INTRODUCCIÓN A HTML5

¿Qué es HTML?

HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Historia del Estándar

En 1989 existían dos técnicas que permitían vincular documentos electrónicos, por un lado los hipervínculos (enlaces) y por otro lado un poderoso lenguaje de etiquetas denominado SGML. Por entonces, Tim Berners-Lee da a conocer a la prensa que estaba trabajando en un sistema que permitirá acceder a ficheros en línea que funcionaba sobre redes de computadoras o máquinas electrónicas basadas en el protocolo TCP/IP.

A principios de 1990, Tim Berners-Lee define por fin el HTML como un subconjunto del conocido SGML y crea algo más valioso aún, el World Wide Web. En 1991, Tim Berners-Lee crea el primer navegador web, que funcionaría en modo texto y sobre un sistema operativo UNIX. Los trabajos para crear un sucesor del HTML, denominado HTML +, comenzaron a finales de 1993. HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se le dio, por lo tanto, el número de versión 2 para distinguirla de las propuestas no oficiales previas. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar, a pesar de ser la base formalmente más parecida al aspecto compositivo de las especificaciones actuales.

El borrador del estándar HTML 3.0 fue propuesto por el recién formado W3C en marzo de 1995. Con él se introdujeron muchas nuevas capacidades; por ejemplo, facilidades para crear tablas, hacer que el texto fluyese alrededor de las figuras y mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatible con HTML 2.0, era demasiado complejo para ser implementado con la tecnología de la época, y cuando el borrador del estándar expiró en septiembre de 1995, se abandonó debido a la carencia de apoyos de los fabricantes de navegadores web. El HTML 3.1 nunca llegó a ser propuesto oficialmente, y el estándar siguiente fue el HTML 3.2, que abandonaba la mayoría de las nuevas características del HTML 3.0 y, a cambio, adoptó muchos elementos desarrollados inicialmente por los navegadores web Netscape y Mosaic. La posibilidad de trabajar con fórmulas matemáticas que se había propuesto en el HTML 3.0 pasó a quedar integrada en un estándar distinto llamado MathML.

En 1997, HTML 4.0 se publicó como una recomendación del W3C. HTML 4.0 adoptó muchos elementos específicos desarrollados inicialmente para un navegador web concreto, pero al mismo tiempo comenzó a limpiar el HTML señalando algunos de ellos como «desaprobados» o deprecated en inglés.

HTML 4.0 implementa características como XForms 1.0 que no necesitan implementar motores de navegación que eran incompatibles con algunas páginas web HTML. En 2004 la W3C reabrió el debate de la evolución del HTML, y se dieron a conocer las bases para la versión HTML5. No obstante, este trabajo fue rechazado por los miembros del W3C y se daría preferencia al desarrollo del XML.

Apple, Mozilla y Opera anunciaron su interés en seguir trabajando en el proyecto bajo el nombre de WHATWG,13 que se basa en la compatibilidad con tecnologías anteriores. En 2006, el W3C se interesó en el desarrollo de HTML5, y en 2007 se unió al grupo de trabajo del WHATWG para unificar el proyecto.

Fuente
Ver tambien: Relaciones y Diferencias entre SGML, HTML y XML

Estructura Base de un Documento HTML

Los "*tags*" que describen la estructura general de un documento y dan una información sencilla sobre él son: **<HTML> <HEAD> <TITLE> <BODY>**. Estas tags no afectan a la apariencia del documento y solo interpretan y filtran los archivos HTML

