

Certificado HTML 5 para móviles.



Manual del participante

2014





Índice

Índice	3
Formación de la comunidad de aprendizaje	4
Objetivo del curso	5
Objetivo.....	5
Introducción	5
Unidad 1 Introducción	7
Historia de HTML y los estándares web	7
La W3C.....	9
El nuevo estándar HTML5	10
Restricciones y advertencias sobre HTML5	11
Unidad 2 Sintaxis.....	12
Elementos nuevos en HTML 5	12
Elementos desechados HTML 4	12
Unidad 3 Estructura de HTML 5.....	13
Estructura	13
Validación.....	16
Unidad 4 Funcionalidades.....	18
Web Forms	18
Audio y video	19
Canvas	20
Contenidos semánticos.....	23
Geolocalización	24
Unidad 5 Primeras Aplicaciones.....	26
Desarrollo del contenido.....	26
Javascript.....	26
CSS	27
Despliegue en diversos navegadores	27
Conclusiones finales	28
Fuentes de consulta	30
Cibergrafía.....	30

Formación de la comunidad de aprendizaje

La formación de la comunidad de aprendizaje es un proceso que debe llevarse a cabo para iniciar cada uno de nuestros cursos.

Su finalidad es crear un clima propicio para la celebración de la actividad instruccional, es decir, generar un entendimiento previo entre el instructor y los participantes sobre los temas que se desarrollarán durante ésta, así como las estrategias educativas que se llevarán a cabo para lograr un mejor aprendizaje.

Un adecuado manejo de la comunidad de aprendizaje es un elemento fundamental para garantizar la satisfacción de uno de los clientes involucrados en la impartición de los cursos: los participantes.

Presentación del Instructor:

- ★ Nombre, profesión, años de experiencia como instructor, experiencia en la impartición del curso, o cursos similares o relacionados.

Alineación de expectativas:

- ★ El instructor recabará las expectativas de los participantes respecto al curso, con el fin de dejarles claro el objetivo del mismo.
- ★ En caso de que alguna expectativa no coincida con los temas que el curso contiene, el instructor dejará claro cuáles de las expectativas expresadas no serán cubiertas con el curso y porqué.
- ★ Las expectativas alineadas serán anotadas en hojas de rotafolio para su revisión al término del curso.
- ★ Durante el desarrollo del curso el instructor deberá cubrir las expectativas alineadas.

Presentación del objetivo del curso:

- ★ El instructor presentará a los participantes el objetivo del curso, aclarando dudas al respecto si las hubiese.

Reglas de oro:

- ★ El instructor promoverá el establecimiento de reglas por parte de los participantes que se observarán a través del curso; por lo que puede proponer: tiempo de tolerancia para iniciar las sesiones, respeto hacia los compañeros, participación de todos en técnicas y ejercicios grupales, etc.; se incluirán todos los puntos que los participantes consideren pertinentes.
- ★ Se anotarán los acuerdos en hojas de rotafolio y se colocarán en un espacio en el que sean visibles a lo largo de todo el curso.

Cumplimiento de expectativas

- ★ Al finalizar el curso el instructor deberá llevar a cabo una revisión de las expectativas alineadas que se anotaron en hojas de rotafolio al inicio del curso
- ★ Se revisará cada una de las expectativas alineadas palomeando las que hayan sido cumplidas, y el instructor explicará de qué manera se llevó a cabo tal cumplimiento.

Objetivo del curso

Objetivo

Al final del curso el asistente podrá desarrollar y desplegar contenidos en dispositivos móviles aprovechando las ventajas de HTML 5.

Introducción

El curso se enfoca en el desarrollo de aplicaciones para dispositivos móviles utilizando estándares abiertos como los son la nueva especificación HTML5, CSS3 y JavaScript.

HTML 5 es la base de la web 2.0 Geolocalización, video, audio, streaming, nuevos componentes y etiquetas. Este curso sirve de introducción para el participante en el desarrollo de este tipo de aplicaciones, empezando por el aprendizaje de los componentes HTML necesarios e incidiendo en las nuevas características. Si unimos las nuevas características de CSS, además de complejidad se puede dotar a nuestras aplicaciones de un diseño impresionante.



