

Guía teórica HTML 5

UNEWEB

Índice

	Pág.
Qué es HTML 5.	4
Novedades de la estructura	5
• Estructura del cuerpo	5
• Etiquetas para contenido específico	5
• Canvas	5
• Bases de datos locales	5
• Web Workers	6
• Aplicaciones web Offline	6
• Geolocalización	6
• Nuevas APIs para interfaz de usuario	6
• Fin de las etiquetas de presentación	6
Novedades en HTML de HTML5	6
• Etiquetas que nos traen soporte a nuevas funcionalidades:	6
• Etiquetas que componen la web semántica	7
Elementos que dan soporte a nuevas funcionalidades	7
➤ Etiquetas para Multimedia	7
• AUDIO	7
• VIDEO	7
• EMBED	7
• SOURCE	7
• TRACK	7

Nuevos elementos de formulario	7
• METER	7
• PROGRESS	7
• DATALIST	7
• KEYGEN	7
• OUTPUT	8
 Dibujos completos en HTML5, lienzo de CANVAS	 8
 Elementos para la web semántica	 9
➤ Secciones dentro de una página	9
• ARTICLE	9
• SECTION	9
• HEADER	9
• FOOTER	9
• ASIDE	9
• NAV	9
 Otros tipos de informaciones	 9
• BDI	9
• MENU	9
• COMMAND	9
• DETAILS	9
• SUMMARY	9
• FIGURE	10
• FIGCAPTION	10
• HGROUP	10
• MARK	10

• TIME	10
• WBR	10
Etiquetas semánticas del HTML 5	10
Qué son etiquetas semánticas	10
Etiquetas semánticas estructurales	11
Las etiquetas semánticas no definen un estilo predeterminado	14
Por qué debemos utilizar las etiquetas semánticas y cómo nos ayudan	15
Elementos eliminados de HTML 4	16
HTML 5. Atributos eliminados	17
Nuevos elementos	19
Nuevos atributos	20
Atributos globales	21
Algunos ejemplos de uso de las nuevas etiquetas	21
Ejemplo de un Formulario Organizado en una tabla	22
Ejemplo básico de etiquetas semánticas haciendo uso de CSS	25
Crear un Acordeón con CSS3	28

Qué es HTML 5.

Es la quinta revisión importante del estándar que mueve internet, HTML y que ha sido impulsado por el aumento de las necesidades de mejorar la funcionalidad y la interactividad de usuarios y aplicaciones. **HTML 5** presenta una amplia gama de mejoras en controles para formularios, **APIs**, multimedia y también en la **estructura** y la **semántica**. El trabajo en **HTML 5** comenzó en el 2004 y es un esfuerzo conjunto entre el **W3C**, **HTML**, **WC** y el **WHATWG**. Muchos aportan su participación, como Apple, Mozilla, Opera, Microsoft y un sinnúmero de personas y empresas.

La aparición de estas revisiones implicará cambios importantes en el desarrollo de páginas web y algunos se aventuran incluso a prever la desaparición de algunas tecnologías de gran calado en la red, como puede ser Adobe Flash. Ciertamente o no, la realidad es que estamos presenciando una evolución radical de los lenguajes principales de la web que aportará grandes mejoras, pero que al mismo tiempo implicará un periodo de adaptación y aprendizaje por parte de programadores y diseñadores.

Ahora convendría explicar qué es exactamente HTML 5, ya que no es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes al desarrollo web. Es decir, HTML 5 no se limita sólo a crear nuevas etiquetas, atributos y eliminar aquellas marcas que están en desuso o se utilizan inadecuadamente, sino que va mucho más allá.

Así pues, HTML 5 es una nueva versión de diversas especificaciones, entre las que se encuentran:

- HTML 4.
- XHTML 1.
- CSS Nivel 2.
- DOM Nivel 2 (DOM = Document Object Model).

A la par, HTML 5 pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. Para ello se están creando unas APIs que permitan trabajar con cualquiera de los elementos de la página y realizar acciones que hasta hoy era necesario realizar por medio de tecnologías accesorias.

Estas API, que tendrán que ser implementadas por los distintos navegadores del mercado, se están documentando con minuciosidad, para que todos los Browsers, creados por cualquier compañía las soporten tal cual se han diseñado. Esto se hace con la intención que no ocurra lo que viene sucediendo en el pasado, que cada navegador hace la guerra por su parte y los que acaban pagándolo son los desarrolladores y a la postre los usuarios, que tienen muchas posibilidades de acceder a webs que no son compatibles con su navegador preferido.

Novedades de la estructura:

Empezamos con decir que el **DOCTYPE** es el encargado de indicarle al navegador web que el documento que está abriendo es un documento HTML, además de indicar que es un documento HTML también le indica la versión del HTML del mismo con el fin de renderizar la página de manera correcta para mostrarla de la mejor manera posible.

Ahora con la llegada de HTML5, el doctype es bastante sencillo y compatible con las versiones anteriores de HTML y XHTML, esta es su sintaxis.-

<!DOCTYPE html>

HTML 5 incluye novedades significativas en diversos ámbitos. Como decíamos, no sólo se trata de incorporar nuevas etiquetas o eliminar otras, sino que supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

- **Estructura del cuerpo:** La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- **Etiquetas para contenido específico:** Hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- **Canvas:** es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plugin.
- **Bases de datos locales:** el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del

cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet.

- **Web Workers:** son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página. Para ello se dispondrá también de un API para el trabajo con los Web Workers.
- **Aplicaciones web Offline:** Existirá otro API para el trabajo con aplicaciones web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- **Geolocalización:** Las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.
- **Nuevas APIs para interfaz de usuario:** temas tan utilizados como el "drag & drop" (arrastrar y soltar) en las interfaces de usuario de los programas convencionales, serán incorporadas al HTML 5 por medio de un API.
- **Fin de las etiquetas de presentación:** todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

Como se puede ver, existirán varios API con los que podremos trabajar para el desarrollo de todo tipo de aplicaciones complejas, que funcionarán online y offline. Quizás se entienda mejor por qué HTML 5 es un proyecto tan ambicioso y que está llevando tanto tiempo para ser elaborado.

