University of Mosul

College of Engineering

Computer Engineering Dept.

Introduction to JavaScript

By

Dr. Mayada Faris Ghanim

JavaScript

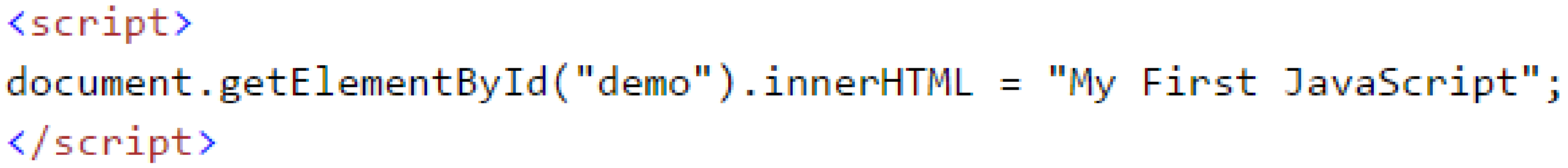
JavaScript is a scripting language used to create and control dynamic website content, i.e. anything that moves, refreshes, or otherwise changes on your screen without requiring you to manually reload a web page. Features like:

- animated graphics
- photo slideshows
- autocomplete text suggestions
- interactive forms

JavaScript Where To

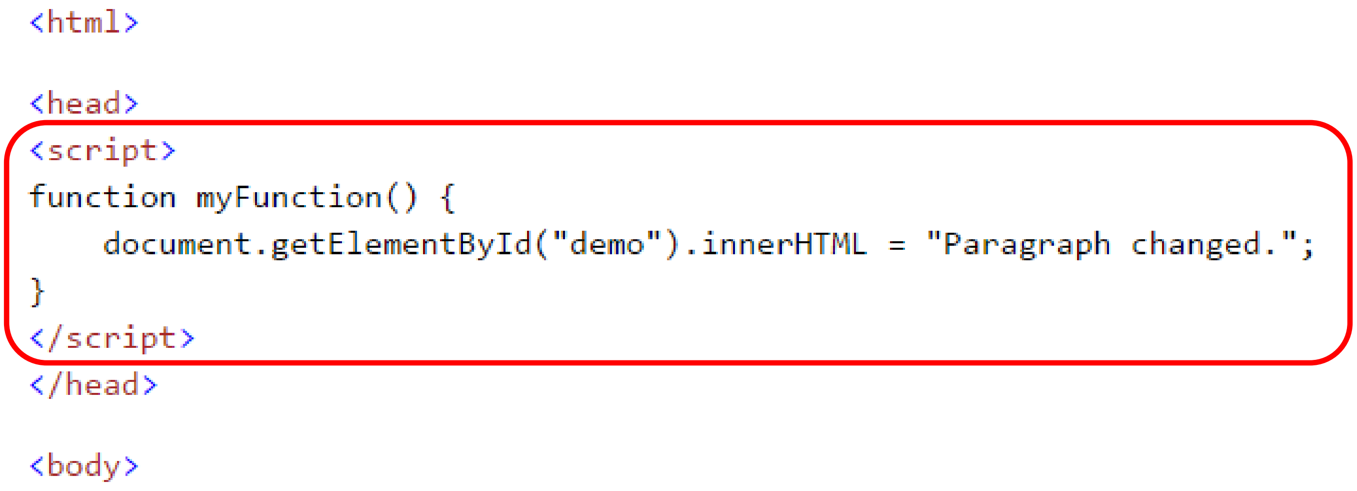
- The <script> Tag

In HTML, JavaScript code must be inserted between <script> and </script> tags.



