

HTML5



HTML: *HyperText Markup Language – Lenguaje de marcas de hipertexto*

1. INTRODUCCIÓN

Los ficheros que contienen una página web, son ficheros **de texto** que tienen la extensión **.html**, (*index.html, practica01.html, cajas.html, etc..*) escritos con un lenguaje de etiquetas denominado **html**.

HTML es el lenguaje de marcado predominante utilizado para describir el contenido, o los datos, en la Web. HTML5 es la última iteración de este lenguaje de marcas, e incluye nuevas características, mejoras en las características existentes, y APIs basadas en scripts.

API (del inglés: *Application Programming Interface*), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Enmascara una compleja tarea de programación para ofrecer un servicio al usuario mucho más **fácil de utilizar**.

Dicho esto, HTML5 no es una reformulación de las versiones anteriores del lenguaje -que incluye todos los elementos válidos tanto de HTML 4 como de XHTML 1.0. Ha sido diseñado con algunos principios primarios en mente para asegurarse de que funciona en casi todas las plataformas, es compatible con los navegadores más antiguos y se ocupa de los errores correctamente.

HTML5 incluye **redefiniciones** de elementos existentes, y **nuevos elementos** que permiten a los diseñadores web ser más expresivos en la **semántica** de su código. El término HTML5, además, se ha utilizado para referirse a una serie de otras nuevas tecnologías y APIs. Algunos de estas incluyen el dibujo, con el elemento <canvas>, almacenamiento fuera de línea, el nuevo <video> y <audio>, la función de arrastrar y soltar (drag-and-drop), fuentes incrustadas, y otros.

HTML5 es, de hecho, una mejora de la combinación entre Javascript, HTML y CSS, el pegamento que lo une todo. A partir de ahora, **HTML** provee los elementos estructurales, **CSS** se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y **Javascript** tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales.

Año	Versión		
		2000	W3C Recommendation: XHTML 1.0
1989	Tim Berners-Lee invented www	2008	WHATWG HTML5 First Public Draft
1991	Tim Berners-Lee invented HTML	2012	WHATWG HTML5 Living Standard
1993	Dave Raggett drafted HTML+	2014	W3C Recommendation: HTML5
1995	HTML Working Group defined HTML 2.0	2016	W3C Candidate Recommendation: HTML 5.1
1997	W3C Recommendation: HTML 3.2	2017	W3C Recommendation: HTML5.1 2nd Edition
1999	W3C Recommendation: HTML 4.01	2017	W3C Recommendation: HTML5.2

Es conveniente que las páginas web definidas en los archivos .html contengan sólo:

- **Estructura** (cajas o contenedores).
- **Contenido** (texto, imágenes, vídeos, enlaces...).

El **aspecto** de la página (maquetación, distribución de las cajas, colores, tipo de fuentes, colores de las fuentes, etc.), aunque también se puede incluir dentro del fichero .html, por cuestiones de legibilidad y economía de código, es mejor separarlo, y definirlo en el archivo **de estilos**, que tiene extensión **.css**. Por ejemplo: *estilos.css*, *estilosweb.css*, *estilospractica01.css*, etc.

Las **etiquetas html** son un conjunto de palabras reservadas que van entre los símbolos **<** y **>**. Deben estar escritas **en minúsculas**. Por ejemplo:

<code><html></code>	<code><h1></code>	<code><div></code>	<code></code>	<code><le></code>
<code><head></code>	<code><h2></code>	<code></code>	<code><object></code>	<code><p></code>
<code><body></code>	<code><a></code>	<code><table></code>	<code></code>	<code></code>

Las etiquetas tienen un significado concreto para el programa que interpreta el contenido del fichero html y lo muestra gráficamente. Este programa se denomina **navegador web** (del inglés web browser) por ejemplo: Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, Safari, Opera, Tor, etc.

El navegador web espera siempre una **estructura básica fija** con unas etiquetas ubicadas en posiciones determinadas que hacen que el programa considere que la página web está correctamente definida.

Básicamente hay dos tipos de etiquetas. En primer lugar aquellas que tienen una etiqueta **de apertura** y otra cierre. Las etiquetas de cierre se diferencian de las de apertura simplemente en que llevan la barra inclinada **/** antes del nombre de la etiqueta:

`<p>` Párrafo `</p>`
`<div>` Caixa o Contenedor `</div>`
`<body>` `<h1>` Capçalera dentro del cuerpo de la página `</h1>` `</body>`

Entre apertura y cierre se sitúa el contenido de la etiqueta, que puede incluir **texto** u otras **etiquetas** que también se deben abrir y cerrar.

Este ejemplo de estructura sería **correcto**:

`<p>` Pàrrafo `` amb `` un texto `</p>`

Y este otro sería **incorrecto**:

`<p>` Pàrrafo `` amb un texto `</p>`

Ya que las etiquetas a que se abren en primer lugar, se deben cerrar después de las que se abren posteriormente.

El otro tipo de etiquetas son aquellas que **no necesitan una etiqueta de cierre**, y simplemente incluyen la barra inclinada `/` al final de la única etiqueta que las compone. Por ejemplo:

`<hr/>`
`
`
``

Las etiquetas pueden incluir modificadores de su significado llamados **parámetros**. Se sitúan a continuación del nombre de la etiqueta, separados por espacios y siempre tienen a continuación el símbolo igual (`=`) y un valor que va entre comillas dobles (`" "`). Hay parámetros que se pueden definir en la mayoría de las etiquetas y otros que son específicos de un conjunto más reducido. Por ejemplo:

`<p style="color:red;">`Pàrrafo de texto en color rojo`</p>`
``
`<h1 style="color:blue;">`Cabecera en color azul`</h1>`
`<div id="contenido">`Caja identificada con el nombre contenido`</div>`

2. ESTRUCTURA BÁSICA <HTML> Y DE CABECERA <HEAD>

Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas.

