

# CAPÍTULO 2 SEMÁNTICA

Una de las novedades que hemos mencionado anteriormente son las etiquetas que se han introducido en HTML5. Existen más de 30 nuevas etiquetas semánticas que pueden ser utilizadas en nuestras páginas estáticas. Estas nuevas etiquetas se podrían clasificar en dos grupos:

- Etiquetas que extienden a las actuales, como `<video>`, `<audio>` o `<canvas>`, y que además añaden nuevas funcionalidades a los documentos HTML, que podemos controlar desde JavaScript y
- etiquetas que componen la web semántica, es decir, que no proponen nuevas funcionalidades pero sirven para estructurar sitios web, y añadir un significado concreto, más allá de las etiquetas generales como `<div>`.

En este capítulo, veremos como transformar nuestra estructura actual de marcado basada en `<div>`, a una estructura que utiliza las nuevas etiquetas estructurales como `<nav>`, `<header>`, `<footer>`, `<aside>`, o `<article>`.

## 2.1 CABECERA DEL DOCUMENTO

Además de las nuevas etiquetas introducidas por HTML5 (que veremos más adelante), el nuevo estándar propone pequeñas mejoras que podemos aplicar en la definición de nuestros documentos, en concreto en la cabecera de los mismos.

### 2.1.1 DOCTYPE

El estándar XHTML deriva de XML, por lo que comparte con él muchas de sus normas y sintaxis. Uno de los conceptos fundamentales de XML es la utilización del DTD o Document Type Definition ("Definición del Tipo de Documento"). El estándar XHTML define el DTD que deben seguir las páginas y documentos XHTML. En este documento se definen las etiquetas que se pueden utilizar, los atributos de cada etiqueta y el tipo de valores

## ÍNDICE DE CONTENIDOS

### Semántica

#### 2.1 Cabecera del documento

#### 2.2 Nuevas etiquetas semánticas

#### 2.3 Estructura de un documento HTML5

#### 2.4 Uso de las nuevas etiquetas semánticas

#### 2.5 Atributos globales

que puede tener cada atributo.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
">
```

Ésta es una de las 15 declaraciones posibles declaradas en los estándares HTML4 y XHTML. En HTML5 se reduce la definición del tipo de documento a una única posibilidad, por lo que no tenemos que preocuparnos de elegir el tipo de documento correcto:

```
<!DOCTYPE html>
```

## 2.1.2 ELEMENTO RAÍZ HTML

En todo documento HTML, su elemento raíz o nodo superior siempre es la etiqueta `<html>`. Hasta ahora, este elemento raíz se definía de la siguiente manera:

```
<html xmlns=http://www.w3.org/1999/xhtml
    lang="en"
    xml:lang="en">
```

No hay ningún problema en mantener esta sintaxis. Si se desea, se puede conservar, ya que es válido en HTML5. Sin embargo, algunas de sus partes ya no son necesarias, por lo que podemos eliminarlas.

El primer elemento del que podemos prescindir es el atributo `xmlns`. Se trata de una herencia de XHTML 1.0, que dice que los elementos de esta página están en el espacio de nombres XHTML, `http://www.w3.org/1999/xhtml`. Sin embargo, los elementos de HTML5 están siempre en este espacio de nombres, por lo que ya no es necesario declararlo explícitamente. Eliminar el atributo `xmlns` nos deja con este elemento de la siguiente manera:

```
<html lang="es" xml:lang="en">
```

En este caso ocurre lo mismo con el atributo `xml:lang`, es una herencia de XHTML que podemos eliminar, quedando finalmente la etiqueta de la siguiente manera:

```
<html lang="en">
```

### 2.1.3 ELEMENTO HEAD

El primer hijo del elemento raíz es generalmente el elemento `head`. El elemento `head` contiene los metadatos que aportan información extra sobre la página, como su título, descripción, autor, etc. Además, puede incluir referencias externas a contenidos necesarios para que el documento se muestre y comporte de manera correcta (como hojas de estilos o *scripts*). Este elemento ha sufrido pequeñas variaciones, pero que debemos tener en cuenta:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; c
harset=UTF-8" />
  <title>My Weblog</title>
  <link rel="stylesheet" type="text/css" href="style-or
iginal.css" />
  <link rel="alternate" type="application/atom+xml"
      title="My Weblog feed"
      href="/feed/" />
  <link rel="search" type="application/opensearchdescri
ption+xml"
      title="My Weblog search"
      href="opensearch.xml" />
  <link rel="shortcut icon" href="/favicon.ico" />
</head>
```

