

**Certificado HTML 5 para móviles.**

****

**Manual del participante**

**2014**

# Índice

[Índice 2](#_Toc403145472)

[Formación de la comunidad de aprendizaje 4](#_Toc403145473)

[Objetivo del curso 5](#_Toc403145474)

[Objetivo 5](#_Toc403145475)

[Introducción 5](#_Toc403145476)

[Unidad 1 Introducción 6](#_Toc403145477)

[Historia de HTML y los estándares web 7](#_Toc403145478)

[La W3C 9](#_Toc403145479)

[El nuevo estándar HTML5 10](#_Toc403145480)

[Restricciones y advertencias sobre HTML5 11](#_Toc403145481)

[Diseño y desarrollo web adaptado a móviles 11](#_Toc403145482)

[Ventajas de HTML5 para las plataformas móviles 13](#_Toc403145483)

[Navegadores para móviles 13](#_Toc403145484)

[Cómo saber si el móvil está preparado para HTML5 16](#_Toc403145485)

[Técnicas de detección 16](#_Toc403145486)

[Unidad 2 Sintaxis 17](#_Toc403145487)

[Contenidos semánticos 17](#_Toc403145488)

[Nuevos elementos semánticos / Estructural 19](#_Toc403145489)

[Nuevos para medios de comunicación / Media 20](#_Toc403145490)

[El nuevo elemento <canvas> 20](#_Toc403145491)

[Nuevos elementos de formulario 20](#_Toc403145492)

[Elementos eliminados de la versión anterior de HTML 20](#_Toc403145493)

[Unidad 3 Estructura de HTML 5 22](#_Toc403145494)

[Estructura 22](#_Toc403145495)

[Validación 27](#_Toc403145496)

[Unidad 4 Funcionalidades 31](#_Toc403145497)

[Web Forms 31](#_Toc403145498)

[1 – Elemento <datalist>: 32](#_Toc403145499)

[Los nuevos atributos: 32](#_Toc403145500)

[2 – Elemento <output>: 34](#_Toc403145501)

[3 – Elemento <keygen>: 34](#_Toc403145502)

[Audio y video 35](#_Toc403145503)

[1 – Múltiples Fuentes para mostrar el video 35](#_Toc403145504)

[Atributos de la etiqueta video 36](#_Toc403145505)

[1 – Múltiples Fuentes para la etiqueta audio 38](#_Toc403145506)

[2 – Atributos de la etiqueta audio 39](#_Toc403145507)

[Canvas 41](#_Toc403145508)

[Creación de un elemento Canvas 41](#_Toc403145509)

[1 – Dibujar Rectángulos Con Canvas 44](#_Toc403145510)

[2 – Trazar Una Línea Con Canvas 46](#_Toc403145511)

[3- Trazar O Dibujar Una Curva Bézier Con Canvas 47](#_Toc403145512)

[4 – Dibujar Círculos Con Canvas 48](#_Toc403145513)

[1 – Comencemos dibujando en SVG 50](#_Toc403145514)

[Geolocalización 53](#_Toc403145515)

[Busque la posición del usuario 54](#_Toc403145516)

[Soporte del navegador 54](#_Toc403145517)

[HTML5 - El uso de Geolocalización 54](#_Toc403145518)

[Gestión de errores y Rechazos 55](#_Toc403145519)

[Ejemplo 55](#_Toc403145520)

[Visualización del resultado en un mapa 56](#_Toc403145521)

[Ejemplo 56](#_Toc403145522)

[Información específica de la ubicación 56](#_Toc403145523)

[El getCurrentPosition () Método - Devolver datos 56](#_Toc403145524)

[Property 57](#_Toc403145525)

[Description 57](#_Toc403145526)

[Objeto Geolocalización - Otros métodos interesantes 57](#_Toc403145527)

[Ejemplo 57](#_Toc403145528)

[Unidad 5 Primeras Aplicaciones 58](#_Toc403145529)

[Desarrollo del contenido 58](#_Toc403145530)

[Javascript 58](#_Toc403145531)

[CSS 59](#_Toc403145532)

[Despliegue en diversos navegadores 59](#_Toc403145533)

[Conclusiones finales 60](#_Toc403145534)

[Fuentes de consulta 61](#_Toc403145535)

[Cibergrafía 61](#_Toc403145536)

# 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.

Bueno antes que todo estan las preguntas básicas, ¿qué es?, ¿para qué es?, ¿En qué me beneficio?, y no pueden hacer falta en este mundo del diseño y la estructura web y específicamente en este tema de **HTML5**

**¿Qué es html 5?** – Simplemente Como su nombre indica, HTML 5 es el sucesor de HTML4. Este trabajo o proyecto de HTML5 se inició a finales de 2003 con un grupo de trabajo que se propuso a hacer un lenguaje que llegara con un conjunto de tecnologías que permiten construir la nueva Web. No fue sino hasta 2007 que el HTML5 del W3C acepta la visión mediante la incorporación en ella del grupo de trabajo. Los principios de diseño son claras: para simplificar el uso de HTML, la formalización de las prácticas actuales, y garantizar la máxima compatibilidad con versiones anteriores.

**HTML5** es la nueva versión del lenguaje de marcado que se usa para estructurar páginas web, que actualmente todavía sigue en su evolución, gracias a él con características nuevas y modificaciones que mejorará significativamente este nuevo estándar.

**¿Qué va a servir?** En primer lugar, el HTML 4 se utiliza para estructurar un documento. El html5 no cambia este objetivo, solo que tendremos más oportunidad de hacer las cosas más simplificadas y de una manera más sencilla. Eso es todo, ahora podemos crear aplicaciones web de una manera menos complicada ya que esta tecnología mejora e incorpora muchos conceptos de programación web, como decir un todo en uno.

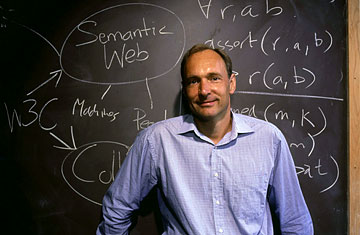
**¿Por qué es mejor HTML5?** Esta es una pregunta muy importante que no requiere una cátedra para dar una respuesta, Simplemente porque es una tecnología que supera a la actual HTML, porque es lo nuevo que estandariza la W3C, porque es nueva tecnología y como toda nueva tecnología siempre viene con cosas que nos van a impresionar, porque llega de la mano de CSS3, una evolución notable de las hojas de estilo que conocíamos y porque revaloriza el papel de JavaScript en la Web, como el lenguaje que “sabe hablar” con las nuevas APIs que llegan con **HTML5**.

**¿Por qué debo de comenzar a trabajar con HTML5?**, La respuesta es muy corta: debes de comenzar a crear tus proyectos en HTML5 porque es una inversión en el futuro, ¡¡ NO que digo futuro!! es para ¡¡YA..!!., la mayoría de los sitios web poderosos del mundo ya implementan este estándar, y si no quieres quedarte atrás, debes de comenzar ya.

**HTML5** lleva a un nivel más alto que el código HTML4 el lenguaje de marcado, ya que será capaz de controlar los eventos y las iteraciones con el usuario. La diferencia esencial no está en el lenguaje en sí mismo, se añaden las etiquetas nuevas en comparación con HTML 4 y no requiere un tipo de documento específico, el verdadero “cambio” es en el manejo de la tela, o las áreas de la página que se puede utilizar como pizarras interactivas (se puede diseñar y gestionar los eventos), todos a través de JavaScript.

En resumen, **HTML5** conduce a una fusión entre JavaScript como lenguaje de programación, HTML como modelo semántico y css3 que es la evolución del css como el lenguaje de los estilos, que se dedica a dar un mejor aspecto a nuestros proyectos.

## 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**](http://es.wikipedia.org/wiki/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](http://es.wikipedia.org/wiki/SGML), pero no fue formalmente reconocida como tal hasta la publicación a mediados de 1993, por la [IETF](http://es.wikipedia.org/wiki/IETF), de una primera proposición para una especificación de HTML: el boceto *Hypertext Markup Language* de Berners-Lee y [Dan Connolly](http://es.wikipedia.org/w/index.php?title=Dan_Connolly&action=edit&redlink=1), el cual incluía una [Definición de Tipo de Documento](http://es.wikipedia.org/wiki/DTD) 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](http://es.wikipedia.org/wiki/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.

**¿Qué hay de nuevo entonces en html5?**

Hay muchas mejoras y cosas nuevas que nos pueden sorprender por ejemplo el tema de formularios, la forma de estructuración en fin…

A continuación alguna de las reglas establecidas para**HTML5**:

* Las nuevas características debe basarse en HTML, CSS, DOM y JavaScript.
* Reducir la necesidad de plugins externos (como Flash).
* Mejor manejo de errores.
* Más marcado para reemplazar secuencias de comandos.
* HTML5 debe ser independiente del dispositivo.
* El proceso de desarrollo debe ser visible para el público.

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>

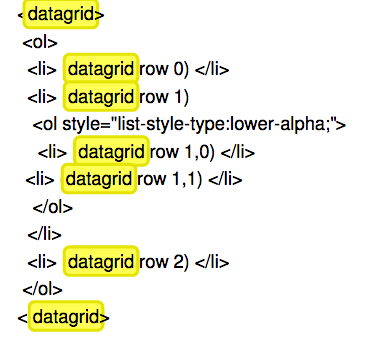
* La etiqueta canvas para dibujar contenido en el navegador (2D y 3D) (Recomendado: Cómo dibujar curvas cuadráticas en un Canvas de HTML5)

<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



* 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.

# Diseño y desarrollo web adaptado a móviles

En este capítulo, veremos los aspectos que debemos tener en cuenta acerca del panorama actual para diseñar y desarrollar soluciones web para móviles. Nos introduciremos en las técnicas de detección de características y pantallas de dispositivos móviles. Además, conoceremos las ventajas que introduce HTML5 en este rubro, y veremos las herramientas y los frameworks que nos ayudarán en el desarrollo y testeo.

**La evolución de la web Mobile**

La navegación web a través de móviles se ha convertido en una tendencia que se mantiene en ascenso constante. Cada vez son más y de mayor potencia los dispositivos que se comercializan, lo que ha generado un gran crecimiento del mercado.

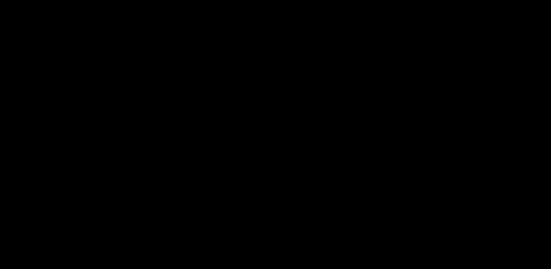
Las tecnologías relacionadas con este fenómeno también han evolucionado en el último tiempo, y las técnicas de desarrollo acompañan estos cambios. Desde las primeras especificaciones del lenguaje WML (*Wireless Markup Language*) y del estándar WAP (*Wireless Application Protoco*l), en la primera época de la Web móvil, hasta el soporte actual de XHTML, HTML5, CSS3 y JavaScript por parte de muchos navegadores mobile, un largo camino se ha transitado y una importante evolución se ha producido tanto en la Web como en los dispositivos.