<!DOCTYPE>

En primer lugar necesitamos indicar el tipo de documento que estamos creando. En contraposición con la definición extensa y compleja utilizada en las diferentes versiones de HTML 4.01 y XHTML 1.0:

`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"`
`"http://www.w3.org/TR/html4/strict.dtd">`

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

Esto en HTML5 es extremadamente sencillo:

`<!DOCTYPE html>`

<html>

En HTML, la estructura tipo árbol tiene su raíz en el elemento <html> que envolverá al resto del código:

```
<!DOCTYPE html>
<html lang="es">
</html>
```

El atributo **lang** en la etiqueta de apertura <html> es el único atributo que necesitamos especificar en HTML5. Este atributo define el idioma humano del contenido del documento que estamos creando. (https://iso639-3.sil.org/code_tables/639/data)

En XHTML, era obligatorio insertar el parámetro xmlns, propio de una aplicación XML, que ahora desaparece:

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

<head>

El código HTML insertado entre las etiquetas <html> tiene que ser dividido entre dos secciones principales. La primera sección es la **cabecera** y la segunda el **cuerpo**. Estas secciones se crean en el código usando los elementos <head> y <body>.

El propósito de <head> sigue siendo exactamente el mismo:

- Definir el **título** de nuestra página web
- Declarar el set de **caracteres** correspondiente,
- Proporcionar **información general** acerca del documento
- Incorporar los **archivos** externos que definen
 - **Estilos**,
 - Códigos **Javascript**
 - *Imágenes necesarias para generar la página en la pantalla.*

La información incorporada en el documento entre estas etiquetas es **invisible para el usuario**, excepto por el título y algunos íconos.

<body>

El cuerpo representa la **parte visible** de todo documento y es especificado entre etiquetas <body> </body>

```
<!DOCTYPE html>
<html lang="es">
  <head>

  </head>
  <body>

  </body>
```

Elementos de <head>:

<meta>

Algunos cambios e innovaciones fueron incorporados dentro de la cabecera. Uno de ellos es la etiqueta que define el **juego de caracteres** a utilizar para mostrar el documento. Una etiqueta <meta> especifica cómo el texto será presentado en pantalla:

HTML5 conlleva una **simplificación**, como en la mayoría de sus novedades, de la etiqueta <meta>. Anteriormente la declaración tenía el siguiente aspecto:

```
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
```

Ahora, la definición del tipo de caracteres es más **corta y simple**, utilizamos el atributo **charset** para especificar el tipo de codificación. Se pueden agregar otras etiquetas <meta> utilizando el atributo **name** con los valores **description** o **keywords** para describir el tipo de metadatos, palabras clave del documento) y el atributo **content** para especificar el valor.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
</head>
<body>
</body>
</html>
```

En HTML5 no es necesario **cerrar etiquetas simples** con una barra al final como sucedía en XHTML, pero es **recomendable** utilizarlas por razones de **compatibilidad**. El anterior código se podría escribir de la siguiente manera, en ambos casos sería HTML5 válido:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1"/>
  <meta name="description" content="Introducción HTML5"/>
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria"/>
</head>
<body>

</body>
</html>
```

<title>

La etiqueta <title>, simplemente especifica el **título del documento**, que será visible en la pestaña del navegador:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
</head>
<body>

</body>
</html>
```

<link>

Este elemento se utiliza para incorporar estilos, imágenes o iconos desde **archivos externos**. Uno de los usos más comunes para <link> es la incorporación de archivos con estilos CSS. Para cargar un icono, tendremos que incluir el elemento link junto el atributo **rel** definido como **icon**, y el atributo **href** con la ubicación del icono, el atributo **type** para especifica el formato, y el atributo **sizes** con el ancho y alto del icono:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
  <link rel="stylesheet" href="estilos.css">

<link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
</head>
<body>

</body>
</html>
```

<script>

Este elemento se utiliza para incorporar **códigos Javascript** desde **archivos externos** o en el propio **cuerpo** del documento:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
  <link rel="stylesheet" href="estilos.css">
  <script src="scripts.js"></script>
  <link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
```

```
</head>  
<body>  
  
</body>  
</html>
```

Cambia con respecto a la versión anterior, puesto que ya no es necesario indicar el tipo de lenguaje que se utilizará, que pasa a ser por defecto Javascript.

