***Declaring variable name in javascript***

**🡪**Choosing meaningful and descriptive variable names is crucial in JavaScript (and in programming in general) for several reasons:

🡪Code Readability: Clear and descriptive variable names make the code more readable and understandable. When someone reads the code, they can quickly grasp the purpose and usage of each variable without needing to dive into its context.

🡪Documentation: Descriptive variable names serve as self-documentation. They convey the intent and purpose of the variable, reducing the need for extensive comments to explain its role in the code.

🡪Maintenance: When you or other developers revisit the code later for maintenance or updates, meaningful variable names make it easier to understand the code's logic and make modifications without introducing errors.

🡪Reduced Cognitive Load: Well-named variables reduce cognitive load by providing clear cues about what the variable represents or stores. Developers can focus on understanding the problem-solving logic rather than trying to decipher cryptic variable names.

**Example:**

🡪// Unclear variable names

let x = 10; // Number of items in stock

let y = 5; // Number of items sold

let z = x - y; // Calculate remaining items

console.log(z); // Output: 5

🡪// Clear variable names

let itemsInStock = 10;

let itemsSold = 5;

let remainingItems = itemsInStock - itemsSold;

console.log(remainingItems); // Output: 5

🡪In the first part of the example, the variables x, y, and z are used to represent the number of items in stock, the number of items sold, and the calculation of remaining items, respectively. While the code technically works, it's difficult to understand at a glance what each variable represents.

🡪In contrast, the second part of the example uses descriptive variable names like itemsInStock, itemsSold, and remainingItems, making it immediately clear what each variable represents. This improves the code's readability and makes it easier to understand the logic behind the calculations.