En la actualidad, gran parte de los aparatos que cuentan con tecnologías para navegar por Internet pueden hacerlo sin inconvenientes por la mayoría de los sitios que se encuentran disponibles hoy en día.

Respecto de la compatibilidad con móviles, debemos tener en cuenta que algunas de las tecnologías que se utilizan en el desarrollo web no son compatibles en todas las plataformas móviles.

Un ejemplo de esto pueden ser Flash y Silverlight, que son soportados solo por algunos de los navegadores mobile disponibles en el mercado.

En lo que se refiere a la interfaz de usuario, existe un cambio de paradigma muy importante. Las tecnologías touch y multi-touch marcan un cambio también para el diseño de interfaces. El mouse deja su papel protagónico a los dedos, y esto nos obliga a repensar la manera de crear los elementos accesibles que componen la interfaz de la Web y también el uso de los eventos, ya que para este tipo de dispositivos no tiene sentido el uso de **onmouseover** u **onmouseout**, por ejemplo.



iPhone ([www.apple.com/es/iphone)](http://www.apple.com/es/iphone) es el smartphone desarrollado por Apple que se destaca por su panta- lla de alta calidad y soporte multi-touch, y también por la compati- bilidad con HTML5 en su navegador.

También es importante tener en cuenta que los usuarios que acceden desde un móvil o una tablet a un sitio o a una aplicación web esperan contar con una versión optimizada para su plataforma y, de ser posible, acceder a una experiencia pensada para aprovechar al máximo los recursos que pueden ofrecer estos tipos de dispositivos.

### Ventajas de HTML5 para las plataformas móviles

En el ámbito del desarrollo mobile, uno de los debates que se plantean es el desarrollo de aplicaciones nativas o aplicaciones web.

Las aplicaciones nativas tienen como ventaja la personalización que puede realizarse para el medio y el acceso al hardware. Sin embargo, por parte de las soluciones web, estos aspectos pueden superarse en la actualidad con la evolución de las técnicas de desarrollo y las nuevas posibilidades que ofrecen a móviles, incluida la opción de que el navegador tenga acceso al hardware en ciertos modelos.

Algunas de las características de HTML5 que pueden resultar especialmente útiles para el desarrollo de móviles tienen que ver con las posibilidades multimedia para audio y video, acceso a la API de geolocalización y características de formulario. Otro de los aspectos que abren un interesante camino para el desarrollo en móviles es la posibilidad de crear aplicaciones web que puedan funcionar offline. También algunas de las características que incorpora CSS3 comienzan a ser soportadas por ciertos navegadores móviles.

### 

### Navegadores para móviles

Existen diversos sistemas operativos para móviles. Algunas compañías, como el caso de Apple, han desarrollado sus propios sistemas para dispositivos de este tipo. Otras empresas han optado por soluciones que puedan adaptarse a diferentes clases de dispositivos, como el claro ejemplo de Android. Si bien la mayoría de los sistemas móviles que se encuentran preparados para conectarse a Internet cuentan con un navegador y software adecuado para hacerlo, también existen compañías que desarrollan versiones de browsers multiplataforma, como por ejemplo, Opera.

A continuación, repasaremos los principales navegadores para plataformas móviles y sus características.

##### **Mobile Safari**

Este navegador es el que utilizan los dispositivos móviles desarrollados por Apple. Basado en WebKit, cuenta con muy buenas opciones de zoom, interfaz multi-touch para acceder mediante gestos de los dedos, y muy buena compatibilidad con características de HTML5 y CSS3. Su motor JavaScript es Nitro, y se destaca su desempeño en el Acid3 test.

Distribuido junto al sistema iOS, ofrece soporte para características como geolocalización, almacenamiento del lado cliente, FormData (XMLHttpRequest Level 2), video (HTML5), formularios avanzados

(HTML5), WebSockets (HTML5), Canvas (HTML5) y SVG (HTML5). Vale recordar que este navegador no soporta Flash.

Una característica interesante que se le ha incorporado a este producto (desde iOS 4.2) es la posibilidad de acceder al acelerómetro y giroscopio del móvil desde el navegador, mediante código programado en JavaScript, gracias al acceso a la Device Orientation API y a los eventos de movimiento (*Device Motion Event*). Podemos encontrar más información para el desarrollo de soluciones web enfocadas en este navegador en Safari Developer Library ([http://](http://developer.apple.com/library/safari/navigation) [developer.apple.com/library/safari/navigation).](http://developer.apple.com/library/safari/navigation)

##### **Navegador Android**

Debemos tener en cuenta que el navegador que se incluye junto a los sistemas operativos Android (conocido como Android Web Browser) dispone de un motor basado en WebKit, y también debemos saber que V8 es su motor de JavaScript. Cabe destacar que este software, al igual que Chrome, son desarrollos de Google.

Este navegador móvil soporta HTML5 y logra buen desempeño en los tests, pero vale aclarar que no soporta todas las características de Chrome.

Este navegador posee soporte para acceso a la API de geolocalización, acceso a hardware (acelerómetro), animaciones con Canvas de HTML5, Web Storge, elementos de formulario HTML5 y estilos con CSS3, entre otras muchas funcionalidades accesibles desde este browser. Es importante aclarar que este navegador puede soportar el reproductor de Flash (dependiendo de la versión del sistema y el hardware).

##### **Internet Explorer Mobile**

Este navegador, en sus primeros tiempos en la década del noventa, fue conocido como Pocket Internet Explorer y se distribuyó con los sistemas para móviles Windows CE. Originalmente, no utilizaba el mismo motor que la versión de escritorio.

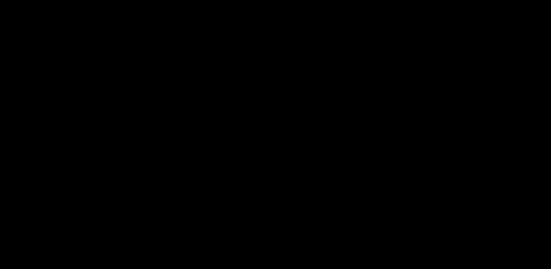
En la actualidad, funciona como navegador predeterminado en Windows Phone 7, utiliza motor Trident, soporta interfaces multitáctiles y tiene diferentes niveles de zoom.

A partir de Internet Explorer Mobile 9, se incorpora soporte a HTML5 y aceleración por hardware, funciones que se pueden apreciar en la versión de escritorio de IE9. Este navegador se incluye junto a

Windows Phone 7, que podemos encontrar en el sitio web que está en la dirección [www.microsoft.com/windowsphone/es-es/default.aspx.](http://www.microsoft.com/windowsphone/es-es/default.aspx)

##### **Firefox para móviles**

Mozilla ha introducido una versión de Firefox pensada especialmente para móviles, cuyo nombre es Firefox for mobile. Este producto, que es una evolución del que anteriormente se denominaba Fennec, hoy por hoy funciona en sistemas Maemo, Windows Mobile 6.0 Professional (o superior) y Android (en este último caso, dependiendo del dispositivo).

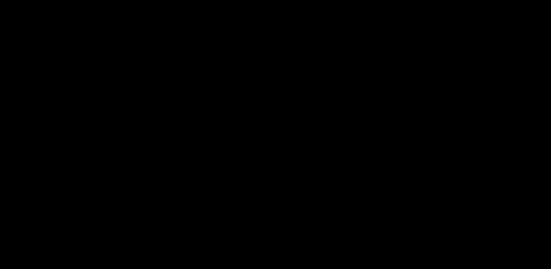


Este navegador para móviles cuenta con el mismo motor de render que Firefox (Gecko), y su motor de JavaScript es JaegerMonkey. Ofrece compatibilidad con HTML5 (Web workers, Offline storage, Canvas y SVG, entre otras características), CSS3 y JavaScript.

##### **Opera Mini y Opera Mobile**

Es importante recordar que Opera es una compañía que, entre sus productos, se destaca en especial por ofrecernos, de manera completamente gratuita, versiones de su navegador para diferentes plataformas, entre las que se encuentran las siguientes:

Opera Mini es una alternativa muy interesante que se basa en un motor de render online (ubicado en los servidores de Opera) para ofrecer un flujo de datos optimizado. Es posible probar una demostración de un simulador de Opera Mini, el cual encontramos en el sitio web que está en la dirección [www.opera.com/mobile/demo.](http://www.opera.com/mobile/demo) Según las características de nuestro dispositivo, es posible optar por Opera Mobile, un completo y potente navegador para móviles, que utiliza el mismo motor que la versión de escritorio (Presto), y que, incluso, brinda soporte para HTML5 y acceso al acelerómetro (solo obtendremos acceso a esta característica si está disponible en el teléfono correspondiente).



##### 

##### **Otros navegadores para móviles**

Además de los navegadores mencionados anteriormente, podemos encontrar otros, entre los que se destacan:

**BlackBerry Browser**: navegador de los dispositivos BlackBerry.

Utilizó el motor de render Mango entre la versión 4.5 y la 5, pasando a WebKit desde la versión 6. En el sitio oficial, podemos encontrar el SDK, y simuladores que permiten realizar desarrollos y pruebas destinadas a esta plataforma móvil.

**NetFront**: desarrollado por la empresa japonesa Access Co.Ltd., funciona en una gran variedad de plataformas. Su motor es NetFront. Soporta HTML 4.01, algunas características de HTML5 (según la versión) y también AJAX. Más información en [www.access-company.com/products/mobile\_solutions/](http://www.access-company.com/products/mobile_solutions/) netfrontmobile/browser/index.html.

**Blazer**: creado por Palm para sus sistemas, está basado en NetFront. Soporta HTML 4.01/XHTML 1.0, WML 1.3 y JavaScript, entre otras características. Podemos ver más sobre sus particularidades en el sitio [www.palm.com/ar/](http://www.palm.com/ar/) mobilemanagers/lifedrive/blazer.html.

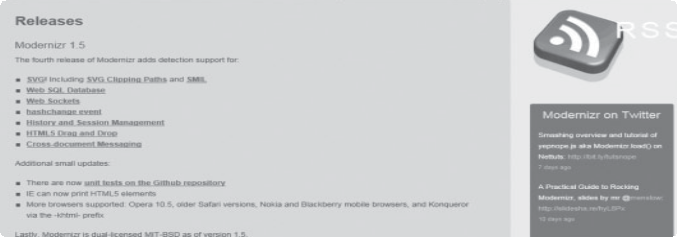
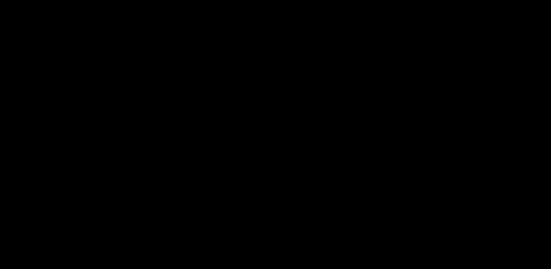
Vale destacar que Symbian, uno de los sistemas móviles más usados, se encuentra entre los precursores del uso de WebKit en su navegador para móviles. El navegador de Symbian Anna es compatible con HTML5 y CSS3.

