perplexity

operators in js

JavaScript uses **operators** to perform operations on values and variables (known as operands). Operators are classified based on the kind of operations they perform. Here are the major types of operators in JavaScript, with examples:

1. Arithmetic Operators

Used for mathematical calculations:

- Addition: `+`
- Subtraction: `-`
- Multiplication: `*`
- Division: `/`
- Remainder (modulo): `%`
- Exponentiation: `**`
- Increment: `++`
- Decrement: `--`

Example:

```
let x = 5;
let y = 2;
let sum = x + y; // 7
let power = x ** y; // 25
```

2. Assignment Operators

Assign values to variables:

- Simple assignment: `=`
- Addition assignment: `+=`
- Subtraction assignment: `-=`
- Multiplication assignment: `*=`
- Division assignment: `/=`
- Remainder assignment: `%=`

Example:

```
let a = 10;
a += 5; // a is now 15
```

3. Comparison Operators

Compare two values:

- Equal: `==`
- Strict equal (type + value): `===`
- Not equal: `!=`
- Strict not equal: `!==`
- Greater than: `>`
- Less than: `<`
- Greater than or equal: `>=`
- Less than or equal: `<=`

Example:

```
5 == '5'; // true
5 === '5'; // false
```

4. Logical Operators

Combine or invert boolean values:

- Logical AND: `&&`
- Logical OR: `||`
- Logical NOT: `!`

Example:

```
true && false; // false
true || false; // true
!true; // false
```

5. String Operators

- Concatenation: `+` (adds strings)

Example:

```
let greeting = "Hello" + " " + "world"; // "Hello world"
```

6. Bitwise Operators

Operate at the bit level:

- AND: `&`
- OR: `|`
- XOR: `^`
- NOT: `~`
- Left shift: `<<`
- Right shift: `>>`
- Zero-fill right shift: `>>>`

Example:

```
5 & 1; // 1
```

7. Ternary Operator (Conditional)

A shorthand for `if…else`:

- `condition ? expr1 : expr2`

Example:

```
let status = age >= 18 ? "Adult" : "Minor";
```

8. Type Operators

- `typeof`: Returns the type of a variable
- `instanceof`: Checks if an object is an instance of a class or constructor

Example:

```
typeof 5; // "number"
arr instanceof Array; // true
```

9. Miscellaneous

- `delete`: Removes a property from an object
- `void`: Discards an expression’s return value
- `,` (comma): Evaluates multiple expressions and returns the last

Example:

```
let x = (a = 1, b = 2, a + b); // x is 3
```

These operators form the basis of most JavaScript expressions and control much of its behavior[1] [2] [3] [4] [5] [6] [7].

⁂

1. https://www.w3schools.com/js/js_operators.asp
2. https://www.programiz.com/javascript/operators
3. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_operators
4. https://www.geeksforgeeks.org/javascript-operators/
5. https://www.freecodecamp.org/news/javascript-operators/
6. https://www.simplilearn.com/tutorials/javascript-tutorial/operators-in-javascript
7. https://www.w3schools.com/jsref/jsref_operators.asp