<HTML>: Limitan el documento e indica que se encuentra escrito en este lenguaje.
<HEAD>: Especifica el prólogo del resto del archivo. Son pocas las "*tags*" que van dentro de ella, destacando la del título.
<TITLE> que será utilizado por los marcadores del navegador e identificará el contenido de la página. Solo puede haber un título por documento, preferiblemente corto aunque significativo, y no caben otras tags dentro de él.
<BODY>: Encierra el cuerpo principal del documento, el contenido.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
 <html lang="es">
   <head>
     <title>Ejemplo</title>
   </head>
   <body>
     <p>ejemplo</p>
   </body>
 </html>
```

¿Qué es el DOM?

DOM o *Document Object Model* es un conjunto de utilidades específicamente diseñadas para

manipular documentos XML. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML. Técnicamente, DOM es una API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente.

Antes de poder utilizar sus funciones, DOM transforma internamente el archivo XML original en una estructura más fácil de manejar formada por una jerarquía de nodos. De esta forma, DOM transforma el código XML en una serie de nodos interconectados en forma de *árbol*.

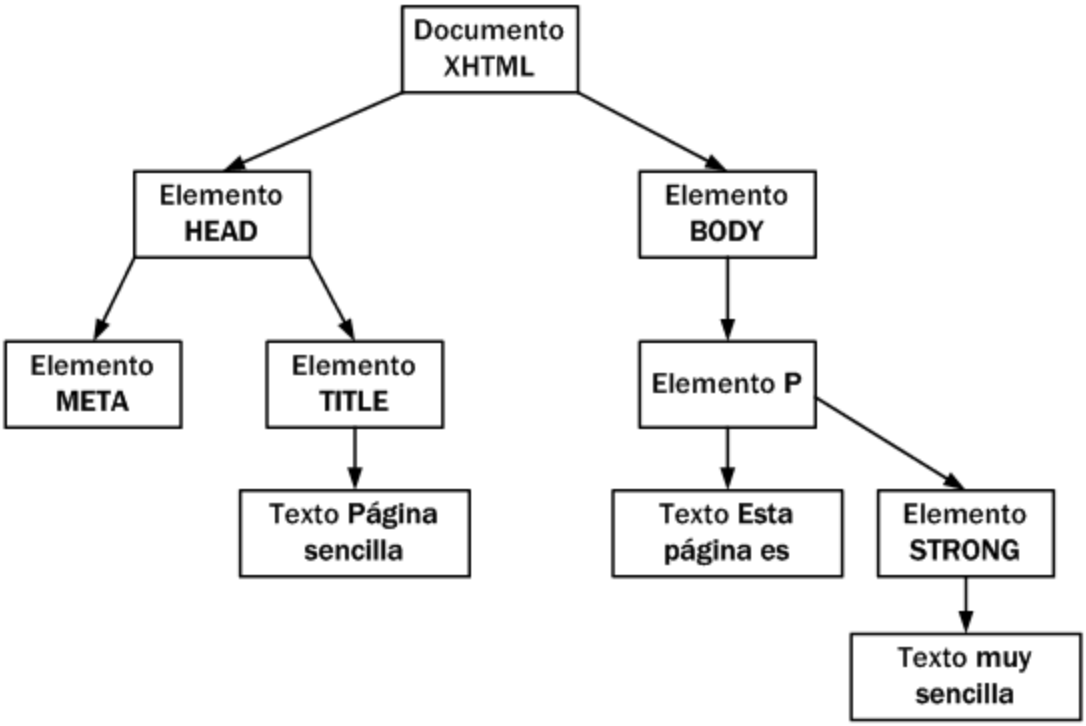
El árbol generado no sólo representa los contenidos del archivo original (mediante los nodos del árbol) sino que también representa sus relaciones (mediante las ramas del árbol que conectan los nodos). Aunque en ocasiones DOM se asocia con la programación web y con JavaScript, la API de DOM es independiente de cualquier lenguaje de programación. De hecho, DOM está disponible en la mayoría de lenguajes de programación comúnmente empleados.

Si se considera la siguiente página HTML sencilla:

```
<!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>
```

Del documento anterior se desprende el siguiente árbol jerárquico de nodos:



Fuente

Atributos

Cualquier etiqueta XHTML puede contener uno o más atributos, separados por espacios, que permiten especificar la etiqueta. El código fuente tiene el siguiente aspecto:

```
<etiqueta atributo="valor de atributo">
<etiqueta atributo1="valor_de_atributo1" atributo2="valor_de_atributo2" >
```

En el XHTML los atributos tienen que estar siempre definidos (mediante el signo igual "=") y el valor del atributo tiene que estar entre comillas dobles o simples. Existen una serie de atributos comunes a todas las etiquetas (salvo la etiqueta <html>) y algunas etiquetas tienen unos atributos específicos. En esta lección se comentan los atributos comunes.

Atributo Class

El atributo class permite definir clases de elementos. Las hojas de estilo pueden hacer referencia a los elementos con atributo class.

Atributo Id

El atributo id permite identificar de forma unívoca un elemento en un documento. Por ello no puede haber dos elementos con el mismo valor en el atributo id.

El nombre de un atributo id tiene que empezar con una letra (mayúscula o minúscula, caracteres latinos) y puede contener dígitos, guiones medios (-), guiones bajos (_), comas y puntos. Si el nombre

contiene puntos, a continuación del punto tiene que haber una letra. Un elemento sólo puede tener un atributo id, es decir, el atributo id no puede contener espacios. Las hojas de estilo pueden hacer referencia a un elemento con atributo id

Atributo Style [No usar en producción, mala práctica, peligro de muerte]

El atributo style permite definir propiedades de estilo a un elemento determinado.

Atributo Title

El atributo title contiene el texto que se muestra en forma de "tooltip" (cuadrito amarillo que aparece cuando se sitúa el ratón encima del elemento).

Etiquetas Básicas

**División **

La etiqueta define una división. Esta etiqueta permite agrupar varios elementos de bloque (párrafos, encabezados, listas, tablas, divisiones, etc). En principio, los navegadores no muestran nada especial cuando se crea una división, salvo que se dé formato a la división con la hoja de estilo.

Una división no puede insertarse dentro de una etiqueta en-línea (,, etc.) o de un bloque de texto (párrafo <p>, encabezado <h1> ... <h6>, preformateado <pre>, lista, etc), pero si puede insertarse dentro de una tabla, de un bloque de cita <blockquote> o de una división .

Parrafo <p>

La etiqueta de texto más común es la etiqueta <p>, pensada para contener párrafos, es decir todo lo que no tenga un significado especial (títulos, etc). A diferencia de los procesadores de textos, en los que se pueden separar dos párrafos mediante párrafos vacíos, si un párrafo <p> no contiene nada, los navegadores no lo muestran (salvo que la hoja de estilo incluya bordes o márgenes).

Encabezados <h1>, <h2>, <h3>, <h4>, <h5> y <h6>

Para los títulos y subtítulos de los apartados de un documento debes utilizar las etiquetas <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. Debes utilizar la etiqueta <h1> para el título principal del documento (a no confundir con la etiqueta <title>, que corresponde al texto de la barra de título de la ventana del navegador). Debes utilizar la etiqueta <h2> en los títulos de los apartados del documento, <h3> para los subapartados de cada apartado, y así sucesivamente.

Hipervínculo <a>

El elemento más importante que tiene una página de internet es el hipervínculo, estos nos permiten cargar otra página en el navegador. Normalmente un navegador al encontrar esta marca muestra un texto subrayado, y al hacer clic con el mouse sobre éste el navegador carga la página indicada por dicho hipervínculo.

```
<a href="URL" title="título del enlace">etiqueta del enlace</a>
```

Preformateado <pre>

La etiqueta <pre> se utiliza cuando se quiere conservar los espacios en blanco y los saltos de línea del texto original. En el resto de etiquetas, los navegadores no muestran ni las líneas en blanco ni varios

espacios en blanco seguidos.

Codigo <code>

Pensada para etiquetar fragmento de código de ordenador. Los navegadores suelen mostrar la etiqueta <code> en tipo de letra monospace (normalmente Courier).

**Salto de Línea
**

La etiqueta
 (del inglés break) permite insertar saltos de línea en un párrafo (o cualquier etiqueta de bloque).

**Listas <ul>, <ol>, **

El lenguaje HTML define tres tipos diferentes de listas para agrupar los elementos: listas no ordenadas <ul> (se trata de una colección simple de elementos en la que no importa su orden), listas ordenadas (similar a la anterior, pero los elementos están numerados y por tanto, importa su orden) y listas de definición <dl> (un conjunto de términos y definiciones similar a un diccionario).