### Cómo saber si el móvil está preparado para HTML5

Debemos saber que, al igual que en un navegador de escritorio, en el caso de los móviles podremos encontrar un variado nivel de compatibilidad con HTML5 y CSS3.

De la misma manera que ocurre con las opciones desktop, es posible trabajar con JavaScript para detectar compatibilidad.

Modernizer (que podemos encontrar en la dirección web [www.modernizr.com)](http://www.modernizr.com/) es una librería sobre la que ya hemos hablado. También puede resultarnos útil en la detección de las características con las que cuenta el navegador del móvil.



### Técnicas de detección

Existen diversas formas para detectar si el usuario que está accediendo a nuestro sitio web lo hace desde una plataforma móvil o si, por el contrario, se maneja desde una de escritorio.

En cualquiera de los casos, también dependeremos de ciertas características del navegador del móvil para poder tener éxito con este proceso.

Si bien existen diferentes soluciones mediante JavaScript o,

Incluso, empleando lenguajes del lado servidor, en esta oportunidad veremos las opciones que nos ofrece CSS, por medio de la regla **@ media** y Media Queries, característica incorporada a partir de CSS3.

**Crear aplicaciones web**

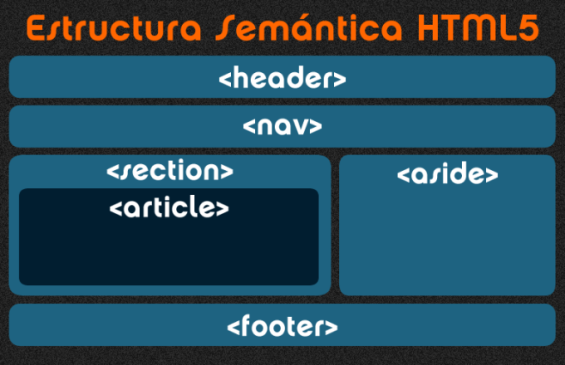
Con el éxito de la Web móvil y el auge de HTML5, han surgido librerías que aprovechan estas características, sumadas a otras técnicas, para ofrecer soluciones eficaces que nos ayuden a resolver nuestros desarrollos con mayor flexibilidad y sin tantas complicaciones.

La creación de aplicaciones web para móviles empleando HTML5 puede ser potenciada gracias al uso de diferentes frameworks, como veremos a continuación.

# 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.

## 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 PAramio

**¿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.

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

#### El nuevo <! DOCTYPE>

Esto es lo primero que encontramos en un documento web. Te reto a que me nombre la cabeza el tipo de documento de XHTML 1.0 Strict o HTML 4.01. No es fácil ¿eh?.

El **<! DOCTYPE>** para la versión HTML 4.01:

|  |  |
| --- | --- |
| 1  2  3 | <!DOCTYPE HTML PUBLIC »  "-//W3C//DTD HTML 4.01//ES" »  "http://www.w3.org/TR/html4/strict.dtd"> |

Aquí el **<! DOCTYPE>** para la versión HTML 1.0:

|  |  |
| --- | --- |
| 1  2  3 | <!DOCTYPE html PUBLIC »  "-//W3C//DTD XHTML 1.0 Strict //ES" »  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> |

si lo recuerdas te felicito porque deberás de ser un duro del diseño y el desarrollo web, de suerte este código del tipo de documento se ha simplificado. En realidad todo lo que dice en los anteriores códigos es “este documento está escrito en HTML 4.01,” o “Este documento está escrito en XHTML 1.0.”, eso es todo.

Ahora en HTML5 El encabezado o la primera línea se ha simplificado considerablemente. Ahora podemos mantener este Doctype en la memoria, sin tener que copiar y pegar. Es solo esto:

|  |  |
| --- | --- |
| 1 | <!DOCTYPE html> |

Es tan corto que todo el mundo lo puede memorizar.  
El <! DOCTYPE> debe ser lo primero que debe ir en un documento HTML 5, antes de la etiqueta <html>.

¡¡Ojo!! <DOCTYPE!>, no es una etiqueta, esto es una instrucción para el navegador que declara la versión de html que se está usando.

En HTML 4.01, todo la declaración del <! DOCTYPE> requieren una referencia a una DTD, HTML 4.01, porque se basa en [SGML](http://es.wikipedia.org/wiki/SGML).

HTML5 no se basa en [SGML](http://es.wikipedia.org/wiki/SGML), y por lo tanto no requiere una referencia a una DTD.

**Importante:** Siempre agregue la declaración <DOCTYPE! html> a los documentos HTML, para que el navegador sepa qué tipo de documento debe de esperar.  
El DOCTYPE se activa en HTML5 en todos los navegadores que tienen un modo estándar, incluyendo los navegadores que no saben nada acerca de HTML5. Por esa razón, usted puede comenzar a usar el DOCTYPE de HTML5 ahora.

Recuerda:

|  |  |
| --- | --- |
| 1 | <!DOCTYPE html> |

Además de los elementos estructurales vimos en el tutorial anterior, HTML 5 introduce una serie de nuevos elementos semánticos que son elementos que sin ninguna duda mejoran la estructura de cualquier sitio, las etiquetas semánticas, son claves para posicionamiento en buscadores y el buen desarrollo web Vamos a Examinarlas y a ver algunos detalles de ellas y también HTML5 incluye nuevos elementos para el manejo de las forma, el dibujo y para el contenido de los medios de comunicación. Y también vamos a ver algunas etiquetas que fueron eliminadas en esta nueva versión de HTML.

en la siguiente tabla te mostrare todo lo nuevo de **HTML5:**

### Nuevos elementos semánticos / Estructural

|  |  |
| --- | --- |
| **Etiqueta** | **Descripción** |
| **<article>** | Define un artículo |
| **<aside>** | Define el contenido aparte del contenido de la página |
| **<bdi>** | Aísla una parte del texto que pueda ser formateada en una dirección distinta de otro texto fuera de ella |
| **<command>** | Define un botón de comando que un usuario puede invocar |
| **<details>** | Define los detalles adicionales que el usuario pueda ver u ocultar |
| **<summary>** | Define un título visible para un elemento <details> |
| **<figure>** | Especifica autónomo de contenido, como ilustraciones, diagramas, fotos, listas de códigos, etc |
| **<figcaption>** | Define un título para un elemento <figure> |
| **<footer>** | Define un pie de página de un documento o sección |
| **<header>** | Define un encabezado de un documento o sección |
| **<hgroup>** | Agrupa un conjunto de etiquetas desde los <h1> hasta los <h6> cuando un título tiene varios niveles |
| **<mark>** | Define el texto marcado / resaltado |
| **<meter>** | Define una medida escalar dentro de una gama. |
| **<nav>** | Define vínculos de exploración |
| **<progress>** | Representa el progreso de una tarea |
| **<ruby>** | Define una anotación de rubí (para el Este de la tipografía de Asia) |
| **<rt>** | Define una explicación / pronunciación de los caracteres (para el Este de la tipografía de Asia) |
| **<rp>** | Define lo que se muestra en los navegadores que no son compatibles con anotaciones de rubí |
| **<section>** | Define una sección de un documento |
| **<time>** | Define una fecha / hora |
| **<wbr>** | Define una posible ruptura del verso |

### Nuevos para medios de comunicación / Media

|  |  |
| --- | --- |
| **Etiqueta** | **Descripción** |
| **<audio>** | Define el contenido de sonido |
| **<video>** | Define un vídeo o una película |
| **<source>** | Define los recursos de medios múltiples para <video> y <audio> |
| **<embed>** | Define un contenedor para una aplicación externa o de contenido interactivo (un plug-in) |
| **<track>** | Define las pistas de texto para <video> y <audio> |

### El nuevo elemento <canvas>

|  |  |
| --- | --- |
| **Etiqueta** | **Descripción** |
| **<canvas>** | Se utiliza para dibujar gráficos sobre la marcha a través de secuencias de comandos (normalmente JavaScript) |

### Nuevos elementos de formulario

|  |  |
| --- | --- |
| **Etiqueta** | **Descripción** |
| **<datalist>** | Especifica una lista de opciones predefinidas para los controles de entrada |
| **<keygen>** | Define un campo generador de par de claves (para las formas) |
| **<output>** | Define el resultado de un cálculo |

### Elementos eliminados de la versión anterior de HTML

Los siguientes elementos o etiquetas DE LA VERSION HTML4 fueron eliminados y su uso no será correcto para la nueva versión de HTML5

* <acronym>
* <applet>
* <basefont>
* <big>
* <center>
* <dir>
* <font>
* <frame>
* <frameset>
* <NOFRAMES>
* <strike>
* <tt>

Que más te puedo decir acerca de estos nuevos elementos, bueno en realidad muchísimo, pero mejor te remito a unas fuentes de mayor información:  
[Ver los elementos de html5 en forma de una tabla periódica](http://joshduck.com/periodic-table.html)



# Unidad 3 Estructura de HTML 5

## Estructura

**Estructura básica de una página en HTML5**

Aunque gran parte de la atención que se tiene sobre **HTML5** gira en torno a las nuevas API, y sus 30 nuevos elementos con la nueva semántica que se puede utilizar en tradicionales páginas estáticas, también en las franja de controles de formularios. Vamos a tocar un poco el tema de lo básico como veníamos tratando hasta ahora, Vamos a ver un poco la estructura básica y algunas mejoras establecidas en la misma.

Así que vamos a ver una estructura html5, que es lo mínimo que se requiere para tener una correcta base para iniciar nuestro proyecto, luego vamos a tratar de analizarla un poco.

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>

Lo primero del **DOCTYPE** es la etiqueta que no pasa de moda la clásica , si te fijas veras que tiene un atributo que es lang, ósea lenguaje aquí definimos en que lenguaje va a ir nuestro documento web, lo puedes en “en” si tu web va a hacer en el idioma Ingles o “es” si está en Español, ya dependerá de tu proyecto, en este caso hace referencia a una web construida en español, y esto ayudara también a muchas personas con algunas discapacidades, que utilizan un lector de pantalla o algo parecido, que más que decirles en qué idioma esta nuestra web. Más información: <http://www.w3schools.com/tags/ref_language_codes.asp>.

