Operators in JavaScript

JavaScript Operators

Operators are symbols or keywords that perform operations on values or variables. They can be classified into several categories based on their functionality.

1. **Arithmetic Operators**
2. **Assignment Operators**
3. **Comparison Operators**
4. **Logical Operators**
5. **Bitwise Operators**
6. **String Operators**
7. **Unary Operators**
8. **Ternary Operator**
9. **Type Operators**

1.Arithmetic Operators.

Used for mathematical calculations

Operator	Description	Example	Output
+	Addition	5 + 2	7
-	Subtraction	5 - 2	3
*	Multiplication	5 * 2	10
/	Division	5 / 2	2.5
%	Modulus (Remainder)	5 % 2	1
**	Exponentiation	5 ** 2	25

2. Assignment Operators

Used to assign values to variables.

Operator	Example	Equivalent To
=	x = 10	Assign 10 to x
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
**=	x **= 2	x = x ** 2

3.Comparison Operators

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

Operator	Description	Example	Output
==	Equal to	5 == '5'	true
===	Strict equal to	5 === '5'	false
!=	Not equal to	5 != '5'	false
!==	Strict not equal to	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 <= 4	false

4. Logical Operators

Used to combine multiple conditions.

Operator	Description	Example	Output
&&	Logical AND	true && false	false
||	Logical OR	true II false	true
!	Logical NOT	!true	false

5. Bitwise Operators

Operate at the bit level.

Operator	Description	Example	Output
&	AND	5 & 1	1
`	`	OR	`5
^	XOR	5 ^ 1	4
~	NOT (Complement)	~5	-6
<<	Left Shift	5 << 1	10
>>	Right Shift	5 >> 1	2
>>>	Zero-fill Right Shift	5 >>> 1	2

6. String Operators

Used for string concatenation.

Operator	Description	Example	Output
+	Concatenation	'Hello' + ' World'	'Hello World'
+=	Concatenation & Assign	let x = 'Hi'; x += '!'	'Hi!'

7. Unary Operators

Operate on a single operand.

Operator	Description	Example	Output
+	Positive	+5	5
-	Negative	-5	-5
++	Increment (Pre/Post)	++x, x++	6, 5
--	Decrement (Pre/Post)	--x, x--	4, 5

8. Ternary Operator

A shorthand for `if-else` statements.

Syntax:

```
condition ? valueIfTrue :
valueIfFalse;
```

Example:

```
let age = 18;

let message = (age >= 18) ? "Adult"
: "Minor";

console.log(message); // Output:
Adult
```

9. Type Operators

Used to check or convert types.

Operator	Description	Example	Output
typeof	Returns the type of a value	typeof "Hello"	"string"
instanceof	Checks instance of an object	[] instanceof Array	true

JavaScript Comments

JavaScript comments are used to add notes, explanations, or temporarily disable parts of the code. They are ignored by the JavaScript engine during execution.

There are two types of comments in JavaScript: **single-line comments** and **multi-line comments**.

1. Single-Line Comments

Start with `//`.

Used for short notes or to disable a single line of code.

Example:

```
// This is a single-line comment
let name = "Alice"; // Variable declaration
console.log(name); // Output: Alice
```

2. Multi-Line Comments

Enclosed between `/*` and `*/`.

Used for longer explanations or to comment out multiple lines of code.

Example:

```
/*
   This is a multi-line comment.
   You can use it to explain complex code or disable multiple lines.
*/
let age = 25;
console.log(age); // Output: 25
```