

Introduction to JavaScript Training

JavaScript Basics

Lesson 1, Activity 1: The Name "JavaScript"

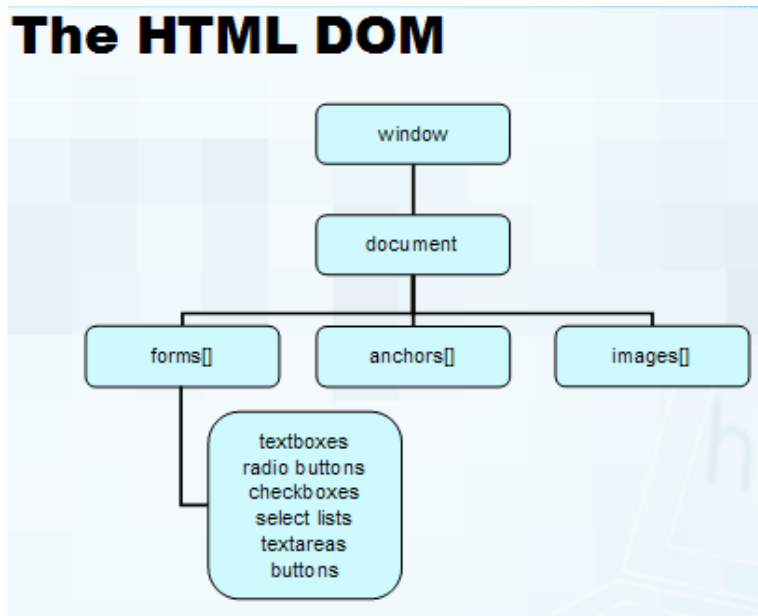
In this course, we refer to the language we are learning as *JavaScript*, which is what it is usually called. However, *JavaScript* was invented by Netscape Communications and is now [owned by Oracle Corporation](#). Microsoft calls its version of the language *JScript*. JavaScript and JScript are both implementations of [EcmaScript](#), but everyone still refers to the language as JavaScript.

Lesson 1, Activity 3: The Name "JavaScript"

In this course, we refer to the language we are learning as *JavaScript*, which is what it is usually called. However, *JavaScript* was invented by Netscape Communications and is now [owned by Oracle Corporation](#). Microsoft calls its version of the language *JScript*. JavaScript and JScript are both implementations of [EcmaScript](#), but everyone still refers to the language as JavaScript.

The HTML DOM

The HTML Document Object Model (DOM) is the browser's view of an HTML page as an object hierarchy, starting with the browser window itself and moving deeper into the page, including all of the elements on the page and their attributes. Below is a simplified version of the HTML DOM.



As shown, the top-level object is `window`. The `document` object is a child of `window` and all the objects (i.e., elements or nodes) that appear on the page (e.g., forms, links, images, tables, etc.) are descendants of the `document` object. These objects can have children of their own. For example, `form` objects generally have several child objects, including textboxes, radio buttons, and select menus.

JavaScript Syntax

Basic Rules

1. JavaScript statements end with semi-colons.
2. JavaScript is case sensitive.
3. JavaScript has two forms of comments:
 - Single-line comments begin with a double slash (`//`).
 - Multi-line comments begin with `/*` and end with `*/`.

Syntax

```
// This is a single-line comment
```

```
/*  
  This is  
  a multi-line  
  comment.  
*/
```

Lesson 1, Activity 4: JavaScript Syntax

Basic Rules

1. JavaScript statements end with semi-colons.
2. JavaScript is case sensitive.
3. JavaScript has two forms of comments:
 - Single-line comments begin with a double slash (//).
 - Multi-line comments begin with "/*" and end with "*/".

Syntax

```
// This is a single-line comment

/*
This is
a multi-line
comment.
*/
```

Lesson 1, Activity 5: Accessing Elements

Dot Notation

In JavaScript, elements (and other objects) can be referenced using dot notation, starting with the highest-level object (i.e, `window`). Objects can be referred to by name or id or by their position on the page. For example, if there is a form on the page named "loginform", using dot notation you could refer to the form as follows:

```
window.document.loginform
```

Assuming that `loginform` is the first form on the page, you could also refer to this way:

```
window.document.forms[0]
```

A document can have multiple form elements as children. The number in the square brackets (`[]`) indicates the specific form in question. In programming speak, every document object contains a *collection* of forms. The length of the collection could be zero (meaning there are no forms on the page) or greater. In JavaScript, collections (and arrays) are zero-based, meaning that the first form on the page is referenced with the number zero (0) as shown in the syntax example above.

