Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

Modelo de objetos de documento

Programación en Internet
Curso 2009-2010

Índice

Programación en Internet – Curso 2009-2010

Introducción

- *Document Object Model (DOM)*
  - Permite acceder a los elementos HTML de un documento → No confundir con los lenguajes de *script*
- *Browser Object Model (BOM)*
  - *Navigator objects*
  - Permite acceder a elementos del navegador
- Incompatibilidades entre distintos navegadores
- Jerarquía objectos ⇔ Estructura documento

Programación en Internet – Curso 2009-2010

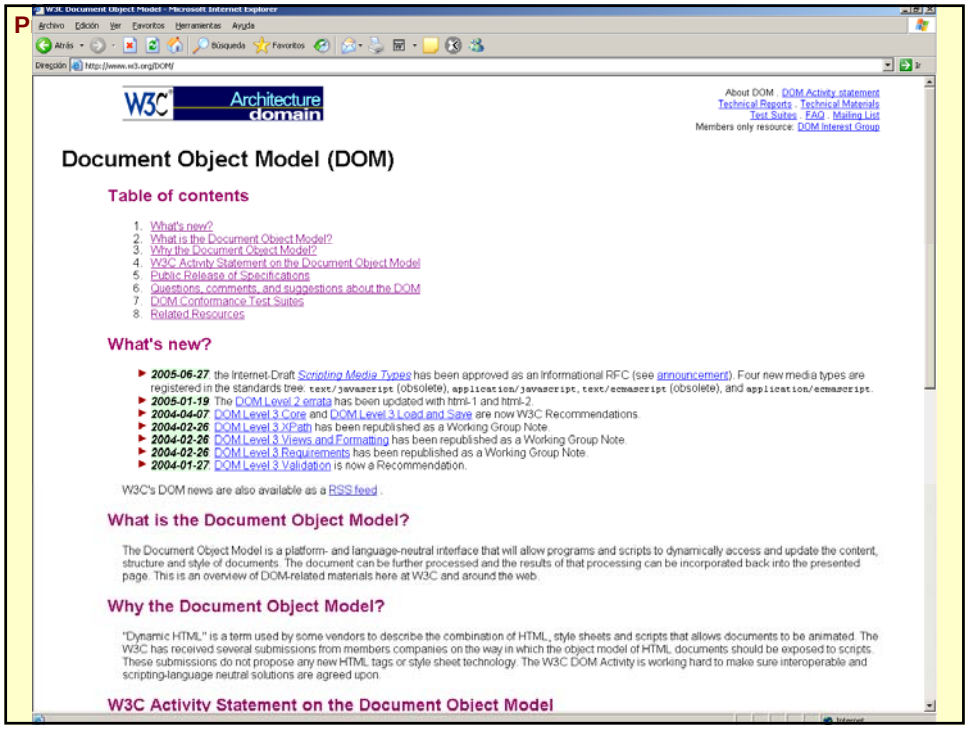
Introducción

- Ventajas:
  - Permite manipular las páginas web en el cliente sin necesidad de procesamiento en el lado del servidor

Programación en Internet – Curso 2009-2010

DOM

- *Document Object Model*
- Especificación de W3C
- Mecanismo para que desde cualquier lenguaje de programación se pueda manipular un documento XML o HTML
- Básicamente:
  - Crea una estructura lógica para el documento (llamada “árbol nodal”)
  - Proporciona un método para leer y manipular el documento



Programación en Internet – Curso 2009-2010

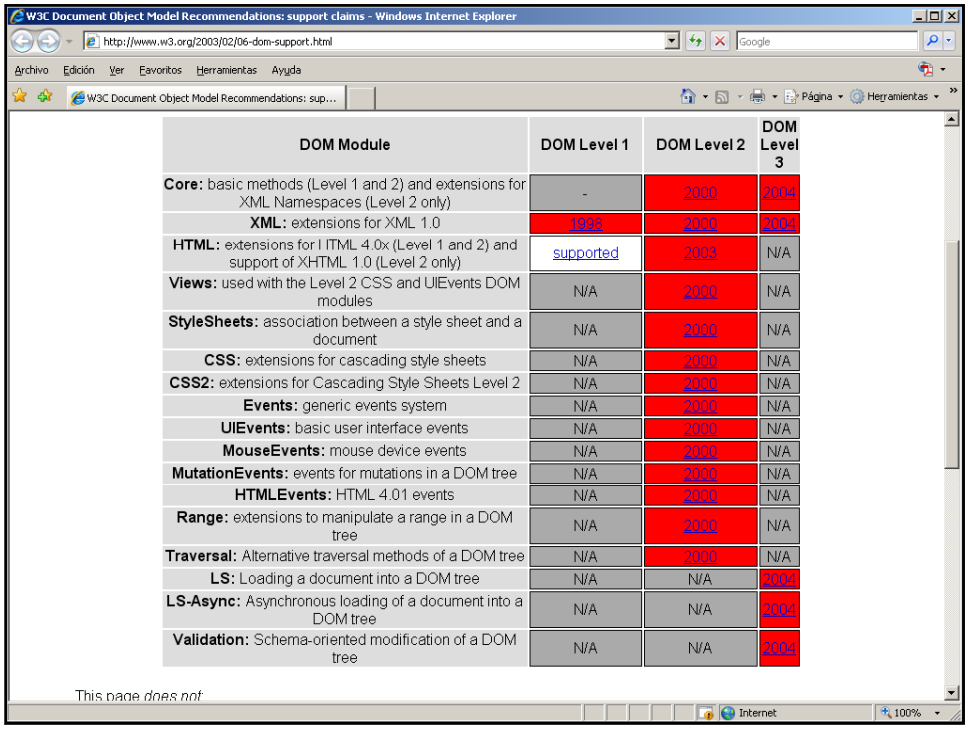
DOM

- Versiones de DOM:
  - Nivel 0
  - Nivel 1
    - *Document Object Model Level 1* (1/10/1998)
    - *Document Object Model Level 1 (Second Edition)* (29/9/2000)
  - Nivel 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*
  - Nivel 3
    - *Document Object Model Level 3 Core* (7/4/2004)
    - *Document Object Model Level 3 Load and Save*
    - *Document Object Model Level 3 Validation*

