

Etiquetas semánticas del HTML5

Qué son las etiquetas semánticas del HTML5, y por qué se recomienda usar aquellas que sirven para definir la estructura del documento en los sitios web modernos.

El propio lenguaje HTML es una de las muchas tecnologías que se ha renovado con la llegada de HTML5. Se han incorporado numerosas etiquetas que no estaban en versiones anteriores del lenguaje de marcación, que podemos clasificar en función de su utilidad o características. Una de estas clasificaciones son las etiquetas semánticas, que ya comenzamos a explicar en el artículo anterior, y que trata sobre las etiquetas nuevas del HTML5.

En el presente artículo 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 cómo 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 este artículo, 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.

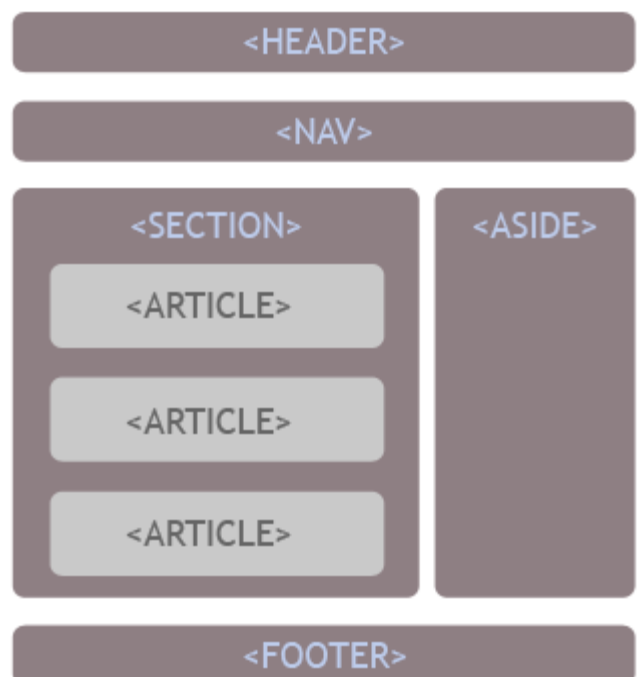
Existirán además otras etiquetas que representan elementos típicos como la barra de navegación, etiqueta NAV.

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.

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 utilizemos 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.

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.

En el próximo artículo explicaremos algunos detalles sobre la compatibilidad de las etiquetas semánticas con los navegadores que se usan en la actualidad y cómo resolver casos indeseables que ocurren cuando los navegadores no las interpretan correctamente.

Tipos de enlaces en HTML

Introducción

Lo que caracteriza a la web como medio de comunicación es la navegación por medio de hipervínculos (links o enlaces). Esta conexión entre archivos permite al usuario elegir el sentido de la navegación.

En el HTML los enlaces se marcan con la etiqueta `<a>` y el atributo principal es **href=""** donde se escribe la ubicación del archivo de destino que puede estar en la misma carpeta que el archivo que lo está llamando, en otra carpeta del mismo sitio o en otro sitio web.

Entre las etiquetas `` se puede colocar cualquier elemento html que funcionará como botón, generalmente se coloca un texto o una imagen. Si es un texto, el navegador por defecto lo muestra en color azul y subrayado. Si es una imagen, algunos navegadores especialmente las versiones anteriores a las actuales, le pueden agregar un borde azul.

Rutas de enlaces:

Se llama ruta al valor que se escribe dentro del atributo **href=""** y se pueden clasificar en dos tipos:

Rutas absolutas:

Las **rutas absolutas** son las que definen la ubicación completa de un archivo en la web y empiezan desde **https://nombredelsitio.com/archivo.html** (el archivo de destino puede ser de cualquier extensión). Por ejemplo si queremos vincular nuestro sitio con Facebook, usaríamos una ruta absoluta y el código sería:

```
<a href="https://www.facebook.com/pagina">Estamos en Facebook</a>
```

Rutas relativas:

Las **rutas relativas** en cambio, definen la ubicación del archivo de destino en relación a la ubicación del archivo que lo está llamando. Ejemplo: si el archivo **index.html** vincula con **quien-soy.html** y están en la misma carpeta, el código sería así:

```
<a href="quien-soy.html">Quién soy</a>.
```

Tipos de enlaces

Existen diferentes tipos de enlaces que detallo a continuación:

Enlaces internos entre archivos .html

Son los enlaces principales de un sitio. Si en el archivo **index.html**, queremos agregar un vínculo a **contacto.html** el código sería así: `Contáctenos`. El texto que queda encerrado entre las etiquetas **<a>** es lo que ve el usuario en color azul y subrayado.

Enlaces externos

Son los enlaces entre diferentes sitios web. Si queremos vincular un sitio web con otro, tenemos que indicar la **ruta absoluta** del sitio escribiendo la **url completa** que conviene copiarla del navegador. En este caso, es común usar el atributo **target=""** que nos permite abrir el enlace en una pestaña nueva. Por ejemplo, si quisiera poner un vínculo a Google y que éste abra en pestaña nueva, sería así: `Ir a Google`

Enlaces internos: anclas

Las anclas son enlaces hacia un punto determinado dentro de un html. Se usa en varios casos, por ejemplo en una página de preguntas frecuentes donde las respuestas están desarrolladas debajo de las preguntas, o en Wikipedia, cuando el texto es extenso el menú tiene vínculos con anclas. También en los textos largos, al finalizar muchas veces se coloca un botón para subir. En los sitios de una sola página donde los botones en realidad hacen scroll, esas son anclas.

