

Diseño, Evaluación y Desarrollo de Interfaces



HTML 5

Luis Rodríguez Baena (luis.rodriguez@upsam.es)

Universidad Pontificia de Salamanca
Escuela Superior de Ingeniería y Arquitectura

Introducción

Algo de historia

- ❑ La última recomendación de HTML del W3C (HTML 4.01) es de 1998.
- ❑ Después abandona el desarrollo de HTML y se centra en XHTML:
 - Recomendación XHTML 1.0 en 2000.
 - Recomendación XHTML 1.1 en 2001.
 - Entre agosto de 2002 y julio de 2006 aparecieron 8 borradores de XHTML 2.0.
 - ✓ Antes de aparecer el noveno borrador, en 2009, el W3C retiró el grupo de trabajo de XHTML a favor de HTML5.
- ❑ En 2004 un grupo de empresas formado por Apple, Mozilla Foundation y Opera Software forman el Web Hypertext Application Technology Working Group (WHATWG, www.wathwg.org) .
 - Su propósito es seguir desarrollando el lenguaje HTML ante el abandono del W3C.
 - Ese mismo año se crea el borrador la especificación Web Applications 1.0 que se convertiría en HTML5.

Introducción

Algo de historia (II)

- ❑ En 2007 el W3C se une al WHATWG en un grupo de trabajo para desarrollar la especificación de HTML5.
 - En 2008 se crea el primer borrador de trabajo (*Working Draft*).
- ❑ Mientras que el W3C quería poner un límite al desarrollo para poder hacer una primera recomendación oficial, el WHATWC quería seguir desarrollando el lenguaje.
 - En 2011 se vuelven a dividir.
- ❑ La especificación HTML5 del W3C es una foto de un momento del desarrollo del lenguaje.
 - Se supone que pasará a recomendación oficial en 2014.
- ❑ El WHATWG sigue desarrollando el lenguaje como un estándar vivo (*live standard*) que continúa evolucionando.

Declaración del tipo de documento

□ Declaración del tipo de documento.

- HTML 4.01 tenía tres tipos de documento:
 - ✓ Transitional, Strict y Frameset
- XHTML, además tenía declaraciones para XHTML 1.1 y XHTML 2.0.
- A diferencia de HTML 4.01 y XHTML, sólo hay un tipo documento.
- La única declaración del tipo de documento es:
`<!DOCTYPE html>`

Sintaxis: HTML y XML

- ❑ HTML5 admite dos estilos de escritura:
 - HTML y XML.
- ❑ La sintaxis HTML permite escribir documentos al estilo de HTML 4.01.
 - No son necesarias las etiquetas de cierre en elementos sin contenido.
 - No es sensible a mayúsculas.
 - Los atributos no tienen por qué ir entrecomillados.
 - Puede haber atributos sin contenido (minimización de atributos o atributos lógicos), por ejemplo `disabled` o `checked`.

```
<input name=sexo id=hombre type=radio checked>
```
- ❑ Pero también permite seguir la sintaxis XHTML.
 - Incluso se pueden combinar las dos en el mismo documento.

El elemento html y elementos de la cabecera

❑ El elemento raíz `html` no varía.

- Un elemento `html` típico sería:

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

❑ Elemento `link`.

- Al enlazar a hojas de estilo, se asume por defecto que el atributo `type` tiene el contenido `"text-css"`.
 - ✓ El atributo `type` no es obligatorio.

❑ Elemento `meta`.

- Aunque se puede seguir utilizando el elemento `meta` para indicar al servidor el tipo de contenido (`html`, `xml`, etc.) y la codificación de caracteres, existe un atributo `charset` que también permite indicar la codificación de caracteres.

- ✓ En HTML 4.01 y XHTML...

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

- ✓ En HTML 5...

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

El elemento `html` y elementos de la cabecera (II)

□ Elemento `style`.

- Se asume por defecto que el atributo `type` tiene el contenido `"text/css"`.
 - ✓ El atributo `type` no es obligatorio.
- Es posible definir estilos dentro del elemento `body`.
 - ✓ En ese caso, el atributo lógico `scoped` permite indicar que las reglas de estilo sólo se aplican en el elemento padre dónde está definido (el atributo `scoped` puede cambiar en la versión definitiva de HTML5).
- En el siguiente ejemplo el estilo sólo se aplica a la primera sección...

```
<section>
  <style>
    h3 {background: red;
        color:white;}
  </style>
  <h3>Sección 1.2</h3>
  <p>Lorem ipsum ... </p>
</section>
<section>
  <h3>...</h3>
  <p>...</p>
```

El elemento html y elementos de la cabecera (III)

□ Elemento `script`.

