



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

Ajax Technology in Web Programming

Sergio Luján Mora

Ajax Technology in Web Programming

API for accessing and manipulating HTML documents

DOM

Index

- Introduction
- DOM
 - DOM Level 0
 - DOM Level 1
- How to access a web form
- BOM

Introduction

- *Document Object Model (DOM)*
 - Allows accessing HTML elements in a document →
It isn't a programming language
- *Browser Object Model (BOM)*
 - *Navigator objects*
 - Allows accessing browser's components
- There are some incompatibilities between different browsers

Introduction

- Benefits:
 - Allows client-side processing without the need of web server processing

DOM

- The *Document Object Model* is an API for HTML and XML documents
- W3C Specification
- It does two things for web developers:
 - it provides a structural representation of the document (tree structure), and
 - it defines the way that that structure is to be accessed and manipulated from script, allowing you to get at the web page as a structured group of nodes
- Essentially, it connects web pages to scripts or programming languages

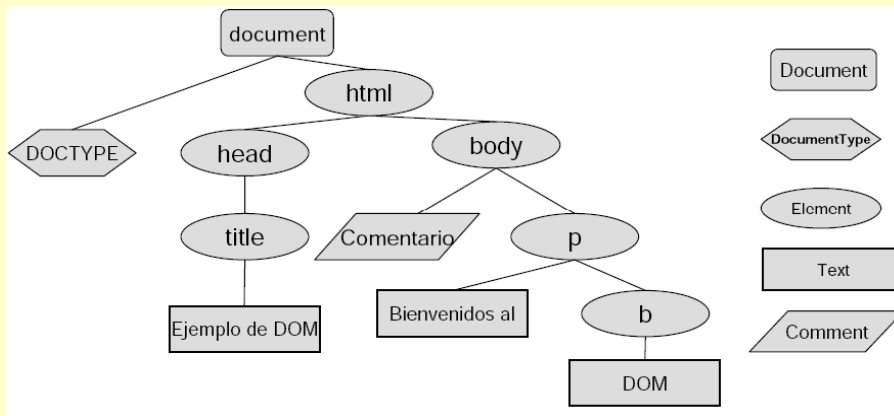
DOM

- Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01>  
<html>  
<head>  
<title>Ejemplo de DOM</title>  
</head>  
<body>  
<!-- es un ejemplo un poco simple -->  
<p style="color:red">Bienvenidos al <b>DOM</b></p>  
</body>  
</html>
```

DOM

- Nodes reflect content and structure of the document:

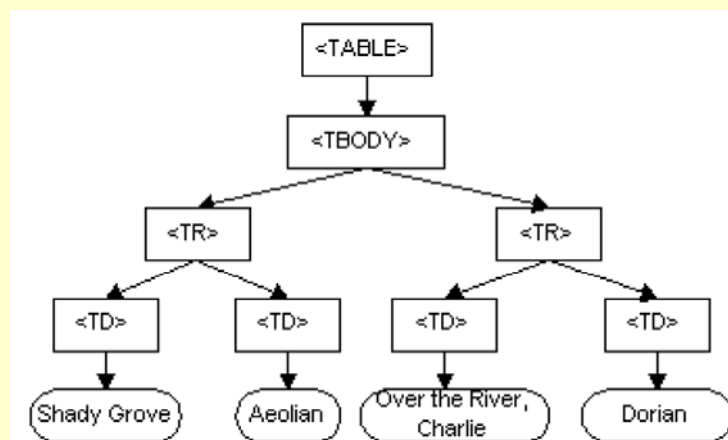


DOM

- Example:

```
<TABLE>
<TBODY>
<TR>
<TD>Shady Grove</TD>
<TD>Aeolian</TD>
</TR>
<TR>
<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>
</TR>
</TBODY>
</TABLE>
```

DOM



DOM

- DOM levels:
 - Level 0
 - Level 1
 - *Document Object Model Level 1* (1/10/1998)
 - *Document Object Model Level 1 (Second Edition)* (29/9/2000)
 - Level 2
 - *Document Object Model Level 2 Core* (13/11/2000)
 - *Document Object Model Level 2 Views*
 - *Document Object Model Level 2 Events*
 - *Document Object Model Level 2 Style*
 - *Document Object Model Level 2 Traversal and Range*
 - *Document Object Model Level 2 HTML*
 - Level 3
 - *Document Object Model Level 3 Core* (7/4/2004)
 - *Document Object Model Level 3 Load and Save*
 - *Document Object Model Level 3 Validation*

DOM

- What does your user agent claim to support?
 - <http://www.w3.org/2003/02/06-dom-support.html>

W3C Document Object Model Recommendations: support claims - Windows Internet Explorer

http://www.w3.org/2003/02/06-dom-support.html

DOM Module	DOM Level 1	DOM Level 2	DOM Level 3
Core: basic methods (Level 1 and 2) and extensions for XML Namespaces (Level 2 only)	-	2000	2009
XML: extensions for XML 1.0	1999	2000	2009
HTML: extensions for HTML 4.0x (Level 1 and 2) and support of XHTML 1.0 (Level 2 only)	supported	2003	N/A
Views: used with the Level 2 CSS and UIEvents DOM modules	N/A	2000	N/A
StyleSheets: association between a style sheet and a document	N/A	2000	N/A
CSS: extensions for cascading style sheets	N/A	2000	N/A
CSS2: extensions for Cascading Style Sheets Level 2	N/A	2000	N/A
Events: generic events system	N/A	2000	N/A
UIEvents: basic user interface events	N/A	2000	N/A
MouseEvents: mouse device events	N/A	2000	N/A
MutationEvents: events for mutations in a DOM tree	N/A	2000	N/A
HTMLEvents: HTML 4.01 events	N/A	2000	N/A
Range: extensions to manipulate a range in a DOM tree	N/A	2000	N/A
Traversal: Alternative traversal methods of a DOM tree	N/A	2000	N/A
LS: Loading a document into a DOM tree	N/A	N/A	2009
LS-Async: Asynchronous loading of a document into a DOM tree	N/A	N/A	2009
Validation: Schema-oriented modification of a DOM tree	N/A	N/A	2009

This page *does not*

W3C Document Object Model Recommendations: support claims - Mozilla Firefox

http://www.w3.org/2003/02/06-dom-support.html

DOM Module	DOM Level 1	DOM Level 2	DOM Level 3
Core: basic methods (Level 1 and 2) and extensions for XML Namespaces (Level 2 only)	-	supported	2009
XML: extensions for XML 1.0	supported	supported	2009
HTML: extensions for HTML 4.0x (Level 1 and 2) and support of XHTML 1.0 (Level 2 only)	supported	supported	N/A
Views: used with the Level 2 CSS and UIEvents DOM modules	N/A	supported	N/A
StyleSheets: association between a style sheet and a document	N/A	supported	N/A
CSS: extensions for cascading style sheets	N/A	supported	N/A
CSS2: extensions for Cascading Style Sheets Level 2	N/A	supported	N/A
Events: generic events system	N/A	supported	N/A
UIEvents: basic user interface events	N/A	supported	N/A
MouseEvents: mouse device events	N/A	supported	N/A
MutationEvents: events for mutations in a DOM tree	N/A	2000	N/A
HTMLEvents: HTML 4.01 events	N/A	supported	N/A
Range: extensions to manipulate a range in a DOM tree	N/A	supported	N/A
Traversal: Alternative traversal methods of a DOM tree	N/A	2000	N/A
LS: Loading a document into a DOM tree	N/A	N/A	2009
LS-Async: Asynchronous loading of a document into a DOM tree	N/A	N/A	2009
Validation: Schema-oriented modification of a DOM tree	N/A	N/A	2009

Terminado

DOM Level 0

- The term "DOM Level 0" refers to a mix (not formally specified) of HTML document functionalities.
- Offered from Netscape Navigator version 3.0 and Microsoft Internet Explorer version 3.0.
- It is also a set of APIs which are not a part of any W3C DOM specification, but are implemented across several browsers → No standard.

DOM Level 0

- `innerHTML`:
 - Sets or retrieves the HTML between the start and end tags of the object.
 - For example, the entire contents of the document body can be deleted by:

```
// Replaces body content with an empty string.  
document.body.innerHTML = "";
```

DOM Level 0

- As there is no public specification for this property, implementations differ widely.
- It should never be used to write parts of a table—W3C DOM methods should be used for that—though it can be used to write an entire table or the contents of a cell.

DOM Level 0

- Example: text changes when cursor mouse is moved