#### 2.1.3.1 CODIFICACIÓN DEL CONTENIDO

Cuando se piensa en "texto", probablemente nos venga a la cabeza una definición de "caracteres y símbolos que veo en la pantalla de mi ordenador". Pero realmente se tratan de bits y bytes. Cada cadena de caracteres que se muestra en la pantalla, se almacena con una codificación de caracteres en particular. Hay cientos de codificaciones de caracteres diferentes, algunos optimizado para ciertos idiomas como el ruso, el chino o inglés, y otros que se pueden utilizar para múltiples idiomas. En términos generales, la codificación de caracteres proporciona una correspondencia entre lo que se muestra en la pantalla y lo que un equipo realmente almacena en la memoria y en el disco.

Se puede pensar en la codificación de caracteres como una especie de

clave de descifrado del texto. Cuando accedemos a una secuencia de bytes, y decidimos que es "texto", lo que necesitamos saber es qué codificación de caracteres se utiliza para que pueda decodificar los bytes en caracteres y mostrarlos (o transformarlos) de manera correcta.

Lo ideal es establecer esta codificación en el servidor, indicando el tipo en las cabeceras de respuesta:

```
Content-Type: text/html; charset="utf-8"
```

Por desgracia, no siempre podemos tener el control sobre la configuración de un servidor HTTP. Por ejemplo, en la plataforma *Blogger*, el contenido es proporcionado por personas, pero los servidores son administrados por Google, por lo que estamos supeditados a su configuración. Aún así, HTML 4 proporciona una manera de especificar la codificación de caracteres en el documento HTML:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

El encabezado HTTP es el método principal, y anula la etiqueta `<meta>` si está presente. Pero no todo el mundo puede establecer encabezados HTTP, por lo que la etiqueta `<meta>` todavía tiene sentido. En el caso de HTML5, es aún más sencillo definir esta etiqueta *meta*:

```
<meta charset="utf-8">
```

Debemos acostumbrarnos a especificar la codificación de nuestros documentos, aunque no vayamos a utilizar caracteres especiales o nuestro documentos no se presente en otros idiomas. Si no lo hacemos, podemos exponernos a **problemas de seguridad**.

### 2.1.3.2 LINKS

Dentro del elemento `head`, las etiquetas `<link>` son una manera de acceder o declarar contenido externo al documento actual, que puede cumplir distintos objetivos:

- Es una hoja de estilo contiene las reglas CSS que su navegador debe aplicar al presente documento.
- Es un feed que contiene el mismo contenido que esta página, pero en

un formato estándar (RSS).

- Es una traducción de esta página en otro idioma.
- Es el mismo contenido que esta página, pero en formato PDF.
- Es el próximo capítulo de un libro en línea de la cual esta página es también una parte.

En HTML5, se separan estas relaciones de enlace en dos categorías:

- Enlaces a recursos externos que se van a utilizar para mejorar el documento actual,
- y enlaces de hipervínculos que son enlaces a otros documentos.

El tipo de relación más utilizado (literalmente) es el siguiente:

```
<link rel="stylesheet" href="style-original.css" type="text/css" />
```

Esta relación es utilizada para indicar el archivo donde se almacenan las reglas CSS que se desean aplicar al documento. Una pequeña optimización que se puede hacer en HTML5 es eliminar el atributo `type`. Sólo hay un lenguaje de estilo para la web, CSS, así que ese es el valor predeterminado para el atributo `type`:

```
<link rel="stylesheet" href="style-original.css" />
```

## 2.2 NUEVAS ETIQUETAS SEMÁNTICAS

En 2004, [Ian Hickson](#), el autor de la especificación de HTML5, analizó 1.000.000.000 de páginas web utilizando el motor de Google, intentando identificar la manera en la que la *web real* estaba construida. Uno de los resultados de este análisis, fue la [publicación](#) de una lista con los nombres de *clases* más utilizados. Este estudio revela que los desarrolladores utilizan *clases* o *IDs* comunes para estructurar los documentos. Esto llevó a considerar que quizás fuese una buena idea crear etiquetas concretas para reflejar estas estructuras.