Unidad 1 Introducción

HTML, siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Es el lenguaje con el que se definen las páginas web.

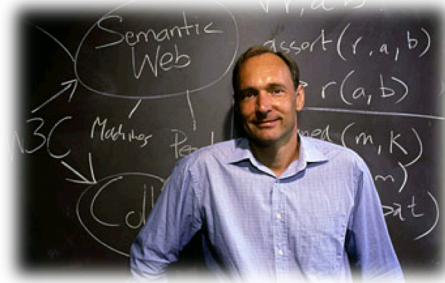
El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.



Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). Sin embargo, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Usualmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores aún mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, aún existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

Historia de HTML y los estándares web

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, quien trabajaba en el Centro Europeo de Investigaciones Nucleares CERN 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. Inicialmente fue desarrollado para que se pueda compartir fácilmente información entre científicos de distintas universidades e institutos de investigación de todo el mundo.

Tim Berners-Lee en 1991 describe 22 elementos que incluyen el diseño inicial y relativamente simple de HTML. Trece de estos elementos todavía existen en HTML 4.

Berners-Lee consideraba a HTML una ampliación de SGML, pero no fue formalmente reconocida como tal hasta la publicación a mediados de 1993, por la IETF, de una primera proposición para una especificación de HTML: el boceto *Hypertext Markup Language* de Berners-Lee y Dan Connolly, el cual incluía una Definición de Tipo de Documento SGML para definir la gramática. El boceto expiró luego de seis meses, pero fue notable por su reconocimiento de la etiqueta propia del navegador Mosaic usada para insertar imágenes sin cambio de línea, que reflejaba la filosofía del IETF de basar estándares en prototipos con éxito. De la misma manera, el boceto competidor de Dave Raggett *HTML+ (Hypertext Markup Format)* (Formato de Marcaje de Hipertexto), de finales de 1993, sugería estandarizar características ya implementadas, como las tablas.

El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre HTML Tags (Etiquetas HTML)

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, adoptaba 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,¹⁸ 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.

La W3C

El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones Miembro, personal a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web Tim Berners-Lee y el Director Ejecutivo (CEO) Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial.

¿Qué hace el W3C?

La principal actividad del W3C es desarrollar protocolos y directrices que aseguren el crecimiento de la Web a largo plazo. Los estándares del W3C definen las partes claves que hacen que la World Wide Web funcione.

¿Dónde está el W3C?

El W3C no sólo tiene una sede física. Existen tres instituciones que "albergan" al W3C: MIT (en Cambridge, Massachusetts, EEUU), ERCIM (en Sophia-Antipolis, Francia) y la Universidad de Keio (cerca de Tokio, Japón).

El equipo del W3C está distribuido por todo el mundo, pero muchas de estas personas se concentran en Cambridge, Massachusetts (EEUU), Sophia-Antipolis (Francia) y Tokio (Japón). Además, el W3C está representado en otras 17 regiones del mundo a través de representantes que se basan en organizaciones. El W3C llama a estos puntos "Oficinas del W3C."

- **Validador**

<http://validator.w3.org/>

- **Tutoriales**

<http://www.w3schools.com/>

- **Apartado W3C CSS**

<http://www.w3.org/Style/CSS/>

El nuevo estándar HTML5

La controversia recae en que ciertas compañías actúan como si HTML5 fuera el estándar, cuando la realidad es que ningún navegador lo soporta al 100% todavía.

La revisión incluye:

- ★ Las etiquetas multimedia para audio y video con soporte a distinto códec.

```
<video src="http://v2v.cc/~j/theora_testsuite/320x240.ogg" controls>  
  Tu navegador no implementa el elemento      <code>video</code>.  
</video>
```

- ★ La etiqueta canvas para dibujar contenido en el navegador (2D y 3D) (Recomendado: Cómo dibujar curvas cuadráticas en un Canvas de HTML5)

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">  
  Your browser does not support the HTML5 canvas tag.  
</canvas>
```

- ★ Las etiquetas datagrid, details, menu y command, que permiten manejar grandes conjuntos de datos