```
<ul>
  <li>Inicio</li>
  <li>Noticias</li>
  <li>Artículos</li>
  <li>Contacto</li>
</ul>
```

Tablas <table>

Las tablas más sencillas de HTML se definen con tres etiquetas: `<table>` para crear la tabla, `<tr>` para crear cada fila y `<td>` para crear cada celda. A continuación se muestra el código HTML de una tabla sencilla:

```
<table>
<!--Encabezado de columnas-->
<tr>
  <th>Curso</th>
  <th>Horas</th>
  <th>Horario</th>
</tr>
<!--Contenido de la Tabla-->
<tr>
  <td>CSS</td>
  <td>20</td>
  <td>16:00 - 20:00</td>
</tr>
<tr>
  <td>HTML</td>
  <td>20</td>
  <td>16:00 - 20:00</td>
</tr>
```

```
</table>
```

Formularios Basicos

HTML es un lenguaje de marcado cuyo propósito principal consiste en estructurar los contenidos de los documentos y páginas web. Sin embargo, HTML también incluye elementos para crear aplicaciones web. El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.

Los formularios más sencillos se pueden crear utilizando solamente dos etiquetas: <form> y <input>. La etiqueta <form> encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta <input> permite definir varios tipos diferentes de elementos (botones y cuadros de texto).

La mayoría de formularios utilizan sólo los atributos action y method. El atributo action indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador.

El atributo method establece la forma en la que se envian los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son GET y POST. De esta forma, casi todos los formularios incluyen el atributo method="get" o el atributo method="post".

Al margen de otras diferencias técnicas, el método POST permite el envío de mucha más información que el método GET. En general, el método GET admite como máximo el envío de unos 500 bytes de información. La otra gran limitación del método GET es que no permite el envío de archivos adjuntos con el formulario. Además, los datos enviados mediante GET se ven en la barra de direcciones del navegador (se añaden al final de la URL de la página), mientras que los datos enviados mediante POST no se pueden ver tan fácilmente.

Si no sabes qué método elegir para un formulario, existe una regla general que dice que el método GET se debe utilizar en los formularios que no modifican la información (por ejemplo en un formulario de búsqueda). Por su parte, el método POST se debería utilizar cuando el formulario modifica la información original (insertar, modificar o borrar alguna información).

El ejemplo más común de formulario con método GET es el de los buscadores. Si realizas una búsqueda con tu buscador favorito, verás que las palabras que has introducido en tu búsqueda aparecen como parte de la URL de la página de resultados.

Del resto de atributos de la etiqueta <form>, el único que se utiliza ocasionalmente es enctype. Como se explica más adelante, este atributo es imprescindible en los formularios que permiten adjuntar archivos.

Atributos del tag <input>

- type = "text | password | checkbox | radio | submit | reset | file | hidden | image | button" - Indica el

tipo de control que se incluye en el formulario

- `name = "texto"` - Asigna un nombre al control (es imprescindible para que el servidor pueda procesar el formulario)
- `value = "texto"` - Valor inicial del control
- `size = "unidad_de_medida"` - Tamaño inicial del control (para los campos de texto y de password se refiere al número de caracteres, en el resto de controles se refiere a su tamaño en píxel)
- `maxlength = "número"` - Máximo número de caracteres para los controles de texto y de password
- `checked = "checked"` - Para los controles checkbox y radiobutton permite indicar qué opción aparece preseleccionada
- `disabled = "disabled"` - El control aparece deshabilitado y su valor no se envía al servidor junto con el resto de datos
- `readonly = "readonly"` - El contenido del control no se puede modificar
- `src = "url"` - Para el control que permite crear botones con imágenes, indica la URL de la imagen que se emplea como botón de formulario
- `alt = "texto"` - Descripción del control

```
<form action="maneja_formulario.php" method="post">
  <fieldset>
    <legend>Datos personales</legend>
    Nombre <br/>
    <input type="text" name="nombre" value="" />
    <br/>
    Apellidos <br/>
    <input type="text" name="apellidos" value="" />
    <br/>
    DNI <br/>
    <input type="text" name="dni" value="" size="10" maxlength="9" />
  </fieldset>

