

Operators in Java Script

- Arithmetic Operators
- Assignment Operators
- Comparison (Relational) Operators
- Logical Operators
- Conditional Operators
- Bitwise Operators
- Special Operators

- **Arithmetic Operators**

Arithmetic operators are used to perform arithmetic between variables and/or values.

Given that y=5, the table below explains the arithmetic operators:

Operator	Description	Example	Result
+	Addition	x=y+2	x=7
-	Subtraction	x=y-2	x=3
*	Multiplication	x=y*2	x=10
/	Division	x=y/2	x=2.5
%	Modulus (division remainder)	x=y%2	x=1
++	Increment	x=++y	x=6
--	Decrement	x=--y	x=4

The + operator can also be used to add string variables or text values together.

To add two or more string variables together, use the + operator.

```
txt1="What a very";  
txt2="nice day";  
txt3=txt1+txt2;
```

- **Assignment Operators**

Assignment operators are used to assign values to JavaScript variables.

Given that x=10 and y=5, the table below explains the assignment operators:

Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
=	x=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

- **Comparison (Relational) Operators**

Comparison operators are used in logical statements to determine equality or difference between variables or values.

Given that x=5, the table below explains the comparison operators:

Operator	Description	Example
==	is equal to	x==8 is false
===	is exactly equal to (value and type)	x===5 is true x==="5" is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

How to Use

Comparison operators can be used in conditional statements to compare values and take action depending on the result:

```
if (age<18) console.log ("Too young");
```

- **Logical Operators**

Logical operators are used to determine the logic between variables or values.

Given that x=6 and y=3, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
	or	(x==5 y==5) is false
!	not	!(x==y) is true

Statement - 1	Logical Operator	Statement - 2	Result
true	&&	true	true
false	&&	true	false
true	&&	false	false
false	&&	false	false

Statement - 1	Logical Operator	Statement - 2	Result
true		true	true
false		true	true
true		false	true
false		false	false

○ **Conditional Operators**

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

The conditional (ternary) operator is the only JavaScript operator that takes three operands:

1. a condition followed by a question mark (?),
2. an expression to execute if the condition is truthy followed by a colon (:),
3. finally the expression to execute if the condition is falsy.

Syntax

variablename = (condition)? Value1 : value2

Example

```
students= (exam=="PASS"? "Welcome " : "Try Again ";
```

If the variable *exam* has the value of "PASS", then the variable *students* will be assigned the value "Welcome" else it will be assigned "Try Again".

○ Bitwise Operators

Bitwise operators perform bit-by-bit logical operations.

The bitwise logical operators treat the operands and result as a series of 32 bits (like an integer).

Operator	Meaning
<<	Shifts the bits to left
>>	Shifts the bits to right
~	Bitwise inversion (one's complement)
&	Bitwise logical AND
	Bitwise logical OR
^	Bitwise exclusive or

○ Special Operators

Name	Operator	Description
Property access	.	Appends an object, method, or property to another object
Array index	[]	Accesses an element of an array
Function call	()	Calls up functions or changes the order in which individual operations in an expression are evaluated
Comma	,	Allows you to include multiple expressions in the same statement
Conditional expression	? :	Executes one of two expressions based on the results of a conditional expression
Delete	delete	Deletes array elements, variables created without the var keyword, and properties of custom objects
Property exists	in	Returns a value of true if a specified property is contained within an object
Object type	instanceof	Returns true if an object is of a specified object type
New object	new	Creates a new instance of a user-defined object type or a predefined JavaScript object type
Data type	typeof	Determines the data type of a variable
Void	void	Evaluates an expression without returning a result