```
<datagrid>
<ol>
<li> datagrid row 0 </li>
<li> datagrid row 1
  <ol style="list-style-type:lower-alpha;">
    <li> datagrid row 1,0 </li>
    <li> datagrid row 1,1 </li>
  </ol>
</li>
<li> datagrid row 2 </li>
</ol>
</datagrid>
```

- ★ **Formularios más inteligentes que nos permitirán hacer cosas como validación de llenado mediante el uso de atributos requeridos, a través de los nuevos tipos email, number, url, datetime, etc. (Recomendado: Cómo usar campos de voz con HTML5)**

```
<input type="text" name="nombre" required/>
```

- ★ **APIs que permiten Arrastrar y Soltar objetos como imágenes, trabajar Off-Line, utilizar Geoposicionamiento, Almacenamiento persistente con base de datos local, Websockets y más.**

```
navigator.geolocation.getCurrentPosition(show_map)
```

Restricciones y advertencias sobre HTML5

La especificación de HTML 5 no ha finalizado, aún pueden sucederse cambios.

No todo funciona en todos los navegadores.

Las aplicaciones para iOS eran lentas e inestables. La apuesta de Facebook por HTML5 no resultó tan exitosa como esperaban, al final se optó por desarrollar aplicaciones nativas para iOS y Android y así depender menos de HTML5.

Unidad 2 Sintaxis

Es una sintaxis mucho más relajada que la impuesta por el XHTML y permite tanto mayúsculas como minúsculas como combinaciones de ambas en nombres de etiqueta y de atributo.

Permite parametrizar los atributos sin comillas, siempre que el valor del atributo no contenga espacios, " ' ` = < ó >.

Los siguientes ejemplos son igualmente válidos:

- ✓ ``
- ✓ ``
- ✓ ``
- ✓ ``
- ✓ ``
- ✓ ``
- ✓ ``

Elementos nuevos en HTML 5

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas `<div>` y ``, pero tienen un significado semántico, como por ejemplo `<nav>` (bloque de navegación del sitio web) y `<footer>`.

Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos `<audio>` y `<video>`. Mejora el elemento `<canvas>`, capaz de renderizar elementos 3D en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE).

Elementos desechados HTML 4

- `<acronym>`
- `<applet>`
- `<basefont>`
- `<big>`
- `<center>`
- `<dir>`
- ``
- `<frame>`
- `<frameset>`
- `<hgroup>`
- `<isindex>`
- `<noframes>`

- `<s>`
- `<strike>`
- `<tt>`
- `<u>`
- `<xmp>`

Unidad 3 Estructura de HTML 5

Estructura

Estructura básica de una página en HTML5

En HTML5 la estructura interna básica de una página web se ha simplificado, reduciendo el código innecesario hasta quedarse con el esqueleto básico, que sería el siguiente:

```
<!doctype html>
<html>

  <head>
    <meta charset="utf-8"/>
    <title>Título de la web</title>
  </head>

  <body>
    Contenido de la web
  </body>

</html>
```

Cada fichero HTML está compuesto de dos partes. El `<head>`, donde se añade toda la información que el navegador necesita pero que no se visualizará y el `<body>`, que será toda la información que contiene la página y que el navegador visualizará.

`<head>`

Todo lo que coloquemos entre las etiquetas del `<head>` y `</head>`, aunque no se visualizará en el navegador, ofrece información importante sobre la propia página, por lo que en el `<head>` se introduce información referente a:

- la codificación de la página (para que aparezcan los acentos correctamente) utilizando 'meta charset'.
- el título de la página, que aparece en la pestaña del navegador (a través de 'title').
- La descripción de la temática de la página para los buscadores (Google entre otros).
- el estilo CSS que la página utilizará (lo trataremos en el tema: Introducción a CSS3).

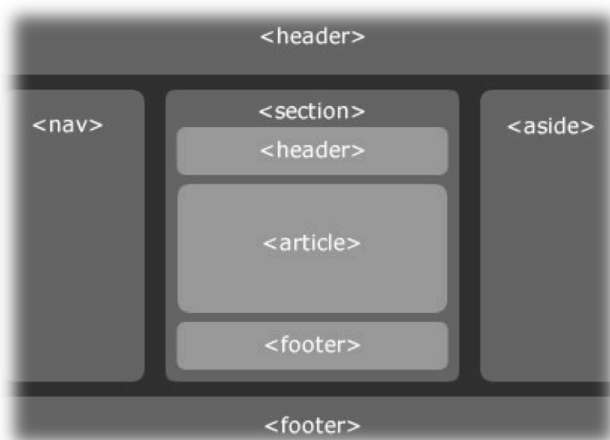
- la programación en JavaScript (en el caso de que se utilice).

Por otra parte, dentro del <body> colocaremos todo aquello que sí se tiene que visualizar en el navegador, entre lo que vamos a destacar las etiquetas semánticas.

Las nuevas etiquetas (semánticas)

Una de las novedades de HTML5 hace referencia a las etiquetas semánticas, que tienen como objetivo crear cajas o apartados de contenido dentro de la página pudiendo aportar un significado semántico interno dependiendo del contenido que contenga. Así, si un apartado tiene como nombre `<header>`, el navegador sabrá que se trata de un encabezado, o si por el contrario se utiliza la etiqueta `<nav>` identificará que es una barra de navegación y que por lo tanto es donde se encuentran los enlaces de la página.

Así, utilizando las etiquetas semánticas, se puede identificar el contenido que cada apartado de la página contiene, huyendo de esta manera del abusivo uso que se hace del omnipresente DIV (que no aporta ninguna información adicional sobre la información de la información que contiene).



Las etiquetas semánticas más importantes son:

- **Header**
- **Nav**
- **Section**
- **Article**
- **Aside**
- **Footer**

A continuación las describiremos:

Header: Es el equivalente a la cabecera de la página web. Contiene el título o nombre de la empresa/titular de la página, logo e información relacionada.

Nav: Este apartado contiene los enlaces (barra de navegación) externos o internos de la página.

Section: Este gran apartado puede agrupar diferentes subapartados (de tipo 'article') de diferentes temas, o bien puede definir un gran apartado de contenido unitario.

Article: Es una pieza independiente de contenido, que puede estar contenida (o no) dentro de un apartado de tipo 'section'.

Aside: Define un bloque de contenido (tangencial) relacionado con el contenido principal, pero que no es esencial para la comprensión del mismo.

Footer: Equivale al pie de página de un apartado concreto ('section' o 'article') o de la página web en general.

Para seleccionar el nombre de estas etiquetas, Ian Hickson (el editor de HTML5) recurrió a las herramientas de Google para extraer datos de más de mil millones de páginas web

reales, repasando los ID y los nombres de clase más utilizados en la web. Si quieres más información puedes acceder a **Google Code: Web Authoring Statistics: Classes**.

Estructura compleja

Una estructura HTML que contenga referencia a un CSS externo, contenga el bloque de código CSS interno y tenga las etiquetas semánticas básicas podría ser el siguiente:

```
<!doctype html>
<html>

  <head>
    <meta charset="utf-8"/>
    <meta name="description" content="Resumen del contenido de la página">
    <title>Título de la página</title>
    <link rel="stylesheet" href="css/estilo.css" type="text/css"/>
    <style type="text/css">
    </style>
  </head>

