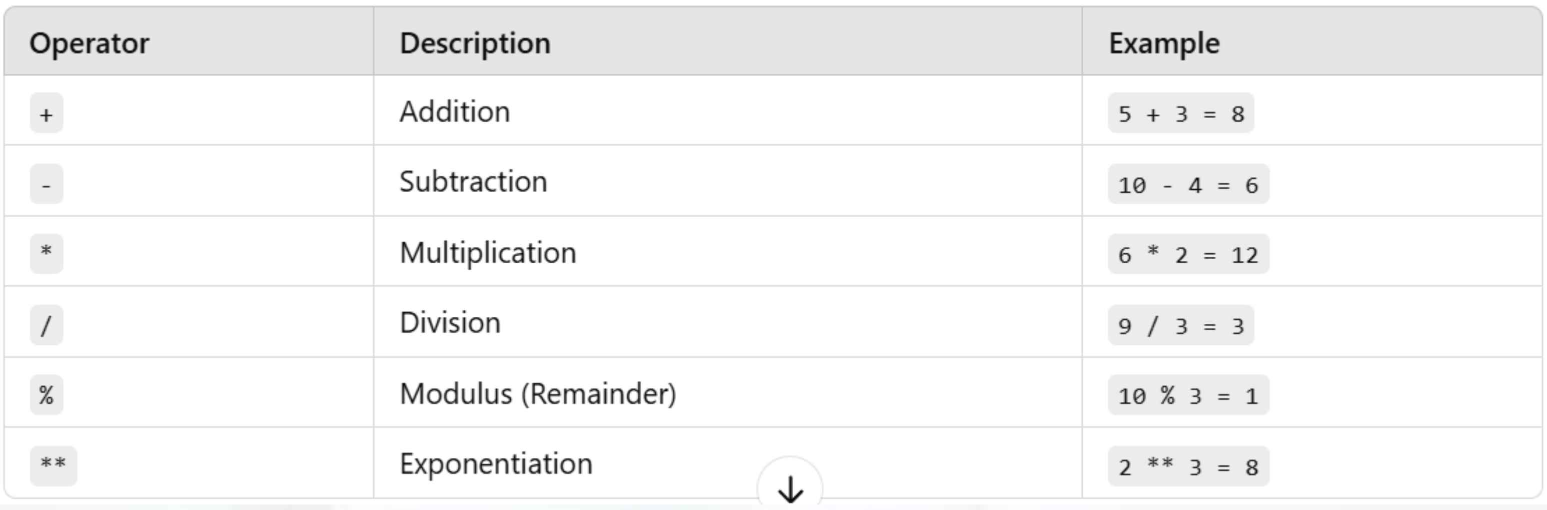
Operators In JS

Operators are symbols that perform operations on values or variables.
There are 8 types of operators in JS.

1️⃣ Arithmetic Operators

Used for mathematical calculations.