```
<script src="scripts.js" type="text/javascript"></script>
```

3. ESTRUCTURA DEL CUERPO <BODY>

Dentro de la estructura del cuerpo (el código entre las etiquetas <body>) se ubicarán los elementos que realmente serán visibles para el usuario del documento.

HTML ofrece diferentes formas de construir y organizar la información dentro del cuerpo de un documento. En los inicios <table> fue utilizado profusamente para estructurar el contenido, pero su uso decayó con el tiempo y fue entonces el elemento <div> el que comenzó a ser preponderante. Webs más interactivas y la integración de HTML, CSS y Javascript, generalizaron el uso de <div>.

Pero este elemento, así como <table>, no provee demasiada información acerca de la parte del cuerpo que está representando. **Cualquier cosa puede ir entre las etiquetas de apertura y cierre de un elemento <div>.** La palabra clave div solo especifica una división en el cuerpo, como la celda de una tabla, pero no ofrece indicio alguno sobre qué clase de división es, cuál es su propósito o qué contiene. **No tiene una connotación semántica.**

Para los usuarios, estas claves o indicios no son importantes, pero para los navegadores, la correcta interpretación del interior del documento que se está procesando puede ser crucial en muchos casos. Tras la revolución de los dispositivos móviles y el surgimiento de diferentes formas en que la gente accede a la web, **la identificación de cada parte del documento es una tarea que se ha vuelto muy relevante.**

HTML5 incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo. Cómo usamos estos nuevos elementos depende de nosotros, pero **las palabras clave otorgadas a cada uno de ellos ayudan a entender sus funciones.** Normalmente una página o aplicación web está dividida en varias áreas visuales para mejorar la experiencia del usuario y facilitar la interactividad. Las palabras claves que representan cada nuevo elemento de HTML5 están íntimamente relacionadas con estas áreas.

Todas las **etiquetas** que veremos a continuación crean un espacio en la página que llamaremos **cajas** (que en realidad son rectángulos **en la pantalla**). En primer lugar veremos las etiquetas más básicas que ya existían antes de HTML5 y en secciones ampliaremos con los nuevos elementos que ya introducen información semántica en el código. Algunas de las etiquetas más importantes son las siguientes:

Etiqueta	Función
<code><div> </div></code>	Crea una caja , por poder estructurar el contenido de la página.
<code><h1> Texto </h1></code>	Cabecera, título principal.
<code><h2> </h2>... <h6> </h6></code>	Cabeceras con medidas progresivamente menores .
<code><p> Párrafo de texto </p></code>	Párrafo , donde meter texto dentro de la página.
<code> texto </code>	Negrita (en algunos sitios se usa <code></code> però se desaconseja su uso)
<code> texto </code>	Cursiva (en algunos sitios se usa <code><i></code> però se desaconseja su uso)
<code><hr /></code>	Inserta una línea horizontal en la página.
<code></code>	Inserta la imagen foto.jpg dentro de la página. En este caso el fichero foto.jpg debe estar en la misma carpeta que el fichero .html
<code>
</code>	Inserta un salto de línea en la página.
<code><table> </table></code>	Indica la presencia de una tabla en la página.
<code><tr> </tr></code>	Inserta una fila en una tabla.
<code><td> </td></code>	Inserta una casilla dentro de la fila de una tabla.
<code> </code>	Lista ordenada
<code> </code>	Lista no ordenada
<code> </code>	Inserta un elemento dentro de una lista .
<code> Google </code>	Asigna un enlace al texto Google, que abre la página web www.google.es .
<code> Documento </code>	Asigna un enlace al texto Documento, que abre el documento que contiene el archivo document.pdf.
<code> Página web</code>	Asigna un enlace al texto Página web, que abre otro fichero .html que nosotros hemos creado.

4. ESTRUCTURA

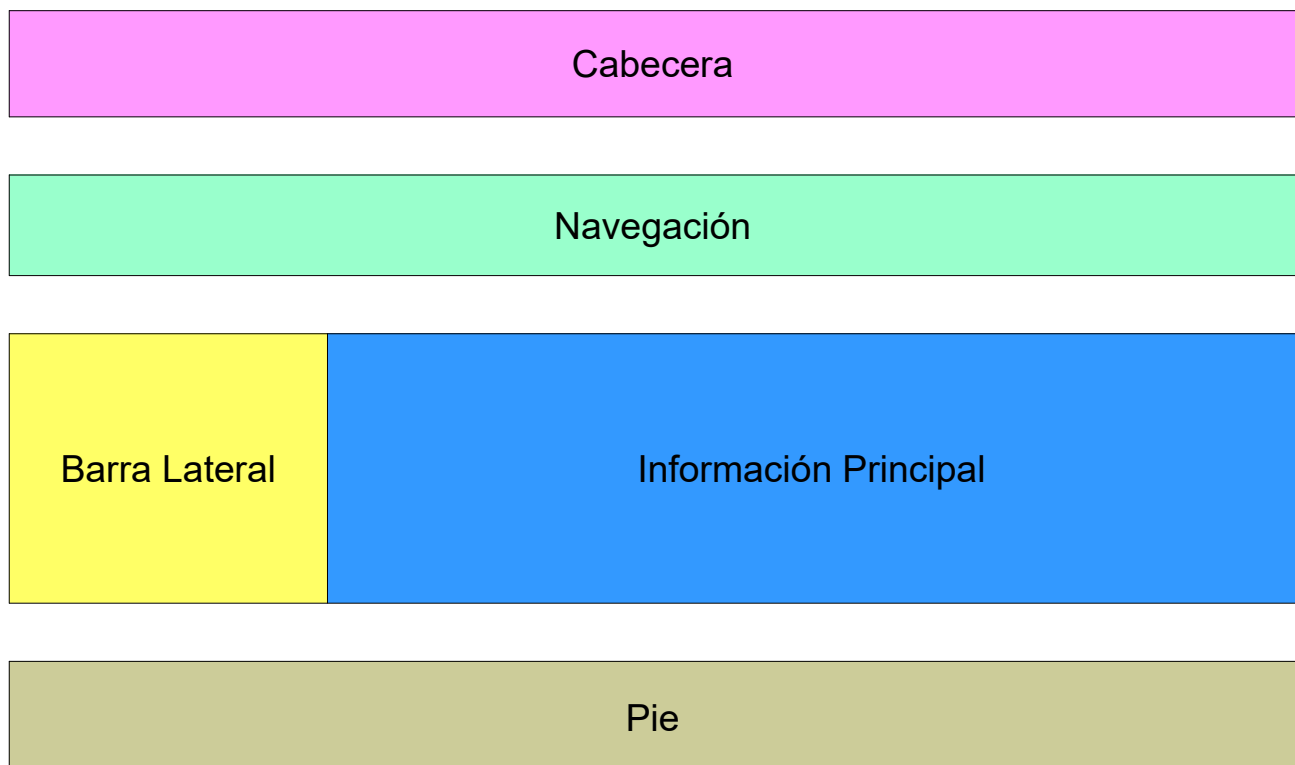
Conviene **estructurar una página web para organizar mejor la información** que queremos que se visualice en el navegador web. Toda etiqueta crea una caja, pero la etiqueta `<div>`, se utiliza exclusivamente por crear cajas de contenidos y es la forma más rudimentaria. Ejemplo:


```
<body>
  <div id="contenedor">
    <div id="cabecera">
      ...
    </div>
    <div id="principal">
      ...
    </div>
    <div id="pie">
      ...
    </div>
  </div>
</body>
```

En el archivo que contiene los estilos (.css) correspondientes a la página web (.html), se definirán, si es preciso, las características de cada una de las cajas definidas en la estructura.

Con HTML5 se incorporan nuevas cajas teniendo en cuenta como se estructuran la mayoría de sitios web.

La siguiente figura representa un diseño común encontrado en la mayoría de los sitios webs. Aunque cada diseñador crea sus propios diseños, en general podremos identificar las siguientes secciones en un sitio web:



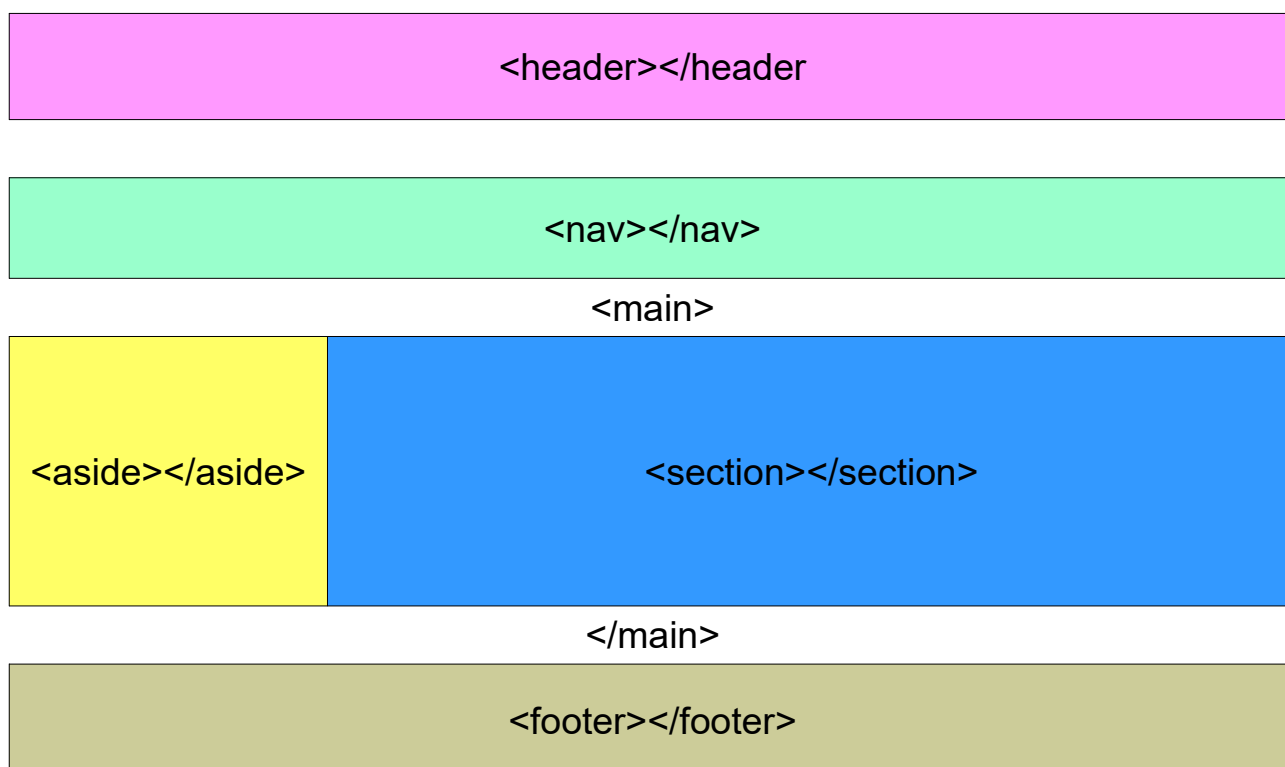
En la parte superior, descrito como **Cabecera**, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del sitio web o la página.

Inmediatamente debajo, podemos ver la Barra de **Navegación** en la que casi todos los desarrolladores ofrecen un menú o lista de enlaces con el propósito de facilitar la navegación a través del sitio.

El contenido más relevante de una página web se encuentra, en casi todo diseño, ubicado en su centro, **Información Principal**. Esta sección presenta información y enlaces valiosos. La mayoría de las veces se divide en varias **filas** y **columnas**. Esta sección es extremadamente flexible y normalmente diseñadores la adaptan de acuerdo con sus necesidades insertando más columnas, dividiendo cada columna en bloques más pequeños o generando diferentes distribuciones y combinaciones.

En la base de un diseño web clásico siempre nos encontramos con una barra más que aquí llamamos **Pie**. Es el área en donde normalmente se muestra información acerca del sitio web, el autor o la empresa, además de algunos enlaces con respecto a reglas, términos y condiciones y toda información adicional que el desarrollador considere importante compartir.

HTML5 considera esta estructura básica y provee **nuevos elementos** para diferenciar y declarar cada una de sus partes. A partir de ahora podemos decir al navegador para qué es cada sección:



Veamos detenidamente cada uno de estos elementos.

<header>

El elemento `<header>` no debe ser confundido con `<head>` usado para construir la

cabecera del documento. Del mismo modo que <head>, la intención de <header> es proveer **información introductoria** (títulos, subtítulos, logos), pero difiere con respecto a <head> en su alcance. Mientras que el elemento <head> tiene el propósito de proveer información acerca de todo el documento, <header> es usado solo para el cuerpo o secciones específicas dentro del cuerpo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
  <link rel="stylesheet" href="estilos.css">
  <link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
  <script src="scripts.js"></script>
</head>
<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
</body>
</html>
```

<nav>

En el ejemplo de disposición de contenido en el interfaz que se ha propuesto, la siguiente sección es la **Barra de Navegación**, generada en HTML5 con el elemento <nav>:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
  <link rel="stylesheet" href="estilos.css">
  <script src="scripts.js"></script>
</head>
<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
```

</body>

</html>

El elemento <nav> se encuentra dentro de las etiquetas <body> pero es ubicado después de la etiqueta de cierre de la cabecera (</header>), no dentro de las etiquetas <header>. Esto es así porque **<nav> no es parte de la cabecera sino una nueva sección.**

La estructura y el orden que elegimos para colocar los elementos HTML5 dependen de nosotros. HTML5 es versátil y simplemente nos proporciona los parámetros y elementos básicos con los que trabajar, pero cómo usarlos será exclusivamente decisión nuestra.

Un ejemplo de esta versatilidad es que el elemento **<nav>** podría ser insertado dentro del elemento <header> o en **cualquier otra parte del cuerpo**. Sin embargo, siempre se debe considerar que estas etiquetas fueron creadas para brindar información a los navegadores y ayudar a cada nuevo programa y dispositivo en el mercado a identificar las partes más relevantes del documento.

Para conservar nuestro código portable y comprensible, es recomendable como buena práctica seguir lo que marcan los estándares y mantener todo tan claro como sea posible. El elemento **<nav>** fue creado para ofrecer ayuda para la navegación, como en **menús principales** o **grandes bloques de enlaces**, y debería ser utilizado de esa manera.

<main>

Con esta nueva etiqueta se pretende englobar toda aquella información que es considerada parte de la información principal de la página. Es decir, el contenido principal de esa página y resto de enlaces y de pequeñas reseñas que enlazan con contenidos principales.

<section>

Siguiendo el diseño estándar propuesto, nos encontramos con las columnas que llamamos **Información Principal** y Barra Lateral. La columna Información Principal contiene la información más relevante del documento y puede ser encontrada en diferentes formas (por ejemplo, dividida en varios bloques o columnas). Debido a que el propósito de estas columnas es más general, el elemento en HTML5 que especifica estas secciones se llama simplemente <section>:

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="iso-8859-1">

<meta name="description" content="Introducción HTML5">

<meta name="keywords" content="HTML5, CSS3, IES Pere Maria">

<title>Lenguajes de Marcas</title>

<link rel="stylesheet" href="estilos.css">

<script src="scripts.js"></script>

<link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">

