

DHTML - Introduction

Contributed by Eddie Traversa
Wednesday, 15 November 2006

Think of DHTML as not a singular technology but a combination of three existing technologies glued together by the Document Object Model (DOM):
{mosbotwordcount}

Think of DHTML as not a singular technology but a combination of three existing technologies glued together by the Document Object Model (DOM):

1. HTML - For creating text and image links and other page elements.
2. CSS - Style Sheets for further formatting of text and html plus other added features such as positioning and layering content.
3. JavaScript - The programming language that allows you to access and dynamically control the individual properties of both HTML and Style Sheets.

{bot_wrgoogle}The way JavaScript accesses the properties of an HTML document is through the Document Object Model (DOM). The job of the DOM is to expose all the attributes of HTML and Style sheets to JavaScript control. All you need to know about the DOM is what JavaScript commands it accepts. Not that easy, as different browsers have their slightly different versions of the DOM, so they access HTML properties differently as well as display them differently.

So how do you locate an HTML element on a page and change its property?

This is the job of JavaScript. Obviously, I cant into all the details of JavaScript or the DOM, but here is an example of how JavaScript can change a visibility of a style sheet layer in both browsers.

Note: That every piece of HTML has a location much like a directory in a phone book. When finding that piece of HTML you have to go through the same hierarchy process of searching for a name in the phone book such as

(state) Washington -> (City) Seattle -> (Listings) j -> (Name) Jessica

In JavaScript, a reference to this would be equivalent to

```
washington.seattle.j.jessica
```

Now Jessica may have additional information such as her address and phone number, so the JavaScript reference would be written this way.

```
washington.seattle.j.jessica.address
```

or

```
washington.seattle.j.jessica.phone
```

Lets transcribe the above metaphor to a DHTML document that contains a

layer [myLayer] with style attributes [top,left,width,height,z-index,visibility,etc] and the layer contains a bit of text "myText" (Note that the visibility attribute is set to hidden)

```
myText
```

In Netscape the address to the DIV layer "myLayer" is

```
document.myLayer
```

in Explorer it is

`document.all.myLayer.style`

The W3C way of identifying the address is

`document.getElementById('myLayer').style`

To access the properties such as visibility under "myLayer" you would use these addresses.

Netscape

`document.myLayer.visibility`

Explorer

`document.all.myLayer.style.visibility`

W3C

`document.getElementById('myLayer').style.visibility`

To change the visibility of this layer you would assign a value to your JavaScript address.

Netscape

`document.myLayer.visibility = "visible";`

Explorer

`document.all.myLayer.style.visibility = "visible";`

W3C

```
document.getElementById('myLayer').style.visibility="visible";
```

Now the previously hidden layer is now visible. This is essentially how DHTML works, but understand there are hundreds and hundreds of attribute properties for text, images, documents and windows. Not all these properties are supported in both browser and sometime accessing a property requires a few more hurdles, but if you stick to the common denominator properties both browser use then life it a bit easier. I recommend the excellent DHTML reference book Dynamic HTML - The Definitive Guide by Danny Goodman (O'Riley Books) It lists all of the DHTML properties and their cross browser compatibilities.

About the Author

Eddie Traversa

DHTML Nirvana <http://nirvana.media3.net/> is a site dedicated to exploring the possibilities of DHTML. It hosts free graphics, dhtml templates and tutorials. Some of the tutorials emphasis is on Flash/DHTML integration.