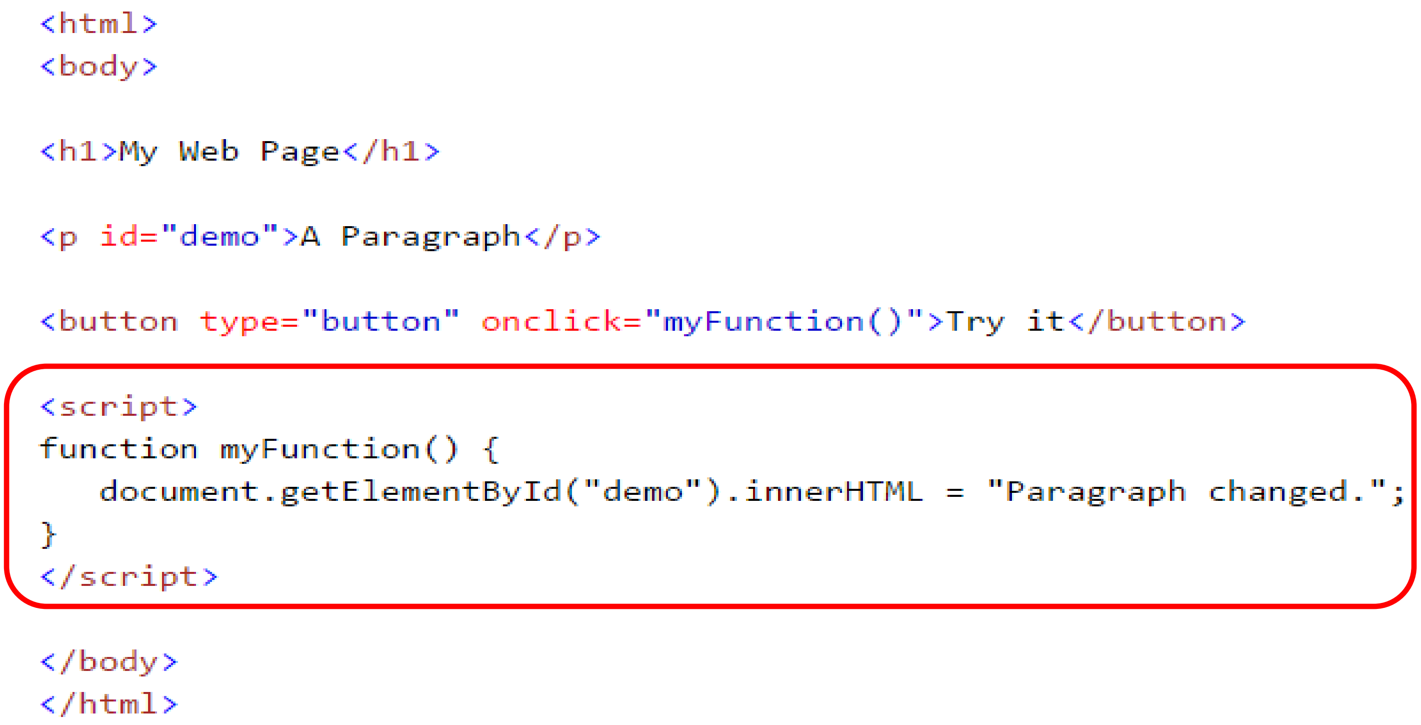
```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

- JavaScript in <head>



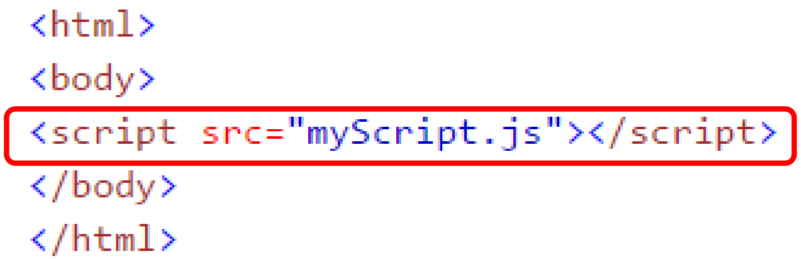
```
<html>

<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>

<body>
```

- JavaScript in <body>



```
<html>
<body>

<h1>My Web Page</h1>

<p id="demo">A Paragraph</p>

<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction() {
   document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>

</body>
</html>
```

- External JavaScript

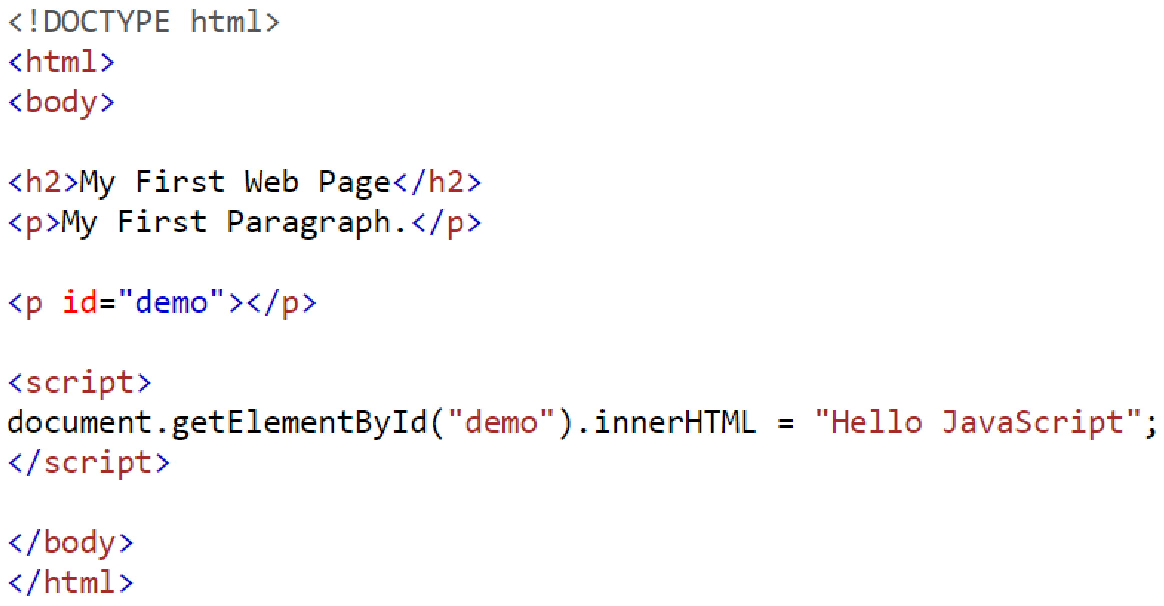


