1. **implicit and explicit in javascript addition is effected by string but subtraction and multiplication are not effected.**

In JavaScript, the behavior you're describing is related to the concept of type coercion. JavaScript is a loosely typed language, and it tries to perform operations even when the operands are of different types. When you use the + operator with a string and another type, JavaScript will attempt to convert the non-string operand to a string and perform concatenation.

Here's an example:

let result = "5" + 2; // result is the string "52"

In this case, the number 2 is implicitly converted to a string and then concatenated with the string "5."

However, this behavior is specific to the + operator. The subtraction (-) and multiplication (\*) operators in JavaScript are not designed for string concatenation. If you try to use them with a string and a non-string operand, JavaScript will attempt to convert the string to a number (if possible) and perform the operation as numeric subtraction or multiplication.

Here's an example:

let resultSubtraction = "5" - 2; // result Subtraction is the number 3

let resultMultiplication = "5" \* 2; // result Multiplication is the number 10

In these cases, the string "5" is implicitly converted to a number before the subtraction and multiplication operations are performed.

It's important to be aware of these behaviors to avoid unexpected results in your code. If you want to ensure that you are working with numbers, you may want to explicitly convert strings to numbers using functions like parseInt or parseFloat.