```
</head>

<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <main>
    <section>

    </section>
  </main>
</body>
</html>
```

Al igual que la Barra de Navegación, la columna **Información Principal es una sección aparte**. Por este motivo, la sección para Información Principal va debajo de la etiqueta de cierre </nav>.

<aside>

En un típico diseño web la columna llamada **Barra Lateral** se ubica al lado de la columna Información Principal. Esta es una columna o sección que normalmente contiene **datos relacionados con la información principal** pero que no son relevantes o igual de importantes.

En el diseño de un blog, por ejemplo, la Barra Lateral contendrá una lista de enlaces. En el ejemplo siguiente, los enlaces apuntan a cada una de las entradas del blog y ofrecen información adicional sobre el autor. La información dentro de esta barra está relacionada con la información principal pero no es relevante por sí misma.

Siguiendo el mismo ejemplo podemos decir que las entradas del blog son relevantes pero los enlaces y las pequeñas reseñas sobre esas entradas son solo una ayuda para la navegación pero no lo que al lector realmente le interesa.

En HTML5 podemos diferenciar esta **clase secundaria de información** utilizando el elemento <aside>:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
```

```
<meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
<title>Lenguajes de Marcas</title>
<link rel="stylesheet" href="estilos.css">
<link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
<script src="scripts.js"></script>
</head>
<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <main>
    <section>

    </section>

    <aside>
      <blockquote>Mensaje número uno</blockquote>
      <blockquote>Mensaje número dos</blockquote>
    </aside>
  </main>
</body>
</html>
```

El elemento <aside> podría estar ubicado del lado derecho o izquierdo de nuestra página de ejemplo, la etiqueta **no tiene una posición predefinida**. El elemento <aside> solo describe la información que contiene, no el lugar dentro de la estructura. Esto se determinará después en el fichero de estilos.

Este elemento **puede estar ubicado en cualquier parte del diseño** y ser usado siempre y cuando su contenido no sea considerado como el contenido principal del documento. Por ejemplo, podemos usar <aside> dentro del elemento <section> o incluso insertado entre la información relevante, como en el caso de una cita.

<footer>

Para finalizar la construcción de la plantilla o estructura elemental de nuestro documento HTML5, lo único que nos queda por hacer es cerrar nuestro diseño para otorgarle un **final al cuerpo del documento**. HTML5 provee un elemento específico para este propósito llamado <footer>:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
```

```
<meta name="description" content="Introducción HTML5">
<meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
<title>Lenguajes de Marcas</title>
<link rel="stylesheet" href="estilos.css">
<link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
<script src="scripts.js"></script>
</head>
<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <main>
    <section>

  </section>
  <aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
  </aside>
</main>
<footer>
  IES Pere Maria Orts i Bosch. <br> Benidorm.
  &copy; Derechos Reservados 2018
</footer>
</body>
</html>
```

En el típico diseño de una página web la sección llamada **Pie** será definida por etiquetas <footer>. Esto es debido a que la barra representa el final (o pie) del documento y esta parte de la página web es normalmente usada para compartir **información general sobre el autor o la organización, como derechos de autor, términos y condiciones de uso**.

Generalmente, el elemento <footer> representará el final del cuerpo de nuestro documento. Sin embargo, <footer> **puede ser usado múltiples veces dentro del cuerpo** para representar también el final de diferentes secciones (del mismo modo que la etiqueta <header>).

5. ELEMENTOS DENTRO DEL CUERPO

La estructura básica de nuestro sitio web fue finalizada, pero aún debe introducirse el **contenido**. Los elementos HTML5 estudiados hasta el momento nos ayudan a identificar cada sección del diseño y asignar un propósito intrínseco a cada una de ellas,

pero lo que es realmente importante para nuestro sitio web se encuentra en el interior de estas secciones.

Ninguno de los nuevos elementos de HTML5 vistos hasta el momento declara algo acerca del contenido. Todos tienen un específico propósito estructural para que dicha estructura del documento HTML pueda ser identificada y reconocida por los navegadores y nuevos dispositivos.

El contenido estará compuesto por diferentes elementos visuales como títulos, textos, imágenes, videos y aplicaciones interactivas, entre otros. Necesitamos poder diferenciar estos elementos y establecer una relación entre ellos dentro de la estructura.

<article>

Del mismo modo que los blogs están divididos en entradas, los sitios web normalmente presentan información relevante dividida en **partes que comparten similares características**. El elemento <article> nos permite identificar cada una de estas partes:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
  <link rel="stylesheet" href="estilos.css">
  <link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
  <script src="scripts.js"></script>
</head>
```