```
<html>
<body>
<script src="myScript.js"></script>
</body>
</html>
```

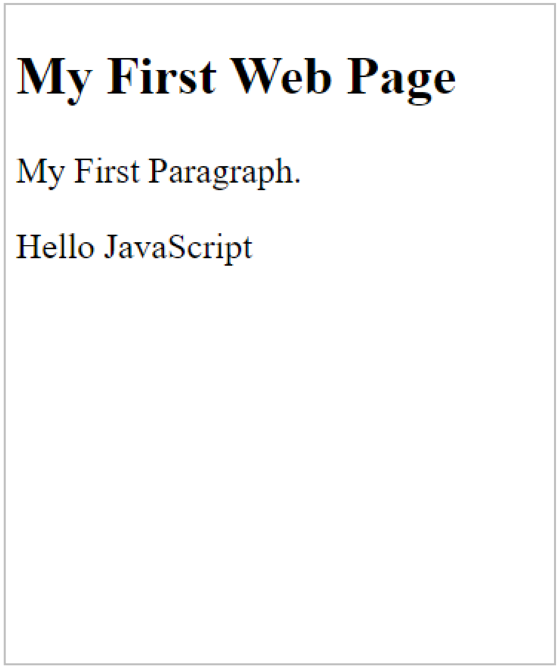
JavaScript Output

1. Writing into an HTML element, using innerHTML.

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```



```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My First Paragraph.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript";
</script>

</body>
</html>
```



My First Web Page

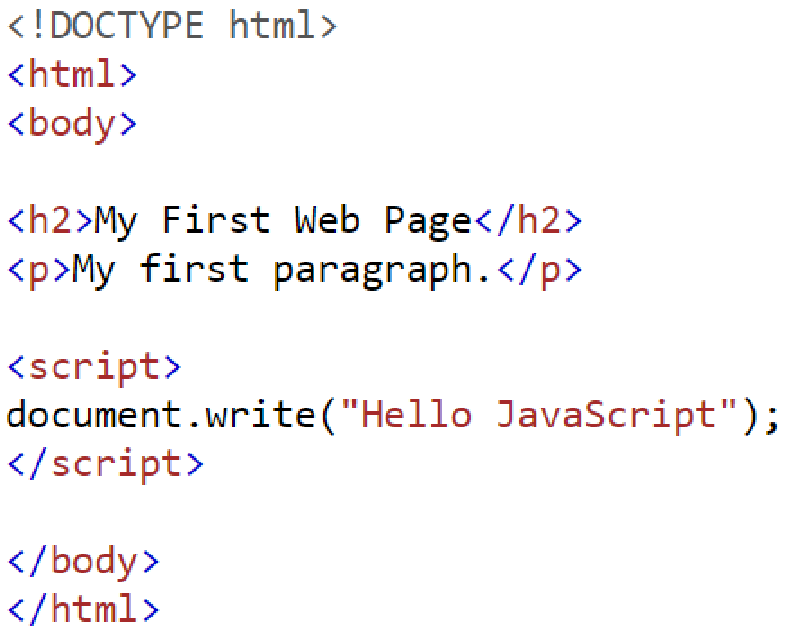
My First Paragraph.

Hello JavaScript

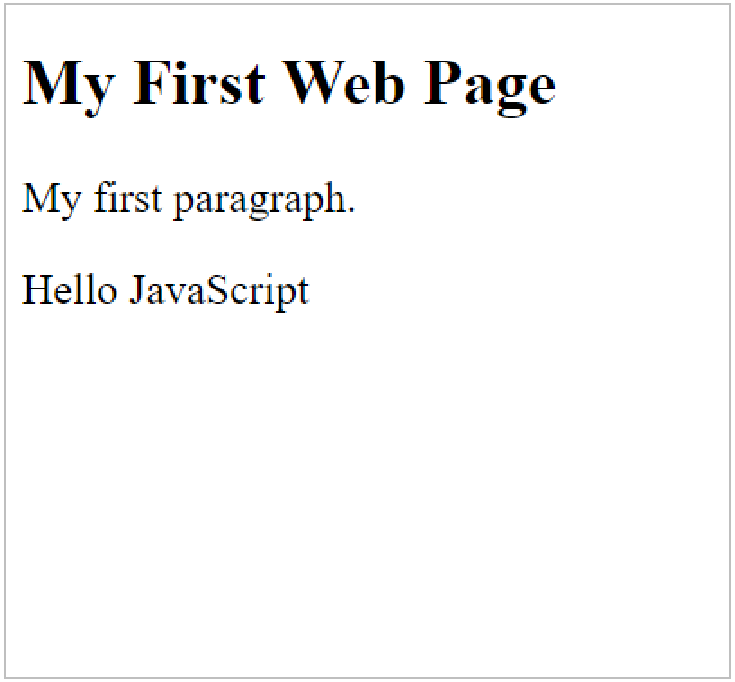
JavaScript Output

2. Writing into the HTML output using document.write().