  <fieldset>
    <legend>Datos de conexión</legend>
    Nombre de usuario<br/>
    <input type="text" name="nombre" value="" maxlength="10" />
    <br/>
    Contraseña<br/>
    <input type="password" name="password" value="" maxlength="10" />
    <br/>
    Repite la contraseña<br/>
    <input type="password" name="password2" value="" maxlength="10" />
  </fieldset>
</form>
```

¿Qué es HTML5?

Segun Wikipedia
HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como *HTML5* y una variante XHTML conocida como sintaxis *XHTML5* que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Segun Cristalab
HTML5 es la actualización de HTML, el lenguaje en el que es creada la web. HTML5 también es un termino de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript.

Segun W3C
HTML5 is the cornerstone of the W3C's open web platform; a framework designed to support innovation and foster the full potential the web has to offer. Heralding this revolutionary collection of tools and standards, the HTML5 identity system provides the visual vocabulary to clearly classify and communicate our collective efforts.

- **SEMANTICS**: Giving meaning to structure, semantics are front and center with HTML5. A richer set of tags, along with RDFa, microdata, and microformats, are enabling a more useful, data driven web for both programs and your users.
- **OFFLINE & STORAGE:** Web Apps can start faster and work even if there is no internet connection, thanks to the HTML5 App Cache, as well as the Local Storage, Indexed DB, and the File API specifications.
- **DEVICE ACCESS:** Beginning with the Geolocation API, Web Applications can present rich, device-aware features and experiences. Incredible device access innovations are being developed and implemented, from audio/video input access to microphones and cameras, to local data such as contacts & events, and even tilt orientation.
- **CONNECTIVITY:** More efficient connectivity means more real-time chats, faster games, and better communication. Web Sockets and Server-Sent Events are pushing (pun intended) data between client and server more efficiently than ever before.
- **MULTIMEDIA:** Audio and video are first class citizens in the HTML5 web, living in harmony with your apps and sites. Lights, camera, action!
- **3D, GRAPHICS & EFFECTS:** Between SVG, Canvas, WebGL, and CSS3 3D features, you're sure to amaze your users with stunning visuals natively rendered in the browser.
- **PERFORMANCE & INTEGRATION:** Make your Web Apps and dynamic web content faster with a variety of techniques and technologies such as Web Workers and XMLHttpRequest 2. No user should ever wait on your watch.
- **CSS3:** CSS3 delivers a wide range of stylization and effects, enhancing the web app without sacrificing your semantic structure or performance. Additionally Web Open Font Format (WOFF) provides typographic flexibility and control far beyond anything the web has offered before.

Doctype y Estructura base

Es un DOCTYPE mucho más simplificado que XHTML (cuyas reglas siguen siendo usadas) y te permite usar

todas las habilidades de HTML5 sin que nada de lo que ya tienes programado deje de funcionar.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title></title>
    </head>
    <body>
    </body>
</html>
```

Etiquetas Semánticas

Las principales etiquetas HTML5 nuevas no tienen una representación especial en pantalla. Todas se comportan como un o un . Pero cada una tiene un significado semántico superior a un simple div o span.

<header>

Hacer cosas como es un poco estúpido cuando el 99% de los proyectos web tienen una cabecera. <header> está diseñada para reemplazar la necesidad de crear divs sin significado semántico.

<hgroup>

Muchos headers necesitan múltiples títulos, como un blog que tiene un titulo y un tagline explicando el blog. <hgroup> permite colocar un h1, h2 y h3 dentro del header sin afectar el SEO, permitiendo usar otro h1 en el sitio. En el HTML actual, sólo puedes usar h1 una vez por sitio o el h1 pierde prioridad de SEO.

<nav>

Igual que <header>, <nav> está diseñado para que ahí coloques la botonera de navegación principal. Puedes colocar cualquier etiqueta dentro, aunque lo recomendado es usar listas <ul>. Sólo puedes tener un <nav> por página.