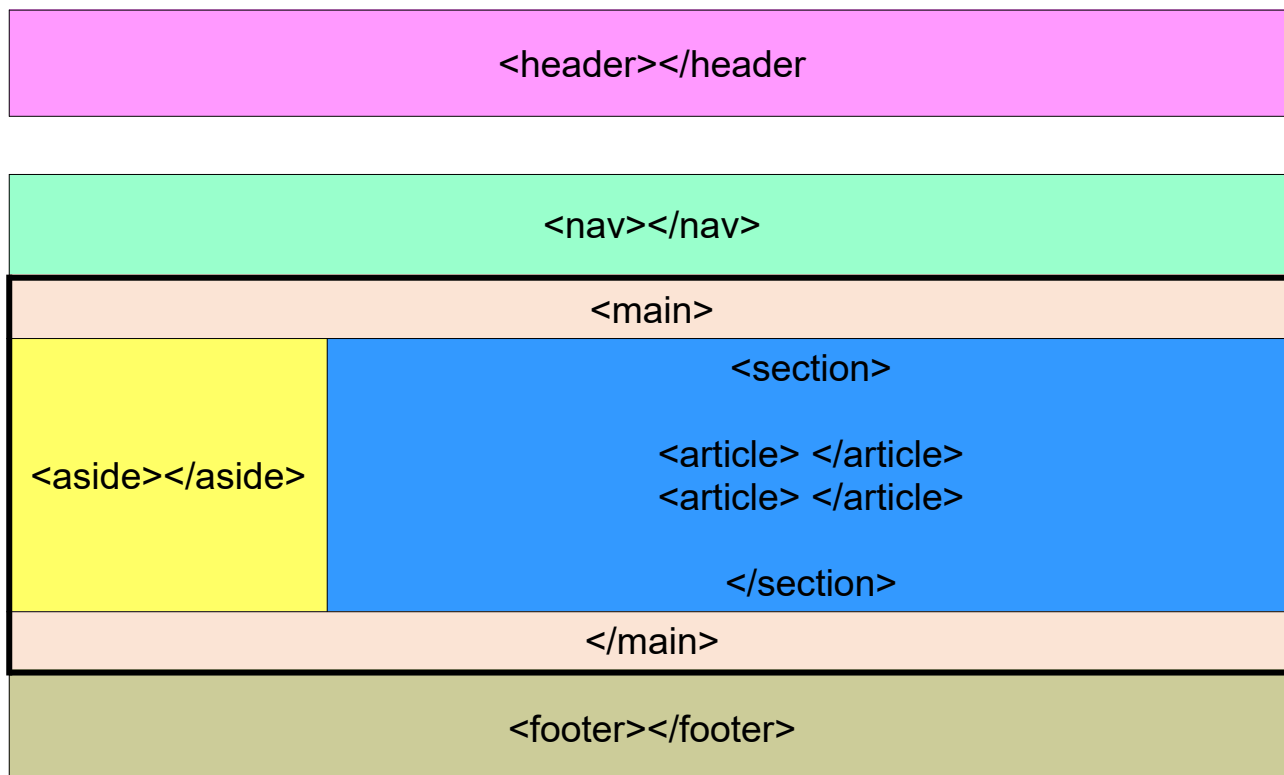
```
<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <main>
    <section>
      <article>
        Texto del mensaje I
      </article>
      <article>
```


Texto del mensaje II

```
</article>
</section>
<aside>
  <blockquote>Mensaje número uno</blockquote>
  <blockquote>Mensaje número dos</blockquote>
</aside>
</main>
<footer>
  IES Pere Maria Orts i Bosch. <br> Benidorm.
  &copy; Derechos Reservados 2018
</footer>
</body>
</html>
```

Las etiquetas `<article>` se encuentran ubicadas **dentro del elemento `<section>`**. Las etiquetas `<article>` en nuestro ejemplo pertenecen a esta sección, son sus hijos, del mismo modo que cada elemento dentro de las etiquetas `<body>` es hijo del cuerpo. Y al igual que cada elemento hijo del cuerpo, las etiquetas `<article>` son ubicadas una sobre otra.

El elemento `<article>` no está limitado por su nombre (**no se limita, por ejemplo, a artículos de noticias**). Este elemento fue creado con la intención de contener **unidades independientes de contenido**, por lo que puede incluir mensajes de foros, artículos de una revista digital, entradas de blog, comentarios de usuarios, etc. Lo que hace es **agrupar porciones de información que están relacionadas** entre sí independientemente de su naturaleza.



Como una parte independiente del documento, el contenido de **cada elemento <article>** tendrá su propia estructura.

Para definir esta estructura, podemos aprovechar la versatilidad de los elementos **<header>** y **<footer>** estudiados anteriormente. Estos elementos son portables y pueden ser usados no solo para definir los límites del cuerpo sino también en cualquier sección de nuestro documento:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="iso-8859-1">
  <meta name="description" content="Introducción HTML5">
  <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
  <title>Lenguajes de Marcas</title>
  <link rel="stylesheet" href="estilos.css">
  <link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
  <script src="scripts.js"></script>
</head>
<body>
  <header>
    <h1>DAW: Lenguajes de Marcas</h1>
  </header>
```

```
<nav>
  <ul>
    <li>principal</li>
    <li>fotos</li>
    <li>videos</li>
    <li>contacto</li>
  </ul>
</nav>
<main>
  <section>
    <article>
      <header>
        <h1>Título del mensaje I</h1>
        <h2>Subtítulo del mensaje I</h2>
        <p>publicado 20-09-2018</p>
      </header>
      Texto del mensaje I
      <footer>
        <p>comentarios (0)</p>
      </footer>
    </article>
    <article>
      <header>
        <h1>Titulo del mensaje II</h1>
        <h2>Subtítulo del mensaje II</h2>
        <p>publicado 23-09-2018</p>
```