Novedades en HTML de HTML5

Ya dentro del lenguaje HTML específicamente, las novedades que nos trae en su versión 5 son una serie de etiquetas útiles en la web actual, algo bien sencillo de aprender y de aplicar a los sitios web, así como el propio lenguaje de marcación es también sencillo de asimilar. Las podemos clasificar en dos partes:

- **Etiquetas que nos traen soporte a nuevas funcionalidades:** es decir, aquellas que nos sirven para extender el HTML, dando soporte a asuntos como el vídeo o el sonido, lienzos donde diseñar dibujos, etc.

- **Etiquetas que componen la web semántica:** algunas etiquetas que realmente no proponen nuevas funcionalidades, sino que sirven para componer sitios indicando qué son los bloques de código de una web, en lugar de como se deben representar.

Elementos que dan soporte a nuevas funcionalidades

Existen numerosas novedades dentro del HTML5 que se han representado por etiquetas o elementos de diversa índole.

Algunas etiquetas son realmente nuevas y otras como EMBED ya existían y se han incorporado al estándar.

➤ **Etiquetas para Multimedia:** Algunos elementos nuevos servirán para integrar contenido multimedia, pues sabemos que cada día esos nuevos tipos de información están más presentes en la Web.

- **AUDIO:** Para insertar sonido dentro de una web.
- **VIDEO:** Para insertar clips de vídeo.
- **EMBED:** Para embeber contenido externo de otro tipo, como el traído de diversos plugins que se comercializan actualmente o se comercializarán en el futuro.
- **SOURCE:** Permite especificar varias fuentes diferentes cuando se insertan elementos en AUDIO y VIDEO.
- **TRACK:** Permite especificar varias pistas de sonido o vídeo para los elementos AUDIO y VIDEO.

Nuevos elementos de formulario:

En el caso del HTML5 y los formularios tenemos que destacar que no solamente se han creado nuevas etiquetas, sino que se ha añadido soporte a las existentes anteriormente. De momento estas son las nuevas etiquetas que nos ofrece.

- **METER:** Para trabajar con medidas y escalas.
- **PROGRESS:** Implementa barras de progreso.
- **DATALIST:** Extensión para crear campos con funcionalidad de autocompletar.
- **KEYGEN:** Genera claves pública y privada para encriptación.

- **OUTPUT:** Realizar y mostrar cálculos matemáticos.

Pero hablando de novedades en formularios, no debemos dejar de remarcar que HTML5 también trae diversos nuevos usos de INPUT, que nos sirven para especificar qué tipo de información concreta queremos introducir en ellos. Ello proporciona utilidad especial a esos campos INPUT, pensando también en los dispositivos móviles y la forma con la que tratan dichos campos. Todo eso se explica con detalle en segmento dedicado a los tipos de inputs.

Dibujos completos en HTML5, lienzo de CANVAS:

Luego tenemos una utilidad nueva que merece la pena verla por separado, puesto que nos va a transformar la manera con la que se experimentará la web. Se trata de un lienzo en donde se puede dibujar cualquier cosa e incluso hacer animación compleja.

CANVAS: Una etiqueta que genera un lienzo en la página donde realizar cualquier tipo de diseño, soporta dibujo de todo tipo de formas, degradados, imágenes, etc.

Nota: La etiqueta CANVAS del HTML simplemente delimita un área de la página donde se puede dibujar, pero para realizar esos dibujos se tiene que utilizar el lenguaje JavaScript, a través del API de Canvas. Es por eso que no ahondaremos en su uso.

