**Arithmetic Formatter  
A Simple Project From FreeCodeCamp**

**1. Project Description:**Arithmetic Formatteris 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’ 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 follow:  
1. Imported REGEX library  
2. Used the def() function to create an object called ‘arithmetic\_arranger()’  
3. ‘arithmetic arranger’ takes 2 arguments with one being a bullion, ‘problems’ and ‘solve = True’  
4. Next I created 4 empty strings that represent the formatting and each row that will be printed (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 remaining 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, to print 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 show the equations with the answer as specified by to 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, bullions, 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, and therefor deferring to assistance 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.