

# HTML5

Antonio Espín Herranz

# Contenidos

- Introducción.
- Estructura global.
- Estructura del cuerpo.
- Nuevos y viejos elementos.
- Formularios Web.
- Audio, Video y Canvas.

# Introducción

- **Html5:** Se considera la unión de HTML, CSS y JavaScript.
- **HTML:** se centra en la estructura del documento.
- **CSS:** Presenta la estructura y el contenido en la pantalla.
- **JS:** Aporta dinamismo.

# Estructura global

- **<!DOCTYPE>**
  - Indica al navegador el tipo del documento.
  - Ayuda al renderizado de la página.
  - `<!DOCTYPE html>`
  - Se simplifica mucho el contenido de la etiqueta doctype.
- **<html>**
  - Sólo dispone del atributo lang, para indicar el idioma del documento.
  - Utilizar los códigos de cada idioma: es, en, de, fr, etc.
  - `<html lang="es">`

# Estructura global

- **<head>**
  - Mismo significado que en versiones anteriores.
  - Dentro de head se especifican archivos adjuntos, etiquetas meta.
  - Códigos de codificación de caracteres: normalmente iso-8859-1.
  - Con las etiquetas meta se indica información necesaria para los buscadores, pero no visible para la página. *No es obligatorio cerrarlas pero es aconsejable por compatibilidad.*
  - **<head>**
    - `<meta charset="iso-8859-1">`
    - `<meta name="description" content="contenido del sitio">`
    - `<meta name="keywords" content="lista palabras clave">`
      - Se separan por comas.

# Estructura global

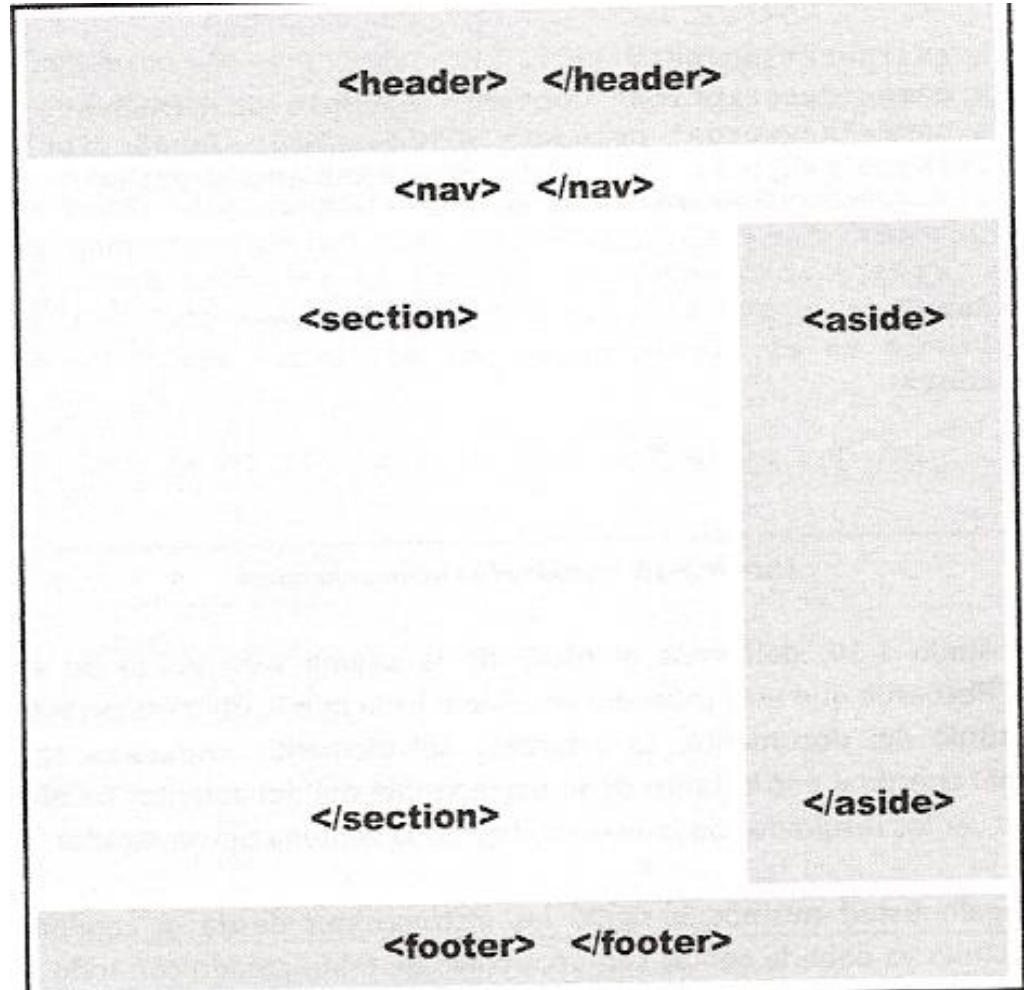
- **<head>**
  - Mas elementos que podemos encontrar:
  - **<title>** idem de versiones anteriores.
  - **<link>** Se simplifica para adjuntar ficheros de estilos.
    - `<link rel="stylesheet" href="fichero.css">`
      - Desaparece el atributo type.
- **<body>**
  - Mismo significado, pero agrega una estructura.

