

1. APLICAR ESTILO A LOS ELEMENTOS ESTRUCTURALES HTML5

La forma de aplicar estilos a los elementos estructurales que introduce HTML5 es exactamente la misma que con los elementos HTML tradicionales.

Así pues, hemos visto que podemos utilizarlos como selectores de las reglas CSS o aplicar estilos mediante identificadores y clases.

Sin embargo, los navegadores que no entienden estos elementos HTML5 no los renderizarán correctamente, ya que los tratan como "elementos desconocidos".

El principal problema de esto es que, por defecto, los tratan como elementos "en línea", por lo que si utilizamos los elementos **header**, **article**, **aside**, **footer**, etc. para estructurar la página, no conseguiremos el resultado deseado.

En la figura siguiente puedes ver la página con la que trabajamos en la lección anterior, pero ahora visualizada con el navegador Internet Explorer 7. Como puedes ver, el resultado dista mucho del que renderiza un navegador moderno.



Hay una forma sencilla de solucionar este problema y es aplicando una regla CSS de carácter general que establezca la propiedad **display: block;** para estos elementos HTML5:

```
header, nav, aside, article, section, footer {  
    display: block;  
}
```

De esta manera, aunque los navegadores antiguos siguen sin entender estos elementos, al menos los tratarán como elementos de bloque, de forma similar a un elemento **div**.

Adicionalmente, para **Internet Explorer 8** y anteriores, es necesario escribir un pequeño fragmento de JavaScript, ya que este navegador no incluye los elementos desconocidos en el DOM del documento, como sí que hacen el resto. Debemos escribir este código JavaScript en la sección **head** de la página:

```
<script type="text/javascript">  
document.createElement("header");  
document.createElement("nav");  
document.createElement("aside");  
document.createElement("article");  
document.createElement("section");  
document.createElement("footer");  
</script>
```

Fíjate que lo único que hace este código es añadir esos elementos en el DOM del documento, ya que **Internet Explorer 8** y anteriores no lo hacen. Este código no afecta en nada al resto de navegadores, entiendan o no los elementos HTML5.

Es importante, sin embargo, recordar que los navegadores antiguos no son capaces de entender el significado de estas etiquetas HTML5, sino que simplemente las tratan como otros elementos de bloque más, de forma similar a cómo lo harían con los elementos **div**.

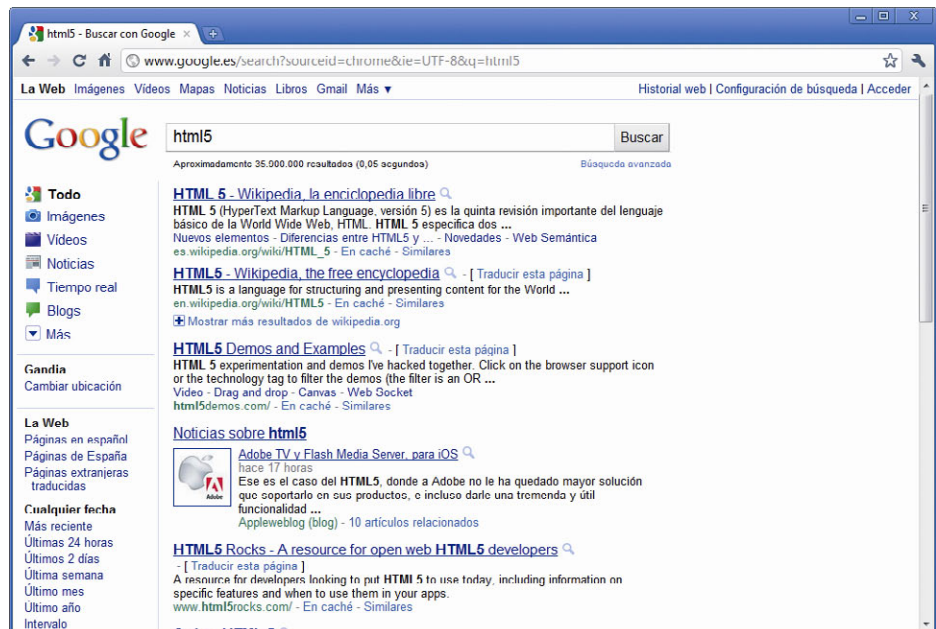
2. OTROS ELEMENTOS SEMÁNTICOS DE HTML5

HTML5 introduce otros elementos semánticos para describir mejor el contenido de las páginas web. En este caso se trata de elementos "en línea", como:

- El elemento **<mark>** se utiliza para marcar o resaltar texto con el propósito de hacer referencia al mismo, debido a su relevancia en otro contexto.