Ejemplo de uso:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Página 1</title>
  </head>
  <body>
    <canvas id="miCanvas">
      Tu navegador no soporta el canvas
    </canvas>
    <script type="text/javascript">
      var canvas=document.getElementById('miCanvas');
      var ctx=canvas.getContext('2d');
      ctx.fillStyle='#FF0000';
      ctx.fillRect(0,0,80,100);
    </script>
  </body>
</html>
```

Elementos para la web semántica

Como decíamos, la segunda clasificación de las nuevas etiquetas del HTML5 está relacionada con lo que se llama la "web semántica". Veamos a continuación una lista de elementos que han sido agregados al HTML para especificar qué son y no cómo se deben mostrar en la página.

➤ Secciones dentro de una página.

Algunas de las nuevas etiquetas nos sirven para decir qué secciones contiene una página:

- **ARTICLE:** Especifica un artículo, es decir, una unidad de contenido.
- **SECTION:** Es una sección dentro de un documento.
- **HEADER:** La cabecera de una página.
- **FOOTER:** El pie de página o informaciones que formen el pie de una sección.
- **ASIDE:** Es una parte de la web que muestra contenido accesorio, generalmente colocado en un panel lateral.
- **NAV:** Con el que colocar el navegador principal de una página web.

Otros tipos de informaciones:

Hay otras muchas etiquetas que nos sirven para definir qué es el contenido que se escribe dentro.

- **BDI:** Define una parte del texto que debe ser entendido aparte de la línea de contenido que se esté escribiendo.
- **MENU:** una lista de opciones que formen parte de un menú.
- **COMMAND:** Uno de los elementos o botones de un menú de opciones.
- **DETAILS:** Detalles o información suplementaria que se puede ver u ocultar por el usuario.
- **SUMMARY:** Encabezamiento para detalles especificados en DETAILS.

- **FIGURE:** es un contenido que ilustre el artículo, como fotos, diagramas, ilustraciones, etc.
- **FIGCAPTION:** El pie o explicación de un FIGURE.
- **HGROUP:** Un grupo de encabezamientos, útil cuando se tiene diversos bloques de encabezamientos del mismo nivel H1, H2...
- **MARK:** Un texto o información que es remarcable.
- **TIME:** Para escribir una fecha, una hora o ambas.
- **WBR:** Define un posible salto de línea.

A continuación veremos unas explicaciones más conceptuales acerca de las etiquetas semánticas y cómo éstas deben cambiar el modo en el que se entiende la composición de una página web.

Etiquetas semánticas del HTML 5.

En el presente segmento vamos a explicar con mayor profundidad qué son las etiquetas semánticas y por qué deben ser utilizadas para mejorar la creación de un documento web. Veremos el conjunto de etiquetas semánticas que sirven para definir la estructura de un sitio y cómo nos pueden ayudar a que nuestra página sea indexada con mayor exactitud por los motores de búsqueda.

Qué son etiquetas semánticas.

Semántica: *Perteneciente o relativo a la significación de las palabras.* Esa es la definición de la Real Academia de la Lengua Española y es así de sencillo es entender este tipo de etiquetas del lenguaje HTML: Aquellas que dan un significado a las partes del documento.

Por decirlo de otra forma, estamos ante **etiquetas que indican qué es el contenido que contienen, en lugar de como se debe formatear** al mostrar el documento HTML en un cliente web.

Para que todas nuestras dudas se despejen, podemos echar mano de un ejemplo. Pensemos en la etiqueta STRONG ¿Qué nos indica? Que un texto debe resaltarse con mayor fuerza con respecto a otros textos que no sean "strong", parecido a lo que nos indica la etiqueta B, que todavía es más concreta, al especificar que un texto debe ser colocado en negrita. Pensemos ahora en una etiqueta DIV, o P. La primera es una división y P es un párrafo. Todo esto no son

más que maneras de decirle al navegador cómo debe formatear una página, qué debe separar en un texto, en una división normal, o en un párrafo, etc.

Pocas de las etiquetas del lenguaje de marcación anteriores a HTML5 indican al documento el significado de los elementos que contienen. O sea, pocas etiquetas indican qué son y qué función representan en el documento HTML, sino que la mayoría sirven para indicar cómo se deben ver en la página una vez renderizada por el navegador.

Pues bien, en HTML5 se han incorporado muchas etiquetas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro. Esas son las etiquetas semánticas, que ganan mucha importancia en el marco del HTML y de la composición de un documento web por ayudar a motores de búsqueda como Google a indexar más correctamente los contenidos de un sitio.

Dentro del etiquetado semántico también tenemos varias funciones, pero para los que nos dedicamos a maquetar páginas web, hay un conjunto de ellas que nos interesan especialmente. Son las que sirven para definir el esquema principal del documento, como HEADER, ARTICLE, FOOTER, etc. Todas esas etiquetas semánticas nos indican qué es el contenido que engloban y cuál es su relación con el conjunto de elementos del documento HTML.

Nota: Hay que añadir que también existen otra serie de etiquetas semánticas nuevas en la versión 5 del lenguaje, que también dicen qué es lo que hay dentro, más que cómo debe representarse. Ejemplos de ellas los vimos en el anterior artículo, como DETAILS, FIGURE, etc. De momento nos interesan menos en lo que respecta a esta guía, pero también hay que tenerlas en cuenta.

Etiquetas semánticas estructurales.