Este tipo de etiquetas que componen la web semántica nos sirven para que cualquier mecanismo automático (un navegador, un motor de búsqueda, un lector de *feeds*...) que lea un sitio web sepa con exactitud qué partes de su contenido corresponden a cada una de las partes típicas de

un sitio. Observando esas etiquetas semánticas estructurales, cualquier sistema podrá procesar la página y saber cómo está estructurada. Veamos algunas de estas etiquetas que introduce HTML5 en este sentido.

- `<section></section>`: se utiliza para representar una sección "general" dentro de un documento o aplicación, como un capítulo de un libro. Puede contener subsecciones y si lo acompañamos de `h1-h6` podemos estructurar mejor toda la página creando jerarquías del contenido, algo muy favorable para el buen posicionamiento web.
- `<article></article>`: representa un componente de una página que consiste en una composición autónoma en un documento, página, aplicación, o sitio web con la intención de que pueda ser reutilizado y repetido. Podría utilizarse en los artículos de los foros, una revista o el artículo de periódico, una entrada de un blog, un comentario escrito por un usuario, un widget interactivo o cualquier otro artículo independiente de contenido. Cuando los elementos de `<article>` son anidados, los elementos interiores representan los artículos que en principio son relacionados con el contenido del artículo externo. Por ejemplo, un artículo de un blog que permite comentarios de usuario, dichos comentarios se podrían representar con `<article>`.
- `<aside></aside>`: representa una sección de la página que abarca un contenido relacionado con el contenido que lo rodea, por lo que se le puede considerar un contenido independiente. Este elemento puede utilizarse para efectos tipográficos, barras laterales, elementos publicitarios, para grupos de elementos de la navegación, u otro contenido que se considere separado del contenido principal de la página.
- `<header></header>`: representa un grupo de artículos introductorios o de navegación. Está destinado a contener por lo general la cabecera de la sección (un elemento `h1-h6` o un elemento `hgroup`), pero no es necesario.
- `<nav></nav>`: representa una sección de una página que enlaza a otras páginas o a otras partes dentro de la página. No todos los grupos de enlaces en una página necesita estar en un elemento `nav`, sólo las secciones que constan de bloques de navegación principales son apropiados para el elemento de navegación.

- `<footer></footer>`: representa el pie de una sección, con información acerca de la página/sección que poco tiene que ver con el contenido de la página, como el autor, el copyright o el año.
- `<hgroup></hgroup>`: representa el encabezado de una sección. El elemento se utiliza para agrupar un conjunto de elementos `h1`–`h6` cuando el título tiene varios niveles, tales como subtítulos o títulos alternativos.
- `<time>`: representa o bien una hora (en formato de 24 horas), o una fecha precisa en el calendario gregoriano (en formato **ISO**), opcionalmente con un tiempo y un desplazamiento de zona horaria.

## 2.3 ESTRUCTURA DE UN DOCUMENTO HTML5

Como hemos visto con las nuevas etiquetas semánticas introducidas en HTML5, éstas aportan un significado concreto al documento que estamos definiendo, y por lo tanto, afectan de manera directa a la forma en la estructuramos el contenido. Vamos a ver como podemos *convertir* nuestra actual página con las nuevas etiquetas introducidas en HTML5.

