

# Curso básico HTML5



¿que es?, ¿para que es?, ¿En que me beneficio?, y no pueden hacer falta en este mundo del diseño y la estructura web y específicamente en este tema de HTML5



# Introduccion

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

**¿Que es html 5?** – Simplemente Como su nombre indica, HTML 5 es el sucesor de HTML4. Este trabajo o proyecto de HTML5 se inicio 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 todavia sigue en su evolucion, 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.

**¿Porqué 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**.

**¿Porqué 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 estandar, y si no quieres quedarte atras, 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 programacion, HTML como modelo semantico y css3 que es la evolucion del css como el lenguaje de los estilos, que se dedica a dar un mejor aspecto a nuestros proyectos.

## ¿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.
- 

## 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 version HTML 4.01:

```
<!DOCTYPE HTML PUBLIC »
```

```
"-//W3C//DTD HTML 4.01//ES" »
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

Aqui el **<! DOCTYPE>** para la version HTML 1.0:

```
<!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 codigo 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:

```
<!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.

HTML5 no se basa en 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:

<!DOCTYPE html>

Mas Informacion: [http://www.w3schools.com/html5/tag\\_doctype.asp](http://www.w3schools.com/html5/tag_doctype.asp)

## Estructura básica

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:

```
<!DOCTYPE html>

<html

<head>

<meta charset="utf-8"/>

<meta "description"content"curso html5 desde 0" />

<title>Mi primera web en html5</title>

</head>

<body>

<!--aquí se despliega todo el contenido-->

</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 esta 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](http://www.w3schools.com/tags/ref_language_codes.asp).

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

```
<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:

```
<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](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.

```
<title>Aqui va el titulo del documento actual</title>
```

Si quieres mas información acerca esta etiqueta title ve a esta

dirección: [http://www.w3schools.com/html5/tag\\_title.asp](http://www.w3schools.com/html5/tag_title.asp)

y el **<body>** que ya lo conocemos y es el mismo. pero si quieres saber mas acerca del body ve a esta dirección: [http://www.w3schools.com/html5/tag\\_body.asp](http://www.w3schools.com/html5/tag_body.asp)

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.

```
<!DOCTYPE html>
```

```
<html
```

```
<head>
```

```
<meta charset="UTF-8"/>
```

```
<meta "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 </a></li>
<li><a quienes somos</a></li>
<li><a portafolio</a></li>
<li><a 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 enlaces</a></li>
<li><a enlaces</a></li>
</ul>
</aside>
```

<footer>

<p>Aquí va el Pie de página, copyright, el menú otra vez, 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.

1° – **<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>`.

2° – **<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>`.

<hgroup>

<h1>Mi curso de html5</h1>

<h2>Aprende desde 0</h2>

</hgroup>

3° – **<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>`.

<nav>

<ul>

<li><a /></li>

<li><a quienes somos</a></li>

<li><a portafolio</a></li>

<li><a contacto</a></li>

</ul>

</nav>

4° – **<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 mas 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 muy 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 mas que css3, pero vamos por partes.

`<header> </header>`

`<nav></nav>`

`<aside>  
</aside>`

`<section>`

`<article> </article>`

`</section>`

`<footer></footer>`

# Hablemos más de los nuevos elementos de HTML5

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 la 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
<b>&lt;article&gt;</b>	Define un artículo
<b>&lt;aside&gt;</b>	Define el contenido aparte del contenido de la página
<b>&lt;bdi&gt;</b>	Aísla una parte del texto que pueda ser formateada en una dirección distinta de otro texto fuera de ella
<b>&lt;command&gt;</b>	Define un botón de comando que un usuario puede invocar
<b>&lt;details&gt;</b>	Define los detalles adicionales que el usuario pueda ver u ocultar
<b>&lt;summary&gt;</b>	Define un título visible para un elemento <details>
<b>&lt;figure&gt;</b>	Especifica autónomo de contenido, como ilustraciones, diagramas, fotos, listas de códigos, etc
<b>&lt;figcaption&gt;</b>	Define un título para un elemento <figure>
<b>&lt;footer&gt;</b>	Define un pie de página de un documento o sección

<b>&lt;header&gt;</b>	Define un encabezado de un documento o sección
<b>&lt;hgroup&gt;</b>	Agrupar un conjunto de etiquetas desde los <h1> hasta los <h6> cuando un título tiene varios niveles
<b>&lt;mark&gt;</b>	Define el texto marcado / resaltado
<b>&lt;meter&gt;</b>	Define una medida escalar dentro de una gama.
<b>&lt;nav&gt;</b>	Define vínculos de exploración
<b>&lt;progress&gt;</b>	Representa el progreso de una tarea
<b>&lt;ruby&gt;</b>	Define una anotación de rubí (para el Este de la tipografía de Asia)
<b>&lt;rt&gt;</b>	Define una explicación / pronunciación de los caracteres (para el Este de la tipografía de Asia)
<b>&lt;rp&gt;</b>	Define lo que se muestra en los navegadores que no son compatibles con anotaciones de rubí
<b>&lt;section&gt;</b>	Define una sección de un documento
<b>&lt;time&gt;</b>	Define una fecha / hora
<b>&lt;wbr&gt;</b>	Define una posible ruptura del verso

## Nuevos para medios de comunicación / Media

### Etiqueta Descripción

<b>&lt;audio&gt;</b>	Define el contenido de sonido
<b>&lt;video&gt;</b>	Define un vídeo o una película
<b>&lt;source&gt;</b>	Define los recursos de medios múltiples para <video> y <audio>
<b>&lt;embed&gt;</b>	Define un contenedor para una aplicación externa o de contenido interactivo (un plug-in)
<b>&lt;track&gt;</b>	Define las pistas de texto para <video> y <audio>

## El nuevo elemento <canvas>

Etiqueta	Descripción
----------	-------------

<b>&lt;canvas&gt;</b>	Se utiliza para dibujar gráficos sobre la marcha a través de secuencias de comandos (normalmente JavaScript)
-----------------------	--

## Nuevos elementos de formulario

Etiqueta	Descripción
----------	-------------

<b>&lt;datalist&gt;</b>	Especifica una lista de opciones predefinidas para los controles de entrada
-------------------------	---

<b>&lt;keygen&gt;</b>	Define un campo generador de par de claves (para las formas)
-----------------------	--

<b>&lt;output&gt;</b>	Define el resultado de un cálculo
-----------------------	-----------------------------------

## Elementos eliminados de la version anterior de HTML

Los siguientes elementos o etiquetas DE LA VERSION HTML4 fueron eliminados y su uso no sera correcto para la nueva version 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 periodica](#)