- Se asume por defecto que el atributo `type` tiene el contenido `"text/javascript"`.
 - ✓ El atributo `type` no es obligatorio.
- Nuevo atributo `async`.
 - ✓ Para scripts externos hace una ejecución asíncrona, es decir, fuerza a que el script se ejecute mientras se sigue generando la página.
 - Este es el modo de trabajo estándar en Chrome e IE, pero no en Firefox y Opera.
 - ✓ Se trata de un atributo lógico.

```
<script src="mijavascript.js" async></script>  
<script src="otrojavascript.js" async="async"></script>
```

Elementos semánticos

- ❑ HTML4 y XHTML no presentan elementos propios para definir la semántica de las distintas partes de la estructura de un documento.
 - Para definir las se utiliza el elemento genérico `div`.
- ❑ HTML5 tiene distintos elementos que permiten definir más claramente la función de cada una de las partes de la estructura de un documento.
- ❑ Los nuevos elementos son:
 - `section`. Representa una sección de un documento o parte de un documento que forma una unidad.
 - `article`. Representa una parte independiente de contenido de un documento.
 - `nav`. Representa una parte de un documento con fines de navegación.
 - `aside`. Representa una pieza de contenido relacionado de alguna forma con el resto de la página.
 - `header`. Representa un información de introducción en la página o en una sección.
 - `footer`. Representa el pie de una sección (o de la página principal).
 - `main`. Se puede usar como contenedor para el contenido principal de una página.
 - `figure`. Representa una parte de contenido independiente que ayuda a la explicación de contenido principal como una imagen, un vídeo, un fragmento de código, etc.
 - `figcaption`. Se utiliza como leyenda a una figura.

Elementos semánticos (II)

□ En HTML 4/ XHTML...

The image shows a screenshot of the W3C Web Accessibility Initiative website with several HTML elements highlighted and annotated with code snippets:

- Header:** The W3C logo and "Web Accessibility initiative" are enclosed in a `<div id="cabecera">` tag.
- Navigation:** A vertical list of links on the left side is enclosed in a `<div id="navegacion">` tag.
- Main Content:** The main heading "Web Accessibility Initiative (WAI)" is enclosed in a `<div id="principal">` tag. Below it, a "Highlights" section is enclosed in a `<div id="highlights">` tag, and a specific news item is enclosed in a `<div class="noticia">` tag.
- Sidebar:** A right-hand sidebar containing "WAI develops..." and "WAI welcomes..." sections is enclosed in a `<div id="sidebar">` tag.
- Footer:** The footer area, including site maps and copyright information, is enclosed in a `<div id="pie">` tag.

Elementos semánticos (III)

□ En HTML 5

The image shows a screenshot of the W3C Web Accessibility Initiative website. The page is annotated with HTML5 semantic elements. The header contains the W3C logo and the text 'Web Accessibility initiative' with the annotation `<header>`. Below the header is a navigation bar with 'W3C Home' and 'WAI develops...'. The main content area is divided into three columns. The left column contains a list of links: 'Web Accessibility Initiative (WAI) Home', 'Getting Started', 'Designing for Inclusion', 'Guidelines and Techniques', 'Planning & Implementing', 'Evaluating Accessibility', and 'Presentations & Tutorials', with the annotation `<nav>`. The middle column contains the main heading 'Web Accessibility Initiative (WAI)' with the annotation `<main>`, followed by a 'Highlights' section with the annotation `<article>`, and a sub-section titled 'Easy Checks - A First Review of Web Accessibility is updated for the holidays' with the annotation `<section>`. The right column contains a list of bullet points under the heading 'WAI develops...' with the annotation `<aside>`. The footer contains links for '[WAI Site Map]', '[Help with WAI Website]', '[Search]', and '[Contacting WAI]', along with contact information and copyright information, with the annotation `<footer>`.

W3C Home

Web Accessibility Initiative (WAI) Home

Getting Started

Designing for Inclusion

Guidelines and Techniques

Planning & Implementing

Evaluating Accessibility

Presentations & Tutorials

Web Accessibility Initiative (WAI)

Highlights

Easy Checks - A First Review of Web Accessibility is updated for the holidays

Have you ever wondered: "Is this web page accessible?" [Easy Checks - A First Review of Web Accessibility](#) provides simple steps to start assessing web accessibility. It is designed for

WAI develops...

- guidelines widely regarded as the international standard for Web accessibility
- support materials to help understand and implement Web accessibility
- resources, through international collaboration

WAI welcomes...

- participation from around the world
- volunteers to review, implement, and promote guidelines
- dedicated participants in

[WAI Site Map] [Help with WAI Website] [Search] [Contacting WAI]

Feedback welcome to wai-eo-editors@w3.org (a publicly archived list) or wai@w3.org (a WAI staff-only list).

Copyright © 1994-2012 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply. Your interactions with this site are in accordance with our [public](#) and [Member](#) privacy statements.

Elementos semánticos

Secciones

- ❑ `article`, `section`, `nav` y `aside` crean secciones del documento.
 - El agente de usuario modifica la jerarquía de secciones con estos elementos (*sectioning content*).
 - ✓ Modifica el algoritmo de secciones (*outline algorithm*)
 - Los elementos de cabecera `h1..h6` también crean secciones implícitas.
- ❑ Elemento `article`.
 - Representa un bloque de contenido completo.
 - La idea es que sea un bloque que se pueda distribuir de forma independiente o ser reutilizado.
 - ✓ Un mensaje de un foro, un artículo de una revista, una entrada de blog, etc.
 - Pueden anidarse.
 - ✓ Estarán relacionados con el artículo principal.
 - ✓ Un artículo anidado podría incluir, en un blog, los comentarios al artículo.

Elementos semánticos

Secciones (II)

□ Elemento `section`.