El ejemplo más claro del uso de **<mark>** lo encontramos cuando estudiamos la página de resultados de una búsqueda. Habitualmente el término o términos que se han utilizado para la búsqueda quedan resaltados.

En ese caso, el uso de **<mark>** sería el más adecuado, ya que no queremos dar mayor o menor importancia a esos términos (para lo que podríamos utilizar **** o ****) sino reflejar el hecho de que son los términos buscados (en la figura siguiente puedes verlo, fíjate cómo el término *html5* aparece resaltado en las páginas del resultado de la búsqueda).



- El elemento **<progress>** representa el estado de completitud o progreso de una tarea.





El progreso puede ser indeterminado, indicando que no se conoce cuándo finalizará la tarea; o un valor numérico indicando el trabajo completado.

Este elemento dispone de dos atributos que establecen su significado:

- **value**: especifica la cantidad de trabajo completado.
- **max**: especifica la cantidad de trabajo necesaria para completar la tarea.

Puede utilizarse, por ejemplo, para especificar el progreso de una copia o descarga de archivos, como puedes ver en la figura de la página siguiente.

<progress>

- Indeterminado 
- Progreso: 0% 
- Progreso: 100% 
- Progreso 50% 

Este es el código correspondiente:

```
<ul>
  <li>
    <label>Indeterminado</label>
    <progress max="100"></progress>
  </li>
  <li>
    <label>Progreso: 0%</label>
    <progress max="100" value="0"></progress>
  </li>
  <li>
    <label>Progreso: 100%</label>
    <progress max="100" value="100"></progress>
  </li>
  <li>
    <label>Progreso 50%</label>
    <progress max="1.0" value="0.5"></progress>
  </li>
</ul>
```

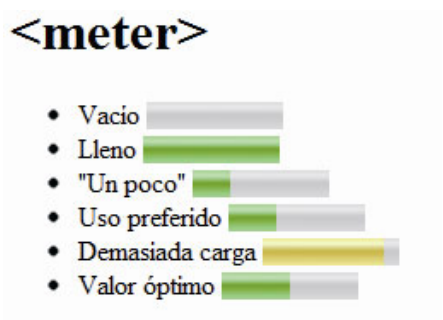
- El elemento **<meter>** sirve para representar una medida dentro de un rango conocido.

Por ejemplo, puede utilizarse para medir el uso del espacio de un disco, representando la parte ocupada y libre del mismo.

Este elemento admite hasta seis atributos que establecen su significado:

- **value**: representa el valor actual, que se entiende teniendo en cuenta el resto de atributos.
- **min**: representa el valor menor posible.
- **max**: representa el valor mayor posible.
- **low**: representa el rango que es considerado como "bajo".
- **high**: representa el rango que es considerado como "alto".
- **optimum**: representa el valor que es considerado como "óptimo".

Además, si el valor "optimum" es más alto que el valor "high", entonces se entiende que cuanto más alto, mejor; mientras que si el valor "optimum" es más bajo que el valor "low", entonces se entiende que cuanto más bajo, mejor.



Y este es el código correspondiente a la imagen anterior:

```
<ul>
  <li>
    <label>Vacío</label>
    <meter value="0"></meter>
  </li>
  <li>
    <label>Lleno</label>
    <meter value="1"></meter>
  </li>
  <li>
    <label>"Un poco"</label>
    <meter min="0.34" max="0.41" value="0.36"></meter>
  </li>
  <li>
    <label>Uso preferido</label>
    <meter min="50" max="250" low="100" high="200"
      value="120"></meter>
  </li>
  <li>
    <label>Demasiada carga</label>
    <meter min="1024" max="10240" low="2048"
      high="8192" value="9216"></meter>
  </li>
  <li>
    <label>Valor óptimo</label>
    <meter value="0.5" optimum="0.8"></meter>
  </li>
</ul>
```

- El elemento <time> se utiliza para representar una determinada hora en formato de 24 horas y/o una fecha concreta.

Por ejemplo, vamos a utilizarlo para la fecha de publicación de los post de este blog.

