

# PCW

---

## PROGRAMACIÓN DEL CLIENTE WEB

---

### Tema 02 - HTML



Dept. de Ciència de la Computació i Intel·ligència *a*rtificial  
Dpto. de Ciencia de la Computación e Inteligencia *a*rtificial



Universitat d'Alacant  
Universidad de Alicante

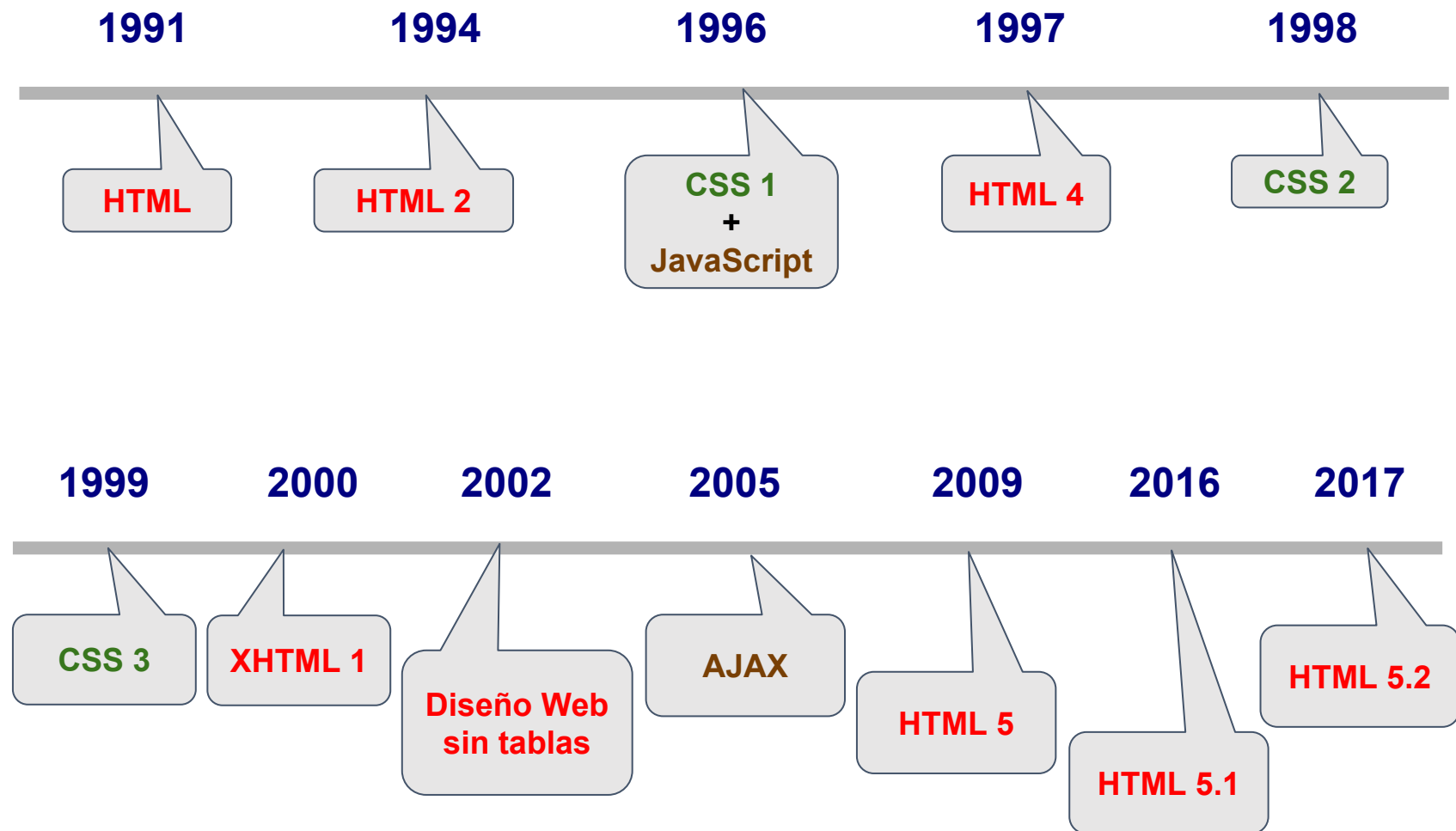
# HTML

# HTML

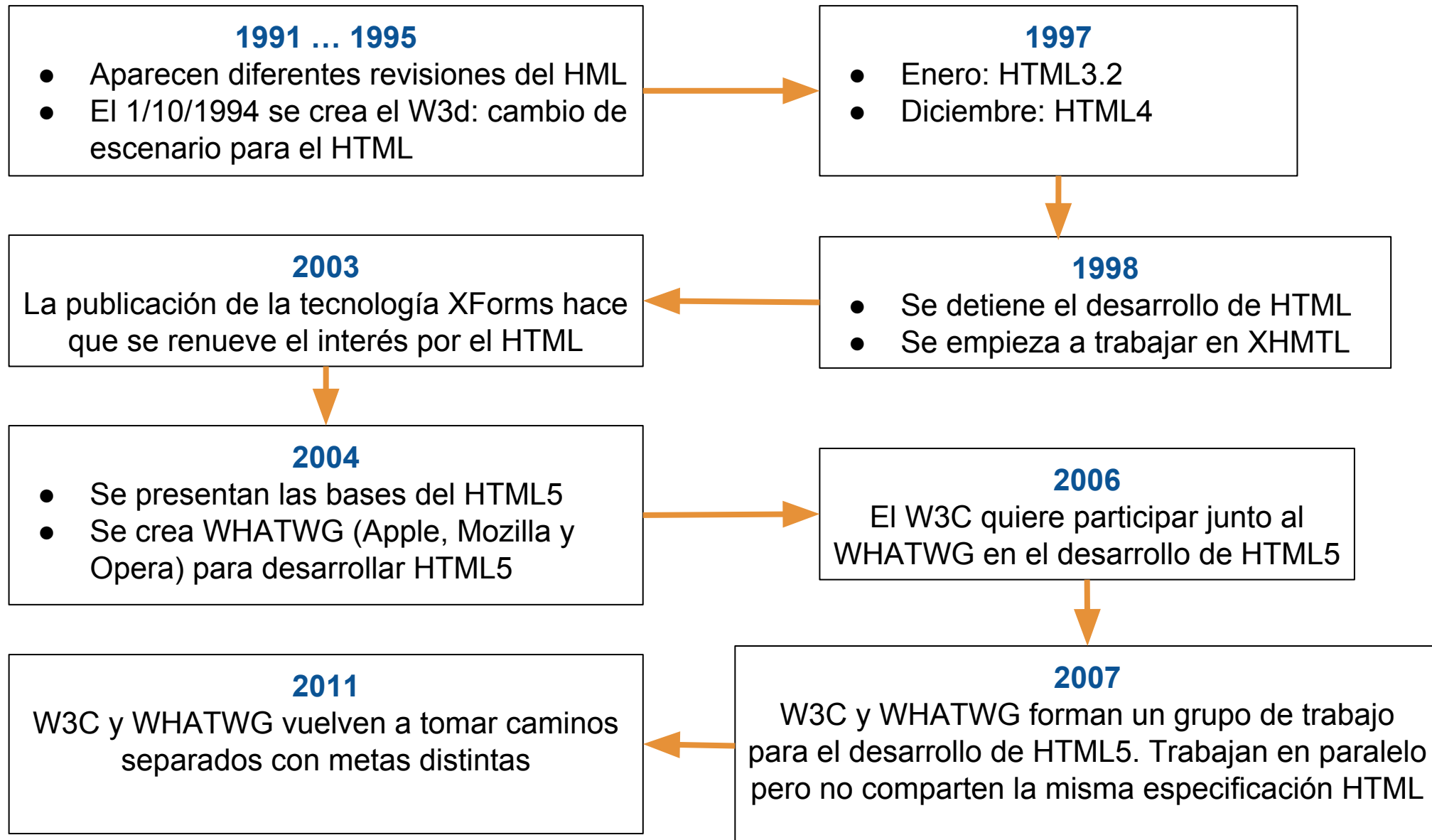
1. Introducción.
2. Sintaxis HTML5.
3. Elementos HTML5.
4. Estructura de un documento HTML5.
5. Otros elementos importantes en HTML5.
6. Formularios HTML5.
7. Elementos `<audio>` y `<video>`.
8. Características interesantes de html

# Introducción

# Evolución de las Tecnologías Web



## Evolución del HTML



## Especificaciones de HTML

### **WHATWG**

**WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP**

<https://html.spec.whatwg.org/multipage/>

### **W3C**

**WORLD WIDE WEB CONSORTIUM**

<http://w3c.github.io/html/>

## ¿Qué es exactamente HTML5?

- NO es sólo la quinta revisión del lenguaje base de la WWW: el HTML
- Engloba un conjunto de nuevas tecnologías para la web (*Video, audio, canvas, drag and drop, ...*)
- Pretende proporcionar soporte para el desarrollo de aplicaciones web.
- Recomendación W3C (28/10/2014)
- HTML 5.1 => Recomendación W3C (01/11/2016).
- HTML 5.2 => Recomendación W3C (14/12/2017).
- HTML 5.3 => Recomendación W3C durante 2018.