- Representa una sección genérica de un documento.
- Suele incluir una cabecera y permite organizar un bloque de información en partes.
 - ✓ Por ejemplo, capítulos de un libro, apartados en un artículo de una revista, cada una de las pestañas en un documento organizado con pestañas (que podrían tener dentro artículos a su vez).
- No se trata de un elemento genérico. Si se utiliza sólo para aplicar un estilo a una parte del contenido, es mejor utilizar el elemento genérico `div`.
- Si el autor va a reutilizar el contenido, por ejemplo, mediante sindicación, puede sustituirse por el elemento `article`.



```
<section id="recomendados">
  <h1>Última hora</h1>
  <article id="noticia1">
    <h2>Noticia 1</h2>
    ...
  </article>
  <article id="noticia2">
    <h2>Noticia 2</h2>
    ...
  </article>
  ...
</section>
```

Elementos semánticos

Secciones (III)

❑ Elemento `nav`.

- Agrupa un conjunto de enlaces que permite navegar a otras páginas del sitio o a partes de la misma página.
- Los enlaces que contienen, pueden agruparse mediante listas.

❑ Elemento `aside`.

- Permite agrupar bloques de contenido con información relacionada tangencialmente con el contenido principal.
 - ✓ Información de contacto, enlaces relacionados a otras páginas, citas literales, anotaciones, etc.

HTML5 in a way that can be supported by older n... accessibility languages like
WAI-ARIA (which I'll say more about later).

Este podría ser
un ejemplo de
elemento aside

■ **Note** At time of this writing, many of the big players in the browser market have really good HTML5 support. It is often the AT that doesn't.

This book pays particular attention to backward compatibility in its general tone and tries to help you support what is known as *legacy user agents*, or older browsers and AT. Approaching the markup and design of your web content in a way that supports both the newer user agents that understand it, as well as those that don't (by giving them something to get their semantic teeth into), is the bedrock of *progressive enhancement*, which is the more enlightened approach to design.

Elementos semánticos

Secciones (IV)

❑ Elemento `header`.

- Representa el contenido introductorio de la sección en la que está contenido.
- En un artículo o sección, podría incluir el título y una entradilla.
- En el elemento `body` contendría la cabecera de la página que se repiten en todo el sitio: cabecera, navegación, herramientas, etc.

❑ Elemento `footer`.

- Representa el pie de página de la sección en la que está contenido.
- Podría incluir quién lo ha escrito, enlaces a documentos relacionados, información de copyright.
 - ✓ Cuando se trata del pie del elemento `body` incluye también información corporativa, enlaces de navegación alternativos, etc.

❑ También son elementos de secciones el elemento `body`, el elemento `address` y las cabeceras `h1..h6`

Elementos semánticos

Elementos de grupo

- ❑ Permiten agrupar otros elementos en una unidad semántica.
- ❑ Elemento `main`.
 - Se usa como contenedor del contenido principal de una página.
 - Sólo puede haber uno por página.
- ❑ Elemento `figure`.
 - Representa contenido referenciado dentro del contenido principal:
 - ✓ Responden a mensajes del contenido principal tipo: “en el siguiente código”, “cómo se puede ver en la figura x”, etc.
 - ✓ Se trata de elementos relacionados con el documento pero que pueden moverse sin afectar al significado del documento (por ejemplo, se pueden mover a un apéndice).
 - Una ilustración, un diagrama, listados de código, etc.
- ❑ Elemento `figcaption`.
 - Sería la leyenda del elemento `figure` que lo contiene.
- ❑ También son elementos de grupo, los elementos `p`, `hr`, `pre`, `blockquote`, `ol`, `ul`, `li`, `dl`, `dt`, `dd` y `div`.

Elementos semánticos

Elementos de texto

□ Elementos nuevos.

- Elemento `mark`.

- ✓ Representa un texto marcado para referencia (por ejemplo un texto añadido o modificado de una versión anterior del documento).
- ✓ Los navegadores marcan el texto en amarillo.

- Elemento `time`.

- ✓ Se utiliza para marcar un elemento con información de fecha/hora, zona horaria, duración, etc.
- ✓ El atributo `datetime` puede contener una especificación de fecha/hora concreta.

- En www.w3.org/TR/html5/text-level-semantics.html#datetime-value se pueden ver los formatos admitidos.

```
<p>El <time datetime="2014-01-06">día de Reyes</time> me  
levanté a las <time>10:00</time></p>
```

Elementos semánticos

Elementos de texto (II)

- ❑ Algunos elementos se han modificado ligeramente.
 - El elemento `a` ya no se puede utilizar como destino de un enlace interno.
 - ✓ Para marcar un destino interno se debe utilizar la sintaxis de XHTML 1.1:
 - `<h1 id="capitulo1">Capitulo 1</h1>`, indicaría el destino de `#capitulo1`.
 - ✓ El atributo `download` indica que la intención del autor es descargar un recurso.
 - Es un atributo lógico.
 - ✓ El atributo `type` advierte el tipo MIME del contenido del enlace (.
 - El elemento `strong` indica texto importante.
 - El elemento `s` (desaprobado en HTML 4.01) ahora se utiliza para indicar que se trata de texto no importante, exacto o erróneo.
 - El elemento `b`, se utiliza sólo para dar un estilo distinto sin importancia semántica.
 - El elemento `i` indica ahora un tipo distinto de texto, como taxonomías científicas, una frase en otro lenguaje, etc. (el uso académico de la letra cursiva).
 - El elemento `u` se utiliza para indicar algún texto que debería ser estilísticamente diferente, como palabras mal escritas.
 - El elemento `cite` indica ahora el título de una obra o del autor, en lugar de una cita o referencia a otro documento.