Echa un vistazo a la etiquetación mucho cuidado. Los que están acostumbrado a escribir XHTML se darán cuenta que es el primer cambio. Ya que estábamos acostumbrados a esto:

|  |  |
| --- | --- |
| 1 | <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> |
|  |  |

Esto es todavía lo puedes usar, ya que en html4 y 5 podemos intercambiar etiquetas en la manera posible que lo necesitemos, pero en html5 el camino más corto es mejor ya que es más fácil de escribir y que ya funciona en todas partes, solo tenemos que escribir esto:

|  |  |
| --- | --- |
| 1 | <meta charset="utf-8"/> |

Ahora sé que en él la etiquetase encuentra muchas metas, pero la verdad es que realmente no vale la pena meterlas en nuestros documentos ya que no tienen ya validez para los buscadores que fueron para las que se crearon principalmente. Ejemplo, la meta keywords, otras como la meta description, si es muy importante ya que ayudara a los buscadores a expresar el contenido de tu pagina.

Si quieres mas información acerca esta etiqueta meta ve a esta dirección: <http://www.w3schools.com/html5/tag_meta.asp>

Y la etiqueta **<title>** que es la que provee el titulo a la pagina, esta es muy importante incluirla a los documentos creados, ya que la asociamos a una identidad por medio de su titulo.

|  |  |
| --- | --- |
| 1 | <title>Aqui va el titulo del docuemento actual</title> |

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).

Bueno vamos a definir una estructura más completa de un documento web utilizando las nuevas etiquetas estructurales. **<header>, <hgroup>**, **<footer>**, **<nav>**, **<aside>**, **<section>** y **<article>**.

Vamos a definir el documento y luego vamos a examinar las etiquetas más a fondo y conocer como las podemos utilizar en nuestros propios proyectos.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8"/>      <meta name="description" content="curso html5 desde 0" />      <title>Mi primera web en html5</title>  </head>  <body>    <header>       <hgroup>         <h1>Mi curso de html5</h1>         <h2>Aprende desde 0</h2>       </hgroup>    <nav>       <ul>          <li><a href="#">Home</a></li>          <li><a href="#">quienes somos</a></li>          <li><a href="#">portafolio</a></li>          <li><a href="#">contacto</a></li>       </ul>    </nav>  </header>    <article>      <h2>Aquí va un post, con su titulo en h2</h2>         <p>Aqui puede ir los contenidos del post</p>    </article>    <article>      <h2>Aquí va un post, con su titulo en h2</h2>        <p>Aqui puede ir los contenidos del post</p>    </article>  <aside>     <h2>Titulo del sidebar</h2>     <p>alguna informacion que se muestra en la barra lateral </p>     <ul>       <li><a href="#">enlaces</a></li>       <li><a href="#">enlaces</a></li>     </ul>  </aside>  <footer>       <p>Aqui va el Pie de pagina, copyright, el menu otravez, etc.</p> </footer>  </body>  </html> |

Hemos utilizado estos elementos **<header>, <hgroup>**, **<footer>**, **<nav>**, **<aside>**, **<section>** y **<article>**. Para marcar y así formar la estructura casi que completa de nuestra página, ahora es evidente notar los nuevos nombres, es el momento para estudiar en detalle un poco más de esas nuevas etiquetas que llamamos estructurales.

1º – **<header>**: Esta etiqueta hace el mismo trabajo que esta <div id=”header”>, esta etiqueta <header> la utilizamos para contener información adicional como logos y ayudas a la navegación, iconos de redes sociales etc. Esta etiqueta tiene su cierre de esta manera </header>.

2º – **<hgroup>**: Muchos headers de las páginas web podrán contener múltiples títulos representados con la etiqueta h1, y de pronto un subtítulo formando con la etiqueta h2. Bueno esta nueva etiqueta <hgroup> permite colocar un h1, h2 y h3 dentro del header sin afectar el SEO, permitiendo usar otro h1 en el sitio. Esta etiqueta tiene su cierre de la siguiente forma, </hgroup>.

|  |  |
| --- | --- |
| 1  2  3  4 | <hgroup>      <h1>Mi curso de html5</h1>      <h2>Aprende desde 0</h2>  </hgroup> |

3º – **<nav>:** La etiqueta <nav> está diseñada para colocar la botonera o navegación principal los normales (Home, quienes somos, portafolio, contáctenos, blog). Puedes colocar cualquier etiqueta dentro, aunque lo recomendado es usar listas <ul> con sus respectivos <li>, su cierre es </nav>.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <nav>       <ul>          <li><a href="#">Home</a></li>          <li><a href="#">quienes somos</a></li>          <li><a href="#">portafolio</a></li>          <li><a href="#">contacto</a></li>       </ul>  </nav> |

4º – **<aside>**: Esta etiqueta es creada para contener información no relevante para el sitio web como da a sospechar su nombre que se traduciría algo así como “a un lado”. En un blog, obviamente el <aside> es la barra lateral de información donde se muestra las categorías, blogroll etc. Esta etiqueta puede contener cualquier cosa desde un reproductor multimedia hasta una galería de imágenes, el cierre de esta etiqueta es </aside>.

5º – **<section>**: Esta etiqueta Define un área de contenido única dentro del sitio. Esta es una de las etiquetas más genéricas de los elementos estructurales ya que podemos agrupar contenidos relacionados por el tema. Ejemplo: En un blog, esta etiqueta sería la zona donde están todos los posts, el cierre de esta etiqueta es </section>.

6º – **<article>**: Esta etiqueta especifica un contenido independiente y autónomo. Define zonas únicas de contenido independiente. Ejemplo: En el home de un blog, cada post sería un <article> y el post y cada uno de sus comentarios serían varios <article>, el cierre respectivo de esta etiqueta es </article>.

7º – **<footer>**: Esta etiqueta lo que define es un pie de página con la información del copyright, autor un menú o lo queramos colocar en el pie de la web, el cierre de esta etiqueta es </footer>.



**NOTA:** Algo muy importante: el caso de ver nuevas etiquetas estructurales no significa que desaparecieron los divs, los podemos seguir utilizando en cuestiones de contenido interno, porque para el tema de la estructura y semántica las nuevas etiquetas nos pueden venir my bien, y es bueno que comencemos a basarnos en ellas.

Bueno en conclusión, utilizando estas nuevas etiquetas podemos estructurar una web basada en HTML5, se vería la estructura algo así como lo vamos a ver en el siguiente cuadro, y claro para una mejor apariencia que más que css3, pero vamos por partes.

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 Stadistics: 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.

Hoy día ya en el futuro hablado desde hace años es increíble que hablemos de compatibilidad, pero es la realidad, hoy día vemos una gran avanzada en las grandes empresas que controlan la web para mejorar este asunto pero desafortunadamente existen todavía residuos del pasado, viejos navegadores que podrían dar algún dolor de cabeza de vez en cuando, cuando ejecutamos HTML5 Y CSS3 Vamos a tener que dar soporte a algunas versiones de internet explorer de pronto no la 6 aunque no faltara quien lo use pero si tratar de lidiar con la versión 7, 8 y 9 que no soportan todas las nuevas etiquetas y elementos.

**Ranking por Versiones, Navegadores de Escritorio Mayo 2012:**

1. Internet Explorer 8: 26.92%
2. Internet Explorer 9: 17%
3. Firefox 12: 11.15%
4. Chrome 18: 9.10%
5. Chrome 19: 7.41%
6. Internet Explorer 6: 17%
7. Safari 5.1: 3.48%
8. Internet Explorer 7: 3.43%
9. Firefox 11: 2.24%
10. Firefox 3.6: 1.81%
11. Opera 11: 1.40%

fuente: [www.netmarketshare.com](http://www.netmarketshare.com/)

Si puedes notar el navegador más usado es Internet Explorer y en su versión 8 y el segundo lugar es para el explorer 9, pues te digo que la versión 8 necesita algunos trucos para su compatibilidad con HTML5. Las versiones más recientes de los navegadores más populares del mercado soportan HTML5. Entre ellos: Internet Explorer 9 ( el muy mejorado IE10), Firefox 9, Chrome 16, Safari 5.1 y Opera 11.60.

En el caso de IE, las versiones 6, 7 y 8 no tienen soporte para HTML5. El primero en dar compatibilidad en esta familia ha sido Internet Explorer 9, lanzado oficialmente en el año 2011.

Así que te daré algunas herramientas que te ayudaran a trabajar correctamente con HTML5 aun en los navegadores antiguos.

1 – [HTML5 Test](http://html5test.com/). nos indica el soporte de HTML5, ítem por ítem, del navegador que estamos usando para pasar el test.



2 – [html5shiv](http://code.google.com/p/html5shiv/) Esta herramienta nos ayuda muchísimo ya que hará que los navegadores viejos obsoletos puedan entender los nuevos elementos de HTML5, basta incluir un pequeño código de enlace en la etiqueta <head> y podremos trabajar con la nueva versión de HTML libremente sin preocuparnos de los navegadores de antaño. Seria algo mas o menos asi:

|  |  |
| --- | --- |
| 1 | &lt;script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"&gt;&lt;/script&gt; |

3 – [Modernizar](http://www.modernizr.com/) esta librería le permite ofrecer “respaldo” de estilo en los navegadores que no soportan HTML5 ciertas características de CSS3. Esta librería puedes detectar si el navegador tiene soporte para múltiples capacidades HTML5, [Modernizar](http://www.modernizr.com/). De una forma sencilla, nos permite incluirla en nuestras páginas y comprobar la compatibilidad con el navegador.



también puedes ver el funcionamiento de esta librería y como instalarla y cómo usarla en este video:

4 – y por ultimo [HTML5 Cross Browser Polyfills](https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-browser-Polyfills). Estas librerías están basadas generalmente en HTML, CSS y especialmente en JavaScript. Dentro de las soluciones que encontraremos se destacan alternativas para audio, video, formularios, geolocalización, SVG, Canvas, Web Storage, Web Sockets, Web Workers, entre otras.

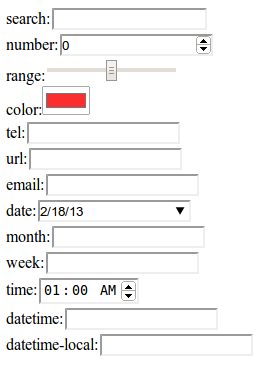
————–

Bueno y tenemos más herramientas disponibles en la web, solo te quería dar esta información y hacerte saber que puedes hacer compatible HTML5 con todos los navegadores, muy pronto ya internet explorer 7 y 8 dejaran de ser usados por las personas, mientras nos toca hacer trucos para sacar proyectos adelantes.

En conclusión. Podemos entrar en este mundo de HTML5 confiadamente, estamos en era de cambio y no sería posible que nos quedemos en el pasado trabajando con antiguas tecnologías.

# Unidad 4 Funcionalidades

## Web Forms

Nuevos atributos para **<input>**

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

Uno de los problemas que teníamos con la versión anterior de HTML es decir la 4, era la validación ya que teníamos que utilizar otra tecnología para su validación, por supuesto con Javascript dábamos el toque que necesitábamos para dar lo que le faltaba a estos formularios.

En HTML5 esto se ha solucionado ya que se hicieron unas grandes mejoras para su propia validación y otras para su fácil manejo en si para verse más atractivos.