# Estructura del cuerpo

- Dentro del elemento body, se agregan nuevas secciones:
  - <header>
  - <nav>
  - <section>
  - <aside>
  - <footer>
- La colocación de estas partes del body se realiza mediante estilos css.

# Estructura del cuerpo

```
<body>  
  <header>  
  </header>  
  
  <nav>  
  </nav>  
  
  <section>  
  </section>  
  
  <aside>  
  </aside>  
  
  <footer>  
  </footer>  
</body>
```





# Estructura del cuerpo

- **<header>**

- Representa la cabecera y está destinado a la colocación de encabezados: <h1> ... <h6>, también para logos, imágenes.

- <body>

- <header><h1>Titulo de la página</h1></header>

- **<nav>**

- Representa la barra de navegación.

- Dentro de este podemos utilizar listas de enlaces.

# Estructura del cuerpo

- **<section>**
  - El elemento central de la página.
  - Información de interés.
  - Sección principal.
- **<aside>**
  - Representa la barra lateral de navegación o información para seleccionar contenidos a mostrar en la section.
  - Puede ubicarse al lado derecho o izquierdo de la section.
  - No tiene ubicación predefinida.

# Estructura del cuerpo

- **<footer>**

- Información de tipo institucional.
- El copyright, nombre de la empresa, enlaces de contacto, toda la información relevante a mostrar.

<footer>

Derechos reservados &copy; 2010-2011

</footer>

# Dentro del cuerpo

- Dentro de la sección **section**, la información a mostrar se puede organizar en **article**.

<section>

<article>

Primer artículo ...