# TABLA PERIODICA DE LOS ELEMENTOS DE HTML5



# hablemos de Compatibilidad de HTML5 con los Navegadores.

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%

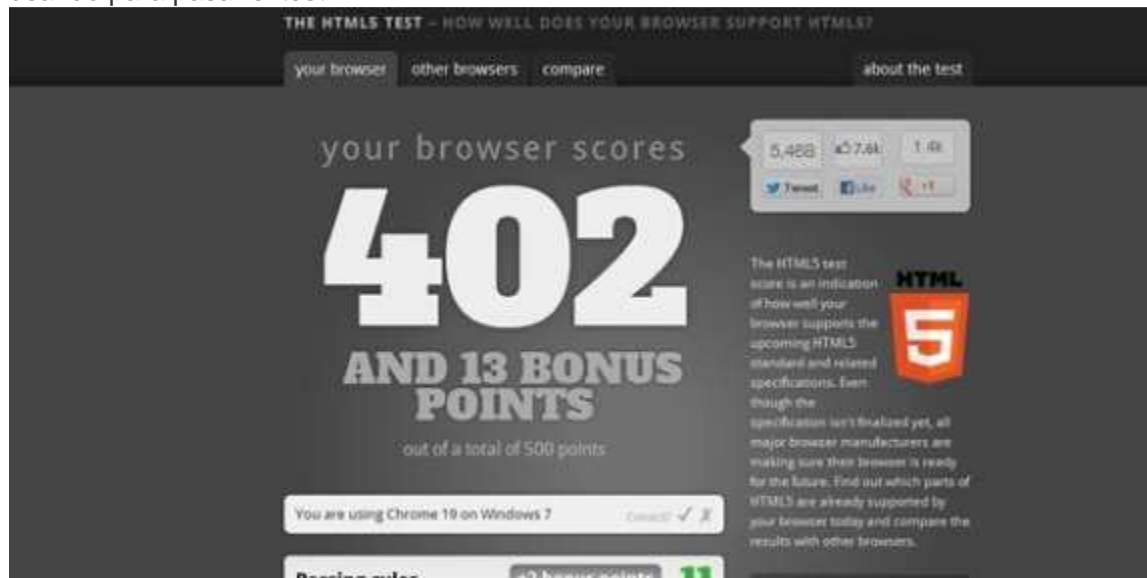
fuelle: [www.netmarketshare.com](http://www.netmarketshare.com)

Si puedes notar el navegador mas usado es Internet Explorer y en su version 8 y el segundo lugar es para el explorer 9, pues te digo que la version 8 nesecita algunos trucos para su conpatibilidad 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. nos indica el soporte de HTML5, ítem por ítem, del navegador que estamos usando para pasar el test.



2 – 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:

```
script "http://html5shiv.googlecode.com/svn/trunk/html5.js"script
```

3 – Modernizr esta libreria le permite ofrecer “respaldo” de estilo en los navegadores que no soportan HTML5 ciertas características de CSS3. Esta libreria puedes detectar si el navegador tiene soporte para multiples capacidades HTML5, Modernizr. De una forma sencilla, nos permite incluirla en nuestras páginas y comprobar la compatibilidad con el navegador.

Modernizr

DESCARGAR DOCUMENTACIÓN RECURSOS NOTICIA

"Modernizr es una parte esencial de mi caja de herramientas de archivos."

— Andy Clarke, y cosas sin sentido

Modernizr es una biblioteca de código abierto de JavaScript que le ayuda a construir la próxima generación de HTML5 y CSS3 sitios web con motor.

¿Por qué utilizar Modernizr?

Aprovechando las nuevas capacidades de HTML5 y CSS3 puede significar el sacrificio de control sobre la experiencia en los navegadores antiguos. Modernizr 2 es el punto de partida para la fabricación de los mejores sitios web y aplicaciones que funcionan exactamente a la derecha, no importa qué navegador o dispositivo para el uso por los visitantes.

Gracias a las pruebas de los nuevos medios y

Descargar Modernizr 2.5.3

Utilice el comando, la versión sin comprime de Desarrollo para desarrollar y aprender de.

A continuación, sumergirse en la producción de herramienta de construcción y recoger sólo las pruebas

Ver la documentación

DESARROLLO

PRODUCCIÓN

Comience con Modernizr

Modernizr le da un mayor control sobre la experiencia a través de la detección de características basada en JavaScript, es importante seguir utilizando las mejores prácticas a través de su proceso de desarrollo. Utilice la mejora progresiva siempre que sea posible, y no sacrificar la accesibilidad para la comodidad o el rendimiento.

- Documentación: Primeros pasos
- Tomando ventaja de HTML5 y CSS3 con Modernizr - Faruk Ateş
- Cómo utilizar Modernizr - Inayati de León

Últimas noticias

04 de febrero 2012

Modernizr 2.5. Supercharged de 2012

Nuestro mayor lanzamiento todavía!

ANTERIOR:

07 de noviembre 2011

¿Cómo funciona Modernizr de Text Suite?

01 de septiembre 2011

Follow us on Twitter

Contribute on github

Subscribe with RSS

Donate to Modernizr

tambien puedes ver el funcionamiento de esta libreria y como instalarla y como usarla en este video:

4 – y por ultimo 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 adelante.

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

# Hablemos de los Microdatos en Html5 (Microdata)

Existe una nueva forma de marcar una frontera semántica en HTML5, se trata de las nuevas posibilidades que tenemos de definir una estructura semántica más fuerte y más personalizada utilizando los microdatos (microdata) que nos van a permitir desarrollar mejor la estructura web dándole una gran prioridad al trabajo del SEO y la indexación a los diferentes buscadores. Los microdatos van a permitir especificar, elementos personalizados en una página Web mediante la sintaxis compuesta de pares nombre-valor con el contenido existente, para entender mejor esto de los Microdatos vamos a ver cómo hacerlo:

Así que vamos a comenzar creando una estructura de un documento HTML5:

```
<!DOCTYPE html>

<html
  <head>
    <meta charset="UTF-8"/>
    <meta "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>
      </nav>
    </header>
    <article>
    </article>
```

```
<aside>
</aside>

<footer>
</footer>

</body>

</html>
```

Como bien sabemos estas nuevas etiquetas de HTML nos van a ayudar mucho para la comunicación con el navegador ya que el va a saber que es un <nav>, que es un <header>, que es un <article> y así toda la estructura de un documento HTML5 con sus nuevas etiquetas.

¿Pero como hago para dar una mejor indexación a lo que es un nombre de una persona, su teléfono, su dirección y así una datos más específicos aparte de las etiquetas estructurales ya establecidas, pero que también nos comunicara el contenido y lo que se refiere de el al navegador?

