

# *Client Side Scripting Using JavaScript*

[www.w3schools.com/js](http://www.w3schools.com/js)



# ***Browser Must Haves***

- HTML Parser/renderer (basic)
- XML Processor
- CSS Parser/renderer
- **Javascript Engine**
- Plugin Framework/API



# *Javascript*

- HTML/XML-embedded
  - Javascript code can be embedded in HTML via
    - An external script file (\*.js) linked to the HTML page via `<link>`
    - Via `<script>` tag (internal)
    - Or embedded in some HTML tag attributes
- Object-based
  - Page/document, browser, window objects +
  - Several other objects
- Scripting Language
  - Scripts are only for controlling/using defined objects.

# *Terms Connected To Javascript*

- Brendan Eich
  - Netscape Communications Corporation
  - Netscape Navigator 2.0B3 (Dec, 1995)
  - *Mocha*
  - *LiveScript*
  - *Jscript*
  - *Java*
  - *ECMA*
  - *ECMAScript*
  - *Javascript 1.7*
- 
-

# *What is JavaScript?*

*www.w3schools.com*

## JavaScript...

- was designed to add interactivity to HTML Pages
- is a scripting language (lightweight programming language)
- is usually embedded directly to HTML pages
- is an interpreted language (scripts execute without preliminary compilation)
- is FREE



# *What can JavaScript do?*

*[www.w3schools.com](http://www.w3schools.com)*

## JavaScript...

- gives HTML designers a programming tool
- can react to events
- can read and write HTML elements
- can be used to validate data
- can be used to detect a visitor's browser
- can be used to create cookies



# *JavaScript is an embedded language*

In HTML we can embed JavaScript (just like in CSS)

- External script file (\*.js extension) linking via <link rel="javascript" >, also <script src="url"/>

```
<link rel="javascript" href="jsfile.js"
      type="text/javascript" />
```

```
<script language="JavaScript"
src="http://sample.com"></script>
```

# *JavaScript is an embedded language*

In HTML we can embed JavaScript (just like in CSS)

- Internal scripts via `<script>` tag, which may be embedded anywhere (in head or body)

```
<html>
<body>
<h1>My First Web Page</h1>
<p id="demo"></p>
<script type="text/javascript">
document.getElementById("demo").in
nerHTML=Date();
</script>
</body>
</html>
```

```
html>
<head>
<script type="text/javascript">
function displayDate()
{
document.getElementById("demo").in
nerHTML=Date();
}
</script>
</head>
<body>
...
```



# *JavaScript is an embedded language*

In HTML we can embed JavaScript (just like in CSS)

- Inline script via the **onxxxxx** attributes of different HTML elements.

```
...  
<input type="button" onclick="show_alert()" value="Show alert box" />  
...
```

# JavaScript Basics

- JavaScript is **Case Sensitive**.
- A semicolon (;) at the end of each statement is **optional**.
  - adding a semicolon at the end of each statement is a good programming practice.
  - Allows multiple statements in one line
- // This is a single line comment  
/\* This is a block/multi-line comment \*/

# JavaScript Variables

Variables have no associated type and may change type during run time.

```
x=10; //variable declaration, global  
var line='To JS or not to JS';
```

```
function getSD(val, arr, trunc) {  
  //function body  
  y = 101;  
}  
//var declarations outside functions are  
//global
```

\*The “var” keyword is optional.

---

# JavaScript Operators

= is the assignment operator

Given  $y=5$ ,

Operator	Description	Example	Result	
+	Addition	$x=y+2$	$x=7$	$y=5$
-	Subtraction	$x=y-2$	$x=3$	$y=5$
*	Multiplication	$x=y*2$	$x=10$	$y=5$
/	Division	$x=y/2$	$x=2.5$	$y=5$
%	Modulus (division remainder)	$x=y\%2$	$x=1$	$y=5$
++	Increment	$x=++y$	$x=6$	$y=6$
		$x=y++$	$x=5$	$y=6$
--	Decrement	$x=--y$	$x=4$	$y=4$
		$x=y--$	$x=5$	$y=4$

# JavaScript Operators

Given  $x=10$ ,  $y=5$ ,

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

The  $+$  operator can also be used to concatenate strings.

```
txt1="CMSC"; txt2="100"; txt3=txt1+txt2;  
txt3="CMSC100"
```

# *JavaScript Operators*

## Adding Strings and Numbers

`x=5+5;`                      ans. 10

`x="5"+"5";`                  ans. “55”

`x=5+"5";`                    ans. “55”

`x="5"+5;`                    ans. “55”

If a number and a string are added, the result will be a string.

---

---

# JavaScript Comparison and Logical Operators

Comparison Operators: Given  $x=5$ ,

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

Logical Operators: Given  $x=6$  and  $y=3$ ,

Operator	Description	Example
<code>&amp;&amp;</code>	and	<code>(x &lt; 10 &amp;&amp; y &gt; 1)</code> is true
<code>  </code>	or	<code>(x==5    y==5)</code> is false
<code>!</code>	not	<code>!(x==y)</code> is true

# *JavaScript Conditional Operator*

```
variablename=(condition)?value1:value2;
```

If condition is “true”, then value1 will be assigned to variable name, otherwise value2 will be assigned.

```
Greeting=(visitor=="PRES")?"Dear  
President ":"Dear ";
```





# JavaScript Conditional Statements

## If statement

```
if (condition)
{
    code to be executed if condition is true
}
```

## If..else statement

```
if (condition)
{
    code to be executed if condition is true
}
else
{
    code to be executed if condition is not true
}
```

---

---

# *JavaScript Conditional Statements*

## If...else if...else statement

```
if (condition1)
{
  code to be executed if condition1 is true
}
else if (condition2)
{
  code to be executed if condition2 is true
}
else
{
  code to be executed if neither condition1 nor
  condition2 is true
}
```

# *JavaScript Switch Statement*

## Switch statement

```
switch(n)
{
case 1:
    execute code block 1
    break;
case 2:
    execute code block 2
    break;
default:
    code to be executed if n is different from
    case 1 and 2
}
```

---

---