HTML5

Programación Multimedia.

G.I.M.

Inmaculada Coma, Francisco Grimaldo

VNIVERSITAT DE VALÈNCIA OpenCourseWare

Resumen

- En este tema:
  - Veremos un resumen de los cambios introducidos por HTML5 respecto a versiones anteriores de HTML y XHTML.
  - Entraremos a ver con detalle algunas de las nuevas funcionalidades, como son la creación de formularios o la inserción de medios en al web, la programación de funciones de mover & arrastrar o de geolocalización.

Introducción.

- Actualmente el w3c está en proceso de desarrollo del estándar HTML 5.
- Última versión: HTML5.1. Candidate recomendation diciembre 2012:
  - http://www.w3.org/TR/html5/
- Último draft:
  - http://www.w3.org/html/wg/drafts/html/master/Overview.html
- Seguimiento de versiones:
  - http://www.w3.org/standards/history/html5

Introducción

- HTML5 introduce etiquetas nuevas que dan soporte a los requisitos de los sitios webs actuales (contenidos semánticos, multimedia..)
- Además, incluye un conjunto APIs de programación específicas para cosas como:
  - Gráficos 2D sobre un canvas.
  - Ejecución de medios.
  - Arrastrar y soltar elementos.
  - Descripción semántica de contenidos (microdata)

Introducción.

- Al igual que con CSS3 el soporte de los nuevos elementos, etiquetas y atributos es desigual en los diferentes navegadores.
  - http://www.findmebyip.com/litmus/
- Se pueden utilizar librerías de Javascript que detectan características de los navegadores (como Modernizr) o que simulan características cuando no están disponibles (polyfills).

HTML5

- Estructura básica de un documento. Sintaxis compatible con versiones anteriores.

```
<!DOCTYPE html>
<html lang="ES">

   <head>
      <title>Este es un ejemplo de HTML5</title>
      <meta charset="UTF-8">
      <link rel="stylesheet" href="estilo.css">
   </head>

   <body>
   Este es el cuerpo de la página, donde irá todo el
   contenido.
   </body>

</html>
```

Descripción en la cabecera

	XHTML	HTML5	
doctype	`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>`	`<!DOCTYPE html>`	HTML5 no está basado en SGML, no necesita dtd
html	`<html xmlns=http://www.w3.org/1999/xhtml lang =“es” xml:lang=“es”>`	`<html lang=“es”>`	El espacio de nombres xmln es requerido por XHTML pero no soportado por HTML
charset	`<meta http-equiv="content-type" content="text/html; charset=UTF-8">`	`<meta charset="UTF-8">`	
link	`<link rel="stylesheet" type="text/css" href="theme.css">`	`<link rel="stylesheet” href="theme.css">`	

Elementos estructurales nuevos

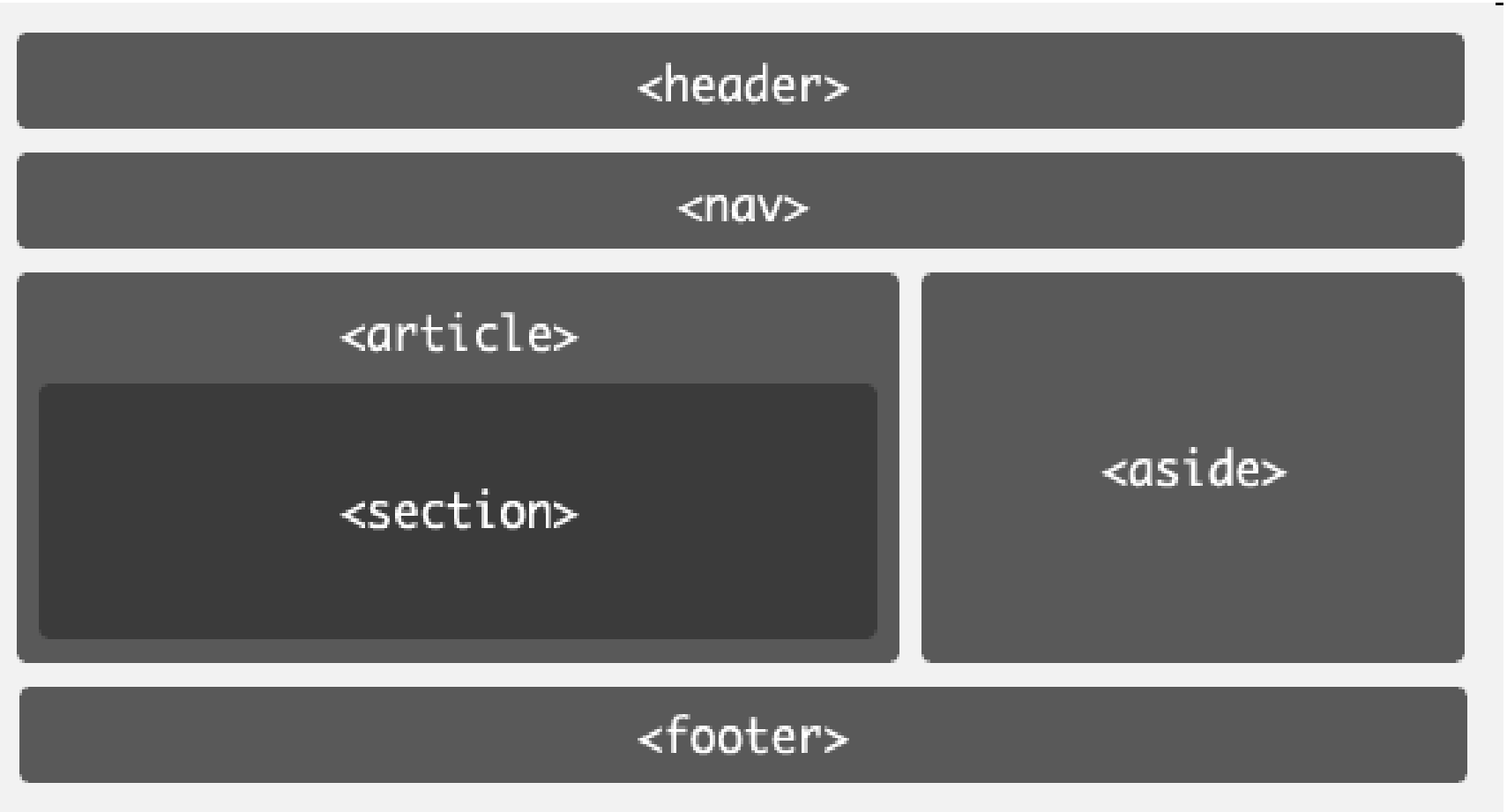
- Los nuevos **elementos estructurales,** facilitan la estructuración de las páginas según su contenido:
  - **section:** representa un documento genérico o sección de una aplicación (ej. Un capítulo, una sección de un capítulo o cualquier cosa con su propio encabezamiento).
  - **article**: entrada independiente de un documento, por ej. un blog o un artículo de un periódico
  - **aside**: pieza de contenido que está parcialmente relacionada con el resto de la página
  - **hgroup:** cabecera de una sección

