Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

Ajax Technology in Web Programming

Sergio Luján Mora

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

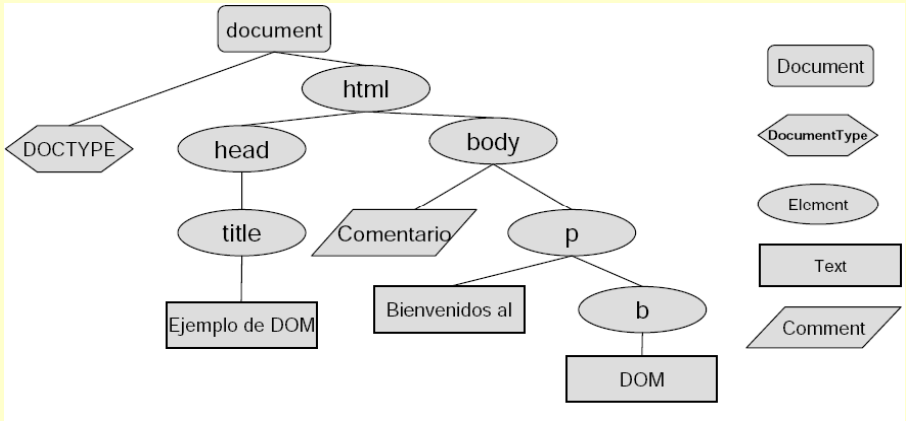
DOM

- 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:

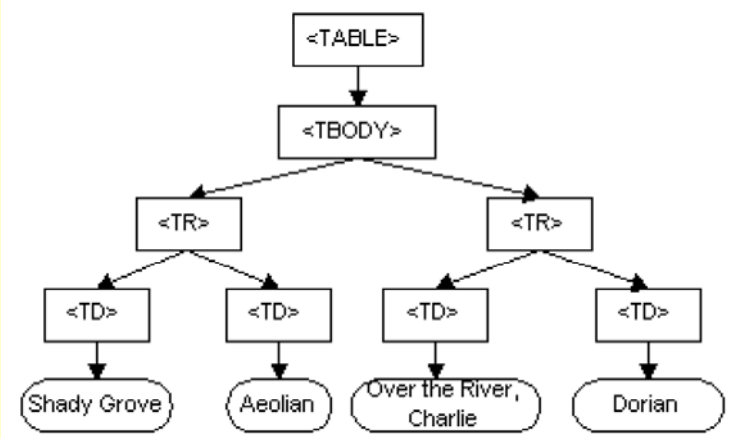


DOM

- 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*

Ajax Technology in Web Programming

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	2004
XML: extensions for XML 1.0	1998	2000	2004
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	2004
LS-Async: Asynchronous loading of a document into a DOM tree	N/A	N/A	2004
Validation: Schema-oriented modification of a DOM tree	N/A	N/A	2004

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	2004
XML: extensions for XML 1.0	supported	supported	2004
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	2004
LS-Async: Asynchronous loading of a document into a DOM tree	N/A	N/A	2004
Validation: Schema-oriented modification of a DOM tree	N/A	N/A	2004

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

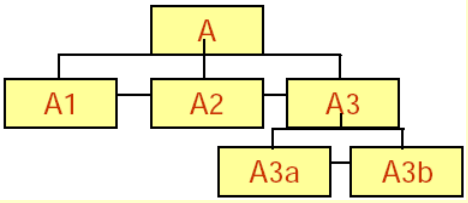
- Every node is a type of `Node`, but there are different subtypes: `Document, DocumentType, Element, Text,Comment, …`
- Attributes are nodes of type `Attr`, but they are not in the tree that represents the document (attribute value is retrieved from a method of the node)

