

LMSGI > UT02 > Apuntes > Anexo I

Tema 2 - Anexo 1. HTML 5

Actualizado al 11/10/2022

2.1.	¿Qué hay nuevo en HTML 5?	2
2.1.1.	DOCTYPE.....	2
2.1.2.	Charset	2
2.1.3.	Un primer ejemplo	2
2.2.	Nuevos elementos.....	2
2.2.1.	Etiquetas semánticas.....	2
2.2.2.	Elementos multimedia	5
2.3.	Controles en formularios	9
2.3.1.	Cuadros de texto especializados	9
2.3.2.	Etiquetado de controles	11
2.3.3.	Atributos extendidos para los controles	11
2.4.	Elementos eliminados	13

2.1. ¿Qué hay nuevo en HTML 5?

2.1.1. DOCTYPE

Se simplifica mucho, ahora es:

```
<!DOCTYPE html>
```

2.1.2. Charset

La forma correcta de indicar el juego de caracteres ahora es:

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

UTF-8 es el valor por defecto, así que no hace falta indicarlo explícitamente.

2.1.3. Un primer ejemplo

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
  </head>
  <body>
    Contenido del documento...
  </body>
</html>
```

2.2. Nuevos elementos

2.2.1. Etiquetas semánticas

2.2.1.1. <header>

Permite marcar una cabecera. Esto no tiene que ver con el elemento head, se trata de una serie párrafos que se marcan de esta forma para indicar que pertenecen a la cabecera de la página.

Realmente una página puede tener varios elementos <header>. Por ejemplo, al nivel de la etiqueta <body> indica que su contenido es la cabecera de la página completa (donde se muestra el título general, logotipo,...). Pero dentro de una etiqueta, por ejemplo <article>, indicaría que su contenido es la cabecera del artículo.

2.2.1.2. <footer>

Similar al anterior, sirve para marcar el pie de una página, sección, artículo etc. Dependiendo del contexto en el que se coloque servirá para una cosa u otra. En el caso del elemento <footer> que se coloque al nivel del elemento <body>, servirá para agrupar los elementos de pie de página que suelen ser información sobre el autor, copyright, términos de uso de la página, etc.

2.2.1.3. <section>

Es un elemento que permite dividir en diferentes partes o secciones un documento. A continuación se puede ver un ejemplo de página dividida en secciones, que además incorpora las etiquetas anteriores <header> y <footer> a nivel de <body>.

<pre><!DOCTYPE html> <html lang="es-ES"> <head> <meta charset="UTF-8"> <title>Ejemplo de página HTML 5</title> </head> <body> <header> <h1>Historia de HTML</h1> <p>Desde 1989 hasta nuestros días</p> </header> <section> <h2>Inicios en HTML. Tim Bernes Lee</h2> <p>.....</p> </section> <section> <h2>Creación de la web. Primeros navegadores</h2> <p>.....</p> </section> <section> <h2>La guerra de los navegadores</h2> <p>.....</p> </section> <section> <h2>De HTML 4 a HTML 5 pasando por XHTML</h2> <p>...</p> </section> <footer> <p>Lenguajes de marcas</p> </footer> </body> </html></pre>	<h2>Historia de HTML</h2> <p>Desde 1989 hasta nuestros días</p> <h3>Inicios en HTML. Tim Bernes Lee</h3> <p>.....</p> <h3>Creación de la web. Primeros navegadores</h3> <p>.....</p> <h3>La guerra de los navegadores</h3> <p>.....</p> <h3>De HTML 4 a HTML 5 pasando por XHTML</h3> <p>...</p> <p>Lenguajes de marcas</p>
---	---

2.2.1.4. <nav>

Se trata de un elemento que marca a su contenido como una sección de enlaces, es decir una barra de navegación. Más adelante con CSS se puede dar un formato especial a dichos enlaces. nav se puede escribir dentro de cualquier elemento HTML de sección (como <section>, <article>, <header>, <footer>, ...).

Los elementos marcados por <nav> pueden ser omitidos por los lectores digitales de páginas web que utilizan, por ejemplo, las personas invidentes. Lo cual facilita la comprensión del texto y así que dicho contenido quede marcado sólo para utilizar los enlaces interiores a <nav>.

2.2.1.5. <article>

Si se observan los elementos descritos anteriormente, parece claro que HTML 5 utiliza como metáfora la forma de distribuir contenidos de un periódico. Así, hay cabeceras, pies, secciones y, con este elemento, artículos.

La idea es colocar dentro de este elemento (que tiene sentido dentro del elemento <section>, o incluso aparecer de forma independiente) es colocar contenido que pueda ser entendido como un todo que describa un tema de forma íntegra.

Nuevamente el formato mediante CSS permitiría que apareciera con el estilo que deseemos.

2.2.1.6. <hgroup>

Permite agrupar varios títulos (elementos h1 a h6) dentro de esta etiqueta para darles estilo común. Se utiliza dentro de <article> para marcar su zona de títulos. Ver ejemplo en <figcaption>.

2.2.1.7. <figure>

Sirve para agrupar los elementos relativos a una imagen como la propia imagen (elemento img), el título de la misma, el pie, los párrafos relativos, etc. Ver ejemplo en <figcaption>

2.2.1.8. <figcaption>

Permite indicar el título de una imagen, dentro de un elemento figure. Ejemplo de uso:

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="UTF-8">
    <title>Ejemplo 2 de página HTML 5</title>
  </head>
  <body>
    <article>
      <hgroup>
        <h2>Murcia</h2>
        <h3>Paisajes</h3>
      </hgroup>
      <figure>
        
        <figcaption> Vista de La Aljorra desde el Pericón </figcaption>
      </figure>
      <p> Los paisajes en la provincia de Murcia... </p>
    </article>
  </body>
</html>
```