Programación en Internet – Curso 2009-2010

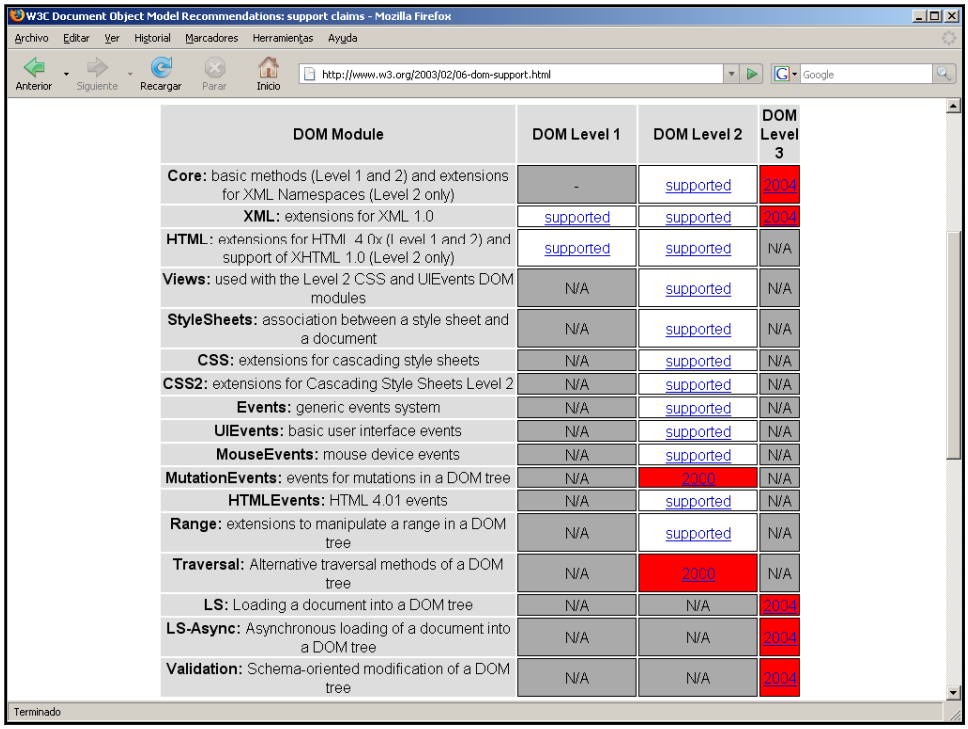
DOM

- What does your user agent claim to support?
  - 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*



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

Programación en Internet – Curso 2009-2010

DOM

- DOM trata un documento HTML (XML) como una jerarquía de nodos
  - Distintos nodos y cada uno tiene permitido solo ciertos hijos de ciertos tipos de nodo
  - Existen algunos tipos de nodo que son hoja → No pueden tener hijos

Programación en Internet – Curso 2009-2010

DOM

- Document:
  - Element (máximo uno)
  - ProcessingInstruction
  - Comment
  - DocumentType (máximo uno)
- DocumentFragment:
  - Element
  - ProcessingInstruction
  - Comment
  - Text
  - CDATASection
  - EntityReference

Programación en Internet – Curso 2009-2010

DOM

- DocumentType: sin hijos
- EntityReference:
  - Element
  - ProcessingInstruction
  - Comment
  - Text
  - CDATASection
  - EntityReference

Programación en Internet – Curso 2009-2010

DOM

- Element:
  - Element
  - Text
  - Comment
  - ProcessingInstruction
  - CDATASection
  - EntityReference
- Attr:
  - Text
  - EntityReference

Programación en Internet – Curso 2009-2010

DOM

- ProcessingInstruction: sin hijos
- Comment: sin hijos
- Text: sin hijos
- CDATASection: sin hijos
- Entity:
  - Element
  - ProcessingInstruction
  - Comment
  - Text
  - CDATASection
  - EntityReference
- Notation: sin hijos

Programación en Internet – Curso 2009-2010

DOM

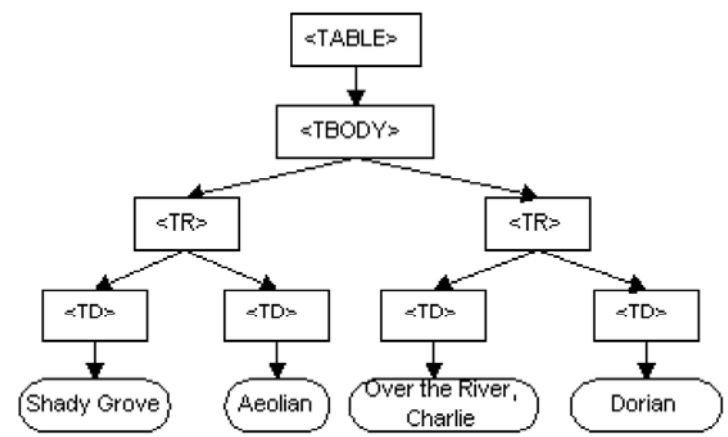
- Ejemplo:

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

Fuente: Document Object Model (DOM) Level 1 Specification, W3C Recommendation 1 October, 1998

Programación en Internet – Curso 2009-2010

DOM



Fuente: Document Object Model (DOM) Level 1 Specification, W3C Recommendation 1 October, 1998

Programación en Internet – Curso 2009-2010

DOM

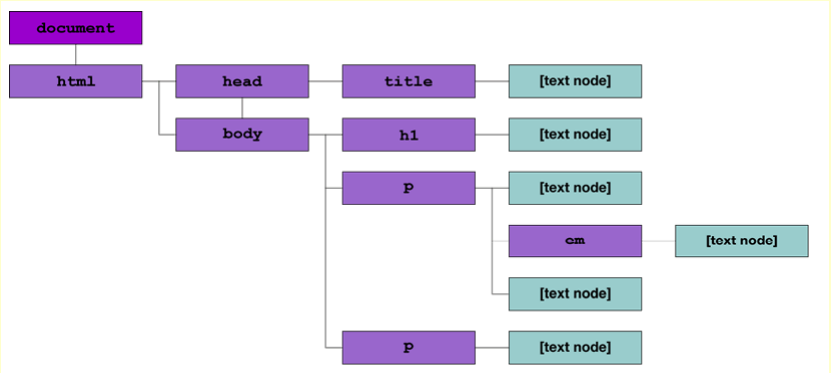
- Ejemplo:

```
<html>
<head>
  <title>This is a Document!</title>
</head>
<body>
  <h1>This is a header!</h1>
  <p id="excitingText">This is a paragraph!
   <em>Excitement</em>!</p>
  <p>This is also a paragraph, but it's not nearly as
   exciting as the last one.</p>
</body>
</html>
```

Fuente: DevOpera, 46: Traversing the DOM, 3 February 2009

Programación en Internet – Curso 2009-2010

DOM



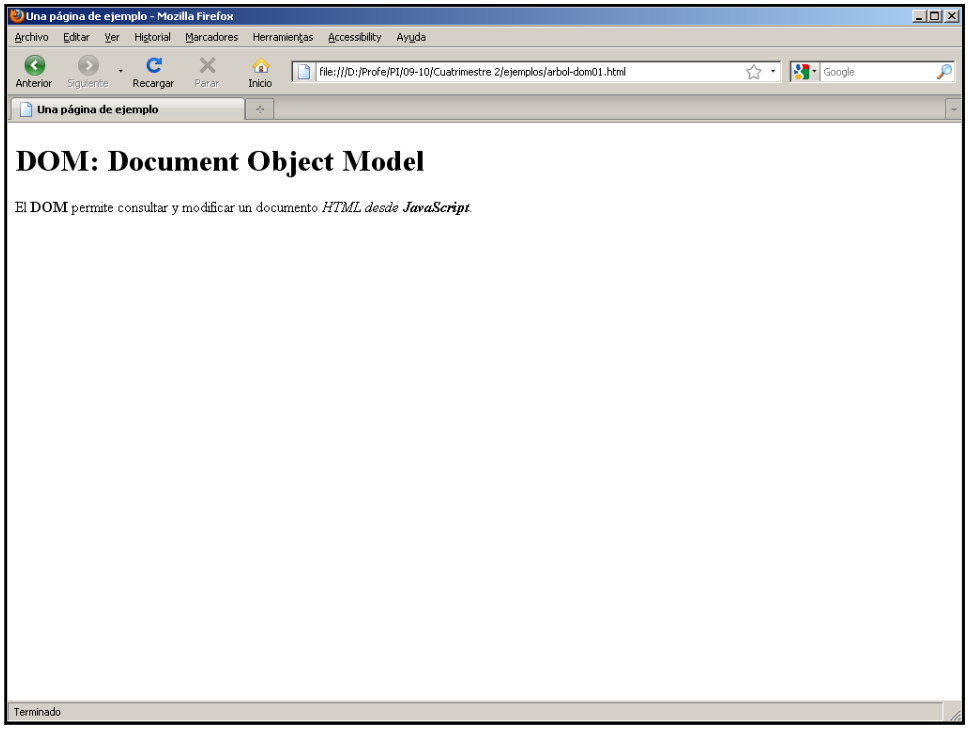
Fuente: DevOpera, 46: Traversing the DOM, 3 February 2009

