En su primera versión XHML 1.0.0 era una mezcla de XML y HTML

DOM: Se trata de un estándar que define como acceder a documentos XHTML.

La creación del Document Object Model o DOM es una de las innovaciones que mas ha influido en el desarrollo de las paginas web dinámicas y de las aplicaciones web más complejas

DOM permite a los programadores de acceder y manipular paginas XHTML.

Para sacar provecho a las utilidades de DOM, es necesario “transformar la página original. Una página HTML normal no es más que una sucesión de caracteres por lo que es un formato muy difícil de manipular. Por ello, los navegadores web transforman automáticamente todas las páginas web en una estructura más eficiente de manipular.

Tipos de nodos DOM:

* Document: Nodo raíz que derivan todos los demás nodos del arbol
* Element: Representa cada una de las etiquetas XHTML
* Attr: Representa cada uno de los atributos de las etiquetas XHTML
* Text: nodo que contiene el teto encerrado por una etiqueta
* Comment. Representa los comentarios incluidos en la pagina XHTML

Propiedades de los nodos:

* Elemento.innerHTML: todo lo que hay entre la etiqueta que abre elemento y la que lo cierra, incluyendo otras etiquetas HTML.
* Elemento.textContent: todo lo que hay entre la etiqueta que abre elemento y la que lo cierra, pero ignorando otras etiquetas HTML
* Elemento.value: devuelve la propiedad ‘value’ de un <input> (en el caso de un <input> de tipo text devuelve lo que hay escrito en el). Como los <inputs> no tienen etiqueta de cierre (</input>) no podemos usar .innerHTML ni textContent.

Acceso desde propiedades del document. Atajos para obtener elementos comunes:

* Document.documentElement.
* Document. Anchor: anclas del documento

getElementByTagName(etiqueta): Esta función devuelve un array con todos los elementos DOM del subárbol cuya etiqueta XHTML sea la indicada en la cadena “etiqueta”.

**Formularios:**

Un formulario web es un elemento que permite recopilar datos de forma ordenada para enviarlos a su validación. Disponen de una arquitectura basada en HTML.

La validación de JS no garantiza que los datos enviados son correctos. Es, simplemente, una cuestión de experiencia de usuario que evita mandar datos erróneos al servidor y hacer operaciones innecesarias. Pero es posible deshabilitar el JS del navegador o utilizar la consola del desarrollador para cambiar estos datos, por lo que no se puede asegurar su validez.

Por eso, la seguridad en la validación de un formulario debe realizarse en el lado del servidor y en la base de datos.

El elemento principal del formulario se marca con <input>

Según la funcionalidad se llaman:

* Controles de formulario
* Campos de formulario