### 2.3.1 ESTRUCTURA TRADICIONAL DE UN DOCUMENTO EN HTML 4

El siguiente código muestra una estructura "clásica" de documento HTML, donde los diferentes contenidos de la web se encuentran agrupados por etiquetas `<div>`. Por sí mismas, estas etiquetas no aportan ningún tipo de significado, y el atributo `id` tampoco se lo proporciona. Si cambiamos `<div id="header">` por `<div id="whatever">`, el significado sigue siendo el mismo, *ninguno*.



```
<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//
EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.
dtd">
<html>
<body>
    <div id="header">
        <a href="/"><img src=logo.png alt="home"></a>
        <h1>My Weblog</h1>
        <p class="tagline">
            A lot of effort went into making this effortl
ess.
        </p>
    </div>
    <div id="nav">
        <ul>
            <li><a href="#">home</a></li>
            <li><a href="#">blog</a></li>
            <li><a href="#">gallery</a></li>
            <li><a href="#">about</a></li>
        </ul>
    </div>
    <div class="articles">
        <div class="article">
            <p class="post-date">October 22, 2009</p>
            <h2>
                <a href="#" title="link to this post">Tra
vel day</a>
            </h2>
            <div class="content">
                Content goes here...
            </div>
            <div class="comments">
                <p><a href="#">3 comments</a></p>
            </div>
        </div>
    </div>
    <div class="aside">
        <div class="related"></div>
        <div class="related"></div>
        <div class="related"></div>
```



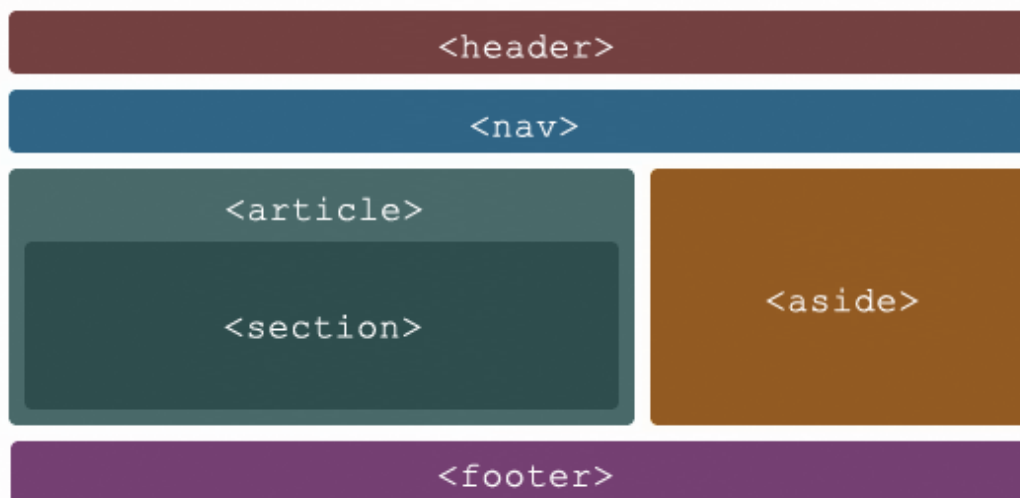
```

</div>
<div id="footer">
  <p>#167;</p>
  <p>#169; 2013#8211;9 <a href="#">Arkaitz Garro<
/a></p>
</div>
</body>
</html>

```

## 2.3.2 ESTRUCTURA TRADICIONAL DE UN DOCUMENTO EN HTML5

Veamos como podemos añadir un significado a este documento, únicamente aplicando las nuevas etiquetas semánticas incluidas en HTML5.



```

<!DOCTYPE html>
<html>
<body>
  <header>
    <a href="/"><img src=logo.png alt="home"></a>
    <hgroup>
      <h1>Title</h1>
      <h2 class="tagline">
        A lot of effort went into making this effortless.
      </h2>
    </hgroup>
  </header>
  <nav>

