

La sintaxis y la estructura en html5



En este segundo capítulo de la **serie sobre HTML5** veremos los elementos que forman la estructura de una web.

En HTML4, el elemento principal para estructurar las páginas web era la famosa caja `<div>`. Con el elemento `<div>` es posible crear zonas de visualización de forma rectangular que identificamos con un id o una clase para luego darle formato con CSS.

El problema de los `<div>`, y esta es una de las cosas que pretende solucionar HTML5, es que se distinguen unos de otros gracias a ese id o esa clase. Identificador que le habrá dado el diseñador web de turno y que **muy probablemente habrá asignado en función de su humor o de sus ganas en ese momento.**

Por esta razón, las cajas `<div>` no son semánticas. Muchas veces no nos aportan ningún dato sobre su contenido.

En HTML5 se ha hecho especial hincapié también en la simplicidad.

Ahora veremos como ha cambiado todo esto enHTML5.

La nueva sintaxis HTML5

En HTML5 destaca sobretodo la simplicidad y la permisividad.

El doctype

La primera línea de un documento corresponde a la DTD (Document Type Declaration) o declaración de tipo de documento y sirve para indicar que versión de lenguaje HTML se ha usado al escribirlo.

En HTML5 esta declaración no puede ser más sencilla:

```
<!DOCTYPE html>
```

```
1 <!DOCTYPE html>
```

Y ya esta, no necesitas nada más.

El documento HTML

En una página web, el elemento `<html>` indica el inicio del contenido HTML. En HTML sólo necesitas indicar el lenguaje de la pagina. Por ejemplo:

```
<html lang=es>
```

```
1 <html lang=es>
```

La codificación de los caracteres

Es frecuente indicar qué codificación de caracteres se ha usado en una página web. Con HTML5 se simplifica con lo indispensable para que los navegadores puedan gestionarla correctamente.

```
<meta charset="UTF-8" />
```

```
1 <meta charset="UTF-8" />
```

Los scripts

Una vez más, se ha simplificado la declaración de scripts de la siguiente forma:

```
<script src="unscript.js">
</script>
```

```
1 <script src="unscript.js"></script>
```

Esto es debido a que ahora se da por sentado que los scripts estan escritos en Javascript.

Los estilos CSS

Seguimos con la simplificación. Ahora para la declaración de los estilos te bastará con:

```
<link
href="tuhojadeestilos.css"
```

```
1 <link href="tuhojadeestilos.css" rel="stylesheet" />
```

La sintaxis de los elementos

Las comillas en HTML5 pasan a ser facultativas para los valores de los atributos. Puedes escribirlos con comillas dobles, comillas simples o sin ellas. Estas tres sintaxis serían correctas:

```
<div class="tuestilo">
<div class='tuestilo'>
```

```
1 <div class="tuestilo">
```

```
2 <div class='tuestilo'>
```

```
3 <div class=tuestilo>
```

En los elementos HTML con una sola etiqueta de apertura como por ejemplo `br` o `img` ya no es necesario indicar el cierre del elemento usando `/`. Por ejemplo:

```

```

```
1 
```

HTML5 permite el uso de mayúsculas y minúsculas indistintamente en la sintaxis de los elementos y atributos. De este modo, podrías declarar un script así:

```
<SCRIPT  
src="TuJavascript.js">
```

```
1 <SCRIPT src="TuJavascript.js"></script>
```

Como puedes ver, HTML5 es muy permisivo a la hora de escribir código. Ofrece una gran flexibilidad y muy pocas restricciones. Lo cual, también puede ser un problema. Imagina un código en el que hayan trabajado dos o tres desarrolladores, con la permisividad de HTML5, si cada uno ha escrito el código a su manera, lo que tu veas al final, puede resultar un auténtico caos.

Es por eso, que te recomiendo seguir unas sencillas pautas que te harán la vida mas fácil a ti y a otros desarrolladores que trabajen sobre tu código.

- Usa sólo minúsculas
- Usa siempre comillas para los valores de los atributos
- Cierra con `/` las etiquetas con una sola etiqueta de apertura.
- Cierra los elementos aunque su etiqueta sea facultativa.
- Utiliza el sangrado para que el código sea mas legible.

Los elementos de la estructura en HTML5

Con la llegada de HTML5 han llegado nuevos elementos de

estructura para mejorar la semántica. Esto es, dotar a estos elementos de un significado.