Veremos con detalle a continuación las etiquetas semánticas que sirven para definir la estructura de un documento HTML y por qué debemos incorporarlas cuanto antes a nuestras técnicas de desarrollo de sitios web.

Las etiquetas semánticas estructurales nos sirven para que los motores de búsqueda, o cualquier otro mecanismo automático que lea un sitio web, sepa con exactitud qué partes de su contenido corresponden a cada una de las partes típicas de un sitio.

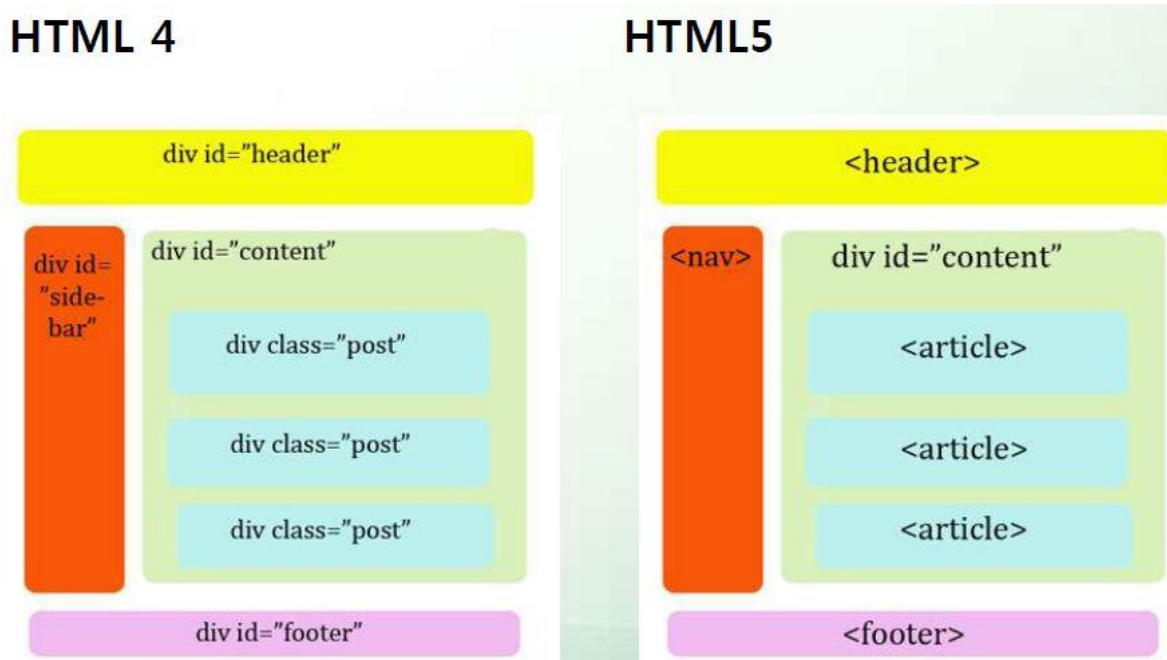
Generalmente, en cualquier documento tenemos una cabecera, un cuerpo y un pie de página, elementos que definen la estructura representados por diversas etiquetas.

De este modo, HTML5 tiene una etiqueta HEADER, que debe englobar aquellos elementos correspondientes a la cabecera de la página. Tiene un FOOTER, que debe englobar todas las etiquetas que componen el pie. Luego, para el cuerpo

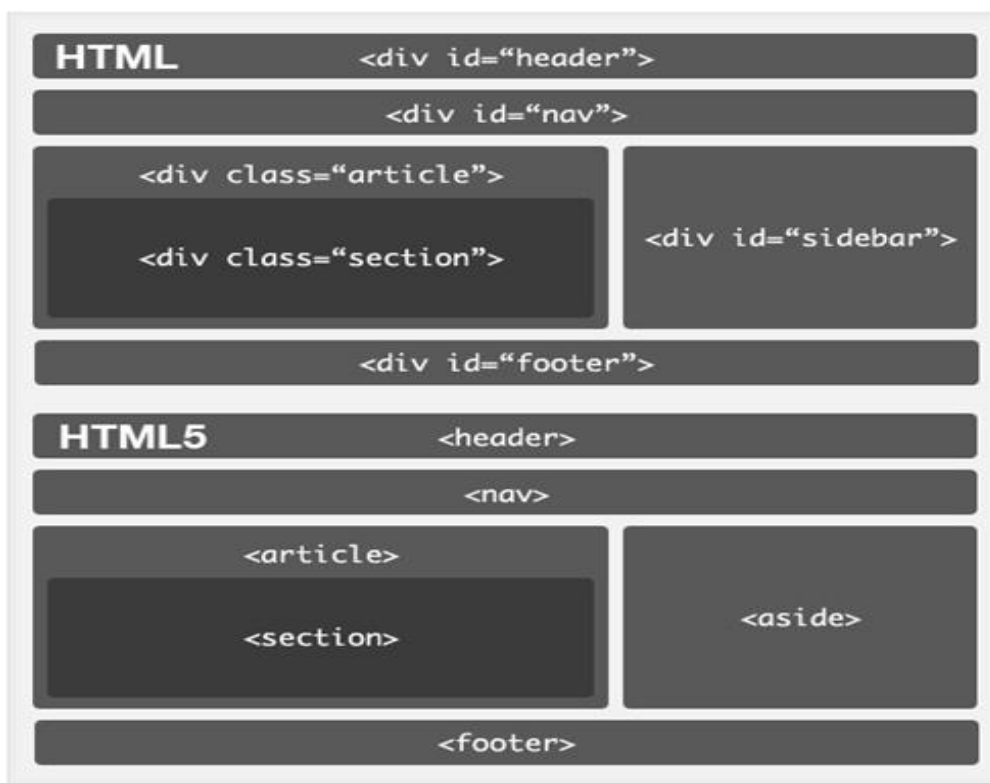
realmente no existe una etiqueta en concreto, sino que se representará por diversas otras como SECTION, ARTICLE o ASIDE. Existirán además otras etiquetas que representan elementos típicos como la barra de navegación, etiqueta NAV.

Observando esas etiquetas semánticas estructurales, cualquier sistema podría procesar la página y saber cómo está estructurada y a qué parte de las típicas de una web corresponde cada contenido. Podrá 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.

A continuación mostramos en una imagen lo anteriormente explicado con respecto a HTML 4.



Otra manera común de estructurar una página:



Como pudieron ver en la imagen existe pues una forma lógica de colocar todos los contenidos dentro de un sitio y unas etiquetas determinadas para contenerlos, atendiendo a la estructura que nosotros queremos comunicar a los sistemas que puedan procesar el documento.

La imagen que acompaña a este texto expresa una de las posibles formas de ordenar el documento con las etiquetas que nos proporciona HTML5.

En la imagen se pueden ver las etiquetas estructurales más habituales, pero hay otras que podríamos utilizar dentro de las etiquetas semánticas. Además, la posición de estos elementos no tiene por qué ser así, incluso podrían anidarse de otras maneras. Quizás en nuestro sitio queramos colocar el navegador en la cabecera, en el lateral derecho o dentro del SECTION. Sería independiente siempre que utilicemos la etiqueta NAV para contenerlo. Asimismo, podrías tener más de un SECTION en el cuerpo de la página, sueltos o colocados dentro de otras etiquetas como ASIDE. Esos SECTION adicionales podrían tener otros ARTICLE o incluso, dentro de un SECTION, hacer otro esquema de cabecera y pie anidando otros HEADER y FOOTER.

Como vemos, las posibilidades de agrupar, ordenar y colocar estas etiquetas para definir la estructura correría por nuestra cuenta, pudiendo generar distintos tipos de estructuras según las necesidades de cada proyecto. No obstante, siempre hay algunas reglas básicas que sí se deben cumplir, como que las unidades de contenido estén en los ARTICLE o que solo exista un navegador con NAV. En casos comunes, que deseemos tener otros navegadores, podríamos optar por echar mano de otras etiquetas semánticas como MENU, asignando al navegador principal la etiqueta NAV y a otros secundarios el MENU.

Nota: Realmente, las etiquetas estructurales están ahí y nosotros somos los que podemos utilizarlas a nuestro gusto, siempre que respetemos la significación que se les ha dado. Hay discusiones de sobra para interpretarlas de una manera o de otra y para decir cómo deberían ser utilizadas. Podemos usar el sentido común para imaginar lo que pueda o no ser correcto.

Las etiquetas semánticas no definen un estilo predeterminado.

Quizás este punto está de más, porque se debe entender a raíz de lo dicho hasta ahora, pero conviene apuntarlo bien para no llevarnos a engaño. Las etiquetas semánticas no tienen un estilo predeterminado que el navegador nos vaya a asignar. Es decir, porque HEADER signifique que es una cabecera, el navegador no va **en ningún caso** a posicionar el elemento en la parte de arriba del documento. Lo mismo con FOOTER, que no lo colocará en la parte de abajo, o ASIDE al lateral.

Somos nosotros los que, mediante CSS, debemos asignar los estilos que queramos se aplique a cada uno de esos elementos del documento HTML. Vuelvo a insistir en que las etiquetas semánticas no indican la forma como se deben mostrar, sino el significado que tienen dentro del contexto del documento.

En este sentido, cabe recordar que desde hacer tiempo se viene trabajando para separar el contenido de la forma y las etiquetas semánticas suponen un paso más hacia ese objetivo. Por ello, todos los estilos y posicionamiento deben quedar a cargo de las CSS.

Por qué debemos utilizar las etiquetas semánticas y cómo nos ayudan.

A algunos les bastará saber que Google recomienda su uso para estar convencidos de utilizarlas, pero las razones de fondo no son caprichosas de una empresa en concreto. Hay muchas compañías grandes detrás de estas nuevas etiquetas semánticas y estructurales y se han creado para facilitar la vida, no solo de Google, sino de cualquier otro sistema que pueda procesar un documento HTML.

La idea con estas etiquetas es añadir información a la Web, como metadatos semánticos que puedan ser interpretados por cualquier sistema informático. Con ello se puede facilitar que cualquier máquina que lee un documento HTML y no está apreciando cómo se presenta en pantalla, sino simplemente su código fuente, pueda saber a qué parte típica pertenece cada pedazo de código, para entender mejor qué es lo que se está tratando en cada caso.

