



neitek_

DOM

Carolina Caballero

DOM

- **DOM** = Document Object Model
- Permite acceder y manipular las páginas XHTML
- DOM transforma los documentos XHTML en un conjunto de elementos llamados *nodos*, que están interconectados y que representan los contenidos de las páginas web y las relaciones entre ellos.
- La unión de todos los nodos se llama "*árbol de nodos*"

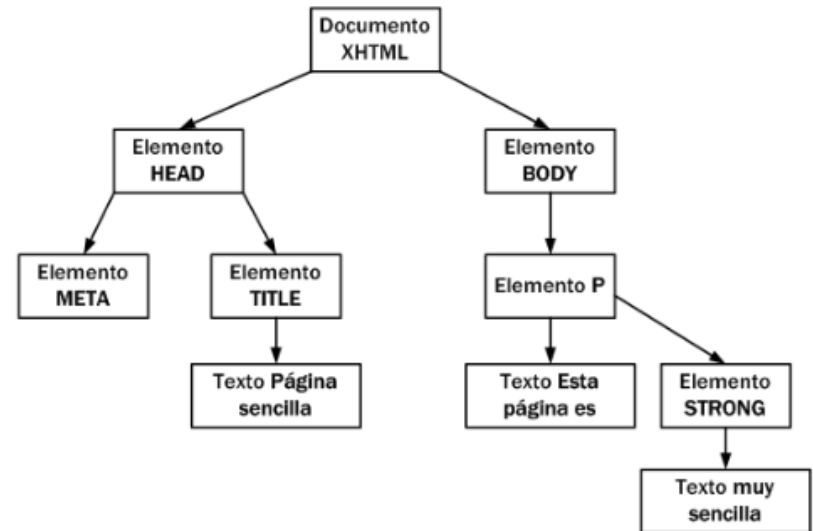
Potencial del DOM

- HTML DOM es un estándar para:
 - cambiar todos los elementos HTML en una página.
 - cambiar todos los atributos HTML en una página.
 - cambiar todos los estilos CSS en una página.
 - quitar elementos y atributos HTML.
 - agregar nuevos elementos y atributos HTML
 - reaccionar a los eventos HTML.
 - crear nuevos eventos HTML.

Árbol de nodos

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Página sencilla</title>
</head>

<body>
<p>Esta página es <strong>muy sencilla</strong></p>
</body>
</html>
```



Reglas de transformación de nodos

- Las etiquetas XHTML se transforman en dos nodos: el primero es la propia etiqueta y el segundo nodo es hijo del primero y consiste en el contenido textual de la etiqueta.

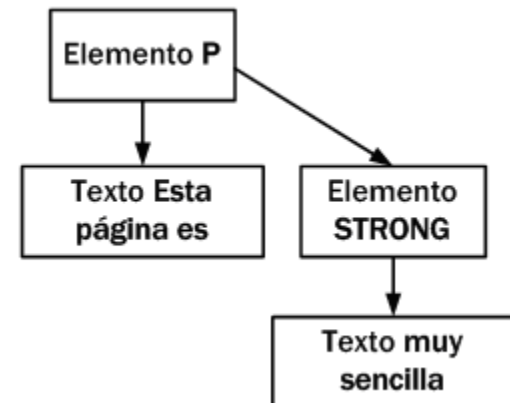
`<title>Página sencilla</title>`



Reglas de transformación de nodos

- Si una etiqueta XHTML se encuentra dentro de otra, se sigue el mismo procedimiento anterior, pero los nodos generados serán nodos hijo de su etiqueta padre.

`<p>Esta página es muy sencilla</p>`



Tipos de nodos

- *Document*, nodo raíz del que derivan todos los demás nodos del árbol.
- *Element*, representa cada una de las etiquetas XHTML. Se trata del único nodo que puede contener atributos y el único del que pueden derivar otros nodos.
- *Attr*, se define un nodo de este tipo para representar cada uno de los atributos de las etiquetas XHTML, es decir, uno por cada par atributo=valor.
- *Text*, nodo que contiene el texto encerrado por una etiqueta XHTML.
- *Comment*, representa los comentarios incluidos en la página XHTML.
- Otros nodos: *DocumentType*, *CDataSection*, *DocumentFragment*, *Entity*, *EntityReference*, *ProcessingInstruction* y *Notation*.

Acceso a nodos

- Solo se puede acceder a los nodos cuando el árbol DOM ha sido construido completamente.
- DOM proporciona dos métodos alternativos para acceder a un nodo específico:
 - acceso a través de sus nodos padre
 - acceso directo.

Acceso directo a los nodos

- Se utilizan los siguientes métodos:
 - `getElementsByTagName(etiqueta)`

```
var parrafos = document.getElementsByTagName("p");
```

- `getElementsByTagName(nombre)`

```
var parrafoEspecial = document.getElementsByTagName("especial");
```

- `getElementById(id)`