Elementos multimedia

- ❑ HTML 5 proporciona ahora elementos para insertar elementos multimedia como audio y video sin necesidad de plugins externos.
 - Elemento `video`. Incrusta un elemento de video.
 - Elemento `audio`. Incrusta un elemento de audio.
 - Elemento `track`. Permite añadir pistas de subtítulos en un elemento `video`.
 - Elemento `source`. Permite especificar distintos formatos de audio o video para su compatibilidad.
 - ✓ No todos los agentes de usuario admiten los mismos formatos de video.
- ❑ Los elementos `audio` y `video` permiten contenido alternativo.
 - El contenido entre las etiquetas puede incluir párrafos, cabeceras, enlaces u otros elementos HTML.
 - Si no está disponible estas funciones, aparecerá un texto alternativo.

```
<video src="unvideocualquiera.mp4">  
  El agente de usuario no admite reproducir videos  
</video>
```

Elementos multimedia

Video

□ Atributos:

- `src`. Contiene un URL con el origen del contenido.
- `poster`. Contiene un URL con la imagen inicial del reproductor.
- `autoplay`, `loop`, `muted`. Atributos lógicos que arrancan automáticamente, repiten o silencian el vídeo.
- `controls`. Atributo lógico que indica si se utilizan los controles nativos del reproductor o el usuario utiliza controles personalizados.
- `width` y `height`. Anchura y altura del reproductor en píxeles o porcentajes.
- `preload`. Establece la forma de cargar el contenido. Toma los valores `none` (no hace una carga previa), `metadata` (solo carga los metadatos) o `auto` (el agente de usuario intenta cargar antes el contenido).

Elementos multimedia

Video (II)

❑ No todos los agentes de usuario soportan todos los formatos de video.

- El elemento `source` permite indicar distintos formatos de video para varios navegadores.
 - ✓ El atributo `src` permite indicar un url con un formato determinado.
 - ✓ El atributo `type` permite indicar el formato de vídeo y codec. Por ejemplo:
 - Para mp4, "video/mp4" (IE, Firefox, Chrome, Safari).
 - Para ogg, "video/ogg" (Firefox, Chrome, Opera).
 - Para WebM, "video/webm" (Chrome, Opera).
 - ...
 - En wiki.whatwg.org/wiki/Video_type_parameters#Browser_Support se puede encontrar una lista de tipos MIME y codecs soportados por los navegadores.

```
<video controls="" preload="auto" height="360" width="471" poster="media/verderon.png">
  <source src="http://www.colimbo.net/media/verderon.webm" type="video/webm">
  <source src="http://www.colimbo.net/media/verderon.ogv" type="video/ogg">
  <source src="http://www.colimbo.net/media/verderon.mp4" type="video/mp4">
  Su navegador no implementa el elemento <code>video</code>.
  <p>Imagen de un verderon en el campo. El vídeo también está disponible
  en <a href="http://www.youtube.com/watch?v=kkgoIt1wfJs">Youtube</a></p>
</video>
```

Elementos multimedia

Audio

- ❑ El elemento `audio` permite añadir un reproductor de audio.
 - Tiene los atributos `src`, `preload`, `autoplay`, `loop`, `muted` y `controls`.
 - También permite indicar varias fuentes mediante el elemento `source`.

Elementos multimedia

Canvas

- ❑ El elemento canvas presenta un lienzo sobre el que se puede dibujar de forma dinámica.
 - Se puede encontrar un tutorial sobre el elemento en [developer.mozilla.org/es/docs/Canvas tutorial](http://developer.mozilla.org/es/docs/Canvas_tutorial).

```
<canvas id="canvas" height="400" width="600">
  El navegador no soporta canvas
  <script>
    var canvas = document.getElementById('canvas'),
        context = canvas.getContext('2d');
    context.lineJoin = 'round';
    context.lineWidth = 30;
    context.font = '24px Helvetica';
    context.fillText('Click anywhere to erase', 175, 200);
    context.strokeStyle = 'goldenrod';
    context.fillStyle = 'rgba(0, 0, 255, 0.5)';
    context.strokeRect(75, 100, 200, 200);
    context.fillRect(325, 100, 200, 200);
    context.canvas.onmousedown = function (e) {
      context.clearRect(0, 0, canvas.width, canvas.height);
    };
  </script>
</canvas>
```

Elementos interactivos

- ❑ El elemento `details` es un widget que permite mostrar y esconder contenido de forma interactiva.
 - Actualmente sólo funciona en Chrome.
 - Incluye un elemento hijo `summary` con el resumen o leyenda del contenido.

```
<p><b>Salida:</b> Madrid 12-feb 14:10</p>
<p><b>Llegada:</b> Cape Town 13-feb 11:40</p>
<details>
  <summary>Detalles del vuelo</summary>
  <p>Salida 14:10 Miercoles , 12 febrero, Terminal 4 Barajas (MAD), Madrid - España</p>
  <p>Llegada 00:15 Jueves, 13 febrero, Terminal 3 Dubai International Airport (DXB),
  Dubai - United Arab Emirates</p>
  <hr/>
  <p>Conexi&oacute;n</p>
  <hr/>
  <p>Salida 03:50 Jueves, 13 febrero, Terminal 3 Dubai International Airport (DXB),
  Dubai - United Arab Emirates</p>
  <p>Llegada 11:40 Jueves, 13 febrero, Cape Town International Airport (CPT),
  Cape Town - South Africa</p>
</details>
```