2.2.1.9. <aside>

Permite marcar texto dentro de un artículo para que no se tenga en cuenta como parte del texto del artículo, sino como un texto aparte que permite realizar aclaraciones al artículo,

referencias, resúmenes remarcados y sobre todo cuadros de texto de estilo periodístico para destacar partes del artículo. Para que este texto aparezca de manera especial, debe dársele formato con CSS.

2.2.2. Elementos multimedia

2.2.2.1. Problemática anterior a HTML 5

Si incorporamos vídeo a nuestra página a través de un enlace normal:

```
<a href="video1.mpeg">Ver vídeo</a>
```

El navegador buscará si disponemos del plugin apropiado para ver el vídeo, si no es así simplemente lo descargará en nuestro ordenador, al no poder mostrar él el contenido.

2.2.2.2. <embed>

Se trata de una etiqueta ya veterana que se utiliza para colocar en una página web elementos no pertenecientes al lenguaje HTML como animaciones Flash, vídeo, audio, etc.

Usa los siguientes atributos:

Atributo	Uso
src	URL al recurso que se desea mostrar
type	Tipo MIME que indica el contenido del recurso que se incorpora con la etiqueta
height	Altura de la ventana que mostrará el recurso
width	Anchura de la ventana que mostrará el recurso

2.2.2.3. <object>

Esta etiqueta está orientada a sustituir a la anterior y permite incorporar cualquier tipo de contenido a una página web. Los atributos posibles son:

Atributo	Uso
data	URL al recurso que se desea mostrar
type	Tipo MIME que indica el contenido del recurso que se incorpora con la etiqueta
height	Altura de la ventana que mostrará el recurso
width	Anchura de la ventana que mostrará el recurso
usemap	Permite indicar el nombre de un mapa de imágenes (usando #nombre) que se usará sobre el objeto
name	Permite indicar el nombre del objeto
form	Indica el nombre del formulario al que pertenece este objeto

2.2.2.4. <param>

Es un elemento interior a <object>, esto es, que permite especificar parámetros a los plugins de los navegadores encargados de mostrar el objeto.

A través de <param> pasamos instrucciones al plugin que reproduzca el objeto. La forma es mediante los atributos:

Atributo	Uso
name	Nombre por el que se reconoce al atributo
value	Valor del atributo

Ejemplo:

```
<object data="mountains.wmv" type="video/x-ms-wmv" width="500" height="300">
  <param name="autoplay" value="true" />
</object>
```

2.2.2.5. <iframe>

Es un elemento que había desaparecido en el estándar pero que HTML 5 ha recuperado.