```

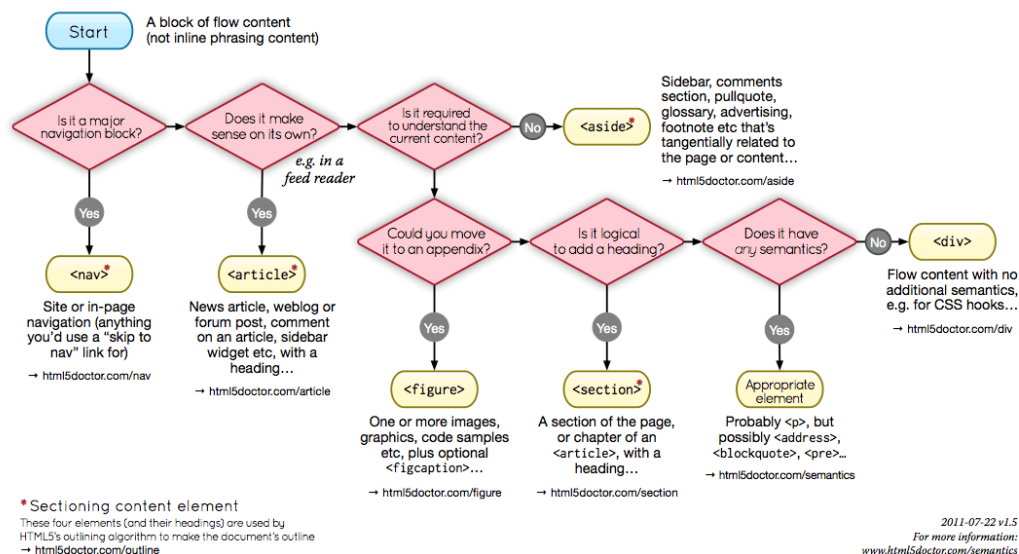
```
<ul>
  <li><a href="#">home</a></li>
  <li><a href="#">blog</a></li>
  <li><a href="#">gallery</a></li>
  <li><a href="#">about</a></li>
</ul>
</nav>
<section class="articles">
  <article>
    <time datetime="2009-10-22">October 22, 2009<
/ time>
    <h2>
      <a href="#" title="link to this post">Tra
vel day</a>
    </h2>
    <div class="content">
      Content goes here...
    </div>
    <section class="comments">
      <p><a href="#">3 comments</a></p>
    </section>
  </article>
</section>
<aside>
  <div class="related"></div>
  <div class="related"></div>
  <div class="related"></div>
</aside>
<footer>
  <p>&#167;</p>
  <p>&#169; 2013&#8211;9 <a href="#">Arkaitz Garro<
/a></p>
</footer>
</body>
</html>
```

Partiendo de la anterior estructura, parece evidente las nuevas etiquetas que debemos utilizar. Esto no siempre es así, y cuando estructuramos contenidos de mucho mayor alcance, lo normal es que nos surjan dudas. Un sencillo algoritmo que nos puede ayudar en la correcta selección de etiquetas, es el que proponen en [HTML5 Doctor](#).

# html5 Doctor HTML5 Element Flowchart

Sectioning content elements and friends

By @riddle & @boblet  
www.html5doctor.com



## 2.4 USO DE LAS NUEVAS ETIQUETAS SEMÁNTICAS

### 2.4.1 <HEADER>

Según la especificación, un elemento `<header>` representa lo siguiente:

*Un grupo de navegación o contenido introductorio. Un elemento header normalmente contiene una sección de encabezado (un elemento `h1-h6` o un elemento `hgroup`), pero puede contener otro tipo de elementos, como una tabla de contenidos, un formulario de búsqueda o cualquier logo importante.*

En nuestro ejemplo, y en la mayoría de los sitios web, la cabecera contiene los primeros elementos de la página. Tradicionalmente el título de la web, su logo, enlaces para volver al inicio... De la manera más simple, nuestra cabecera quedaría de esta forma:

```
<header>
  <a href="/"><img src=logo.png alt="home"></a>
  <h1>Title</h1>
</header>
```

También es muy común que los sitios web muestren un lema o subtítulo bajo el título principal. Para dar mayor importancia a este subtítulo, y

relacionarlo de alguna manera con el título principal de la web, es posible agrupar los dos titulares bajo un elemento `<hgroup>`.

```
<header>
  <a href="/"><img src=logo.png alt="home"></a>
  <hgroup>
    <h1>Title</h1>
    <h2 class="tagline">
      A lot of effort went into making this effortl
ess.
    </h2>
  </hgroup>
</header>
```

## 2.4.2 <NAV>

Según la especificación, un elemento `<nav>` representa lo siguiente:

*El elemento `<nav>` representa una sección de una página que enlaza con otras páginas o partes de la misma página: una sección con enlaces de navegación.*

El elemento `<nav>` ha sido diseñado para identificar la navegación de un sitio web. La navegación se define como un conjunto de enlaces que hacen referencia a las secciones de una página o un sitio, pero no todos los enlaces son candidatos de pertenecer a un elemento `<nav>`: una lista de enlaces a patrocinadores o los resultados de una búsqueda, no forman parte de la navegación principal, sino que corresponden con el contenido de la página.