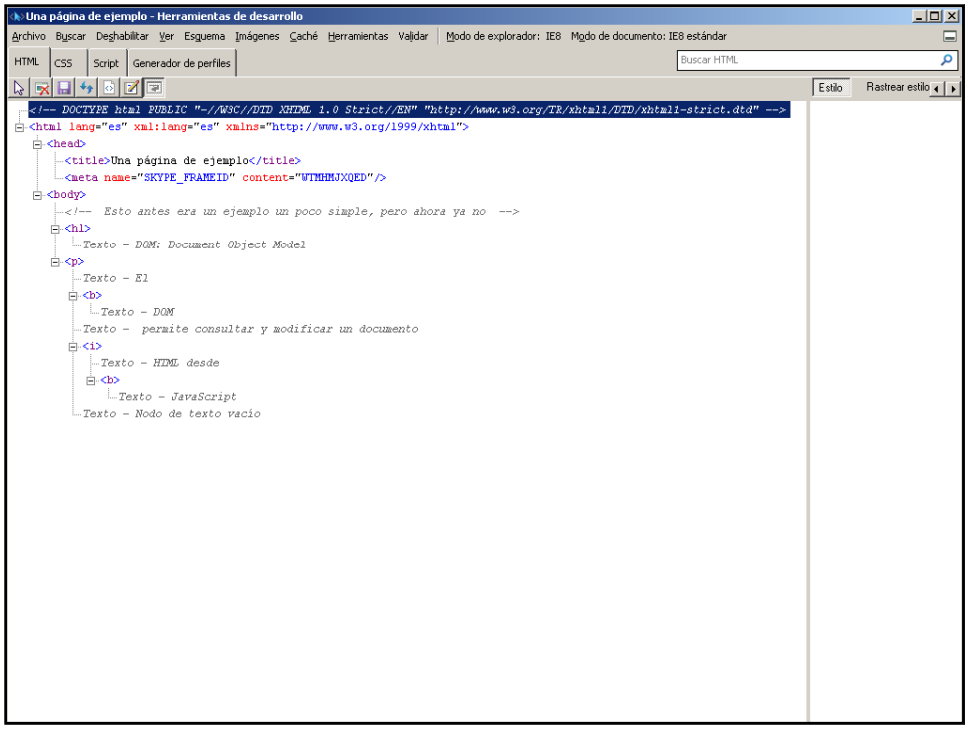
Programación en Internet – Curso 2009-2010

