

Client Side Technologies: DHTML



Mosh Teitelbaum

mosh.teitelbaum@evoch.com

evoch, LLC

DHTML

- **DHTML = Dynamic HTML**
It allows you to build rich client interfaces and to modify them dynamically
- **There is no DHTML standard**
It is not a W3C, IEEE, ISO or anything else standard
- **DHTML is a collection of several standards**
DHTML consists of HTML/XHTML, CSS, DOM and JavaScript (or ECMAScript)

DHTML In A Nutshell

- DHTML is too rich to cover in an hour
The technologies are way too rich to fully cover in this presentation. This presentation will:
 - 1) Briefly introduce each technology with a quick example
 - 2) Give a high-level overview of how to use each technology
 - 3) Show some more advanced uses for the various technologies and review how each one works
 - 4) Provide resources for further exploration of each technology

HTML / XHTML

- **HTML = HyperText Markup Language**
Allows you to define and structure the contents of your document. Latest (last?) version is 4.01.
- **XHTML = XML HyperText Markup Language**
XML-Compliant definition of HTML. Current version is XHTML 1.1 which added no new HTML tags to HTML 4.01
- **Contents, not design**
HTML/XHTML was never intended to convey design

HTML / XHTML Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
    "http://www.w3.org/TR/html4/loose.dtd">  
<HTML>  
<HEAD>  
    <TITLE>Sample</TITLE>  
</HEAD>  
<BODY>  
    <P>This is a sample paragraph</P>  
</BODY>  
</HTML>
```

CSS

- **CSS = Cascading Style Sheets**
Allows you to define the styles to apply to your document.
Latest version is 2.1.
- **Design, not content**
CSS is intended to separate design from content
- **Very powerful**
CSS is much more powerful than HTML design tags

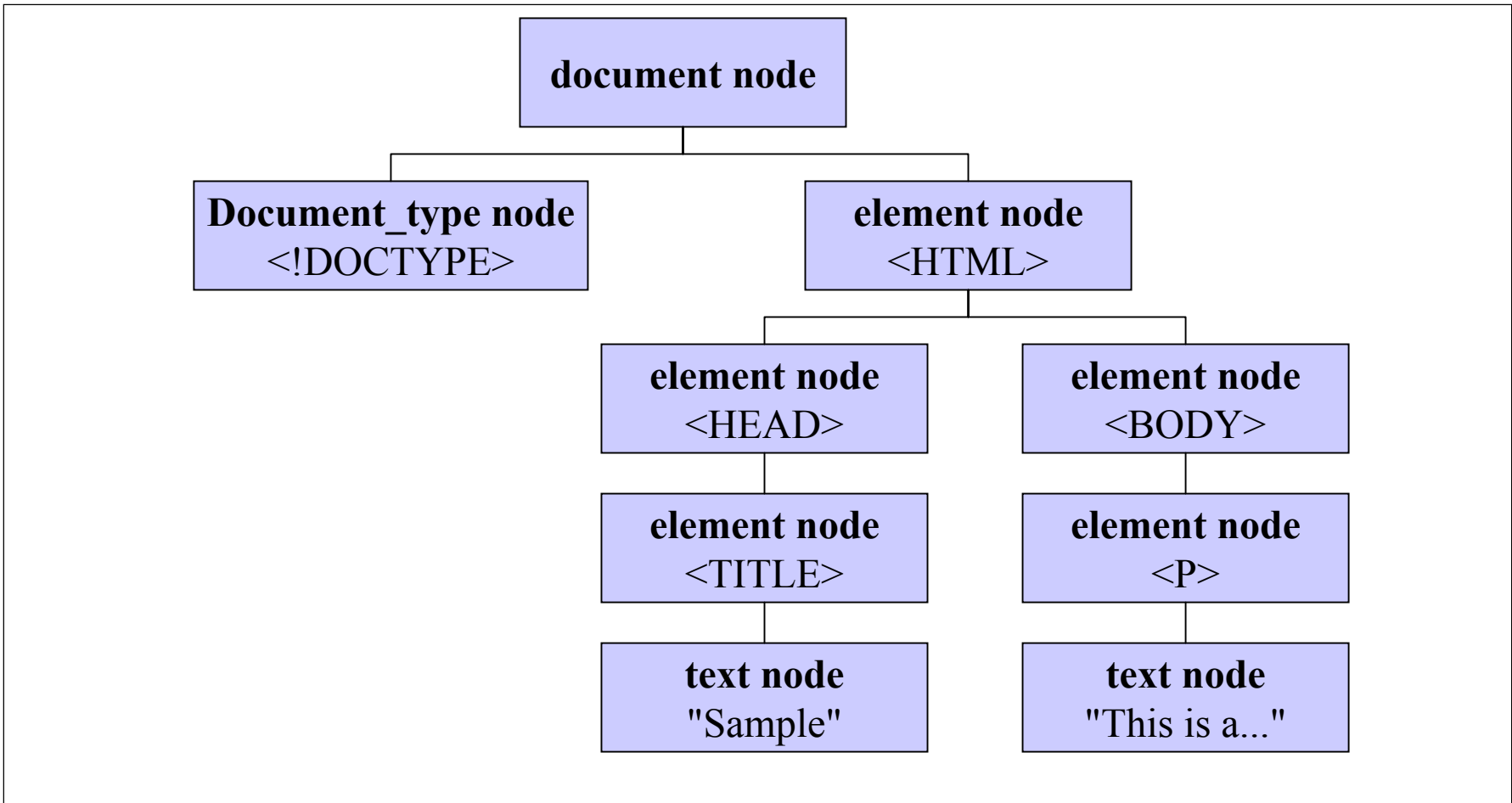
CSS Example