Como ocurre con los elementos `<header>`, `<footer>` y el resto de nuevas etiquetas, no estamos obligados a utilizar un único elemento `<nav>` en toda la página. Es posible que tengamos una navegación principal en la cabecera de la página, una tabla de contenidos o enlaces en el pie de la página, que apuntan a contenidos secundarios. Todos ellos son candidatos a pertenecer a un elemento `<nav>`.

```
<nav>
  <ul>
    <li><a href="#">home</a></li>
    <li><a href="#">blog</a></li>
```

```
<li><a href="#">gallery</a></li>
<li><a href="#">about</a></li>
</ul>
</nav>
```

### 2.4.3 <FOOTER>

Según la especificación, un elemento `<footer>` representa lo siguiente:

*Representa el pie de una sección. Un pie tradicionalmente contiene información acerca de su sección, como quién escribió el contenido, enlaces relacionados, copyright y similares.*

Al igual que ocurre con el elemento `<nav>`, podemos tener tantos elementos `<footer>` como sea necesario. Lo normal es que nuestro sitio web disponga de al menos un pie principal, que contiene los avisos legales (privacidad, condiciones del servicio, copyright...), mapa del sitio web, accesibilidad, contacto y otros muchos enlaces que pueden ir incluidos en un elemento `<nav>`.

### 2.4.4 <ARTICLE>

Según la especificación, un elemento `<article>` representa lo siguiente:

*Este elemento representa un contenido completo, auto-contenido en un documento, página, aplicación o sitio web, que es, en principio, independiente de ser distribuido y reutilizado, por ejemplo en un RSS. Puede ser un post de un foro, un artículo de un periódico o revista, una entrada de un blog, un comentario de un usuario, un widget o cualquier otro elemento independiente.*

*Cuando los artículos están anidados, los artículos interiores representan contenido que en principio está relacionado con el artículo que los contiene. Por ejemplo, una entrada de un blog puede aceptar comentarios de usuarios, que están incluidos dentro del contenido principal y relacionados con el mismo.*

Por lo tanto, la etiqueta `<article>` se utiliza para encapsular contenido, que tiene significado en sí mismo, y que puede ser distribuido y reutilizado en otros formatos de datos. No nos referimos únicamente a contenidos clásicos de texto, sino que incluso un contenido multimedia con su transcripción, un mapa o email pueden ser totalmente válidos para ser

incluidos en una etiqueta `<article>`.

```
<section>
  <h1>Comments</h1>
  <article id="c1">
    <footer>
      <p>Posted by: <span>George Washington</span><
/p>
      <p><time datetime="2009-10-10">15 minutes ago
</time></p>
    </footer>
    <p>Yeah! Especially when talking about your lobby
ist friends!</p>
  </article>
  <article id="c2">
    <footer>
      <p>Posted by: <span itemprop="name">George Ha
mmond</span></p>
      <p><time datetime="2009-10-10">5 minutes ago<
/time></p>
    </footer>
    <p>Hey, you have the same first name as me.</p>
  </article>
</section>
```

### 2.4.5 <SECTION>

A diferencia del elemento `<article>`, este elemento es utilizado para dividir el documento (o artículos) en diferentes áreas, o como su propio nombre indica, en secciones. Según la especificación, un elemento `<section>` representa lo siguiente:

*Representa una sección genérica de un documento o aplicación. Una sección, en este contexto, es un grupo temático de contenido, que generalmente incluye una cabecera.*

Consideremos el siguiente marcado válido en HTML 4:

```
<h1>Rules for Munchkins</h1>
<h2>Yellow Brick Road</h2>
<p>It is vital that Dorothy follows it—so no selling
bricks as "souvenirs"</p>
```

```

<h2>Fan Club uniforms</h2>
<p>All Munchkins are obliged to wear their "I'm a friend
  of Dorothy!" t-shirt when representing the club</p>
<p><strong>Vital caveat about the information above:
  does not apply on the first Thursday of the month.</s
strong></p>

