Web Development Programming Language

By: Dr. Galal AL-Marzoqi

Operators in JavaScript

➢ Type of Operators in JavaScript

JavaScript supports the following types of operators:

- Arithmetic Operators
- Assignment Operators
- Comparison operators
- Logical Operators

• Arithmetic Operators

In JavaScript, arithmetic operators take numerical values (literals or variables) as their operands and return a single numerical value. There are four standard arithmetic operators, addition (+), subtraction (-), multiplication (*), and division (/). See table below. Multiplication

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation
/	Division
%	Modulus
++	Increment
--	Decrement

✓ Example in Addition

```
<html>
 <body>
   <script>
     var x = 100 + 50;
     document.write("The result will be: ", x);
   </script>
 </body>
</html>
```

Result: 150

```
<html>
  <body>
    <script>
      var a = 100;    var b = 50;
      var x = a + b;
     document.write("The result will be: ", x);
    </script>
   </body>
</html>
```

Result: 150

✓ Example in Subtraction

```
<html>
 <body>
   <script>
     var x = 100 - 50;
     document.write("The result will be: ", x);
   </script>
 </body>
</html>
```

Result: 50

```
<html>
  <body>
    <script>
      var a = 100;    var b = 50;
      var x = a - b;
     document.write("The result will be: ", x);
    </script>
   </body>
</html>
```

Result: 50

✓ Example in Multiplication

```
<html>
 <body>
   <script>
     var x = (100 * 50);
     document.write("The result will be: ", x);
   </script>
 </body>
</html>
```

Result: 5000

```
<html>
  <body>
    <script>
      var a = 100;    var b = 50;
      var x = (a * b);
     document.write("The result will be: ", x);
    </script>
   </body>
</html>
```

Result: 5000

✓ Example in Division

```
<html>
 <body>
   <script>
     var x = (100 / 50);
     document.write("The result will be: ", x);
   </script>
 </body>
</html>
```

Result: 2

```
<html>
  <body>
    <script>
      var a = 100;    var b = 50;
      var x =  (a / b);
     document.write("The result will be: ", x);
    </script>
   </body>
</html>
```

Result: 2

✓ **Example in Addition (+) with Multiplication (*)**

```
<html>
 <body>
   <script>
        var a = 3;
        var x = (100 + 50) * a;
    document.write("The result will be: ", x);
   </script>
 </body>
</html>
```

Result: 450
X= (100+50) = 150 * 3 = 450

```
<html>
  <body>
    <script>
      var a = 100;    var b = 50;  var c =3;
      var x = (a + b) * c;
     document.write("The result will be: ", x);
    </script>
   </body>
</html>
```

Result: 450
X= (100+50) = 150 * 3 = 450

- **Assignment Operators**

Assignment operators assign values to JavaScript 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
>>=	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 in = Operator**

```
<html>
  <body>
<h2>The = Operator</h2>
   <script>
     var x = 10;
     document.write("Result:", x);
   </script>
 </body>
</html>
```

Result: 10

✓ **Example in + = Operator**

```
<html>
  <body>
    <h2>The += Operator</h2>
    <script>
      var x = 10;
        x += 5;
      document.write("Result:", x);
    </script>
  </body>
</html>
```

Result: X+= 5 -> X=X+5 -> X=10+5=15

✓ **Example in - = Operator**

```
<html>
 <body>
<h2>The = Operator</h2>
  <script>
      var x = 10;
        x - = 5;
      document.write("Result:", x);
   </script>
 </body>
</html>
```

Result: X- = 5 -> X=X-5 -> X=10-5=5

✓ **Example in * = Operator**

```
<html>
 <body>
   <h2>The += Operator</h2>
   <script>
     var x = 10;
       x *= 5;
     document.write("Result:", x);
   </script>
 </body>
</html>
```

Result: X*= 5 -> X=X*5 -> X=10*5=50

✓ **Example in / = Operator**

```
<html>
 <body>
<h2>The = Operator</h2>
  <script>
      var x = 10;
        x / = 5;
      document.write("Result:", x);
   </script>
 </body>
</html>
```

Result: X/ = 5 -> X=X/5 -> X=10 / 5= 2

✓ **Example in % = Operator**

```
<html>
 <body>
   <h2>The += Operator</h2>
   <script>
     var x = 10;
       x %= 5;
     document.write("Result:", x);
   </script>
 </body>
</html>
```