```
</header>
    Texto del mensaje II
<footer>
    <p>comentarios (0)</p>
</footer>
</article>
</section>
<aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
</aside>
</main>
<footer>
    IES Pere Maria Orts i Bosch. <br> Benidorm.
    &copy; Derechos Reservados 2018
</footer>
</body>
</html>
```

Los dos mensajes insertados en el código fueron contruidos con el elemento `<article>` y tienen una estructura específica. En la parte superior de esta estructura incluimos las etiquetas `<header>` conteniendo el título definido con los elementos `<h1>`, `<h2>` y `<p>`, debajo se encuentra el contenido mismo del mensaje y, después del texto, vienen las etiquetas `<footer>` especificando la cantidad de comentarios recibidos.

<figure> y <figcaption>

La etiqueta `<figure>` fue creada para ser aún más específicos a la hora de declarar el contenido del documento. Antes de que este elemento sea introducido, no podíamos identificar el contenido que era parte de la información, pero a la vez independiente, como **ilustraciones, fotos, vídeos, etc.** Normalmente estos elementos son parte del contenido relevante pero **pueden ser extraídos o movidos a otra parte sin afectar o interrumpir el flujo del documento**. Cuando nos encontramos con esta clase de información, las etiquetas `<figure>` pueden ser usadas para identificarla:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="iso-8859-1">
    <meta name="description" content="Introducción HTML5">
    <meta name="keywords" content="HTML5, CSS3, IES Pere Maria">
    <title>Lenguajes de Marcas</title>
    <link rel="stylesheet" href="estilos.css">
    <link rel="icon" href="img/logo.png" type="image/png" sizes="16x16">
    <script src="scripts.js"></script>
</head>
<body>
    <header>
        <h1>DAW: Lenguajes de Marcas</h1>
    </header>
```

```
<nav>
  <ul>
    <li>principal</li>
    <li>fotos</li>
    <li>videos</li>
    <li>contacto</li>
  </ul>
</nav>
<main>
  <section>
    <article>
      <header>
        <h1>Título del mensaje I</h1>
        <h2>Subtítulo del mensaje I</h2>
        <p>publicado 20-09-2018</p>
      </header>
      Texto del mensaje I
      <figure>
        
        <figcaption>
          Esta es la imagen del mensaje I
        </figcaption>
      </figure>
      <footer>
        <p>comentarios (0)</p>
      </footer>
    </article>
    <article>
      <header>
        <h1>Título del mensaje II</h1>
        <h2>Subtítulo del mensaje II</h2>
        <p>publicado 23-09-2018</p>
      </header>
      Texto del mensaje II
      <footer>
        <p>comentarios (0)</p>
      </footer>
    </article>
  </section>
  <aside>
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
  </aside>
</main>
<footer>
  IES Pere Maria Orts i Bosch. <br> Benidorm.
  &copy; Derechos Reservados 2018
</footer>
</body>
</html>
```

En el primer mensaje del código anterior, después del texto se inserta una imagen (****). A menudo el texto es enriquecido con imágenes o

videos. Las etiquetas **<figure>** nos permiten envolver estos complementos visuales y **diferenciarlos así de la información más relevante**.

Se puede observar también un elemento extra dentro de **<figure>**. Normalmente, unidades de información como imágenes o videos son descritas con un corto texto debajo. HTML5 provee un elemento para ubicar e identificar estos **títulos descriptivos**. Las etiquetas **<figcaption>** encierran el **texto relacionado con <figure>** y establecen una relación entre ambos elementos y su contenido.

6.NUEVOS ELEMENTOS INCORPORADOS A HTML5

HTML5 fue desarrollado con la intención de **simplificar, especificar y organizar el código**. Para lograr este propósito, nuevas etiquetas y atributos fueron agregados y HTML fue completamente integrado a CSS y Javascript. Estas incorporaciones y mejoras de versiones previas están relacionadas no solo con nuevos elementos sino también con cómo usamos los ya existentes.

<mark>

La etiqueta **<mark>** fue agregada para **resaltar parte de un texto** que originalmente no era considerado importante pero ahora es **relevante acorde con las acciones del usuario**. El ejemplo que más se ajusta a este caso es un **resultado de búsqueda**. El elemento **<mark>** resaltará la parte del texto que concuerda con el texto buscado:

`El <mark>ordenador</mark> está ocupado`

Si un usuario realiza una búsqueda de la palabra “ordenador”, por ejemplo, los resultados podrían ser mostrados con el código anterior. La frase del ejemplo representa los resultados de la búsqueda y **las etiquetas <mark> en el medio encierran lo que era el texto buscado** (la palabra “ordenador”). En algunos navegadores, esta palabra será resaltada con un fondo amarillo por defecto, pero siempre podemos sobrescribir estos estilos con los nuestros utilizando CSS.

En el pasado, normalmente obteníamos similares resultados usando el elemento ****. El agregado de **<mark>** tiene el objetivo de cambiar el significado y otorgar un nuevo propósito para éstos y otros elementos relacionados:

- **** es para indicar **énfasis** (reemplazando la etiqueta **<i>** que utilizábamos anteriormente).
- **** es para indicar **importancia**.
- **<mark>** es para **resaltar** texto que es relevante de acuerdo con las **circunstancias**.
- **** debería ser usado **solo cuando no hay otro elemento más apropiado** para la situación.

<small>

Este elemento era utilizado anteriormente con la intención de presentar cualquier texto con letra pequeña. La palabra clave hacía referencia al tamaño del texto,

independientemente de su significado. En HTML5, el nuevo propósito de <small> es **presentar la llamada letra pequeña**, como impresiones legales, descargos, etc.

<small>Derechos Reservados © 2018 IES Pere Maria Orts</small>

<cite>

Otro elemento que ha cambiado su naturaleza para volverse más específico es <cite>. Ahora las etiquetas <cite> **encierran el título de un trabajo**, como un libro, una película, una canción, etc.

Los contenidos del libro <cite>HTML5 CSS3 for the Real World</cite>

<address>

El elemento <address> es un viejo elemento convertido en un elemento estructural. No necesitamos usarlo previamente para construir nuestra plantilla, sin embargo podría ubicarse perfectamente en algunas situaciones en las que debemos presentar **información de contacto** relacionada con el contenido del elemento <article> o el cuerpo completo.

Este elemento debería ser incluido dentro de <footer>, como en el siguiente ejemplo:

```
<article>
<header>
  <h1>Título del mensaje </h1>
</header>
  Este es el texto del mensaje
<footer>
  <address>
    <a href="http://iesperemaria.edu.gva.es/iesperemaria/">IES Pere Maria Orts</a>
  </address>
</footer>
</article>
```

<time>

En cada <article> de la última plantilla, se incluye la fecha indicando cuándo se publicó el mensaje. Para esto usamos un simple elemento <p> dentro de la cabecera (<header>) del mensaje, pero existe un elemento en HTML5 específico para este propósito. El elemento <time> nos permite declarar un texto comprensible para humanos y navegadores que representa **fecha y hora**:

```
<article>
<header>
  <h1>Título del mensaje II</h1>
```

<time datetime="2018-09-23" pubdate>publicado 23-09-2018</time>

</header>

Texto del mensaje II

</article>

Ahora, el elemento <p> usado en ejemplos previos se ha reemplazado por el nuevo elemento <time> para mostrar la fecha en la que el mensaje fue publicado. El atributo **datetime** para especificar un format de **fecha comprensible para el navegador** (timestamp).