En este capítulo, vamos a ver una pequeña muestra a las nuevas características de los formularios en

HTML5. Y entenderemos que sus nuevas características son muy importantes para la web de hoy.

Ahora, ten en cuenta que no todos los navegadores soportan estas novedades y quienes lo hacen muestran los resultados a su manera, pero Empecemos:

Todos sabemos que La etiqueta **<input>** especifica un campo de entrada donde el usuario puede introducir datos. Bueno, en HTML5 vemos que tiene varios tipos de entrada o **<input>**. Estas nuevas características permiten un mejor control de entrada y validación. Ejemplos, tenemos:

**<input type=text>**: especifica que el usuario debe de llenar el campo con texto  
**<input type=email>**: especifica que el usuario debe de llenar el campo con un email valido  
**<input type=url>**: especifica que el usuario debe de llenar el campo con url  
**<input type=color>**: especifica que el usuario debe de llenar el campo con un color  
**<input type=date>**: especifica que el usuario debe de llenar el campo con una fecha  
**<input type=month>**: especifica que el usuario debe de llenar el campo con unos de los meses

para ver mas opciones ve aquí: <http://www.w3schools.com/html5/html5_form_input_types.asp>

### 1 – Elemento <datalist>:

Este elemento proporcionar un “autocompletar” en funciones de los <input>. Es decir cuando el usuario se situé dentro del campo y comience a escribir este campo va a desplegar una cantidad de opciones predefinidas anteriormente en su configuración. Ejemplo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <input list="Paises-europeos" />    <datalist id="Paises-europeos">    <option value="España">    <option value="Francia">    <option value="Alemania">    <option value="Portugal">    <option value="Holanda">  </datalist> |

Si comienzas a escribir con la letra e, el por supuesto te autocompletara españa. Así funciona.

### Los nuevos atributos:

Además de los nuevos tipos de entrada, este tiene varios atributos nuevos para especificar el comportamiento y las limitaciones: Autocomplete, min, max, pattern, múltiple, y step.

También para las listas hay un nuevo atributo, es el elemento <datalist>, que especifica una lista de opciones predefinidas para un elemento <input>

##### - Hablemos de los nuevos atributos para <input> y para <form>

**1 – El atributo autocomplete:**

Que vendría siendo “Autocompletar” La mayoría de los navegadores tienen algún tipo de funcionalidad de autocompletar. El atributo autocomplete permite controlar cómo funciona esto. El atributo autocomplete específica si un campo de formulario o de entrada debe tener autocompletado de encendido o apagado, es decir On o Off.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <form action="form.php" autocomplete="on">    Nombre:<input type="text" name="fname" /><br />    Apellido: <input type="text" name="lname" /><br />    E-mail: <input type="email" name="email" autocomplete="off" /><br />    <input type="submit" />  </form> |

El atributo autocomplete trabaja con <form> y con los <input> siguientes: ext, search, url, tel, email, password, datepickers, range, and color.

**2- El atributo autofocus:**

El autofocus o enfoque automático proporciona una forma declarativa para enfocar un control de formulario durante la carga de la página. Anteriormente, un desarrollador necesita para escribir JavaScript utilizando control.focus (). La nueva forma permite al navegador hacer cosas inteligentes como no centrar el control si el usuario ya está escribiendo en otro.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <form action="form.php">    Nombre:<input type="text" name="fname" autofocus="autofocus"/><br />    Apellido: <input type="text" name="lname" autofocus="autofocus"/><br />    E-mail: <input type="email" name="email" autofocus="autofocus" /><br />    <input type="submit" />  </form> |

Esto funciona correctamente en chrome.

**3- El atributo múltiple:**

El atributo múltiple especifica que el usuario puede introducir más de un valor en el elemento <input>. Ej: <input type=email múltiples> permite al usuario introducir varias direcciones de correo electrónico.

|  |  |
| --- | --- |
| 1  2 | Seleccione varias imagenes: <input type="file" name="img" multiple="multiple" /><br/>  Entre varios Email:<input type="email" multiple="multiple"> |

Anteriormente solo se soportaba en navegadores que implementan WebKit como Chrome.

**4- El atributo required:**

El atributo required o requerido en un elemento se establecerá en un <input>, y automáticamente hace que el usuario se vea obligado a llenar el campo para continuar, es decir el navegador no permitirá que se envié la forma sin que el input con este atributo este vacío.

|  |  |
| --- | --- |
| 1  2  3  4 | <form action="demo\_form.asp">    Nombre: <input type="text" name="name" required="required" />    <input type="submit" />  </form> |

El atributo requerido trabaja con los tipos de entrada: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

**5- El atributo min and max:**

El atributo min and max o mínimo y máximo nos hace especificar el valor mínimo y máximo para un elemento <input>.

|  |  |
| --- | --- |
| 1 | Escoja de 1 a 5: <input type="number" name="quantity" min="1" max="5" /> |

El atributo min a max funciona con los siguientes tipos de entrada: number, range, date, datetime, datetime-local, month, time and week.

—————————————–

Son aproximadamente 18 atributos que podemos utilizar, pero Ojo la mayoría está con solo soporte para navegadores que implementan webkit como chrome y opera.

### 2 – Elemento <output>:

El elemento <output> es para mostrar los resultados de un cálculo o de otro tipo con la escritura.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0     <input type="range" name="a" value="50" />100    +<input type="number" name="b" value="50" />    =<output name="x" for="a b"></output>  </form> |

Por ejemplo aquí se realiza un cálculo del atributo step que nos genera un resultado y el del atributo range que nos presenta un slider que podemos graduar para dar un resultado. Y el resultado total lo sacamos utilizando el elemento <output>

### 3 – Elemento <keygen>:

El propósito del elemento <keygen> es proporcionar una forma segura de autenticar a los usuarios. La etiqueta <keygen> especifica un campo generador de un par de claves en un formulario, cuando se envía el formulario, dos claves se generan, uno público y uno privado.

**En Conclusion:**

El apoyo crece en cuanto a los elementos de entrada y los atributos de los formularios en HTML5, para crear formularios más completos se requerirá menos y menos de JavaScript para la validación del lado del cliente y las mejoras de la interfaz de usuario. Aunque muchos de estos atributos no sean soportados por los navegadores excluyendo chrome no hay duda que debemos aprender a usarlos por que saldrán técnicas que harán posible su compatibilidad o a medida que pasa el tiempo y las actualizaciones de los navegadores se ejecuten contaremos ya con poderosas herramientas para nuestro trabajo.

## Audio y video

En este artículo, vamos a ver de pronto no todo acerca los videos en HTML5 pero si lo más relevante e importante de esta nueva etiqueta Video.

HTML ha hecho cambios muy grandes en la web, que quizás la gente común no note pero a los que trabajamos en este campo se no se nos hacen grandioso, una de estas mejores es la de poder incluir directamente video y audio en el código, para depender de el mismo HTML y no de flash como lo veníamos haciendo (bueno ahora tenemos otra alternativa para elegir), en este tutorial veremos el tema del Video en HTML5.

Una buena razón para utilizar videos en nuestro HTML5 es por la decisión de Apple de no soportar Flash en el iPhone y el IPAD. IOS, el sistema operativo que se ejecuta en estos teléfonos, tiene soporte para vídeo MP4.

Comencemos por lo básico, para añadir un video vamos a necesitar el siguiente código:

|  |  |
| --- | --- |
| 1 | 1 |

Bueno para tener más control a esta etiqueta podemos agregar width and height es decir alto y ancho como lo hacemos con las imágenes, quedaría así el código más completo:

|  |  |
| --- | --- |
| 1 | <video src="video.mp4" width="375" height="280"></video> |

El atributo src curso define la dirección del archivo de vídeo, así como para la etiqueta img. También estamos especificando las dimensiones con los atributos width y height .



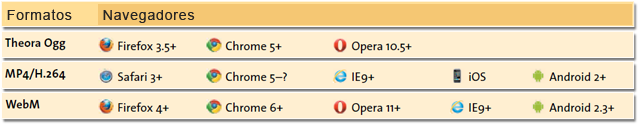
### 1 – Múltiples Fuentes para mostrar el video

Bueno como no hay un formato estándar para los videos en HTML5 debemos de incluir varios archivos de videos con diferentes fuentes de formatos ya que no todos los navegadores leen el mismo formato, y así nos aseguraremos que el video se muestre en todos los navegadores modernos. Esto es lo que hay que hacer hoy en día porque no sabemos si los navegadores o sus fabricantes se pondrán de acuerdo para elegir un solo formato que rija la web con la etiqueta video.

Lo que hacemos es quitar el src dentro de la etiqueta video y crear varios src apuntando a cada video independientemente dentro de los tags <video></video>. Sería algo así:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <video controls="controls" width="375 height="280">      <source src="sintel.ogg" type="video/ogg"/>      <source src="sintel.mp4" type="video/mp4"/>      <source src="sintel.webm" type="video/webm"/>      <!--comentario para los navegadores que no soportan ninguna fuente de video -->Tu navegador no soporta video HTML5  </video> |

A continuación te muestro el formato soportado por cada navegador, es decir los que debemos utilizar o especificar para cada navegador específicamente.



Bueno, no sé si te estarás preguntando como haces para conseguir estos formatos de video para incluirlos en la web. Pero si es así, aquí te dejo unos links para que descargues algunos programas convertidores de videos que te ayudaran a obtener múltiples formatos de video y especialmente los que necesitaremos para insertar en HTML5.

1 – Miro Video Converter (<http://www.mirovideoconverter.com>).  
2 – Handbrake ([http://handbrake.fr](http://handbrake.fr/)).  
3 – Media Converter ([http://www.mediaconverter.org](http://www.mediaconverter.org/)).

Nota: IExplorer 8 y las versiones anteriores no soportan la etiqueta video.