Elementos estructurales nuevos

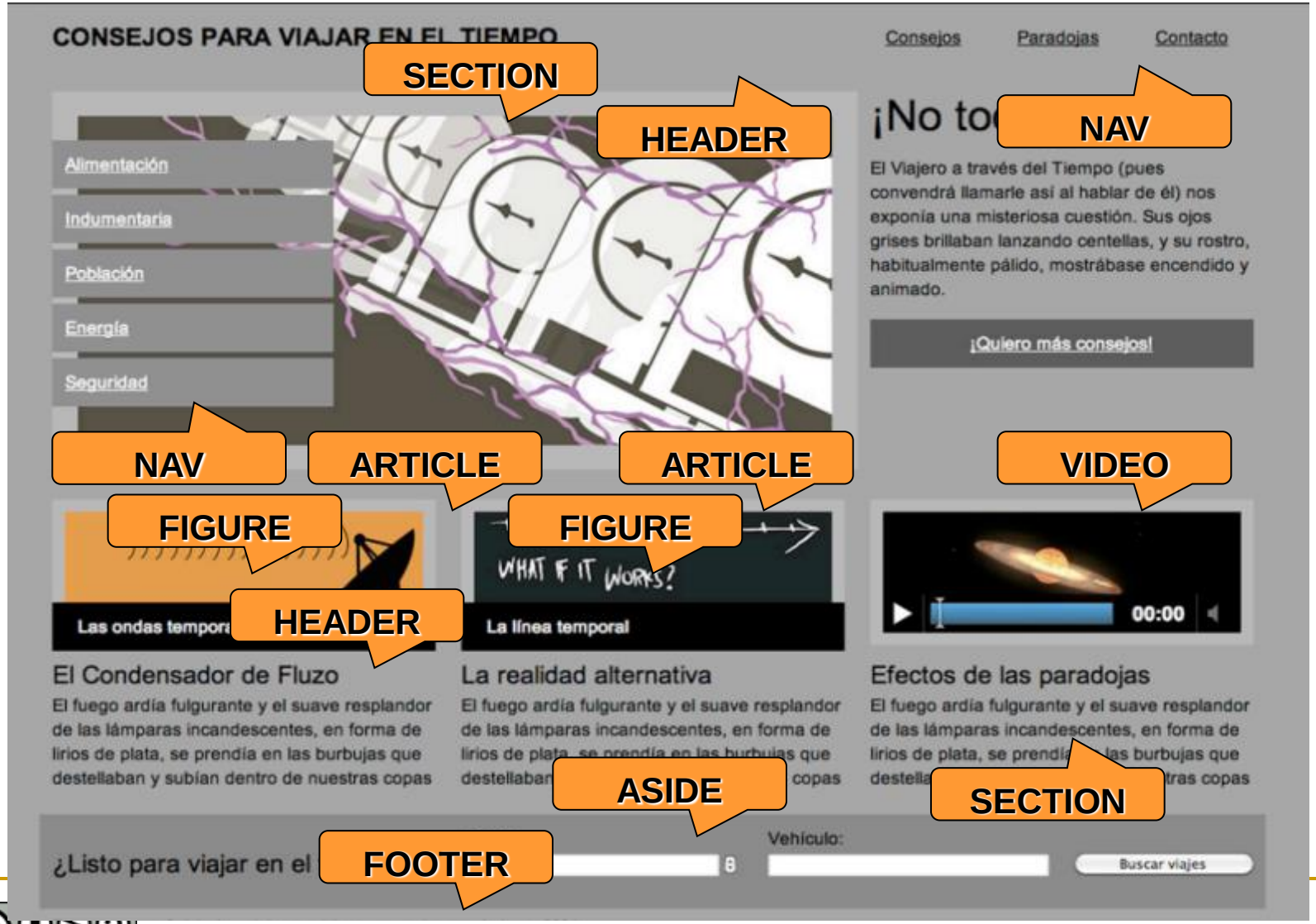
- **header**: cabecera de una sección o bloque.
- **footer**: final de sección que puede contener información del autor, copyright, etc.
- **nav**: parte de un documento destinada a la navegación
- **figure:** sirve para asociar texto a algún contenido embebido sea una imagen o un vídeo

```
<figure>
   <video src="ogg"></video>
   <figcaption> Example </figcaption>
</figure>
```

Elementos estructurales nuevos



Elementos estructurales nuevos



Otros elementos nuevos.

