionaries, and lists to display user entered information and save that information to a JSON file. This assignment was similar to Assignment 5, but the difference was that we needed to make our code more modular by accomplishing this using functions and classes.

We started our code with a header that contained important information for the user including the title, description of the program, and a change log. Next, we imported the json module so that we could work with json files (Fig. 1).

```
# ------------------------------------------------------------------------------------------ #
# Title: Assignment06
# Desc: This assignment demonstrates using functions
# with structured error handling
# Change Log: (Who, When, What)
#   Miles Devine, 8/7/2024, Created script
# ------------------------------------------------------------------------------------------ #
import json
```

Fig. 1- (Header with important information and imported module)

Next, we went on to define the constants as well as the variables that were not local to our functions. The value of the constants will not change throughout the script. The next step was to define our classes while making sure that we gave each of the descriptive document strings. These give the users insight into exactly what purpose the functions inside of the class serve (Fig. 2).

```python
class FileProcessor:
    """
    A collection of processing layer functions that work with Json files.

    (Who, When, What)
    M. Devine, 8/7/2024, Created Class
    """
    1 usage
    @staticmethod
    def read_data_from_file(file_name: str, student_data: list):
        """
        This function reads data from a file and saves it to a two-dimensional list.

        ChangeLog: (Who, When, What)
        M. Devine, 8/7/2024, Create function
        """
        try:
            file = open(file_name, "r")
            student_data = json.load(file)
            file.close()
        except FileNotFoundError as e:
            IO.output_error_messages( message: "The file must exist before running this script...\n",e)
        except Exception as e:
            IO.output_error_messages( message: "There was an unspecified error...\n",e)
        finally:
            if file.closed == False:
                file.close()
        return student_data
```

Fig. 2-(Descriptive document string for a class)

Once we had defined our functions we modified the Assignment 06 starter code and plugged in our functions. This greatly reduced the lines of code that were within the body of the script (Fig. 3).

```python
#Start the body of the text

students = FileProcessor.read_data_from_file(file_name=FILE_NAME,student_data=students) #Read the data from the file into a two-dimensional list

while (True): #Start the while loop
    IO.output_menu(menu=MENU) #Output the menu of options to the user

    menu_choice = IO.input_menu_choice() #Capture and store the user's choice

    if menu_choice == "1":
        IO.input_student_data(student_data=students) #Capture the user's inputs regarding student registration and append them to a two-dimensional list
        continue
    elif menu_choice == "2":
        IO.output_student_courses(student_data=students) #Display the current data
        continue
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students) #Save the current data to a json file
        continue
    elif menu_choice == "4":
        input("\nWaiting for you to press enter....") #Pause before exiting the program
        break #Break out of the while loop
```

Fig. 3- (Body of the script when using functions)

I tested the script in PyCharm as well as the command console to make sure that the program will work for a user regardless of where they are trying to run it from.

Summary

In summary, this assignment reinforced some of the lessons learned in previous weeks and gave us a chance to work through the new content from this module. While this assignment was similar to the last one, I feel it stepped up the difficulty a bit. Creating classes of functions gives us a different way to think about programming. It's a lot to think about but I can see the benefits of it, especially as our code continues to get more complex.