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”);`

Ajax Technology in Web Programming

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`

Ajax Technology in Web Programming

How to access a web form

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

BOM

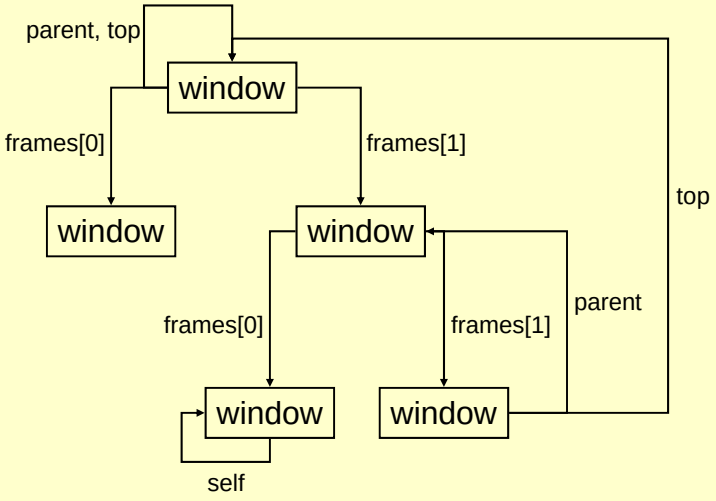
- *Browser Object Model*
- Under specification process by W3C:
  - Window Object 1.0 W3C Working Draft 07 April 2006
- Objects:
  - `window`
  - `history`
  - `location`
  - `navigator`
- Root object: `window` (it can be omitted)

```
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)`

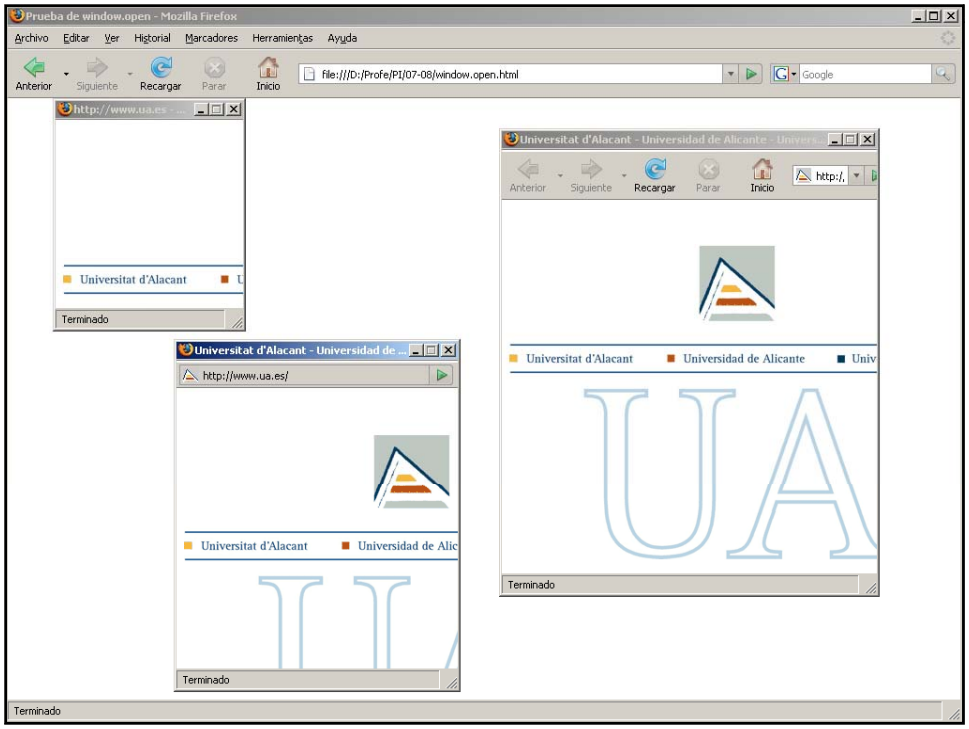
Ajax Technology in Web Programming

Object window

Parámeto	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

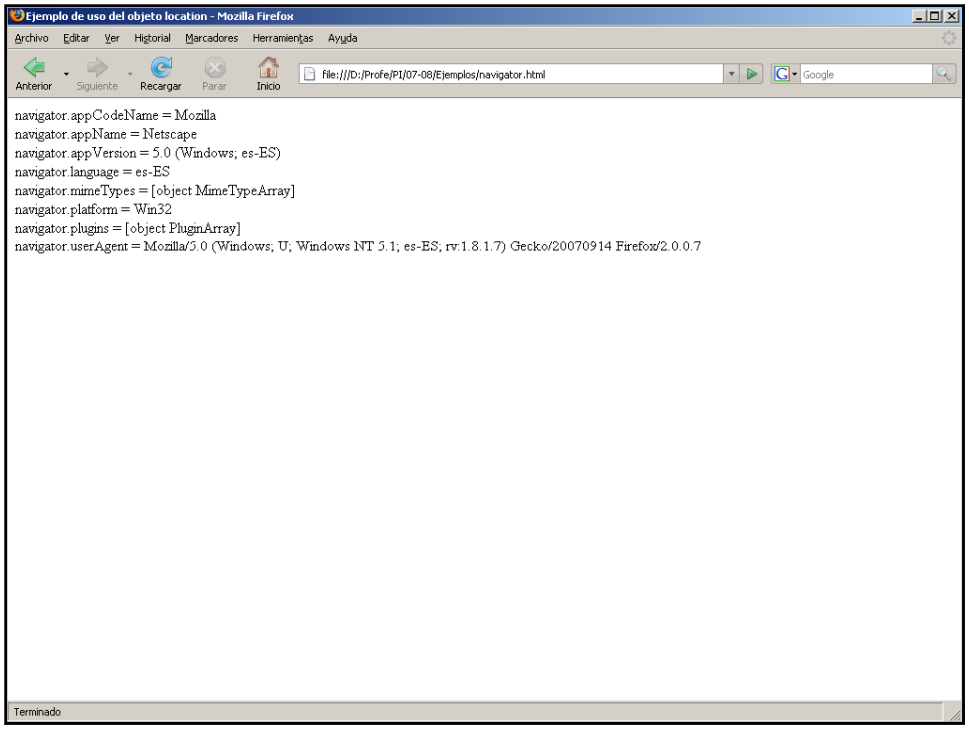
- Information about the browser (version, operating system, etc.)
- Properties:
  - appCodeName, appName, appVersion, platform

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 />");
```