La idea original es colocar un documento dentro de otro documento, es decir sirve para incrustar contenido de una dirección dentro de la página que le hace referencia.

Su popularidad actual se debe a Youtube, ya que es el elemento que sugiere la página para colocar los vídeos de esta plataforma en nuestras páginas.

```
><iframe src="https://www.youtube.com/embed/bHQqvYy5KYo"
frameborder="0" allowfullscreen>...</iframe>
```

Sus principales atributos son:

Atributo	Uso
src	URL al recurso que se desea mostrar
width	Anchura del objeto en nuestra página
height	Altura del objeto en nuestra página

2.2.2.6. <video>

Es un elemento que sólo funciona en HTML 5 y es el método propuesto para incorporar vídeos a una página web.

Atributo	Uso
src	URL al vídeo que se desea mostrar
width	Anchura del vídeo en nuestra página
height	Altura del vídeo en nuestra página
autoplay	Usa el valor fijo autoplay para indicar que el vídeo se inicia automáticamente en cuanto se descargue
loop	Usa el valor fijo loop para indicar que el vídeo se ejecuta automáticamente una y otra vez
controls	Con valor controls indica que el vídeo mostrará los controles de reproducción (pausa, play,...)
preload	Recomendación sobre cómo debemos realizar la descarga. Posibilidades: <ul style="list-style-type: none"> • auto. El vídeo se descarga en cuanto se carga la página. • none. El vídeo no se descarga. Cuando el usuario pulse play, se descargará. • metadata. Descarga los metadatos del vídeo, no el vídeo en sí.
poster	Permite indicar la dirección URL a una imagen que se mostrará mientras el vídeo no se está reproduciendo. Si no se usa este atributo, se usa el primer fotograma del vídeo como póster.

2.2.2.7. <source>

Este elemento se utiliza dentro del anterior <video>.

Se denomina *códec* (de codificar, decodificar), al software que se usa para crear los vídeos. Un vídeo actual usa un formato comprimido de datos, por lo que para crearlos se codifican los datos y para mostrarle, se decodifica.

Eso significa que para ver un vídeo en un formato concreto, se necesita el *códec* apropiado. No siempre están disponibles en el navegador todos los *códec*s existentes. HTML 5 ha conseguido subsanar en parte la situación, mediante la etiqueta <source>.

La idea es que el vídeo se codifique en varios formatos distintos. Luego hacemos referencia a las diferentes posibilidades mediante la etiqueta <source>, dentro de la etiqueta <video>.

Después, el navegador usará el formato que sea capaz de decodificar, según los *códec*s que tenga disponibles para ello.

El elemento <source> fundamentalmente dispone de dos atributos:

Atributo	Uso
src	URL al vídeo en el formato concreto indicado por type.
type	Tipo MIME del vídeo. Además, dentro del tipo MIME se pueden indicar los <i>códec</i> s concretos que necesitamos para decodificar el vídeo.

Un ejemplo de uso de source:

```
<video autoplay="autoplay" controls="controls" poster="caratula.jpg" >
  <source src="video.mp4" type="video/mp4;codecs='avc1.42E01E, mp40a.40.2' " />
  <source src="video.ogv" type="video/ogg;codecs='theora, vorbis' " />
  El navegador no puede mostrar el vídeo
</video>
```

2.2.2.8. <track>

Al igual que la anterior, se usa dentro de <video>. Permite vincular un archivo con pistas de subtítulos al vídeo. También capítulos y otros metadatos. Sus atributos principales son:

Atributo	Uso
src	URL al archivo VTT o SRT que contiene los subtítulos u otros metadatos.
kind	Tipo de subtítulos, que pueden ser: <ul style="list-style-type: none"> subtitles. Subtítulos normales captions. Igual que el anterior pero con posibilidad de añadir sonidos. descriptions. Texto alternativo e independiente de la película. chapters. Contiene los capítulos para navegar por la película. metadata. Metadatos de la película.
label	Nombre de los subtítulos para reconocerlos en caso de haber indicado varios
srclang	Idioma de los subtítulos