```

En este caso, y desde un punto de vista semántico, es complicado deducir si el texto *Vital caveat about the information above: does not apply on the first Thursday of the month.* pertenece al contenido completo o está relacionado con la sección *Fan Club uniforms*. Gracias a la etiqueta `<section>`, es muy sencillo separar e identificar a qué sección pertenece cada contenido:

```

<article>
  <h1>Rules for Munchkins</h1>
  <section>
    <h2>Yellow Brick Road</h2>
    <p>It is vital that Dorothy follows it—so no sell
ing
      bricks as "souvenirs"</p>
  </section>
  <section>
    <h2>Fan Club uniforms</h2>
    <p>All Munchkins are obliged to wear their "I'm a
friend
      of Dorothy!" t-shirt when representing the cl
ub</p>
  </section>
  <p><strong>Vital caveat about the information above:
    does not apply on the first Thursday of the month
.</strong></p>
</article>

```

```

<article>
  <h1>Rules for Munchkins</h1>
  <section>
    <h2>Yellow Brick Road</h2>
    <p>It is vital that Dorothy follows it—so no sell
ing
      bricks as "souvenirs"</p>

```

```

</section>
<section>
  <h2>Fan Club uniforms</h2>
  <p>All Munchkins are obliged to wear their "I'm a
friend
      of Dorothy!" t-shirt when representing the cl
ub</p>
  <p><strong>Vital caveat about the information abo
ve:
      does not apply on the first Thursday of the m
onth.</strong></p>
</section>
</article>

```

Como podemos observar en los dos ejemplos anteriores, es muy sencillo agrupar contenido que pertenece a una misma sección, permitiendo incluirlo dentro de un contexto semántico.

Otra de las posibilidades que nos ofrece esta etiqueta, es la de dividir nuestro documento en secciones, que incluyen contenido de temáticas diferentes entre sí. Si además queremos separar estos contenidos visualmente en dos columnas, lo lógico sería utilizar las tradicionales etiquetas `<div>` para agrupar los artículos según su temática, y posteriormente aplicar estilos CSS o JavaScript para presentarlos en forma de *pestañas*.

En este caso, la etiqueta `<div>` no nos aporta ningún significado semántico, tan sólo estructural. La etiqueta `<section>` es la encargada de añadir semántica en estos casos:

```

<section>
  <h1>Articles about llamas</h1>
  <article>
    <h2>The daily llama: Buddhism and South American
camelids</h2>
    <p>blah blah</p>
  </article>
  <article>
    <h2>Shh! Do not alarm a llama</h2>
    <p>blah blah</p>
  </article>

```



```
</section>
<section>
  <h1>Articles about root vegetables</h1>
  <article>
    <h2>Carrots: the orange miracle</h2>
    <p>blah blah</p>
  </article>
  <article>
    <h2>Eat more Swedes (the vegetables, not the people)</h2>
    <p>blah blah</p>
  </article>
</section>
```

## 2.4.6 <ASIDE>

Según la especificación, un elemento `<aside>` representa lo siguiente:

*Una sección de una página que consiste en contenido tangencialmente relacionado con el contenido alrededor del elemento, y puede considerarse separado de este contenido. Estas secciones son normalmente representadas como elementos laterales en medios impresos. Este elemento puede utilizarse para contener citas, anuncios, grupos de elementos de navegación y cualquier otro contenido separado del contenido principal de la página.*

Dentro de un artículo, por ejemplo, puede ser utilizado para mostrar contenido relacionado como citas u otros artículos relacionados.