Para eso están los Microdatos, para trabajar con ellos vamos a requerir 3 elementos,

- 1º – Requiere un **<itemscope>** que es un contenedor de datos
  - 2º – Requiere **<itemprop>** que es el dato propiamente, es decir lo que utilizamos para agregar la propiedad.
  - 3º – Requiere **<itemtype>** Definimos el vocabulario que vamos a utilizar, es decir donde definimos los datos de la persona
- Vamos a ver un Claro ejemplo definiendo algunos de mis datos:

```
<div itemscopeitemtype="<a href="httpwwwdatavocabularyorgPerson" target="blank" data-mce-
href="httpwwwdatavocabularyorgPerson">http://www.data-vocabulary.org/Person/</a>"

<p>Mi nombre es Johan Ricardo,

    Soy diseñador Web,

    Mi web es: http://johanricardo.260mb.com,

    Vivo en Colombia

</p>

</div>
```

Bien, ya tenemos definido lo que es el <itemscope> por medio de un div, y con el itemtype lo que hacemos es definir los datos donde el navegador va a recoger ciertos datos que se daran como las propiedades de cada dato pero ¿Cómo sabe el navegador cual es mi nombre y cuál es mi profesión y cuál es mi pagina web?, pues realmente la respuesta es negativa, el navegador no sabrá nada,

claro está, hasta ahora no sabe nada. Por lo tanto tengo que utilizar la etiqueta <itemprop>, y sería así:

```
itemscope itemtype="<a href="vocabularyPerson" target="_blank" data-mce-  
href="vocabularyPerson">http://www.data-vocabulary.org/Person/</a>"
```

```
<p>nombre itemprop""Johan Ricardo
```

```
dicenitemprop""Johanso
```

```
itemprop""title""ñ
```

```
itemprop""johanricardo260mb.com
```

```
twitter itemprop""contact""tutosytips
```

Después de esto el navegador sabrá cual es mi nombre porque le he puesto la propiedad name, cual es mi apodo por medio de la propiedad Nick y así los demás datos serán interpretados correctamente. Para saber las diferentes propiedades del itemprop ve a esta dirección: <http://www.data-vocabulary.org/Person/>).

En resumen: los microdatos son una mejora para la web, ya que van a permitir a los motores de búsquedas como Google y demás, que la información sea fácilmente indexada, y así detecten todo tipo de información por ejemplo la información del usuario, esto es algo poderoso que realmente debemos utilizar en nuestros proyectos.

# Formularios en HTML5. Nuevos elementos y Nuevos atributos.

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](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 situó dentro del campo y comience a escribir este campo va a desplegar una cantidad de opciones predefinidas anteriormente en su configuración. Ejemplo:

```
<input "Paises-europeos" />
```

```
<datalist "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 porsupuesto te autocompletara españa. Asi 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 especifica si un campo de formulario o de entrada debe tener autocompletado de encendido o apagado, es decir On o Off.

```
<form action="form.php"autocomplete

Nombre:<input "text""fname" /><br />

Apellido: <input "text""lname" /><br />

E-mail: <input "email""email"autocomplete"off" /><br />

<input "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.

```
<form action="form.php"

  Nombre:<input "text""fname"autofocus"autofocus"/><br />

  Apellido: <input "text""lname"autofocus"autofocus"/><br />

  E-mail: <input "email""email"autofocus"autofocus" /><br />

  <input "submit" />

</form>
```

Esto funciona correctamente en chrome.

## 3- El atributo multiple:

El atributo multiple 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.

```
  Seleccione varias imagenes<input "file""img"multiple"multiple" /><br/>

  Entre varios Email:<input "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 envíe la forma sin que el input con este atributo este vacío.

```
<form action="demo_form.asp"

  Nombre: <input "text""name"required"required" />

  <input "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>`.

```
Escojade1a5<input "number""quantity" />
```

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.

[Para ver todos los demas atributos ve aqui.](#)

---

## 2 – Elemento `<output>`:

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

```
<form oninput"x.value=parseInt(a.value)+parseInt(b.value)"  
  
<input "range"value />  
  
<input "number"value />  
  
<output </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>`

[Para saber mas de este nuevo elemento ve aqui:](#)

## 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.

Para saber de este nuevo elemento ve aqui:

**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.

# Hablemos de los Videos en HTML5

En este tutorial o 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 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:

```
<video src="videomp4"></video>
```

bueno para tener mas control a esta etiqueta podemos agregar width and height es decir alto y ancho como lo hacemos con las imagenes, quedaria asi el codigo mas completo:

```
<video "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 – Multiples 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 por que no sabemos si los navegadores o sus fabricante 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>. Seria algo asi:

```
<video controls"controls"width"375 height="280">
```

```
<source src="sintelogg" type="videoogg"/>
```

```
<source src="sintelmp4" type="videomp4"/>
```

```
<source src="sintelwebm" type="videowebm"/>
```

```
<!--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.

Formatos	Navegadores				
Theora Ogg	Firefox 3.5+	Chrome 5+	Opera 10.5+		
MP4/H.264	Safari 3+	Chrome 5-7	IE9+	iOS	Android 2+
WebM	Firefox 4+	Chrome 6+	Opera 11+	IE9+	Android 2.3+

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

2 – Handbrake

3 – Media Converter

*Nota: IE Explorer 8 y las versiones anteriores no soportan la etiqueta video.*

## 2 – Atributos de la etiqueta video

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

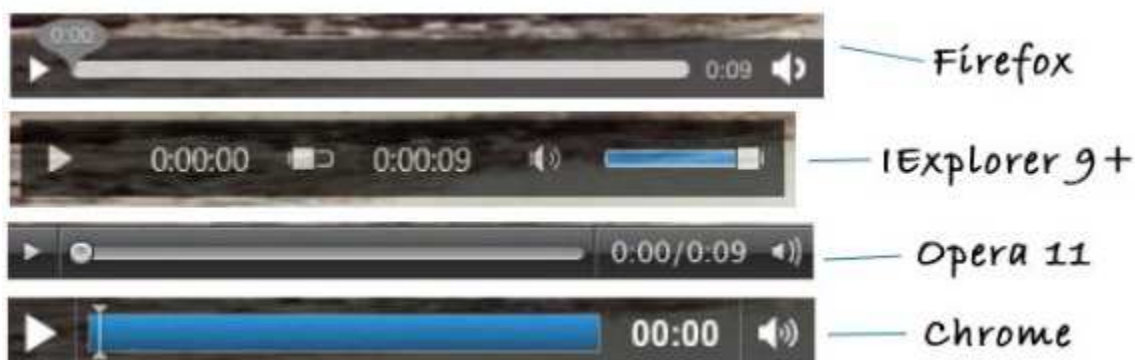
**\*Autoplay:** Lo que hace este atributo es iniciar automáticamente el video una vez carga la pagina sin necesidad que el usuario de al play.

```
<video autoplay">video autoplay
```

<!--Por ser boolean se puede utilizar de ambas maneras-->

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

```
<video controls">video controls
```



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

```
<video width="320" height="240" controls="controls">
```

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

```
<source "video.ogv" type="video/ogg" />
```

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

```
<video controls="controls" width="375" height="280">
```

```
<source src="sintelogg" type="video/ogg" />
```

```
<source src="sintelmp4" type="video/mp4" />
```

```
<source src="sintelwebm" type="video/webm" />
```

```
</video>
```

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

```
<video controls="controls" loop="loop"></video>
```

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

```
<video controls="controls" muted="muted"></video>
```

**\*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.

