JavaScript operators

JavaScript operators are fundamental symbols that perform operations on values and variables. These values are called operands. Operators are categorized based on the type of operation they perform and the number of operands they require.

Here's a brief explanation of the main types of operators in JavaScript:

1. **Assignment Operators:**
   - Used to assign values to variables.
   - The most common is the simple assignment operator =.
   - Shorthand assignment operators combine an arithmetic or bitwise operation with assignment, e.g., +=, -=, *=, /=, %=, **=.
   - **Example:** let x = 10; x += 5; // x is now 15
2. **Arithmetic Operators:**
   - Perform mathematical calculations.
   - Include: + (addition), - (subtraction), * (multiplication), / (division), % (modulus - remainder after division), ** (exponentiation), ++ (increment), -- (decrement).
   - **Example:** let result = 10 + 5; // result is 15
3. **Comparison Operators:**
   - Compare two values and return a Boolean (true or false) result.
   - Include: == (loose equality), === (strict equality), != (loose inequality), !== (strict inequality), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to).[1]
   - **Example:** let isEqual = (5 === '5'); // isEqual is false (strict equality checks type)
4. **Logical Operators:**
   - Combine or modify Boolean expressions.
   - Include: && (logical AND), || (logical OR), ! (logical NOT).
   - **Example:** if (age > 18 && hasLicense) { /* do something */ }
5. **Bitwise Operators:**
   - Perform operations on the binary representation of numbers.
   - Less commonly used in general web development but important for low-level operations.
   - Include: & (AND), | (OR), ^ (XOR), ~ (NOT), << (left shift), >> (right shift), >>> (unsigned right shift).
   - **Example:** let x = 5 & 1; // x is 1 (binary 101 & 001 = 001)
6. **String Operators:**
   - Primarily the + operator for string concatenation.
   - **Example:** let fullName = "John" + " " + "Doe"; // fullName is "John Doe"
7. **Ternary (Conditional) Operator:**
   - A shorthand for an if...else statement.
   - Syntax: condition ? expressionIfTrue : expressionIfFalse
   - **Example:** let status = (age >= 18) ? "Adult" : "Minor";

8. **Type Operators:**
    - typeof: Returns a string indicating the type of an operand.
    - instanceof: Checks if an object is an instance of a particular class or constructor function.
    - **Example:** console.log(typeof 10); // "number"
9. **Spread Operator (...):**
    - Expands an iterable (like an array or string) into individual elements.
    - Useful for creating new arrays/objects, passing arguments to functions.
    - **Example:** let arr1 = [1, 2]; let arr2 = [...arr1, 3, 4]; // arr2 is [1, 2, 3, 4]
10. **Nullish Coalescing Operator (??):**
    - Provides a default value when the left-hand operand is null or undefined.
    - **Example:** let name = user.name ?? 'Guest';
11. **Optional Chaining Operator (?.):**
    - Allows you to safely access properties of an object that might be null or undefined without throwing an error.
    - **Example:** let city = user.address?.city;

Understanding these operators is crucial for writing effective and efficient JavaScript code, as they dictate how data is manipulated and how program flow is controlled.