  <body>
    <header>
    </header>

    <section id="contenido">
      Contenido de la web
    </section>

    <footer>
    </footer>
  </body>

</html>
```

Validación

Testea qué tan compatible es tu navegador con HTML 5, con la siguiente herramienta:

- **HTML5 Test** es un sitio que ejecuta un test para ver cuán apto es tu navegador con el nuevo HTML5. Sólo tienen que entrar al sitio y esperar unos segundos a que se ejecute el test y nos de los resultados, ofreciendo un sistema de puntaje hasta 400 puntos dependiendo de cuantas características de HTML5 soporta tu navegador.

<http://html5test.com/>



Por ejemplo, **Chrome 14.0** fue el ganador con 340 puntos de 400 posibles. Este navegador ha implementado la mayoría de la funcionalidad para ser compatible con **HTML5**. Adicionalmente, muchas de las características que no han sido añadidas ya son soportadas parcialmente. Así que **Google Chrome** está por mucho al frente de la carrera.



En seguida se encuentra **Mozilla Firefox 7.0** con 313 puntos, Todavía les faltan ciertos elementos, pero obtuvo puntos bonus por su excelente soporte a video y sus reglas de parsing.

Internet Explorer, como de costumbre, está muy por detrás de los demás. Con su versión **Internet 8** obteniendo 32 puntos de 400, y en su versión 9 obteniendo 141 puntos. Es muy mala calificación para lo que alguna vez fue el navegador más popular del mundo. **Internet Explorer** ha estado tratando de acatar los estándares internacionales en cuanto a **CSS** se refiere, sin embargo ha dado mucho que desear, ya que ni siquiera soporta al 100% los estándares **CSS3**, y no parece haber muchos cambios a futuro.



Así que, aquí están las estadísticas. Por el momento el único navegador que te puede llevar lejos es **Google Chrome**, y, hasta que todos los navegadores se actualicen, es muy recomendable usar **HTML5** solo esporádicamente en tus diseños, hasta que realmente sea un verdadero estándar.

Ejercicio.

Analiza 5 sitios web en el validador de w3c

En este ejercicio vamos validar cinco sitios web, con la herramienta que nos proporciona **W3C**.

Sigamos los siguientes pasos:



Unidad 4 Funcionalidades

Web Forms

Nuevos atributos para <input>

- **autocomplete**
- **autofocus**
- **form**
- **formation**
- **formenctype**
- **formmethod**
- **formnovalidate**
- **formtarget**
- **height and width**
- **list**
- **min and max**
- **multiple**
- **color**
- **pattern (regexp)**
- **placeholder**
- **required**
- **step**

search:

number:

range:

color:

tel:

url:

email:

date:

month:

week:

time:

datetime:

datetime-local:

Nuevos atributos para <form>

autocomplete Atributo para autocompletar datos en formularios.

```
<form action="" autocomplete="on">
  Primer nombre: <input type="text" name="nombre" >
  <br/>
  Apellidos: <input type="text" name="apellidos">
  <br/>
  Email: <input type="email" name="email">
  <br/>
  Fecha de nacimiento: <input type="date" name="fecha">
  <br/>
  Número Favorito 1-5 <input type="number" name="quantity" min="1"
max="5">
  <br/>
  <input type="submit" value="Enviar">
```

novalidate se puede establecer en elementos form que tienen restricciones de validación para permitir el envío del formulario sin que se realice una validación.

```
<form action="" novalidate >
  Primer nombre: <input type="text" name="nombre" required>
  <br/>
  Apellidos: <input type="text" name="apellidos">
  <br/>
  Email: <input type="email" name="email">
  <br/>
  <input type="submit" value="Enviar">
</form>
```

Audio y video

Anteriormente no existía un standard para manejar archivos de audio en un web site. Solo se podía manejar audio con ayuda de plug-in

HTML 5 ha implementado una etiqueta para el manejo de archivos de audio **<audio>**

Ejemplo

```
<audio controls>
  <source src="tesone.mp3" type="audio/mpeg">
  Tu navegador no soporta html5
</audio>
```

Nota: Soporte de archivos MP3, Wav y Ogg

Atributos de la etiqueta <audio>

- ✓ **autoplay** Especifica si el audio comenzará a ejecutarse tan pronto como esté listo.
- ✓ **controls** Especifica si los controles (inicio/paro) se deben mostrar.
- ✓ **loop** Especifica si el audio debe comenzar de nuevo una vez haya terminado.
- ✓ **preload** Especifica cómo se debe cargar el audio cuando se cargue la página.
- ✓ **src** Especifica la URL del origen del audio.



HTML 5 ha implementado una etiqueta para el manejo de archivos de video **<video>**

Atributos de la etiqueta **<video>**

- ✓ **autoplay** Especifica si el vídeo comenzará a ejecutarse tan pronto como esté listo.
- ✓ **controls** Especifica si los controles (inicio/paro) se deben mostrar.
- ✓ **height** Especifica la altura del reproductor de vídeo.
- ✓ **loop** Especifica si el vídeo debe comenzar de nuevo una vez haya terminado.
- ✓ **muted** Especifica si el audio del vídeo debe estar apagado.
- ✓ **poster** Especifica una imagen para ser mostrada mientras el vídeo se carga y hasta que el usuario pulse el botón de inicio.
- ✓ **preload** Especifica cómo se debe cargar el vídeo cuando se cargue la página.



Ejemplo