# Tecnologías HTML5



## Semantics

- Aporta significado a la estructura
- Conjunto más rico en etiquetas: nuevas etiquetas semánticas



## Offline & Storage

- Las aplicaciones web funcionan más rápido e, incluso, sin conexión gracias a *HTML5 App Cache*, *Local Storage*, *Indexed DB* y *File API*



## Device Access

- *Geolocation API*
- Captura de audio/video
- Datos locales: contactos, eventos, ...



## Connectivity

- Mejor comunicación y más rápida
- Chats en tiempo real
- *Web Sockets*
- *Server-Sent Events (SSE)*



## Multimedia

- Elementos *audio* y *video* compatibles con aplicaciones y sitios web



## 3D, Graphics, Effects

- *SVG*, *Canvas*, *Web GL*, *CSS3 3D*



## Performance & Integration

- Funcionamiento más rápido de aplicaciones web y contenido web dinámico
- *Web Workers*
- *XMLHttpRequest 2*



## CSS3

- Gran diversidad de estilos y efectos, mejorando la aplicación web sin sacrificar la estructura o el rendimiento
- Tipografía flexible: *Web Open Font Format (WOFF)*

# Sintaxis HTML5

## Sintaxis HTML5

- **Los nombres de los elementos** (o etiquetas) HTML y sus atributos son invariantes a mayúsculas/minúsculas. Por ejemplo: `<div>`, `<DIV>`, `<dIV>` o `<Div>` es el mismo elemento HTML.
- Los valores de los **atributos** se pueden escribir entre comillas, simples o dobles; o también sin ellas, siempre y cuando no contengan un espacio en blanco ni los caracteres `"`, `'`, ```, `=`, `<`, `>`. Incluso, se puede dejar únicamente el nombre del atributo si su valor es una cadena vacía.
- Los elementos *void* pueden llevar (`</>`) o no (`<>`) la barra final.
- Algunos elementos HTML se pueden utilizar sin elemento de cierre en determinadas circunstancias.

# Sintaxis HTML5

Ejemplo (aunque no a seguir) de uso válido de sintaxis HTML5:

```
<!DoCtYpe hTm1>
<Html lAng='es'>
  <head>
    <tITle>Sintaxis poco cuidada, pero
      v&aacute;lida</title>
  </HEAD>
  <boDY>
    <p>Ah&iacute; va!.
    <P><bR/><BR>Los elementos <I>void</i> pueden tener
      barra final o no.</p>
    <ol>
      <LI class=green>Butter
      <li CLASs="red">Bread</LI>
      <lI claSS=blue>Milk
    </OL>
  </BodY>
</htML>
```

## Sintaxis HTML5

Otro ejemplo de sintaxis HTML5 válida, aunque no recomendable:

```
...
<table>
  <thead>
    <tr><th>ID<th>Measurement<th>Average<th>Maximum
  <tbody>
    <tr><td><th scope=rowgroup>Cats<td><td>
    <tr><td>93<th scope=row>Legs<td>3.5<td>4
    <tr><td>10<th scope=row>Tails<td>1<td>1
  <tbody>
    <tr><td><th scope=rowgroup>English
      speakers<td><td>
    <tr><td>32<th scope=row>Legs<td>2.67<td>4
    <tr><td>35<th scope=row>Tails<td>0.33<td>1
</table>
...
```

## Sintaxis HTML5

- Requiere la especificación del **tipo de documento** mediante una única declaración *DOCTYPE*.

**<!DOCTYPE html>**

La declaración se ha simplificado con respecto a versiones anteriores de html. Es invariante a mayúsculas o minúsculas, aunque todo lo que no sea escribirla en mayúsculas se considera una *declaración obsoleta*. Su única función es indicar al navegador que el documento se ha escrito utilizando la sintaxis HTML5.

### Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    ...
```

## Sintaxis HTML5

- Aunque no es obligatorio, se recomienda especificar **la codificación de caracteres**. En HTML5 se ha simplificado al máximo y se realiza mediante un elemento *meta*:

**`<meta charset="utf-8">`**

Debe estar dentro de los 1024 primeros bytes del documento. Normalmente es la primera línea tras el elemento `<head>`.

### Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    ...
```

## Sintaxis HTML5

- Los elementos `<meta>` proporcionan información en forma de **metadatos** sobre el documento html. Se declaran en el `<head>` del documento.

Algunos de los elementos `<meta>` más importantes son:

- Para especificar la codificación de caracteres:  
`<meta charset="utf-8"/>`
- Breve resumen del contenido de la página:  
`<meta name="description" content="Texto resumen">`
- Controlar cómo indexan y recorren el sitio web los motores de búsqueda:  
`<meta name="robots" content="noindex, nofollow">`
- Evitar que el navegador redimensione el contenido en dispositivos móviles:  
`<meta name="viewport" content="width=device-width, initial-scale=1">`
- Evitar que el navegador recargue la página de la cache:  
`<meta http-equiv="cache-control" content="no-cache">`



## Sintaxis HTML5

- **Especificación del idioma.** En HTML5 también se ha simplificado y se realiza mediante el atributo *lang* del elemento *html*:

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

El valor del atributo *lang* es el código ISO 639-1 del idioma en el que se encuentra el contenido del elemento *html* al que se aplica el atributo.

### Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    ...
```

## Sintaxis HTML5

- Otras simplificaciones.

- Etiqueta `<script>`. Si el lenguaje de script es JavaScript, no es necesario especificar el atributo *type="text/javascript"*, puesto que es el lenguaje de script que se asume por defecto.
- Etiqueta `<link>`. Tampoco es necesario especificar el atributo *type* cuando se enlaza a un archivo CSS.

### Ejemplo:

```
<!DOCTYPE html>
  <html lang="es">
    <head>
      <meta charset="utf-8">
      <link rel="stylesheet" href="estilo.css">
      <script src="archivo.js"></script>
      ...
```

## Sintaxis HTML5

- **Caracteres especiales (á, ü, ñ, ¿, ...).** Es conveniente escribirlos mediante su *referencia numérica* o su *nombre de entidad*.

Carácter español	Nombre de entidad	Referencia numérica	Carácter español	Nombre de entidad	Referencia numérica
Á	&Aacute;	&#193;	Ú	&Uacute;	&#218;
á	&aacute;	&#225;	ú	&uacute;	&#250;
É	&Eacute;	&#201;	Ü	&Uuml;	&#220;
é	&eacute;	&#233;	ü	&uuml;	&#252;
Í	&Iacute;	&#205;	«	&laquo;	&#171;
í	&iacute;	&#237;	»	&raquo;	&#187;
Ñ	&Ntilde;	&#209;	¿	&iquest;	&#191;
ñ	&ntilde;	&#241;	¡	&iexcl;	&#161;
Ó	&Oacute;	&#211;	€	&euro;	&#128;
ó	&oacute;	&#243;	Pts		&#8359;

## Sintaxis HTML5

### Recomendaciones:

- Escribir los nombres de los elementos y sus atributos en minúsculas.
- Escribir los valores de los atributos de los elementos HTML entre comillas dobles.
- Escribir los elementos de cierre del correspondiente elemento de apertura, aunque no sea necesario.
- No utilizar barra final en los elementos *void* de HTML.
- Escribir los caracteres especiales mediante su código de referencia o nombre de entidad.
- Usar un servicio de validación de documentos HTML:  
<http://validator.w3.org/>

# Elementos HTML5

## Elementos HTML5

- **Algunos** elementos **se mantienen** de la especificación anterior de html.
- **Otros** elementos **se han eliminado** por diversas razones y, aunque siguen siendo interpretados por los navegadores, no pertenecen a la especificación HTML5.
- Por último, **se han añadido nuevos** elementos html para, entre otras cosas, añadir más semántica a las distintas partes del documento; estandarizar el tratamiento de elementos como vídeo o audio, haciéndolo independiente de plugins externos de terceros; y mejorar la experiencia de usuario (UX).
- La lista completa de elementos con su descripción se puede encontrar en:

<https://html.spec.whatwg.org/multipage/semantics.html#semantics>

## Elementos HTML5

- El origen de los nuevos elementos que incorpora la especificación HTML5 es diverso:
  - ✓ Algunos se importaron de XHTML2
  - ✓ Otros se han creado como resultado de un estudio realizado por Google y Opera entre otros, sobre el uso de nombres para los atributos *class* e *id*, eligiendo aquéllos más comúnmente utilizados. Algunos ejemplos se pueden ver en la siguiente tabla:

Antes de HTML5	Nueva etiqueta en HTML5
<code>&lt;div id="header"&gt;</code>	<code>&lt;header&gt;</code>
<code>&lt;div id="nav"&gt;</code>	<code>&lt;nav&gt;</code>
<code>&lt;div id="footer"&gt;</code>	<code>&lt;footer&gt;</code>
<code>&lt;div class="article"&gt;</code>	<code>&lt;article&gt;</code>
<code>&lt;div class="section"&gt;</code>	<code>&lt;section&gt;</code>

## Clasificación de los elementos en HTML5

Los elementos o etiquetas HTML se pueden agrupar según su funcionalidad de la siguiente manera:

- El elemento *documento*: **html**. Representa la raíz de un documento HTML.
- Metadatos del documento.
  - **head**. Representa una colección de metadatos de documento.
  - **title**. Representa el título o nombre del documento.
  - **base**. Permite especificar la *URL* base del documento.
  - **link**. Permite enlazar el documento con otros recursos.
  - **meta**. Representa otra serie de metadatos que no se pueden expresar con los elementos `title`, `base`, `link`, `style` o `script`.
  - **style**. Permite incluir información de estilo embebida en el documento.



# Clasificación de los elementos en HTML5

- Secciones.
  - **body**. Representa el contenido del documento.
  - **article**. Representa una composición completa o, autocontenida, en un documento, página, aplicación o *site*. Es decir, contenido reusable o redistribuible de forma independiente.
  - **section**. Representa una sección genérica de un documento o aplicación. Es decir, agrupa contenido de una misma temática.
  - **nav**. Representa una sección de una página que enlaza a otras páginas o a otras partes dentro de la página: una sección con enlaces de navegación.
  - **aside**. Representa una sección de una página cuyo contenido está tangencialmente relacionado con el contenido principal de la página, pero considerado de forma separada. Es por ello que podría eliminarse y no influiría en el contenido de la página.
  - **h1**, **h2**, **h3**, **h4**, **h5**, **h6**. Representan encabezados para las secciones en las que se incluyan. **h1** es el de mayor importancia y **h6** es el de menor importancia.

# Clasificación de los elementos en HTML5

- Secciones (y //).
  - **header**. Representa contenido introductorio para el elemento que lo contiene.
  - **footer**. Representa un pie para el elemento que lo contiene.
  - **address**. Representa la información de contacto para el elemento **article** o **body** que lo contiene.
- Agrupación de contenido.
  - **p**. Representa un párrafo.
  - **hr**. Representa un cambio de temática en el contenido.
  - **pre**. Representa un bloque de texto preformateado.
  - **blockquote**. Representa contenido citado de otra fuente.
  - **ol**. Representa una lista ordenada de artículos.
  - **ul**. Representa una lista desordenada de artículos.
  - **li**. Representa un artículo de una lista.
  - **dl**. Representa una lista de asociación de pares nombre-valor.
  - **dt**. Representa el nombre del par de una lista de asociación.

## Clasificación de los elementos en HTML5

- Agrupación de contenido (y //).
  - **dd**. Representa la descripción o valor del par de una lista de asociación.
  - **figure**. Representa un contenido autocontenido, opcionalmente con una descripción o leyenda, como fotos, diagramas, etc.
  - **figcaption**. Representa una descripción o leyenda para el elemento **figure** que lo contiene.
  - **main**. Representa el contenido principal del cuerpo (**<body>**) de un documento o aplicación.
  - **div**. No tiene significado semántico. Se suele utilizar para agrupar elementos consecutivos.
- Elementos semánticos a nivel de texto.
  - **a**. Si tiene un atributo **href**, representa un hiperenlace; si no, representa un marcador de posición al que se puede llegar mediante un hiperenlace.

## Clasificación de los elementos en HTML5

- Elementos semánticos a nivel de texto (y //).
  - **em**. Se utiliza para enfatizar su contenido.
  - **strong**. Se utiliza para aportar una fuerte importancia, seriedad, o urgencia sobre su contenido.
  - **small**. Representa para mostrar texto de tipo *letra pequeña* tal como advertencias, responsabilidades legales, etc.
  - **s**. Representa contenido que ya no es exacto o ya no es relevante.
  - **cite**. Representa una referencia a un trabajo creativo. Debe incluir el título del trabajo, o el nombre del autor, o una *URL* de referencia, etc.
  - **q**. Representa una cita inline, es decir, dentro de un texto.
  - **dfn**. Representa la definición de un término.
  - **abbr**. Representa una abreviación o acrónimo.
  - **ruby**. Se utiliza para anotaciones [ruby](#).
  - **rb**. Representa el texto base de una anotación *ruby*.

## Clasificación de los elementos en HTML5

- Elementos semánticos a nivel de texto (y III).
  - **rt**. Representa el texto *ruby* de una anotación *ruby*.
  - **rtc**. Representa un contenedor de texto *ruby* para el componente de texto *ruby* en una anotación *ruby*.
  - **rp**. Se utiliza para mostrar texto alternativo cuando el navegador no soporta anotaciones *ruby*.
  - **data**. Se utiliza para proporcionar, en su atributo **value**, un formato *legible para máquina* de un contenido que, por su naturaleza, se muestra al usuario en un formato distinto.
  - **time**. Se utiliza para proporcionar, en su atributo **datetime**, un formato *legible para máquina* de un contenido de fecha, hora, fecha/hora, duración de tiempo, huso horario o semana del año.
  - **code**. Representa un fragmento de código fuente de un programa.
  - **var**. Representa una variable en una expresión matemática o en un contexto de programación.

## Clasificación de los elementos en HTML5

- Elementos semánticos a nivel de texto (y /V).
  - **samp**. Representa una muestra o un resultado citado de otro programa o sistema de computación.
  - **kbd**. Representa información proveniente del usuario, normalmente obtenida a través del teclado.
  - **sub**, **sup**. Representan un subíndice y un superíndice, respectivamente.
  - **i**. Se utiliza para destacar texto dentro de un párrafo al ser distinto de su contexto por algún motivo como, por ejemplo, tratarse de un término técnico, una frase idiomática, etc.
  - **b**. Se utiliza para destacar texto dentro de un párrafo sobre el que se quiere atraer la atención por cuestiones funcionales, sin que ello implique importancia adicional. Ejemplos de uso serían palabras claves en el resumen de un documento, nombres de productos en una revisión, los comandos que admite un software en un manual de instrucciones, etc.

## Clasificación de los elementos en HTML5

- Elementos semánticos a nivel de texto (y V).
  - **u**. Se utiliza para subrayar palabras con errores tipográficos.
  - **mark**. Se utiliza para resaltar o destacar texto en un documento por su relevancia en otro contexto.