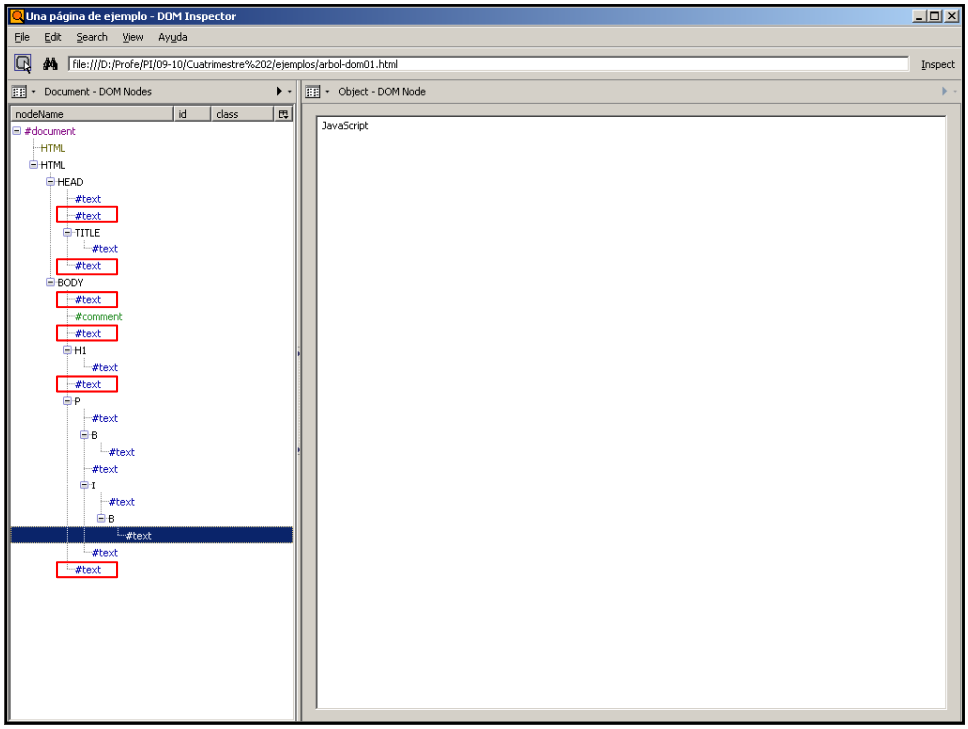
DOM

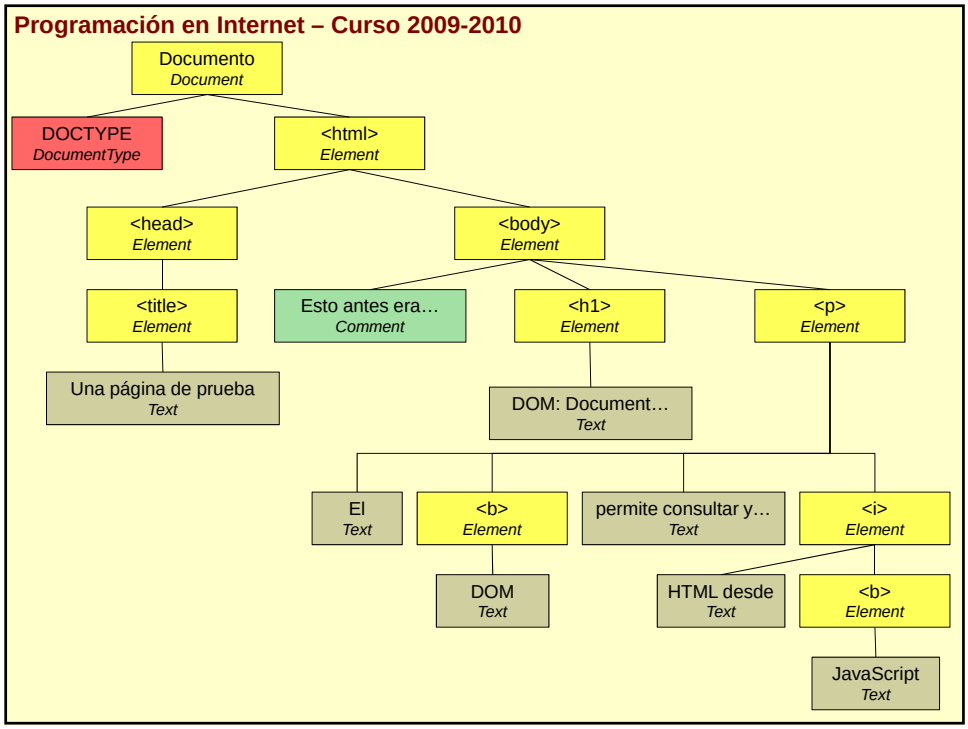
- Ejemplo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es"
   lang="es">
<head>
<title>Una página de ejemplo</title>
</head>
<body>
<!-- Esto antes era un ejemplo un poco simple, pero ahora ya no
   -->
<h1>DOM: Document Object Model</h1>
<p>
El <b>DOM</b> permite consultar y modificar un documento
   <i>HTML desde <b>JavaScript</b></i>
</p>
</body>
</html>
```


Una página de ejemplo - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Accessibility Ayuda
Anterior Siguiente Recargar Parar Inicio
file:///D:/Profe/PI/09-10/Cuatrimestre 2/ejemplos/arbol-dom01.html
Google
Una página de ejemplo
DOM: Document Object Model
El DOM permite consultar y modificar un documento HTML desde JavaScript.
Terminado


Una página de ejemplo - Herramientas de desarrollo
Archivo Buscar Deshabilitar Ver Esquema Imágenes Caché Herramientas Validar Modo de explorador: IE8 Modo de documento: IE8 estándar
HTML CSS Script Generador de perfiles
Buscar HTML
Estilo Rastrear estilo
<!-- DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" -->
<html lang="es" xml:lang="es" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Una página de ejemplo</title>
<meta name="SKYPE_FRAMEID" content="WTMHMJXQED"/>
<body>
<!-- Esto antes era un ejemplo un poco simple, pero ahora ya no -->
<h1>
Texto - DOM: Document Object Model
<p>
Texto - El
<b>
Texto - DOM
Texto - permite consultar y modificar un documento
<i>
Texto - HTML desde
<b>
Texto - JavaScript
Texto - Nodo de texto vacío


Una página de ejemplo - DOM Inspector
File Edit Search View Ayuda
file:///D:/Profe/PI/09-10/Cuatrimestre%202/ejemplos/arbol-dom01.html
Inspect
Document - DOM Nodes
Object - DOM Node
nodeName
id
class
JavaScript
#document
HTML
HEAD
TITLE
BODY
H1
P
B
I
#text
#comment


Programación en Internet – Curso 2009-2010
Documento
Document
DOCTYPE
DocumentType
<html>
Element
<head>
Element
<body>
Element
<title>
Element
Esto antes era…
Comment
<h1>
Element
<p>
Element
Una página de prueba
Text
DOM: Document…
Text
El
Text
<b>
Element
permite consultar y…
Text
<i>
Element
DOM
Text
HTML desde
Text
<b>
Element
JavaScript
Text

Programación en Internet – Curso 2009-2010

DOM Level 0

- Nivel 0:
  - No es una recomendación (no existe como tal)
  - "DOM Level 0“ refiere una mezcla de funcionalidades para trabajar con documentos HTML
  - Ofrecido desde Netscape Navigator version 3.0 y Microsoft Internet Explorer version 3.0

Programación en Internet – Curso 2009-2010

DOM Level 0

- `innerHTML:`
  - Asigna o devuelve el código HTML entre las etiquetas de inicio y final del objeto
  - Como no existe un especificación formal de esta propiedad, su implementación difiere
  - Nunca se debería de utilizar para escribir partes de una tabla → Usar los métodos de W3C DOM

DOM Level 0

- Ejemplo: eliminar el contenido entero de una página

```
// Sustituye el contenido de la etiqueta body
// por una cadena vacía
document.body.innerHTML = "";
```

Programación en Internet – Curso 2009-2010

DOM Level 0

- Ejemplo: texto que cambia al desplazar el puntero del ratón

```
<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>
```

Programación en Internet – Curso 2009-2010

DOM Level 1

- Nivel 1
  - *Document Object Model Level 1* (1/10/1998): definición de las estructuras de datos e interfaces básicos del modelo DOM
  - *Document Object Model Level 1 (Second Edition)* (29/9/2000)
  - Dos partes:
    - *Core*: conjunto fundamental de interfaces de bajo nivel que representan cualquier documento estructurado e interfaces extendidos para representar documentos XML
    - *HTML*: proporciona interfaces adicionales de alto nivel para un acceso más apropiado a documentos HTML

Programación en Internet – Curso 2009-2010

DOM Level 1

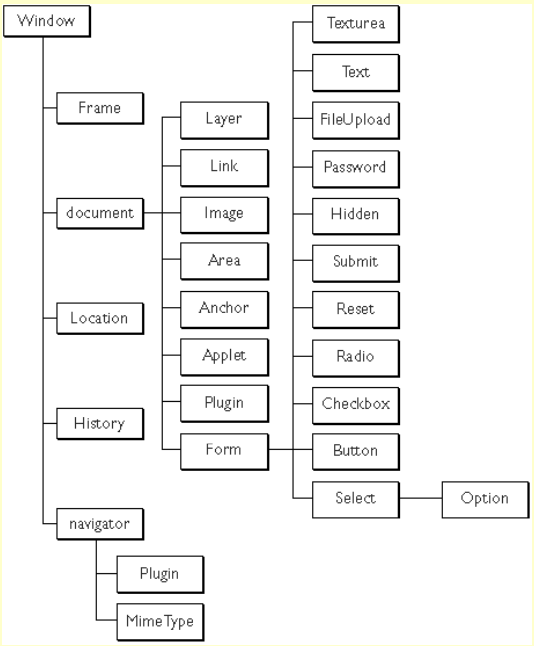
- Cada nodo es de tipo `Node`, pero existen subtipos: `Document, DocumentType, Element, Text,Comment, …`
- Atributos son nodos del tipo `Attr`, pero no aparecen en el árbol que representa el documento (hay que acceder a través de métodos del nodo)

Programación en Internet – Curso 2009-2010

DOM Level 1

- Objetos=propiedades+métodos+eventos
- Descendientes son propiedades y objetos a la vez

Programación en Internet – Curso 2009-2010



DOM Level 1

- `document` ⇔ `<body>…</body>`
- Propiedades:
  - `alinkColor, bgColor, fgColor, linkColor, vlinkColor`
  - `forms, images, title`
  - `lastModified, referrer`
- Métodos:
  - `write, writeln`

DOM Level 1

- Eventos:

`onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload`

DOM Level 1

- onClick
  - Se invoca al recibir un click el objeto
- onChange
  - Se invoca al cambiar el valor o estado del control
- onBlur y onFocus
  - 'Salta' al obtener (Focus) o perder (Blur) el enfoque el control correspondiente.

DOM Level 1

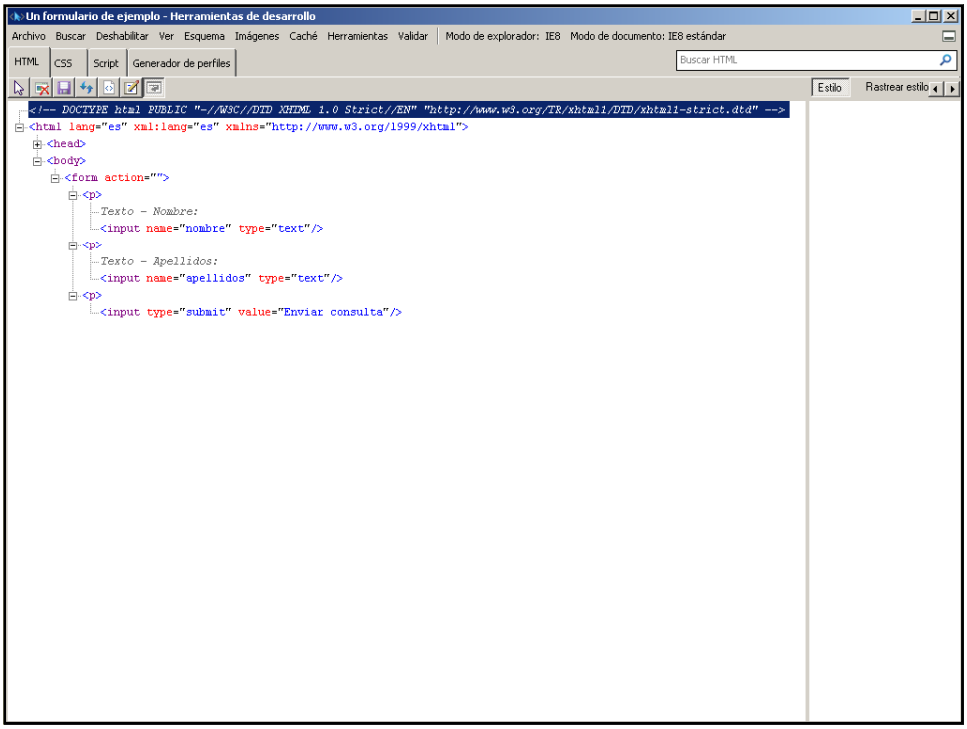
- Cada nodo posee un conjunto de propiedades que lo relacionan con sus "familiares":
  - `childNodes`
  - `firstChild`
  - `lastChild`
  - `parentNode`
  - `nextSibling`
  - `prevSibling`

Programación en Internet – Curso 2009-2010

DOM Level 1

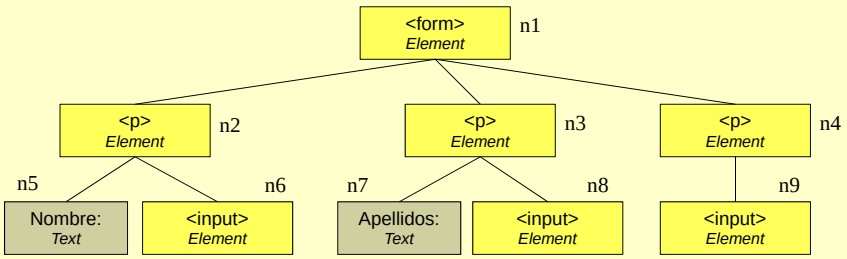
- Ejemplo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Un formulario de ejemplo</title>
</head>
<body>
<form action="">
<p>
Nombre: <input type="text" name="nombre" /></p>
<p>
Apellidos: <input type="text" name="apellidos" /></p>
<p>
<input type="submit" /></p>
</form>
</body>
</html>
```