- Contenidos multimedia y externos:
  - **video** y **audio**
  - **embed**
  - **canvas:** para generar dinámicamente gráficos o juegos
- Otros:
  - **mark** : texto marcado; **progress**: progreso de una tarea; **meter**: indicar ciertas medidas (ej. Uso de disco); **time**: marcar un momento temporal o fecha; **command:**comando que puede ser invocado; **details**: información adicional; **datalist**: puede ser utilizado en comboxes; **keygen**; **output**

Cambios en elementos

- Desaparecen **elementos** respecto a HTML 4.0.
  - Elementos que desaparecen porque sólo hacen referencia al aspecto son manejados ahora mediante CSS: basefont; big; center; font; s; strike; tt; u
  - Elementos que desaparecen por razones de usabilidad y accesibilidad: frame; framset; noframes; (en su lugar usar iframe)
  - applet-> reemplazado por object

Cambios en elementos

- Desaparecen **atributos** de elementos, principalmente todos los que hacen referencia a apariencia y formato del documento que deben ser gestionados con CSS.
  - align, background, bgcolor, border, cellspacing, cellpaddding, height, marginheight, marginwidth, nowrap, scrolling, size, vspace, width…

HTML5 y Javascript

- Al igual que en las versiones anteriores tenemos un API de programación Javascript que nos permite:
  - Recoger eventos sobre los elementos de la página.
  - Modificar dinámicamente contenidos a través de la jerarquía de objetos del DOM.
- HTML5 incluye nuevos eventos sobre determinados objetos y explota las posibilidades de acceso al árbol de nodos para funcionalidades avanzadas como el drag&drop.

HTML5 y Javascript

- Podemos recoger eventos sobre los diferentes elementos de una página de varias formas:
- En el código HTML:

```
<img id="imag" src="file.jpg" onclick="ProcessClick(event)">
```

- Dentro de un script:

```
var im = getElementById("imag");
im.onclick = function ProcessClick(event) {.. }
```

- Con listeners (addEventListener, removeEventListener):

```
var im = getElementById("imag");
im.addEventListener('click', ProcessClick(event),true);
```

- El evento que se recibe contiene información útil (coordenadas, teclas pulsadas, etc):
  - http://www.w3.org/TR/DOM-Level-3-Events/

HTML5 y Javascript

- En HTML aparecen eventos nuevos.
- Eventos del objeto window:
  - onafterprint, onbeforeprint, onbeforeunload, onerror, onhaschange, onload, onmessage, onoffline, onpagehide, onpageshow, onpopstate, onredo, onresize, onstorage, onundo, onunload.
- Eventos de teclado (sin cambios)
  - onkeydown, onkeypress, onkeyup
- Eventos de ratón:
  - onclick, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondrop, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onmousewheel, onscroll.
- Eventos de formularios y medios (los veremos).

HTML5

- Veamos con más detalle algunos de estos elementos nuevos de HTML5 y las posibilidades de diseño que nos ofrecen.

Etiquetas semánticas

- **Microdata**: permite añadir descripciones semánticas a los contenidos de forma que los buscadores puedan extraer información estructurada. Se añaden etiquetas descriptivas dentro de los elementos HTML.
- **Microformats**: Existen vocabularios para describir contenidos
  - http://microformats.org/

```
<section itemscope itemtype="http://schema.org/Person">
        Hello, my name is
        <span itemprop="name">John Doe</span>,
        I am a
        <span itemprop="jobTitle">graduate research assistant</span>
        at the
        <span itemprop="affiliation">University of Dreams</span>.
        My friends call me
        <span itemprop="additionalName">Johnny</span>.
        You can visit my homepage at
        <a href="http://www.JohnnyD.com" itemprop="url">www.JohnnyD.com</a>.
        <section itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
                I live at
                <span itemprop="streetAddress">1234 Peach Drive</span>
                <span itemprop="addressLocality">Warner Robins</span>
                ,
                <span itemprop="addressRegion">Georgia</span>.
        </section>
</section>
```

http://en.wikipedia.org/wiki/Microdata_(HTML5)

FORMULARIOS

- Tipos de controles
- Tipos de controles input y atributos.
- Selectores CSS
- Eventos

Formularios

- Tipos de controles
  - input
  - label
  - button
  - select
  - optgroups
  - option
  - textarea
  - keygen
  - fieldset
  - legend
  - datalist
  - progress
  - meter
  - output

Input

```
<input name="text" type="text">
```

Button Botón

```
<button type="button"> Botón </button>
```

Textarea

```
<textarea name="area" rows="5" cols="40"readonly>  </textarea>
```

Select, option, optgroup

Lecture 01: Powers of Ten
I: Classical Mechanics
Lecture 01: Powers of Ten
Lecture 02: 1D Kinematics
2 Electricity and Magnestism
Lecture 01: What holds our world together?
Lecture 02: Electric Field

```
<select name="c">
  <optgroup label="I: Classical Mechanics">
  <option value="1.1">Lecture 01: Powers of Ten </option>
  <option value="1.2">Lecture 02: 1D Kinematics</option>
  </optgroup>
  <optgroup label="2 Electricity and Magnestism">
  <option value="2.1">Lecture 01: What holds our world together?</option>
  <option value="2.2">Lecture 02: Electric Field</option>
  </optgroup>
</select>
```

Fieldset y legend

Elegir opción:
Uno
Dos

```
<fieldset>
  <legend> Elegir opción: </legend>
  <p><label> <input type="radio" name="size"> Uno </label></p>
  <p><label> <input type="radio" name="size"> Dos </label></p>
</fieldset>
```