  - **bdi**. Se utiliza para mostrar texto aislado de su entorno por tratarse de texto formateado bidireccionalmente.
  - **bdo**. Se utiliza para controlar cómo se formatea la direccionalidad de un texto, sobrescribiendo la que tuviera por defecto.
  - **span**. No tiene significado semántico. Se utiliza para aplicar estilo CSS sin ningún propósito semántico.
  - **br**. Representa un salto de línea.
  - **wbr**. Representa la posibilidad de aplicar un salto de línea.

## Clasificación de los elementos en HTML5

- Ediciones del documento.
  - **ins**. Representa contenido añadido al documento.
  - **del**. Representa contenido eliminado del documento.
- Contenido embebido.
  - **picture**. Es un contenedor que proporciona múltiples fuentes para el elemento **img** que contiene, permitiendo que el navegador elija el recurso adecuado en función de la resolución de la pantalla, densidad de pixel, formato de la imagen, etc.
  - **source** (cuando se utiliza con el elemento **picture**). Permite múltiples conjuntos de fuentes alternativas para elementos **img**.
  - **img**. Representa una imagen y el contenido alternativo.
  - **iframe**. Representa un contexto de navegación anidado.
  - **embed**. Representa un punto de integración con aplicaciones externas o contenido interactivo, normalmente no HTML.
  - **object**. Representa un recurso externo que se tratará en función de su naturaleza.



## Clasificación de los elementos en HTML5

- Contenido embebido (y II).
  - **param**. Define parámetros para *plugins* invocados por un elemento **object**.
  - **video**. Se utiliza para reproducir vídeos o películas y archivos de audio con leyenda.
  - **audio**. Representa un sonido o una transmisión de audio.
  - **source**. Permite especificar múltiples recursos multimedia alternativos para los elementos multimedia.
  - **track**. Permite especificar recursos de texto externos y explícitos para elementos multimedia.
- Datos tabulares.
  - **table**. Representa datos con más de una dimensión, en forma de tabla.
  - **caption**. Representa el título del elemento **table** que lo contiene.

## Clasificación de los elementos en HTML5

- Datos tabulares (y //).
  - **colgroup**. Representa un grupo de una o más columnas en el elemento **table** que lo contiene.
  - **col**. Representa una o más columnas en el elemento **colgroup** que lo contiene.
  - **tbody**. Representa un bloque de filas que es el cuerpo del elemento **table** que lo contiene.
  - **thead**. Representa el bloque de filas consistente en las etiquetas (cabeceras) de las columnas del elemento **table** que lo contiene.
  - **tfoot**. Representa un grupo de filas que se corresponden con el pie de las columnas del elemento **table** que lo contiene.
  - **tr**. Representa una fila de celdas en un elemento **table**.
  - **td**. Representa una celda en un elemento **table**.
  - **th**. Representa una celda cabecera de columna en un elemento **table**.

## Clasificación de los elementos en HTML5

- Formularios.
  - **form**. Representa una colección de elementos de formulario, algunos de los cuales pueden ser valores editables que se pueden enviar a un servidor para su procesamiento.
  - **label**. Representa un título en una interfaz de usuario. Se puede asociar a un control de formulario específico de forma explícita.
  - **input**. Representa un campo de datos de un tipo determinado. Generalmente lleva un control de formulario asociado para permitir al usuario editar los datos.
  - **button**. Representa un botón etiquetado por su contenido.
  - **select**. Representa un control para seleccionar una entre un conjunto de opciones.
  - **datalist**. Representa un conjunto de elementos **option** que representan opciones predefinidas para otros controles.
  - **optgroup**. Representa un grupo de elementos **option** con una etiqueta común.

## Clasificación de los elementos en HTML5

- Formularios (y //).
  - **option**. Representa una opción en un elemento **select** o en una lista de sugerencias en un elemento **datalist**.
  - **textarea**. Representa un control de edición de texto plano multilínea.
  - **output**. Representa el resultado de un cálculo o una acción realizada.
  - **progress**. Representa el progreso de realización de una tarea.
  - **meter**. Representa una medición escalar dentro de un rango conocido.
  - **fieldset**. Representa un conjunto de controles de formulario agrupados bajo un nombre común.
  - **legend**. Representa un título para el resto de contenidos el elemento **fieldset** que lo contiene.

## Clasificación de los elementos en HTML5

- Elementos interactivos.
  - **details**. Representa un elemento para mostrar información adicional al usuario y que permite mostrar u ocultar su contenido.
  - **summary**. Representa un resumen, título, o leyenda para el resto del contenido del elemento **details** padre que lo contiene.
  - ~~**menu**. Representa un grupo de órdenes en un menú contextual *pop-up*.~~\*
  - ~~**menuitem**. Representa una orden de un menú contextual *pop-up* perteneciente al elemento **menu** que lo contiene.~~\*

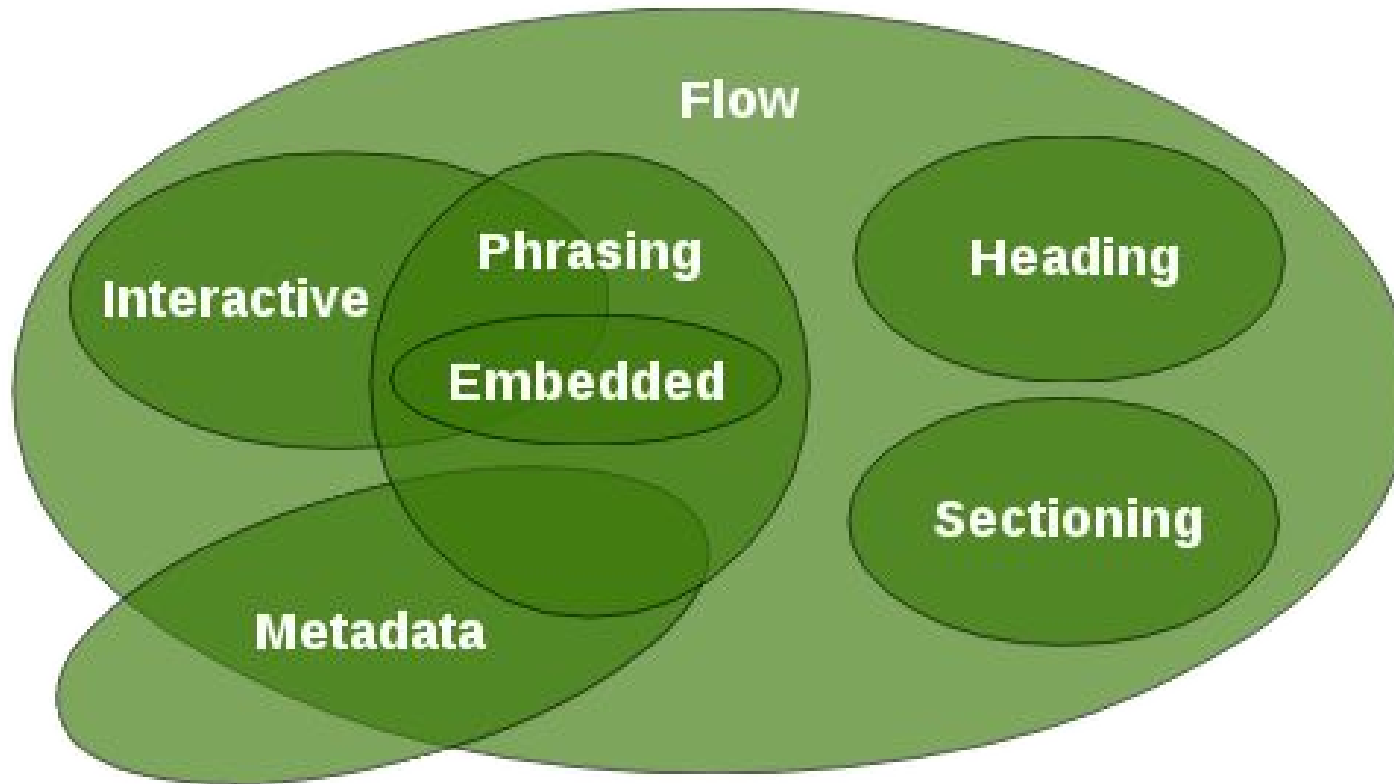
\* *Deprecated* desde la publicación de HTML 5.2 W3C Recommendation, el 14 de diciembre de 2018.

## Clasificación de los elementos en HTML5

- Elementos de *script*. Permiten a los desarrolladores añadir interactividad al documento.
  - **script**. Permite incluir código *script* (dinámico), por lo general *JavaScript*, y bloques de datos en los documentos.
  - **noscript**. Permite presentar un html diferente en aquellos navegadores en los que no soporten código *script* o lo tengan deshabilitado.
  - **template**. Se utiliza para declarar fragmentos de HTML que se pueden clonar e insertar en el documento mediante código *script*.
  - **canvas**. Proporciona código *script* junto con un *canvas* (área de dibujo) de tipo bitmap dependiente de la resolución. Este *canvas* se puede utilizar para dibujar cualquier contenido visual: gráficas, gráficos de juegos, fotos, etc, sobre la marcha.

## Clasificación de los elementos en HTML5

El WHATWG agrupa las etiquetas o elementos de HTML5 en las siguientes categorías, según el contenido.



## Clasificación de los elementos en HTML5

- **Metadatos** (*metadata content*). Contenido que establece la presentación o comportamiento del resto del contenido del documento; o la relación de éste con otros documentos.

Etiquetas: base, command, link, meta, noscript, script, style, title.

- **Encabezados** (*heading content*). Elementos que definen la cabecera de una sección, tanto si aparece explícitamente mediante algún elemento de la categoría anterior, o bien implícitamente con el uso de estos elementos en si.

Etiquetas: h1, h2, h3, h4, h5, h6, hgroup.

- **Secciones** (*sectioning content*). Son los elementos que pueden contener una cabecera y un pie.

Etiquetas: article, aside, nav, section.



## Clasificación de los elementos en HTML5

- **Contenido de flujo** (*flow content*). Es el texto y la mayoría de los elementos que se utilizan en el cuerpo de los documentos.
- Etiquetas: a, abbr, address, area\*, article, aside, audio, b, bdi, bdo, br, button, canvas, cite, code, command, data, datalist, del, details, dfn, dialog, div, dl, em, embed, fieldset, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hr, i, iframe, img, input, ins, kbd, keygen, label, link\*, main, map, mark, math, menu, meta\*, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style\*, sub, svg, table, textarea, time, u, ul, var, video, wbr.

\* Estos elementos se considerarán contenido de flujo si cumplen ciertas condiciones

## Clasificación de los elementos en HTML5

- **Contenido de texto o redacción** (*phrasing content*). Es el texto del documento, así como los elementos utilizados para su marcado.

Etiquetas: a\*, abbr, area, audio, b, bdi, bdo, br, button, canvas, cite, code, command, data, datalist, del\*, dfn, em, embed, i, iframe, img, input, ins\*, kbd, keygen, label, link\*, map\*, mark, math, meta\*, meter, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, textarea, time, u, var, video, wbr

- \* Estos elementos se considerarán contenido de texto si cumplen ciertas condiciones

## Clasificación de los elementos en HTML5

- **Contenido embebido** (*embedded content*). Es el contenido que importa otra fuente dentro del documento.

Etiquetas: audio, canvas, embed, iframe, img, math, object, svg, video.

- **Contenido interactivo** (*interactive content*). Contenido destinado especialmente a la interacción con el usuario.

Etiquetas: a, audio\*, button, details, embed, iframe, img\*, input\*, keygen, label, menu\*, object\*, select, textarea, video\*.

- \* Estos elementos se considerarán contenido interactivo si cumplen ciertas condiciones

# Estructura de un documento HTML5

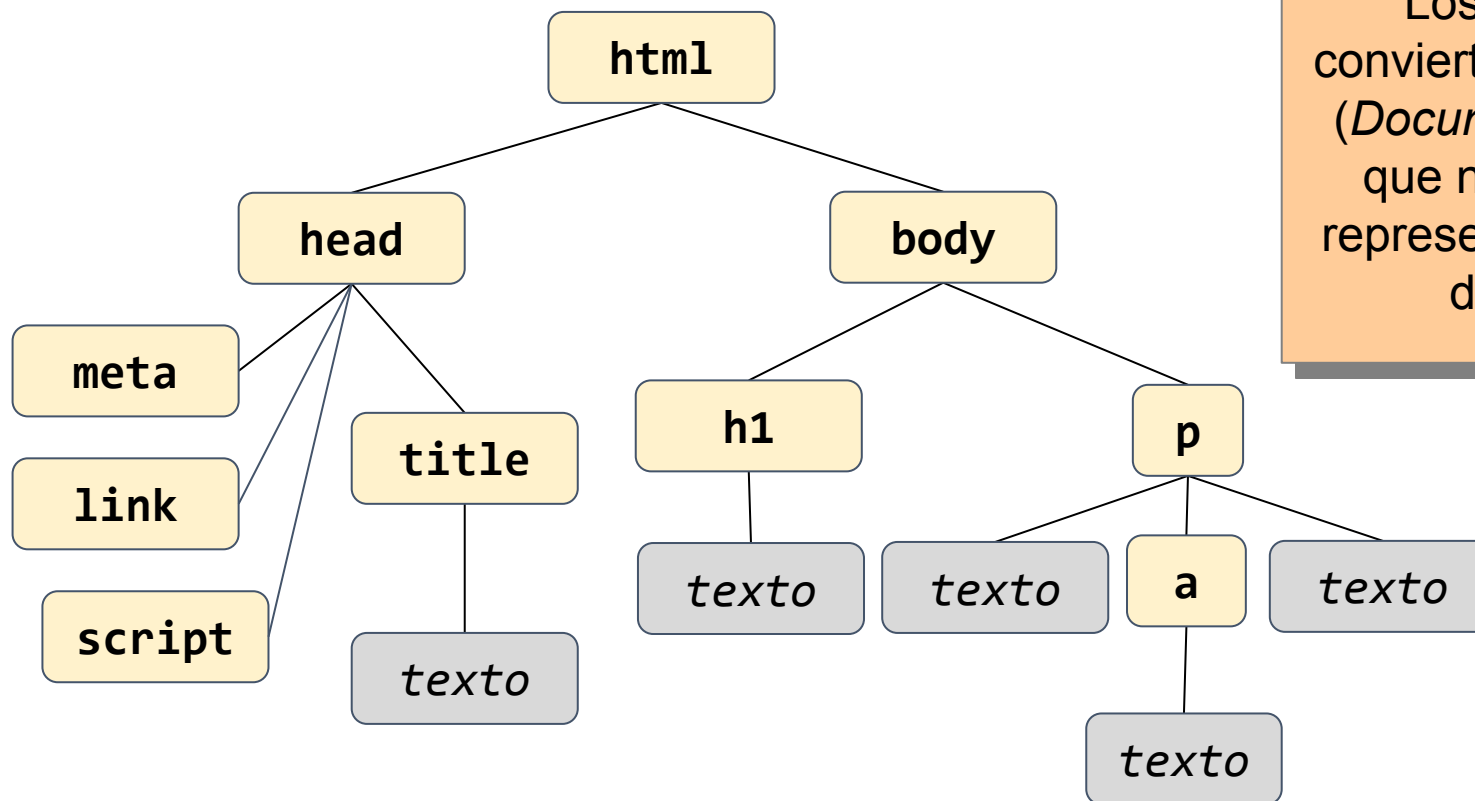
# Estructura de un documento HTML5

## - Estructura básica:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="estilo.css">
    <script src="archivo.js"></script>
    <title>P&acute;gina de ejemplo</title>
  </head>
  <body>
    <h1>P&acute;gina de ejemplo</h1>
    <p>Esto es una <a href="demo.html">simple</a>
      muestra.</p>
    <!-- esto es un comentario -->
  </body>
</html>
```