Hasta ahora esta fecha se había incluido simplemente como un texto más, por lo que el navegador no puede identificarla como tal:

```
<p>Publicado por <a href="#"
class="autor">Administrador</a> - 14 de julio</p>
```

Para representar únicamente una fecha, se debe utilizar este formato: **AAAA-MM-DD**

Es decir, primero el año con cuatro dígitos, después el mes con dos dígitos y, finalmente, el día también con dos dígitos. Todo ello separado por guiones.

Para representar una hora en concreto, utiliza la notación de 24 horas: **HH:MM**

Es decir, dos dígitos para la hora, dos puntos y dos dígitos para los minutos. Opcionalmente, se pueden añadir los segundos con el formato **HH:MM:SS** e incluso centésimas de segundo con el formato **HH:MM:SS.mmm**

Si deseas proporcionar la fecha y la hora conjuntamente, tendrás que separar ambos datos con un **T** y añadir al final la zona horaria.

Por ejemplo, la fecha del 14 de julio de 2011, a las 15:59:14 de España se indicaría así:

2011-07-14T15:59:14+02:00

donde +02:00 es la zona horaria de España en verano.

```
<p>Publicado por <a href="#"
class="autor">Administrador</a> -
<time datetime="2011-07-14T15:59:14+02:00" pubdate>14 de
julio</time></p>
```

Se utiliza la etiqueta **<time>** y con el atributo **datetime** se especifica el valor de hora y/o fecha.

Entre la etiqueta de apertura y cierre se incluye la hora que realmente se visualizará en la página web.

Ten en cuenta que si no incluyes el atributo **datetime**, entonces deberás incluir su valor entre las etiqueta de apertura y cierre, por lo que el valor *14 de julio* no tendría el formato correcto.

Finalmente, la etiqueta **<time>** dispone del atributo opcional **pubdate**. Si lo incluimos, estaremos indicando que se trata de la fecha de publicación del artículo o de todo el documento, si no aparece dentro de una etiqueta **<article>**.



Solo puede haber una etiqueta **<time>** con el atributo **pubdate** por cada **<article>** y una sola para el documento general.

Como has podido comprobar, en HTML5 es muy importante proporcionar elementos semánticos que pueda utilizar el diseñador para describir lo mejor posible el contenido de las páginas web.

Claro está que es precisamente el diseñador el que decidirá hacerlo o no. Por ejemplo, la página actual seguiría siendo igualmente válida según la especificación de HTML5 con y sin el uso de **<time>**.

La especificación de HTML5 **aconseja** representar las fechas de esta forma, pero no lo establece como obligatorio.

3. ELEMENTOS ARTICLE ANIDADADOS

Continuamos con otro tema interesante, referente a la posibilidad de incluir elementos estructurales en el interior de otros elementos estructurales.

Y vamos a verlo con un ejemplo al que estarás habituado. Seguro que has podido comprobar que en muchos blogs se permite realizar comentarios.

Esos comentarios suelen aparecer debajo del artículo o post relacionado. Pues bien, una forma de representar esto es incluir uno o más elementos **article** (los comentarios) dentro del elemento **article** correspondiente al post.

Aquí la especificación de HTML5 lo deja bastante claro: *"Cuando los elementos article aparecen anidados, el elemento interior representa contenido relacionado con el contenido del artículo exterior."*

Este es el caso de un post y sus comentarios. Veamos cómo hacerlo en la página correspondiente al primer post del blog.

En este caso tenemos la misma distribución que la página inicial, pero el contenido del artículo está completo y en lugar de aparecer los enlaces **"Leer más"** y **"Comentarios"** en el pie del artículo, aparece la lista de comentarios realizados en el post.

Puedes verlo en la figura de la página siguiente.

La estructura que se ha seguido es la de mantener los comentarios en el pie o elemento **footer del artículo**. Dentro de dicho elemento **footer** aparecerá cada comentario como un elemento **article** anidado.

Voz, entonación, mirada y presencia

Publicado por Administrador - 14 de julio

Existe una serie de variables personales que influirán decisivamente en el desarrollo de una presentación y que conviene tenerlas en cuenta para adecuarlas convenientemente a cada situación.



Así pues, la **voz** y **entonación** hacen referencia a cómo pronunciar la presentación, es decir, el volumen y tono adecuado para ser oído claramente.

En la vida cotidiana ocurre que, incluso las personas que en una conversación normal se expresan de forma enérgica y clara y anticipan bien las palabras, cuando se encuentran frente a una audiencia formal pueden hablar de forma apagada, inaudible, monótona, etc.