```
<p onmouseover="this.innerHTML='<b>Mouse  
out to change back.</b>'"  
  onmouseout="this.innerHTML='<i>Mouse  
over again to change.</i>'">  
<i>Mouse over this text to change it.</i>  
</p>
```

DOM Level 1

- Two parts:
 - **Core**: a low-level set of fundamental interfaces that can represent any structured document, as well as defining extended interfaces for representing an XML document.
 - **HTML**: provides additional, higher-level interfaces that are used with the fundamental interfaces to provide a more convenient view of an HTML document.

DOM Level 1

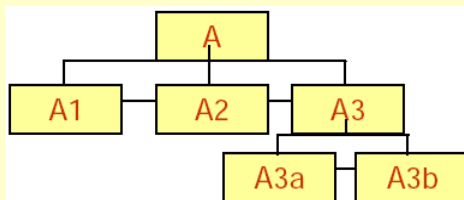
- Every node is a type of `Node`, but there are different subtypes: `Document`, `DocumentType`, `Element`, `Text`, `Comment`, ...
- Attributes are nodes of type `Attr`, but they are not in the tree that represents the document (attribute value is retrieved from a method of the node)

DOM Level 1

- Every node has a set of properties that link to the “relatives”:
 - childNodes
 - firstChild
 - lastChild
 - parentNode
 - nextSibling
 - prevSibling

DOM Level 1

- Example:



A.firstChild = A1
A.lastChild = A3
A.childNodes.length = 3
A.childNodes[0] = A1
A.childNodes[1] = A2
A.lastChild.firstChild = A3a
A3b.parentNode.parentNode = A
A1.nextSibling = A2
A3.prevSibling = A2
A3.nextSibling = null

DOM Level 1

- `getElementById("elementID")`
 - `var element = document.getElementById("myTable")`
- `getElementsByTagName("tagName")`:
 - `var images = document.getElementsByTagName("img")`
 - The special value "*" matches all tags.
- `getAttribute("attrName")`:
 - `var element = document.getElementById("myTable").getAttribute("width");`

DOM Level 1

```
// HTML:  
// <div id="d"><p>Content</p>  
// <p>Further Elaborated</p>  
// </div>  
  