```
<video poster="URL"><!--utilizamos href para la ubicación de la imagen-->
```

```
<video poster="href="imagenesposter/video.jpg">
```

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

```
<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 la etiqueta 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 incomodo que dichos archivos de audios solo eran reproducido si el navegador en cuestión tenia 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 etique 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 bastaria con abrir la etiqueta audio colocarle le fuente por medio de src que nos dice la ubicacion del archivo y cerramos la etiqueta audio, el código básico que necesitaremos es el siguiente:

```
<audio src="Audiowav"></audio>
```

## 1 – Multiples Fuentes para la etiqueta audio









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

```
<audio controls  
  
<source "Audio.mp3""audio/mpeg" />  
  
<source "Audio.ogg""audio/ogg" />  
  
<source "Audio.wav""audio/wav" />  
  
</audio>
```



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

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

Formatos	Navegadores			
Mp3	 IE9+	 Chrome 5+	 Safari 5+	
Wav	 Chrome 5+	 Safari 5+	 Opera 11+	 Firefox 4+
Ogg	 Firefox 4+	 Chrome 6+	 Opera 11+	

*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 cuales son:

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

```
<audio src"Audiowav"autoplay</audio>
```

<!-- o tambien seria correcto-->

```
<audio src"Audiowav"autoplay"autoplay"</audio>
```

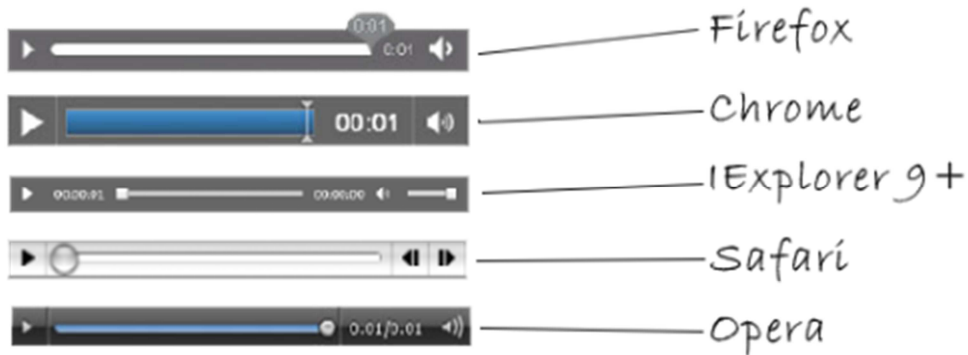
\* **atributo controls:** Lo que hace este atributo booleano es hacer que el reproductor de audio de cada navegador muestre los respectivos controles de: reproducir, pausa, volumen etc... esta sería la forma de utilizarlo:

```
<audio src"Audiowav"controls</audio>
```

<!-- o tambien seria correcto-->

```
<audio src"Audiowav"controls"controls"</audio>
```

Hay que tener en cuenta las diferentes formas de mostrarse 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.

```
<audio src="Audiowav" loop</audio>
```

<!-- o tambien seria correcto-->

```
<audio src="Audiowav""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 parametros: *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.

```
<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.

```
<audio "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:

```
<audio controls
```

```
<source src="Audiowav" type="audiowav" />
```

```
<source src="Audio.ogg" type="audio/ogg" />
```

```
<source src="Audiomp3" type="audiompeg" />
```




```
</audio>
```

---

# Empecemos a trabajar con canvas en HTML5

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 esta (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.

Primero veamos el soporte de canvas en los diferentes navegadores:

Soporte de navegadores de la etiqueta <canvas>					
 IE9+	 Safari 2.0+	 Chrome 3.0+	 Firefox 3.0+	 Opera 10.0+	 Android 1.0+

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>.

```
<canvas>
```

Perdon, tu navegador no soporta la etiqueta Canvas.

```
</canvas>
```

En el código anterior podemos ver la apertura y cierre de el 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.

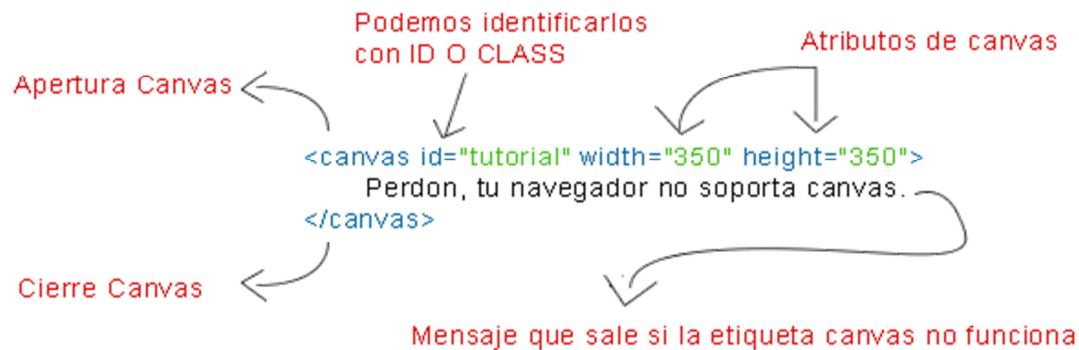
```
<canvas "tutorial"width"350"height"350"
```

Perdon, 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 ancho \* 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.

<html>

</head>

<body>

<!-- Comienzo de el lienzo canvas -->

<canvas "tutorial"width"350"height"350"style"border: 1px solid #000000;"

Perdon, tu navegador no soporta canvas.

</canvas>

<!-- fin de el lienzo canvas -->

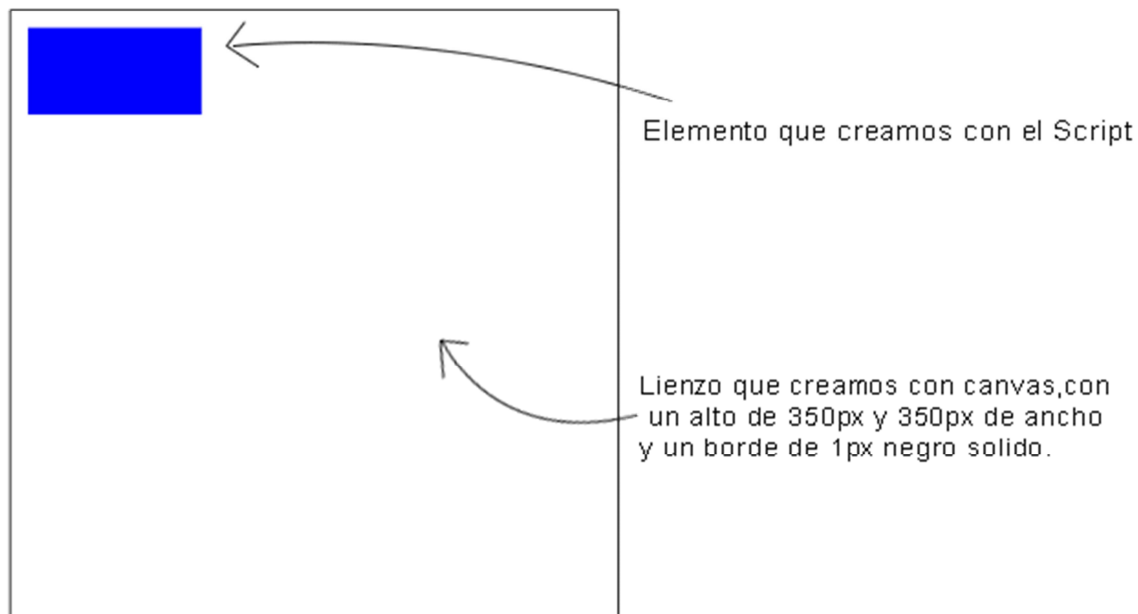
<!-- Comienzo de el script -->

<script "text/javascript"