Progress

```
<progress id="p" max=100><span>0</span>%</progress>
```

Meter

```
<meter min="0" max="100" value="75"> Meter </meter>
```

Datalist

a
b
c

```
<input name="datalist" type="datalist" list="mydata"/>
<datalist id="mydata">
    <option label="a" value="a"/>
    <option label="b" value="b"/>
    <option label="c" value="c"/>
</datalist>
```

Formularios

- Tipos de elementos **input**
  - text
  - checkbox
  - radio
  - password
  - color
  - date
  - datetime
  - datetime-local
  - email
  - month
  - number
  - range
  - search
  - tel
  - time
  - url
  - week

Formularios

- Atributos del input:
  - accept align alt autocomplete autofocus checked disabled form height list max maxlenght min multiple name pattern placeholder readonly required size src step type value width
  - formaction formenctype formmethod formnovalidate formtarget
- No todos los atributos son válidos para todos los tipos de datos
  - http://www.w3.org/html/wg/drafts/html/master/forms.html#the-input-element
  - http://www.w3schools.com/tags/tag_input.asp

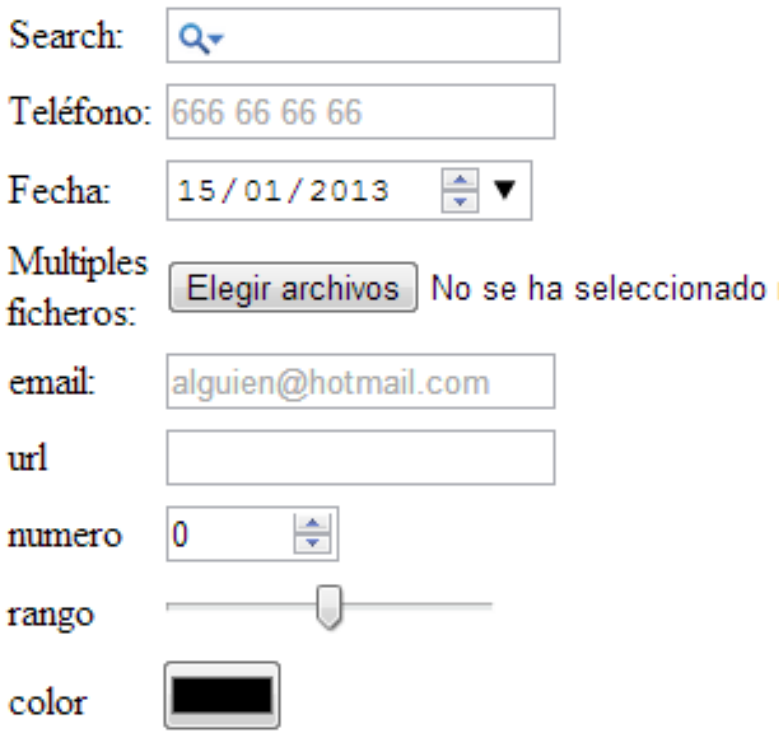
Atributos permitidos a cada control

	Hidden	Text, Search, URL, Telephone	E-mail	Password	Date and Time, Date, Month, Week, Time	Local Date and Time, Number	Range	Color	Checkbox, Radio Button	File Upload	Submit Button	Image Button	Reset Button, Button
accept	-	-	-	-	-	-	-	-	-	X	-	-	-
alt	-	-	-	-	-	-	-	-	-	-	-	X	-
autocomplete	-	X	X	X	X	X	X	X	-	-	-	-	-
checked	-	-	-	-	-	-	-	-	X	-	-	-	-
formaction	-	-	-	-	-	-	-	-	-	-	X	X	-
formenctype	-	-	-	-	-	-	-	-	-	-	X	X	-
formmethod	-	-	-	-	-	-	-	-	-	-	X	X	-
formnovalidate	-	-	-	-	-	-	-	-	-	-	X	X	-
formtarget	-	-	-	-	-	-	-	-	-	-	X	X	-
height	-	-	-	-	-	-	-	-	-	-	-	X	-
list	-	X	X	-	X	X	X	X	-	-	-	-	-
max	-	-	-	-	X	X	X	-	-	-	-	-	-
maxlength	-	X	X	X	-	-	-	-	-	-	-	-	-
min	-	-	-	-	X	X	X	-	-	-	-	-	-
multiple	-	-	X	-	-	-	-	-	-	X	-	-	-
pattern	-	X	X	X	-	-	-	-	-	-	-	-	-
placeholder	-	X	X	X	-	-	-	-	-	-	-	-	-
readonly	-	X	X	X	X	X	-	-	-	-	-	-	-
required	-	X	X	X	X	X	-	-	X	X	-	-	-
size	-	X	X	X	-	-	-	-	-	-	-	-	-
src	-	-	-	-	-	-	-	-	-	-	-	X	-
step	-	-	-	-	X	X	X	-	-	-	-	-	-
width	-	-	-	-	-	-	-	-	-	-	-	X	-

Atributos y funciones preconstruidos para el control de datos.