¿Cuáles son esos elementos? No te preocupes, los vemos a continuación.

El elemento **<header>**

El nuevo elemento **<header>** se utiliza para definir una zona de visualización para las cabeceras. Puedes definir cabeceras tanto a nivel de página como de una zona determinada (un artículo, un menú, etc...)

De este modo, puedes tener una cabecera para toda la página que sería la típica cabecera en la parte alta de la pagina con un logo, un menu, etc... O puedes tener una cabecera para una zona determinada como por ejemplo un artículo, en el que la cabecera podría incluir el título, el autor y la fecha.

El elemento **<footer>**

Funciona de un modo muy parecido al **<header>**. Del mismo modo, podrás determinar un **<footer>** para toda la página o utilizarlo en diferentes secciones de la web como por ejemplo un sidebar o un artículo.

A nivel de página sería la típica zona en la parte baja de la web, donde se suelen incluir datos de contacto, enlaces, etc... A nivel de zonas, y siguiendo con el ejemplo anterior, en el **<footer>** de un artículo podrías incluir por ejemplo la categoría a la que pertenece ese artículo o el autor si no lo has puesto en el **<header>**

El elemento **<nav>**

Como su nombre ya deja intuir, el elemento **<nav>** sirve para definir una zona de navegación con vínculos. Lo que vendría a ser un menú de toda la vida. Este elemento esta pensado para la navegación principal de la web y para definir vínculos entre las diferentes páginas.

El elemento **<section>**

La función del elemento **<section>** es agrupar elementos relacionados entre sí. De este modo, podrás por ejemplo, agrupar dentro de un mismo elemento un contenido con su título y su pie de página.

El elemento **<article>**

El **<article>** sirve para definir un contenido autónomo e independiente, que pueda ser usado en otra parte de la web sin que por ello pierda su significado.

A modo de ejemplo, usarías el elemento **<article>** para un artículo de un blog, como este que estas leyendo ahora mismo, o una noticia de un periódico o incluso un comentario de un blog.

El elemento **<aside>**

La función del elemento **<aside>** es mostrar un contenido relacionado al contenido al cuál esta vinculado. Puede tratarse de sidebars, zonas de widgets, complementos sobre un artículo, etc...

En la mayoría de los casos un **<aside>** se ubicará en un lateral del contenido con el que están relacionados. Pero recuerda que no tiene porque ser así. La finalidad de las nuevas etiquetas es dotar de semántica al contenido de una web, para la ubicación de los elementos debes usar CSS.

Los **<div>**

Por supuesto, la llegada de HTML5, no significa que tengas que erradicar las cajas **<div>**. Todavía puedes usarlas y **aun tienen su utilidad**.

Si deseas ampliar esta información o profundizar mas en alguno de los elementos, te recomiendo que leas la recomendación del 28 de Octubre de 2014 de la [W3C](#).