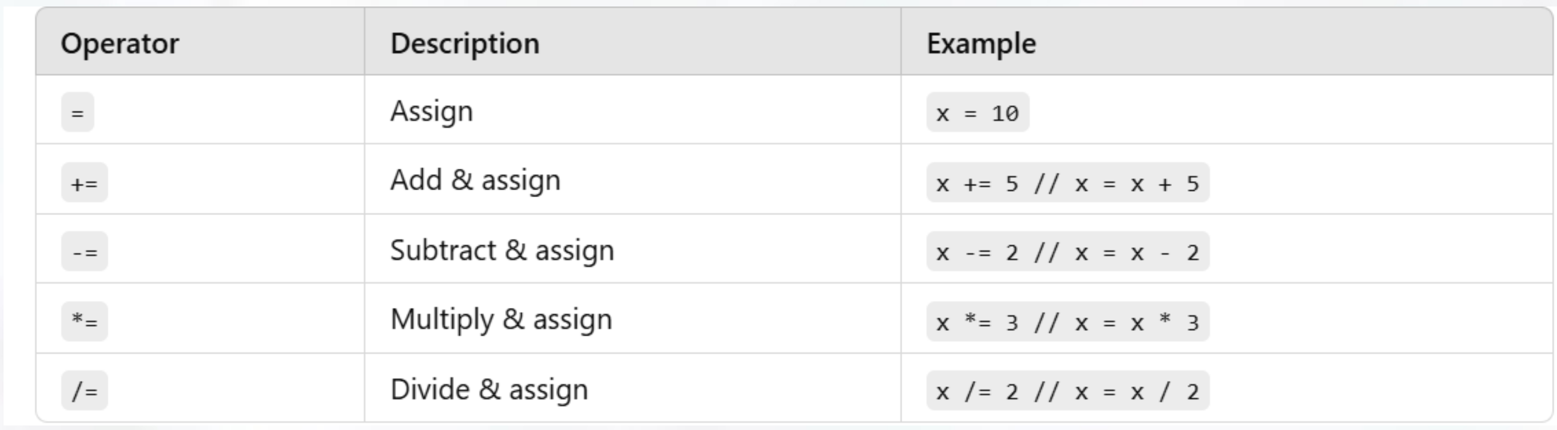


Operator	Description	Example
`+`	Addition	`5 + 3 = 8`
`-`	Subtraction	`10 - 4 = 6`
`*`	Multiplication	`6 * 2 = 12`
`/`	Division	`9 / 3 = 3`
`%`	Modulus (Remainder)	`10 % 3 = 1`
`**`	Exponentiation	`2 ** 3 = 8`

```
let a = 10, b = 3;
console.log(a % b); // Output: 1
```

2 Assignment Operators

Used to assign values to variables.

