Capítulo 3: Los elementos de la estructura en HTML5

A. Los elementos de la estructura en HTML4 62
B. Los elementos de la estructura en HTML5 64
C. El atributo semántico "role" 67
D. Ejemplos de estructura en HTML5 69
E. Las técnicas de formato 71

A. Los elementos de la estructura en HTML4

1. Las cajas

En HTML4, el elemento principal para estructurar las páginas web es la famosa "caja ". Con el elemento `<div>`, es posible crear zonas de visualización de forma rectangular. Cada una de esas zonas, de esas cajas `<div>`, puede ser identificada de manera exclusiva, mediante un código de identificación, y de ese modo, podemos aplicarle un formato con CSS. Ese código de identificación único se establece con el atributo `id`.

Ejemplo:

```
<div id="bannerSuperior">
 ...
</div>
```

También podemos usar las clases, cuando este formato se repita en la página y no se trate, por lo tanto, de un formato único.

Ejemplo:

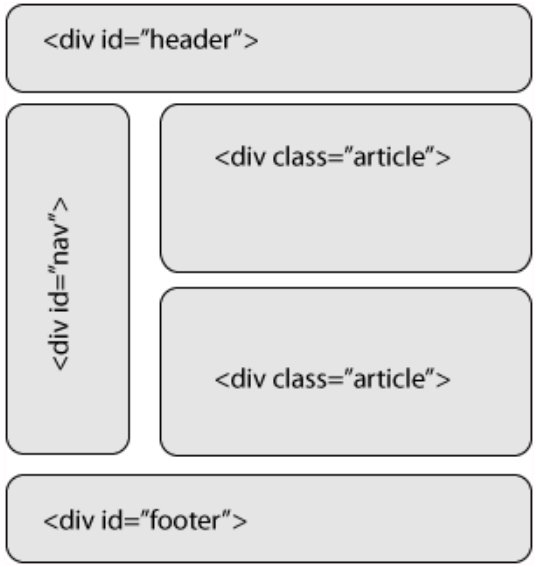
```
<div class="comentarios">
 ...
</div>
```

A continuación, una vez que haya determinado la estructura, podrá aplicar el diseño con las hojas de estilo CSS, mediante los selectores de identificación que correspondan al código de identificación o a la clase en cuestión.

Ejemplo:

```
#bannerSuperior {
 ...
}
.comentarios{
 ...
}
```

En el siguiente ejemplo, podemos ver la estructura de una página de tipo blog con cajas `<div>`:



2. La "divitis"

Con HTML4 y CSS 2.1, la estructura de las páginas, cuando esta es relativamente compleja, puede requerir una gran cantidad de cajas `<div>`. Podemos llegar a crear páginas que contengan decenas y decenas de `<div>`. Si usted cae en ese exceso, habrá contraído la enfermedad llamada "divitis", es decir, el uso excesivo de cajas `<div>`.

3. Un contenido no semántico

Todas esas cajas `<div>` presentan otro problema: el de la semántica de los contenedores. Cada caja `<div>` se distingue de las demás por su código de identificación único, que el diseñador web le habrá asignado en función de su "humor" o de sus "ganas" en ese momento.

Por ese motivo, las cajas `<div>` no son semánticas: el contenido esperado no está definido por ningún parámetro. Una caja `<div>` identificada con `id="izquierda"` no nos aporta ningún dato sobre su contenido. Podría tratarse de una barra de navegación, de información legal o de cualquier otro tipo de contenido.

Ahora bien, la evolución del HTML se dirige hacia un mayor uso de la semántica.

© Editions ENI - All rights reserved

B. Los elementos de la estructura en HTML5

1. Los nuevos elementos de la estructura

Con el HTML5 llegan nuevos elementos de estructura semántica. Estos nuevos elementos han sido definidos y se les ha asignado un nombre tras un largo análisis de los nombres más usados en las cajas `<div>`.

2. El elemento <header>

El nuevo elemento `<header>` permite insertar una zona de visualización para las cabeceras. Puede definirse esta cabecera para toda la página o para una zona determinada: artículo, menú lateral...