```
<video width="320" height="240" controls>  
  <source src="buenfin.mp4" type="video/mp4">  
  Tu navegador no soporta video  
</video>
```

Nota: Soporte de archivos MP4, WebM y Ogg



Canvas

El elemento canvas (lienzo) puede definirse como un entorno para crear imágenes dinámicas en el navegador (2D y 3D) (Recomendado: Cómo dibujar curvas cuadráticas en un Canvas de HTML5)

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">  
Your browser does not support the HTML5 canvas tag.  
</canvas>
```

El potencial de canvas reside en su habilidad para actualizar su contenido en tiempo real. Si usamos esa habilidad para responder a eventos de usuario, podemos crear herramientas y juegos que anteriormente a la nueva especificación hubiesen requerido de un plugin externo como Flash.

Analiza el siguiente código



```
<canvas id="micanvas" width="350" height="350" style="border: 1px solid #000000;">
Tu navegador no soporta canvas.
</canvas>
<script>
var miCanvas=document.getElementById("micanvas");
var cxt=miCanvas.getContext("2d");
cxt.fillStyle="blue";
cxt.fillRect(10,10,100,50);
</script>
```

Puedes consultar los siguientes ejemplos de canvas en 3D

- <http://www.kevs3d.co.uk/dev/html5logo/>
- http://www.kevs3d.co.uk/dev/canvask3d/k3d_test.html
- <http://seb.ly/demos/canvas3d/canvas3d2.html>
- <http://seb.ly/demos/GravityParticles/ParticlesForces3D2.html>

Práctica la elaboración de canvas en <http://www.html5.com.ar/html5/>

Ejercicio

Actividad: Dibujando HTML5 con HTML5

En este ejercicio vamos a dibujar.

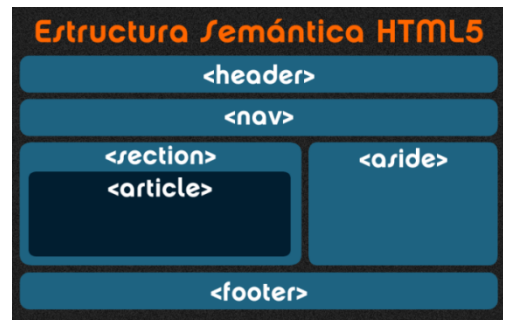
Sigamos los siguientes pasos:

Contenidos semánticos

La semántica es una de las funciones en las que más se diferencia la plataforma web de otras plataformas de aplicaciones. Los desarrolladores suelen ignorar esta función o restarle importancia, pero su dominio puede reportar muchos beneficios para los proyectos.

Pues te permite saber qué contenidos se han colocado en la cabecera, elemento HEADER, cuáles son los temas de los que se habla en una web, los ARTICLES, o cuáles son los enlaces a las secciones principales del sitio, colocados en la etiqueta NAV.

Podemos definir un esquema de documento HTML utilizando los elementos h1 a h6: , por ejemplo:



```

<h1>Genbeta Dev</h1>
<h2>Editores</h2>
<p>Sigue todos los posts de nuestros editores en Genbeta Dev</p>
<h3>Txema Rodriguez</h3>
<p>Apasionado por la tecnología trabaja en Madrid...</p>
<h3>Jorge Ruvira</h3>
<p>Ingeniero Técnico de informática de sistemas por la...</p>
<h3>Carlos Paramio</h3>
<p>Tiene 33 años y vive en la ciudad de Algeciras Cádiz...</p>
<small>Los editores son todos unos frikis del quince</small>
  
```

Lo anterior generaría este esquema

Genbeta Dev

- _ Editores
 - _ Txema Rodriguez
 - _ Jorge Ruvira
 - _ Carlos PArmio

¿Por qué hacer semántica?