Square Bracket Notation

Objects can also be referenced using square bracket notation as shown below:

```
window['document']['loginform']  
  
// and  
  
window['document']['forms'][0]
```

Dot notation and square bracket notation are completely interchangeable. Dot notation is much more common; however, as we will see later in the course, there are times when it is more convenient to use square bracket notation.

The Implicit window Object

The `window` object is always the implicit top-level object and therefore does not have to be included in references to objects. For example, `window.document.forms[0]` can be shortened to `document.forms[0]`.

Lesson 1, Activity 6: Where Is JavaScript Code Written?

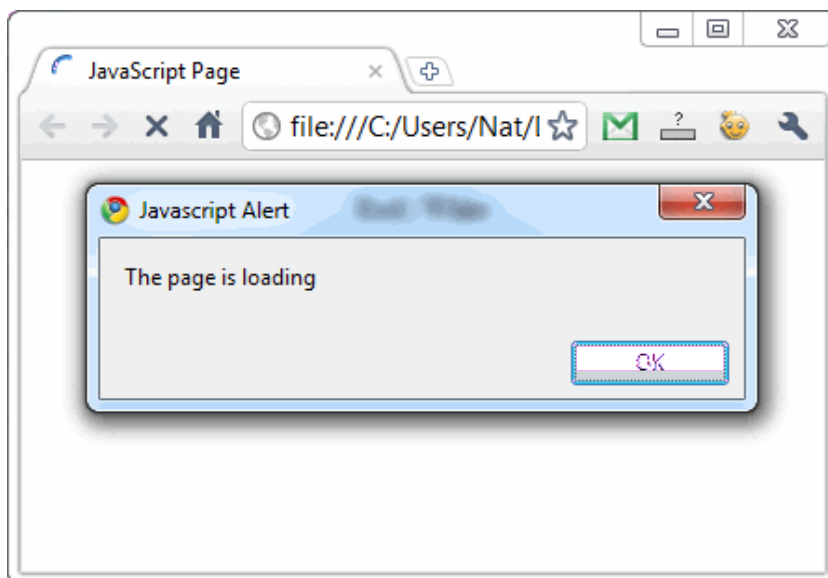
JavaScript code can be written inline (e.g, within HTML tags called event handlers), in `script` blocks, and in external JavaScript files. The page below shows examples of all three.

Code Sample:

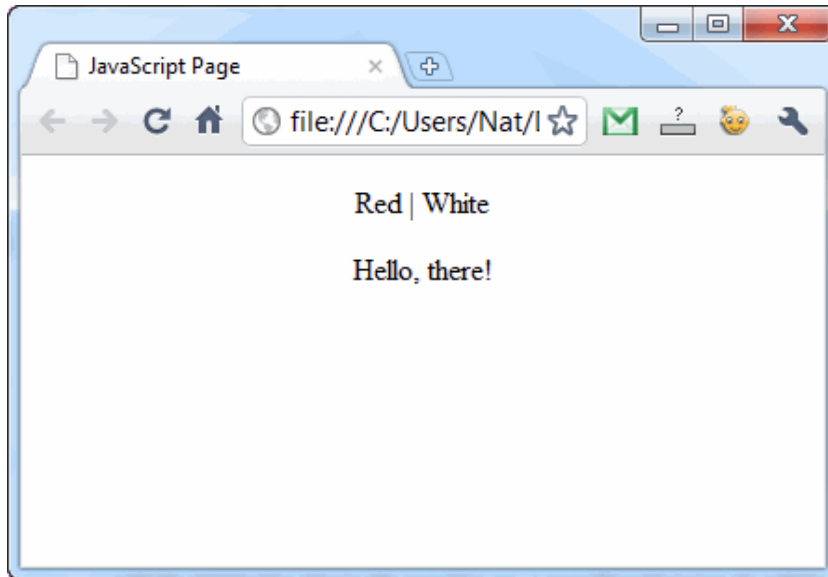
JavaScriptBasics/Demos/javascript-1.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Page</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  window.alert("The page is loading");
</script>
</head>
<body>
<p>
  <span onclick="document.bgColor='red';">Red</span> |
  <span onclick="document.bgColor='white';">White</span>
</p>
<script type="text/javascript" src="script-1.js"></script>
</body>
</html>
```

As this page loads, an alert will pop up that says "The page is loading" as shown below:



After the user clicks the OK button, the page will finish loading and will appear as follows:



Code Sample:

JavaScriptBasics/Demos/script-1.js