Programación en Internet – Curso 2009-2010

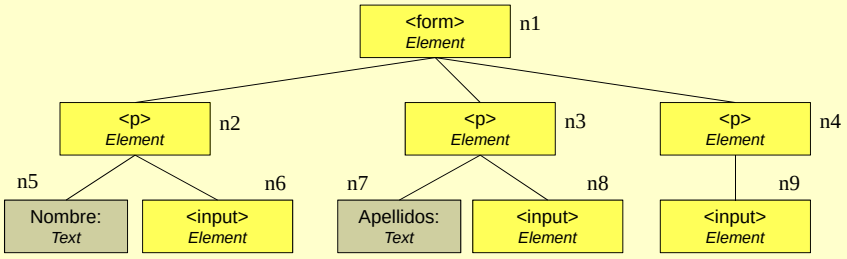
DOM Level 1



- n1.firstChild =
- n1.lastChild =
- n1.childNodes.length =
- n1.childNodes[0] =
- n1.childNodes[1] =
- n1.lastChild.firstChild =
- n9.parentNode.parentNode =
- n2.nextSibling =
- n4.prevSibling =
- n4.nextSibling =

Programación en Internet – Curso 2009-2010

DOM Level 1



- n1.firstChild = **n2**
- n1.lastChild =**n4**
- n1.childNodes.length = **3**
- n1.childNodes[0] = **n2**
- n1.childNodes[1] = **n3**
- n1.lastChild.firstChild = **n9**
- n9.parentNode.parentNode = **n1**
- n2.nextSibling = **n3**
- n4.prevSibling = **n3**
- n4.nextSibling = **null**

DOM Level 1

- getElementById(“elementID”)
  - Útil para recuperar un elemento concreto

```
var element = document.getElementById(“myTable”)
```

- getElementsByName(“name”):
  - Útil para recuperar los controles de un formulario con un mismo nombre
- getElementsByTagName(“tagName”):
  - Útil para recuperar un cojunto de elementos de un mismo tipo

```
var images = document.getElementsByTagName(“img”)
```

  - El valor especial "*" representa todas las etiquetas
- getAttribute(“attrName”):
  -

```
var element = document.getElementById(“myTable”).
getAttribute(“width”);
```

Programación en Internet – Curso 2009-2010

DOM Level 1

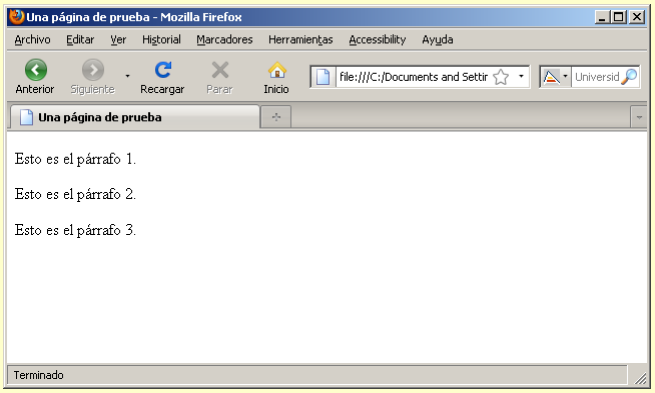
```
// HTML:
// <div id="d"><p>Esto es un párrafo.</p>
// <p>Esto es otro párrafo</p>
// </div>

d = document.getElementById("d");
alert(d.innerHTML);
```

Programación en Internet – Curso 2009-2010

DOM Level 1

- Cómo acceder a un elemento:
  - Queremos mostrar el texto del segundo párrafo



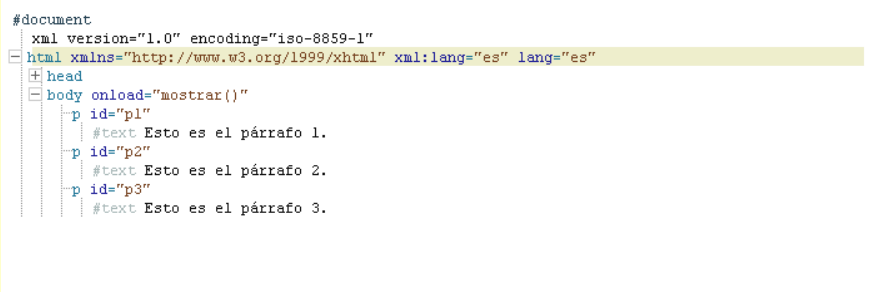
Programación en Internet – Curso 2009-2010