```
document.getElementById("tutorial")  
getContext  
fillStyle"blue"  
fillRect  
</script>  
</body>  
</html>
```

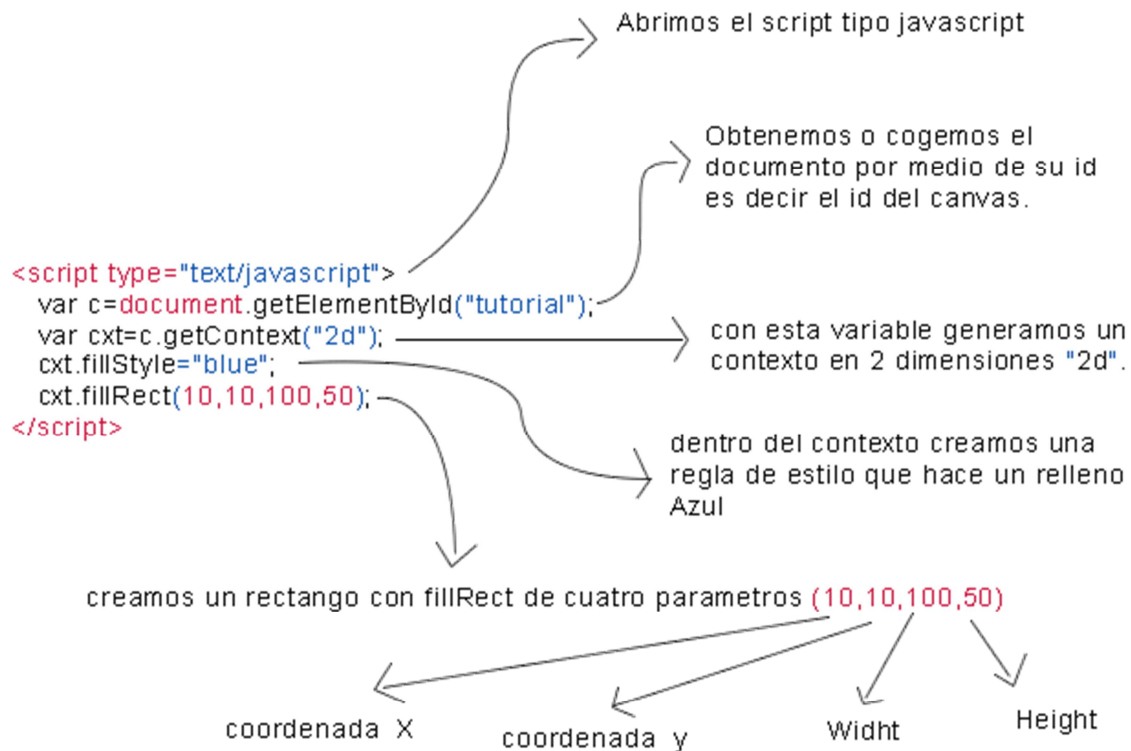
Tendremos el siguiente resultado:

*Resultado del código anterior*



Ahora analicemos el script que nos produjo el cuadro azul:

# Explicando el script



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 básicas con canvas

En este primer acercamiento a canvas vimos como hacer funcionar la etiqueta canvas que es nueva en HTML5, en este tutorial veremos cómo avanzar un poco mas y lograr crear formas básicas con canvas.

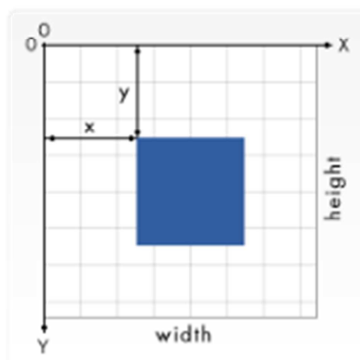
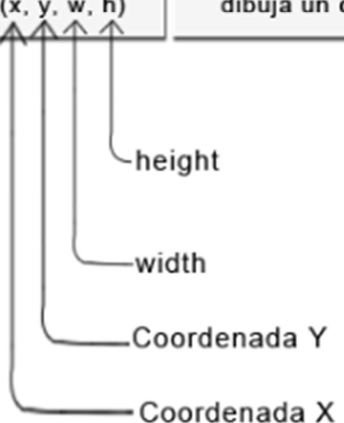
Bueno, antes de comenzar debemos de saber algo de canvas en cuantos a las formas básicas, es que solo admite rectángulos de forma nativa, ahora esto no significa que no podemos crear otras formas, esto lo vamos a poder lograr combinando rutas por medio de funciones.

## 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:

Nombre	Función	Returns
<code>clearRect(x, y, w, h)</code>	Borra el rectángulo, será transparente	void
<code>fillRect(x, y, w, h)</code>	dibuja un rectángulo relleno	void
<code>strokeRect(x, y, w, h)</code>	dibuja un contorno rectangular	void



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 el cuadro anterior te explico cómo funcionan las coordenadas de desplazamiento y el lienzo del elementos en canvas.

Veamos un Ejemplo:

```
<html>

<head>

</head>

<body>

<canvas "tutorial"width"150"height"150"style" border: 2px solid #000000"></canvas>

  <script "text/javascript"

documentgetElementById'tutorial'

getContext

// Formas rectangulares

fillRect

clearRect

strokeRect

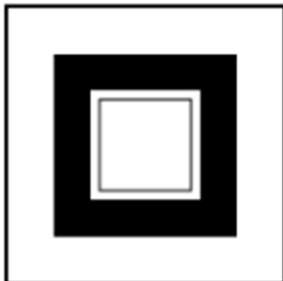
</script>

</div>

</body>

</html>
```

El resultado seria el siguiente cuadro:



```
<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>
```

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.