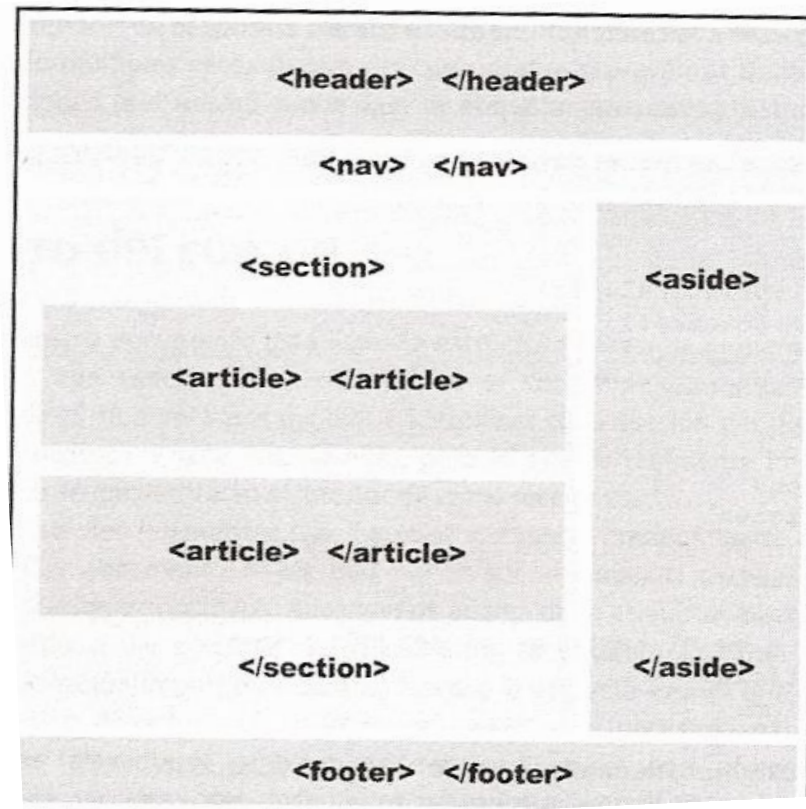
</article>

<article>

Segundo artículo ...

<article>

</section>



# Dentro del cuerpo

- Dentro de un article podemos tener una estructura formada por header y footer.
- Estas dos etiquetas se pueden usar para la estructura del documento y otros elementos como la etiqueta article.

```
<article>
  <header>
    <h1>Título del artículo</h1>
  </header>
  Contenido del primer artículo
  <footer>
    <p>Comentarios(0)</p>
  </footer>
</article>
```

# Dentro del cuerpo

- **<hgroup>**
  - Cuando dentro de una etiqueta header colocamos varios tipos de encabezados, por ejemplo: un título y un subtítulo.
  - En este caso tenemos que utilizar una etiqueta hgroup.

**<header>**

**<hgroup>**

**<h1>Título del mensaje</h1>**

**<h2>Subtítulo del mensaje</h2>**

**</hgroup>**

**</header>**

# Dentro del cuerpo

- **<figure>** y **<figcaption>**
    - Con figure podemos definir el contenido del documento para hacer referencia a imágenes.
- ```
<article>  
  <header>...</header>  
  Contenido en texto del primer artículo  
  <figure>  
      
    <figcaption>Una descripción de la imagen</figcaption>  
  </figure>  
  <footer>  
    <p>Comentarios (0)</p>  
  </footer>  
</article>
```
- Con **figcaption** podemos poner una descripción a la imagen del primer artículo.

# Nuevos y viejos elementos

- **<mark>**
  - Esta etiqueta es para resaltar parte de un texto que originalmente no era importante pero ahora lo es por las acciones del usuario.
  - `<span>Curso de <mark>HTML5</mark></span>`
  - En algunos navegadores esta palabra se marca con fondo amarillo.
  - Otras etiquetas parecidas pueden ser:
    - `<em>` Para dar énfasis.
    - `<strong>` es para dar importancia.
- **<small>**
  - El nuevo propósito de small es presentar la llamada letra pequeña: impresiones legales, etc.
  - `<small>Todos los derechos reservados &copy; 2012</small>`



# Nuevos y viejos elementos

- **<cite>**

- Ahora se utiliza para indicar el título de un libro, de una película, una canción.
- `<span>`
  - Consulta el libro `<cite>Programación Java</cite>`
- `</span>`

- **<address>**

- Se utiliza para presentar información de contacto relacionada con el contenido `<article>` o el cuerpo completo.

`<article>`

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

Contenido del mensaje

`<footer>`

`<address>`

`<a href="http:// ..." >Contacto</a>`

`</address>`

`</footer>`

`</article>`

# Nuevos y viejos elementos

- **<time>**

- Representa una fecha comprensible para humanos y navegadores.
- El atributo `datetime` es para el navegador.

`<article>`

`<header>`

`<h1>Titulo del mensaje</h1>`

`<time datetime="2011-10-12" pubdate>Publicado 12-10-2012</time>`

`</header>`

Este es el contenido del mensaje

`</article>`

- El atributo **pubdate** indica que el valor del atributo `datetime` coincide con la fecha de publicación.