DOM Level 1

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Una página de prueba</title>
<script type="text/javascript">
function mostrar() {
  // Muestra el contenido del segundo párrafo de tres formas distintas
}
</script>
</head>
<body onload="mostrar()"><p id="p1">Esto es el párrafo 1.</p><p
   id="p2">Esto es el párrafo 2.</p><p id="p3">Esto es el párrafo
   3.</p></body>
</html>
```

Programación en Internet – Curso 2009-2010

DOM Level 1



Programación en Internet – Curso 2009-2010

DOM Level 1

```
<script type="text/javascript">
function mostrar() {
  var b, p;
  // El body de la página
  b = document.getElementsByTagName("body")[0];
  // El segundo párrafo es el primer hermano del primer párrafo
  p = b.firstChild.nextSibling;
  alert(p.firstChild.nodeValue);

  // El segundo párrafo es el segundo hijo
  p = b.childNodes[1];
  alert(p.firstChild.nodeValue);

  // El segundo párrafo tiene el id "p2"
  p = document.getElementById("p2");
  alert(p.firstChild.nodeValue);
}
</script>
```

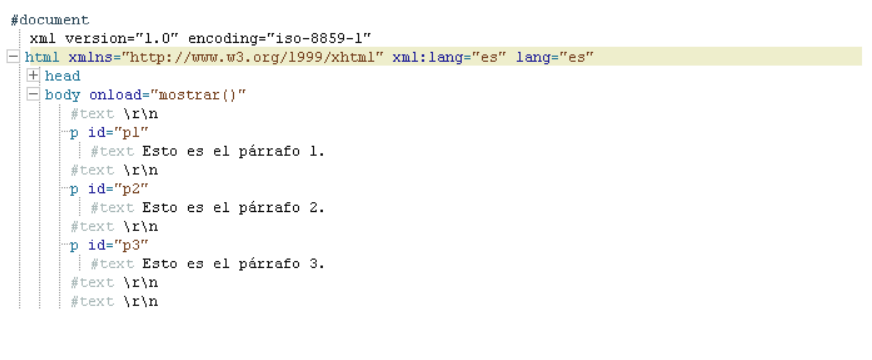
Programación en Internet – Curso 2009-2010

DOM Level 1

```
<body onload="mostrar()">
<p id="p1">
Esto es el párrafo 1.
</p>
<p id="p2">
Esto es el párrafo 2.
</p>
<p id="p3">
Esto es el párrafo 3.
</p>
</body>
</html>
```

Programación en Internet – Curso 2009-2010

DOM Level 1



Programación en Internet – Curso 2009-2010

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);`

Programación en Internet – Curso 2009-2010

DOM Level 1

- Cómo acceder al texto de una etiqueta:

```
// Crea un nuevo elemento de tipo párrafo
var p = document.createElement("p");

// Obtiene un párrafo de un documento
var p = document.getElementById("p1");
```

DOM Level 1

```
// Incorrecto: esto es DOM 0, no es estándar
p.innerHTML = txt;
// Correcto, es del estándar, pero en Internet Explorer
   no funciona
p.textContent = txt;
// Incorrecto, no es del estándar, es sólo de Internet
   Explorer (y algún otro), pero en el resto no funciona
p.text = txt;
// Incorrecto, no es del estándar, es sólo de Internet
   Explorer (y algún otro), pero en el resto no funciona
p.innerText = txt;
// Sólo tiene sentido con los siguientes tipos de nodo:
   CDATASection, Comment, ProcessingInstruction, Text
// Con un párrafo no debe funcionar
p.nodeValue = txt;
```

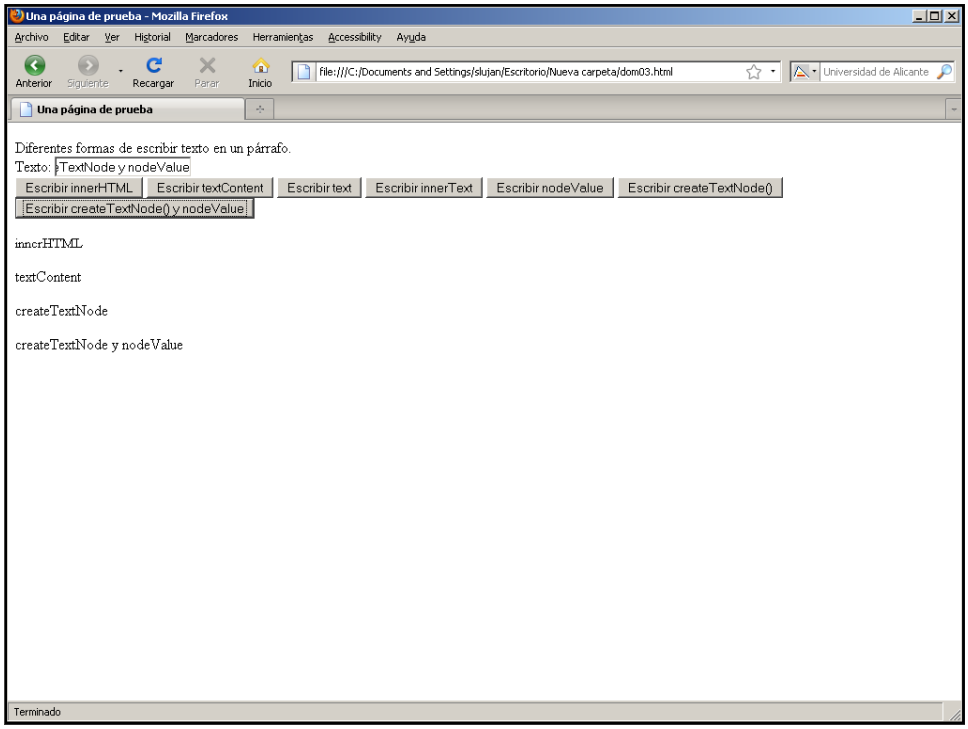

DOM Level 1

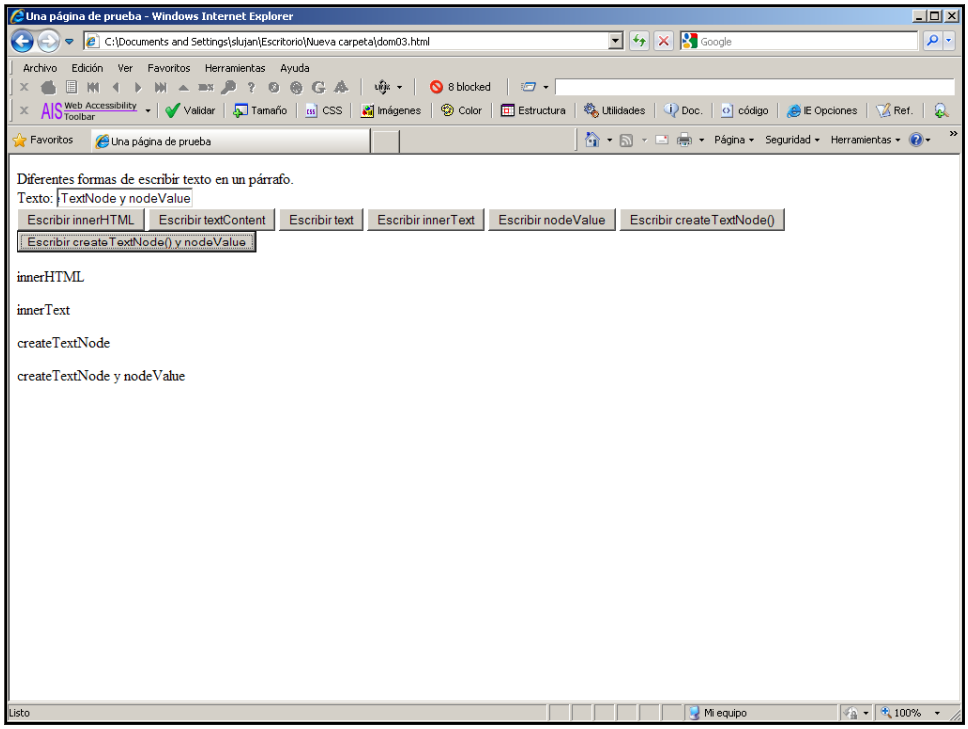
- ¿Cómo se tiene que hacer?

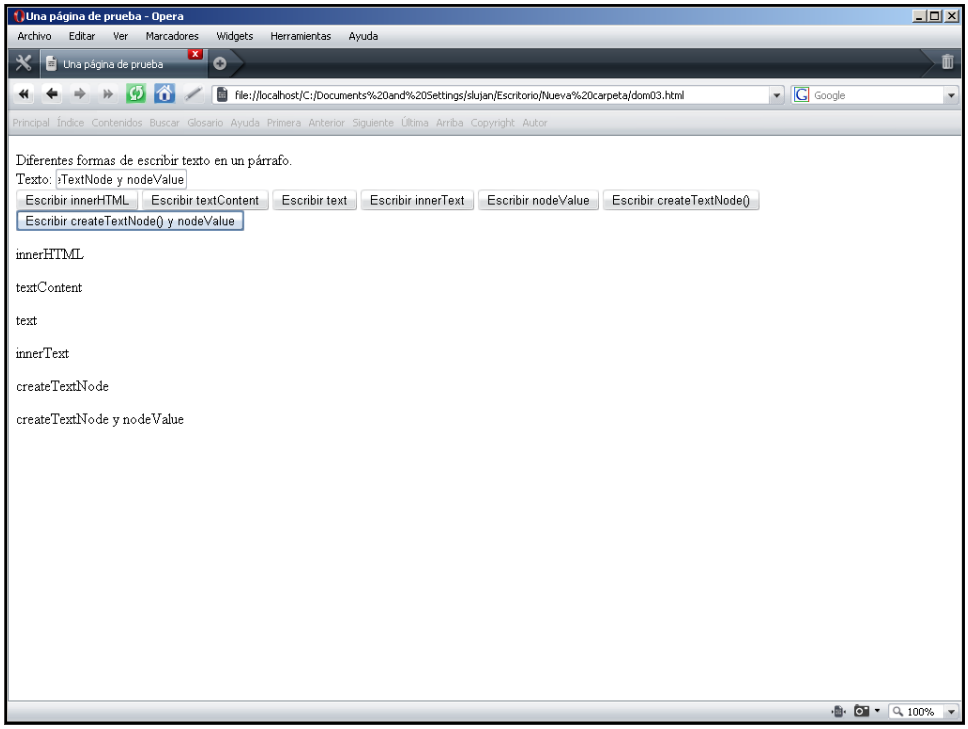
```
var p = document.createElement("p");