```
<STYLE TYPE="text/css">  
  BODY {  
    background-color: #CCCCCC;  
  }  
  
  P {  
    border: 1px solid black;  
    background-color: #FFFFFF;  
    margin-bottom: 1em;  
  }  
</STYLE>
```

DOM

- **DOM = Document Object Model**
Defines a hierarchical model of the document structure through which all document elements may be accessed
- **Nodes**
The W3C DOM defines element of a document is a *node* of a particular type
- **Node Types**
Common types are: document node, element node, text node, attribute node, comment node, document-type node

DOM Example



JavaScript

- **JavaScript**
Allows you to add conditional, client-side logic and behavior to your document
- **JavaScript != JAVA**
Even though they have similar names, they are very different
- **Very powerful**
Current versions are incredibly powerful... fully object-oriented, conditional logic, DOM interaction, more

JavaScript Example

```
<SCRIPT TYPE="text/javascript">
<!--
    function pushButton() {
        if ( confirm("Push a button") ) {
            alert("You pushed \"OK\"");
        } else {
            alert("You pushed \"Cancel\"");
        }
    }
// -->
</SCRIPT>
```

DHTML Example

- Style Switcher

Using JavaScript, this example dynamically "applies" the selected CSS style sheet, changing the design on the fly.

- JavaScript interacts with DOM and cookies
- Shows ability of CSS to affect design w/o changing HTML

```
function setActiveStyleSheet(title) {  
    var i, a, main;  
    for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {  
        if(a.getAttribute("rel").indexOf("style") != -1 && a.getAttribute("title")) {  
            a.disabled = true;  
            if(a.getAttribute("title") == title) a.disabled = false;  
        }  
    }  
}
```

Differences Between HTML and XHTML

- HTML tags and attributes must be lowercase
- All attribute values must be quoted
- All elements that can contain others require end tags
- Empty elements must either have an end tag or self-close
- All attributes must be name/value pairs
- The `name` attribute is deprecated. Use `id` instead.
- Some others...

Standards vs. Quirk Mode

- 2 Modes of Operation

All modern browsers support 2 modes: Standards Mode for standard-compliant code and "Quirks" Mode for older or non-compliant code

- !DOCTYPE Specifies the Mode

A properly formatted DOCTYPE declaration puts the browser in Standards Mode. Otherwise, it's Quirks Mode.

```
<!DOCTYPE OutermostTag RespOrg PublicIdentifier SystemIdentifier>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML and CSS

- **<STYLE>** tag

Delineates inline styles

```
<STYLE TYPE="text/css"> /* Styles go here... */ </STYLE>
```

- **<LINK>** tag

References external style sheets. Allows for alternates.

```
<LINK REL="stylesheet" HREF="default.css" TYPE="text/css">
```