### Atributos de la etiqueta video

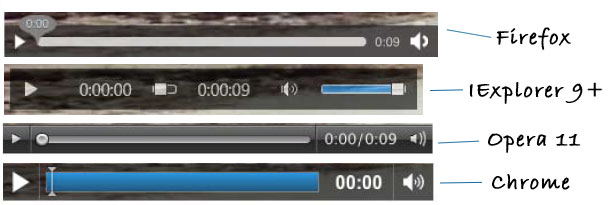
La etiqueta video en HTML5 cuenta con varios atributos para ampliar su funcionamiento, vamos a ver cuáles son:

**\*Autoplay:** Lo que hace este atributo es iniciar automáticamente el video una vez carga la página sin necesidad que el usuario de al play.

|  |  |
| --- | --- |
| 1  2 | <video autoplay="autoplay"> o <video autoplay>  <!--Por ser boolean se puede utilizar de ambas maneras--> |

**\*controls:** Este atributo nos muestra los controles clásicos de (play, pausa, volumen, tiempo de reproducción, etc.) en el player del video (cada navegador muestra su propio reproductor predefinido)

|  |  |
| --- | --- |
| 1 | <video controls="controls"> o <video controls> |



**\*Width and Height**: especificamos la altura y alto del video, lo hacemos en pixeles

|  |  |
| --- | --- |
| 1 | <video width="320" height="240" controls="controls"> |

**\*Src**: El atributo src especifica la ubicación (URL) del archivo de vídeo.

|  |  |
| --- | --- |
| 1 | <source src="video.ogg" type="video/ogg" /> |

Este atributo src tiene un atributo que es **type**, que especifica los recurso de los medios de comunicación. La forma correcta de incluir el atributo src seria este:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <video controls="controls" width="375 height="280">     <source src="sintel.ogg" type="video/ogg"/>     <source src="sintel.mp4" type="video/mp4"/>     <source src="sintel.webm" type="video/webm"/>  </video> |

**\*Loop:** el atributo loop o bucle especifica que el video comenzará otra vez, cada vez que se haya terminado.

|  |  |
| --- | --- |
| 1 | <video controls="controls" loop="loop"> o <video loop> |

**\*Muted:** este atributo específico que la salida de audio del vídeo debe ser silenciado.

|  |  |
| --- | --- |
| 1 | <video controls="controls" muted="muted"> o <video muted> |

**\*poster**: este atributo especifica una imagen que se muestra mientras el vídeo se descarga, o hasta que el usuario pulsa el botón de reproducción. Si este atributo no está incluido, el primer fotograma del vídeo se utilizará en su lugar.

|  |  |
| --- | --- |
| 1  2 | <video poster="URL"> <!--utilizamos href para la ubicación de la imagen-->  <video poster="href="imágenes/poster-video.jpg"> |

**\*Preload:** empezará a precargar el video independientemente de las acciones del usuario sobre el player.

|  |  |
| --- | --- |
| 1 | <video preload="auto|metadata|none"> |

Puede tener las propiedades:

***Auto:***el navegador debe cargar todo el video cuando se carga la página  
***Metadata:***el navegador debe cargar los metadatos sólo cuando se carga la página  
***None:***el navegador no se debe cargar el vídeo cuando se carga la página

Este atributo no se utiliza mucho.

**HABLEMOS DE LAS ETIQUETAS DE AUDIO EN HTML5**

Añadir sonido o archivos de audio a nuestras páginas web suele ser muy importante para algunos proyectos en específico, pero resultaba un poco incómodo que dichos archivos de audios solo eran reproducido si el navegador en cuestión tenía instalado algún plugin (por ejemplo flash).

Por suerte ahora en HTML5 contamos con una nueva etiqueta o un nuevo elemento que especifica un estándar para incluir archivos de audio en nuestras páginas web sin necesidad de plugins que hagan este trabajo.

Se trata de la etiqueta **<audio>** y en este tutorial vamos a hablar de ella, sus atributos y ajustes y compatibilidad con los diferentes navegadores web.

Esta etiqueta funciona igual a su hermano la etiqueta video, claro con diferentes formatos, pero su estructura es parecida.

Para reproducir un archivo de audio en HTML 5 bastaría con abrir la etiqueta audio colocarle le fuente por medio de src que nos dice la ubicación del archivo y cerramos la etiqueta audio, el código básico que necesitaremos es el siguiente:

|  |  |
| --- | --- |
| 1 | <audio src=”Audio.wav”></audio> |

### 1 – Múltiples Fuentes para la etiqueta audio

Al igual que la etiqueta de video y de los diferentes formatos, acá en audio tenemos que especificar lo mismo, múltiples formatos ya que no hay una definida que implemente todos los navegadores. Para esto vamos a utilizar estos formatos WAV, Ogg Vorbis y MP3.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <audio controls>  <source src="Audio.mp3" type="audio/mpeg" />  <source src="Audio.ogg" type="audio/ogg" />  <source src="Audio.wav" type="audio/wav" />  </audio> |

De esta forma triplicamos el espacio preparado para el archivo pero garantizamos su perfecta compatibilidad.

A continuación te presento una tabla con los diferentes principales navegadores y sus archivos de audios soportados:



Nota: IExplorer 8 y las versiones anteriores no soportan la etiqueta audio.

### 2 – Atributos de la etiqueta audio

La etiqueta audio en HTML5 cuenta con varios atributos para ampliar su funcionamiento, vamos a ver cuáles son:

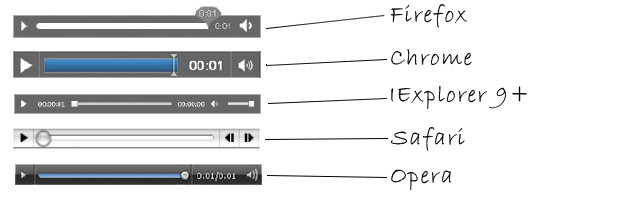
\***atributo autoplay:**Bueno este atributo booleano es bastante intuitivo, lo que básicamente hace es cargar y ejecutar el sonido cuando cargue la página.

|  |  |
| --- | --- |
| 1  2  3 | <audio src=”Audio.wav” autoplay></audio>  <!-- o también sería correcto-->  <audio src=”Audio.wav” autoplay="autoplay"></audio> |

\***atributo controls:** Lo que hace este atributo booleano es hacer que el reproductor de audio de cada navegador muestre los respectivos controles de: reproducir, pausa, volumen etc… esta sería la forma de utilizarlo:

|  |  |
| --- | --- |
| 1  2  3 | <audio src=”Audio.wav” controls></audio>  <!-- o también sería correcto-->  <audio src=”Audio.wav” controls="controls"></audio> |

Hay que tener en cuenta las diferentes formas de mostrase el reproductor en los diferentes navegadores, recuerda que no son iguales y que podemos ajustarlos con css. Estos son los reproductores en los principales navegadores:



\***atributo loop:** Otro atributo booleano, que indica que el audio es ser repetido continuamente, es decir que funcione como bucle, se termina y vuelve a comenzar.

|  |  |
| --- | --- |
| 1  2  3 | <audio src=”Audio.wav” loop></audio>  <!-- o también sería correcto-->  <audio src=”Audio.wav” loop="loop></audio> |

\***atributo preload:** empezará a precargar el video independientemente de las acciones del usuario sobre el player. Al igual que en la etiqueta video este atributo generalmente no es utilizado, esta atributo tiene tres parámetros: auto, metadata y none.

**auto:**el navegador debe cargar todo el audio cuando se carga la página**metadata:**el navegador debe cargar los metadatos sólo cuando se carga la página**none:** informa al navegador que no hay necesidad de realizar la precarga.

|  |  |
| --- | --- |
| 1 | <audio preload="auto|metadata|none"> |

**\*atributo src:** Este atributo src especifica la ubicación (URL) del archivo de audio, ya sea una url interna o externa.

|  |  |
| --- | --- |
| 1 | <audio src="URL"> |

Este atributo src tiene un atributo que es **type**, que especifica los recurso de los medios de comunicación una forma correcta para utilizar este atributo es asi:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <audio controls>     <source src=”Audio.wav” type=”audio/wav" />     <source src=”Audio.ogg” type=”audio/ogg" />     <source src=”Audio.mp3” type=”audio/mpeg" />  </audio> |



## Canvas

Una de las grandes novedades de HTML5 es el elemento canvas que en español la podemos llamar “lienzo”, que lo podemos utilizar para dibujar gráficos en el documento web utilizando secuencias de comando (generalmente javascript). Vamos a poder dibujar cualquier cosa que nos podamos imaginar claro está (utilizando javascript), esto es realmente buenos para nuestros proyectos web, por que mejorara el rendimiento de las misma ya que no vamos a tener que cargar muchas imágenes como antes, ya que con Canvas vamos a crear formas, líneas, degradados, textos, arcos, patrones y muchas cosas más, como el manipular pixeles de las imágenes y manipular videos. En este tutorial veremos lo básico para empezar a trabajar con canvas y en posteriores tutoriales avanzaremos mucho más.

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.

Primero veamos el soporte de canvas en los diferentes navegadores:

soporte canvas navegadores

Bueno antes de empezar ten en cuenta que Usar el elemento <canvas> no es muy difícil, pero sí es necesario un conocimiento básico de HTML y JavaScript . Ahora si, a Comenzar

### Creación de un elemento Canvas

El primer paso para la utilización de canvas es añadir su etiqueta <canvas> y su cierre </canvas>.

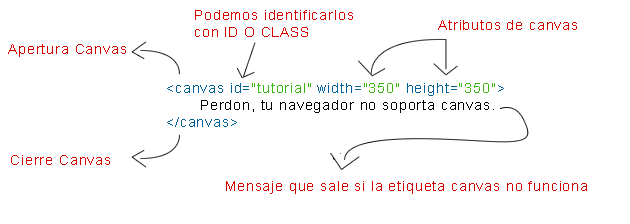
|  |  |
| --- | --- |
| 1  2  3 | <canvas>      Perdón, tu navegador no soporta la etiqueta Canvas.  </canvas> |

En el código anterior podemos ver la apertura y cierre del elemento canvas, dentro de estas etiquetas podemos colocar un mensaje que solo será leído o solo se interpretara si el navegador no soporta canvas. Podemos colocar cualquier mensaje incluso un link para invitar a actualizar su navegador.

|  |  |
| --- | --- |
| 1  2  3 | <canvas id="tutorial" width="350" height="350">      Perdón, tu navegador no soporta canvas.  </canvas> |

A este elemento le podemos aplicar 2 atributos width, height (ancho y alto) estos atributos son opcionales es decir podemos especificar o no. Ahora, al no especificar el ancho ni el alto el valor inicial será 300px de alto \* 150px de alto.

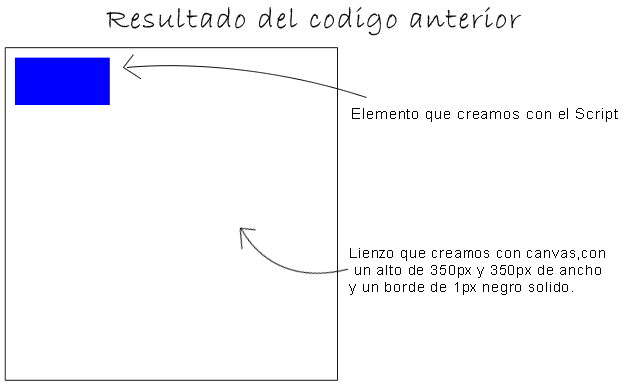
Ahora, también le podemos identificarlo por medio de un ID o una Clase para poder personalizarlo mejor por medio de CSS y manipularlo o identificarlo mejor con Javascript.