```
document.write("Hello JavaScript")
```



```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
document.write("Hello JavaScript");
</script>

</body>
</html>
```



My First Web Page

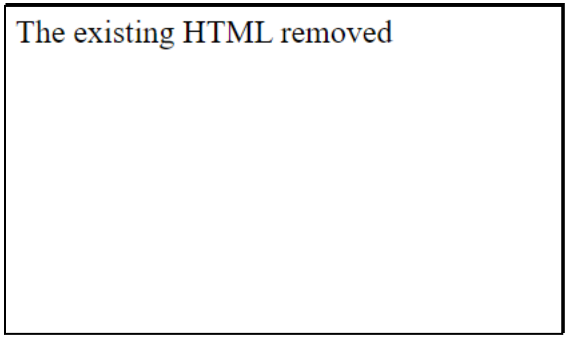
My first paragraph.

Hello JavaScript

JavaScript Output

Note that using document.write() after an HTML document is loaded, will **delete all existing HTML**

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write('The existing
HTML removed')">Try it</button>

</body>
</html>
```



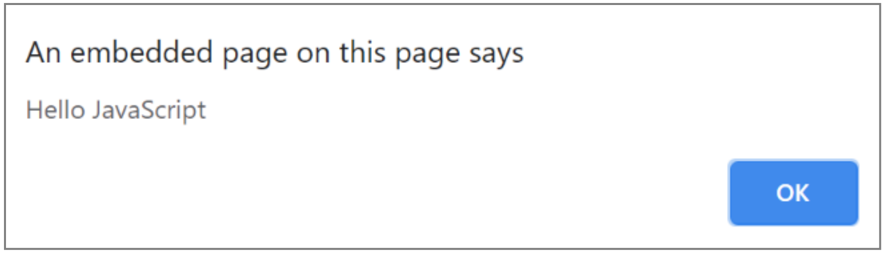
The existing HTML removed

JavaScript Output

3. Writing into an alert box, using window.alert().

```
window.alert("Hello JavaScript");
```

```
<!DOCTYPE html>
<html>
<body>

<script>
window.alert("Hello JavaScript");
</script>

</body>
</html>
```



An embedded page on this page says

Hello JavaScript

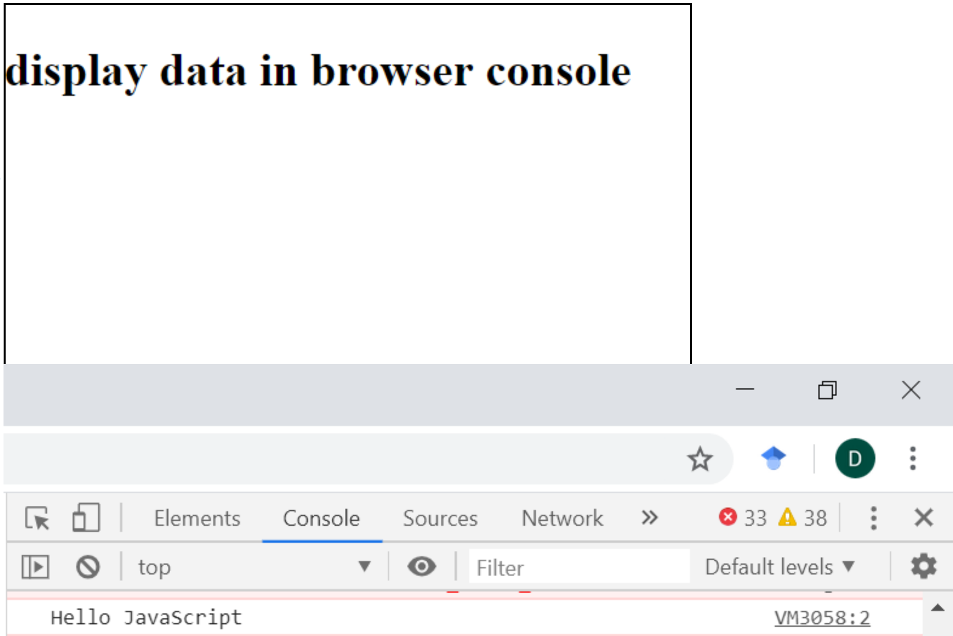
JavaScript Output

4. Writing into the browser console, using console.log().

```
console.log("Hello JavaScript");
```

For debugging purposes, you can call the console.log() method in the browser to display data.

