

JavaScript Operators

What are Operators?

Operators are symbols used to perform **mathematical, logical, and comparison operations** on values (operands).

Example:

```
let z = x + y;
```

Here:

- x and y → operands
 - + → operator
-

Types of JavaScript Operators

JavaScript has many types of operators:

1. Arithmetic Operators
 2. Assignment Operators
 3. Comparison Operators
 4. Logical Operators
 5. String Operators
-

1 Arithmetic Operators

Arithmetic operators are used to perform **math operations**.

Operator Description

+	Addition
-	Subtraction
*	Multiplication
/	Division

Operator Description

% Modulus (remainder)

** Exponentiation

++ Increment

-- Decrement

Addition (+)

```
let x = 5;
```

```
let y = 2;
```

```
let z = x + y; // 7
```

Subtraction (-)

```
let z = 5 - 2; // 3
```

Multiplication (*)

```
let z = 5 * 2; // 10
```

Division (/)

```
let z = 5 / 2; // 2.5
```

Modulus (%)

```
let z = 5 % 2; // 1
```

Increment (++)

```
let x = 5;
```

```
x++;
```

```
// x = 6
```

Decrement (--)

```
let x = 5;
```

```
x--;
```

```
// x = 4
```

Exponentiation (**)

```
let z = 5 ** 2; // 25
```

Operator Precedence

Operators follow **math rules**.

```
let x = 100 + 50 * 3; // 250
```

Multiplication happens first.

Using brackets:

```
let x = (100 + 50) * 3; // 450
```

2 Assignment Operators

Assignment operators **assign values** to variables.

Operator Example Same As

=	x = y	x = y
---	-------	-------

+=	x += y	x = x + y
----	--------	-----------

-=	x -= y	x = x - y
----	--------	-----------

*=	x *= y	x = x * y
----	--------	-----------

/=	x /= y	x = x / y
----	--------	-----------

%=	x %= y	x = x % y
----	--------	-----------

**=	x **= y	x = x ** y
-----	---------	------------

Example

```
let x = 10;
```

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

3 String Operators

String Concatenation (+)

```
let first = "John";  
let last = "Doe";  
let full = first + " " + last;  
  
// John Doe
```

Using += with Strings

```
let text = "Hello";  
text += " World";  
  
// Hello World
```

String + Number

```
let x = 5 + 5;    // 10  
let y = "5" + 5;  // "55"  
let z = "Hello" + 5; // "Hello5"
```

Rule:

If string + number → result is **string**

Comparison Operators

Comparison operators compare values and **return true or false**.

Operator Description

==	Equal value
===	Equal value & type
!=	Not equal
!==	Not equal value or type
>	Greater than
<	Less than

Operator Description

`>=` Greater or equal

`<=` Less or equal

Example

```
let x = 5;
```

```
x > 8; // false
```

```
x == "5"; // true
```

```
x === "5"; // false
```

Used in Conditions

```
if (age < 18) {  
    text = "Too young";  
}
```

String Comparison

```
"A" < "B"; // true
```

```
"2" > "12"; // true
```

Strings are compared **alphabetically**, not numerically.

Comparing Different Data Types

```
2 < "12"; // true
```

```
2 < "John"; // false
```

```
"2" < "12"; // false
```

Always convert to proper type:

```
age = Number(age);
```

5 Logical Operators

Logical operators are used with **conditions**.

Operator Description

`&&` AND

`||` OR

`!` NOT

Example

```
let age = 20;
```

```
let hasID = true;
```

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

Logical Assignment Operators (ES2020)

AND Assignment (`&&=`)

```
let x = true;
```

```
x &&= 10; // x = 10
```

OR Assignment (`||=`)

```
let x = false;
```

```
x ||= 10; // x = 10
```

Nullish Assignment (`??=`)

```
let x;
```

```
x ??= 10; // x = 10
```