La voz debe servir como vehículo para comunicar un mensaje sereno, interesante, racional y serio. Adoptar un estilo serio, un tono de voz interesante.

Puede ser útil recordar las siguientes siglas mnemotécnicas:

PAPPERR: Proyección, articulación, pronunciación, enunciación, repetición y rapidez.

Comentarios

Comentario de Juan - 14 de julio
Gracias por el artículo, ha sido interesante.

Comentario de Administrador - 14 de julio
Gracias a ti por seguir tan fielmente el blog.

Un saludo.

A continuación se muestra el código correspondiente a la sección de comentarios:

```
<article>
...
<footer>
  <section id="comentarios_post">
    <h1>Comentarios</h1>
    <article> <!-- comentario 1 -->
      <header>
```



```

        <strong>Comentario de Juan - <time datetime="2011-
07-14T17:12:58+02:00">14 de julio</time></strong>
    </header>
    <p>Gracias por el artículo, ha sido interesante.</p>
</article>
<article> <!-- comentario 2 -->
    <header>
        <strong>Comentario de Administrador - <time
datetime="2011-07-14T17:20:12+02:00">14 de julio</time>
    </strong>
    </header>
    <p>Gracias a ti por seguir tan fielmente el
blog.</p>
    <p>Un saludo.</p>
    </article>
</section>
</footer>
</article>

```

Así pues, vemos que dentro del elemento **footer** se ha insertado un elemento **section** para representar los comentarios del artículo.

Parece bastante adecuado hacerlo de esta forma, ya que la sección empieza con un título **h1** y se corresponde con contenido relacionado con el artículo.

Además, el identificador **comentarios_post** nos puede servir como marcador para el hipervínculo que aparece en la página principal del blog y que apunta a los comentarios de cada post.

Seguidamente, encontramos el primer comentario. Para ello, se utiliza un elemento **article**.

Es importante recordar que estamos en el interior del elemento **article** principal, el correspondiente al post. Por lo tanto, este otro elemento **article** es un elemento anidado.

Lo mismo podemos decir respecto al otro elemento **article** representando el segundo comentario.

Así pues, tenemos una distribución con sentido para los comentarios:

- Rodeando a todos los comentarios tenemos un elemento **section** y
- cada comentario se incluye en su propio elemento **article**.

Esta estructura podría complicarse todavía más. Por ejemplo, si queremos anidar comentarios, es decir, representar un comentario que responde a otro anterior, podríamos hacerlo repitiendo esta distribución, pero ahora el artículo "*principal*" sería el comentario inicial.

4. EL ESQUEMA DEL DOCUMENTO

Finalizamos la lección estudiando el concepto de "*esquema del documento*".

Hasta la aparición de HTML5, la única forma de estructurar un documento en títulos y párrafos era a través de las etiquetas **h1**, **h2**, etc. y **p**.

El diseñador debía emplear estas etiquetas de una forma lógica, utilizando las etiquetas de título mayor (por ejemplo, **h1**) para remarcar un punto o sección de mayor importancia que otras en las que utilizaba títulos menores (por ejemplo, **h2**).

De esta forma, no tenía sentido disponer de una etiqueta **h1** posterior a otra **h2** en el mismo bloque de la página.

Pues bien, aunque el uso de las etiquetas de título **h1...h6** no ha variado prácticamente, HTML5 ha introducido elementos que determinan secciones del documento en las que se crea un ámbito independiente del resto de secciones o del documento en sí.

¿Qué significa esto? Pues que el uso de las secciones en HTML5 permite que el diseñador pueda utilizar las etiquetas de título sin importar el orden del último título del nivel superior empleado.

Los elementos HTML5 que inician una nueva sección en el esquema del documento son: **article**, **section**, **nav** y **aside**.

Esto lo hemos podido ver anteriormente, al comprobar cómo insertar los comentarios de un post.

Si te fijas en la línea siguiente tenemos una etiqueta de título **h1** en el interior del elemento **section**:

```
<section id="comentarios_post">  
  <h1>Comentarios</h1>
```

Esto no parece que sea ningún problema con lo que conocemos hasta ahora, pero sí que lo sería en HTML 4.01.

¿Por qué? Pues porque aparece después de una etiqueta de título **h2** (para el subtítulo del blog) o **h3** (para el título del post).