# Nuevos y viejos elementos

- **Ejemplos de time:**

`<time datetime="2012-10-08T12:00">12h del 8 de  
octubre de 2012</time>`

`<time datetime="12:00">Las 12 horas</time>`

`<time datetime="2012">Año 2012</time>`

`<time>8 de octubre de 2012</time>`

# Nuevos y viejos elementos

- En HTML5 podemos implementar duraciones. Utilizando el prefijo P para períodos, D para días, H para horas, M para minutos, y S para segundos.
  - `<time datetime="P4D">` es una duración de 4 días.
- Si usamos una T despues de P nos permite todavía ser mas precisos:
  - `<time datetime="PT23H 9M 2.343S">`
    - Es una duración de 23 horas, 9 minutos y 2343 segundos.

# Formularios Web

- El elemento **form** no cambia mucho, añade los siguientes atributos:
  - **autocomplete**: valores posibles on / off. Con el valor off los input tendrán la función autocompletar desactivada.
  - **novalidate**: Por defecto los formularios son automáticamente validados. Se puede desactivar con este atributo:
    - `<form name="miFormulario" novalidate="novalidate">`

# input

- *En HTML5 no es necesario cerrar la etiqueta input pero se aconseja por compatibilidad entre navegadores.*
- **DATOS DE IDENTIFICACIÓN:**
  - Una interesante novedad en HTML5 es la ampliación de la etiqueta *input* para simplificar el ingreso y la validación de variados elementos de identificación y localización. Los nuevos tipos de input relacionados con esta característica son:
  - **tel**, para el ingreso de números de teléfono.
    - `<input type="tel" name="mitelfono">`
  - **url**, para ingresar una dirección web. **Validado**
    - `<input type="url" name="url">`
  - **email**, para la dirección de correo electrónico: **Validado**
    - `<input type="email" name="email">`

# input

- **Fechas y Horas:** (depende del navegador nos puede construir un calendario o no).
  - Soporte para introducir:
    - Date:** `<input type="date" id="fecha" name="fecha" />`
    - Week: `<input type="week" id="fecha2" name="fecha2" />`
    - Month: `<input type="month" id="fecha3" name="fecha3" />`
    - Time:** `<input type="time" id="fecha4" name="fecha4" />`
    - Datetime: `<input type="datetime" id="fecha5" name="fecha5" />`
    - Datetime-local: `<input type="datetime-local" id="fecha6" name="fecha6" />`

# input

- date : selecciona la fecha completa, día-mes-año.
- datetime: selecciona la hora completa más el día, mes, año y zona horaria (timezone).
- datetime- local: incluye la hora actual, día, mes y año (no incluye zona horaria).
- time: selecciona la hora actual en horas y minutos.
- month: selecciona el mes y el año.
- week : indica la semana (un número de 1 a 52) y el año.



# input

- Trabajo con números (sólo en algunos navegadores):
  - **Range**: dibuja en pantalla un control y permite al usuario seleccionar un valor a partir de un deslizador.
    - Att:
      - Min, max, step: valores mínimos, máximos y el paso.
      - `<input type="range" name="deslizador" min="1" max="100" step="1">`
  - **Number**: Para introducir números, permite al usuario seleccionar los valores. Mediante dos botones:
    - `<input type="number" name="numero" id="numero" min="5" max="100" step="5" />`
    - Mismos atributos que el anterior.