	Hidden	Text, Search, URL, Telephone	E-mail	Password	Date and Time, Date, Month, Week, Time	Local Date and Time, Number	Range	Color	Checkbox, Radio Button	File Upload	Submit Button	Image Button	Reset Button, Button
checked	·	·	·	·	·	·	·	·	X	·	·	·	·
files	·	·	·	·	·	·	·	·	·	X	·	·	·
value	default	value	value	value	value	value	value	value	default/on	filename	default	default	default
valueAsDate	·	·	·	·	X	·	·	·	·	·	·	·	·
valueAsNumber	·	·	·	·	X	X	X	·	·	·	·	·	·
list	·	X	X	·	X	X	X	X	·	·	·	·	·
selectedOption	·	X	X	·	X	X	X	X	·	·	·	·	·
select()	·	X	X	X	·	·	·	·	·	·	·	·	·
selectionStart	·	X	X	X	·	·	·	·	·	·	·	·	·
selectionEnd	·	X	X	X	·	·	·	·	·	·	·	·	·
setSelection Range()	·	X	X	X	·	·	·	·	·	·	·	·	·
stepDown()	·	·	·	·	X	X	X	·	·	·	·	·	·
stepUp()	·	·	·	·	X	X	X	·	·	·	·	·	·

Formularios




```
<input type="search" name="search" />

<input type="tel" name="tel"
       placeholder="666 66 66 66 "
       pattern="[0-9]{9}"
       title="El número de tel. debe tener 9 cifras"/>

<input type="date" name="date"
       value="2013-01-15" min="2013-01-01" max="2013-08-15/" />

<input type="file"  name="fichero"
       multiple="multiple" />

<input type="email" name="email"
       placeholder="alguien@hotmail.com"/>

<input type="url" name="homepage"/>

<input type="number" name="number"
       value="0" min="0" max ="100" step="1"/>

<input type="range" name="range"
       value="25" min="0" max="50"/>

<input type="color" name="color" />
```

Formularios. Selectores CSS

- :required :optional
- :valid :invalid
- :in-range :out-of-range

```
<input name="email" type="email" placeholder="alguien@hotmail.com"/>
```


```
<style>
:invalid
{
    border-color:#FF0000;
    box-shadow: 0 0 5px rgba(255,0,0,.5);
}
</style>
```

Formularios

- Eventos en formularios
  - onblur, onchange, onfocus, onselect, onsubmit, oncontextmenu, onformchange, onforminput, oninput, oninvalid, ~~onreset~~

MULTIMEDIA EN HTML5

- Audio y vídeo en HTML5.
  - Etiquetas de audio y vídeo, propiedades.
  - Múltiples sources
  - Múltiples tracks
- Control de medios con javascript:
  - HTMLMediaElement
  - Eventos

Elementos multimedia: audio y vídeo

- HTML5 permite incrustar vídeo y audio de forma sencilla y sin plug-in y controlar la reproducción de los dichos medios.

	Formato	Tipo MIME	Navegadores
VIDEO	MP4	video/mp4	Chrome6+,IE9+, Safari 5+
	WebM	video/webm	Chrome6+,Firefox 3.6+, Opera 10+
	Ogg	video/ogg	Chrome6+,Firefox 3.6+, Opera 10+
AUDIO	Mp3	audio/mpeg	Chrome6+,IE9+, Safari 5+
	Ogg	audio/ogg	Chrome6+,Firefox 3.6+, Opera 10+
	Wav	audio/wav	Chrome6+,Firefox 3.6+,Safari 5+ Opera 10+

http://drafts.htmlwg.org/html/master/embedded-content-0.html#the-video-element

Elementos multimedia: audio y vídeo.

- Aparte de los formatos de ficheros muchas de las etiquetas y funcionalidades que vamos a ver no son soportadas por diferentes navegadores:
- http://www.longtailvideo.com/html5/

Vídeo y audio.

- Video y audio. Atributos comunes:
  - **src**: indica la url del medio a reproducir
  - **crossorigin:** acceso a medios ubicados en otros sitios web
    - **anonymous** |use-credentials
  - **preload**: controla si el medio es precargado;
    - none | metadata | auto
  - **mediagroup:** para agrupar varios medios y controlarlos simultáneamente. Ej: dos vídeo, uno vídeo hablado y otro interpretado en lenguaje de signos.
  - **autoplay**: para que el medio se reproduzca automáticamente;
  - **loop**: el medio se reproduce de manera repetitiva
  - **muted**: el sonido aparecerá inicialmente desactivado
  - **controls**: para que se muestre barra de control debajo del medio
    - Atributos booleanos, aparecen o no

Vídeo y audio

- vídeo: atributos específicos
  - **height**: altura del contenedor del vídeo
  - **width**: ancho del contenedor vídeo
  - **poster**: url con la imagen que se muestra si el vídeo no está disponible; la que se ve antes de darle a “play”

```
<video src=“mivideo.mp4” controls autoplay loop muted>
</video>
```

- En lugar de src, puede contener varios elementos “**source**”, sirve para dar alternativas de un mismo fichero.
  - source. Atributos: src, type y media.

```
<audio controls autoplay>
  <source src=“x.ogg" type="audio/ogg" />
  <source src=“x.mp3" type="audio/mpeg" />
</audio>
```

Vídeo y audio. MediaController.

- Cuando tenemos varios medios agrupados con **mediagroup** se genera un **mediacontroller** que controla su ejecución simultánea**.**

```
<video poster="some.png" controls mediagroup="migrupo">
        <source src="videofile.mp4" type="video/mp4" />
        <source src="videofile.ogg" type="video/ogg" />
</video>
<video poster="sign.png" mediagroup="migrupo">
        <source src="videofilesign.mp4" type="video/mp4" />
        <source src="videofilesign.ogg" type="video/ogg" />
</video>
<audio mediagroup="migrupo">
        <source src="audiofile.mp3" type="audio/mp3" />
        <source src="audiofile.ogg" type="audio/ogg" />
</audio>
```

Vídeo y audio. Tracks.

- Otra opción que proporciona HTML5 es tener múltiples **tracks** que se añaden al elemento principal y que también serán controlados por un MediaController común.
- **track**: El medio puede contener dentro un conjunto de elementos *track* que permiten tener varias pista (ej. subtítulos).
  - Atributos: kind, src, srclang, label, default

```
<track kind="subtitle"s src="brave.en.vtt"
  srclang="en" label="English">
```

- Todavía escaso soporte en los navegadores.