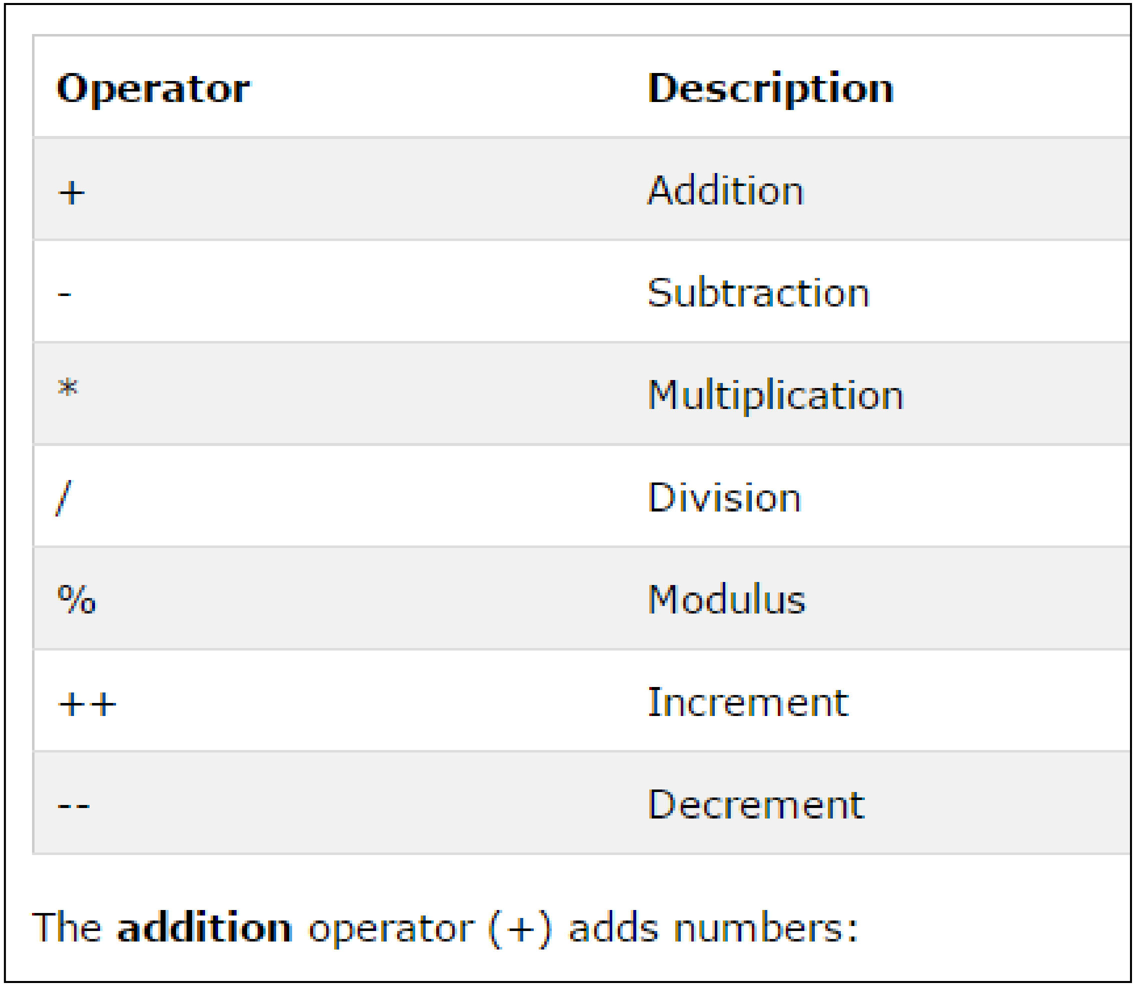
```
<!DOCTYPE html>
<html>
<body>

<h2>display data in browser console</h2>

<script>
console.log("Hello JavaScript");
</script>

</body>
</html>
```



display data in browser console

```
Hello JavaScript                                    VM3058:2
```