Debemos considerar que este elemento se puede usar desde dos puntos de vista:

- a nivel de la página: es la típica cabecera de la página, que a menudo aparece en lo alto de la pantalla, con un logotipo, un eslogan, un menú de navegación principal...
- a nivel del contenido: permite disponer de una zona de introducción del contenido que se incluya a continuación.

Esta es la definición del W3C para el elemento `<header>`: "*The header element represents a group of introductory or navigational aids. A header element is intended to usually contain the section's heading (an h1–h6 element or an hgroup element), but this is not required. The header element can also be used to wrap a section's table of contents, a search form, or any relevant logos.*"

Como puede ver, el elemento `<header>` no tiene que aparecer obligatoriamente en la estructura de la página.

3. El elemento <footer>

El nuevo elemento `<footer>` permite insertar una zona de visualización para los pies de página. Volvemos a encontrarnos con la misma semántica de los encabezados. Es posible definir un pie de página para la página completa, o bien para cualquier otra zona de visualización o para un artículo.

Esta es la definición del W3C: "*The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like*."

Su uso es similar al de los `<header>`, pero aplicado a los pies de página. Por lo tanto, no debemos seguir al "pie de la letra" la traducción literal de "pie de página". Se trata más bien de un "pie de zona de visualización"; y esta zona de visualización puede ser una página, una sección, un artículo...

4. El elemento <nav>

El elemento `<nav>`, como su nombre indica, está pensado para la visualización de una zona de navegación con vínculos hipertexto. Cuidado, esto no quiere decir que solo podrá tener una zona de navegación en cada página, o que tenga que crear tantos elementos `<nav>` como zonas de navegación haya en sus páginas; bastará con que los identifique correctamente. El elemento `<nav>` está pensado en especial para la navegación principal del sitio web, para la inserción de vínculos entre las páginas de dicho sitio web.

La definición del W3C es bastante clara: "*The nav element represents a section of a page that links to other pages or to parts within the page: a section with navigation links. Note: Not all groups of links on a page need to be in a nav element – the element is primarily intended for sections that consist of major navigation blocks. In particular, it is common for footers to have a short list of links to various pages of a site, such as the terms of service, the home page, and a copyright page. The footer element alone is sufficient for such cases; while a nav element can be used in such cases, it is usually unnecessary*."

5. El elemento <section>

El elemento `<section>` permite agrupar elementos que tengan la misma temática. De este modo, podemos agrupar en un mismo elemento un contenido, con su título y su pie de página.

Esta es la definición del W3C: "*The section element represents a generic section of a document or application. A section, in this context, is a thematic grouping of content, typically with a heading*."

6. El elemento <article>

El elemento `<article>` permite insertar un contenido autónomo, que puede volver a usarse en otro lugar del sitio web, sin que por ello se anule su significado.

Esta es la definición del W3C: "*The article element represents a self-contained composition in a document, page, application, or site and that is, in principle, independently distributable or reusable, e.g. in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content*."

Claro está, un artículo podría tener una cabecera (`<header>`), un pie de página (`<footer>`) y varios títulos (`<hx>`).

7. El elemento <aside>

El elemento `<aside>` permite mostrar un contenido relacionado con el contenido al cual esté vinculado. Puede tratarse de barras laterales (*sidebars*), de zonas de widgets, de complementos sobre artículos, etc.

© Editions ENI - All rights reserved

8. El elemento <main>

Último en llegar a los borradores del W3C, el elemento `<main>` permite representar el contenido principal de la página.

La definición del W3C lo indica bien: *"The main element represents the main content of the body of a document or application. The main content area consists of content that is directly related to or expands upon the central topic of a document or central functionality of an application"*.

De esta forma, una página podrá estar perfectamente estructurada con un elemento `<main>` para el contenido de la página, un encabezado `<header>`, contenidos con elementos `<article>` y un pie de página `<footer>`.

El W3C indica una serie de restricciones para el uso del elemento `<main>`:

- un solo elemento `<main>` por página,
- no debe utilizarse dentro (elemento descendente) de los elementos `<article>`, `<aside>`, `<footer>`, `<header>` o `<nav>`,
- debería estar asociado con el `role main (<main role="main">)` para una mejor accesibilidad de los sitios web.

Observe que, con HTML5.2, podemos utilizar varios elementos `<main>` en la misma página. Es su responsabilidad diferenciarlos con selectores CSS.