Si pensamos en HTML 4.01, el uso de la etiqueta **h1** en la sección de comentarios es incorrecta ya que representa una parte menos importante (desde el punto de vista del esquema del documento) que el subtítulo del blog.

Sin embargo, gracias al ámbito de sección que introduce algunos elementos HTML5, como **article**, **section**, **nav** y **aside**, esto es posible.



Así pues, un título **h1** que está dentro de un elemento **section** puede tener realmente un significado equivalente a **h4** en el esquema del documento (como ocurre en este caso).

Sin entrar en mayores detalles referente a este complejo asunto, sí que interesa conocer que esto puede tener algún efecto en la forma de aplicar estilos mediante CSS.

Así pues, aunque el título **h1** incluido en la sección de comentarios tiene un significado equivalente a un título **h4** en el esquema del documento, sigue siendo la etiqueta **h1** para la hoja de estilo en cascada, por lo que se utilizará el formato asignado a dicha etiqueta. Esto no parece adecuado en este caso.

La forma de corregir este problema de formato será utilizar un selector más específico para que solo afecte al uso de la etiqueta **h1** dentro de la sección de comentarios:

```
#comentarios_post h1 {
    font-size: 1.5em;
}
```

Lo mismo podríamos decir respecto de los detalles del aspecto que tienen los bloques de comentarios. Como se aplica el formato de los elementos **article**, tenemos el mismo aspecto para un comentario que para el post en sí. Podemos cambiar también esta circunstancia con un selector más específico:

```
body > article {
    border-bottom-style: dotted;
    border-bottom-width: 3px;
    border-bottom-color: black;
    margin-bottom: 20px;
    clear: both;
    overflow: hidden;
}

article article {
    font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
    font-size: 11px;
    border-bottom-style: solid;
    border-bottom-width: 1px;
    padding-top: 10px;
}
```

En la primera regla el selector especifica los elementos **article** "hijos" de **body**, por lo que el estilo afectará únicamente a los post del blog.

Por su parte, la segunda regla sirve para aplicar un formato específico a los elementos **article** correspondiente a los comentarios. En este caso el selector especifica los elementos **article** "descendientes" de otros elementos **article**.

Fíjate que no se ha utilizado el selector de "hijos" en la segunda regla, aunque también habría sido válido en este ejemplo, porque de esta forma serviría si tuviéramos comentarios dentro de otros comentarios.

Las primeras etiquetas HTML5 que hemos conocido tienen el propósito de crear una Web mejor estructurada y, sobre todo, más semántica, con más significado.

El diseñador podría obviar el uso de estas etiquetas, diseñando las páginas web como lo ha hecho hasta ahora, y seguiría siendo código válido. Sin embargo, utilizando los elementos estructurales HTML5 añadirá valor a sus diseños.



La etiqueta <hgroup>

Se trata de otra etiqueta HTML5 que podemos utilizar desde el punto de vista semántico.

Se emplea para representar "*grupos de encabezados*". Por ejemplo, en el blog que hemos diseñado tenemos que el título principal del blog es "**Cómo preparar presentaciones eficaces**" y, debajo de él, aparece el subtítulo "**Un blog para directivos**". Para ello se han utilizado las etiquetas **h1** y **h2**, respectivamente.

La pregunta que se plantea ahora es: ¿tiene sentido que aparezca el subtítulo en el esquema del documento?

Pues realmente no, ya que esquematizando el contenido de la página web no se ve claro en qué nivel debería aparecer. Por ejemplo, ¿ese subtítulo tiene más importancia que los títulos de los distintos post?

Para eliminar el subtítulo del esquema del documento, podemos utilizar la etiqueta **<hgroup>**.

Así pues, con este código:

```
<hgroup>
  <h1>Cómo preparar presentaciones eficaces</h1>
  <h2>Un blog para directivos</h2>
</hgroup>
```

solo aparece en el esquema del documento el título **h1** (el de mayor nivel, independientemente del orden en que aparezca).

Es importante no confundir **<hgroup>** con **<header>**. La etiqueta **<hgroup>** sirve para evitar la circunstancia que acabamos de relatar, mientras que **<header>** tiene mucho más significado al marcar una sección de encabezado o introducción.

Así pues, dentro de **<hgroup>** solo podremos incluir etiqueta de título **h1...h6**, mientras que el contenido de **<header>** puede incluir cualquier elemento que sirva para introducir la sección.