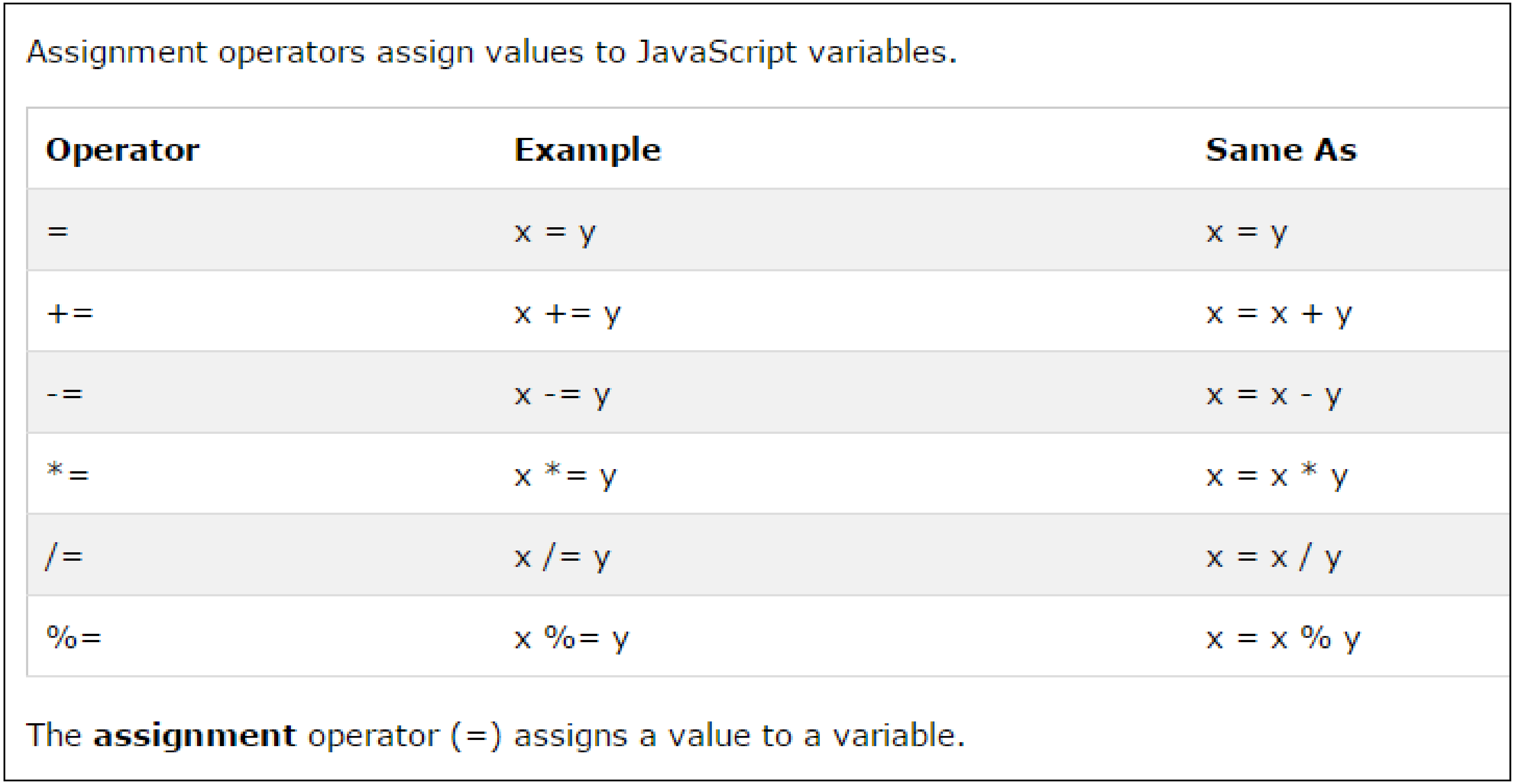
JavaScript Arithmetic Operators



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

The **addition** operator (+) adds numbers:

JavaScript Assignment Operators

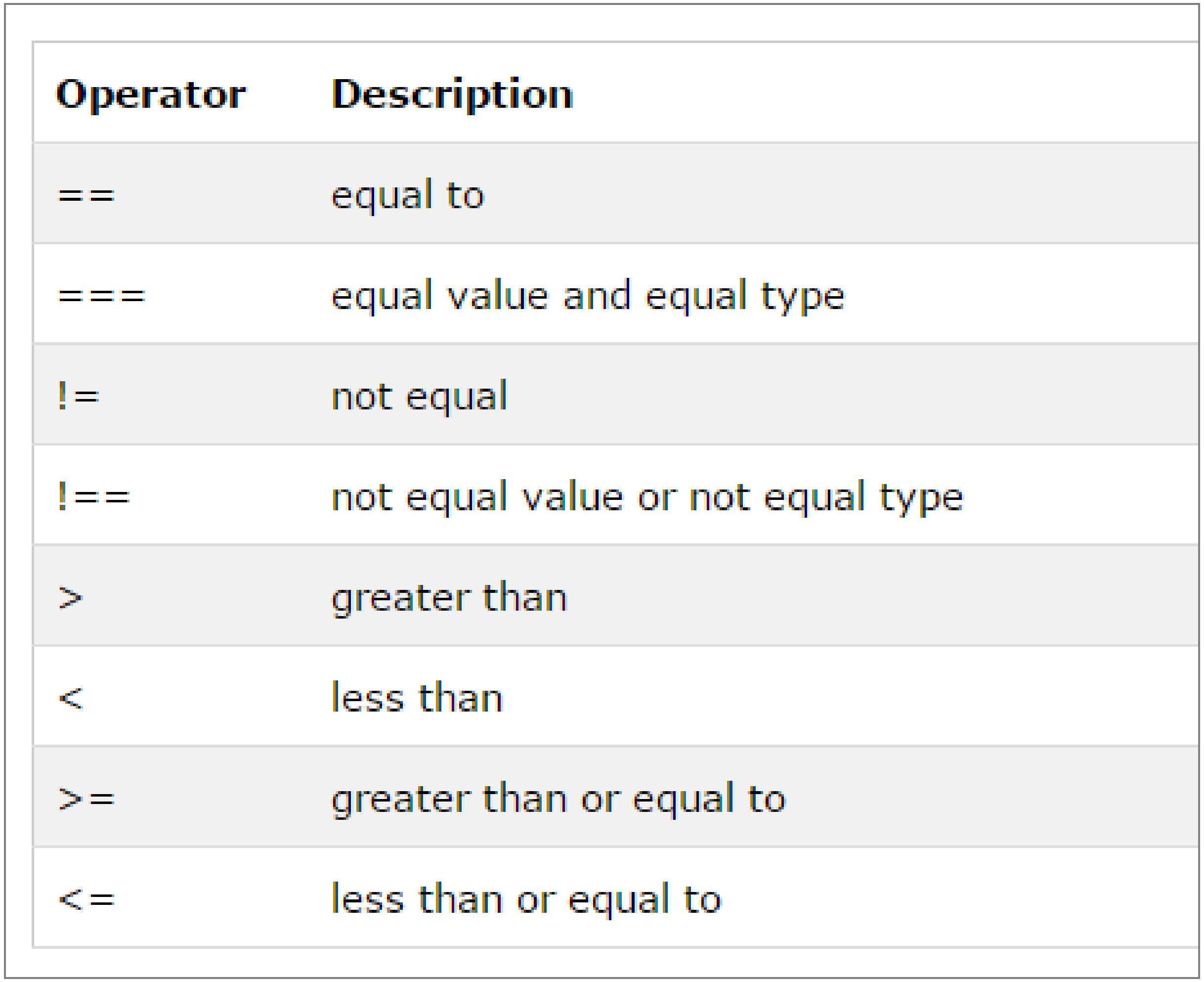


Assignment operators assign values to JavaScript variables.

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

The **assignment** operator (=) assigns a value to a variable.

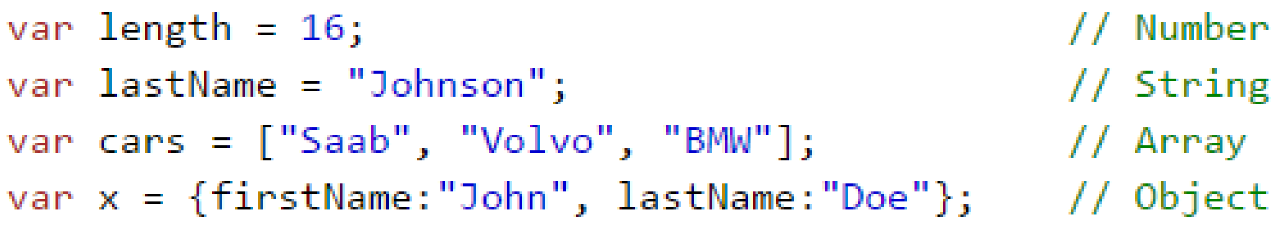
JavaScript Comparison and Logical Operators



Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

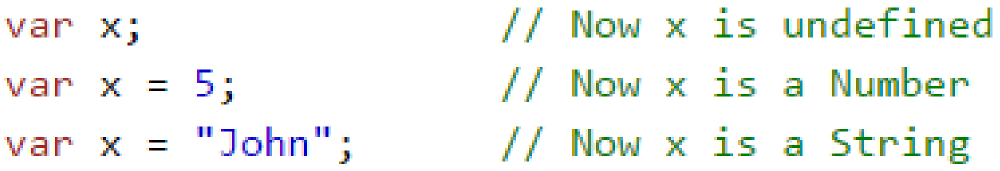
JavaScript Data Types

- JavaScript variables can hold many **data types**: String, Number, Boolean, Array, Object.