# Estructura de un documento HTML5

Un documento HTML es un árbol de elementos y texto. Cada elemento se corresponde con una etiqueta HTML de apertura y una de cierre, aunque en algunos casos esto puede no cumplirse.



Los navegadores lo convierten en un árbol **DOM** (*Document Object Model*), que no es más que una representación en memoria del documento.

## Estructura de un documento HTML5

# Etiquetas estructurales

# Estructura de un documento HTML5

## Etiquetas estructurales

<b>&lt;header&gt;</b>	Permite añadir contenido de encabezamiento a un documento, artículo, sección, ...
<b>&lt;footer&gt;</b>	Representa el pie de un documento, artículo, sección, ...
<b>&lt;nav&gt;</b>	Encierra enlaces de navegación o cualquier otro contenido que permita desplazarse, únicamente, dentro del mismo sitio web. Se trata de una sección con enlaces de navegación.
<b>&lt;section&gt;</b>	Se utiliza para separar distintas áreas en base a funcionalidad o contenido. Es algo más genérica que el resto de etiquetas.
<b>&lt;article&gt;</b>	Sirve para encerrar fragmentos individuales de contenido relacionado, tales como entradas individuales de un blog.
<b>&lt;aside&gt;</b>	Alberga información relacionada con el flujo principal del documento, pero no tiene cabida en él.



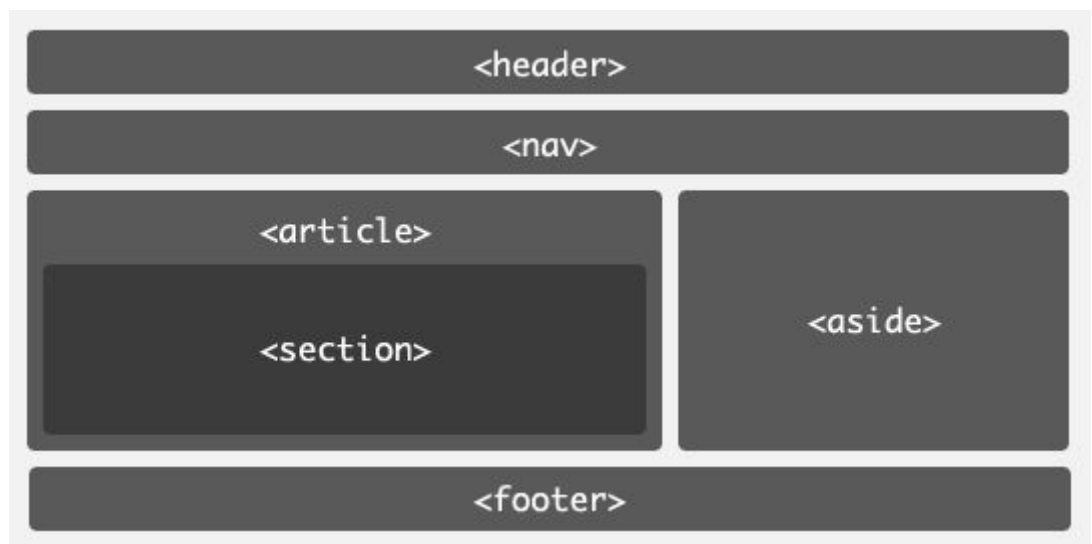
# Estructura de un documento HTML5

## Etiquetas estructurales

- **Antes de HTML5:** Se utilizaba el elemento `<div>`, con distintos valores de atributo *id* y/o *class*, para estructurar la página.



- **Con HTML5:** Se utilizan los nuevos *elementos estructurales*, dejando el elemento `<div>` para aplicar estilo.



# Estructura de un documento HTML5

## Etiquetas estructurales

### <header>

Permite añadir contenido de encabezamiento, no sólo a una página web, sino también a cualquier otro elemento estructural de html5.

### Ejemplo:

```
<header>
  <h1>Título principal</h1>
  <h2>Descripción</h2>
  <p>Última modificación: <time datetime="2014-02-11">el
    martes</time></p>
</header>
```

# Estructura de un documento HTML5

## Etiquetas estructurales

### **<nav>**

Encierra enlaces de navegación o cualquier otro contenido (por ejemplo, un formulario de búsqueda) que permita desplazarse, únicamente, dentro del mismo sitio web, ya sea a otras páginas, o a otras zonas dentro de la misma página.