Elementos interactivos (II)

Salida: Madrid 12-feb 14:10

Llegada: Cape Town 13-feb 11:40

► Detalles del vuelo

Salida: Madrid 12-feb 14:10

Llegada: Cape Town 13-feb 11:40

▼ Detalles del vuelo

Salida 14:10 Miercoles , 12 febrero, Terminal 4 Barajas (MAD), Madrid - España

Llegada 00:15 Jueves, 13 febrero, Terminal 3 Dubai International Airport (DXB), Dubai - United Arab Emirates

Conexión

Salida 03:50 Jueves, 13 febrero, Terminal 3 Dubai International Airport (DXB), Dubai - United Arab Emirates

Llegada 11:40 Jueves, 13 febrero, Cape Town International Airport (CPT), Cape Town - South Africa

Tablas

- ❑ Mantienen los mismos elementos.
- ❑ Se han eliminado los atributos de presentación:
 - El elemento `table`, además de los globales, sólo tiene el atributo `border`.
 - ✓ `border = "1"`, indica que se trata de una tabla con contenido tabular.
 - ✓ Sin atributo `border` o con `border = ""` o `"0"` indica que se trata una tabla de maquetación
 - Para el elementos `td`, los únicos atributos (además de los globales) son `colspan`, `rowspan` y `headers`.
 - `col` y `colgroup`, además de los globales, sólo tienen el atributo `span`.
 - El elemento `th`, tiene además los atributos `scope` y `abbr`.
 - Los elementos `tfoot`, `tbody`, `thead`, `tr` y `caption`, sólo tiene atributos globales.

Tablas (II)

- ❑ Aunque el atributo `summary` ya no existe para el elemento `table`, por cuestiones de accesibilidad es necesario incluir una explicación del contenido de la tabla (Criterio de conformidad 1.3.1 de las WCAG 2.0).
- ❑ Se puede hacer:
 - Poniendo una explicación cerca de la tabla.
 - Utilizando el atributo `aria-describedby` en el elemento `table`.

```
<p id="desc">Explicación...</p>
<table aria-describedby="desc">
...
</table>
```
 - En el elemento `caption`.
 - Encerrando la tabla en un elemento `figure`, mediante el elemento `figcaption`.
 - Utilizando el nuevo elemento `details` y `summary`.

Tablas (II)

```
<table>
  <caption>Entradas y salidas del almacen
  <details>
    <summary>Descripción de la
      tabla</summary>
    <p>Esta tabla representa las entradas
      y salidas del almacen de los
      productos tijeras y chinchetas. Al
      final aparece el total de entradas
      y salidas de productos del
      almacen.</p>
  </details>
</caption>
<tr>
  <th rowspan="2" id="mes2">Meses</th>
  ...
  ...
</table>
```

Entradas y salidas del almacen				
► Descripción de la tabla				
Meses	Entradas		Salidas	
	Tijeras	Chinchetas	Tijeras	Chinchetas
Enero	234	332	343	456
Febrero	323	123	654	56
Marzo	345	543	343	343
Total	902	998	1200	855

Formularios

- ❑ HTML5 tiene nuevos elementos de formulario y nuevos atributos para los elementos.
 - No están totalmente implementados en todos los navegadores.
- ❑ Ahora los controles de los formularios no tienen por qué estar contenidos dentro del mismo formulario.
 - Los elementos `input`, `output`, `select`, `textarea`, `button`, `label`, `object`, y `fieldset` tienen un atributo `form` que puede tomar como valor el formulario al que pertenecen.
- ❑ Algunos controles se validan automáticamente (sin necesidad de Javascript).
 - La validación puede hacer referencia al formato del dato (por ejemplo, una fecha) o a los valores requeridos.
 - El atributo `novalidate` del elemento `form` impide la validación de los campos.
 - ✓ Los elementos `input` de tipo `submit` e `image` tienen un atributo `formnovalidate` que impiden la validación al enviar el formulario (sobreescriben el atributo `novalidate` del formulario).

Formularios

Nuevos valores del atributo type en el elemento input

❑ Valor tel.

- Indica un teléfono.
- Elimina los retornos de carro.
- No hace una validación especial.
 - ✓ Para validarlo podemos utilizar el atributo `maxlength` y el nuevo atributo `pattern`.
 - ✓ Por ejemplo:

```
<input type="tel" id="tel" name="telefono" pattern="[1-9][0-9]{8}"/>
```

Validaría un teléfono que comenzara por un número del 1 al 9 y luego tuviera 8 dígitos del 0 al nueve.
 - ✓ El valor de `pattern` es una expresión regular de Javascript (véase developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp).

❑ Valor search.

- Indica un cuadro de búsqueda.
- Sólo elimina los retornos de carro.

❑ Valor url.

- Valida un url absoluto (incluido el protocolo).
- Realmente sólo comprueba si incluye el protocolo (por ejemplo, `http://` o `ftp://`)

❑ Valor email.