<section>

Define un área de contenido única dentro del sitio. En un blog, sería la zona donde están todos los posts. En un video de youtube, habría un section para el video, uno para los datos del video, otro para la zona de comentarios.

<article>

Define zonas únicas de contenido independiente. En el home de un blog, cada post sería un article. En un post del blog, el post y cada uno de sus comentarios sería un <article>.

<aside>

Cualquier contenido que no esté relacionado con el objetivo primario de la página va en un aside. En un blog, obviamente el aside es la barra lateral de información. En el home de un periódico, puede ser el área de indicadores económicos.

<footer>

Este es obvio. Es el pie de página y todo lo que lo compone.

Otras Etiquetas Importantes

Las etiquetas semánticas, a pesar de ser claves para posicionamiento en buscadores y buen desarrollo web, no son la razón por la que todo el mundo habla de HTML5. Video, audio y animación vectorial están en la lista de prioridades y en la boca de todas las personas que evangelizan su uso. Específicamente, las nuevas etiquetas son:

<video>

Insertar video sin necesidad de plugins. Es muy fácil usarla, pero cada navegador soporta codecs diferentes de video, lo que hace necesario recodificar un video en múltiples codecs.

<audio>

Lo mismo que video, pero sin video. Puede usar múltiples formatos, en especial mp3, pero también depende del navegador.

<input *>

Input ya existía como la etiqueta para insertar cajas de texto y botones. Ahora es más poderosa, con la capacidad de insertar cajas tipo "email" que se autovalidan, calendarios tipo "date", sliders, números, entre otras.

<canvas>

Un área de dibujo vectorial y de bitmaps con Javascript. Es un API de dibujo entero para Javascript.

<svg>

Una etiqueta, igual que <img>, para insertar dibujos y animaciones vectoriales al estilo de Flash. Todo basado en el estándar abierto SVG (Scalable Vector Graphics), derivado de XML.

Soporte en Navegadores

Internet Explorer, es tal vez la principal razón por la cual la adopción de HTML5 al principio fue poca. Sin embargo, ahora existen formas para que las etiquetas semánticas de HTML5 y atributos de CSS3 funcionen en IE:

- **HTML5 Enabling Script**: Permite usar las etiquetas semanticas dentro de IE6, 7 y 8 como si fueran divs normales, estilizables por CSS. Sin este script, las etiquetas son ignoradas en IE y es imposible agregarles diseño a ellas o cualquier elemento dentro de ellas.
- **IE-CSS3**: Usando arcanas técnicas (DirectX y VML), este raro script permite usar cosas como bordes

redondeados y sombras sobre objetos de CSS3 en IE6, 7 y 8. Es magia negra, les digo!
- **Modernizr**: Cuando todo falla, con Modernizr puedes detectar si el navegador tiene soporte para multiples capacidades de Javascript, HTML5 y CSS3. Si no, tu mismo puedes codear la solución o alternativa.

Gracias a estos scripts podemos ser desarrolladores felices.

Nuevas Capacidades de Javascript

Javascript, el lenguaje favorito del desarrollador de frontend y experiencias de usuario en la web, ha recibido muchas habilidades nuevas. Esta es una lista de las más importantes.

Web Storage

Una cookie es la forma más casposa de guardar información en el lado del cliente. También es la única forma. Las cookies no pueden guardar más de 4KB por cookie, 100KB por dominio. Muy poco. Pero al mismo tiempo, todo el contenido de las cookies va pegado a cualquier petición HTTP que hagas al servidor. Lo que significa que por cada vez que el usuario recarga la página o baja una imagen, tiene que subir los KB que pesan todas las cookies que le hemos dado.

Web Storage soluciona este problema. Son variables que puedes guardar en el disco del usuario, con soporte en todos los navegadores (incluyendo IE8), puedes guardar hasta 5MB y no sólo texto. Cualquier tipo de datos cabe en un Storage.