d = document.getElementById("d");  
alert(d.innerHTML);
```

DOM Level 1

- **createElement("tagName")**
 - `var element = document.createElement("tbody");`
- **appendChild(element)**
 - `element.appendChild(newChild);`
- **replaceChild(element, element)**
 - `element.replaceChild(newElement, oldElement);`
- **setAttribute(name, value)**
 - `document.getElementById("myTable").setAttribute("width", 300);`

How to access a web form

- **Three ways:**
 - `document.forms[n]`
 - `document.forms["miForm"]`
 - `document.miForm`
- **General syntax for accessing a property of a control in a form:**
 - `document.nForm.nControl.propiedad`

How to access a web form

- **Checkbox:**
 - checked
 - defaultChecked
 - value
- **Radio button:**
 - checked
 - defaultChecked
 - value

How to access a web form

- **Text field, password field and text area:**
 - defaultValue
 - value
- **Selection list:**
 - length
 - options → selected, text, value
 - selectedIndex

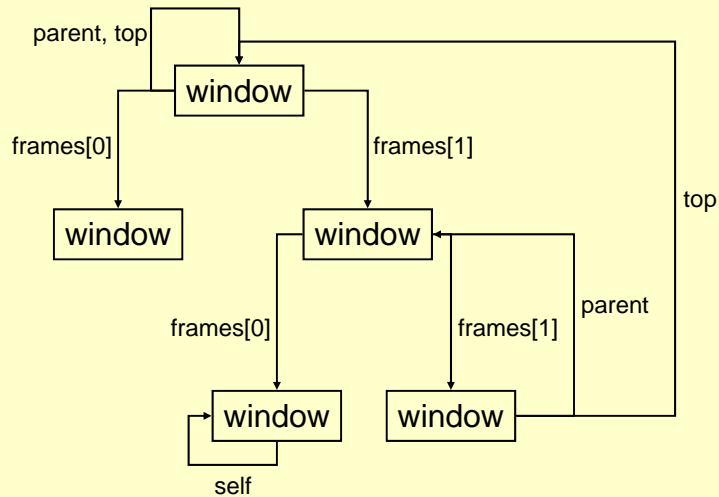
BOM

- *Browser Object Model*
- Under specification process by W3C:
 - Window Object 1.0 W3C Working Draft 07 April 2006
- Objects:
 - window
 - history
 - location
 - navigator
- Root object: window (it can be omitted)
 - `window.navigator.appName`
 - `navigator.appName`

Object window

- Main object
- Represents a window (body) or a frame (frameset)
- Properties (some are objects):
 - document, frames, history, location
 - parent, self, top
 - status **and** defaultStatus

Object window



Object window

- Each frame is represented as a window object
- `self` property retrieves a reference to the current window or frame
- `parent` property retrieves the parent of the window in the object hierarchy
- `top` property retrieves the topmost ancestor window, which is its own parent
- If `self` is equal to `self.parent`, then the current window doesn't have a parent, meaning it is the topmost window

Object window

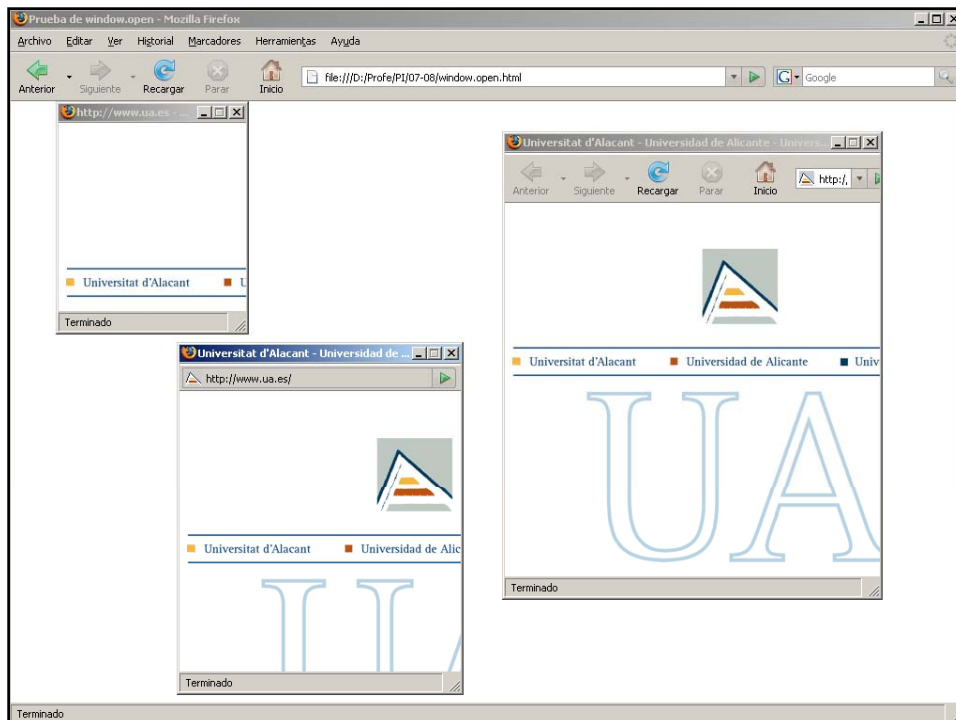
- **Methods:**
 - `alert(mensaje)`
 - `clearInterval(intervalID)`
 - `clearTimeout(timerID)`
 - `confirm(mensaje) → Ok(true)/Cancel(false)`
 - `prompt(mensaje[, valorInicial])`
 - `close`
 - `open(URL, nombre[, características])`
 - `setInterval(expresión, milisegundos)`
 - `setTimeout(expresión, milisegundos)`

Object window

Parámetro	Tipo valor	Descripción
toolbar	boolean	display a toolbar
location	boolean	display the location text box
directories	boolean	display the special link buttons
status	boolean	display a status bar
menubar	boolean	display the menus at the top of the window
scrollbars	boolean	display scrollbars if the document is larger than the window
resizable	boolean	allow the window to be resized
width	integer	the width of the window (in pixels)
height	integer	the height of the window (in pixels)
top	integer	the top position of the window (in pixels)
left	integer	the left position of the window (in pixels)

Object window