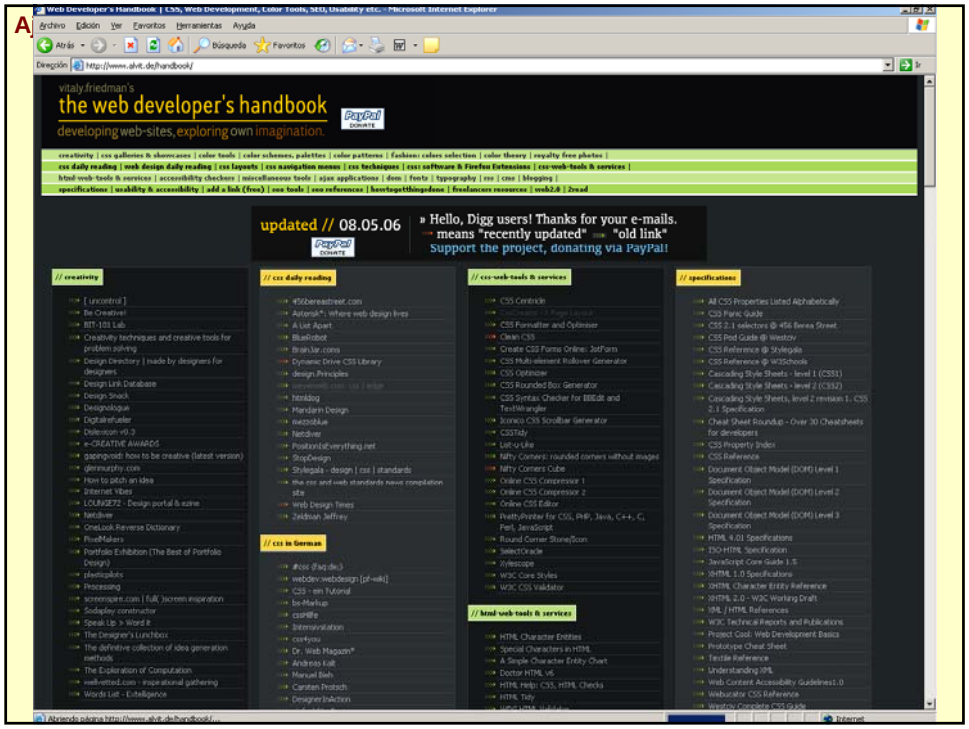
Object navigator

```
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>
```