// Correcto, parece que funciona en todos
p.appendChild(document.createTextNode(txt));

// Correcto, parece que funciona en todos
var tn = document.createTextNode("");
tn.nodeValue = txt;
p.appendChild(tn);
```


Una página de prueba - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Accessibility Ayuda
Anterior Siguiente Recargar Parar Inicio
file:///C:/Documents and Settings/slujan/Escritorio/Nueva carpeta/dom03.html
Universidad de Alicante
Una página de prueba
Diferentes formas de escribir texto en un párrafo.
Texto: TextNode y nodeValue
Escribir innerHTML
Escribir textContent
Escribir text
Escribir innerText
Escribir nodeValue
Escribir createTextNode()
Escribir createTextNode() y nodeValue
innerHTML
textContent
createTextNode
createTextNode y nodeValue
Terminado


Una página de prueba - Windows Internet Explorer
C:\Documents and Settings\slujan\Escritorio\Nueva carpeta\dom03.html
Google
Archivo Edición Ver Favoritos Herramientas Ayuda
8 blocked
Web Accessibility Toolbar
Validar
Tamaño
CSS
Imágenes
Color
Estructura
Utilidades
Doc.
código
IE Opciones
Ref.
Favoritos
Una página de prueba
Página
Seguridad
Herramientas
Diferentes formas de escribir texto en un párrafo.
Texto: TextNode y nodeValue
Escribir innerHTML
Escribir textContent
Escribir text
Escribir innerText
Escribir nodeValue
Escribir createTextNode()
Escribir createTextNode() y nodeValue
innerHTML
innerText
createTextNode
createTextNode y nodeValue
Listo
Mi equipo
100%



Programación en Internet – Curso 2009-2010

DOM Level 2 - 3

- Versiones de DOM:
  - Nivel 2
    - *Document Object Model Level 2 Core* (13/11/2000)
    - *Document Object Model Level 2 Views*: permite mediante programas acceder dinámicamente a la representación de un documento
    - *Document Object Model Level 2 Events*: proporciona a los programas un sistema genérico de eventos
    - *Document Object Model Level 2 Style*: permite a los programas el acceso y la modificación dinámica de las hojas de estilo asociadas
    - *Document Object Model Level 2 Traversal and Range*: permite a los programas la identificación y el recorrido de rangos de documentos
    - *Document Object Model Level 2 HTML*: permite a los programas el acceso y la modificación dinámica de documentos HTML
  - Nivel 3
    - *Document Object Model Level 3 Core* (7/4/2004)
    - *Document Object Model Level 3 Load and Save*
    - *Document Object Model Level 3 Validation*

Programación en Internet – Curso 2009-2010

Cómo acceder a un formulario

- Tres formas:
  - `document.forms[n]`
  - `document.forms["miForm"]`
  - `document.miForm`
- Estructura general de acceso a un control:
  - `document.`*`nForm`*`.`*`nControl`*`.`*`propiedad`*

Programación en Internet – Curso 2009-2010

Cómo acceder a un formulario

- Campos de verificación (checkbox)
  - `checked`
  - `defaultChecked`
  - `value`
- Campos excluyentes (radio)
  - `checked`
  - `defaultChecked`
  - `value`

Programación en Internet – Curso 2009-2010

Cómo acceder a un formulario

- Campos de texto y áreas de texto
  - `defaultValue`
  - `value`
- Listas de selección:
  - `length`
  - `options` → `selected, text, value`
  - `selectedIndex`

Programación en Internet – Curso 2009-2010

BOM

- *Browser Object Model*
- En proceso de estandarización por W3C:
  - Window Object 1.0 W3C Working Draft 07 April 2006
- Objetos:
  - `window`
  - `history`
  - `location`
  - `navigator`
- Raíz: `window` (se puede eliminar)