```
<html>
<head>
<title>Prueba de window.open</title>
<script type="text/javascript">
window.open("http://www.ua.es/", "ventana1", "width=200,
height=200, toolbar=no, menubar=no, location=no,
directories=no");
window.open("http://www.ua.es/", "ventana2", "width=300,
height=300, location=yes");
window.open("http://www.ua.es/", "ventana3", "width=400,
height=400, menubar=no, toolbar=yes, location=yes");
</script>
</head>
<body>
</body>
</html>
```



Object history

- Array of visited URLs in the current window
- Methods:
 - `back()`: previous visited page
 - `forward()`: following page
 - `go(n)`: go to page 'n' in the list of URLs

Object location

- Holds the URL of the current page shown in the window (or frame)
- Properties:
 - `host`, `hostname`, `href`, `pathname`, `port`, `protocol`, `search`, `hash`
- Methods:
 - `reload()`, `replace()`

Object navigator

- Information about the browser (version, operating system, etc.)
- Properties:
 - appCodeName, appName, appVersion, platform

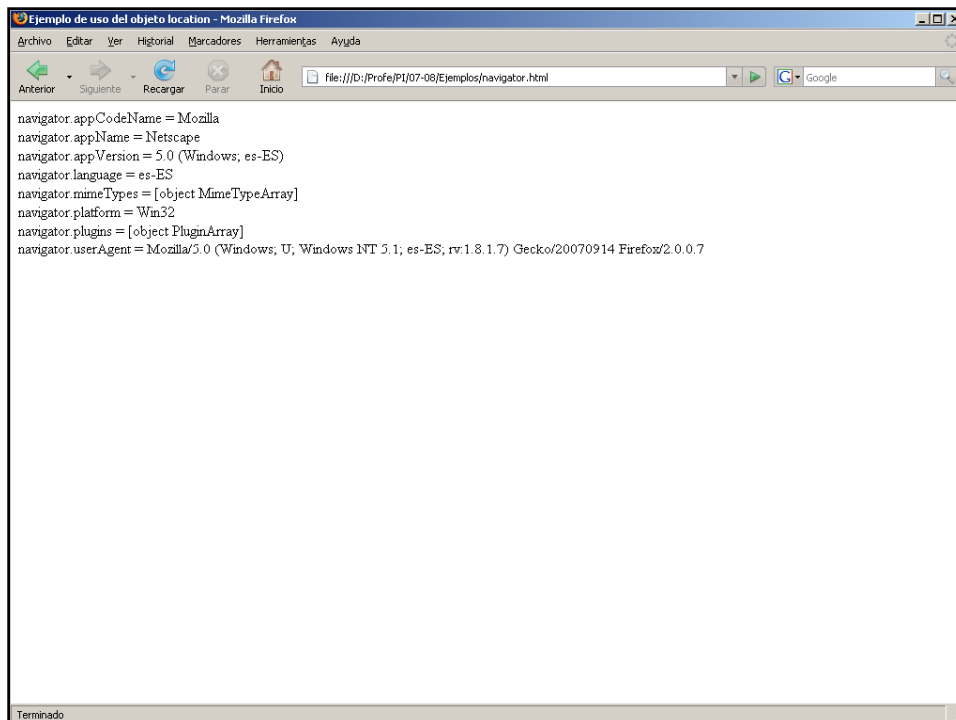
Object navigator