- **STYLE** attribute

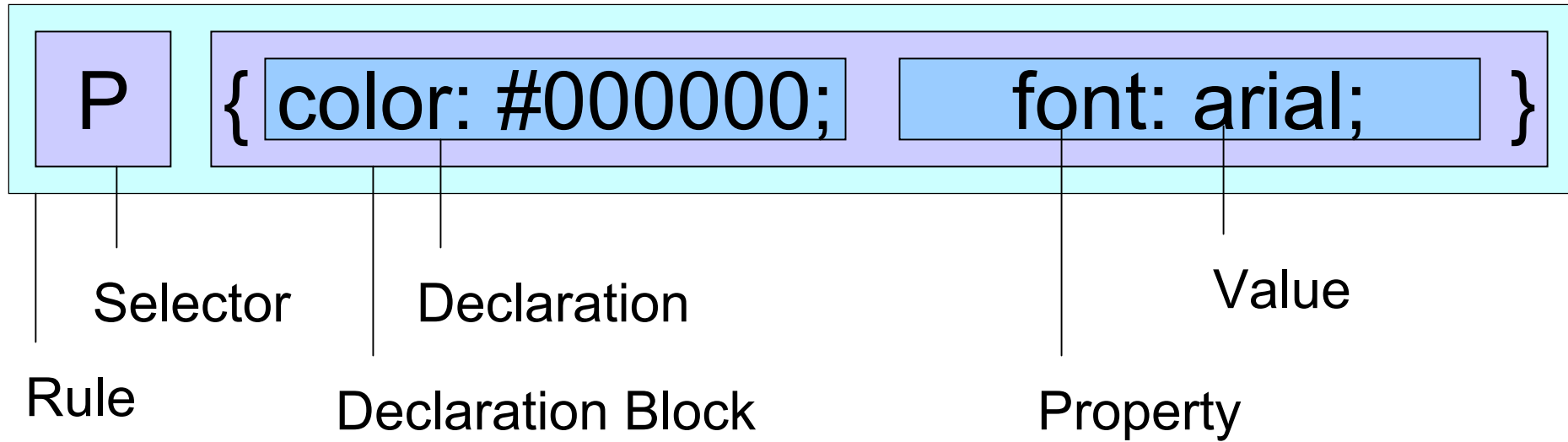
Defines inline styles for a specific block of HTML code

```
<P STYLE="color: #FF0000; font-weight: bold;"> some text </P>
```

CSS: Syntax

- **@import Directive**
Loads an external style sheet. Does not allow alternates.
Not supported in some older browsers.
- **Rules**
Defines which styles to apply to which elements
- **Selectors**
Specifies the element or type of element that style affects
- **Declarations**
Specifies CSS properties and values