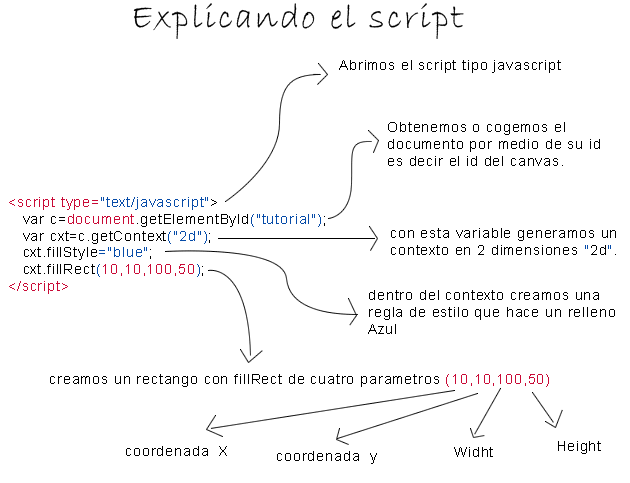
#### Primer Ejercicio con canvas.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <html>  </head>  <body>    <!-- Comienzo de el lienzo canvas -->  <canvas id="tutorial" width="350" height="350" style="border: 1px solid #000000;">      Perdón, tu navegador no soporta canvas.  </canvas>  <!-- fin de el lienzo canvas -->    <!-- Comienzo de el script  -->  <script type="text/javascript">     var c=document.getElementById("tutorial");     var cxt=c.getContext("2d");     cxt.fillStyle="blue";     cxt.fillRect(10,10,100,50);  </script>  </body>  </html> |

Tendremos el siguiente resultado:



Ahora analicemos el script que nos produjo el cuadro azul:



Claro, esto es lo más simple de canvas, como dije al principio, podemos crear muchas cosas sorprendentes con esta nueva herramienta de HTML llamada canvas o lienzo en español.

DIBUJEMOS FORMAS BASICAS CON CANVAS

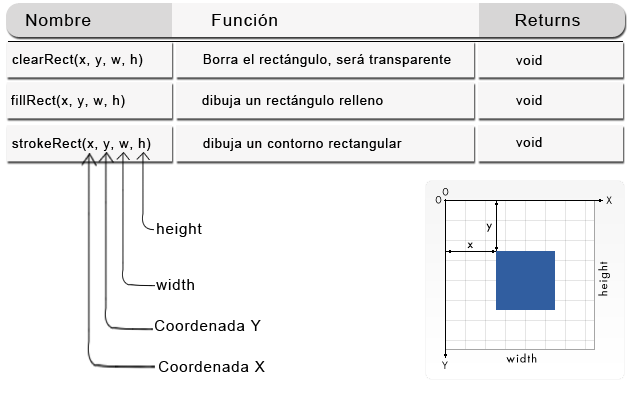
En este [primer acercamiento a canvas](http://www.tutosytips.com/dia-12-empecemos-a-trabajar-con-canvas-en-html5/) vimos cómo hacer funcionar la etiqueta canvas que es nueva en HTML5, en este tutorial veremos cómo avanzar un poco más y lograr crear formas básicas con canvas.

Bueno, antes de comenzar debemos de saber algo de canvas en cuantos a las formas básicas, es que solo admite rectángulos de forma nativa, ahora esto no significa que no podemos crear otras formas, esto lo vamos a poder lograr combinando rutas por medio de funciones.

## 1 – Dibujar Rectángulos Con Canvas

Vamos a comenzar viendo cómo podemos hacer para crear rectángulos, que viene a ser lo más básico que podemos hacer con canvas.

Métodos simples con la cual vamos a crear rectángulos:

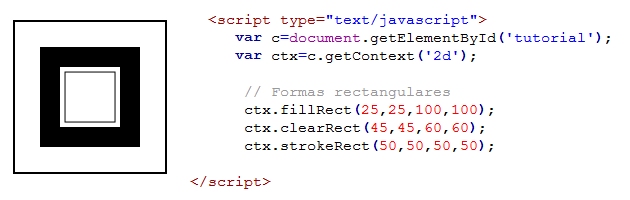


Los tres métodos toman cuatro argumentos (x,y,w,h) que son los desplazamientos desde la esquina superior del elemento, el ancho y alto, bueno en al cuadro anterior te explico cómo funcionan las coordenadas de desplazamiento y el lienzo del elementos en canvas.

Veamos un Ejemplo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <html>    <head>    </head>     <body>        <canvas id="tutorial" width="150" height="150" style=" border: 2px solid #000000"></canvas>        <script type="text/javascript">           var c=document.getElementById('tutorial');           var ctx=c.getContext('2d');              // Formas rectangulares            ctx.fillRect(25,25,100,100);            ctx.clearRect(45,45,60,60);            ctx.strokeRect(50,50,50,50);        </script>      </div>    </body>  </html> |

El resultado seria el siguiente cuadro:



En resumen esto fue lo que hicimos:

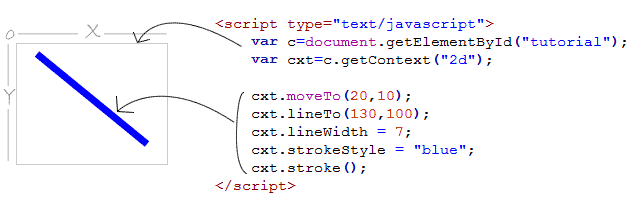
1º – se crea un cuadro negro de 100\*100px con la función FillRect  
2º – se elimina un cuadro de 60\*60px con la función ClearRect  
3º – se crea un contorno rectangular de 50\*50px con StrokeRect

## 2 – Trazar Una Línea Con Canvas

Una línea es el elemento más básico que podemos dibujar, ya sea en un papel o en HTML5 usando la nueva etiqueta canvas, vamos a ver cómo crear líneas con canvas.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | <html>    <head>      <title>creando un circulo con canvas</title>    </head>       <body>        <canvas id="tutorial" width="150" height="120" style="border: 1px solid #ccc;">           Tu navegador no soporta canvas. Actualizalo        </canvas>      <script type="text/javascript">          var c=document.getElementById("tutorial");          var cxt=c.getContext("2d");            cxt.moveTo(20,10);          cxt.lineTo(130,100);          cxt.lineWidth = 7;          cxt.strokeStyle = "blue";          cxt.stroke();      </script>    </body>  </html> |

Ahora veremos el resultado y su explicación.



**cxt.moveTo(20,10);**= Creamos un punto de partida para la línea en el punto (x,y) de acuerdo a las coordenadas partiendo del eje 0.

**cxt.lineTo(130,100);**= Creamos un punto de llegada para la línea en el punto (x,y) de acuerdo a las coordenadas partiendo del eje 0.

**cxt.lineWidth = 7;** =Establecemos un ancho para la línea de 7px.

**cxt.strokeStyle = “blue”;** = Asignamos un color a la línea, en este caso será azul.

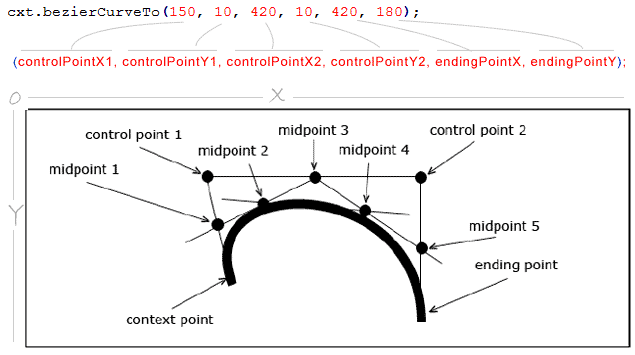
**cxt.stroke();** = hacemos que la línea sea visible con el color del trazo

## 3- Trazar O Dibujar Una Curva Bézier Con Canvas

Para crear curvas con líneas con canvas vamos a necesitar más parámetros en las funciones y utilizar más lógica matemática para saber dónde colocar los puntos para hacer las curvas. Veamos el siguiente código que nos crea una línea curva.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | <html>    <head>      <title>creando un circulo con canvas</title>    </head>       <body>        <canvas id="myCanvas" width="600" height="250" style="border:1px  solid black;">  </canvas>      <script type="text/javascript">         var c=document.getElementById("myCanvas");         var cxt=c.getContext("2d");           cxt.lineWidth = 10;         cxt.strokeStyle = "black";         cxt.moveTo(180, 130);           cxt.bezierCurveTo(150, 10, 420, 10, 420, 180);         cxt.stroke();      </script>    </body>  </html> |

La función que realmente hay nueva y que tenemos que explicar es **cxt.bezierCurveTo**. Que es la que nos va a permitir crear este efecto Bézier.



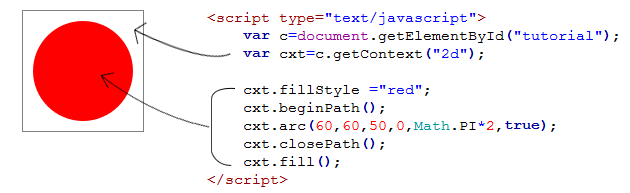
## 4 – Dibujar Círculos Con Canvas

Hay método en canvas que nos permite dibujar círculos y en su defecto arcos. Se trata de cxt.arc, vamos a ver cómo crear un círculo.

Veamos el siguiente código HTML:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | <html>    <head>      <title>creando un circulo con canvas</title>    </head>       <body>        <canvas id="tutorial" width="120" height="120" style="border: 1px solid gray;">           Tu navegador no soporta canvas. Actualizalo        </canvas>      <script type="text/javascript">          var c=document.getElementById("tutorial");          var cxt=c.getContext("2d");            cxt.fillStyle ="red";          cxt.beginPath();          cxt.arc(60,60,50,0,Math.PI\*2,true);                  cxt.closePath();          cxt.fill();      </script>    </body>  </html> |

El resultado seria el siguiente:



Explicación:

**cxt.fillStyle =”red”;**= Creamos un tipo de relleno rojo al elemento que se creara.

**cxt.beginPath();** = Sirve para empezar un recorrido, no se le añadió ningún parámetro.

**cxt.arc(60,60,50,0,Math.PI\*2,true);** = Esta función es la más importante ya que nos crea un arco y nos sirve para hacer el circulo, y los parámetros son: 60 en su centro en X, 60 en su centro en Y, 50px radio en desde el centro, que es el punto que creamos con las coordenadas X, Y, luego empezamos un circulo desde 0 hasta el final del arco expresado en radianes por qué no soporta grados, (1 vuelta completa de circulo es igual a 2 PI Radianes), y sacamos a PI de la librería Math, entonces obtenemos Math.PI\*2 y por ultimo true para decir que queremos ver la parte positiva del arco es decir se hace el recorrido de acuerdo a como giran las manecillas del reloj.

**cxt.closePath();** = Cerramos el recorrido.

**cxt.fill();** = Rellenamos el recorrido y así obtenemos un circulo sencillo con canvas.

———————————–

Como debes de saber, hay muchas cosas para crear con canvas, esto solo es lo mas básico que hay, pero durante poco tiempo seguiremos avanzando.