```
<html>
<head>
<title>Ejemplo de uso del objeto location</title>
</head>
<body>
<p>
<script type="text/javascript">
document.writeln("navigator.appCodeName = " + navigator.appCodeName);
document.writeln("<br />");
document.writeln("navigator.appName = " + navigator.appName);
document.writeln("<br />");
document.writeln("navigator.appVersion = " + navigator.appVersion);
document.writeln("<br />");
document.writeln("navigator.language = " + navigator.language);
document.writeln("<br />");
document.writeln("navigator.mimeTypes = " + navigator.mimeTypes);
document.writeln("<br />");
```

Ajax Technology in Web Programming

Object navigator

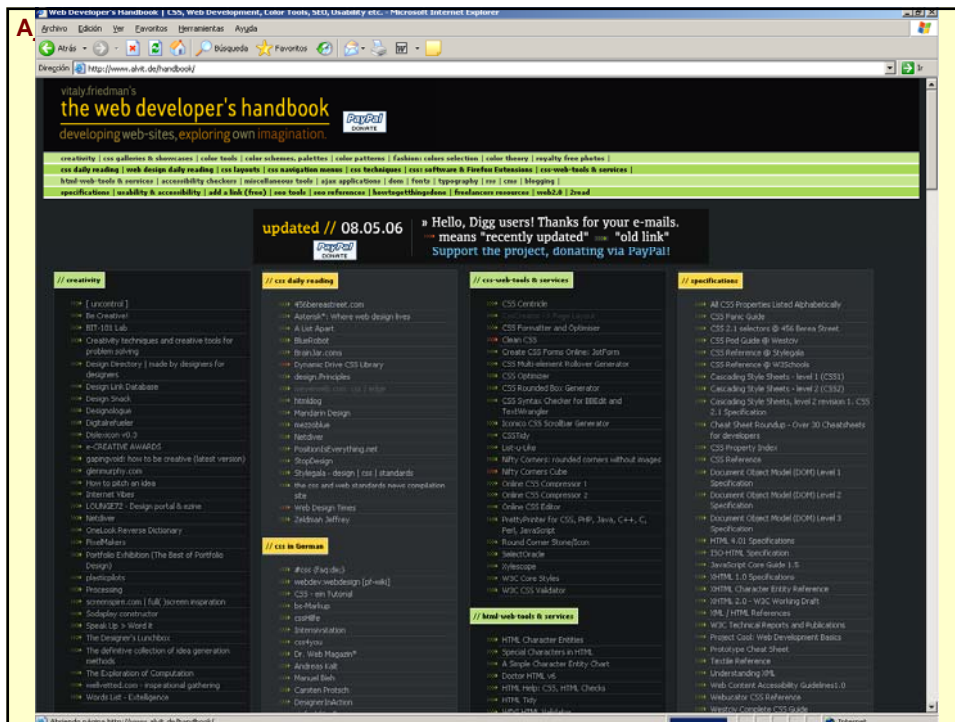
```
document.writeln("navigator.platform = " + navigator.platform);  
document.writeln("<br />");  
document.writeln("navigator.plugins = " + navigator.plugins);  
document.writeln("<br />");  
document.writeln("navigator.userAgent = " + navigator.userAgent);  
document.writeln("<br />");  
</script>  
</p>  
</body>  
</html>
```



Ajax Technology in Web Programming

More Information

- <http://www.alvit.de/handbook/>
- <http://www.w3schools.com/>
- <http://www.w3.org/>



W3Schools Online Web Tutorials - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio http://www.w3schools.com

W3 Schools

The best things in life are free

THE LARGEST WEB DEVELOPER'S SITE ON THE NET

Full Web Building Tutorials - All Free

At W3Schools you will find all the Web-building tutorials you need, from basic HTML and XHTML to advanced XML, SQL, Database, Multimedia and WAP.

Select your tutorial from the menu on the left

Full Web Building References

Our references cover all Web-building technologies, including W3C standards like HTML, XHTML, CSS, XML, and other technologies like JavaScript, PHP, ASP, SQL and much more.

Try It Yourself On-Line Examples

At W3Schools you will find thousands of cut-and-paste examples. With our on-line HTML editor you can edit the examples and experiment with the code on-line.

Quick and Easy Learning

Because time is valuable, we deliver quick and easy learning. At W3Schools, you can study everything you need to learn in an accessible and handy format.

"Never increase, beyond what is necessary, the number of entities required to explain anything" — William of Ockham (1285-1349)

Where to Start

what does a Web developer have to know?

W3Schools will answer this, and help you become a professional Web developer, well prepared for the future.

For the beginner: [Go to our Web Building Primer](#)

For the developer: [Go to our Web Building Tutorial](#)

Internet Joke

Profrase: "I used to download this tutorial. But I found a better book..."

HTML Tutorials

- Learn HTML
- Learn XHTML
- Learn CSS
- Learn TCP/IP

XML Tutorials

- Learn XML
- Learn XSL
- Learn XSLT
- Learn XSL-FO
- Learn XPath
- Learn XQuery
- Learn XLink
- Learn XPointer
- Learn XDT
- Learn Schema
- Learn XSL-DM
- Learn XForms
- Learn SOAP
- Learn WSDL
- Learn RDF
- Learn RSS
- Learn WAP
- Learn Web Services

Browser Scripting

- Learn JavaScript
- Learn HTML DOM
- Learn XHTML
- Learn VBScript
- Learn AJAX
