Q1. What are the different types of operators in JavaScript? Explain with examples.

Ans: Operators in JavaScript are used to perform operations on variables and values. JavaScript has several types of operators. Below are the main operator types with examples.

1. Arithmetic Operators

Used to perform mathematical calculations.

Operator	**Description**	**Example**
+	Addition	5 + 2 → 7
-	Subtraction	5 - 2 → 3
*	Multiplication	5 * 2 → 10
/	Division	10 / 2 → 5
%	Modulus (remainder)	5 % 2 → 1
++	Increment	x++
--	Decrement	x--

Example:

let a = 10;

let b = 3;

console.log(a + b);

2. Assignment Operators

Used to assign values to variables.

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

Example:

```
let x = 10;

x += 5;   // x = 15
```

3. Comparison Operators

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

Operator	**Description**	**Example**
==	Equal to	5 == "5" → true
===	Strict equal	5 === "5" → false
!=	Not equal	5 != 3 → true
!==	Strict not equal	5 !== "5" → true
>	Greater than	10 > 5 → true
<	Less than	3 < 5 → true
>=	Greater than or equal to	5 >= 5 → true
<=	Less than or equal to	4 <= 6 → true

Example:

```
let a = 5;

let b = 10;

console.log(a < b);    // true
```

4. Logical Operators

Used to combine multiple conditions.

Operator	**Description**	**Example**
&&	Logical AND	true && false → false
`		`
!	Logical NOT	!true → false

Example:

```
let age = 20;
let hasID = true;

if (age >= 18 && hasID) {
    console.log("Allowed");
}
```

Q2. What is the difference between == and === in JavaScript?

Ans: In JavaScript, == and === are comparison operators, but they behave differently.

== (Loose Equality Operator)

- Compares values only.
- Performs type conversion (type coercion) before comparison.
- Different data types can be considered equal.

Example:

```
5 == "5"   // true
true == 1 // true
null == undefined // true
```

=== (Strict Equality Operator)

- Compares both value and data type.
- Does not perform type conversion.
- Values must be exactly the same.

Example:

```
5 === "5"   // false
true === 1 // false
null === undefined // false
```