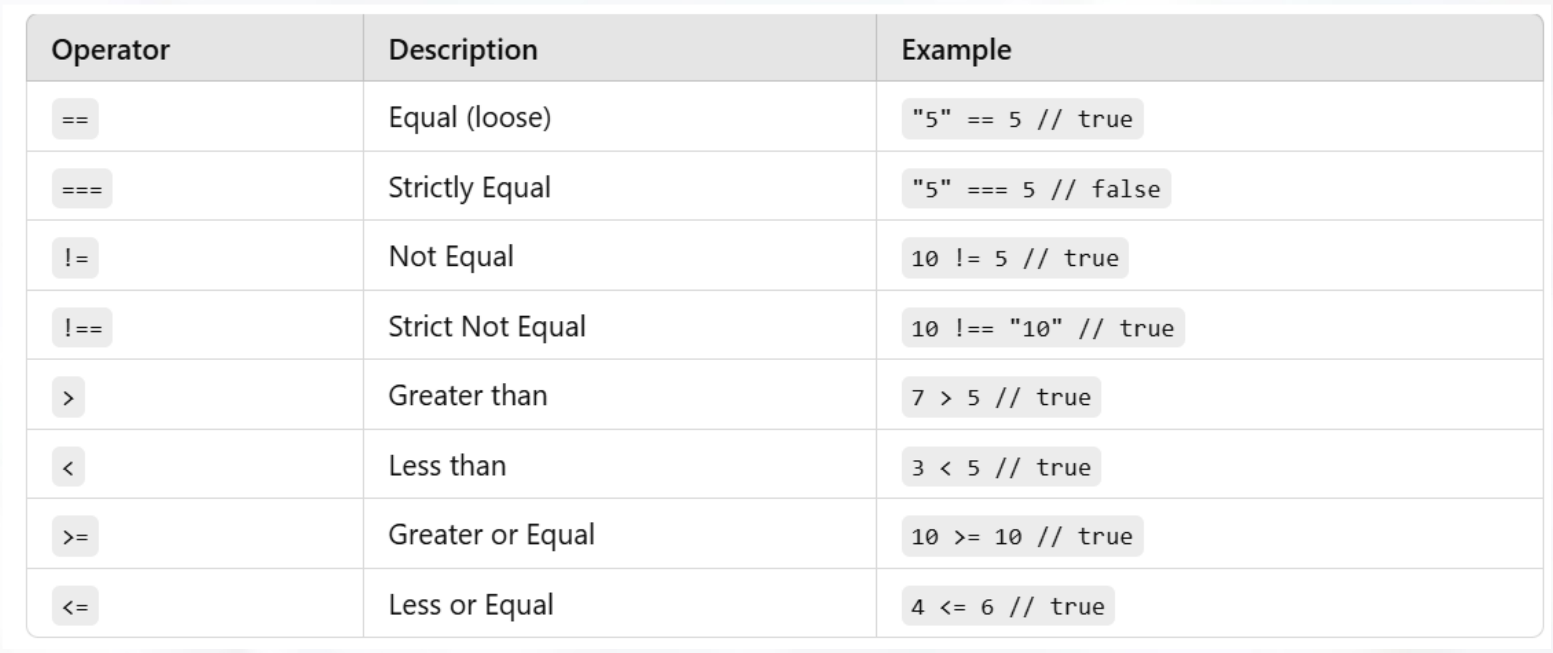


Operator	Description	Example
`=`	Assign	`x = 10`
`+=`	Add & assign	`x += 5 // x = x + 5`
`-=`	Subtract & assign	`x -= 2 // x = x - 2`
`*=`	Multiply & assign	`x *= 3 // x = x * 3`
`/=`	Divide & assign	`x /= 2 // x = x / 2`

```
let x = 5;
x += 3;
console.log(x); // Output: 8
```

3 Comparison Operators

Used to compare values, returning true or false.

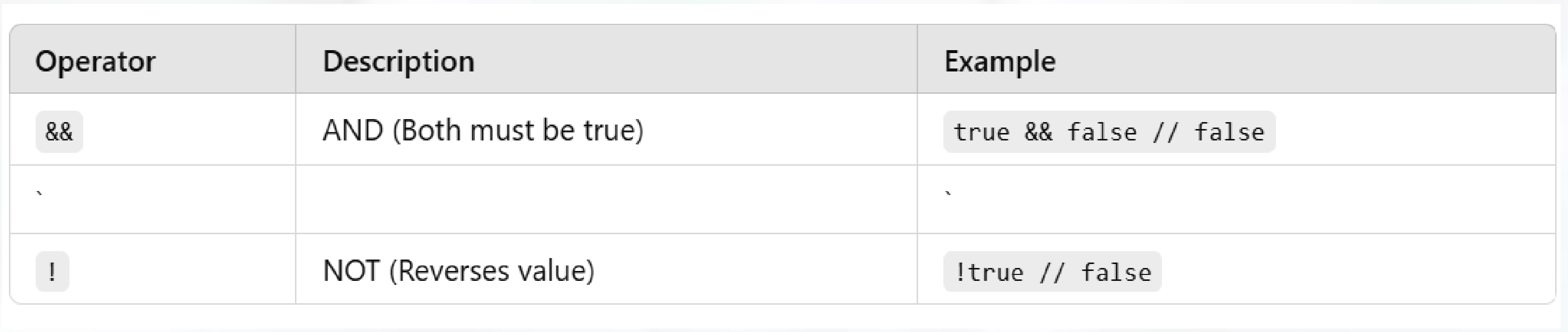


Operator	Description	Example
`==`	Equal (loose)	`"5" == 5 // true`
`===`	Strictly Equal	`"5" === 5 // false`
`!=`	Not Equal	`10 != 5 // true`
`!==`	Strict Not Equal	`10 !== "10" // true`
`>`	Greater than	`7 > 5 // true`
`<`	Less than	`3 < 5 // true`
`>=`	Greater or Equal	`10 >= 10 // true`
`<=`	Less or Equal	`4 <= 6 // true`

```
console.log(5 == "5"); // Output: true (loose comparison)
console.log(5 === "5"); // Output: false (strict comparison)
```

4 Logical Operators

Used to combine multiple conditions.