HTML5 nos permite incluir archivos o imágenes vectoriales directamente al código base, lo hacemos por medio de SVG o “Scalable Vector Graphics”. **SVG** es un archivo especifico que nos permite describir gráficos vectorialmente usando XML, y como debes de saber a diferencia de los formatos tradicionales de la imágenes (GIF, JPEG, PNGY TIFF) las imágenes vectoriales siempre conservan su calidad, nitidez y aspecto no importa cuanto sea esta escalada. Podemos utilizar SVG para hacer muchas tareas que hacemos con CANVAS como alguna figura geométrica, algún texto, degradados o patrones, es muy interesante y yo diría que hacer estas cosas suele resultar más fácil hacerlas con **SVG** que con CANVAS.

Antes de comenzar miremos el soporte de los diferentes navegadores con respecto SVG.

soporte svg en los navegadores

### 1 – Comencemos dibujando en SVG

El código más sencillo de entender para utilizar SVG es el siguiente:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <html>  <head>  </head>  <body>     <svg height=100 width=100>      <circle cx=50 cy=50 r=50 fill=red />     </svg>  </body>  </html> |

Donde creamos el elemento <svg> y lo cerramos </svg>, también le damos una altura y anchura, Luego dentro de el elemento svg dibujamos un circulo con cx=50 es decir su centro x es 50 y de igual para el centro y, r=50 es decir su radio es de 50 y fill o relleno será rojo. Lo que nos resultaría un círculo rojo.

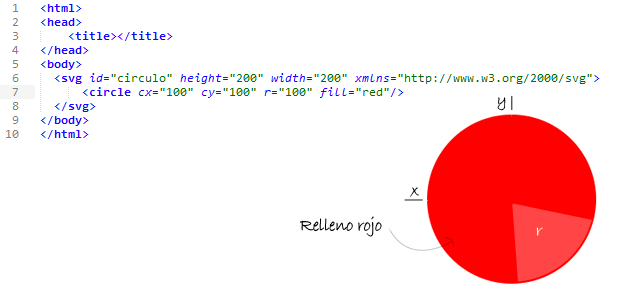
Una forma efectiva y de más posibilidades que tenemos con svg es definir los gráficos o contenido por medio de el formato XML.

La declaración seria la siguiente:

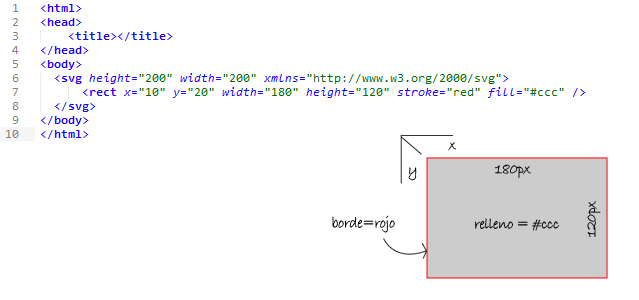
|  |  |
| --- | --- |
| 1  2 | <svg xmlns="http://www.w3.org/2000/svg" version="1.1">  </svg> |

Claro, también le podemos dar un id, una altura y una anchura a este elemento si es requerido. Pero no debe de faltar la declaración que nos indica el lenguaje xml.

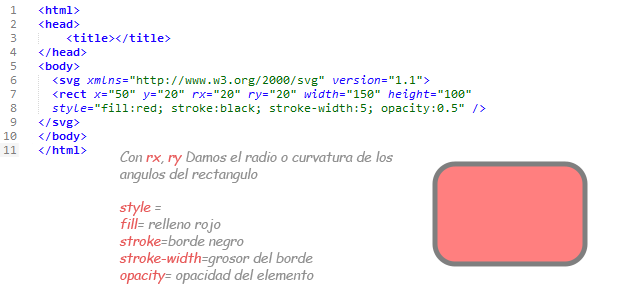
**Dibujando un círculo:**



**Ahora dibujemos un cuadrado:**



**Añadiendo algunos estilos a los elementos:**



Y así podemos obtener y jugar con los gráficos o formas que queramos, solo debemos especificar que elemento predefinidos vamos a utilizar y ajustar sus parámetros básicos.

Definición para las demás formas:

* Rectángulo <rect>
* Círculo <circle>
* Elipse <ellipse>
* Línea <line>
* Polilínea <polyline>
* Polígono <polygon>
* Ruta <path>

Ok, en esta primera parte te he presentado lo básico del trabajo con el elemento SVG de HTML5, en la segunda parte vamos a avanzar un poco más. Nos vemos en ella.

**Ejercicio**

**Actividad: Dibujando HTML5 con HTML5**

En este ejercicio vamos a dibujar.

## 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**

## Busque la posición del usuario

La API de HTML5 Geolocalización se utiliza para obtener la posición geográfica de un usuario.

Dado que esto puede comprometer la privacidad del usuario, la posición no está disponible a menos que el usuario lo aprueba.

## Soporte del navegador

Internet ExplorerFirefoxOperaGoogle ChromeSafari

Internet Explorer 9 +, Firefox, Chrome, Safari y Opera Geolocalización apoyo.

**Nota:** Geolocalización es mucho más precisa para los dispositivos con GPS, como el iPhone.

## HTML5 - El uso de Geolocalización

Utilice el método getCurrentPosition () para obtener la posición del usuario.

El ejemplo siguiente es un ejemplo sencillo Geolocalización devolver la latitud y longitud de la posición del usuario:

**Ejemplo**

<script>  
var x = document.getElementById("demo");  
function getLocation() {  
    if (navigator.geolocation) {  
        navigator.geolocation.getCurrentPosition(showPosition);  
    } else {  
        x.innerHTML = "Geolocation is not supported by this browser.";  
    }  
}  
function showPosition(position) {  
    x.innerHTML = "Latitude: " + position.coords.latitude +   
    "<br>Longitude: " + position.coords.longitude;   
}  
</script>

Ejemplo explicó:

* Compruebe si se admite Geolocalización
* Si se admite, ejecutar el método getCurrentPosition (). Si no, mostrar un mensaje al usuario
* Si el método getCurrentPosition () tiene éxito, devuelve un coordenadas se oponen a la función especificada en el parámetro (showPosition)
* La función showPosition () obtiene las pantallas de la Latitud y Longitud

El ejemplo anterior es un script muy básico Geolocalización, sin tratamiento de errores.

## Gestión de errores y Rechazos

El segundo parámetro del método getCurrentPosition () se utiliza para controlar los errores. Especifica una función a ejecutar en caso de no obtener la ubicación del usuario:

## Ejemplo

function showError(error) {  
    switch(error.code) {  
        case error.PERMISSION\_DENIED:  
            x.innerHTML = "User denied the request for Geolocation."  
            break;  
        case error.POSITION\_UNAVAILABLE:  
            x.innerHTML = "Location information is unavailable."  
            break;  
        case error.TIMEOUT:  
            x.innerHTML = "The request to get user location timed out."  
            break;  
        case error.UNKNOWN\_ERROR:  
            x.innerHTML = "An unknown error occurred."  
            break;  
    }  
}

Códigos de error:

* Permiso denegado - El usuario no permitió Geolocalización
* Posición disponible - No es posible obtener la ubicación actual
* Tiempo de espera - La operación ha agotado

## Visualización del resultado en un mapa

Para mostrar el resultado en un mapa, es necesario tener acceso a un servicio de mapas que pueden utilizar la latitud y longitud, como Google Maps:

## Ejemplo

function showPosition(position) {  
    var latlon = position.coords.latitude + "," + position.coords.longitude;  
  
    var img\_url = "http://maps.googleapis.com/maps/api/staticmap?center=  
    "+latlon+"&zoom=14&size=400x300&sensor=false";  
  
    document.getElementById("mapholder").innerHTML = "<img src='"+img\_url+"'>";  
}

En el ejemplo anterior se utilizan los datos de latitud y longitud vueltos a mostrar la ubicación en un mapa de Google (el uso de una imagen estática).

[Google Map Guión](http://translate.googleusercontent.com/translate_c?depth=1&hl=es&prev=/search%3Fq%3Dw3c%2Bhtml5%26es_sm%3D93%26biw%3D1366%26bih%3D662&rurl=translate.google.com.mx&sl=en&u=http://www.w3schools.com/html/tryit.asp%3Ffilename%3Dtryhtml5_geolocation_map_script&usg=ALkJrhihYSKjIUidqtUb0jx0uwwN4iFFGA)   
Cómo utilizar una secuencia de comandos para mostrar un mapa interactivo con un marcador, zoom y las opciones de arrastre.

## Información específica de la ubicación

Esta página demuestra cómo mostrar la posición de un usuario en un mapa. Sin embargo, la geolocalización también es muy útil para obtener información específica de la ubicación.

Ejemplos:

* Hasta a la fecha de la información local
* Mostrando puntos de interés cerca del usuario
* Turn-by-Turn Navigation (GPS)

## El getCurrentPosition () Método - Devolver datos

El método getCurrentPosition () devuelve un objeto si tiene éxito. Las propiedades de latitud, longitud y precisión siempre se devuelven. Las otras propiedades continuación se devuelven si está disponible.

|  |  |
| --- | --- |
| Property | Description |
| coords.latitude | The latitude as a decimal number |
| coords.longitude | The longitude as a decimal number |
| coords.accuracy | The accuracy of position |
| coords.altitude | The altitude in meters above the mean sea level |
| coords.altitudeAccuracy | The altitude accuracy of position |
| coords.heading | The heading as degrees clockwise from North |
| coords.speed | The speed in meters per second |
| timestamp | The date/time of the response |

## Objeto Geolocalización - Otros métodos interesantes

watchPosition () - Devuelve la posición actual del usuario y continúa regresar posición actualizada como el usuario se mueve (como el GPS en un coche).

clearWatch () - Detiene el metodo watchPosition ().

El siguiente ejemplo muestra el método watchPosition (). Es necesario un dispositivo GPS precisa para probar esto (como el iPhone):

## Ejemplo

<script>  
var x = document.getElementById("demo");  
function getLocation() {  
    if (navigator.geolocation) {  
        navigator.geolocation.watchPosition(showPosition);  
    } else {  
        x.innerHTML = "Geolocation is not supported by this browser.";  
    }  
}  
function showPosition(position) {  
    x.innerHTML = "Latitude: " + position.coords.latitude +   
    "<br>Longitude: " + position.coords.longitude;   
}  
</script>

# 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 ocasión has oído hablar de AJAX. Bien, AJAX significa “Asynchronous JavaScript And XML”… sí, 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 smartphone, tablet y de escritorio.

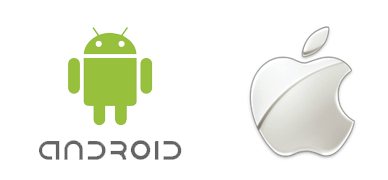
<http://demos.jquerymobile.com/1.4.2/>

**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/html5shim/

# 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/Consorcio/ |
| **Editor:** | W3C |

|  |  |
| --- | --- |
| **Autor:** | Wikipedia |
| **Título:** | HTML 5 |
| **Vínculo:** | http://es.wikipedia.org/wiki/HTML5#Nuevos\_elementos |
| **Editor:** | Wikipedia |