El formato de este valor deberá seguir un **patrón** similar al del siguiente ejemplo: **2015-09-23T12:43:02**. También incluimos el atributo **pubdate**, el cual solo es agregado para indicar que el valor del atributo **datetime** representa la fecha de publicación.

<code>

Este elemento representa código de programación. Se usa en conjunción con el elemento <pre> para presentar código de programación en el formato original.

<article>

<header>

<h1>Título del mensaje II</h1>

<time datetime="2018-09-23" pubdate>publicado 23-09-2018</time>

</header>

Texto del mensaje II

<pre>

<code>

```
function cambiarColor() {  
    document.body.style.backgroundColor = "#0000FF";  
}
```

```
document.addEventListener("click", cambiarColor);
```

</code>

</pre>

</article>

<data>

Este elemento representa datos genéricos. Puede incluir el atributo **value** para especificar el valor en formato ordenador. **Ejemplo:** <data value="32">Treinta y Dos</data>).

7. ESTILOS

Este tema se ampliará posteriormente aunque conviene mencionarlo para tener

una visión más global del funcionamiento de las webs. Cambiar el aspecto de la página web se puede hacer de varias maneras, nosotros lo haremos introduciendo los estilos en otro fichero con extensión **.css**. También se puede hacer indicando los estilos dentro de cada etiqueta o a el apartado **head** de la página.

Para **relacionar un fichero html con el correspondiente fichero .css**, se utiliza una línea en la sección de cabecera **<head>**, con la etiqueta **<link>** como la siguiente:

```
<link rel="stylesheet" href="estilos.css" type="text/css" />
```

Donde se indica:

- El nombre y la ubicación del fichero **.css** con el parámetro **href** (en este caso **"estilos.css"**, en la misma carpeta que el fichero html)
- Qué tipo de relación tendrá este fichero con el archivo html con el parámetro **rel** (en este caso hoja de estilos, **"stylesheet"**)
- De qué tipo de fichero se trata con el parámetro **type** (un fichero de texto que contiene código css, **"text/css"**).

Una vez indicado donde se definirán los estilos de la página, hay tres maneras para asociar el aspecto con las etiquetas del fichero html:

- **id**: identificador **único** en toda la página web.

En el archivo .html	En el archivo .css
<pre><div id="cabecera"> <h1> Cabecera de la página </h1> <p> párrafo de texto de la página </p> </div></pre>	<pre>#cabecera { background-color:#0000FF; color:#FFFFFF; }</pre>

- **class**: conjunto de características que se pueden asignar a varios elementos de una página:

En el archivo .html	En el archivo .css
<pre><p class="claseParrafo"> párrafo de texto con el aspecto definido en la clase claseParrafo definido en un archivo .css </p></pre>	<pre>.claseParrafo{ background-color:#00FF00; color:#FFFF00; }</pre>

- Introduciendo directamente el tipo de etiqueta que queremos cambiar en todo el documento:

En el archivo .html	En el archivo .css
<pre><h1> Título </h1></pre>	<pre>h1 { background-color:#FF0000;</pre>


```
color:#FF00FF;  
border-style:solid;  
border-width:1px;  
border-color:#FF00FF;  
}
```

8. COMENTARIOS

Cuando los ficheros se hacen grandes, es difícil saber exactamente a que corresponde cada parte del código escrito. Para hacer más legibles y comprensibles los ficheros tenemos la posibilidad de escribir comentarios **explicativos** del contenido, que **no afectarán a la presentación de la página en el navegador web**.

- En los ficheros **.html** los comentarios se sitúan entre los símbolos **<!--** y **-->**

Ejemplo:

<!--Comentario en un .html: esta línea no aparece en el navegador Web -->

- En los ficheros **.css** los comentarios se sitúan entre los símbolos **/*** y ***/**

Ejemplo:

/* Comentario en un fichero de estilos */

9. VALIDACIÓN

Para empezar a dotar de estilo a la página html, es conveniente comprobar anteriormente que está correctamente **formada** mediante el uso de un validador, una aplicación que identifica los errores cometidos al crear el fichero (faltan etiquetas, el nombre de la etiqueta o el parámetro es incorrecto, etc.). Por ejemplo el de la página del w3c.

<http://validator.w3.org>

10. CARACTERES ESPECIALES

Las páginas web pueden conllevar problemas de todo tipo con los caracteres **especiales**. Suelen escribirse en el código HTML de una forma especial: empiezan por &, seguido de un nombre establecido y acaban con un punto y coma.

Carácter	En el código	Inglés
<	<	Less Than
>	>	Greater Than
&	&	Ampersand
"	"	Quotation Marks

€	€	Euro
.	·	Middle Dote
Espacio en blanco protegido	 	Non Breakable Space

También hay representaciones para caracteres con acentos **y otros signos** de puntuación:

Carácter	En el código
Á	Á
à	à
ñ	ñ
Ü	Ü
ü	ü

Podemos encontrar listas más completas en Internet.

11. EJEMPLOS DE USO DE ETIQUETAS

Ejemplo Lista Ordenada:

1. Elemento 1
2. Elemento 2
3. Elemento 3
4. Elemento 4

```
<ol>
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
<li>Element 4</li>
</ol>
```

Ejemplo Lista No Ordenada:

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

```
<ul>
<li>Element 1</li>
<li>Element 2</li>
<li>Element 3</li>
<li>Element 4</li>
</ul>
```

Ejemplo tabla:

A	B
C	D

```
<table>
  <tr>
    <td> A </td>
    <td> B </td>
  </tr>
  <tr>
    <td> C </td>
    <td> D </td>
  </tr>
</table>
```