- Comprueba que tiene una arroba entre dos grupos de caracteres.
- Se puede afinar más con el atributo `pattern`.

```
<input type="email" id="email" name="email" pattern=".*\@.*\..*">
```

Formularios

Nuevos valores del atributo `type` en el elemento `input` (II)

- ❑ Valores de `type` para fechas y horas:
 - Valor `date`.
 - ✓ Representa una fecha.
 - ✓ En los elementos `input` con `type="date"` los atributos `max` y `min` permiten indicar una fecha máxima y mínima en formato "aaaa-mm-dd".
 - Valor `time`.
 - ✓ Representa una hora.
 - ✓ Los elemento `input` de este tipo también tienen los atributos `max` y `min`.
 - ✓ Los elementos `input` de este tipo tienen el atributo `step`.
 - Indica el salto, en segundos, que se produce al pulsar las fechas en los widgets.
 - Los siguientes valores de atributos puede que no se mantengan en la versión definitiva:
 - ✓ `datetime`. Representa una fecha y hora.
 - ✓ `month`. Representa un mes.
 - ✓ `week`. Representa una semana de un año.
 - ✓ `datetime-local`. Representa una fecha y hora sin zona horaria.
 - En la actualidad, Firefox e IE no los soportan y Chrome, Opera y Safari los soportan parcialmente.

Formularios

Nuevos valores del atributo type en el elemento input (III)

Valor number.

- Permite introducir números.
- Algunos agentes de usuario presentan un widget de tipo up/down.
- Permite los atributos `max`, `min` y `step`.

Valor range.

- Permite introducir valores numéricos mediante un slider.
- Permite los atributos `max`, `min` y `step`.

Valor color.

- Permite especificar un color.
- Devuelve un color en formato `#RRVVAA`.

Formularios

Elemento output

- ❑ Se utiliza para representar el resultado de un cálculo.
- ❑ Para utilizarlo,
 - En un elemento `form`, el evento `oninput` contendría el valor calculado del contenido del elemento `output` del tipo:
`idControl.value=expresión`
 - Dentro del formulario deberían ir los controles que proporcionan los valores de entrada de las expresión.
 - Al modificar el valor de los controles del formulario se actualizaría el contenido del elemento `input`.
 - Atributo `for`.
 - ✓ Una cadena que especifica los controles que proporcionan los valores de entrada.
 - Atributo `form`.
 - ✓ Indica el formulario al que pertenece el control.

```
<form onsubmit="return false" oninput="resultado.value = parseInt(a.value) +
                                                                    parseInt(b.value) ">
  <input id="a" type="number" value="0"> +
  <input id="b" type="number" value="0"> =
  <output id="resultado"></output>
</form>
```

Año (input de tipo range):

2007

```
<form oninput="resultadoanno.value = anno.value">
  ...
  <input type="range" id="anno" name="anno" min="2000" max="2014" value="2000"/>
  <output id="resultadoanno" value="2000">2000</output> <output id="resultado"></output>
  ...
</form>
```

Formulario

Elemento `datalist`

- ❑ El elemento `datalist` permite presentar en un `input` de tipo `text` una serie de opciones predefinidas.
- ❑ El elemento tiene anidados elementos `option` con los valores que se presentan.
 - En cada elemento `option`:
 - ✓ El atributo `value` indica el valor que va a tomar el control.
 - ✓ El atributo `label` indica el valor que se mostrará.
 - Algunos agentes de usuario muestran los dos valores.
- ❑ En el elemento `input`, el atributo `list` permite relacionarlo con la lista.

Formulario

Elemento datalist (II)

```
<label for="provincia">Código provincia (input enlazado a un datalist):</label>
<input type="text" name="provincia" id="provincia" list="provincias"/>
<datalist id="provincias">
  <option value="02" label="Albacete"/>
  <option value="03" label="Alicante/Alacant"/>
  <option value="04" label="Almería"/>
  <option value="50" label="Zaragoza"/>
  <option value="51" label="Ceuta"/>
  <!-- Se puede utilizar este formato para option-->
  <option value="52">Melilla</option> <!-- Se puede utilizar este formato para option-->
</datalist>
```

Código provincia (input enlazado a un datalist):

a

- Albacete
- Alicante/Alacant
- Almería
- Zaragoza
- Ceuta
- Melilla

Código provincia (input enlazado a un datalist):

02	Albacete
03	Alicante/Alacant
04	Almería
50	Zaragoza
51	Ceuta
52	Melilla

Formularios

Elementos `meter` y `progress`

- ❑ El elemento `meter` representa una medida escalar mediante una barra.
 - Por ejemplo el uso de disco, la relevancia de una búsqueda, los votos de un candidato, etc.
 - Entre `<meter>` y `</meter>` debería haber una representación textual de los valores que representa la barra (por ejemplo, "3 de 10").
 - Atributos:
 - ✓ `value`. El valor actual de la barra (obligatorio).
 - ✓ `max` y `min`, los valores máximos y mínimos que representan la barra.
 - ✓ `high`, el valor mayor del rango posible.
 - ✓ `low`, el valor menor del rango posible.
 - ✓ `optimum`, el valor óptimo del rango.
- ❑ El elemento `progress` representa el progreso de una tarea.
 - El atributo `max` representa el valor máximo.
 - El atributo `value` representa el valor actual.

Formularios

Elementos meter y progress (II)

```
<p>Espacio en disco:  
<meter min="0" max="345" value="117"  
  title="gb">117 de 345 GB.</meter></p>
```

```
<form id="descargar">  
  <progress id="progreso" max="100"  
    value="0"></progress>  
  <span id="textoProgreso">0</span>  
  <input type="submit" id="iniciar"  
    value="Comenzar tarea"  
    onClick="actualizarProgreso()" />  
</form>
```

Ejemplo de meter

Espacio en disco: 

Ejemplo de meter

Espacio en disco: 117 de 345 GB.