En lo que respecta al motor de Google, o cualquier otro robot de indexación, será capaz de leer el documento y saber qué parte es la cabecera, que parte contiene los enlaces a las principales secciones del sitio, qué parte contendrá las unidades de contenido, etc.

Así Google sabrá mejor de qué trata tu página, o cuales son los distintos temas que se están desarrollando en estos momentos. Sabrá qué enlaces debe prestar especial atención para recorrer tu sitio completo y cuáles de esos enlaces son menos importantes. En definitiva, las máquinas sabrán identificar mejor qué hay en una página, y cuáles son las partes más importantes de entre todo el contenido que tiene.

Por poner ejemplos más concretos, Google será capaz de ver tus ARTICLE para saber qué post has publicado y cuál es el contenido más interesante que estás tratando en estos momentos. Podrá ver tu NAV para saber cómo organizas las secciones de tu sitio. Los POST los puede indexar de manera preferente como nuevos artículos que publicas con cierta periodicidad.

Un ejemplo interesante es lo que Google podrá hacer con tus enlaces del NAV. Si te has fijado, cuando buscas por un sitio en concreto a partir de su nombre, muchas veces aparece ese sitio en primer lugar con una serie de sub-enlaces con las secciones principales. Pues con tu navegador colocado en NAV estarías diciéndole a Google que esas son las secciones principales del sitio, que debe mostrar como páginas alternativas cuando buscan por él en el buscador.

Conclusión.

Estas etiquetas semánticas del HTML5 son un paso adelante en cuanto a crear una web con un contenido mejor formado, más útil y procesable con mayor facilidad por cualquier sistema informático. Hoy puede que solo nos interese como una manera de optimizar los sitios para una mejor indexación en buscadores, pero sentarán las bases para conseguir en el futuro un ecosistema web más saludable.

Elementos eliminados de HTML 4.

En esta sección hablaremos de algunos de los elementos eliminados y las razones de por qué son prohibidos:

Los siguientes elementos (muy usados hace pocos años) se quitan de HTML5 porque son **puramente presentacionales** (no tienen semántica) y todo el tema estético se debe tratar con CSS:

- **basefont**
- **big**
- **center**
- **font**
- **s**
- **strike**
- **tt**
- **u**
- Los elementos para **trabajar con frames** (frame, frameset y noframes) se quitan del estándar por razones obvias: afectan negativamente a la usabilidad y accesibilidad de la web. Además, prácticamente rompen la web, y si se necesita algo similar se puede acudir a los iframe, más potentes y mejor pensados.
- El elemento **acronym** se elimina simplemente porque crea confusión sobre su uso, y los desarrolladores no entienden demasiado bien para qué usarlo.

Las abreviaciones y acrónimos se pueden marcar con abbr, que sí se mantiene en el estándar.

- El elemento **applet** se ha declarado obsoleto y hoy en día no se utiliza. El elemento object reemplaza sus funciones y es lo común hoy en día.
- El elemento **isindex** se quita definitivamente. En la era de las cavernas se utilizaba para mandar información al servidor, pero con la llegada de los formularios su uso es arcaico y poco útil.
- El elemento **dir** también se declara obsoleto (ya lo era en HTML4), y simplemente se recomienda usar listas normales con ul.
- El elemento **noscript** se mantiene en HTML pero no en XML/XHTML, ya que su contenido está en HTML. No estoy muy de acuerdo con este movimiento, pero así será.

HTML 5. Atributos eliminados

Para empezar, todos los atributos referentes a la presentación han sido eliminados, por la misma razón de antes: CSS sirve mejor ese propósito. Recuerdo que el atributo style (que contiene CSS) es ahora universal y puede ser aplicado a todos los elementos, así que si queremos indicar su presentación sin añadir una hoja de estilos aparte, tendremos que usar este atributo. Atención a la lista porque esto sí que es importante, ya que algunos de estos elementos son muy usados, aunque otros están muy obsoletos:

- Atributo **align** en todos los elementos.

- Atributos **alink**, **link**, **text** y **vlink** en el elemento body.
- Atributo **background** en el elemento body.
- Atributo **bgcolor** en los elementos table, tr, td, th y body.
- Atributo **border** en todos los elementos.
- Atributos **cellpadding** y **cellspacing** en el elemento table.
- Atributos **char** y **charoff** en los elementos col, colgroup, tbody, td, tfoot, th, thead y tr.
- Atributo **clear** en el elemento br.
- Atributo **compact** en los elementos dl, menu, ol y ul.
- Atributo **frame** en el elemento table.
- Atributo **frameborder** en el elemento iframe.
- Atributo **height** en los elementos td y th.
- Atributos **hspace** y **vspace** en los elementos img y object.
- Atributos **marginheight** y **marginwidth** en el elemento iframe.
- Atributo **noshade** en el elemento hr.
- Atributo **nowrap** en los elementos td y th.
- Atributo **rules** en el elemento table.
- Atributo **scrolling** en el elemento iframe.
- Atributo **size** en el elemento hr.
- Atributo **type** en los elementos li, ol y ul.
- Atributo **valign** en los elementos col, colgroup, tbody, td, tfoot, th, thead y tr.
- Atributo **width** en los elementos hr, table, td, th, col, colgroup y pre.