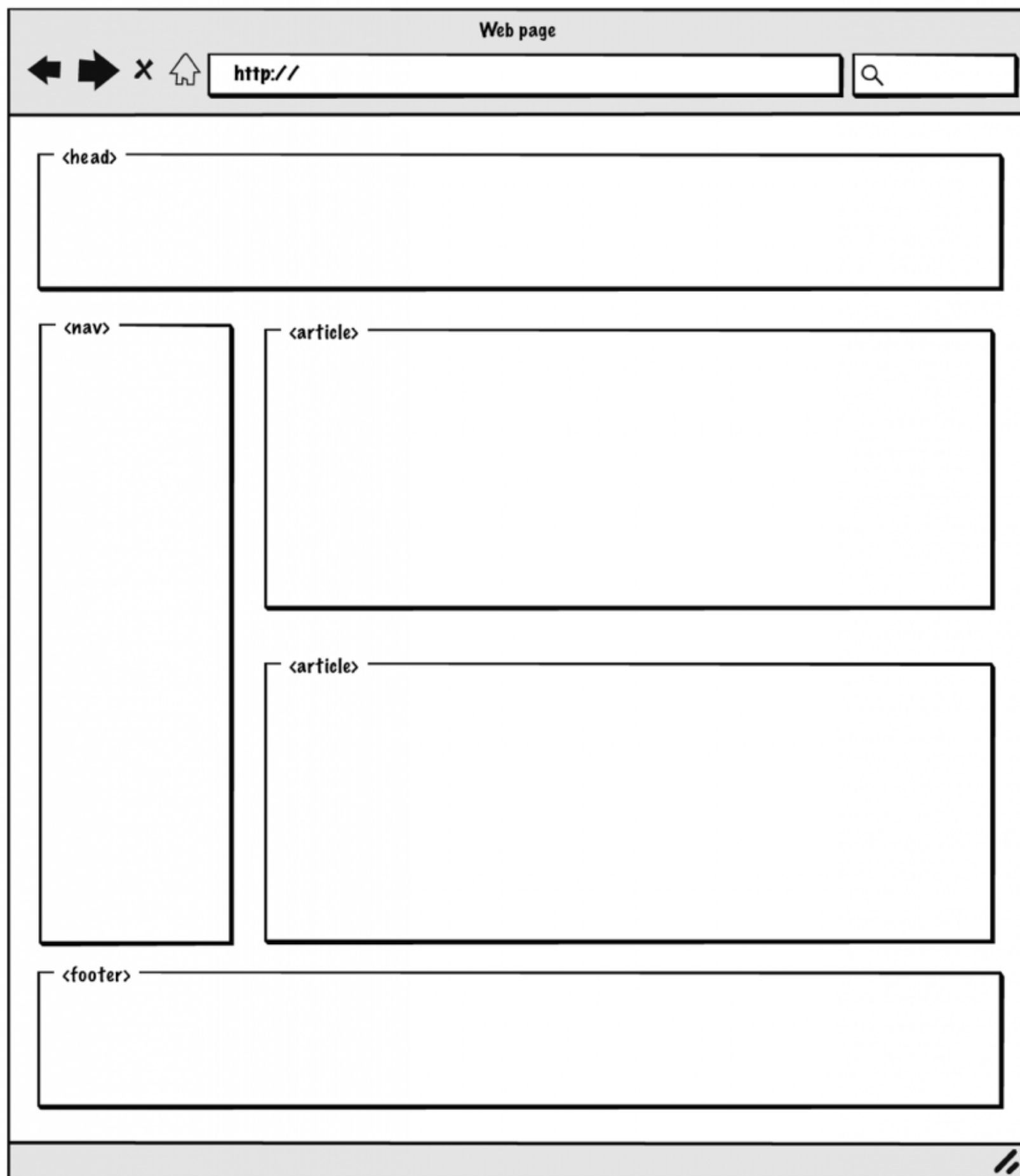
Y por supuesto, no te preocupes si todo esto te ha sonado un poco a chino.

Ahora veremos algunos ejemplos de utilización de estos elementos

estructurales para que entiendas un poco mejor el uso de cada uno de ellos.