- Learn EXX
- Learn WMLScript

Server Scripting

- Learn SQL
- Learn asp
- Learn ASP
- Learn PHP

.NET (dotnet)

- NET Microsoft
- NET ASP
- NET Mobile

Multimedia

- Learn Media
- Learn SMIL
- Learn SVG
- Learn Flash

Web Building

- Web W3C
- Web Browsers
- Web Quality
- Web Semantics
- Web Careers
- Web Hosting
- Web Certification

REFERENCES

- HTML 4.01
- XHTML 1.0
- CSS 2.0
- JAVASCRIPT
- HTML DOM
- PHP 5.1
- XSLT 1.0
- XPath 1.0
- XSL-FO
- XML 1.1
- ASCII Reference
- Entity Reference
- HTML Color Names

EXAMPLES

- HTML Examples
- CSS Examples
- XML Examples
- DOM Examples
- W3C Examples
- JavaScript Examples
- XML Examples
- ASP Examples
- PHP Examples
- ASP.NET Examples
- SQL Examples

QUIZZES

- HTML Quiz
- XHTML Quiz
- CSS Quiz
- XML Quiz
- JavaScript Quiz
- SQL Quiz
- PHP Quiz
- ASP Quiz

CERTIFICATION

- HTML Certification
- XML Certification
- ASP Certification

QUICK STARTERS

- Microsoft HTML
- Microsoft CSS
- Microsoft JavaScript
- Microsoft VBScript

VALIDATION

- Validate HTML
- Validate CSS
- Validate XHTML

Document Object Model (DOM) Specifications - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio http://www.w3.org/DOM/DOMTR

W3C Architecture domain

About DOM DOM Activity statement
Technical Reports - Technical Materials
Test Suites - FAQ - Mailing List
Members only resource: DOM Working Group

Document Object Model (DOM) Technical Reports

See also [other W3C Technical Reports](#).

Table of contents

- [Document Object Model Level 1](#)
- [Document Object Model Level 2](#)
- [Document Object Model Level 3](#)
- [Others](#)

Document Object Model Level 1

[Document Object Model Level 1 \(W3C Recommendation\)](#)
1 October 1998, Vidar Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Robert Sutor, Chris Wilson, Lauren Wood
[\[errata\]](#)

[Document Object Model Level 1 \(Second Edition\) \(W3C Working Draft\)](#)
29 September 2000, Vidar Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Robert Sutor, Chris Wilson, Lauren Wood

Document Object Model Level 2

[Document Object Model Level 2 Core \(W3C Recommendation\)](#)
13 November 2000, Arnaud Le Hors, Philippe Le Hégaré, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion, Steve Byrne
[\[errata\]](#)

[Document Object Model Level 2 Views \(W3C Recommendation\)](#)
13 November 2000, Arnaud Le Hors, Laurence Cable
[\[errata\]](#)

[Document Object Model Level 2 Events \(W3C Recommendation\)](#)
13 November 2000, Tom Fritley
[\[errata\]](#)

[Document Object Model Level 2 Style \(W3C Recommendation\)](#)
13 November 2000, Chris Wilson, Philippe Le Hégaré, Vidar Apparao
[\[errata\]](#)

[Document Object Model Level 2 Traversal and Range \(W3C Recommendation\)](#)
13 November 2000, Joe Kesselman, Jonathan Robie, Mike Champion, Peter Sharpe, Vidar Apparao, Lauren Wood
[\[errata\]](#)

[Document Object Model Level 2 HTML \(W3C Recommendation\)](#)
9 January 2003, Johnny Sterback, Philippe Le Hégaré, Arnaud Le Hors, Chris Wilson, Ian Jacobs, Mike Champion, Scott Isaacs, Vidar Apparao
[\[errata\]](#) [\[issues list\]](#) [\[implementation report\]](#) [\[C6 Disclosure\]](#)