Arithmetic Formatter

A Simple Project from FreeCodeCamp

1. Project Description:
Arithmetic Formatter is a simple and deliberately limited math calculating function. It can only add and subtract numbers of a maximum length of 4 digits. This project was one of the recommended projects at the end of the course 'Scientific Computing with Python' from freecodecamp.org that I had just completed. This project was 50/50 self-guided and follow along tutorials.

2. Problem Statement:
The problem this project aimed to solve was my complete lack of any coding experience outside of following along to tutorials. Up until this point I had never coded in a self-guided environment.

3. Solution Implementation:
The solution was a direct response to the 'Arithmetic Formatter' assignment at the end of the 'Scientific Computer with Python' course on freecodecamp.org. The main goal was to create a function that receives a list of strings that are arithmetic problems and returns the problems arranged vertically, side-by-side and with rules attached. The solution was as follows:
1. Imported REGEX library
2. Used the def() function to create an object called 'arithmetic_arranger()'
3. 'arithmetic_arranger()' takes 2 arguments with the second being a Boolean, 'problems' and 'solve = True'
4. Next I created 4 empty strings that represent the formatting of each row that will be printed at the end in the terminal (first, second, lines, calculated)
5. Then I created a for loop to iterate through 'problems', the first argument of 'arithmetic_arranger()'
6. Used split() to break up the equation into desired segments
7. Then I wrote a series of 'if' and 'elif' statements to encode the limits and appropriate error codes
8. The remainder of the solution I used was from a YouTube tutorial (credit below) from 5:09-7:00
9. The solution was tedious formatting and involved max(), len(), rjust(), int(), str(), and print() functions, and the above 4 variables (first, second, lines, calculated)
10. This part of the solution involved a lot of trial and error with inputting the correct rjust() values so the rows in the equation would line up when printed in the terminal
11. Then I used an 'if' statement, a bullion, print(), and the new line function ('\n'), to say that if 'solve' (the 2nd argument arithmetic arranger takes) = false, print variable 'first', 'second' and 'lines' with a new line in between, so that it prints just the equations without the answer
12. Finally, I used the 'else' statement to say if 'solve' = true, print variable 'first', 'second', 'lines' and 'calculated', to print the equations with the answer, as specified by the assignment

Tutorial I followed: https://youtu.be/6X6pj92PQiw?t=300

4. Results and Impact:
The result is a functioning and intentionally limited math solving function. Although the function did not have practical use, the project served its purpose by providing me the opportunity to code in a self-guided environment for the first time and become more familiar with Python in general.

5. Technical Skills Demonstrated:
Regex, object creation, strings, for loops, if statements, elif statements, else statements, Booleans, the split(), len(), print(), max(), rjust() and range() functions

6. GitHub Repository:

https://github.com/Davidooj/Projects/blob/main/Arithmatic_Formatter.py

7. Key Takeaways:
Breaking Down Problems: Initially daunting, the 'Arithmetic Formatter' assignment became manageable when broken down into steps. This process highlighted the importance of breaking down problems into smaller steps both for learning and execution. Also, it helped define the parts of the assignment I couldn’t complete, which guided me to seek tutorials to learn.

Learning from Others' Problem-Solving: Watching tutorials and observing others' approaches to coding challenges was immensely insightful. It provided new strategies and perspectives into other problem-solving methods, thereby enhancing my understanding of the problem and its solution.