Vídeo y audio. Tracks.

```
<video controls poster="miposter.jpg" width="4820"
height="200">

<source src="source.m4v" type="video/mp4" />
<source src="source.webm" type="video/webm" />

<track kind="subtitles" src="source.vtt" srclang="en"
label="English subtitles">

</video>
```

NOTA: para que funcione los archivos del track hay que subirlo a un servidor, ¡en local no va!

Vídeo y audio.

- Además de los objetos “video” y “audio” el DOM nos proporciona un interfaz de programación conjunto para los elementos “**media**” (vídeo y audio), que permite acceder a sus propiedades y modificar algunas de ellas.

- Veamos este interfaz

  interface **HTMLMediaElement** : HTMLElement

  http://www.w3.org/html/wg/drafts/html/master/embedded-content-0.html#media-elements

Media. Atributos y métodos del DOM.

- Estado de error
  - **error ***: devuelve un error con el estado del medio
- Estado de la red
  - **src, crossorigin, preload-**
  - **currentSrc:** devuelve la URL del medio actual
  - **buffered**: devuelve un objeto TimeRanges con las partes cargadas en el buffer
  - **networkState**: devuelve el estado de la red
  - **load**() : carga de nuevo el elemento
  - **canPlayType**(): comprueba si el vídeo puede ejecutar el medio

(*) Atributos todavía no soportados en la mayoría de los navegadores actuales

(*) Solo chrome y Safari

Media. Atributos y métodos del DOM

- Ready state
  - **readyState:** devuelve si el medio está preparado
  - **seeking:** devuelve si el usuario está actualmente buscando en el medio
- MediaController
  - **controller *:** devuelve el objeto MediaController controlador del medio

Media. Atributos y métodos del DOM

- Estado de reproducción
  - **play**() : inicia la reproducción
  - **pause**(): pausa la reproducción
  - **currentTime:** modifica o devuelve la posición actual de reproducción (en segundos)
  - **duration**: devuelve la duración del medio
  - **paused**: modifica o devuelve si el medio está pausado

```
var myVideo=document.getElementById("video1");
function playVid(){
  myVideo.play();
}
function pauseVid(){
  myVideo.pause();
}
```

Media. Atributos y métodos del DOM

- **startDate ***: devuelve un objeto Date que representa el tiempo transcurrido desde el inicio
- **defaultPlaybackRate *:** modifica o devuelve la velocidad de reproducción por defecto
- **playbackRate ****: modifica o devuelve la velocidad de reproducción
- **ended**: devuelve si la reproducción ha finalizado
- **played ****: devuelve un objeto TimeRanges con las partes reproducidas.
- **seekable:** devuelve un objeto TimeRanges con las partes que se pueden buscar
- **autoplay, loop**

Media. Atributos y métodos del DOM

- Controles
  - **volume:** modifica o devuelve el volumen
  - **defaultMuted *:** modifica o devuelve si el medio está silenciado
  - **controls, muted**
- Tracks
  - **audioTracks ***: devuelve un objeto AudioTrackList
  - **textTracks *** Devuelve un objeto TextTrackList con los tracks disponibles
  - **videoTracks *** Devuelve un objeto VideoTrackList object con los tracks de video disponibles
  - **addTrackText**(): añade un track de texto

Media. Atributos y métodos del DOM

```
interface TimeRanges
{
        readonly attribute unsigned long length;
        double start(unsigned long index);
        double end(unsigned long index);
};
```

Media. Atributos y métodos del DOM

- Ejemplo

```
<audio id="miAudio">
  <source src=Churchill.mp3 type=audio/mp3>
</audio>

<div>
  <button onclick="document.getElementById('miAudio').play()">Play the Audio</button>
  <button onclick="document.getElementById('miAudio').pause()">Pause the Audio</button>
  <button onclick="document.getElementById('miAudio').volume+=0.1">Increase Volume</button>
  <button onclick="document.getElementById('miAudio').volume-=0.1">Decrease Volume</button>
</div>
</body>
```

Media. Eventos

- Los eventos que podemos recoger sobre los objetos tipo media son:
- Estado de la red:
  - onloadstart; onprogress; onsuspend; onabort; onerror; onemptied; onstalled;
- Estado de reproducción:
  - onloadedmetadata; onloadeddata; oncanplay; oncanplaythrough; onplaying; onwaiting;
  - onseeking; onseeked; onended;
  - ondurationchange; ontimeupdate; onplay; onpause; onratechange; onvolumechange;
- http://www.w3.org/html/wg/drafts/html/master/embedded-content-0.html#event-definitions

Elementos multimedia

- Se pueden crear también dinámicamente los objetos de audio y vídeo.

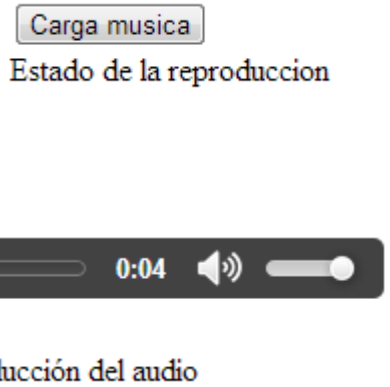
```
var miaudio = new Audio("file.mp3");
```

```
<script language="javascript" type="text/jscript">
function cargaAudio()
{
    var miaudio = new Audio("Churchill.mp3");
    miaudio.addEventListener('loadeddata',AudioCargado(event),false);
    miaudio.controls=true;

    var divAudio = document.getElementById('capaAudio');
    divAudio.appendChild(miaudio);

    miaudio.play();
}
function AudioCargado(event)
{
    var et = document.getElementById('etiqueta');
    et.innerHTML ="Se inicia la reproducción del audio";
}
</script>
</head>

<body>
<div id="capaAudio"> </div>
<input type="button" name="audio" id="audio" value="Carga musica" onclick="cargaAudio()"/><br />
<label id="etiqueta"> Estado de la reproduccion </label>
</body>
```



HTML5 Drag/Drop

- Drag & Drop: arrastrar y soltar forma parte del estándar, no es necesario hacer uso de funciones javascript (JQuery tiene algunas) que lo implementen. Esto nos permite:
  - Seleccionar un elemento que habremos marcado con arrastrable .
  - Arrastrarlo a otra posición que permite que el objeto se quede en ella.