### **Ejemplo:**

```
<header>
  <h1>Título principal</h1>
  <h2>Descripción</h2>
  <p>Última modificación: <time datetime="2014-02-11">el martes</time></p>
  <nav>
    <h2>Navegación</h2>
    <ul>
      <li><a href="home.php">Home</a></li>
      <li><a href="about.php">About</a></li>
      <li><a href="contact.php">Contact</a></li>
    </ul>
  </nav>
</header>
```

# Estructura de un documento HTML5

## Etiquetas estructurales

### **<section>**

Se utiliza para separar distintas áreas en base a funcionalidad o contenido; o dividir un artículo o una historia en diferentes secciones. Es algo más genérica que el resto de etiquetas pero tiene más significado semántico que `<div>`. Si lo que se desea es simplemente aplicar estilo, se recomienda utilizar `<div>`. Normalmente suele llevar un título o un encabezado (h1-h6) que identifica la temática del contenido.

### **Ejemplo:**

```
<section>
  <!-- Aquí va el contenido -->
</section>
```

# Estructura de un documento HTML5

## Etiquetas estructurales

### <article>

Está relacionado con <section>, pero tiene mayor significado semántico. El elemento <article> sirve para encerrar fragmentos individuales de contenido relacionado, tales como entradas individuales de un blog, vídeos, imágenes o noticias.

### Ejemplo:

```
<section>
  <!-- Sección de noticias de deportes -->
  ...
  <article>
    <!-- Primera noticia -->
  </article>
  <article>
    <!-- Segunda noticia -->
  </article>
  ...
</section>
```

# Estructura de un documento HTML5

## Etiquetas estructurales

### `<article>` (y `II`)

La versatilidad de las etiquetas `<section>` y `<article>` es todavía mayor, ya que se pueden anidar secciones dentro de artículos y viceversa.

### Ejemplo:

```
<article>
  <!-- Noticia de deportes con una estructura determinada -->
  <section class="introduccion">
    ...
  </section>
  <section class="contenido">
    ...
  </section>
  <section class="resumen">
    ...
  </section>
</article>
```

# Estructura de un documento HTML5

## Etiquetas estructurales

### <aside>

Alberga información relacionada con el contenido principal del documento, pero que no es necesaria para que éste tenga sentido por sí mismo. Suele tratarse de información complementaria.

### Ejemplo:

```
<section>
  <h2>Comentarios</h2>
  <article>
    <!-- Primer comentario sobre la noticia -->
  </article>
  <article>
    <!-- Segundo comentario sobre la noticia -->
  </article>
</section>
<aside>
  <h2>Categorías</h2>
  <span><a href="/futbol">Fútbol</a></span>
</aside>
```

# Estructura de un documento HTML5

## Etiquetas estructurales

### **<footer>**

Al igual que ocurría en el caso de <header>, el propósito del elemento <footer> está bastante claro y no es otro que encerrar el contenido del pie del documento, o bien del elemento estructural que lo contiene.

### **Ejemplo:**

```
<article>
  <header>
  </header>
  ...
  <footer>
    <a href=".">Volver a index</a>
  </footer>
</article>
```



# Otros elementos importantes en HTML5

## Otros elementos importantes en HTML 5

### <main>

Representa el contenido principal del <body> del documento o de la aplicación web. Incluye contenido que es único para el documento y excluye cualquier otro que se repita en otras páginas del sitio, tal como enlaces de navegación (salvo que sean al contenido del propio <main>), copyright, logos, formularios de búsqueda, etc. Sólo puede haber un elemento <main> por documento.

### Ejemplo:

```
<main>
  <!-- Contenido principal del documento -->
  <h2>Aquí va el título</h2>
  <p>Lorem Ipsum dolor set amet</p>
  ...
</main>
```

## Otros elementos importantes en HTML5

### **<figure> y <figcaption>**

El elemento **<figure>** se utiliza para destacar diagramas, imágenes, códigos de ejemplo, etc, susceptible de ir en un anexo. En el flujo principal del documento se trata como una única unidad de contenido y autosuficiente, pudiéndose eliminar del documento sin afectar al significado de éste. En este caso debería llevar un título o una leyenda que se indicará mediante el elemento **<figcaption>**, aunque el uso de **<figcaption>** es optativo.

### **Ejemplo:**

```
<figure>
  
  <figcaption>Foto 12: Llegada al punto de destino.</figcaption>
</figure>
```

## Otros elementos importantes en HTML5

### <time>

Permite codificar una fecha y/o una hora en un formato que puede ser leído por navegadores, robots web, etc. Para ello se utiliza su atributo ***datetime*** cuyo formato es el siguiente:

AAAA-MM-DDThh:mm:ssDZH

AAAA: Año, con cuatro cifras

MM: Mes, con dos cifras

DD: Día del mes, con dos cifras

hh: Hora, de 00 a 23

mm: Minutos, de 00 a 59

ss: Segundos, de 00.0 a 59.99999

DHZ: Designador de Zona Horaria. Posibles valores: Z (hora Zulu ó GMT); +02:00 (GMT+02:00); -05:00 (GMT-05:00)

T: Carácter separador. Puede ser sustituido por un (y sólo uno) espacio.

### Ejemplo:

```
<p>Inicio previsto: <time datetime="2012-09-10T09:32:27.5403+02:00">  
10 de septiembre a las 9:32, hora de España</time></p>
```

## Otros elementos importantes en HTML5

### <progress>

Este elemento se utiliza para representar el progreso de una tarea. Utiliza dos atributos: **value** y **max**. El atributo *value* indica cuánta tarea se lleva realizada y *max* indica el total de trabajo a realizar para completar la tarea (por defecto 1). Para hacer indeterminado el progreso de una tarea, basta con no utilizar el atributo *value*.

### Ejemplo:

```
<ul>
  <li><label>Indeterminado:</label>
    <progress max="100"></progress></li>
  <li><label>Progreso (determinado): 0%</label>
    <progress value="0" max="10"></progress></li>
  <li><label>Progreso (determinado): 100%</label>
    <progress value="3254" max="3254"></progress></li>
  <li><label>Progreso (determinado): 57%</label>
    <progress value="0.4" max="0.7"></progress></li>
</ul>
```

## Otros elementos importantes en HTML5

### <meter>

Representa un valor escalar dentro de un rango conocido, o un valor fraccionario. Ejemplos de uso serían: mostrar el uso de disco; el resultado de una votación, etc. Admite los atributos ***value***, ***min***, ***max***, ***low***, ***high***, ***optimum***. Los atributos *min* y *max* indican los valores mínimo y máximo, respectivamente, que puede mostrar el elemento. *Value* es el valor “medido” que se representa. Los atributos *low* y *high* permiten dividir el rango en tres zonas: baja, media y alta. Si *optimum* es un valor mayor que *high*, indica que cuanto más alto sea el valor, mejor; pero si es menor que *low*, lo que indica es lo contrario.

### Ejemplo:

```
<label>Nota media:</label><meter min="0" max="10" low="3"
high="6" optimum="8" value="5.6" title="Nota media">5.6</meter>
```

```
<label>Num. medio fallos:</label><meter min="0" max="100"
value="32" title="Num. medio fallos">32</meter>
```

## Otros elementos importantes en HTML5

### <wbr>

Representa una posibilidad de ruptura en una línea de texto. Evita rupturas no deseadas y aleatorias hechas por el navegador.

### Ejemplo:

```
<p>El músculo externo<wbr>cleido<wbr>mastoi<wbr>deo  
(Sterno<wbr>cleido<wbr>mastoi<wbr>deus), común<wbr>mente  
abreviado ECOM, es un músculo robusto situado a los lados del  
cuello.</p>
```

El músculo externo  
cleidomastoideo  
(Sternocleidomastoi  
deus), comúnmente  
abreviado ECOM, es  
un músculo robusto  
situado a los lados del  
cuello.

Una **solución alternativa** es utilizar el guión llamado ***soft hyphen***. Para ello se utiliza la referencia numérica del carácter **&shy;** en lugar de <wbr>. El resultado final es el mismo pero con la ventaja de mostrar un guión en la separación de palabras.

```
<p>El músculo externo&shy;cleido&shy;mastoi&shy;deo  
(Sterno&shy;cleido&shy;mastoi&shy;deus), común&shy;mente  
abreviado ECOM, es un músculo robusto situado a los lados del  
cuello.</p>
```

El músculo esterno-  
cleidomastoideo  
(Sternocleidomastoi-  
deus), comúnmente  
abreviado ECOM, es  
un músculo robusto  
situado a los lados del  
cuello.

## Otros elementos importantes en HTML5

### <mark>

Representa un texto resaltado en un documento por ser relevante y hacia el que se quiere atraer la atención del usuario. Por ejemplo, en un texto puede resaltar cada ocurrencia de una palabra buscada.

### Ejemplo:

```
<p>Resultados de la búsqueda del término <mark>HTML 5</mark>:</p>
<section id="resultados">
  <article>
    <h2><a href="http://en.wikipedia.org/wiki/HTML_5">
      <mark>HTML 5</mark> - Wikipedia, the free encyclopedia
    </a>
  </h2>
  <p><mark>HTML 5</mark> is the next major revision of
    <mark>HTML</mark> ("hypertext markup language"), the core markup
    language of the World Wide Web. The WHATWG started work on
    the ...
```



## Otros elementos importantes en HTML5

### <datalist>

Engloba un conjunto de elementos **<option>** que corresponden a posibles valores predefinidos para un control **<input>**. Está ligado al control **<input>** usando el atributo **list** de éste, cuyo valor debe ser igual al atributo **id** del elemento **<datalist>**.