A continuación, para ver como embeber una película haciendo referencia a dos versiones con diferente códec y dos pistas de subtítulos opcionales, en Inglés y Noruego:

```
<video width="320" height="240" controls>
  <source src="forrest_gump.mp4" type="video/mp4">
  <source src="forrest_gump.ogv" type="video/ogg">
  <track src="subtitles_en.vtt" kind="subtitles" srclang="en" label="English">
  <track src="subtitles_no.vtt" kind="subtitles" srclang="no" label="Norwegian">
</video>
```

2.2.2.9. <audio>

Se trata de un elemento comparable al video, pero en este caso para incorporar audio a la página. De hecho, los siguientes atributos están vigentes con el mismo significado que para el vídeo: "src", "autoplay", "loop", "controls" y "preload".

Con el audio hay el mismo problema con la cuestión de los códecs y los formatos, por lo que también es habitual convertir el audio a distintos formatos y dar opciones dentro de la etiqueta <audio> gracias a la etiqueta <source>. Ejemplo:

```
<audio controls="controls" autoplay="autoplay">
  <source src="audio1.ogg" type="audio/ogg">
  <source src="audio2.mp3" type="audio/mpeg">
  No se puede reproducir el archivo de audio
</audio>
```


2.3. Controles en formularios

El nuevo estándar HTML ha mejorado notablemente los formularios gracias a nuevos controles. La pega es que no todos los controles son soportados por los navegadores; hay controles que algunos navegadores no permiten utilizar.

Cuando un navegador no comprende un control HTML 5, suele traducirle como un cuadro de texto normal.

2.3.1. Cuadros de texto especializados

La diferencia entre ellos es el uso del atributo *type*. Los buenos navegadores que reconocen estos cuadros modifican su comportamiento para:

- Facilitar al usuario la entrada de datos.
- Validar los datos y evitar confirmarlos hasta que sean correctos.
- Mostrar el teclado en pantalla (en caso de requerirlo) más adecuado.

Posibilidades:

2.3.1.1. `<input type="number">`

Acepta sólo números. El navegador se asegura de no aceptar texto. El atributo `maxlength` permite indicar el máximo de dígitos del número.

2.3.1.2. `<input type="email">`

Especializado en direcciones de correo electrónico.

2.3.1.3. `<input type="url">`

Acepta sólo direcciones URL.

2.3.1.4. `<input type="date">`

Acepta sólo fechas válidas. Usa el formato de fecha configurado en el sistema operativo del usuario que visita la página. Los navegadores además proporcionan un cuadro visual más sencillo para recoger la fecha.

2.3.1.5. `<input type="time">`

Acepta sólo horas válidas; funciona igual que el cuadro anterior.

2.3.1.6. `<input type="datetime">`

Acepta fecha y hora.

2.3.1.7. **<input type="month">**

Acepta sólo números del 1 al 12, referidos a un mes.

2.3.1.8. **<input type="search">**

Presenta un cuadro de texto pensado para hacer búsquedas.

2.3.1.9. **<input type="tel">**

Permite introducir números de teléfono. Muy útil para mostrar el teclado telefónico en dispositivos móviles.

2.3.1.10. **<input type="range">**

Presenta un control “deslizante” (a modo de potenciómetro) para elegir un valor de entre un rango. Los atributos max y min establecen los límites máximo y mínimo del control. El atributo step indica cuánto se mueve el control (si de uno en uno, de dos en dos,...).

2.3.1.11. **<input type="color">**

Presenta un control de selección de colores. El color se toma en formato #xxxxxx donde cada x es una cifra hexadecimal. Los dos primeros para la proporción de rojo (de 0 a 255 en decimal), los dos siguientes para la de verde y los dos últimos para la de azul. Es decir, lo toma en el formato habitual de colores de HTML.

2.3.1.12. **<datalist>**

En esta ocasión no hablamos de ningún control especializado sino de la posibilidad de indicar, mediante esta etiqueta <datalist> y sus sub-etiquetas <option>, un conjunto de valores o posibilidades para cualquier control de los anteriores.