```
document.write("<p>Hello, there!</p>");
```

The text "Hello, there!" is written dynamically by the code in JavaScriptBasics/Demos/script-1.js. We will look at the code in this file and in JavaScriptBasics/Demos/javascript-1.html again shortly.

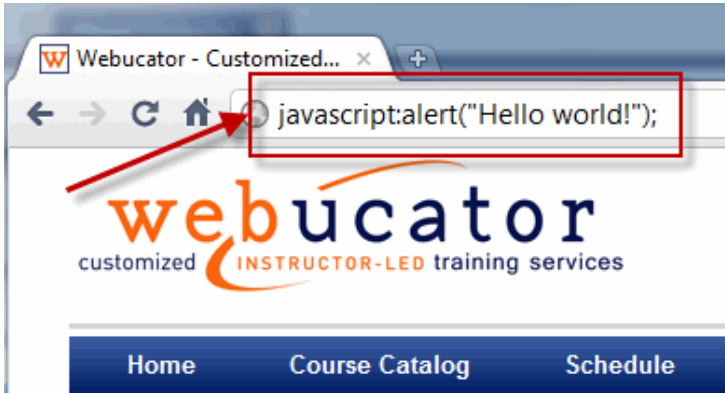
In HTML 4, the `<script>` tag must contain a `type` attribute set to `text/javascript` like this:
`<script type="text/javascript">`.

In HTML5, the assumed (default) value for `type` is `text/javascript` so it's not necessary to include the attribute, but it doesn't hurt.

Lesson 1, Activity 8: javascript: Pseudo-Protocol

Try this:

1. Open your browser.
2. In the location bar, type in `javascript:alert("Hello world!");` like so:



3. Press **Enter**.

You should get an alert reading "Hello world!". The `javascript:` prefix is called a pseudo-protocol, because it mimics the protocol syntax (e.g, like `http:`, `ftp:`, and `mailto:`). It provides an easy way to test JavaScript on a page. It can also be used in links as in the demo below:

Code Sample:

<JavaScriptBasics/Demos/psuedo-protocol.html>

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Pseudo-Protocol</title>
</head>
<body>
<p><a href="javascript:alert('Hello world!');">Say hello</a></p>
</body>
</html>
```

When the user clicks the link, the JavaScript alert will execute.

This is generally considered a bad practice. The preferred way to handle this is to use the `href` attribute to provide an alternative page in case JavaScript is not enabled. You could provide a generic page with a "JavaScript is required" message or, even better, you could provide a page that accomplishes the same task as the alert as in the following the demo:

Code Sample:

<JavaScriptBasics/Demos/better-than-psuedo-protocol.html>

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Better than Pseudo-Protocol</title>
</head>
<body>
<p><a href="hello-world.html" onclick="alert('Hello world!'); return false;">Say hello</a></p>
</body>
</html>
```

When the user clicks the link...

- If JavaScript is enabled, the alert pops up and the `return false;` statement explicitly cancels the link's default behavior.
- If JavaScript is NOT enabled, the link takes the user to [hello-world.html](#).

Lesson 1, Activity 9: JavaScript Objects, Methods and Properties

JavaScript is used to manipulate or get information about objects in the HTML DOM. Objects in an HTML page have methods (actions, such as opening a new window or submitting a form) and properties (attributes or qualities, such as color and size).

To illustrate objects, methods and properties, we will look at the code in [JavaScriptBasics/Demos/javascript-2.html](#), a slightly modified version of [JavaScriptBasics/Demos/javascript-1.html](#), which we looked at earlier, and at the code in [JavaScriptBasics/Demos/script-2.js](#)

Code Sample:

[JavaScriptBasics/Demos/javascript-2.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript Page</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  //Pop up an alert
  window.alert("The page is loading");
</script>
</head>
<body>
<p>
  <span onclick="document.bgColor = 'red';">Red</span> |
  <span onclick="document.bgColor = 'white';">White</span> |
  <span onclick="document.bgColor = 'green';">Green</span> |
  <span onclick="document.bgColor = 'blue';">Blue</span>
</p>
<script type="text/javascript" src="script-2.js"></script>
</body>
</html>
```