### Ejemplo:

```
...
Vehículo: <input type="text" name="vehiculo" list="vehiculos">
...
<datalist id="vehiculos">
  <option value="Coche">
  <option value="Moto">
  <option value="Camión">
  <option value="Furgoneta">
</datalist>
...
```

**Nota:** Los elementos **<option>** no llevan **</option>**, salvo que tengan contenido.

# Otros elementos importantes en HTML 5

## Atributos HTML5

# Otros elementos importantes en HTML 5

## Atributos HTML5

### **contenteditable**

Este atributo es de tipo enumerado y permite hacer que un elemento sea un área editable. El usuario puede cambiar el contenido y manipular el marcado. Al ser un atributo de tipo enumerado, sus posibles valores son “true” o “false”.

### Ejemplo:

```
<h1>Lista de tareas:</h1>

<ul contenteditable="true">
  <li>Sacar a pasear al perro</li>
  <li>Comprar el pan</li>
  <li>Ir al banco</li>
</ul>
```

# Otros elementos importantes en HTML 5

## Atributos HTML5

### **data-\*** : Atributos personalizados

En html5 se pueden definir atributos personalizados para todos los elementos html, simplemente utilizando el prefijo “**data-**” seguido de una cadena de al menos un carácter y que no debería contener letras mayúsculas.

### Ejemplo:

```
<h3>Lista de vehículos:</h3>

<ul>
  <li data-clase-vehiculo="coche">Renault Megane</li>
  <li data-clase-vehiculo="moto">Yamaha YZF-R1</li>
  <li data-clase-vehiculo="coche">Ford Focus</li>
  <li data-clase-vehiculo="camion">Mercedes Benz</li>
</ul>
```

# Otros elementos importantes en HTML 5

## Atributos HTML5

### **async** : Permite ejecutar scripts asíncronamente

Permite que el navegador siga parseando el html mientras se carga un código externo de javascript. Se ejecutará tan pronto esté cargado.

#### Ejemplo:

```
<html>
  <head>
    ...
    <script src="..." async></script>
  </head>
  <body>
    ...
  </body>
</html>
```

Se puede utilizar en combinación con el atributo **defer** para que el código se ejecute tras cargar el html

# Formularios HTML 5

## Formularios HTML 5

- ✓ **NO son XForms**
- ✓ **Tratan con la semántica y la funcionalidad, NO con la apariencia.**
- ✓ **Nuevas funcionalidades**
  - ☐ Nuevos tipos de entrada de datos.
  - ☐ Nuevas funciones y atributos.

## Formularios HTML 5

### `<input type="...">`

En html5 se definen nuevos valores para el atributo ***type*** del elemento `<input>`. Utilizando estos nuevos tipos se consigue delegar la validación de algunos datos de entrada en el navegador. Los valores para el atributo ***type*** son los siguientes:

Valor de <i>type</i>	Propósito
<code>button</code>	Representa un botón
<code>checkbox</code>	Representa una casilla de verificación
<code>file</code>	Permite elegir uno o más ficheros al usuario
<code>hidden</code>	Permite incluir valores ocultos en un formulario
<code>image</code>	Permite utilizar imágenes como botones en formularios
<code>radio</code>	Permite crear grupos de botones de opciones en los que sólo se puede seleccionar una de ellas



# Formularios HTML 5

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

Valor de <i>type</i>	Propósito
<b>reset</b>	Representa un botón que permite limpiar los campos del formulario
<b>submit</b>	Representa un botón que envía el formulario al servidor
<b>text</b>	Representa un campo de texto de una única línea
<b>password</b>	Representa un campo de texto de una única línea que permite la introducción de una contraseña de forma segura
<b>tel</b>	Para números de teléfono
<b>email</b>	Direcciones de correo electrónico
<b>url</b>	Direcciones de páginas web
<b>search</b>	Proporciona un control donde introducir un texto a buscar
<b>range</b>	Selector numérico dentro de un rango de valores, normalmente se visualiza como un <i>slider</i> .

# Formularios HTML 5

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

Valor de type	Propósito
<b>number</b>	Permite la introducción de un número entero o decimal.
<b>date</b>	Para introducir una fecha
<b>datetime</b>	Para introducir una fecha y hora
<b>datetime-local</b>	Para introducir una fecha y hora local
<b>month</b>	Para introducir un mes y año
<b>time</b>	Para introducir una hora
<b>week</b>	Para introducir una semana de un año
<b>color</b>	Para introducir un color

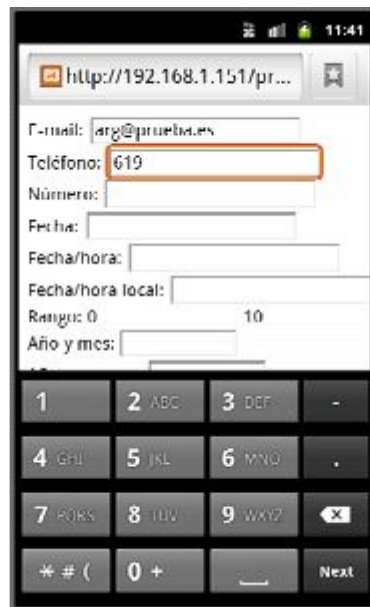
## Formularios HTML 5

`<input type="...">` (y ///)

E-mail: `<input type="email" name="email">`  
Teléfono: `<input type="tel" name="tlf">`  
Número: `<input type="number" name="numero">`



Mobile browser screenshot showing an HTML5 form. The email input field is highlighted with an orange border. The form includes fields for E-mail, Teléfono, Número, Fecha, Fecha/hora, Fecha/hora local, Rango (0 to 10), and Año y mes.



Mobile browser screenshot showing an HTML5 form. The telephone input field is highlighted with an orange border. The form includes fields for E-mail, Teléfono, Número, Fecha, Fecha/hora, Fecha/hora local, Rango (0 to 10), and Año y mes.



Mobile browser screenshot showing an HTML5 form. The number input field is highlighted with an orange border. The form includes fields for E-mail, Teléfono, Número, Fecha, Fecha/hora, Fecha/hora local, Rango (0 to 10), and Año y mes.

## Formularios HTML 5

`<input type="...">` (y /V)

### Ejemplo:

Fecha: `<input type="date" name="fecha">`

Fecha/hora: `<input type="datetime" name="fhora">`

Año y mes: `<input type="month" name="ames"><br>`

Año y semana: `<input type="week" name="asemana"><br>`

Fecha:

Fecha/hora:  UTC

Año y semana:

	Lun	Mar	Mie	Jue	Vie	Sab	Dom
30	31	1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	

Hoy

	Lun	Mar	Mie	Jue	Vie	Sab	Dom
30	31	1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	

Hoy

Seman:	Lun	Mar	Mie	Jue	Vie	Sab	Dom
31	30	31	1	2	3	4	5
32	6	7	8	9	10	11	12
33	13	14	15	16	17	18	19
34	20	21	22	23	24	25	26
35	27	28	29	30	31	1	2
36	3	4	5	6	7	8	9

Hoy

## Formularios HTML 5

`<input type="...">` (y V)

### Ejemplo:

```
Rango: -5<input type="range" name="rango" min="-5" max="25" value="16">16
```

Rango: -5  25

Si no se indican los atributos *min* y *max*, sus valores por defecto son 0 y 10, respectivamente.

# Formularios HTML 5

## Nuevos atributos

Aparecen nuevos atributos tanto para **<form>** y para **<input>**:

- Para **<form>**: **autocomplete**, **novalidate**
- Para **<input>**: **autocomplete**, **autofocus**, **form**, **formaction**, **formenctype**, **formmethod**, **formnovalidate**, **formtarget**, **height** y **width** (sólo para type=image), **list**, **min**, **max**, **multiple**, **pattern** (regexp), **placeholder**, **required**, **step**

# Formularios HTML 5

## Nuevos atributos

- ✓ **autocomplete**. Valores admitidos: on/off. Se puede aplicar a `<input>` y a `<form>`. Cuando este atributo se encuentra activado en un formulario, el navegador tratará de autocompletar los valores de los campos basándose en valores que el usuario hubiera introducido con anterioridad.

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  ...  
</form>
```

Se puede utilizar para cancelar el autocompletado en algún campo:

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  Nombre:<input type="text" name="fname" ><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit" value="Enviar">  
</form>
```

## Formularios HTML 5

### Nuevos atributos

- ✓ **autofocus**. Se trata de un atributo booleano que permite indicar qué control tendrá el foco cuando se cargue la página, permitiendo al usuario empezar a escribir en él sin tener que seleccionarlo manualmente.

