

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands. For example:

```
var sum=10+20;
```

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	$10+20 = 30$
-	Subtraction	$20-10 = 10$
*	Multiplication	$10*20 = 200$
/	Division	$20/10 = 2$
%	Modulus (Remainder)	$20\%10 = 0$
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

JavaScript Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
<code>==</code>	Is equal to	<code>10==20 = false</code>
<code>===</code>	Identical (equal and of same type)	<code>10===20 = false</code>
<code>!=</code>	Not equal to	<code>10!=20 = true</code>
<code>!==</code>	Not Identical	<code>20!==20 = false</code>
<code>></code>	Greater than	<code>20>10 = true</code>
<code>>=</code>	Greater than or equal to	<code>20>=10 = true</code>
<code><</code>	Less than	<code>20<10 = false</code>
<code><=</code>	Less than or equal to	<code>20<=10 = false</code>

JavaScript Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
&	Bitwise AND	$(10 == 20 \ \& \ 20 == 33) = \text{false}$
 	Bitwise OR	$(10 == 20 \ \ 20 == 33) = \text{false}$
^	Bitwise XOR	$(10 == 20 \ ^ \ 20 == 33) = \text{false}$
~	Bitwise NOT	$(\sim 10) = -10$
<<	Bitwise Left Shift	$(10 << 2) = 40$
>>	Bitwise Right Shift	$(10 >> 2) = 2$
>>>	Bitwise Right Shift with Zero	$(10 >>> 2) = 2$

JavaScript Logical Operators

The following operators are known as JavaScript logical operators.

Operator	Description	Example
&&	Logical AND	<code>(10==20 && 20==33)</code> = false
 	Logical OR	<code>(10==20 20==33) =</code> false
!	Logical Not	<code>!(10==20) = true</code>

JavaScript Assignment Operators

The following operators are known as JavaScript assignment operators.

Operator	Description	Example
=	Assign	10+10 = 20
+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
=	Multiply and assign	var a=10; a=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

JavaScript Special Operators

The following operators are known as JavaScript special operators.

Operator	Description
(?:)	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.