# input

- **search:**

- La etiqueta *search* no es más que un típico campo de input de texto, sólo que con un formato similar al muy conocido de búsqueda de contenido. Algunos navegadores agregan a la derecha del casillero una “x” para eliminar el texto ingresado.
- `<input type="search" name="buscador">`

# input

- **Color:**
  - Seleccionar un color. Se espera un número hexadecimal del estilo: #00FF90.
  - Algunos navegadores muestran un desplegable para seleccionar el color.
  - `<input type="color" name="color" id="color" />`

# Nuevos atributos

- **Autofocus** fuerza el foco (la posición del cursor) dentro de un campo, una vez que se carga la página.
  - `<input type="tel" id="telefono" name="telefono" autofocus="autofocus" />`
- **Autocomplete** permite el rellenado automático del input en base al texto que vaya escribiendo el usuario, en base a los textos introducidos anteriormente. Puede utilizarse como atributo de un campo de entrada o del formulario completo.
  - `<input type="email" id="correo" name="correo" autocomplete="on" />`
- **Placeholder** muestra un texto por defecto en el input hasta que este tiene el foco, en ese momento el texto desaparece para que el usuario ingrese el texto que desee.
  - `<input type="url" id="url" name="url" placeholder="http://" />`
- **Required** es un atributo que se aplica a un campo de entrada e impide que se envíe el formulario si el campo no se a rellenado.
  - `<input type="text" id="nombre" name="nombre" required="required" />`

# Nuevos atributos

- **Multiple:** permite introducir varios valores en el mismo campo, tienen que ir separados por comas (y sin blancos).
  - Se permite en elementos como email y file.
    - `<input type="email" id="correo" name="correo" multiple="multiple"/>`
  - Se puede ver también así:
    - `<input type="email" id="correo" name="correo" multiple />`

# Nuevos atributos

- **Pattern:**

- Permite establecer un patrón con el que se valida el campo.
- Utiliza expresiones regulares.
  - `<input type="text" pattern="[0-9]{5}" name="cp" id="cp" />`

- **Form:**

- Permite declarar elementos del formulario que están fuera de las etiquetas form.
- Hace referencia al **nombre** del form.

```
<nav>
```

```
  <p>
```

```
    <input type="search" name="buscador" form="miFormulario" />
```

```
  </p>
```

```
</nav>
```

# Nuevos elementos para formularios

- **Datalist:**

- Permite crear una lista de sugerencias que luego se utilizará dentro de un campo de formulario.
- Creamos una lista con datalist, y luego se referencia desde otro control con el atributo list.

```
<datalist id="info">
```

```
  <option value="112">Emergencias</option>
```

```
  <option value="091">Policía</option>
```

```
  <option value="060">Ambulancias</option>
```

```
</datalist>
```

```
<p>Telefono de asistencia: <input type="tel"  
  name="tno" id="tno" list="info" /></p>
```

# Nuevos elementos para formularios

- **Progress**

- Muestra una barra de progreso. Indica el progreso o estado de una tarea, indicamos un valor y un máximo.
  - `<progress id="progreso" value="5" max="10"></progress>`

- **Meter**

- Muestra una escala para representar una medida. Atributos: min, max (límites de la escala), value: determina el valor medio y low, high y optimum para segmentar la escala.
- Por ejemplo: para representar la seguridad de una clave.

- **Output**

- Representa el resultado de un cálculo. Normalmente se utiliza para mostrar el resultado de un cálculo en javascript.
- `<output>valor</output>`
- **Atributos:**
  - name: el nombre,
  - for: hace referencia a los ids de los controles que influyen en el cálculo.



# Ejemplo: output

```
<form onsubmit="return false" oninput="o.value =  
parseInt(a.value) + parseInt(b.value)">  
  <input name="a" type="number" step="any"> +  
  <input name="b" type="number" step="any"> =  
  <output name="o"></output>  
</form>
```

2  + 3  = 5

↑                      ↑                      ↑  
<input type="number">      <output>

# Ejemplo: output

- También se puede utilizar la propiedad: **valueAsNumber** de **number** para convertir directamente a número.

```
<form onsubmit="return false" oninput="o.value =  
a.valueAsNumber + b.valueAsNumber">  
  <input name="a" id="a" type="number" step="any"> +  
  <input name="b" id="b" type="number" step="any"> =  
  <output name="o" for="a b"></output>  
</form>
```

