

JS-OPERATORS



ANECO ACADEMY

JavaScript Operators

An operator performs some operation on single or multiple operands and produces a result

<Left operand> operator <right operand>

For example, in 1 + 2,
+ sign is an operator and
1 is left side operand and
2 is right side operand

Types of Operators

1. Arithmetic Operators
2. Comparison Operators
3. Logical Operators
4. Assignment Operators
5. Conditional Operators or ternary operator

JavaScript Operators - Arithmetic

Operator	Description
+	Adds two numeric operands.
-	Subtract right operand from left operand
*	Multiply two numeric operands.
/	Divide left operand by right operand.
%	Modulus operator. Returns remainder of two operands.
++	Increment operator. Increase operand value by one.
--	Decrement operator. Decrease value by one.

Example: Arithmetic Operation

```
var x = 5, y = 10;
```

```
var z = x + y; //performs addition and returns 15
```

```
z = y - x; //performs subtraction and returns 5
```

```
z = x * y; //performs multiplication and returns 50
```

```
z = y / x; //performs division and returns 2
```

```
z = x % 2; //returns division remainder 1
```

```
var x = 5;
```

```
x++; //post-increment, x will be 5 here and 6 in the next line
```

```
++x; //pre-increment, x will be 7 here
```

```
x--; //post-decrement, x will be 7 here and 6 in the next line
```

```
--x; //pre-decrement, x will be 5 here
```

JavaScript Operators - Comparison

Operators	Description
==	Compares the equality of two operands without considering type .
===	Compares equality of two operands with type .
!=	Compares inequality of two operands.
>	Returns a boolean value true if the left-side value is greater than the right-side value; otherwise, returns false.
<	Returns a boolean value true if the left-side value is less than the right-side value; otherwise, returns false.
>=	Returns a boolean value true if the left-side value is greater than or equal to the right-side value; otherwise, returns false.
<=	Returns a boolean value true if the left-side value is less than or equal to the right-side value; otherwise, returns false.

Example: Comparison Operators

```
var a = 5, b = 10, c = "5";  
var x = a;
```

```
a == c; // returns true
```

```
a === c; // returns false
```

```
a == x; // returns true
```

```
a != b; // returns true
```

```
a > b; // returns false
```

```
a < b; // returns true
```

```
a >= b; // returns false
```

```
a <= b; // returns true
```

JavaScript Operators - Logical

Operator	Description
&&	&& is known as AND operator. It checks whether two operands are non-zero or not (0, false, undefined, null or "" are considered as zero). It returns 1 if they are non-zero; otherwise, returns 0.
	is known as OR operator. It checks whether any one of the two operands is non-zero or not (0, false, undefined, null or "" is considered as zero). It returns 1 if any one of of them is non-zero; otherwise, returns 0.
!	! is known as NOT operator. It reverses the boolean result of the operand (or condition). !false returns true, and !true returns false.

Example: Logical Operators

```
var a = 5, b = 10;
```

```
(a != b) && (a < b); // returns true
```

```
(a > b) || (a == b); // returns false
```

```
(a < b) || (a == b); // returns true
```

```
!(a < b); // returns false
```

```
!(a > b); // returns true
```

JavaScript Operators – Ternary or condition operator

- JavaScript provides a special operator called ternary operator :? that assigns a value to a variable based on some condition.
- Ternary operator ?: is a short form of if-else condition

Syntax:

```
<condition> ? <value1> : <value2>;
```

Example:

```
var a = 10, b = 5;
```

```
var c = a > b? a : b; // value of c would be 10
```

```
var d = a > b? b : a; // value of d would be 5
```