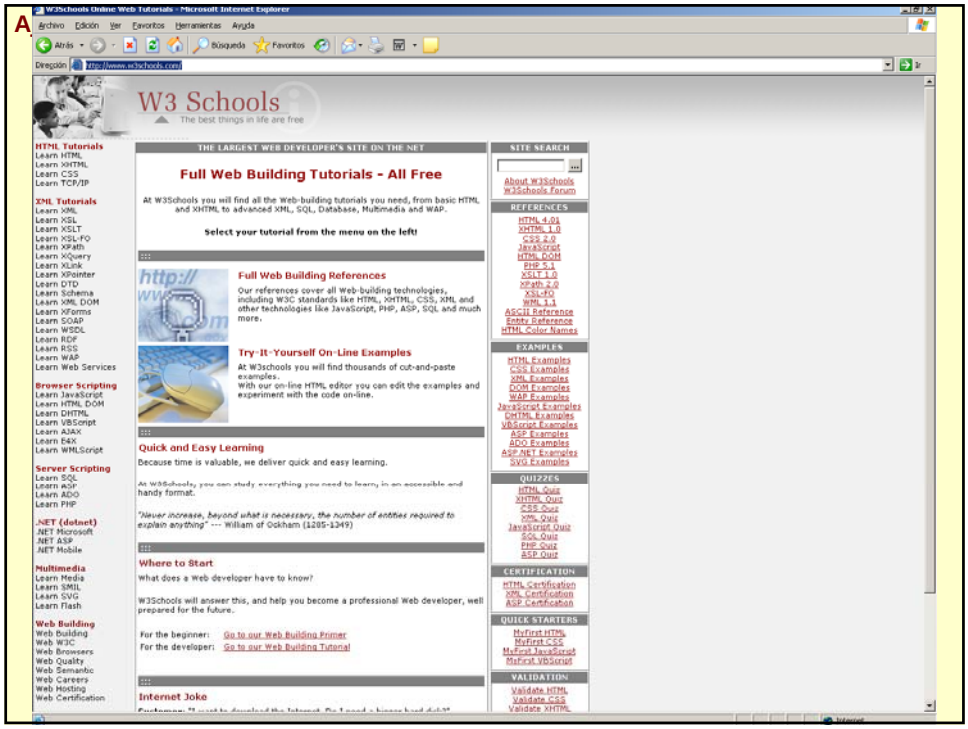


More Information

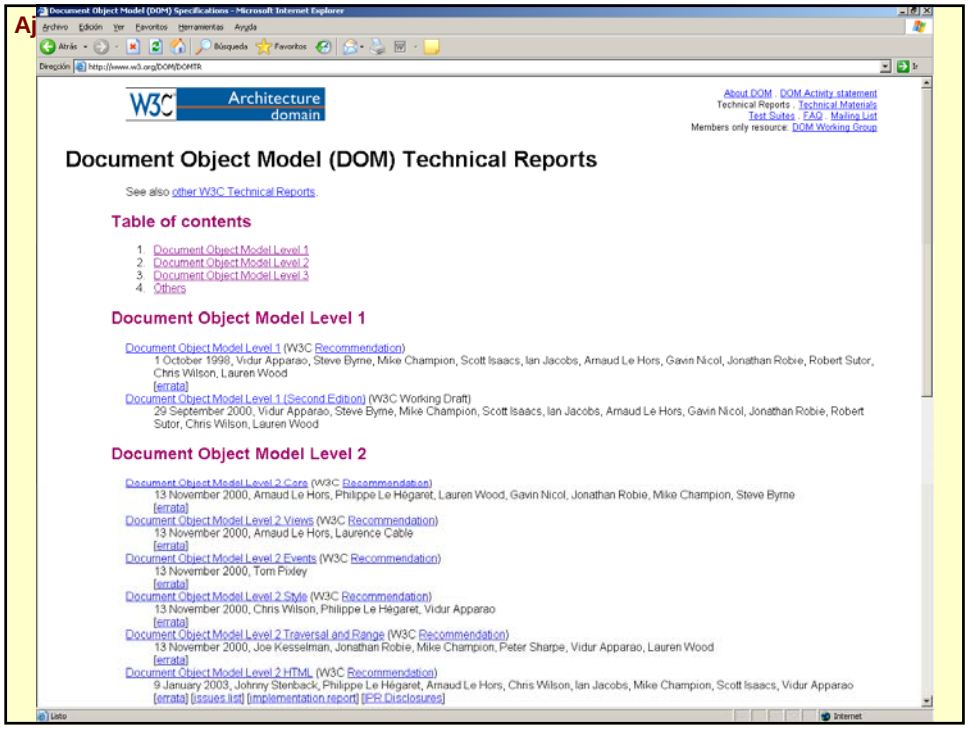
- http://www.alvit.de/handbook/
- http://www.w3schools.com/
- http://www.w3.org/



A


W3Schools Online Web Tutorials - Microsoft Internet Explorer
W3 Schools
The best things in life are free
Full Web Building Tutorials - All Free
Full Web Building References
Try-It-Yourself On-Line Examples
Quick and Easy Learning
Where to Start
Internet Joke

Aj


Document Object Model (DOM) Specifications - Microsoft Internet Explorer
Architecture domain
Document Object Model (DOM) Technical Reports
Table of contents
Document Object Model Level 1
Document Object Model Level 2