CSS: Rule Structure



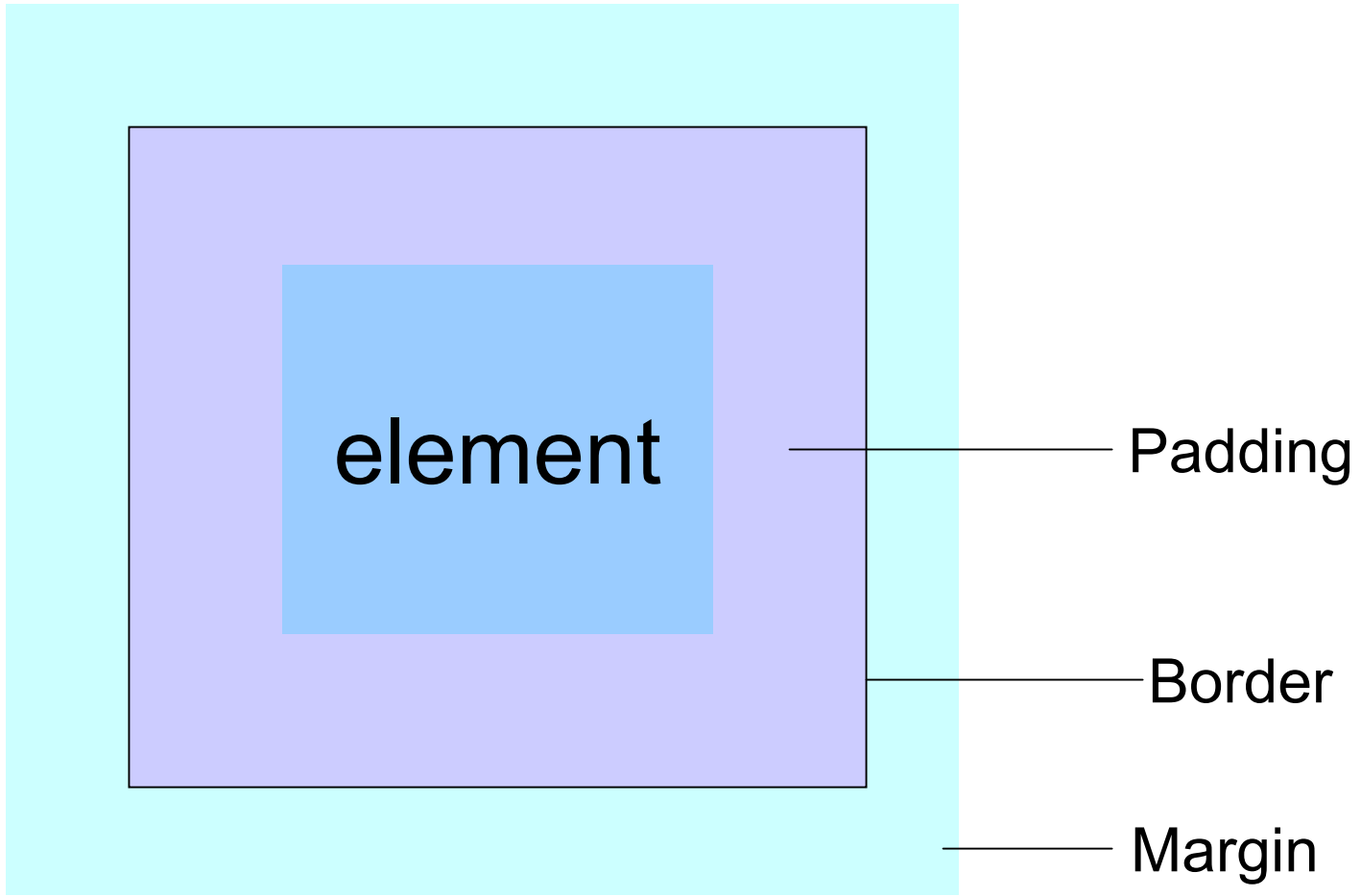
CSS: Selector Types

<code>P { color: black; }</code>	<code>/* Element Selector */</code>
<code>P, H1, H2 { color: black; }</code>	<code>/* Grouping Selector */</code>
<code>* { color: black; }</code>	<code>/* Universal Selector */</code>
<code>P.urgent, .Error { color: black; }</code>	<code>/* Class Selector */</code>
<code>#Menu { color: black; }</code>	<code>/* ID Selector */</code>
<code>*[title], A[href][title] { color: black; }</code>	<code>/* Attribute Selector */</code>
<code>A[title="home page"] { color: black; }</code>	<code>/* Exact Attribute Selector */</code>
<code>A[title~="foo"] { color: black; }</code>	<code>/* Partial Attribute Selector */</code>
<code>*[lang "en"] { color: black; }</code>	<code>/* Particular Attribute Selector */</code>
<code>UL LI UL { color: black; }</code>	<code>/* Descendant Selector */</code>
<code>P > STRONG { color: black; }</code>	<code>/* Child Selector */</code>
<code>H1 + P { color: black; }</code>	<code>/* Adjacent Sibling Selector */</code>
<code>A:hover { color: black; }</code>	<code>/* Pseudo-Class Selector */</code>
<code>P:first-letter { font-size: 200%; }</code>	<code>/* Pseudo-Element Selector */</code>

CSS: Common Declaration Properties

background	background-attachment	background-color
background-repeat	border	bottom
color	cursor	display
float	font	font-family
font-size	font-style	font-weight
height	left	letter-spacing
line-height	list-style-image	list-style-position
list-style-type	margin	overflow
padding	position	right
text-align	text-decoration	text-indent
text-transform	top	vertical-align
visibility	white-space	width
word-spacing	word-wrap	z-index

CSS: Box Model



HTML and JavaScript

- **<SCRIPT> tag**

Delineates inline code or references external code files

```
<SCRIPT TYPE="text/javascript">
```

```
    // Code goes here...
```

```
</SCRIPT>
```

```
<SCRIPT TYPE="text/javascript" SRC="code.js"></SCRIPT>
```

- **Event Attributes**

Defines event handlers for events of specific elements

```
<INPUT TYPE="Button" onClick="alert('Hi there!');" VALUE="Hi">
```

```
<IMG SRC="blue.gif"
```

```
    onMouseOver="this.src='red.gif';" onMouseOut="this.src='blue.gif';" >
```

JavaScript

- **Full, feature-rich language**
Supports all standard control mechanisms: conditionals, loops, variables, try/catch/throw, functions, "objects"
- **Very powerful**
Earlier versions were limited. Current version is not.
- **Syntactically similar to CFScript**
CFScript syntax was based on JavaScript
- **Access to most browser features/properties**
Cannot normally access local file system, etc.