La razón es que las etiquetas p y h1 transmiten información extra acerca del contenido. Estas dicen “esto es un párrafo” y “esto es un título en el primer nivel”, respectivamente.

El autor hace todo el esfuerzo para asegurar que el etiquetado organiza y estructura el contenido.

Geolocalización



La API de geolocalización permite al usuario compartir su ubicación a las aplicaciones web si así lo desea. Por razones de privacidad, al usuario se le pide que confirme el permiso para proporcionar información de ubicación.

Para obtener la ubicación actual del usuario, puede llamar al método `getCurrentPosition()`.

Esto inicia una solicitud asíncrona para detectar la posición del usuario, y consulta el hardware de posicionamiento para obtener información actualizada. Cuando se determina la posición, la función de callback es ejecutada. Si lo desea, puede proporcionar otra función de callback que se ejecuta si se produce un error. Un tercer parámetro opcional, es un objeto de opciones donde se puede establecer la edad máxima de la posición devuelta, el tiempo de espera para una solicitud y si se requiere una alta precisión para la posición.

Cuando `getCurrentPosition()` ejecutan la función de éxito, le envían a ésta un objeto con dos propiedades, `timestamp` y `coords`.

timestamp es una marca de tiempo expresada en milisegundos, que indica el momento en que se ha obtenido la ubicación.

A su vez **coords** es un contenedor para las propiedades que más nos interesan:

- ★ **latitude:** La latitud de la posición en grados decimales. Un valor positivo equivale a «latitud norte», y uno negativo a «latitud sur».
- ★ **longitude:** La longitud de la posición en grados decimales. Un valor positivo equivale a «longitud este», y uno negativo a «longitud oeste».
- ★ **altitude:** La altitud, expresada en metros.
- ★ **accuracy:** La precisión de la estimación de la latitud y la longitud, expresada en metros; indica la distancia de error máxima entre la posición estimada y la posición real.
- ★ **altitudeAccuracy:** La precisión de la estimación de la altitud, expresada en metros; indica la distancia de error máxima entre la altura estimada y la altura real.
- ★ **heading:** Indica la dirección del movimiento en comparación con una ubicación anterior, y se expresa en grados decimales con respecto al norte real, en el sentido de las agujas del reloj —es decir, que 45, por ejemplo, significa noreste—. Como se puede intuir, esta propiedad sólo tendrá valor si se está obteniendo la ubicación con `watchPosition()`.
- ★ **speed:** La velocidad del movimiento, expresada en metros por segundo. Como **heading**, sólo tendrá valor en el objeto de posición que devuelva `watchPosition()`.

La localización por wifi se produce por la triangulación de los puntos donde se encuentran los routers que tenemos a nuestro alrededor, seguro que hemos visto alguna vez al coche de Google, ¿verdad?, pues también lleva una antena wifi escaneando los puntos de acceso y registrando su ubicación.



Si queremos estas variables para situar en nuestra web un mapa, por ejemplo, podemos consultar el API de google para obtener uno, pasándole estos parámetros.

Ejemplo de Geolocalización

Actividad: Localizándote

En este ejercicio vamos a crear una aplicación de tracking sencilla utilizando la API de geolocalización de HTML5. Vamos a recuperar las coordenadas desde el API y los vamos a mostrar en un mapa.

Sigamos los siguientes pasos:

Unidad 5 Primeras Aplicaciones

Desarrollo del contenido

Cuando HTML5 se combina con JavaScript y CSS3, puede hacer cosas bastante curiosas, y esta afirmación es especialmente cierta en el desarrollo de aplicaciones móviles.

Las principales plataformas móviles, incluyendo iOS (Apple), Android y Blackberry tienen navegadores que son compatibles con HTML5. Los desarrolladores que crean aplicaciones web HTML5 para equipos de sobremesa, pueden utilizar el mismo código para el desarrollo de una aplicación para iPad o cualquier otra aplicación para tablets.

Javascript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico.