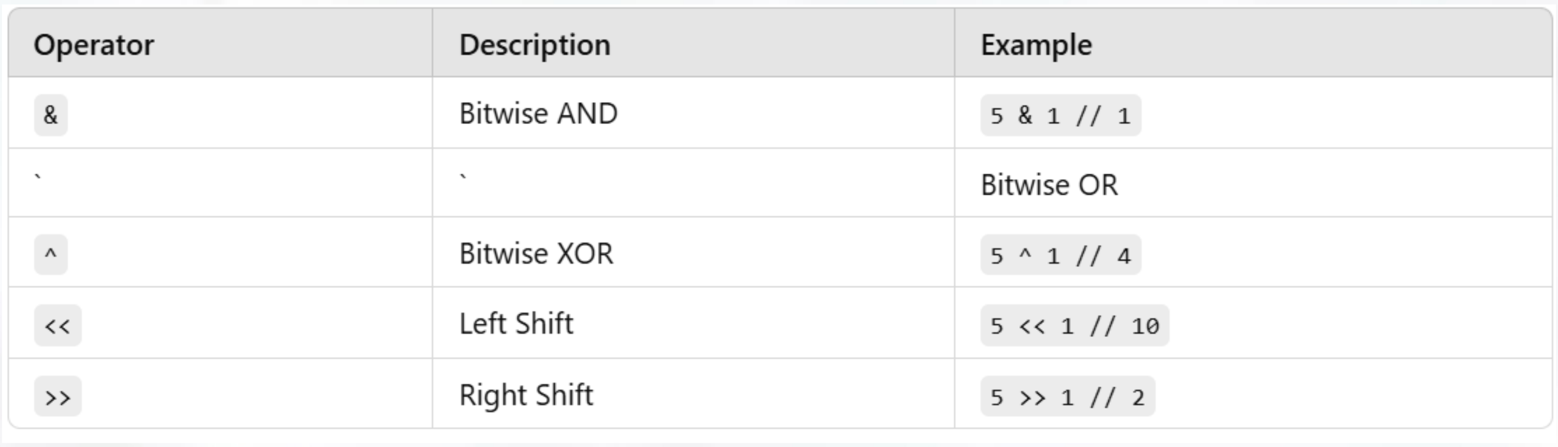


Operator	Description	Example
`&&`	AND (Both must be true)	`true && false // false`
`		`
`!`	NOT (Reverses value)	`!true // false`

```
let age = 20;
console.log(age > 18 && age < 30); // Output: true
```

5 Bitwise Operators

Operate at the binary level (bit-by-bit).



Operator	Description	Example
`&`	Bitwise AND	`5 & 1 // 1`
`	`	Bitwise OR
`^`	Bitwise XOR	`5 ^ 1 // 4`
`<<`	Left Shift	`5 << 1 // 10`
`>>`	Right Shift	`5 >> 1 // 2`

```
console.log(5 & 3); // Output: 1 (Binary: 101 & 011)
```

6 Ternary Operator

A shorthand for if-else conditions.

Syntax

```
condition ? true_value : false_value;
```

```
let score = 85;
let result = (score >= 50) ? "Pass" : "Fail";
console.log(result); // Output: Pass
```

7 String Operators

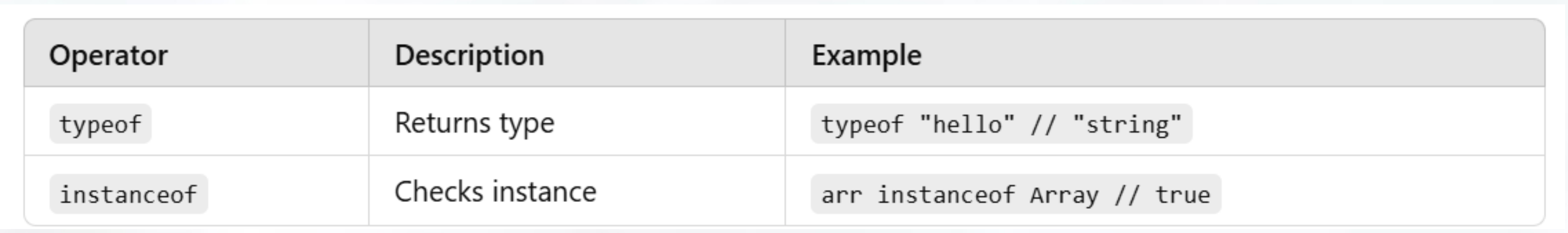
Used to manipulate strings.

Operator	Description	Example
`+`	Concatenation	`"Hello" + " World"`
`+=`	Append	`let s = "Hi"; s += " there!"`

```
let greeting = "Hello" + " World";
console.log(greeting); // Output: Hello World
```

8️⃣ Type Operators

Used to check or convert data types.



Operator	Description	Example
`typeof`	Returns type	`typeof "hello" // "string"`
`instanceof`	Checks instance	`arr instanceof Array // true`

```
console.log(typeof 42); // Output: number
console.log([1, 2, 3] instanceof Array); // Output: true
```