

Callie Weber
5/19/2024
Foundations of Python Programming
Assignment 06
<https://github.com/cweber08/IntoToProg-Python-Mod06>

Creating a Python Script

Introduction

This assignment builds on the last by adding common techniques for improving scripts: functions, classes, and the separation of concerns.

Drafting the Code

I started this assignment by opening the getting started script provided. It had all the processes laid out and everything pre-defined. I turned each step of the script into functions.

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

    Changelog: (Who, When, What)
    Cweber, 5/19/2024, Created Script
    """

    1 usage
    def read_data_from_file(file_name: str, student_data: list):
        """ This function reads data from a json file and loads it into a list of dictionary rows

        Changelog: ( Who, when, What)
        Cweber, 5/19/2024, Created function

        :return: list
        """
        try:
            file = open(file_name, "r")
            student_data = json.load(file)
            file.close()
        except Exception as e:
            IO.output_error_message(message="Error: There was a problem with reading the file.", error=e)

        finally:
            if file.closed == False:
                file.close()
        return student_data
```

I also created a function for step three where our data was being written out to a json and then cleaned up the code for step 3 to call the function:

Callie Weber

5/19/2024

Foundations of Python Programming

Assignment 06

<https://github.com/cweber08/IntoToProg-Python-Mod06>

```
def write_data_to_file(file_name: str, student_data: list):
    """ This function writes data to a json file and with data from a list of dictionary rows

    Changelog: ( Who, when, What)
    Cweber, 5/19/2024, Created function

    :return: None
    """
    try:
        file = open(file_name, "w")
        json.dump(student_data, file)
        file.close()
        IO.output_student_and_course_names(student_data=student_data)
    except Exception as e:
        message = "Error: There was a problem with writing to the file.\n"
        message += "Please check that the file is not open by another program."
        IO.output_error_message(message=message, error=e)
    finally:
        if file.closed == False:
            file.close()
```

To keep this short, I did very similar work under the # Presentation” header replacing steps 1, 2, and 3 with functions as well. A lot of the code that was originally under each if/elif statement could easily be repurposed for the functions, this makes the code for the if/elif statements much shorter and cleaner to simply call a function that’s defined once:

```
# Present choices and Process the data
while (True):
    IO.out_menu(menu=MENU)
    menu_choice = IO.input_menu_choice()

    # Input user data
    if menu_choice == "1": # This will not work if it is an integer!
        students = IO.input_student_data(student_data=students)
        continue

    # Present the current data
    elif menu_choice == "2":
        IO.output_student_and_course_names(students)
        continue

    # Save the data to a file
    elif menu_choice == "3":
        FileProcessor.write_data_to_file(file_name=FILE_NAME, student_data=students)
        continue

    # Stop the loop
    elif menu_choice == "4":
        break # out of the loop
    else:
        print("Please only choose option 1, 2, or 3")

print("Program Ended")
```

Additionally there is a variety of error messaging for each function so the user can identify if/when errors occurred.

Callie Weber
5/19/2024
Foundations of Python Programming
Assignment 06
<https://github.com/cweber08/IntoToProg-Python-Mod06>

Summary

This homework was lengthy and took me quite a while to put it together. I'm very grateful for the homework demo video which helped me in understanding quite a few of the functions much better.