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  
Nombre:<input type="text" name="fname" autofocus><br>  
E-mail: <input type="email" name="email" autocomplete="off"><br>  
  
<input type="submit" value="Enviar">  
  
</form>
```



# Formularios HTML 5

## Nuevos atributos

- ✓ **required**. Este atributo booleano permite especificar que el usuario debe rellenar el campo antes de enviar el formulario.

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  
  Nombre:<input type="text" name="fname" autofocus><br>  
  E-mail: <input type="email" name="email" required><br>  
  
  <input type="submit" value="Enviar">  
  
</form>
```

# Formularios HTML 5

## Nuevos atributos

- ✓ **placeholder**. Permite mostrar dentro del control `<input>` un pequeño texto indicando al usuario qué debe introducir en él, o en qué formato.

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  
Nombre:<input type="text" name="fname" placeholder="Nombre  
completo"><br>  
E-mail: <input type="email" name="email"><br>  
  
<input type="submit" value="Enviar">  
  
</form>
```



Nombre:

E-mail:

# Formularios HTML 5

## Nuevos atributos

- ✓ **multiple**. Este atributo booleano sirve para indicar si se permite al usuario introducir más de un valor (separados por espacios) en un mismo control.

```
<form action="send.php" method="get" autocomplete="on">
```

```
  Para: <input type="email" name="email" multiple required  
        autofocus><br>
```

```
  Asunto:<input type="text" name="fname"><br>
```

```
  Texto:<input type="text" name="texto"><br>
```

```
  Ficheros adjuntos: <input type="file" name="fichero" multiple>  
  <input type="submit" value="Enviar">
```

```
</form>
```

## Formularios HTML 5

### Nuevos atributos

- ✓ **pattern**. Especifica una expresión regular que debe evaluar el valor introducido en el control. La expresión regular debe seguir el patrón de las expresiones regulares de JavaScript. En la url <http://html5pattern.com/> se pueden consultar las expresiones regulares más comunes.

```
<form action="demo_form.asp" method="get" autocomplete="on">  
  Nombre:<input type="text" name="fname"><br>  
  E-mail: <input type="email" name="email"  
    pattern="^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$"><br>  
  <input type="submit" value="Enviar">  
</form>
```

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp)

# Elementos <audio> y <video>

## Elementos <audio> y <video>

Un fichero de audio o vídeo no es más que un fichero contenedor (del estilo de los fichero *zip*) que contiene diferentes archivos: pistas de audio, pistas de vídeo y metadatos adicionales como pueden ser los subtítulos.

### Formatos de vídeo más conocidos:

- Audio Video Interleave (\*.avi)
- Flash Video (\*.flv)
- MPEG4 (\*.mp4)
- Matroska (\*.mkv)
- Ogg (\*.ogg)
- WebM (\*.webm)

### Codecs de audio:

- AAC
- MPEG-3
- Ogg Vorbis

### Codecs de vídeo:

- H.264
- VP8
- Ogg Theora

## Elementos <audio> y <video>

### <audio>

Representa un sonido o un flujo de audio. Admite los siguientes atributos:

- **src**: dirección del recurso de audio.
- **preload**: atributo enumerado (none|metadata|auto) que indica si debería, y cómo, cargarse el fichero de audio.
- **autoplay**: atributo booleano. Permite iniciar el audio nada más cargar.
- **loop**: atributo booleano. Permite repetir infinitamente el audio.
- **muted**: atributo booleano para silenciar o no el audio por defecto.
- **controls**: atributo booleano para mostrar o no los controles de audio.

### Ejemplo:

```
<audio src="audio.wav" controls autoplay preload="metadata">  
  Tu navegador no soporta el elemento audio.  
</audio>
```

## Elementos <audio> y <video>

### <video>

Se utiliza para vídeo o películas, y ficheros de audio con títulos. Admite los mismos atributos que <audio> y, además, los siguientes:

- **poster**: dirección de un fichero imagen a mostrar mientras no esté disponible el recurso de vídeo.
- **width** y **height**: permiten indicar al navegador el ancho y alto que tendrá el vídeo cuando se cargue. Es recomendable utilizar estos atributos para evitar el efecto de redimensionado del elemento <video>.

### Ejemplo:

```
<video src="movie.mp4" width="320" height="240" controls  
preload="auto">
```

Tu navegador no soporta el elemento video.

```
</video>
```



## Elementos `<audio>` y `<video>`

### `<source>`

El elemento `<source>` permite especificar múltiples alternativas para un recurso multimedia, ya sea audio o vídeo. Sólo se puede utilizar como hijo de un elemento multimedia, delante de cualquier elemento de flujo o elemento `<track>`. Admite los atributos:

- ✓ **src**: dirección del recurso multimedia.
- ✓ **type**: tipo del recurso multimedia.

### Ejemplos:

```
<audio controls>
  <source src="audio.wav" type="audio/wav">
  <source src="audio.mp3" type="audio/mpeg">
  Tu navegador no soporta el elemento audio.
</audio>

<video width="320" height="240" controls preload="auto">
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Tu navegador no soporta el elemento video.
</video>
```

## Elementos <audio> y <video>

### <track>

El elemento <track> permite especificar pistas de texto externas tales como subtítulos, textos, descripciones, títulos de capítulos, etc, a recursos multimedia. Admite los atributos:

- **kind**: tipo de la pista: *subtitles*, *captions*, *descriptions*, *chapters*, *metadata*.
- **src**: dirección del fichero de texto asociado a la pista.
- **srclang**: idioma de texto de la pista.
- **label**: título de la pista.
- **default**: permite indicar cuál es la pista por defecto.

### Ejemplos:

```
<video width="320" height="240" controls preload="auto">
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  <track src="sub_esp.vtt" kind="subtitles" srclang="es" label="Español">
  <track src="sub_en.vtt" kind="subtitles" srclang="en" label="English">
</video>
```

# Características interesantes de HTML

# Características interesantes de HTML

## Imágenes adaptativas

La versión 5.1 de HTML introduce el elemento `<picture>` y los atributos `srcset` y `sizes` para el elemento `<img>` que permiten la carga condicional de imágenes.

Esta carga adaptativa de imágenes se puede hacer en función de cuatro aspectos:

- **device-pixel-ratio**. En función del tamaño de píxel del dispositivo.
- **art direction**. En función del tamaño de la ventana del navegador.
- **viewport**. En función del tamaño físico de la pantalla del dispositivo.
- **image format**. En función del formato de la imagen.

# Características interesantes de HTML

## Imágenes adaptativas

### <picture>

Permite especificar diferentes fuentes, mediante el elemento **<source>**, para el elemento **<img>** que contiene. Si se especifican elementos **<source>**, éstos deben ir antes del elemento **<img>**.

Cuando el elemento **<source>** se utiliza junto al elemento **<picture>** admite los siguientes atributos:

- **srcset**: permite indicar una o más imágenes candidatas, en función de la densidad de píxel del dispositivo, separadas por comas.
- **media**: permite especificar reglas media query para utilizar o no el elemento **<source>**.
- **type**: permite especificar el formato de la imagen para que el navegador elija el formato más adecuado.

# Características interesantes de HTML

## Imágenes adaptativas

### <picture>

Ejemplo: utilizando *art direction*

```
<picture>
  <source media="(min-width: 45em)" srcset="img_grande.jpg">
  <source media="(min-width: 32em)" srcset="img_mediana.jpg">
  
</picture>
```

El código anterior:

- para anchos de ventana mayores de 45em, se mostrará `img_grande.jpg`;
- para anchos de ventana mayores de 32em y menores de 45em, se mostrará `img_mediana.jpg`;
- para el resto de anchos, se mostrará `img_pequenya.jpg`.

# Características interesantes de HTML

## Imágenes adaptativas

### <img>

Este elemento admite dos nuevos atributos en la revisión 5.1 de la especificación HTML para la inclusión de imágenes adaptativas en un documento html. Estos dos atributos son:

- **srcset**: permite indicar una o más imágenes candidatas, en función de la densidad de píxel del dispositivo o del tamaño de la imagen, separadas por comas.
- **sizes**: permite especificar tamaño con el que se mostrará la imagen en función de las características del dispositivo.

Lo que se puede conseguir utilizando estos dos nuevos atributos de **<img>** se puede conseguir también usando el elemento **<picture>**.

# Características interesantes de HTML

## Imágenes adaptativas

Ejemplos: utilizando *art direction*

```

```

El código anterior:

- para anchos de ventana de hasta 320px, se mostrará `img_pequenya.jpg` con un ancho de 150px;
- para anchos de ventana de hasta 640px, se mostrará `img_mediana.jpg` con un ancho de 300px;
- para el resto de anchos, se mostrará `img_pequenya.jpg` con un ancho de 600px.



# Características interesantes de HTML

## Imágenes adaptativas

Ejemplos: Utilizando *device-pixel-ratio*

- Con elemento <picture>

```
<picture>
  <source srcset="imagen_1_5x.jpg 1.5x, imagen_2x.jpg 2x">
  
</picture>
```

- Con elemento <img>

```

```