Ejemplos de estructura en HTML5

Estructura simple



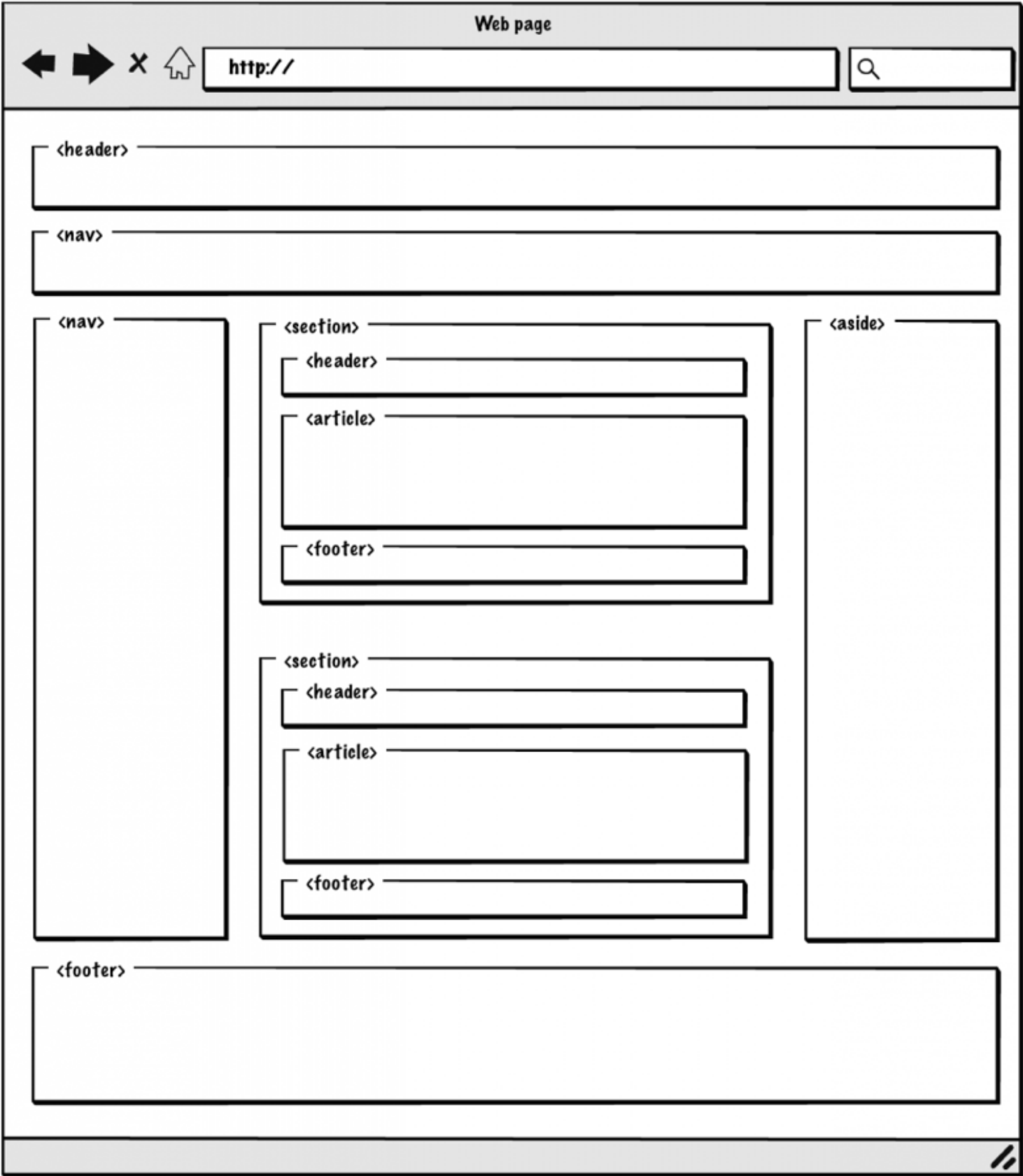
En este primer ejemplo tendrías un `<header>` con los elementos del encabezado de la pagina como el logo o el eslogan de la web.

En la sección <nav> tendrías el menú con los vínculos que permiten navegar por la web.

Dos secciones <article> que evidentemente serán los artículos de la web.

Y terminas con un <footer> que podría contener por ejemplo el copyright y los avisos legales con sus vínculos.

Estructura un poco mas compleja



En este segundo ejemplo, tendríamos un <header> como en el anterior.

Debajo del <header> aparece el primer cambio. Un <nav> horizontal que contendría el menú principal de la web con los vínculos a las diferentes páginas.