```
window.navigator.appName
    navigator.appName
```

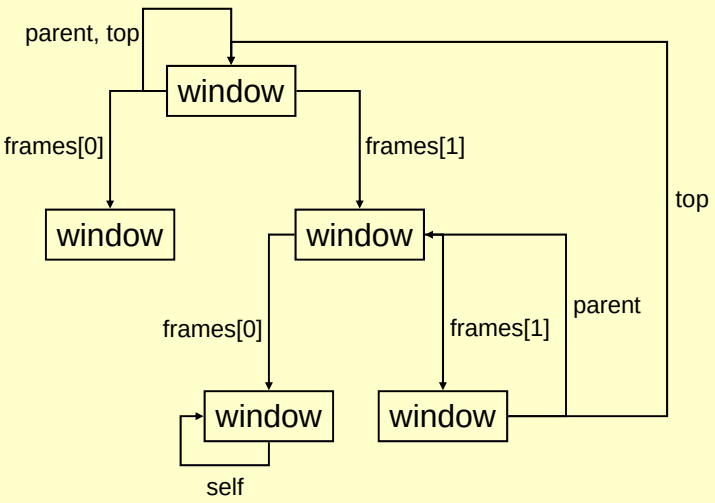
Programación en Internet – Curso 2009-2010

Objeto window

- Objeto principal
- Ventana (body) o marco (frameset)
- Propiedades (algunas son objetos):
  - document, frames, history, location
  - parent, self, top
  - status y defaultStatus (texto de la barra de estado de la ventana del navegador)

Programación en Internet – Curso 2009-2010

Objeto window



Programación en Internet – Curso 2009-2010

Objeto window

- Cada marco es un objeto window
- parent devuelve el objeto window padre en la jerarquía de objetos
- top devuelve el objeto window superior en la jerarquía de antecesores
- Para el objeto window raíz o principal, parent y top apuntan a sí mismo → Útil para saber si hay marcos

Programación en Internet – Curso 2009-2010

Objeto window

- Métodos
  - `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)`

Programación en Internet – Curso 2009-2010

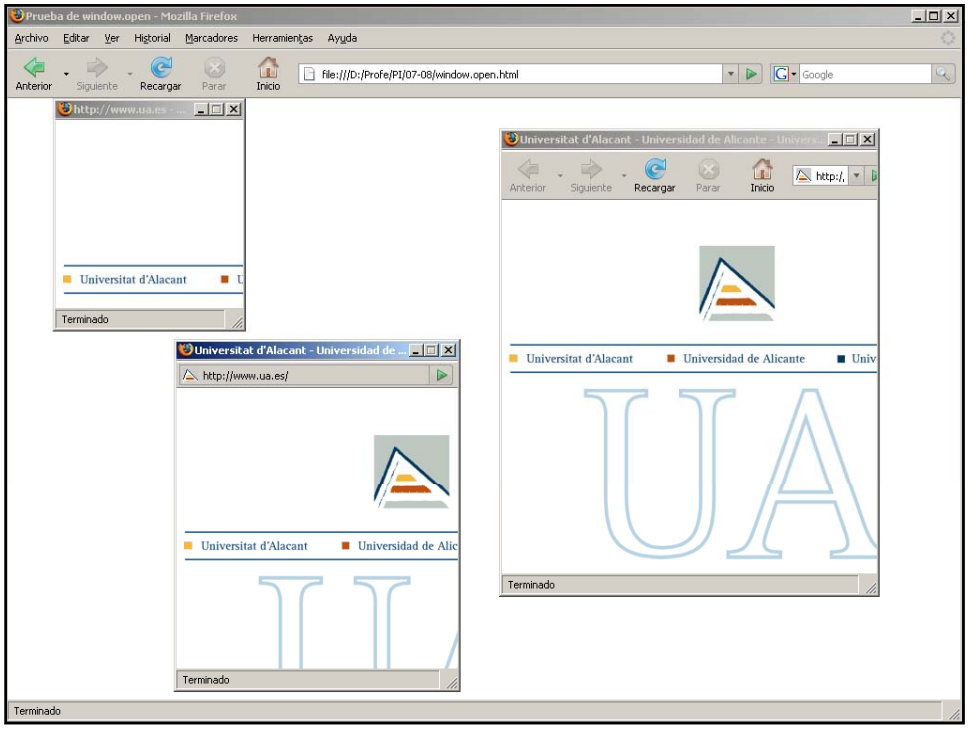
Objeto window

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)

Programación en Internet – Curso 2009-2010

Objeto 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>
```



Programación en Internet – Curso 2009-2010

Objeto history

- Array de URLs visitadas en la ventana actual
- Métodos:
  - back: página visitada anteriormente.
  - forward: siguiente página del histórico
  - go(n): ir a la página 'n' del histórico

Objeto location

- Contiene la URL del objeto window
- Propiedades:
  - host, hostname, href, pathname, port, protocol, search, hash
- Métodos:
  - reload, replace

Objeto navigator

- Información sobre el navegador (el programa cliente)
- Propiedades:
  - appCodeName, appName, appVersion, platform

Programación en Internet – Curso 2009-2010

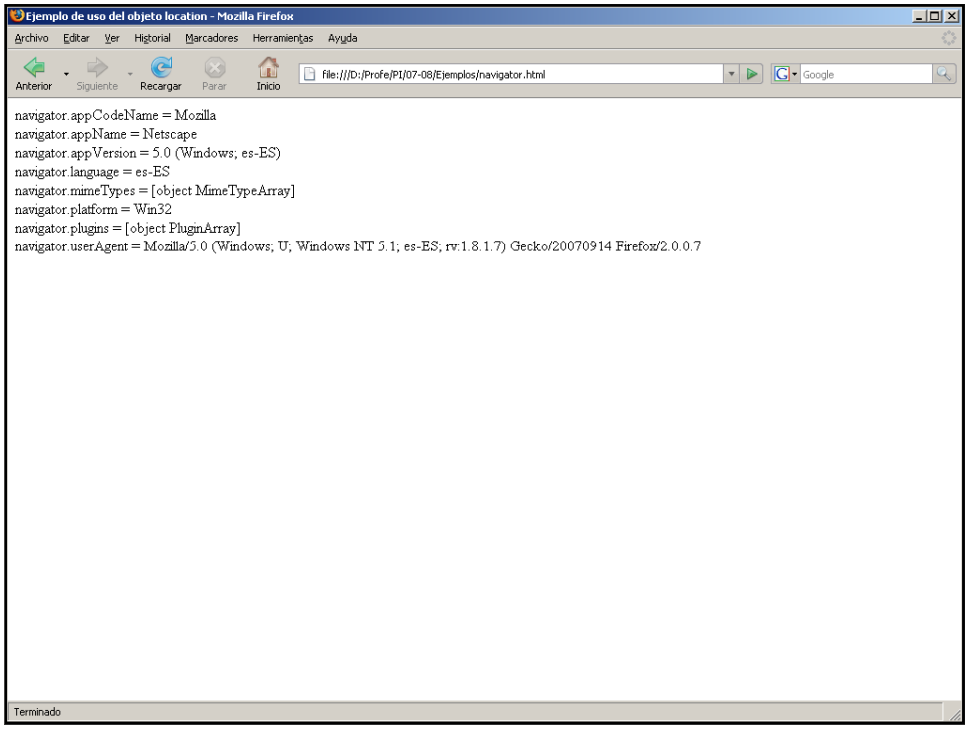
Objeto navigator

```
<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 />");
```

Programación en Internet – Curso 2009-2010

Objeto 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>
```


Ejemplo de uso del objeto location - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
Anterior Siguiente Recargar Parar Inicio
file:///D:/Profe/PI/07-08/Ejemplos/navigator.html
Google
navigator.appCodeName = Mozilla
navigator.appName = Netscape
navigator.appVersion = 5.0 (Windows; es-ES)
navigator.language = es-ES
navigator.mimeTypes = [object MimeTypeArray]
navigator.platform = Win32
navigator.plugins = [object PluginArray]
navigator.userAgent = Mozilla/5.0 (Windows; U; Windows NT 5.1; es-ES; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7
Terminado