# Video

- De una forma simple se puede mostrar un video con la etiqueta video:

```
<video width="640" height="360"
src="http://www.youtube.com/demo/google_main.mp4" controls
autobuffer><p> Try this page in Safari 4! Or you can <a
href="http://www.youtube.com/demo/google_main.mp4">download
the video</a> instead.</p>
</video>
```

```
<video src="movie.ogv" controls="controls">
</video>
```

El atributo “controls” proporciona los controles nativos que ofrece HTML5 para la etiqueta video

# Video

- El elemento vídeo permite añadir múltiples elementos, y con éstos elementos direccionar a distintos archivos de video.
- El navegador usará el primer formato reconocido de entre los formatos que le introduzcamos en la etiqueta **<source>** por lo que usaría el que aceptase y el resto los ignoraría:

```
<video width="320" height="240" controls="controls">  
  <source src="movie.ogg" type="video/ogg" />  
  <source src="movie.mp4" type="video/mp4" />  
  Your browser does not support the video tag.  
</video>
```

# Video: Formatos Disponibles

- **Ogg:** archivos Ogg con códec de vídeo Theodora y códec de audio Vorbis.
- **MPEG4:** archivos MPEG4 con códec de vídeo H.264 y códec de audio ACC.
- **WebM:** archivos WebM con códec de vídeo VP8 y códec de audio Vorbis.

# Video: Atributos

- **Autoplay:** Especifica que el vídeo comenzará a reproducirse tan pronto como esté listo
- **Controls:** Especifica que los controles se mostrará como un botón de reproducción.
- **Height(píxeles):** *Especifica la altura del reproductor de vídeo*
- **Loop:** Especifica que el archivo multimedia se iniciará de nuevo, una vez que haya terminado.
- **Preload:** Especifica que el vídeo se cargará en la carga de la página, y listo para reproducir. Se ignora si "autoplay" está activado.
- **Src:** Especifica la dirección URL del vídeo a reproducir.
- **Width(píxeles):** *Especifica el ancho del reproductor de vídeo.*

# Audio

- Etiqueta similar al Video.
- Formatos:
  - Ogg
  - MP3
  - WAV.

```
<audio src="song.ogg" controls="controls">  
</audio>
```

# Audio: Atributos

- **Autoplay:** Especifica que el audio comenzará a reproducirse tan pronto como esté listo
- **Controls:** Especifica que los controles se mostrará como un botón de reproducción.
- **Loop:** Especifica que el archivo multimedia se iniciará de nuevo, una vez que haya terminado.
- **Preload:** Especifica que el audio se cargará en la carga de la página, y listo para reproducir. Se ignora si "autoplay" está activado.
- **Src:** Especifica la dirección URL del audio a reproducir.



# Canvas

- Permite dibujar dentro de la pantalla del Navegador en modo gráfico.
- Es un área rectangular, en el que el programador controla todos y cada uno de los píxeles que lo forman, creando, modificando y añadiendo efectos para conseguir el dibujo requerido.
- Es necesario el uso de **JavaScript** para dibujar.

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

# Buenas prácticas con HTML5

- Utilizar generadores de plantillas básicas.
  - <http://switchtohtml5.com/>
  - <http://shikiryu.com/html5/>
  - <http://www.initializr.com/>
- Guía de etiquetas:
  - [http://www.inmotionhosting.com/infographics/img/html5\\_cheat\\_sheet\\_tags.png](http://www.inmotionhosting.com/infographics/img/html5_cheat_sheet_tags.png)
- Comprobar la compatibilidad de elementos en los distintos navegadores:
  - <http://caniuse.com/>
- Saber cómo hacer compatibles HTML5 y versiones antiguas de Internet Explorer
  - <https://code.google.com/p/html5shim/>
  - Consiste en agregar un fichero en la cabecera para hacerlo compatible.  

```
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>  
<![endif]-->
```
- Usar los elementos correctamente:
  - si HTML5 nos da tags específicos como header, footer, article, section, aside, nav.... debemos usarlos y olvidar los viejos div para esos casos.
- Utilizar validadores de código:
  - <http://validator.w3.org/>

# Enlaces de interés

- <http://slides.html5rocks.com/#landing-slide>