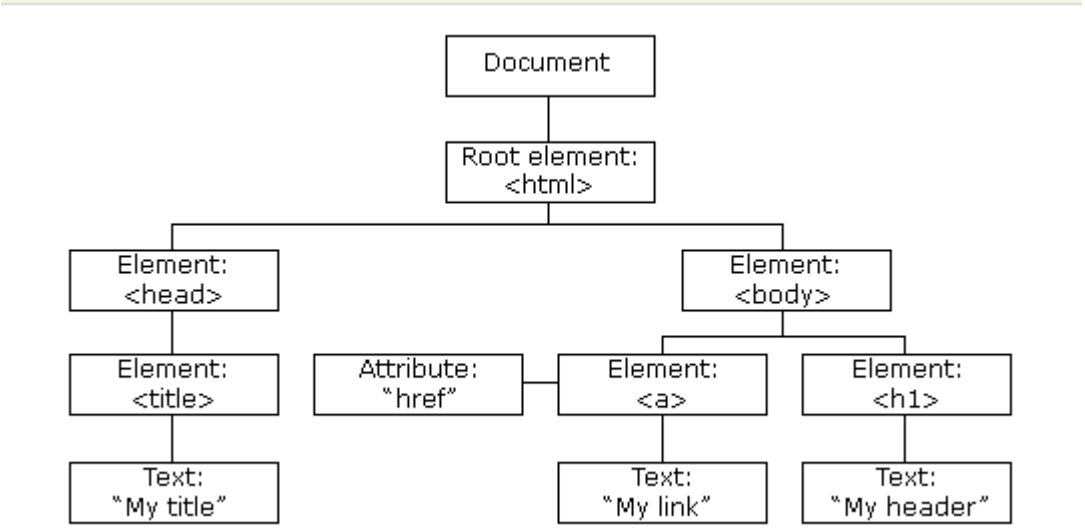
  - http://www.w3schools.com/html/html5_draganddrop.asp
  - http://www.w3.org/html/wg/drafts/html/master/editing.html#dnd

DOM

- Para poder mover elementos de nuestra página necesitamos acceder al DOM y a las funciones que permiten acceder a los nodos y cambiarlos de posición.
  - http://www.w3schools.com/js/js_htmldom_elements.asp
  - http://www.w3schools.com/htmldom/default.asp

DOM

- El HTML DOM genera un árbol por cada página HTML. Cada elemento HTML se corresponde con un nodo.
- Mediante Javascript podemos acceder a esos nodos y recoger valores o modificar elementos



Ejemplo de árbol generado para un documento. w3chools.com.

DOM

- Además, para cada tipo de nodo tenemos un conjunto de métodos y propiedades.
- Algunos métodos generales que nos serán de utilidad:
  - getElementById(*id*) – devuelve el nodo especificado con un id
  - getElementsByClassName()
  - appendChild(*node*), removeChild(), replaceChild(), insertBefore() – inserta, elimina, reemplaza un nuevo nodo hijo
  - cloneNode(bool) –bool = true si se quiere clonar un nodo y false si se quiere clonar el contenido

```
var c = document.getElementById('padre').cloneNode(false);
```

  - createAttribute(), createElement(), createTextNode()
  - getAttribute(), setAttribute()

DOM

- Y algunas propiedades:
  - innerHTML – el texto dentro de un nodo
  - parentNode, childNodes – nodo padre o hijos de uno
  - attributes – atributos de un nodo

DOM. Ejemplos de uso.

```
<p id="p1">Hello World!</p>
<script>
   document.getElementById("p1").innerHTML="New text!";
</script>

<script>
   document.getElementById("p1").style.color="blue";
</script>

<script>
   var p=document.createElement("p");
   var node=document.createTextNode("This is new.");
   p.appendChild(node);

   var element=document.getElementById("p1");
   element.appendChild(p);
</script>
```

Drag & Drop

- Para el drag & drop tenemos:

1. Eventos que se añaden en el elemento que se mueve y en el que recibe.
2. Atributos que permiten elementos que se pueden mover o pueden recibir.
3. Un sistema de almacenamiento de los datos a ser transferidos mediante un interfaz *DataTransfer*

Drag & Drop

- Los eventos que se generan durante el drag & drop son:
  - **dragstart**: se inicia la operación drag & drop. Sirve, por ej.,para cambiar la apariencia de un elemento cuando empezamos a arrastralo.
  - **dragenter, dragleave, dragover:** diferentes pasos del proceso que sirven pueden utilizarse para dar pistas visuales de donde entra o sola el elemento arrastrado.
  - **drag**: continua la operación de arrastre
  - **drop**: cuando un elemento recibe el arrastrado
  - **dragend**: cuando acaba la operación de arrastre (si es modificado su apariencia en el dragstart aquí podríamos volver al valor inicial).

Drag & Drop

- Los atributos de los elementos:
  - **draggable**: se añade en el origen para especificar que ese elemento podrá ser arrastrable.
  - **dropzone**: se añade en el destino y permite especificar el tipo de datos que puede aceptar (texto, imágenes..).

Drag & Drop. Transferencia de datos.