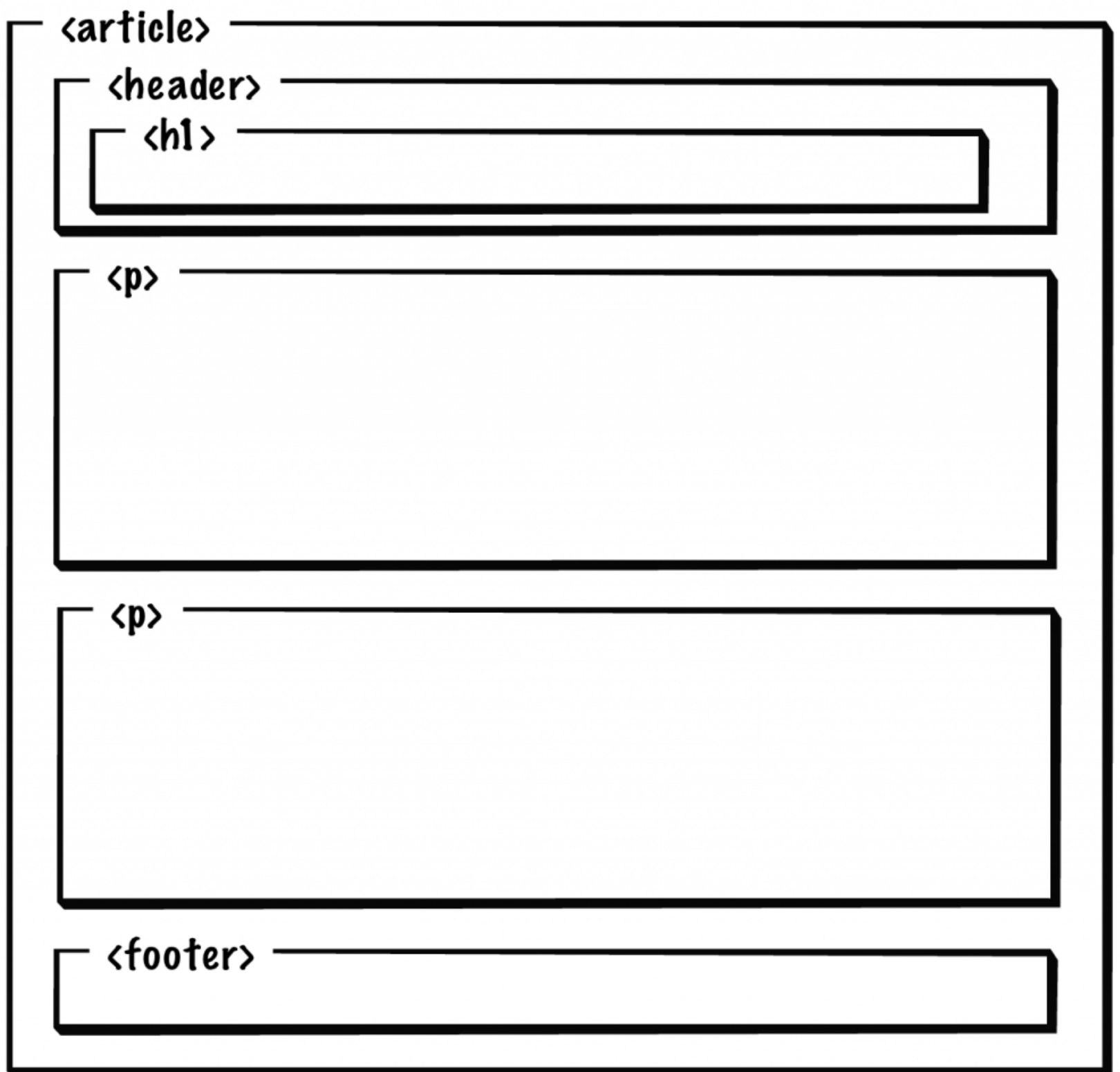
A la izquierda, otro elemento <nav> con el menú secundario que contendría vínculos directamente relacionados con el contenido de esa página.

El contenido principal, en el centro, con dos elementos <section> para distinguir dos tipos de contenido diferentes. Cada <section> contiene un <header> para el título, un <article> con el contenido textual y un <footer> con la información de la categoría y el autor, por ejemplo.

A la derecha, un elemento <aside> que podría mostrar otra información relacionada con el contenido como por ejemplo contenidos relacionados.

Termina la estructura nuevamente con un <footer> típico donde podría aparecer, como no, el copyright y los avisos legales con sus vínculos.

Estructura de un artículo



En este último ejemplo, vemos una posible estructura para un artículo.

El artículo, evidentemente, lo definiremos con un elemento `<article>`.

Ya dentro del artículo, primero encontraríamos un `<header>` con un `<h1>` para el título del artículo.

Seguimos con el contenido textual, dentro de dos elementos `<p>`.

Y terminamos con un elemento `<footer>` donde podríamos definir la categoría, la fecha y el autor.

Como has podido ver, los nuevos elementos de HTML5, además de hacer el

código más semántico, también te permitirán entender mucho mejor un código ya escrito con menos esfuerzo. Ordenar y clasificar mucho mejor el contenido de tu web también será mucho más sencillo.

Hoy te he querido mostrar algunos ejemplos para que te hagas una idea de por donde van los tiros. En próximas entregas, veremos todo esto con ejemplos reales, y por supuesto, paso a paso.

¿Y tu, ya usas HTML5 para crear la estructura de tu web?