```
<html>

<head>

<title>creando un circulo con canvas</title>

</head>

<body>

<canvas "tutorial"width"150"height"120"style"border: 1px solid #ccc;"

    Tu navegador no soporta canvas. Actualizalo

</canvas>

    <script "text/javascript"

documentgetElementById"tutorial"

getContext

moveTo

lineTo

lineWidth

strokeStyle"blue"

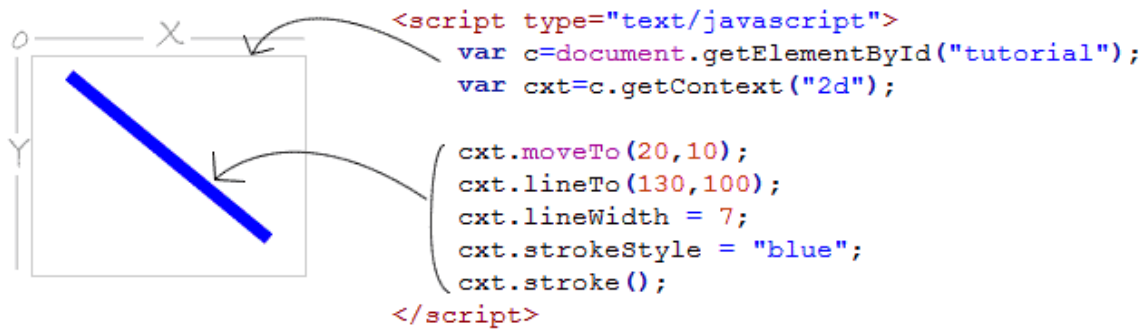
stroke

</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.

```
<html>

<head>

<title>creando un círculo con canvas</title>

</head>

<body>

<canvas "myCanvas"width"600"height"250"style"border:1px
solid black;"

</canvas>

<script "text/javascript"

document.getElementById"myCanvas"

getContext
```

```

lineWidth
strokeStyle"black"

moveTo

bezierCurveTo

stroke

</script>

</body>

</html>

```

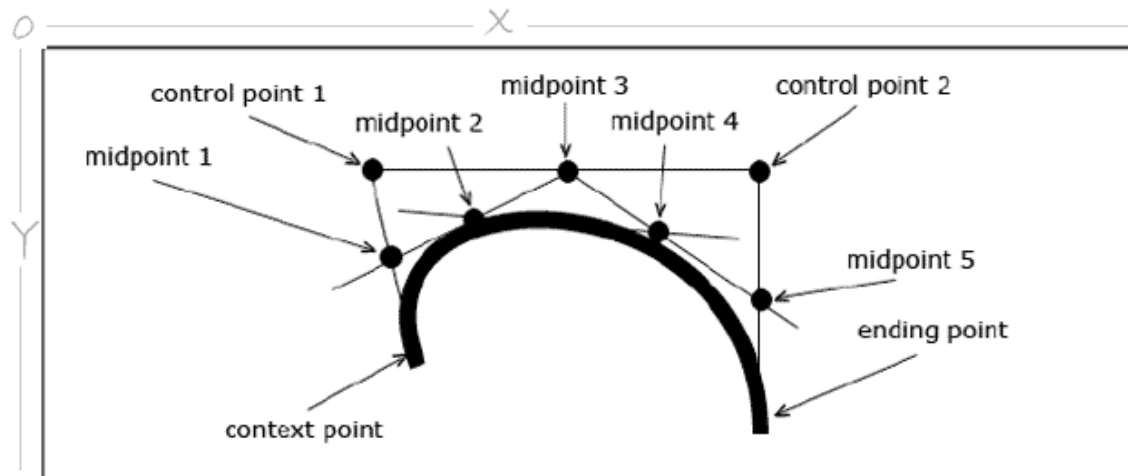
La función que realmente hay nueva y que tenemos que explicar es **cxt.bezierCurveTo**. Que es la que nos va a permitir crear este efecto Bézier.

```

cxt.bezierCurveTo(150, 10, 420, 10, 420, 180);

```

(controlPointX1, controlPointY1, controlPointX2, controlPointY2, endingPointX, endingPointY);



## 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:

```
<html>
```

```

<head>

<title>creando un circulo con canvas</title>

</head>

<body>

<canvas "tutorial"width"120"height"120"style"border: 1px solid gray;"

    Tu navegador no soporta canvas. Actualizalo

</canvas>

    <script "text/javascript"

documentgetElementById"tutorial"

getContext

fillStyle"red"

beginPath

closePath

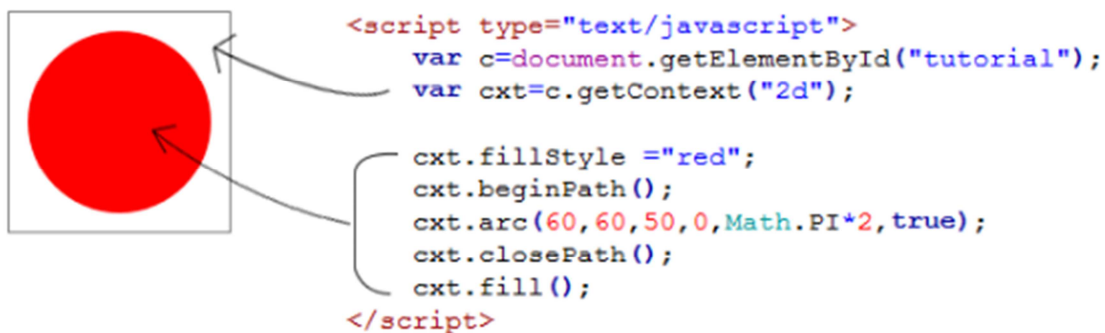
</script>

</body>

</html>

```

El resultado seria el siguiente:



Explicacion:

**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 círculo, 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 círculo desde 0 hasta el final del arco expresado en radianes por que no soporta grados, (1 vuelta completa de círculo 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 círculo 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.

# Dibujemos el logo de HTML5 solo con Canvas

Seguimos viendo lo que es lo de nuevo de HTML5, en este tutorial vamos a avanzar un poco más y vamos a profundizar más en canvas. Vamos a hacer el logo oficial de HTML5 utilizando las técnicas de canvas que vimos anteriormente.

Este logo lo vamos a armar por partes, cada parte de el logo según el tono de color será independiente, será de la siguiente manera:



Te presentare el documento completo y luego te explicaré el proceso para hacer este **logo de HTML5 con canvas**.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML5 Logo con Canvas</title>
```

```
<meta charset="UTF-8"
```

```
<script language="JavaScript"
```

```
factorvalue
```

```
fontfamily"65px 'Gill Sans Ultra Bold', sans-serif"
```

```
function
```

```
documentgetElementById'canvas'getContext
```

```
fontfamily
```

```
dologo
```

```
functiondologo
```

```
offsety
```

fillText"HTML"

translateoffsety

// Escudo de fondo fondo naranja

fillStyle"#E34C26"

beginPath

moveTo

lineTo

lineTo

lineTo

lineTo

closePath

// naranja iluminado de la parte de la derecha

fillStyle"#F06529"

beginPath

moveTo

lineTo

lineTo

lineTo

closePath

// Gris claro del 5 de su parte izquierda

fillStyle"#EBEBEB"

beginPath

moveTo

lineTo

lineTo

lineTo

lineTo

lineTo

lineTo

lineTo

closePath

beginPath

moveTo

lineTo

lineTo

lineTo

lineTo

lineTo

closePath

// el 5 en lado de la derecha

fillStyle"#FFFFFF"

beginPath

moveTo

lineTo

lineTo

lineTo

lineTo

lineTo