- **dataTransfer**: propiedad del evento “DragEvent” que se utiliza para almacenar los datos que van a moverse. En el evento dragstart establecemos los elementos que se mueve y en el evento drop los recogemos.
- Métodos:
  - **setData** (format, data): añade los datos que se moverán

    ```
    e.dataTransfer.setData('text/html', this.innerHTML);
    ```

  - **getData** (format): recoge los datos

    ```
    this.innerHTML = e.dataTransfer.getData('text/html');
    ```

  - **clearData**([format]): elimina los datos

Drag & Drop. Transferencia de datos.

- Propiedades de dataTransfer:
  - **effectAllowed**: limite el tipo de arrastre permitido.
    - none | copy | copyMove | link | linkMove | move |all | uninitialize

```
e.dataTransfer.effectAllowed = 'move';
```

  - **dropEffect**: devuelve/modifica el tipo de operación activa, falla si no es una permitida
    - none | copy | link | move
  - **items**: lista de items que se arrastran
  - **types**: array con los formatos establecidos en el evento dragstart
  - **files**: lista de ficheros que se son arrastrados (si es el caso)

Drag & Drop. Ejemplo.

- Ejemplo básico de D & D:

http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_draganddrop)

  - Hacemos un elemento arrastrable y recogemos el evento “ondragstart”

```
<img draggable="true” ondragstart=“drag(event)”>
```

  - Especificar el comportamiento cuando se arrastra, decimos que los datos se almacenarán para ser transferidos:

```
function drag(ev){
   ev.dataTransfer.setData(“image/gif",ev.target.id);
}
```

Drag & Drop. Ejemplo.

- Al elemento sobre el que es arrastrado otro le añadimos dos eventos:
  - **ondragover**: donde especificamos que este elemento permite soltar otros en él.
  - **ondrop**: donde especificamos el comportamiento cuando se suelte allí el elemento

```
<div id="div1" ondrop = "drop(event)"
   ondragover = "allowDrop(event)"> </div>
```

Drag & Drop. Ejemplo.

- Por defecto los elementos no permiten que otros se puedan soltar dentro, hay que modificar su valor por defecto llamando al método *preventDefault()*

```
function allowDrop(ev){
   ev.preventDefault();
}
```

- Al soltar cogeremos los datos transferidos desde el arrastre y los añadiremos colgándolos como un nuevo nodo:

```
function drop(){
   ev.preventDefault();
   var data=ev.dataTransfer.getData('image/gif');
   ev.target.appendChild(document.getElementById(data));
}
```

Drag & Drop

- Algunos ejemplos:
  - http://www.html5rocks.com/es/tutorials/dnd/basics/#toc-dragstart

HTML5 Geolocation

- Si el navegador lo soporta podemos obtener las coordenadas mediante el API de geolocalización:
  - http://www.w3.org/TR/geolocation-API/
- Con esto se consultan las coordenadas que el navegador puede obtener mediante GPS, dirección de red, Wifi… Dependiendo de cómo se obtengan serán más precisas o menos.
- Según tipo de conexión y sistema, necesitaremos **subir la página al servidor** para que funcione correctamente.
- El objeto **navigator** contiene un objeto **geolocation** con el método getcurrentPosition() al que le pasamos una función de callback que recibirá un objeto Position.

HTML5 Geolocation

```
function getLocation(){
    if (navigator.geolocation){
    navigator.geolocation.getCurrentPosition(showPosition);
       }
    else{
   alert("Geolocation not supported ”);}
   }

function showPosition(position){
var latitud = position.coords.latitude;
var longitud = position.coords.longitude;
alert('Tus coordenadas son: ('+latitud+','+longitud+')‘);
}
```

http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_geolocation

Otras funcionalidades de HTML5

- Web storage: sustituye a las cookies
  - http://www.w3schools.com/html/html5_webstorage.asp
- Web workers:
  - En Javascript la ejecución de los scripts paraliza el resto de la página hasta que éste se ejecuta.
  - Para evitar esto se pueden crear hilos, llamados web workers en javascript, que ejecutarán los scripts de forma paralela sin bloquear la página.
    - http://www.w3schools.com/html/html5_webworkers.asp
    - http://www.w3.org/TR/workers/

Bibliografía y referencias

- HTML 5 and CSS3: Visual QuickStart Guide, Seventh Edition. E. Castro, B. Hyslop. Safari Books Online:
  - http://proquest.safaribooksonline.com/book/web-development/html/9780131382022

- HTML5 Media. S. Powers. O'Reilly media.
  - http://proquest.safaribooksonline.com/book/web-development/html/9781449308063

Bibliografía y referencias

- HTML5.1. Candidate recomendation diciembre 2012:
  - http://www.w3.org/TR/html5/
- Versiones:
  - http://www.w3.org/standards/history/html5
- W3C schools:
  - http://www.w3schools.com/html/html5_intro.asp
- Microdata y microformats:
  - http://www.w3.org/html/wg/drafts/microdata/master/
  - http://microformats.org/

Bibliografía y referencias. Media.

- w3c – HTML5 Embedded content:
  - http://www.w3.org/html/wg/drafts/html/master/embedded-content-0.html#embedded-content-0
  - http://www.w3schools.com/html/html5_video.asp
  - http://www.w3schools.com/html/html5_audio.asp
- Controlling Media with javascript, apple developers:
  - https://developer.apple.com/library/safari/#documentation/AudioVideo/Conceptual/Using_HTML5_Audio_Video/ControllingMediaWithJavaScript/ControllingMediaWithJavaScript.html#//apple_ref/doc/uid/TP40009523-CH3-SW1
- Using HTML5 audio and video:
  - https://developer.mozilla.org/en-US/docs/HTML/Using_HTML5_audio_and_video