Result: X%= 5 -> X=X%5 -> X=10%5=0

- **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	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
!=	not equal	x != 8	true
>	greater than	x > 8	false
<	less than	x < 8	true
>=	greater than or equal to	x >= 8	false
<=	less than or equal to	x <= 8	true

✓ **Example in comparison operators**

```
<html>
 <body>
  <h2>JavaScript Comparison</h2>
  <script>
    var x = 5;
    document.write("Comparison value:", x = = 8);
  </script>
 </body>
</html>
```

Result: false

```
<html>
 <body>
  <h2>JavaScript Comparison</h2>
  <script>
    var x = 8;
    document.write("Comparison value:", x = = 8);
  </script>
 </body>
</html>
```

Result: true

- **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

✓ **Example in Logical operators**

```
<html>
  <body>
    <h2>JavaScript Comparison</h2>
    <script>
      var x = 6;       var y = 3;
      document.write( (x < 10 && y > 1) );
    </script>
  </body>
</html>
```

Result: true

```
<html>
  <body>
    <h2>JavaScript Comparison</h2>
    <script>
      var x = 6;       var y = 3;
      document.write( (x < 10 && y < 1) );
    </script>
  </body>
</html>
```

Result: false

✓ **Example in Logical operators**

```
<html>
 <body>
  <h2>JavaScript Comparison</h2>
  <script>
   var x = 6;      var y = 3;
  document.write( (x = = 5 || y = = 5) + "<br>" +  (x = = 6 || y = = 0));
  </script>
  </body>  </html>
```

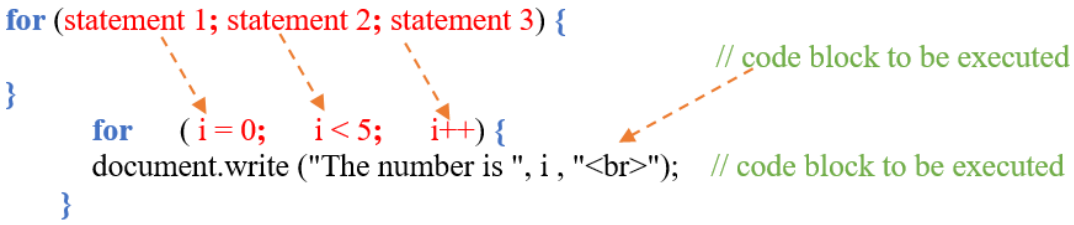
Result: false

Result: true

Iteration and conditional commands

➢ For loop

The for loop has the following syntax:



Statement 1 is executed (one time) before the execution of the code block.
Statement 2 defines the condition for executing the code block.
Statement 3 is executed (every time) after the code block has been executed.

✓ Example	✓ Result

```
<html>
 <body>
  <h2>JavaScript For Loop</h2>
  <script>
      var i;
      for ( i = 0; i < 5; i + +) {
      document.write ("The number is ", i, "<br>");
      }
  </script>
 </body>
</html>
```

```
JavaScript For Loop

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
```

✓ Example	✓ Result

```
<html>
<body>
<h2>JavaScript For Loop</h2>
<script>
var names = ["BMW", "Volvo", "Saab", "Ford"];
var i, len;
for (i = 0, len = names.length; i < len; i++) {
 document.write(names[i] + "<br>");
}
</script>
</body>
</html>
```

```
JavaScript For Loop

BMW
Volvo
Saab
Ford
```

From the example above, you can read:

Statement 1 sets a variable before the loop starts (var i = 0).

Statement 2 defines the condition for the loop to run (i must be less than 5).

Statement 3 increases a value (i++) each time the code block in the loop has been executed.

- **The While Loop**

The while loop loops through a block of code as long as a specified condition is true.

Syntax

```
while (condition) {
  // code block to be executed
}
```

✓ Example	✓ Result

<html>
<body>
 JavaScript While Loop
<script>
var i = 0;
while (i < 5) {
document.write ("The number is ", i, "
")
i++;
}
</script>
</body>
</html>
	JavaScript While Loop
	
	The number is 0
	The number is 1
	The number is 2
	The number is 3
	The number is 4