Code Sample:

[JavaScriptBasics/Demos/script-2.js](#)

```
/*
This script simply outputs
  "Hello, there!"
to the browser.
*/
document.write("<p>Hello, there!</p>");
```

Methods

Methods are the verbs of JavaScript. They cause things to happen.

window.alert()

HTML pages are read and processed from top to bottom. The JavaScript code in the initial `script` block at the top of [JavaScriptBasics/Demos/javascript-2.html](#) calls the `alert()` method of the `window` object. When the browser reads that line of code, it will pop up an alert box and will not continue processing the page until the user presses the OK button. Once the user presses the button, the alert box disappears and the rest of the page loads.

document.write()

The `write()` method of the `document` object is used to write out code to the page as it loads. In [JavaScriptBasics/Demos/script-2.js](#), it simply writes out "Hello, there!"; however, it is more often used to write out dynamic data, such as the date and time on the user's machine.

Arguments

Methods can take zero or more arguments separated by commas.

Syntax

```
object.method(argument1, argument2);
```

The `alert()` and `write()` methods shown in the example above each take only one argument: the message to show or the HTML to write out to the browser.

Properties

Properties are the adjectives of JavaScript. They describe qualities of objects and, in some cases are writable (can be changed dynamically).

document.bgColor

The `bgColor` property of the `document` object is read-write. Looking back at [JavaScriptBasics/Demos/javascript-2.html](#), the four `span` elements use the `onclick` event handler to catch click events. When the user clicks on a `span`, JavaScript is used to change the value of the `bgColor` property to a new color.

The `bgColor` property of `document` is deprecated, meaning it should not be used anymore. Instead, `document.body.style.backgroundColor` should be used to change the background color of the page. However, we do not get to the `style` property in this course, so we use `bgColor` for learning purposes. In practice, you can substitute `document.body.style.backgroundColor` for `document.bgColor`.

The Implicit window Object

The `window` object is always the implicit top-level object and therefore does not have to be included in references to objects. For example, `window.document.write()` can be shortened to `document.write()`. Likewise, `window.alert()` can be shortened to just `alert()`.

Lesson 1, Activity 11: Event Handlers

In [JavaScriptBasics/Demos/javascript-2.html](#), we used the `onclick` event handler to call JavaScript code that changed the background color of the page. Event handlers are attributes that force an element to "listen" for a specific event to occur. Event handlers all begin with the letters "on". The table below lists the HTML event handlers with descriptions.

HTML Event Handlers

Event Handler	Elements Supported	Description
<code>onblur</code>	<code>a</code> , <code>area</code> , <code>button</code> , <code>input</code> , <code>label</code> , <code>select</code> , <code>textarea</code>	the element lost the focus
<code>onchange</code>	<code>input</code> , <code>select</code> , <code>textarea</code>	the element value was changed
<code>onclick</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a pointer button was clicked
<code>ondblclick</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a pointer button was double clicked
<code>onfocus</code>	<code>a</code> , <code>area</code> , <code>button</code> , <code>input</code> , <code>label</code> , <code>select</code> , <code>textarea</code>	the element received the focus
<code>onkeydown</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a key was pressed down
<code>onkeypress</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a key was pressed and released
<code>onkeyup</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a key was released
<code>onload</code>	<code>frameset</code>	all the frames have been loaded
<code>onload</code>	<code>body</code>	the document has been loaded
<code>onmousedown</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a pointer button was pressed down
<code>onmousemove</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a pointer was moved within
<code>onmouseout</code>	All elements except <code>applet</code> , <code>base</code> , <code>basefont</code> , <code>bdo</code> , <code>br</code> , <code>font</code> , <code>frame</code> , <code>frameset</code> , <code>head</code> , <code>html</code> , <code>iframe</code> , <code>isindex</code> , <code>meta</code> , <code>param</code> , <code>script</code> , <code>style</code> , <code>title</code>	a pointer was moved away

onmouseover	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer was moved onto
onmouseup	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer button was released
onreset	form	the form was reset
onselect	input, textarea	some text was selected
onsubmit	form	the form was submitted
onunload	frameset	all the frames have been removed
onunload	body	the document has been removed

For a full reference of HTML event handler attributes, see the [list of attributes and the elements they apply to on the W3C web site](#).