Javascript es lo que se utiliza en los sitios web para hacer esos efectos molones de sliders, mover bloques de contenido (drag&drop), básicamente cualquier tipo de interactividad entre el usuario y la página web utiliza Javascript, por ejemplo:

- Seguramente en más de una de AJAX. Bien, AJAX significa JavaScript And XML"... sí,  ocasión has oído hablar "Asynchronous Javascript."
- JQuery es un framework de Javascript

Herramientas que proporciona JQuery

El ThemeRoller: es una herramienta web que nos permite crear temas personalizados para nuestras aplicaciones móviles sin necesidad de escribir una sola línea de código CSS. Sólo hay que arrastrar y soltar algunos componentes para crear verdaderas obras de arte. Luego, podemos compartirlas mediante su URL o descargarlas en un archivo ZIP, listas para producción.

<http://themeroller.jquerymobile.com/>

Los Demos: es un marco HTML5 interfaz de usuario táctil optimizada diseñada para hacer que los sitios web sensibles y aplicaciones que se puede acceder a todos los dispositivos smartpone, tablet y de escritorio.

<http://demos.jquerymobile.com/1.4.2/>

El API

<http://api.jquerymobile.com/>

Llevando JQuery Mobile a una App

Android - iOS



CSS



Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentosXML, incluyendo SVG y XUL.

En otras palabras más llanas, cuando entramos en una página web vemos que existen colores de fondo, que la información se muestra en columnas, que los enlaces tienen un color y al pasar sobre él con el ratón cambian de color, que existen menús desplegables, etc. Todo esto se consigue con CSS.

Despliegue en diversos navegadores

Dado que HTML5 llama cada vez más la atención a fin de marcar nuestras las tendencias de las nuevas páginas la única manera de conseguir que IE reconozca los nuevos elementos, como <article>, <section>, etc. es utilizar el shiv (navaja) HTML5

<https://code.google.com/p/html5shiv/>

Conclusiones finales

Está claro que HTML5 ha dejado ya de ser el futuro para convertirse en el presente. Ya la gran mayoría de navegadores modernos soportan gran cantidad de los nuevos elementos del estándar lo que hace que su uso se ya posible hoy día. Pero HTML5 no es lo único que ha evolucionado, conjunto a él ha crecido CSS3 y la nueva generación de JavaScript.

La nueva versión de CSS nos va a permitir que el diseño de páginas web cambie por completo, esta nueva versión nos va a permitir hacer cosas que antes eran impensables, podremos incluso dibujar con CSS, hacer que nuestras webs cobren vida, y otras muchas cosas. Esto hará que los diseñadores que quieran mantenerse al día tenga que evolucionar, tenga que cambiar el concepto de hacer diseños web como si fueran artistas y pasar a hacer web como si fueran programadores haciendo uso de las nuevas herramientas que nos ofrece CSS3. Lo que está claro que la tecnología avanza y que todo el que se dedique al desarrollo web, ya sean programadores o diseñadores tendrán que ir de la mano de la evolución para no quedarse estancado en el pasado.





Fuentes de consulta

Cibergrafía

Autor:	Saúl González Fernández
Título:	Pruebas para el futuro inmediato: geolocalización
Vínculo:	http://digitalicon.es/blog/pruebas-futuro-inmediato-geolocalizacion/
Editor:	

Autor:	Diego García y Rubén Lara
Título:	Using geolocation
Vínculo:	https://developer.mozilla.org/es/docs/WebAPI/Using_geolocation
Editor:	Mozilla Foundation

Autor:	Álvarez Miguel Ángel
Título:	Introducción al HTML
Vínculo:	http://www.desarrolloweb.com/articulos/534.php
Editor:	Desarrollo Web

Autor:	López Javier
Título:	Estructura semántica de HTML5
Vínculo:	http://www.jabmultimedia.com/t1_estructura.html
Editor:	Jabmultimedia

Autor:	W3C
Título:	Sobre el W3C
Vínculo:	http://www.w3c.es/Consortio/
Editor:	W3C

Autor:	Wikipedia
Título:	HTML 5
Vínculo:	http://es.wikipedia.org/wiki/HTML5#Nuevos_elementos
Editor:	Wikipedia