Para ello, indicaremos al control el identificador de la lista, mediante el atributo *list*, dándole el valor que tenga la lista en su atributo identificador *id*. Veamos un ejemplo:

```
<input list="browsers">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

2.3.2. Etiquetado de controles

En lugar de poner los títulos de los controles poniendo texto directamente, se recomienda usar la etiqueta `<label>` que mediante el atributo *for* puede asociarse al control para el que hace de rótulo (a *for* debe dársele el mismo valor que tenga el control en su atributo *id*).

Esto permite una mejor usabilidad. Los navegadores reconocen `<label>` y, por ejemplo, al hacer clic en el texto de la etiqueta, el control referido en su atributo "for" recibe el foco.

Un ejemplo de control de entrada de datos que se rotula usando `<label>` sería:

```
<form action="formulario.php" method="get">
  <label for="nombre">Nombre completo:</label>
  <input type="text" id="nombre" />
  <br />
  <input type="submit" value="enviar"/>
  <br />
</form>
```

2.3.3. Atributos extendidos para los controles

2.3.3.1. *required*

Este atributo obliga a rellenar con algún valor el control en el que se usa. Es decir hace que un determinado control sea de obligado rellenado en un formulario. Uso:

```
<input type="text" name="texto" id="texto" required />
<input type="text" name="texto" id="texto" required="required" />
```

La primera forma es aceptada en HTML. Sólo la segunda se acepta en XHTML.

2.3.3.2. *multiple*

Permite en los cuadros de entrada de texto que el usuario pueda indicar varios valores si los separa con comas. Al igual que *required*, puede aparecer con valor (xhtml) o sin él.

2.3.3.3. *pattern*

Permite colocar una expresión regular en un cuadro de texto que, obligatoriamente, tendrá que cumplir el cuadro en el que se use el atributo. En el siguiente ejemplo se habilita un cuadro de texto que sólo acepta introducir 5 letras mayúsculas y tres números:

```
<input type="text" pattern="[A-Z]{5}[0-9]{3}" id="gustos" name="texto" />
```

2.3.3.4. **placeholder**

Se denomina así a un texto que aparece sobre una caja de entrada de texto, pero antes de que escribamos nada, palidecido, a modo de ejemplo o ayuda de lo que se espera que escribamos. El siguiente ejemplo nos aclarará qué es. Vemos en gris “Ej. 12345678Z”. Dicho texto desaparecerá cuando comencemos a teclear el DNI. Eso es el *placeholder*:

Indique aquí el DNI:

Para conseguirlo, se ha usado este código HTML:

```
<label for="dni">Indique aquí el DNI:</label>  
<input type="text" id="dni" required placeholder="Ej. 12345678Z" />
```

2.3.3.5. **autocomplete**

Permite activar (valor *on*) o desactivar (valor *off*) el autocompletado del navegador. El autocompletado es la opción que permite a los usuarios cuando rellenan un formulario ver entradas habituales que han escrito en el mismo u otros formularios.

Conviene no hacer uso si, por ejemplo estamos ante datos sensibles (como contraseñas).

2.3.3.6. **min, max y range**

Son atributos que se pueden utilizar en muchos tipos de cuadros (number, date, time, range,...) que almacenan valores numéricos o similares.

- *min* y *max* establecen los límites del cuadro. Por ejemplo si el cuadro es numérico y ponemos *min*="1" y *max*="50", eso impedirá poner valores en el cuadro fuera de esos límites.
- El parámetro *step* indica el mínimo incremento de valor en el cuadro; si en el ejemplo anterior ponemos *step*="3" del valor 1 saltaremos al 4. Los controles del cuadro (en el caso de los cuadros numéricos, las flechitas) obedecen a esa configuración.

2.4. Elementos eliminados

A continuación, se muestran etiquetas eliminadas en HTML5 y cuáles usar en su lugar.

CSS hace referencia a que, en lugar de utilizar una etiqueta dentro de HTML, es preceptivo aplicar el formato desde una hoja de estilos basada en este estándar.

Etiqueta	Usar en su lugar
<code><acronym></code>	<code><abbr></code>
<code><applet></code>	<code><object></code>
<code><basefont></code>	CSS
<code><big></code>	CSS
<code><center></code>	CSS
<code><dir></code>	<code></code>

Etiqueta	Usar en su lugar
<code></code>	CSS
<code><frame></code>	
<code><frameset></code>	
<code><noframes></code>	
<code><strike></code>	CSS
<code><tt></code>	CSS