## 2.4.7 <FIGURE>

Según la especificación, un elemento `<figure>` representa lo siguiente:

*The figure element represents some flow content, optionally with a caption, that is self-contained and is typically referenced as a single unit from the main flow of the document.*

*The element can be used to annotate illustrations, diagrams, photos, code listings, etc. This includes, but is not restricted to, content referred to from the main part of the document, but that could, without affecting the flow of the document, be moved away from that primary content, e.g. to the side of the page, to dedicated pages,*

| *or to an appendix.*

Hasta ahora, no había una manera correcta de poder añadir un subtítulo o una leyenda a un contenido concreto, como explicar una figura o atribuir una imagen a un fotógrafo. Gracias a la etiqueta `<figure>` podemos contener una imagen (o un vídeo, ilustración o bloque de código) en un elemento y relacionarlo con un contenido concreto:

```
<figure>
  
  <figcaption>
    Bruce and Remy welcome questions
    <small>Photo &copy; Bruce's mum</small>
  </figcaption>
</figure>
```

En conveniente recordar que el atributo `alt` indica el texto a mostrar cuando la imagen no está disponible, y no está pensada para contener una descripción de la imagen, y mucho menos para duplicar lo ya indicado en la etiqueta `<figcaption>`.

Ejercicio 1

[Ver enunciado](#)

## 2.5 ATRIBUTOS GLOBALES

HTML5 también incluye nuevos atributos globales que pueden ser asignados a cualquier elemento. Son los siguientes:

### 2.5.1 ACCESSKEY

El atributo `accesskey` permite a los desarrolladores especificar un atajo de teclado que permite activar un elemento a asignarle el foco. Este atributo ya existía en HTML 4, aunque ha sido utilizado en muy pocas ocasiones. Como HTML5 está pensado para aplicaciones, y algunos usuarios siguen prefiriendo los atajos de teclado, este atributo no ha sido eliminado, y ahora está disponible para cualquier elemento.

Para evitar conflictos con otros atajos de teclado, o con los propios del navegador, ahora esta etiqueta permite asignar alternativas en este atributo. Un ejemplo incluido en la especificación:

```
<input type="search" name="q" accesskey="s 0">
```

Esto quiere decir que este elemento es accesible a través de dos atajos de teclado, a través de la tecla `s` o a través de la tecla `0` (en ese orden).

## 2.5.2 CONTENTEDITABLE

Inventado por Microsoft, e implementado por el resto de los navegadores, la etiqueta `contenteditable` es ahora parte de la especificación oficial.

La introducción de esta etiqueta significa principalmente dos cosas:

- Primero, los usuarios pueden editar los contenidos de un elemento que incluya esta etiqueta. Este elemento debe ser seleccionable y el navegador debe proporcionar una marca que indique la posición actual del cursor.
- Y segundo, es posible cambiar el formato del texto del contenido, añadiendo negritas, cambiar la fuente, añadir listas, etc.

Este atributo es de tipo *booleano*, por lo que su valor puede ser *true* o *false*. Al acceder desde JavaScript a este atributo, hay que tener en cuenta su notación *lowerCamelCase*, siendo el nombre de la propiedad del DOM `contentEditable`. Además, existe otra propiedad llamada `isContentEditable`, que indica si el elemento es editable o no.

Finalmente, el contenido que ha sido seleccionado por el usuario, puede ser objeto de modificaciones, como hemos comentado antes. A través del comando `element.execCommand()` es posible indicar el tipo de modificación (poner en negrita, copiar, cambiar la fuente...), siempre que el documento se haya indicado como editable.

```
document.designMode = 'on';
```

Si se desea almacenar los cambios realizados en el contenido, es necesario enviarlo al servidor. No existe ningún API o método en JavaScript que nos posibilite esta acción, por lo que debemos utilizar algún tipo de tecnología tipo AJAX.

## 2.5.3 DATA-\* (CUSTOM DATA ATTRIBUTES)

HTML5 permite crear atributos personalizados para los elementos. Estos atributos son utilizados para pasar información a JavaScript. Como

veremos en el capítulo correspondiente, hasta ahora se utilizaba el atributo *class* para de alguna manera almacenar información asociada con elementos, pero esto cambia radicalmente con estos atributos.

```
<ul id="vegetable-seeds">
  <li data-spacing="10cm" data-sowing-time="March to June"
">Carrots</li>
  <li data-spacing="30cm" data-sowing-time="February to M
arch">Celery</li>
  <li data-spacing="3cm" data-sowing-time="March to Septe
mber">Radishes</li>
</ul>
```

#### 2.5.4 DRAGGABLE

Este atributo indica que el elemento indicado puede ser *arrastable*. Lo veremos en el capítulo correspondiente.