```
lineTo
lineTo
closePath
beginPath
moveTo
lineTo
lineTo
lineTo
closePath
</script>
</head>
<body onLoad"init();"
<canvas "canvas"width"600"height"400"
    Tu navegador no soporta canvas
</canvas>
</body>
</html>
```

Explicación de cada parte del código que creamos para formar el logo.

*Clic para ampliar un poco mas la imagen*

```

<!DOCTYPE html>
<html>
<head>
<title>HTML5 Logo con Canvas</title>
<meta charset="UTF-8">
<script language="JavaScript">
var ctx;
var factorvalue = 1;
var fontfamily = "65px 'Gill Sans Ultra Bold', sans-serif";
function init() {
ctx = document.getElementById('canvas').getContext('2d');
ctx.font = fontfamily;
dologo();
}

```

Generamos el contexto del canvas, la fuente que vamos a utilizar en el lienzo

```

function dologo() {
var offsety = 80 ;
ctx.fillText("HTML", 31,60);
ctx.translate(0,offsety);

```

Iniciamos la función dologo que es donde dibujaremos el logo, ajustamos las coordenadas, dibujamos las letras HTML,

```

// Escudo de fondo fondo naranja
ctx.fillStyle = "#E34C26";
ctx.beginPath();
ctx.moveTo(39, 250);
ctx.lineTo(17, 0);
ctx.lineTo(262, 0);
ctx.lineTo(239, 250);
ctx.lineTo(139, 278);
ctx.closePath();
ctx.fill();

```

**HTML**



Pintamos este escudo que formamos por medio de línea con lineTo y moveTo, y lo rellenamos con ese color utilizando fillStyle

```

// naranja iluminado de la parte de la derecha
ctx.fillStyle = "#F06529";
ctx.beginPath();
ctx.moveTo(139, 257);
ctx.lineTo(220, 234);
ctx.lineTo(239, 20);
ctx.lineTo(139, 20);
ctx.closePath();
ctx.fill();

```

**HTML**



Con este bloque de código generamos el reflejo del escudo de la derecha

```

// Gris claro del 5 de su parte izquierda
ctx.fillStyle = "#EBEBEB";
ctx.beginPath();
ctx.moveTo(139, 113);
ctx.lineTo(98, 113);
ctx.lineTo(96, 82);
ctx.lineTo(139, 82);
ctx.lineTo(139, 51);
ctx.lineTo(62, 51);
ctx.lineTo(70, 144);
ctx.lineTo(139, 144);
ctx.closePath();
ctx.fill();

```

**HTML**



Con este bloque de código generamos la primera parte del 5, que tiene como color #EBEBEB que es un gris claro.

```

ctx.beginPath();
ctx.moveTo(139, 193);
ctx.lineTo(105, 184);
ctx.lineTo(103, 159);
ctx.lineTo(72, 159);
ctx.lineTo(76, 207);
ctx.lineTo(139, 225);
ctx.closePath();
ctx.fill();

```

**HTML**



Generamos la otra parte del 5 en su parte gris claro

```

// el 5 en lado de la derecha
ctx.fillStyle = "#FFFFFF";
ctx.beginPath();
ctx.moveTo(139, 113);
ctx.lineTo(139, 144);
ctx.lineTo(177, 144);
ctx.lineTo(173, 184);
ctx.lineTo(139, 193);
ctx.lineTo(139, 225);
ctx.lineTo(202, 207);
ctx.lineTo(210, 113);
ctx.closePath();
ctx.fill();

```

**HTML**



Generamos la parte del 5 de la derecha, que tiene como relleno #FFFFFF que es un blanco limpio

```

ctx.beginPath();

```

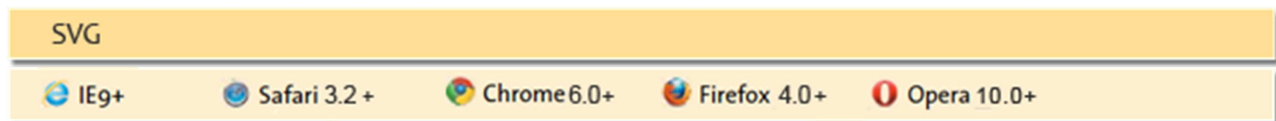
**HTML**

Con Canvas podemos hacer muchas cosas que nos pueden sorprender, es solo profundizar más en el tema con dedicación y los frutos se verán pronto.

# SVG en HTML5

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 específico que nos permite describir gráficos vectorialmente usando XML, y como debes de saber a diferencia de los formatos tradicionales de las imágenes (GIF, JPEG, PNG y 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.



## 1 – comencemos dibujando en SVG

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

```
<html>

<head>

</head>

<body>

<svg height=100width=100

<circle cx=50cy=50r=50fill=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 círculo 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:

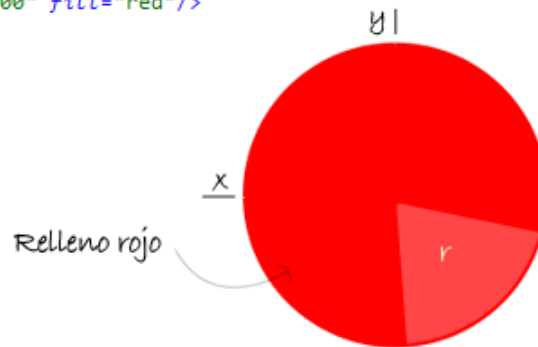
```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"

</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.

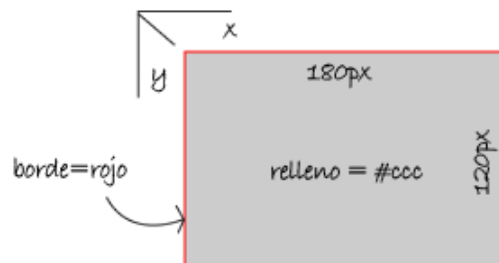
### Dibujando un círculo:

```
1  <html>
2  <head>
3  |   <title></title>
4  </head>
5  <body>
6  |   <svg id="circulo" height="200" width="200" xmlns="http://www.w3.org/2000/svg">
7  |   |   <circle cx="100" cy="100" r="100" fill="red"/>
8  |   </svg>
9  </body>
10 </html>
```



### Ahora dibujemos un cuadrado:

```
1  <html>
2  <head>
3  |   <title></title>
4  </head>
5  <body>
6  |   <svg height="200" width="200" xmlns="http://www.w3.org/2000/svg">
7  |   |   <rect x="10" y="20" width="180" height="120" stroke="red" fill="#ccc" />
8  |   </svg>
9  </body>
10 </html>
```



### Añadiendo algunos estilos a los elementos:

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <svg xmlns="http://www.w3.org/2000/svg" version="1.1">
7     <rect x="50" y="20" rx="20" ry="20" width="150" height="100"
8       style="fill:red; stroke:black; stroke-width:5; opacity:0.5" />
9   </svg>
10 </body>
11 </html>
```

Con **rx**, **ry** Damos el radio o curvatura de los  
ángulos del rectángulo

**style =**  
**fill=** relleno rojo  
**stroke=** borde negro  
**stroke-width=** grosor del borde  
**opacity=** opacidad del elemento



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.