El punto de destino tiene que estar marcado con el atributo **id="algo"** y en el enlace se coloca un **#** (numeral) seguido del nombre. Por ejemplo: `Pregunta 4` "salta" a este párrafo: `<p id="respuesta4">acá está escrita la respuesta</p>`

Enlaces para ampliar una imagen

Es común armar una galería de fotos con imágenes miniaturas que vinculan a imágenes ampliadas. Si combinamos este recurso con código javascript, logramos galerías de fotos muy atractivas. El código HTML es así: `` El navegador le muestra al usuario la foto miniatura y si cliqueea en ella, le muestra la foto ampliada.

Enlaces para descargar archivos

En realidad, dentro del atributo **href=""** podemos colocar la ruta hacia cualquier tipo de archivo. Si el navegador reconoce la extensión, lo abre por ejemplo: html, jpg, png, gif, svg, pdf, etc. Pero si no lo reconoce o es un archivo comprimido (.rar, .zip), el navegador le ofrece al usuario descargarlo. Ejemplo:

```
<a href="fotos.rar">Descargá todas las fotos</a>
```

Enlaces a una dirección de correo

Se puede vincular una dirección de correo para que abra en el programa de correo predeterminado. El código se escribe así:

```
<a href="mailto:info@dominio.com">Consúltenos a info@dominio.com</a>
```

Este recurso está bastante cuestionado por dos razones: primero porque muchos usuarios utilizan correo webmail (Gmail, Yahoo, Hotmail) por lo que este enlace no les sirve (incluso molesta) y segundo porque les estamos facilitando a los robots que encuentren nuestro correo y lo incluyan en una base de datos para spam.

Listas no ordenadas:

```
<ul>
```

Las listas no ordenadas van dentro de la etiqueta

```
<ul>
```

 HTML y de su cierre

```
</ul>
```

. Cada punto que queramos añadir a la lista, lo haremos dentro de la etiqueta

```
<li>
```

 y su cierre.

Si no le indicamos nada a la etiqueta

```
<li>
```

 HTML, ésta se generará de forma automática. Pero si queremos definir nosotros mismo el símbolo del punto, podemos gracias al atributo `type`.

Así podemos hacer que la lista esté definida por puntos negros (`li type="disc"`), por puntos con el fondo blanco (`li type="circle"`) o por cuadrados (`li type="square"`). Aunque esta apariencia dependerá del

navegador. Hay algunos navegadores que dan otras apariencias a estos mismos atributos.

Vamos con un pequeño ejemplo en código:

```
<ul>

<li type="circle">Esto es un tipo de punto.</li>

<li type="square">Este es otro.</li>

<li type="disc">Y este es otro diferente.</li>

</ul>
```

Cuyo resultado visual será el que veremos a continuación:

- Esto es un tipo de punto.

- Este es otro.

- Y este es otro diferente.

Listas ordenadas:

Las listas ordenadas van enmarcadas dentro de las etiquetas .

Cada punto de la lista se escribe con la misma etiqueta que en las no numeradas: . Pero al ser listas ordenadas los símbolos serán números y éstos se irán generando automáticamente por orden, conforme escribamos nuevos puntos.

En las listas ordenadas podemos hacer que el primer punto comience con el número que nosotros queramos. Lo conseguiremos gracias al atributo

“value”. Los siguientes puntos que escribamos se generarán automáticamente por orden, partiendo de ese número.

Por ejemplo, si queremos que nuestra lista empiece por el número 20, sólo deberemos escribir lo siguiente:

```
<ol>  
  
<li value="20">Este será el número 20. </li>  
  
<li>Este será el 21. </li>  
  
<li> Este será el 22. Y así sucesivamente. </li>  
  
</ol>
```

Y el resultado será el siguiente:

20. Este será el número 20.

21. Este será el 21.

22. Este será el 22. Y así sucesivamente.

Listas de definiciones: <dl>, <dt> y <dd>

Si lo que vamos a hacer es un listado de definiciones, podemos usar las etiquetas <dl>, <dt> y <dd>. Vamos a explicarlas por partes:

La etiqueta <dl> viene de los términos ingleses "Definiton list" y nos indica que dentro de ella, entre ella y su cierre, va a ir una definición.

La etiqueta <dt> viene de los términos "Definition term" y dentro de ella irá el término que vamos a definir. Para entendernos mejor, dentro de <dt> iría el título de la definición.

La etiqueta <dd> viene de los términos "Definition description" y nos dice que dentro de ésta irá la definición.

Si escribimos varios listados de definición, éstas se separarán automáticamente entre ellas para facilitar su diferenciación.

Aquí podemos ver un ejemplo de código de dos listado de definición:

```
<dl>
```

```
<dt>Aquí va el término que definiremos</dt>
```

```
<dd>Y aquí dentro irá la definición propiamente dicha.</dd>
```

```
</dl>
```

```
<dl>
```

Anon. n.d. "Tipos de Enlaces En HTML." Retrieved April 8, 2021c
(<https://www.laurachuburu.com.ar/tutoriales/enlaces-html.php>).