

**Question 1: What are the different types of operators in JavaScript? Explain with examples.**

JavaScript has many operators, but the most commonly used are:

### 1) Arithmetic Operators

These operators are used to perform **mathematical operations**.

**Examples:**

Operator	Meaning	Example
+	Addition	$5 + 3 = 8$
-	Subtraction	$10 - 4 = 6$
*	Multiplication	$2 * 4 = 8$
/	Division	$10 / 2 = 5$
%	Modulus (remainder)	$10 \% 3 = 1$
++	Increment	let a = 5; a++; // 6
--	Decrement	let b = 5; b--; // 4

### 2) Assignment Operators

These operators are used to **assign values** to variables.

**Examples:**

Operator	Meaning	Example
=	Assign value	let x = 10;
+=	Add and assign	x += 5; // x = x + 5
-=	Subtract and assign	x -= 2;

Operator	Meaning	Example
<code>*=</code>	Multiply and assign	<code>x *= 3;</code>
<code>/=</code>	Divide and assign	<code>x /= 2;</code>
<code>%=</code>	Mod remainder and assign	<code>x %= 3;</code>

### 3) Comparison Operators

These operators are used to **compare two values**.  
Output is always **true** or **false**.

**Examples:**

Operator	Meaning	Example
<code>==</code>	Equal	<code>5 == "5" → true</code>
<code>===</code>	Strict equal	<code>5 === "5" → false</code>
<code>!=</code>	Not equal	<code>4 != 3 → true</code>
<code>&gt;</code>	Greater than	<code>10 &gt; 8 → true</code>
<code>&lt;</code>	Less than	<code>3 &lt; 5 → true</code>
<code>&gt;=</code>	Greater or equal	<code>7 &gt;= 7 → true</code>
<code>&lt;=</code>	Less or equal	<code>5 &lt;= 10 → true</code>

### 4) Logical Operators

These operators are used to combine conditions.

**Examples:**

Operator	Meaning	Example
<code>&amp;&amp;</code>	Logical AND	<code>(5 &gt; 3 &amp;&amp; 10 &gt; 5) → true</code>

Operator	Meaning	Example
----------	---------	---------

	Logical OR	(5 > 8    10 > 5) → true
--	------------	--------------------------

!	Logical NOT	!(5 > 3) → false
---	-------------	------------------

## Question 2: What is the difference between == and === in JavaScript?

### == (Double Equal) → Loose Comparison

- Only checks **value**, not data type.
- Converts both values to the same type before comparing.

#### Example:

```
5 == "5" // true (same value)
```

### === (Triple Equal) → Strict Comparison

- Checks **value AND data type** both.
- No type conversion happens.

#### Example:

```
5 === "5" // false (number ≠ string)
```