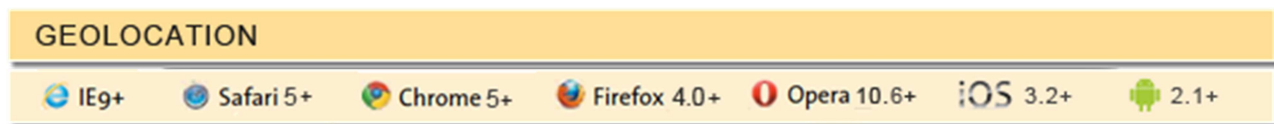
# Geolocalización Parte-1

## Obteniendo las coordenadas del usuario

Seguimos con la serie de tutoriales de HTML5 y vamos a ver cómo funciona el Localizador en html5. La Geolocalización permitirá a sus visitantes a compartir su ubicación actual por medio de su latitud y longitud (*esto si el usuario lo permite*), dependiendo de cómo se está visitando su sitio su ubicación puede ser determinada por cualquiera de las siguientes maneras:

- dirección IP (IP address)
- Conexión de red inalámbrica (wireless network connection)
- GPS (GPS hardware on the device)
- GSM/CDMA Cell IDs

Bueno también el éxito de la Geolocation también depende del navegador que se esté usando, así que vamos a ver cuáles de ellos lo soportan.



La geolocalización es mucho más preciso para los dispositivos con GPS, como el iPhone.

### Propiedades de la ubicación:

La posición especifica la ubicación geográfica actual del dispositivo, esta se expresa como un conjunto de coordenadas geográficas junto con la información de partida y velocidad

PROPIEDAD	VALOR
coords.latitude	Latitud, en grados decimales
coords.longitude	Longitud, en grados decimales
coords.altitude	Altitud, en metros
coords.accuracy	Nivel de precisión de las coordenadas, en metros
coords.altitudeAccuracy	Nivel de precisión de la altitud en metros
coords.heading	Dirección en la que se desplaza el aparato que proporciona las coordenadas, en grados (de 0° a 360°), comenzando desde el norte y contando en el sentido de las agujas del reloj.
coords.speed	Velocidad de desplazamiento del aparato, en metros por segundo.
timestamp	Momentos en que la posición fue adquirida, en formato timestamp (fecha y hora de la respuesta)..

### Geolocation nos proporciona los siguientes métodos

Los métodos JavaScript que utiliza para la ubicación de un usuario los hace a través del Geolocation object contenido en window.navigator.geolocation.

MÉTODO	DESCRIPCIÓN
getCurrentPosition ()	Este método recupera la ubicación geográfica actual del usuario.
watchPosition()	Este método recupera actualizaciones periódicas sobre la ubicación geográfica actual del dispositivo.
clearWatch()	Este método cancela una llamada watchPosition en curso.

Bueno, pienso que haciendo más práctico este asunto vamos a entender mejor el funcionamiento de la geolocalización en html5, solo vamos a obtener las coordenadas del usuario y el el segundo tutorial veremos como traducir esto en un mapa de Googlemaps. Vamos a ver el código completo y luego vamos a analizarlo.

```
<!DOCTYPE html>
```

```
<html
```



```
<head>

<meta charset="utf-8"/>

<title>Ubicación</title>

  <script "text/javascript"

functionlocalizar

navigator.geolocation.getCurrentPositionubicacionerror

functionubicacionposicion

contenedordocument.getElementById"mapa"

latitudposicion.coords.latitude

longitudposicion.coords.longitude

precisionposicion.coords.accuracy

alert"Lat="latitud" - Long="longitud" - Precision="precision

functionerrorerror

error

alert"Error Desconocido"

error

alert"No fue posible contactarte"

error

alert"No hay una ubicacion disponible"

error

alert"Tiempo agotado"

alert"Error Desconocido"

</script>

</head>

<body>

<button onClick"localizar()"Ver mi Ubicación </button>

</body>

</html>
```

Ok, vamos a ver qué significa todo este código, vamos a verlo parte por parte.

## 1 Rutina que obtiene la ubicación del usuario

```
functionlocalizar
```

```
    navigator.geolocation.getCurrentPositionubicacionerror
```

// Dentro de las etiquetas de <script> Creamos una función llamada localizar, y dentro de ella colocamos navigator.geolocation.getCurrentPosition(mapa,error);que lo que hacemos es llamar la función localización del navegador del usuario y nos va a devolver la longitud y latitud del mismo.

Dentro de getCurrentPosition tenemos la opción de incluir dos funciones que luego ampliaremos, pero la primera se cumple si es posible ubicar a el usuario y la segunda si no se puede, a la primera le llamamos ubicación y la segunda error.

## 2 Rutina que imprime las coordenadas

```
functionubicacionposicion
```

```
    contenedor=document.getElementById"ubicacion"
```

```
    latitudposicioncoordslatitude
```

```
    longitudposicioncoordslongitude
```

```
    precisionposicioncoordsaccuracy
```

```
    alert"Lat="latitud" - Long="longitud" - Precision="precision"
```

// En el código anterior creamos la función ubicación, que nos mostrara la ubicación siempre y cuando no haya algún error, dentro creamos tres variables, que por medio de las coordenadas de las propiedades de ubicación vamos a obtener la ubicación exacta, la primera para obtener la latitud, después la longitud y luego la precisión, esto lo mostramos en una ventana emergente de javascript por medio de un alert. Así, ya podemos obtener la ubicación del usuario.

## 3 Rutina de error

```
functionerrorerror
```

```
    error
```

```
    alert"Error Desconocido"
```

```
    error
```

```
    alert"No fue posible contactarte"
```

```
    error
```

```
    alert"No hay una ubicacion disponible"
```

```
    error
```

```
alert("Tiempo agotado")
```

```
alert("Error Desconocido")
```

// Creamos la función error, que es la que tomara el papel principal cuando el método *getCurrentPosition* tenga problemas para obtener la ubicación. Vemos que tenemos unos errores del 0 al 3,

- 0: Error desconocido
- 1: Permiso denegado, el usuario no permite que el navegador envíe datos de geolocalización.
- 2: No se ha podido obtener una ubicación.
- 3: Se ha superado el tiempo de espera máximo para obtener la ubicación.

Lo que hacemos es especificar los errores cada uno con su respectivo mensaje a mostrar en caso de que se de alguno.

#### 4

```
<body>
```

```
<button onClick="localizar()"Ver mi Ubicación </button>
```

```
</body>
```

// Luego en el body creamos un botón con el atributo `onClick="localizar()"` que lo que hacemos es que cuando la persona de click al botón se ponga en marcha la función madre de todo el código de geolocalizar que hemos creado.

Ten en cuenta que debes de decidir si aceptar o no aceptar el acceso de la geolocalización en tu navegador y si ocurre algún problema revisa los permisos de tu navegador.