Como podemos ver, algunos de estos atributos sí que se mantienen para ciertos elementos, como la anchura y altura en las imágenes. Sin embargo estos no son los únicos atributos que se eliminan, también hay otros que se quitan por redundancia, por evitar confusiones, por su bajo uso o porque simplemente se han quedado obsoletos.

- Atributo **accesskey** en los elementos a, area, button, input, label, legend y textarea.
- Atributos **rev** y **charset** en los elementos link y a.
- Atributos **shape** y **coords** en el elemento a.
- Atributo **longdesc** en los elementos img y iframe.

- Atributo **target** en el elemento link.
- Atributo **nohref** en el elemento area.
- Atributo **profile** en el elemento head.
- Atributo **version** en el elemento html.
- Atributo **name** en los elementos img y a. Para obtener un comportamiento similar se recomienda usar id.
- Atributo **scheme** en el elemento meta.
- Atributos **archive**, **classid**, **codebase**, **codetype**, **declare** y **standby** en el elemento object.
- Atributos **valuetype** y **type** en el elemento param.
- Atributo **language** en el elemento script.
- Atributo **summary** en el elemento table.
- Atributos **axis** y **abbr** en los elementos td y th.
- Atributo **scope** en el elemento td.

Nuevos elementos

Los tiempos modernos requieren nuevos elementos para proporcionar una web más semántica, completa y homogénea. Para ello se han añadido una buena serie de elementos que nos permitirán encapsular más nuestro contenido.

- `<article />` elemento que nos permite declarar un trozo del contenido como artículo. Ideal para blogs o periódicos.
- `<aside />` representa un trozo de contenido que se relaciona muy levemente con el resto del contenido.
- `<dialog />` elemento que permite reprogramar conversaciones.
- `<figure />` podrás usarlo para asociar con un caption un contenido incrustado, como por ejemplo gráficos o vídeo.
- `<footer />` Sección de la página para contener información sobre el autor, copyright, etc,...
- `<header />` representa a la sección de cabecera.
- `<nav />` representa la sección de la página orientada a la navegación.
- `<section />` elemento que indica que se trata de una sección genérica.
- `<audio />` y `<video />` para el contenido multimedia.
- `<embed />` es un elemento dedicado para contenido de plugins.
- `<m />` representa el texto marcado.
- `<meter />` usado para representar medidas, por ejemplo el tamaño del disco usado...
- `<time />` usado para mostrar fechas y/o tiempo.

- `<canvas />` usado para mostrar gráficos renderizados en tiempo real, por ejemplo gráficos, juegos, etc,...
- `<command />` relacionado con los comandos que el usuario puede invocar.
- `<datagrid />` ideal para mostrar un árbol de datos o una tabla tabulada.
- `<details />` muestra información adicional si el usuario lo demanda.
- `<datalist />` junto con el nuevo atributo `list` para los `<input />` puede ser usado para crear comboboxes.
- `<event-sources />` puede ser usado para capturar eventos enviados desde servidor.
- `<output />` nos indica que tipo de salida vamos producir con nuestra página.
- `<progress />` representa una barra de proceso de una tarea, por ejemplo descargar.

Los elementos de entrada `<input />` dispondrán de una serie de tipos (type) nuevos para indicar los diferentes tipos de elementos de entrada posibles.

- `datetime`
- `datetime-local`
- `date`
- `month`
- `week`
- `time`
- `number`
- `range`
- `email`
- `url`

Nuevos atributos

HTML 5 ha introducido una gran cantidad de nuevos elementos para varios elementos de los que ya disponemos en la HTML4.

- `media`: Para conseguir una mayor consistencia con el link en los elementos `<a />`
- `ping`: Especificaremos una lista separada por espacios donde produciremos un ping cuando se siga el enlace, para los elementos `<area />` y `<a />`
- `target`: Disponible para mejorar la consistencia con el elemento `<a />`.
- `autofocus`: Destinado para indicar el elemento `<input />` (no hidden), `<select />`, `<textarea />` o `<button />` que ha de coger el foco al cargar la página.
- `form`: Atributo para `<input />`, `<output />`, `<select />`, `<textarea />`, `<button />` y `<fieldset />` que permite que se asocien con un simple formulario.
- `replace`: atributo para `<input />`, `<button />` y `<form />` que le afectará cuando el contenido del elemento sufra algún cambio.

- data: Para <form />, <select /> y <datalist />.
- required: Para elementos <input /> (Excepto hidden e image) y <textarea />, indica que el campo es obligatorio.
- inputmode: Atributo para <input /> y <textarea />.
- disabled: Para <fieldset />, permite desactivar el fieldset completo.
- autocomplete, min, max, pattern, step: Para elementos <input /> permite delimitar las posibilidades de nuestros elementos de entrada.
- list: Para elementos <datalist /> y <select />.
- template: Para <input /> y <button /> podrá usarse para repetir templates.
- scoped: Para elemento <style />, permitirá usar hojas de estilo "scoped".
- async: Para el elemento <script /> el ajax hecho atributo.

Atributos globales

- Aparte de los ya existentes: class, dir, id, lang y title.
- contenteditable: indica que se trata de un área editable.
- contextmenu: Puede ser usado como punto de menú contextual proporcionado por el usuario.
- draggable: indica que se trata de un elemento draggable.
- tabindex: indica la posición numérica a la que llegaremos pulsando la tecla TAB.
- irrelevant: atributo que indica que el contenido no es relevante.
- repeat, repeat-start, repeat-min, repeat-max: atributos referentes a las iteraciones.

Algunos ejemplos de uso de las nuevas etiquetas.

Formularios:

```
<form action="." oninput="range_control_value.value =
range_control.valueAsNumber">
  <table>
    <tr>
      <th>
        Nombre:
      </th>
      <td>
        <input type="text" name="name_control" autofocus required />
      </td>
    </tr>
    <tr>
      <th>
```

	Correo Electrónico:
	<input ><="" name="email_control" required="" td="" type="email"/>
	URL:
	<input ><="" name="url_control" placeholder="Escribe la URL de tu página web personal" td="" type="url"/>
	Fecha:
	<input ><="" name="date_control" td="" type="date"/>
	Tiempo:
	<input ><="" name="time_control" td="" type="time"/>
	Fecha y hora de nacimiento:
	<input ><="" name="datetime_control" td="" type="datetime-local"/>
	Mes:
	<input ><="" name="month_control" td="" type="month"/>


```

<tr>
  <th>
    Semana:
  </th>
  <td>
    <input type="week" name="week_control" />
  </td>
</tr>
<tr>
  <th>
    Número (min -10, max 10):
  </th>
  <td>
    <input type="number" name="number_control" min="-10"
    max="10" value="0" />
  </td>
</tr>
<tr>
  <th>
    Intervalo (min 0, max 10):
  </th>
  <td>
    <input type="range" name="range_control" min="0" max="10"
    value="0" />
    <output for="range_control" name="range_control_value"
    >0</output>
  </td>
</tr>
<tr>
  <th>
    Teléfono:
  </th>
  <td>
    <input type="tel" name="tel_control" />
  </td>
</tr>
<tr>
  <th>
    Término de búsqueda:
  </th>
  <td>
    <input type="search" name="search_control" />
  </td>
</tr>
<tr>
  <th>
    Color Favorito:

```

```

        </th>
        <td>
            <input type="color" name="color_control" />
        </td>
    </tr>
    <tr>
        <th colspan="2">
            <input type="submit" value="Enviar!" />
        </th>
    </tr>
</table>
</form>

```

Con este código podremos observar el siguiente resultado:

Nombre:	<input type="text"/>
Correo Electrónico:	<input type="text"/>
URL:	<input type="text" value="Escripe la URL de tu página"/>
Fecha:	<input type="text" value="dd/mm/aaaa"/>
Tiempo:	<input type="text" value="--:-- ----"/>
Fecha y hora de nacimiento:	<input type="text" value="dd/mm/aaaa --:-- ----"/>
Mes:	<input type="text" value="----- de ----"/>
Semana:	<input type="text" value="Semana --, ----"/>
Número (min -10, max 10):	<input type="text" value="0"/>
Intervalo (min 0, max 10):	<input type="range" value="0"/>
Teléfono:	<input type="text"/>
Término de búsqueda:	<input type="text"/>
Color Favorito:	<input type="color" value="#000000"/>
<input type="submit" value="Enviar!"/>	

En cada uno de los capos podremos ver la funcionalidad que nos brinda.

Ejemplo básico de etiquetas semánticas haciendo uso de CSS:

archivo.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Titulo</title>
    <link rel="stylesheet" type="text/css" href="estilo.css">
  </head>
  <body>
    <div id="centrado">
      <header>
        <hgroup>
          <h1>(( TITULO ))</h1>
          <h2>(( DESCRIPCION DE LA WEB ))</h2>
        </hgroup>
        <div id="logotipo">(( LOGOTIPO ))</div>
      </header>
      <div id="contenido">
        <nav>
          (( MENU DE NAVEGACION ))
        </nav>
        <aside>
          ASIDE
        </aside>
        <article>

          <section id="izquierda">
            Sección de la izquierda
          </section>

          <section id="derecha">
            Sección de la derecha
          </section>
        </article>
      </div>
      <footer>
        <section id="copyright">(( Datos Copyright ))</section>
        <section id="contacto">(( Datos de Contacto ))</section>
        <section id="enlaces">(( MapaWeb, Política, etc. ))</section>
      </footer>
    </div>
  </body>
```

</html>

Archivo estilo.css

```
header{

    width: 100%;
    height: 150px;
    border-spacing: 5px;
    background-color: blue;
}

nav{
    width: 100%;
    height: 40px;
    background-color: yellow;
}

aside{
    width: 35%;
    height: 400px;
    float: right;
    display: block;
    background-color: green;
}

article{
    width: 65%;
    height: 400px;
    float: left;
    display: block;
    background-color: purple;
}

section#izquierda{
    position: relative;
    display: block;
    float: right;
    width: 46%;
    height: 320px;
    background-color: gray;
    margin-right: 2%;
    margin-left: 2%;
    margin-top: 5%;
}

section#derecha{
```

```

position: relative;
display: block;
float: left;
width: 46%;
height: 320px;
background-color: orange;
margin-left: 2%;
margin-left: 2%;
margin-top: 5%;
}

```

```

footer{
width: 100%;
height: 70px;
float: left;
display: block;
background-color: olive;
}

```

Resultado:

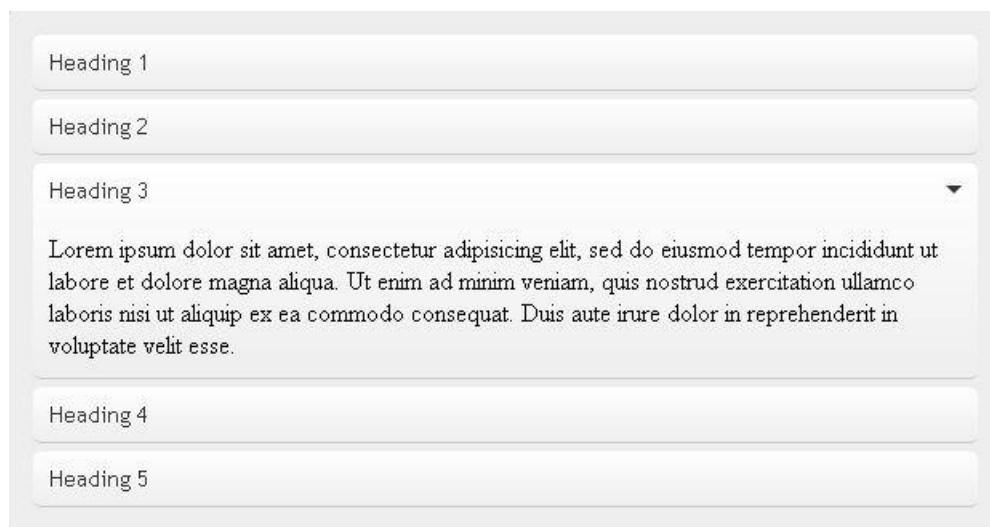


Crear un Acordeón con CSS3

En esta práctica aprenderás cómo crear un acordeón animado con instrucciones sencillas de CSS3.

Pero primero que todo, ¿qué es un acordeón en el mundo de la web? En términos simples, y al igual que en la vida real, es un elemento organizador que permite empaquetar muchos contenidos en un espacio reducido.

De esta forma, al finalizar esta práctica tendremos el siguiente resultado:



Para lograr este resultado debemos hacer el siguiente código html:

```
<!DOCTYPE html> <html>
<head>
  <title>Crear un acordeon con CSS3 </title>
  <link type="text/css" rel="stylesheet" href="estilo.css">
</head>
<body>
  <div class="accordion">
    <section id="one">
      <h2><a href="#one">Heading 1</a></h2>
      <div>

        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
        eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
        minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex
        ea commodo consequat.</p>
```

```

</div>

</section>
<section id="two">
  <h2><a href="#two">Heading 2</a></h2>
  <div>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
    ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo consequat.</p>
  </div>
</section>

<section id="three"> <h2><a href="#three">Heading 3</a></h2>
  <div>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
    ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo consequat.</p>
  </div>
</section>
<section id="four"> <h2><a href="#four">Heading 4</a></h2>
  <div>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
    ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo consequat. </p>
  </div>
</section>
<section id="five"> <h2><a href="#five">Heading 5</a></h2>
  <div>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
    ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo consequat. </p>
  </div>
</section>
</div>
</body>
</html>

```

Luego debemos crear un archivo css llamado estilo y darle la apariencia deseada de acordeón de la siguiente manera:

Estilo.css

```
#contenedor {
  width: 100%;
  height: 300px;
  border-spacing: 5px;
  background-color: blue;
}
#contenidos {
  display: block;
  background-color: pink;
}
#principal, #secundario, #lateral {
  display: block;
}
#principal {
  background-color: red;
  width: 60%;
  float: right;
}
#secundario, #lateral {
  width: 20%;
}

body {
  padding: 0;
  margin: 0;
}

section {
  display: block;
}

.accordion {
  background-color: #eee;
  border: 1px solid #ccc;
  width: 600px;
  padding: 10px;
  margin: 50px auto;
  /* Bordes redondeados */
  -moz-border-radius: 3px;
  -webkit-border-radius: 3px;
  border-radius: 3px;
  /* Sombra del contenedor */
  -moz-box-shadow: 0 1px 0 #999;
  -webkit-box-shadow: 0 1px 0 #999;
```



```

    box-shadow: 0 1px 0 #999;
}

.accordion section
{ border-bottom: 1px solid #ccc;
  margin: 5px;
  /* Fondo en degradado de colores */
  background-color: #fff;
  background-image: -webkit-gradient(linear, left top, left bottom, from(#fff),
to(#eee));
  background-image: -webkit-linear-gradient(top, #fff, #eee);
  background-image: -moz-linear-gradient(top, #fff, #eee);
  background-image: -ms-linear-gradient(top, #fff, #eee);
  background-image: -o-linear-gradient(top, #fff, #eee);
  background-image: linear-gradient(top, #fff, #eee);
  /* Bordes redondeados */
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  border-radius: 5px;
}

.accordion h2, .accordion p {
  margin: 0;
}

.accordion p {
  padding: 10px;
}

.accordion h2 a {
  display: block;
  position: relative;
  font: 14px/1 'Trebuchet MS', 'Lucida Sans';
  padding: 10px;
  color: #333;
  text-decoration: none;
  -moz-border-radius: 5px;
  -webkit-border-radius: 5px;
  border-radius: 5px;
}

.accordion h2 a:hover {
  background: #fff;
}

.accordion h2 + div {
  height: 0;
}

```

```
overflow: hidden;
/* Transiciones */
-moz-transition: height 0.3s ease-in-out;
-webkit-transition: height 0.3s ease-in-out;
-o-transition: height 0.3s ease-in-out;
transition: height 0.3s ease-in-out;
}

.accordion :target h2 a:after {
content: "";
position: absolute;
right: 10px;
top: 50%;
margin-top: -3px;
border-top: 5px solid #333;
border-left: 5px solid transparent;
border-right: 5px solid transparent;
}

.accordion :target h2 + div {
height: 100px;
}
```