Java script Operators:

JavaScript operators are symbols or keywords used to perform operations on values or variables. There are various types of operators in JavaScript. Here's a detailed overview of the most commonly used ones with examples:

**1. Arithmetic Operators**

These operators are used to perform mathematical operations.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| + | Addition | 5 + 3 | 8 |
| - | Subtraction | 5 – 3 | 2 |
| \* | Multiplication | 5 \* 3 | 15 |
| / | Division | 5 / 2 | 2.5 |
| % | Modulus (remainder) | 5 % 3 | 2 |
| ++ | Increment | let a = 5; a++ | 6 (a becomes 6 after increment) |
| -- | Decrement | let a = 5; a-- | 4 (a becomes 4 after decrement) |

**Example:**

let x = 10;

let y = 3;

console.log(x + y); // 13

console.log(x - y); // 7

console.log(x \* y); // 30

console.log(x / y); // 3.3333333333333335

console.log(x % y); // 1

**x++; // x becomes 11**

**console.log(x); // 11**

**x--;//10 becomes 10**

**console.log(x); // 10**

**2. Assignment Operators**

Used to assign values to variables.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| = | Assigns a value to a variable | let x = 10; | x = 10 |
| += | Adds and assigns | x += 5; | x = x + 5 |
| -= | Subtracts and assigns | x -= 3; | x = x - 3 |
| \*= | Multiplies and assigns | x \*= 2; | x = x \* 2 |
| /= | Divides and assigns | x /= 2; | x = x / 2 |
| %= | Modulus and assigns | x %= 3; | x = x % 3 |

**Example:**

let a = 10;

a += 5; // a becomes 15

a -= 3; // a becomes 12

a \*= 2; // a becomes 24

a /= 6; // a becomes 4

a %= 3; // a becomes 1

**3. Comparison Operators**

Used to compare two values and return a boolean (true or false).

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| == | Equal to | 5 == 5 | true |
| === | Strict equal to (checks both value and type) | 5 === '5' | false |
| != | Not equal to | 5 != 3 | true |
| !== | Strict not equal to (checks both value and type) | 5 !== '5' | true |
| > | Greater than | 5 > 3 | true |
| < | Less than | 5 < 3 | false |
| >= | Greater than or equal to | 5 >= 5 | true |
| <= | Less than or equal to | 5 <= 3 | false |

**Example:**

console.log(5 == '5'); // true (loose equality)

console.log(5 === '5'); // false (strict equality)

console.log(5 != 3); // true

console.log(5 > 3); // true

console.log(5 <= 5); // true

**4. Logical Operators**

Used to combine multiple conditions.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| && | Logical AND (both must be true) | true && false | False |
| || | Logical OR (any one must be true) | True || false | Logical OR (at least one must be true)  True |
| ! | Logical NOT (inverts boolean) | !true | False |

**Example:**

let x = true;

let y = false;

console.log(x && y); // false (both must be true)

console.log(x || y); // true (at least one must be true)

console.log(!x); // false (negates the value)

**5. Conditional (Ternary) Operator**

A shorthand for an if-else statement.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| condition ? expr1 : expr2 | If condition is true, return expr1, otherwise expr2 | x > 10 ? 'Greater' : 'Smaller' | 'Smaller' |

**Example:**

let x = 8;

let result = x > 10 ? 'Greater' : 'Smaller';

console.log(result); // Smaller

**6. Bitwise Operators**

These operators work on the binary representation of numbers.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| & | AND | 5 & 3 | 1 |
| ` | ` | OR | `5 |
| ^ | XOR | 5 ^ 3 | 6 |
| ~ | NOT | ~5 | -6 |
| << | Left shift | 5 << 1 | 10 |
| >> | Right shift | 5 >> 1 | 2 |
| >>> | Unsigned right shift | 5 >>> 1 | 2 |

**Example:**

console.log(5 & 3); // 1 (binary 101 & 011 = 001)

console.log(5 | 3); // 7 (binary 101 | 011 = 111)

console.log(~5); // -6 (inverts the bits of 5)

**7. Type Operators**

Used to check the type of variables.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| typeof | Returns the type of a variable | typeof 10 | number |
| instanceof | Checks if an object is an instance of a specific class | x instanceof Array | true (if x is an Array) |

**Example:**

console.log(typeof 10); // 'number'

console.log(typeof "Hello"); // 'string'

console.log([1, 2, 3] instanceof Array); // true

**8. Spread Operator**

Used to expand or spread elements of an iterable (like arrays or objects).

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| ... | Expands elements or properties | let arr = [1, 2, 3]; let arr2 = [...arr]; | [1, 2, 3] |

**Example:**

let arr = [1, 2, 3];

let newArr = [...arr, 4, 5];

console.log(newArr); // [1, 2, 3, 4, 5]

**9. Destructuring Assignment**

Used to unpack values from arrays or properties from objects into distinct variables.

**Array Destructuring:**

let [a, b] = [10, 20];

console.log(a); // 10

console.log(b); // 20

**Object Destructuring:**

let person = { name: "John", age: 30 };

let { name, age } = person;

console.log(name); // John

console.log(age); // 30

**Summary**

JavaScript has a wide variety of operators that allow you to manipulate data efficiently. These include **arithmetic**, **comparison**, **logical**, **bitwise**, **conditional**, and more specialized operators like the **spread operator** and **destructuring assignment**. Understanding these operators helps you write cleaner, more concise, and more efficient code.

Bottom of Form