The getElementById() Method

A very common way to reference HTML elements is by their ID using the `getElementById()` method of the document object as shown in the example below.

getElementById()		
Description	Syntax	Parameter
Used to accesses/manipulate the first element with the specified id.	<code>document.getElementById("id")</code>	The "id" parameter is required. This refers to the id of the HTML element you want to access/manipulate

Code Sample:

[JavaScriptBasics/Demos/getElementById.html](#)

```

---- C O D E   O M I T T E D ----

<p>
  <span onclick="document.getElementById('divRed').bgColor = 'red';">
    Red</span> |
  <span onclick="document.getElementById('divOrange').bgColor = 'orange';">
    Orange</span> |
  <span onclick="document.getElementById('divGreen').bgColor = 'green';">
    Green</span> |
  <span onclick="document.getElementById('divBlue').bgColor = 'blue';">
    Blue</span>
</p>
<table>

```

```
<tr id="divRed"><td>Red</td></tr>
<tr id="divOrange"><td>Orange</td></tr>
<tr id="divGreen"><td>Green</td></tr>
<tr id="divBlue"><td>Blue</td></tr>
</table>
</body>
</html>
```

Lesson 1, Activity 13: Using Event Handlers

Duration: 15 to 25 minutes.

In this exercise, you will use event handlers to allow the user to change the background color of the page.

1. Open [JavaScriptBasics/Exercises/color-changer.html](#) for editing.
2. Modify the page so that...
 - when it is finished loading an alert pops up reading "The page has loaded!" You can make use of [hello.js](#).
 - when the "Red" button is *clicked*, the background color turns red and an alert pops up reading "The background color is now Red."
 - when the "Green" button is *double-clicked*, the background color turns green and an alert pops up reading "The background color is now Green."
 - when the "Orange" button is *clicked down*, the background color turns orange and an alert pops up reading "The background color is now Orange."
 - when the mouse button is *released* over the "Blue" button, the background color turns blue and an alert pops up reading "The background color is now Blue."

Code Sample:

[JavaScriptBasics/Exercises/color-changer.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Color Changer</title>
<link href="style.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
  window.alert("The page is loading.");
</script>
</head>
<body>
<form>
  <p>Click the button to turn the page:
  <input type="button" value="Red"></p>
  <p>Double click the button to turn the page:
  <input type="button" value="Green"></p>
  <p>Click down on the button to turn the page:
  <input type="button" value="Orange"></p>
  <p>Release the mouse while on the button to turn the page:
  <input type="button" value="Blue"></p>
</form>
<hr>
</body>
</html>
```

Challenge

1. Add functionality so that when the user presses any key, the background color turns white.
2. Add a "Black" button. When the user hovers over this button and presses the mouse button down, the

background color should turn black. When the user releases the mouse button, the background color should turn white.

Solution:

JavaScriptBasics/Solutions/color-changer.html

```

---- C O D E   O M I T T E D ----

<script type="text/javascript">
  window.alert("The page is loading.");
</script>
</head>
<body>
<form>
  <p>Click the button to turn the page:
  <input type="button" value="Red"
    onclick="document.bgColor = 'red';
      alert('The background color is now Red.');"></p>
  <p>Double click the button to turn the page
  <input type="button" value="Green"
    ondblclick="document.bgColor = 'green';
      alert('The background color is now Green.');"></p>
  <p>Click down on the button to turn the page
  <input type="button" value="Orange"
    onmousedown="document.bgColor = 'orange';
      alert('The background color is now Orange.');"></p>
  <p>Release the mouse while on the button to turn the page
  <input type="button" value="Blue"
    onmouseup="document.bgColor = 'blue';
      alert('The background color is now Blue.');"></p>
</form>
<hr>
<script type="text/javascript" src="hello.js"></script>
</body>
</html>

```

Challenge Solution:

JavaScriptBasics/Solutions/color-changer-challenge.html

```

---- C O D E   O M I T T E D ----

<body onkeypress="document.bgColor = 'white';
  alert('The background color is now White.');">
<form>

---- C O D E   O M I T T E D ----

  <p>Press any key to turn the page back to white.</p>
  <p>Click the button to turn the page
  <input type="button" value="Black"
    onmousedown="document.bgColor = 'black';"

```

```
    onmouseup="document.bgColor = 'white';"></p>  
</form>  
<hr>  
<script type="text/javascript" src="hello.js"></script>  
</body>  
</html>
```