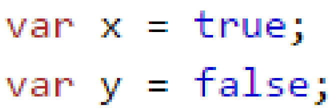
```
var length = 16;                                  // Number
var lastName = "Johnson";                         // String
var cars = ["Saab", "Volvo", "BMW"];              // Array
var x = {firstName:"John", lastName:"Doe"};       // Object
```

- JavaScript Has Dynamic Types



```
var x;               // Now x is undefined
var x = 5;           // Now x is a Number
var x = "John";      // Now x is a String
```

- JavaScript Booleans

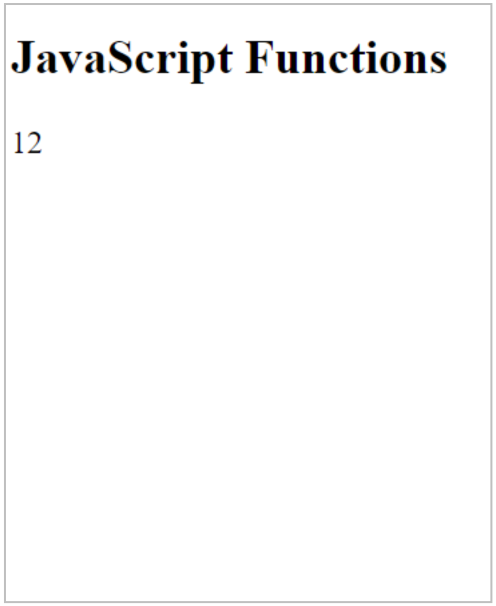


```
var x = true;
var y = false;
```

Functions

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Functions</h2>
<p id="demo"></p>

<script>
function myFunction(p1, p2) {
  return p1 * p2;
}
document.getElementById("demo").innerHTML = myFunction(4, 3);
</script>

</body>
</html>
```



JavaScript Functions

12

JavaScript Objects

```
var objectName = {propertyName:value, propertyName:value,..,
                  propertyName:value};
```

- Object properties can be accessed in two way :

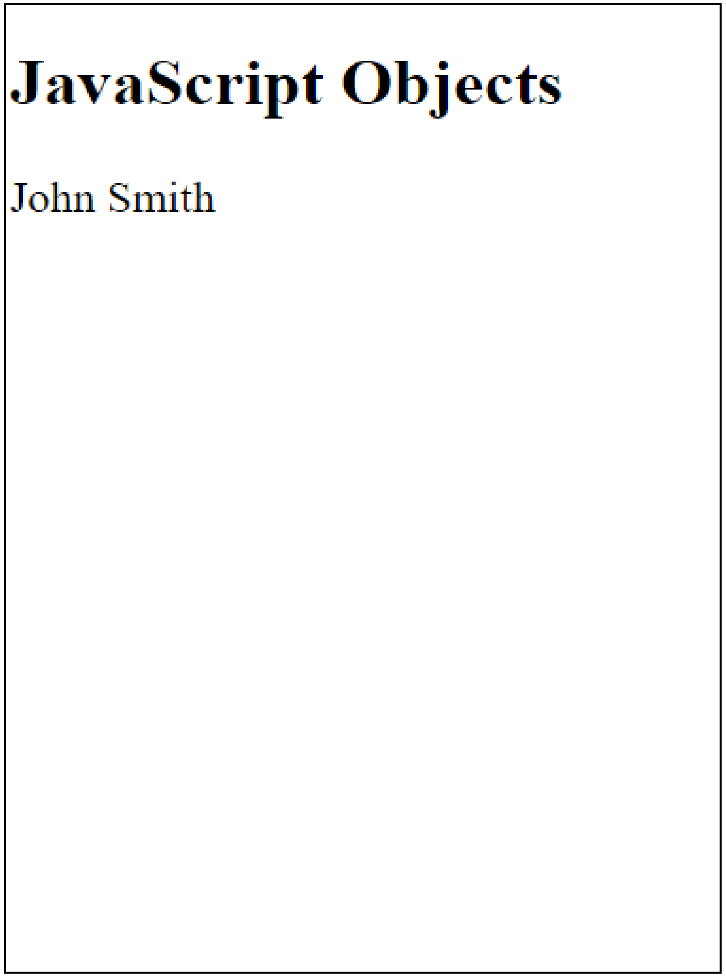
```
objectName.propertyName or
objectName["propertyName"]
```

- Object methods are **actions** that can be performed on objects. A method is a function stored as a property and can be accessed using:

```
objectName.methodName()
```

Example

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Objects</h2>

<p id="demo"></p>

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Smith",
  id       : 5566,
  fullName : function() {
    return person.firstName + " " + person.lastName;
  }
};
// Display data from the object:
document.getElementById("demo").innerHTML = person.fullName();
</script>

</body>
</html>
```



JavaScript Objects

John Smith

JavaScript If...Else Statements

```
if (condition) {
  //  block of code to be executed if
the condition is true
}
```

JavaScript Switch Statement

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

JavaScript For Loop

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

JavaScript While Loop

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

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