```
var cabecera = document.getElementById("cabecera");
```

Creación de nodos

- Requiere 4 pasos:
 1. Creación de un nodo tipo Element
 2. Creación de un nodo tipo Text
 3. Añadir el nodo Text como nodo hijo del nodo Element
 4. Añadir el nodo Element a la página, en forma de nodo hijo del nodo correspondiente donde queremos insertar el elemento

Creación de nodos

```
// Crear nodo de tipo Element
var parrafo = document.createElement("p");

// Crear nodo de tipo Text
var contenido = document.createTextNode("Hola Mundo!");

// Añadir el nodo Text como hijo del nodo Element
parrafo.appendChild(contenido);

// Añadir el nodo Element como hijo de la pagina
document.body.appendChild(parrafo);
```

Creación de nodos

- *createElement* (etiqueta) – crea un nodo tipo Element
- *createTextNode* (contenido) – crea un nodo tipo Text
- *nodoPadre.appendChild* (nodoHijo) – añade un nodo como hijo de otro nodo

Eliminación de nodos

- Para borrar nodos:
 - Obtener el nodo a borrar
 - *removeChild* desde el elemento padre se invoca el *removeChild*.

```
var parrafo = document.getElementById("provisional");  
parrafo.parentNode.removeChild(parrafo);
```

```
<p id="provisional">...</p>
```

Acceso a contenido

- innerHTML : esta propiedad sirve para obtener o cambiar cualquier elemento HTML , incluyendo <HTML> y <BODY>

```
HTMLElementObject.innerHTML
```

```
HTMLElementObject.innerHTML = text
```

```
document.getElementById("demo").innerHTML = "Paragraph changed!";
```

Acceso a contenido

- innerHTML : esta propiedad sirve para obtener o cambiar cualquier elemento HTML , incluyendo <HTML> y <BODY>

```
HTMLElementObject.innerHTML
```

```
HTMLElementObject.innerHTML = text
```

```
document.getElementById("demo").innerHTML = "Paragraph changed!";
```

Acceso a atributos XHTML

- Mediante DOM, es posible acceder a todos los atributos XHTML y todas las propiedades CSS.
- Los atributos XHTML de los elementos de la página se transforma automáticamente en propiedades de los nodos.

Acceso a atributos XHTML

```
var enlace = document.getElementById("enlace");  
alert(enlace.href); // muestra http://www...com
```

```
<a id="enlace" href="http://www...com">Enlace</a>
```

```
var parrafo = document.getElementById("parrafo");  
alert(parrafo.style.fontWeight); // muestra "bold"
```

```
<p id="parrafo" style="font-weight: bold;">...</p>
```

Acceso a atributos XHTML

```
var imagen = document.getElementById("imagen");  
alert(imagen.style.margin);
```

```

```

Nombres de atributos XHTML

- Generalmente el mismo nombre de la propiedad.
- Si el nombre es compuesto cambia de la siguiente manera:

`font-weight` se transforma en `fontWeight`

- El único atributo XHTML que no cumple esta regla es `class` (puesto que `class` es reservada por JavaScript), por lo que cambia a `className`.

Buscando objetos HTML

Property	Description
document.anchors	Returns all <a> elements that have a name attribute
document.applets	Returns all <applet> elements (Deprecated in HTML5)
document.baseURI	Returns the absolute base URI of the document
document.body	Returns the <body> element
document.cookie	Returns the document's cookie
document.doctype	Returns the document's doctype
document.documentElement	Returns the <html> element
document.documentMode	Returns the mode used by the browser
document.documentURI	Returns the URI of the document
document.domain	Returns the domain name of the document server
document.domConfig	Obsolete. Returns the DOM configuration
document.embeds	Returns all <embed> elements

Buscando objetos HTML

document.forms	Returns all <form> elements
document.head	Returns the <head> element
document.images	Returns all elements
document.implementation	Returns the DOM implementation
document.inputEncoding	Returns the document's encoding (character set)
document.lastModified	Returns the date and time the document was updated
document.links	Returns all <area> and <a> elements that have a href attribute
document.readyState	Returns the (loading) status of the document
document.referrer	Returns the URI of the referrer (the linking document)
document.scripts	Returns all <script> elements
document.strictErrorChecking	Returns if error checking is enforced
document.title	Returns the <title> element
document.URL	Returns the complete URL of the document

Ejercicios

- ◉ Usando Exercise01.html obten:
 - ◉ Número de enlaces de la página
 - ◉ Dirección a la que enlaza el penúltimo enlace
 - ◉ Numero de enlaces que enlazan a <http://prueba>
 - ◉ Número de enlaces del tercer párrafo
- ◉ Usando Exercise02.html:
 - ◉ Haz la funcion `agrega()` para agregar nuevos nodos y añadelos a la lista existente.
 - ◉ Verifica que no exista previamente el nuevo elemento a agregar.

PREGUNTAS

neitek_