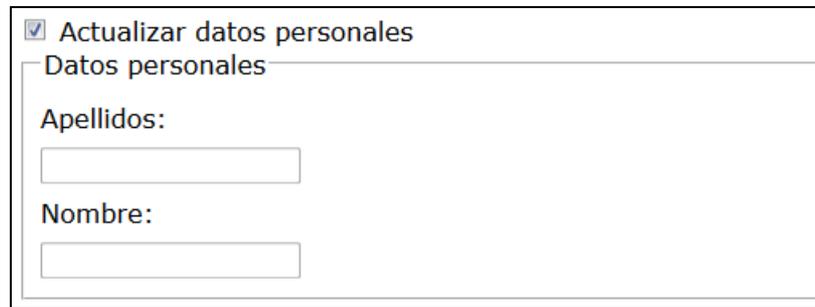
Ejemplo de progress

 58%

Formularios

Nuevos atributos

- ❑ El elemento `fieldset`, ahora incluye un atributo lógico `disabled` para desactivar todos los controles del elemento.



A screenshot of a web form. At the top, there is a checked checkbox labeled "Actualizar datos personales". Below it is a `fieldset` element with a legend "Datos personales". Inside the fieldset, there are two text input fields: one labeled "Apellidos:" and one labeled "Nombre:". The entire fieldset is visually disabled, indicated by a light gray background and a border.

```
<input id="actualizar" type="checkbox" name="actualizar" checked
      onchange="document.getElementById('datospersonales').disabled=this.checked"; />
<label class="check" for="actualizar">Actualizar datos personales</label>

<fieldset id="datospersonales" disabled="disabled">
  <legend>Datos personales</legend>
  <label for="apellidos">Apellidos:</label>
  <input type="text" id="apellidos" name="apellidos" />
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre"/>
</fieldset>
```

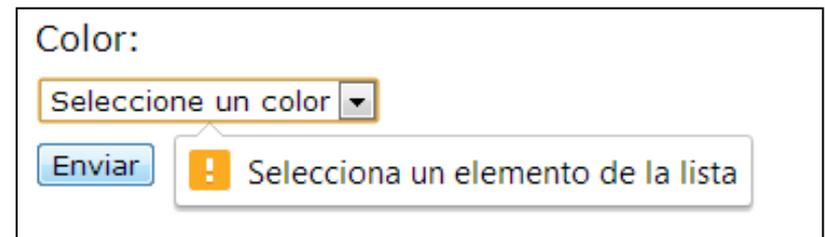
Formularios

Nuevos atributos (II)

□ Atributo `required`.

- Atributo lógico.
- Para los elementos `select`, `textarea` e `input` (menos los `input` de tipo `hidden`, `image` o cualquier tipo de botón).
 - ✓ En los elementos `select`, el primer elemento `option` de la lista debe tener un atributo `value` vacío.
- Indica que el campo es obligatorio.
 - ✓ Si el agente de usuario lo admite, aparecerá un mensaje indicando que el campo es obligatorio.

```
<label for="materiales">Color:</label>-  
<select name="colores" id="colores" required>  
  <option value="">Selecione un color</option>  
  <option value="rojo">Rojo</option>  
  <option value="verde">Verde</option>  
  <option value="azul">Azul</option>  
</select>
```



The screenshot shows a web form with the label "Color:". Below the label is a dropdown menu with the text "Selecione un color" and a downward arrow. To the right of the dropdown is a blue button labeled "Enviar". Below the dropdown and button is a validation message box with a red exclamation mark icon and the text "Selecciona un elemento de la lista".

Formularios

Nuevos atributos (III)

❑ Atributo `autofocus`.

- Atributo lógico.
- Para los elementos `select`, `textarea` e `input` (menos los `input` de tipo `hidden`, `image` o cualquier tipo de botón).
- Al cargar la página el control con el atributo `autofocus` será el que recibe el foco.
 - ✓ Sólo puede haber un control con el atributo `autofocus` activado.
 - ✓ El orden del foco contará a partir de ese elemento.

❑ Atributo `placeholder`.

- Para los `input` de tipo `text` y los elemento `textarea`.
- Se utiliza para mostrar un mensaje (un ejemplo del formato, una pequeña ayuda) en el cuadro de texto.

Formularios

Nuevos atributos (IV)

❑ Atributo `autocomplete`.

- Atributo lógico.
- Para los `input` de tipo `text` y los elemento `textarea`.
- El valor del atributo "on" activa el relleno automático.
 - ✓ El navegador debería tener activada la opción.
- El valor del atributo "off" lo desactiva.
 - ✓ Por ejemplo para datos sensibles, como contraseñas.

❑ Atributo `contenteditable`.

- No se trata de un atributo específico para controles de formulario.
 - ✓ Se puede aplicar a cualquier elemento.
- Hace que el contenido se pueda modificar.
- Se puede acceder al nuevo contenido, por ejemplo, con `elemento.textContent`.

Formularios

Página de prueba

- ❑ Todos estos elementos se pueden probar en el navegador móvil o de escritorio en:

<http://goo.gl/T0qtr0>.