Web Workers

¿Ustedes sabían que Javascript sólo puede hacer una cosa al tiempo? Gran parte de la razón por la que Wave falló y las web apps son simples es porque la multitarea es imposible. Web Workers soluciona eso. Web Workers permite tener multiples .js corriendo en paralelo en una misma página. Haciendo tareas complejas más veloces gracias al multithreading. Para más información acerca de Websockets visita este **Link**

Web Sockets

Igual que XMLSockets en Actionscript, Web Sockets permite hacer aplicaciones multiusuario en tiempo real, como juegos, chats, notificaciones, etc. Si el navegador no tiene soporte de Web Sockets, es posible usar implementaciones multiuser en Javascript como PubSubHubBub. Para más información acerca de Websockets visita este **Link**

Arrastrar y soltar

Vete a gmail, crea un email e intenta arrastrar un archivo del explorador de archivos al mail. Verás que es posible adjuntarlo con sólo arrastrarlo. El gesto de arrastrar y soltar ahora es posible gracias a HTML5. Puedes traer trozos de datos o archivos enteros.

Geolocalización

El navegador hará uso de muchos métodos (GPS, Skyhook, Google Geo, IP) para darte la latitud y longitud de tus usuarios. Obviamente, ellos tienen que dar permiso. Lo mejor es que funciona en cualquier PC, no sólo en teléfonos ¿No me crees? Prueba este demo y dime en los comentarios qué tan cerca de ti te

encontró.

Web Semántica

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante.

Objetivo

La Web ha cambiado profundamente la forma en la que nos comunicamos, hacemos negocios y realizamos nuestro trabajo. La comunicación prácticamente con todo el mundo en cualquier momento y a bajo coste es posible hoy en día. Podemos realizar transacciones económicas a través de Internet. Tenemos acceso a millones de recursos, independientemente de nuestra situación geográfica e idioma. Todos estos factores han contribuido al éxito de la Web. Sin embargo, al mismo tiempo, estos factores que han propiciado el éxito de la Web, también han originado sus principales problemas: sobrecarga de información y heterogeneidad de fuentes de información con el consiguiente problema de interoperabilidad.

La Web Semántica ayuda a resolver estos dos importantes problemas permitiendo a los usuarios delegar tareas en software. Gracias a la semántica en la Web, el software es capaz de procesar su contenido, razonar con este, combinarlo y realizar deducciones lógicas para resolver problemas cotidianos automáticamente.

Componentes Principales

Para obtener esa adecuada definición de los datos, la Web Semántica utiliza esencialmente **RDF**, SPARQL, y **OWL**, mecanismos que ayudan a convertir la Web en una infraestructura global en la que es posible compartir, y reutilizar datos y documentos entre diferentes tipos de usuarios.

- RDF proporciona información descriptiva simple sobre los recursos que se encuentran en la Web y que se utiliza, por ejemplo, en catálogos de libros, directorios, colecciones personales de música, fotos, eventos, etc.
- SPARQL es lenguaje de consulta sobre RDF, que permite hacer búsquedas sobre los recursos de la Web Semántica utilizando distintas fuentes datos.
- OWL es un mecanismo para desarrollar temas o vocabularios específicos en los que asociar esos recursos. Lo que hace OWL es proporcionar un lenguaje para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas. Las ontologías, que se encargan de definir los términos utilizados para describir y representar un área de conocimiento, son utilizadas por los usuarios, las bases de datos y las aplicaciones que necesitan compartir información específica, es decir, en un campo determinado como puede ser el de las finanzas, medicina, deporte, etc. Las ontologías incluyen definiciones de conceptos básicos en un campo determinado y la relación entre ellos.

Fuentes y Recursos

http://www.htmlya.com.ar/
http://www.mclibre.org/consultar/amaya/index.html
http://www.librosweb.es/xhtml/
http://www.cristalab.com/tutoriales/introduccion-a-html5-c92171l/
http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica
http://www.w3.org/html/logo/
http://joshduck.com/periodic-table.html
http://switchtohtml5.com/
http://www.html5rocks.com/en/tutorials/

HTML5 Browsers Support: http://caniuse.com/ , http://html5test.com/