Advanced DHTML Example 1

- **Bouncing Balls**

Using a combination of all DHTML technologies, this example dynamically creates and tracks the movements of an unlimited number of bouncing balls.

- Creates new content dynamically
- DOM manipulation to render animation and live data

Advanced DHTML Example 2

- Zip Code Lookup

Uses XMLHttpRequest object to retrieve XML-formatted data without requiring a screen refresh. Data is retrieved from the server in the background.

- Retrieves data from server in the background
- DOM manipulation to parse XML document

XMLHttpRequest Object Methods

Method	Description
<code>abort()</code>	Aborts the current request
<code>getResponseHeaders()</code>	Returns all sets of response headers as a string
<code>getResponseHeader("label")</code>	Returns the value of the specified response header as a string
<code>open("method", "URL"[,asyncFlag[,"username"[,"password"]])</code>	Sets various request properties of the request including URL, method (get or post), and asynchronous handling flag.
<code>send(content)</code>	Sends the request along with optional POST content
<code>setRequestHeader("label", "value")</code>	Sets a header to be sent with the request

XMLHttpRequest Object Properties

Method	Description
onreadystatechange	The event handler that will be fired on change of state
readyState	Integer indicating object status: 0 = uninitialized 1 = loading 2 = loaded 3 = interactive 4 = complete
responseText	Response from server in string format
responseXML	DOM-compatible object of response data
status	Numeric HTTP status code returned by server
statusText	HTTP status message returned by server

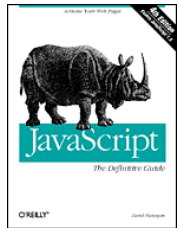
Resources: Books (1/2)



Dynamic HTML: The Definitive Reference, 2nd Edition

By Danny Goodman

<http://www.oreilly.com/catalog/dhtmlref2/index.html>



JavaScript: The Definitive Guide, 4th Edition

By David Flanagan

<http://www.oreilly.com/catalog/jscript4/index.html>

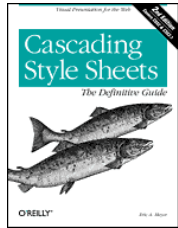


HTML & XHTML: The Definitive Guide, 5th Edition

By Chuck Musciano, Bill Kennedy

<http://www.oreilly.com/catalog/html5/index.html>

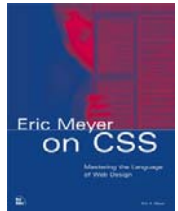
Resources: Books (2/2)



Cascading Style Sheets: The Definitive Guide, 2nd Edition

By Eric Meyer

<http://www.oreilly.com/catalog/css2/index.html>



Eric Meyer on CSS

By Eric Meyer

<http://www.ericmeyeroncss.com/>



More Eric Meyer on CSS

By Eric Meyer

<http://more.ericmeyeroncss.com/>

Resources: Online

W3C Website

<http://www.w3.org/Style/CSS/>

(CSS)

<http://www.w3.org/DOM/>

(DOM)

<http://www.w3.org/MarkUp/>

(HTML/XHTML)

css-discuss listserv

<http://www.css-discuss.org/>

css-discuss Wiki

<http://css-discuss.incutio.com/>

JavaScript Message Board

<http://www.aspmessageboard.com/forum/jscript.asp>

XMLHttpRequest() Information

<http://developer.apple.com/internet/webcontent/xmlhttpreq.html>

<http://www.xml.com/lpt/a/2005/02/09/xml-http-request.html>

Resources: Developer Tools

Web Developer Extension for Firefox and Mozilla

By Chris Pederick

<http://www.chrispederick.com/work/firefox/webdeveloper/>

LiveHTTPHeaders for Firefox and Mozilla

By David Savard

<http://livehttpheaders.mozdev.org/>

Web Development Tools built into Firefox and Mozilla

By Mozilla.org

<http://www.mozilla.org/products/firefox/>

IE DOM Inspector for Internet Explorer (not free)

By IEInspector Software, LLC

<http://www.ieinspector.com/dominspector/index.html>

Closing

- Questions?
- Download Presentation and Source Code
<http://www.evoch.com/Community/presentations.cfm>
- Contact Info
Mosh Teitelbaum
evoch, LLC
mosh.teitelbaum@evoch.com
<http://www.evoch.com/>