Compatibilidad

- ❑ Al tratarse de un borrador, no todas las versiones de los navegadores soportan las nuevas características de HTML5 y ningún navegador las soporta plenamente.
 - Por ejemplo, para los nuevos elementos semánticos:
 - ✓ IE empieza a soportar las características a partir de la versión 9, Firefox a partir de la 4, Chrome a partir de la 6, Opera a partir de la 11 y Safari a partir de la 5.
 - ✓ Se trata de versiones anteriores a 2010.
 - ✓ Esto supone dejar fuera, aproximadamente, al 10% de los usuarios.
- ❑ Se puede ver el soporte HTML5 de los navegadores en:
 - HTML5test (html5test.com/)
 - Can I use (caniuse.com/)
 - La referencia de HTML5 en w3Schools (www.w3schools.com/tags/default.asp) ofrece, por cada elemento, que versiones de navegadores actuales la soportan.

Compatibilidad (II)

- ❑ Existen bibliotecas Javascript que permiten utilizar algunos elementos de HTML5 en versiones antiguas.
 - En general, casi todos tratan de mejorar el funcionamiento de IE.
 - ✓ El número de usuarios de IE que utilizan versiones antiguas es muy superior al de otros navegadores.
- ❑ html5shiv es un script Javascript que permite reconocer la existencia de los nuevos elementos html.
- ❑ modernizr es una biblioteca Javascript que permite determinar si nuestro navegador soporta características de HTML5 y CSS.
 - Puede incluir también html5shiv.

Compatibilidad

html5shiv

- ❑ No añada las funcionalidades nuevas.
- ❑ Simplemente crea en nuestra página los nuevos elementos.
 - De esta forma las reglas CSS podrán reconocer elementos como `header`, `footer`, `aside`, `section`, `article`, etc.
- ❑ Puede dar soporte tanto a IE como a versiones antiguas de otros navegadores.
- ❑ Se puede descargar de code.google.com/p/html5shim/ o github.com/aFarkas/html5shiv.
- ❑ ¿Cómo se utiliza?
 - En el elemento `head`, después de la declaración de estilos incluirlo en un "comentario condicional"
 - Los comentarios condicionales sólo son reconocidos por IE.
 - ✓ Sólo se cargaría el script con versiones anteriores a la 9.

```
<!-- se supone que el script está en la carpeta dist -->
<!--[if lt IE 9]>
    <script src="dist/html5shiv.js"></script>
<![endif]-->
```

Compatibilidad

modernizr

- ❑ Por sí sólo no añade nuevos elementos o nuevas funcionalidades a la página.
- ❑ Se puede descargar de modernizr.com/.
 - Se puede descargar una versión de desarrollo con todas las posibilidades y paquetes.
 - También se puede crear un built sólo con los elementos que nos interesen.
- ❑ Es un paquete para detectar las capacidades de nuestro navegador.
- ❑ También facilita la “mejora progresiva” para compatibilizar estilos con CSS3 en navegadores que no lo soporten.
 - Aplicar estilos alternativos para navegadores que no soporten CSS3.

Compatibilidad modernizr (II)

- ❑ modernizr también puede incluir html5shiv.
- ❑ En modernizr.com/download/ se puede crear un built de modernizr sólo con html5shiv.
 - Seleccionar sólo las opciones por omisión:
 - ✓ html5shiv v3.7
 - ✓ Modernizr.load
 - ✓ CSS Classes
 - Seleccionar el botón "Generate"
 - Seleccionar el botón "Download" y asignar un nombre al built (por ejemplo modernizr.js).
 - Para incluir modernizr hay que cargar el script en el elemento `head`, después de las declaraciones de estilo.

```
<!-- se supone que el script se ha llamado modernizr.js-->  
<script src="modernizr.js"></script>
```

Compatibilidad

Pruebas

□ Ejemplo en IE8 con modernizr

Ejemplo con modernizr

[Enlace 1](#) [Enlace 2](#) [Enlace 3](#) [Enlace 4](#)

<h3>Artículo 1</h3> <hr/> <h4>Sección 1.1</h4> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p> <h4>Sección 1.2</h4> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p>	<h3>Artículo 2</h3> <hr/> <h4>Sección 2.1</h4> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p> <h4>Sección 2.2</h4> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p>	<h3>Enlaces relacionados</h3> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p>
---	---	---

© Luis Rodríguez Baena, 2014

Compatibilidad

Pruebas

□ Ejemplo en IE8 sin modernizr

Ejemplo sin nada

[Enlace 1](#) [Enlace 2](#) [Enlace 3](#) [Enlace 4](#)

Artículo 1

Sección 1.1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

Sección 1.2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

Artículo 2

Sección 2.1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

Compatibilidad

Pruebas (II)

❑ Ejemplo sin utilizar ninguna herramienta:

- <http://goo.gl/vBHqj3>



❑ Ejemplo con modernizr.

- <http://goo.gl/V2qMjm>



Compatibilidad

Polyfills

- ❑ Son correctores de compatibilidad (shim) en Javascript que replican las API estándar para navegadores más antiguos.
 - En github.com/Modernizr/Modernizr/wiki/HTML5-Cross-browser-Polyfills se puede ver una lista de polyfills para algunas de las características de HTML5.
 - Por ejemplo, para que esté disponible la función de video, se puede utilizar el polyfill html5media (<https://github.com/etianen/html5media>).

```
<script src="http://api.html5media.info/1.1.6/html5media.min.js"></script>
```

- ❑ Se puede ver un ejemplo en goo.gl/VWZzaE.

