Overview of Application

My client-server application allows the client to send mathematical expressions to the server for evaluation. The server performs the requested calculations and returns the results to the client. The implemented operations include:

- Calculating the power of a number: calculate base ^ exponent
- Calculating the nth root of a number: calculate root degree number

The application is designed to handle these operations over a network connection using socket communication. The client connects to the server, sends mathematical expressions as commands, and receives the computed results.

Client → Server Message Format

The client sends mathematical expressions to the server using the following message format:

Command: <operation> <operands>

- <operation> is a string representing the desired mathematical operation. It can be either "calculate" for power calculation or "calculate root" for root calculation.
- <operands> are the numbers involved in the operation, such as the base, exponent, and degree, separated by spaces.

Examples:

- To calculate the power of 2^3, the client sends: calculate 2 ^ 3
- To calculate the cube root of 27, the client sends: calculate root 3 27

Server → Client Message Format

The server responds to the client with messages that include the results of the mathematical operations or error messages. The server's response format is as follows:

Response: <message>
<message> is a string containing either the computed result or an error message.
Examples:

- For the calculation 2 ^ 3, the server responds with: Response: The result of 2 ^ 3 is 8.0.
- For the calculation root 3 27, the server responds with: Response: The 3-th root of 27 is 3.0.

In case of an error, the server might respond with: Response: Invalid expression format. Use 'calculate base ^ exponent' or 'calculate root degree number'.

Example Output

Here are examples of expected output for the implemented RPC (Remote Procedure Call) operations:

Client sends calculate 2 ^ 3:

Server responds with: Response: The result of 2 ^ 3 is 8.0.
Client sends calculate root 3 27:

Server responds with: Response: The 3-th root of 27 is 3.0.
Client sends calculate 4 ^ 0.5:

Server responds with: Response: The result of 4 ^ 0.5 is 2.0.
Client sends an invalid command, e.g., calculate invalid command:

Server responds with: Response: Invalid command.

Acknowledgments

I am, admittedly, a bit out of practice when it comes to Python. While I'm more comfortable with Java, I'm not overly worried about the Python aspects of this class. While this assignment may have taken me a bit longer than my peers, I'm still happy with my final results. I should acknowledge, however, that I used GeeksForGeeks as well as W3Schools to refamiliarize myself with the syntax.