Diseño Web Responsive HTML5 y CSS3

Unidad 6: Multimedia y Formularios

Contacto: info@elearning-total.com
Web: www.elearning-total.com

Indice

Objetivos

Que el alumno logre:

- Conocer los nuevos elementos semánticos incorporados en la versión HTML5.

Multimedia en HTML 5

La nueva versión del estándar se preocupa sobre todo por la semántica de las páginas web. En el campo del multimedia, incorpora dos nuevas directivas que actúan como contenedores de video y audio: **<video>** y **<audio>**.

Con **<video>**, el navegador ya contaría con la capacidad de interpretar correctamente el contenido, sin requerir de los elementos externos.

<VIDEO>

Repasando la historia de Internet, podemos detenernos en el año 2004. No existía Youtube. Por lo mismo, si querías alojar y compartir en un navegador cualquier vídeo había que enfrentarse a los problemas de plataforma, de ancho de banda y de usabilidad para el usuario final.

Si nos encontrábamos con algún enlace que llevara a ver videos en la web seguramente empezarían a aparecer los logos de Real Player, de Windows Media Player o incluso de Quicktime.

Del lado del servidor los sysadmins tenían que luchar contra Real Media Server, Windows Media Server que era parte del IIS y otro montón de opciones, pero era complicado unificar. Adobe por su parte había logrado importantes avances para que desde archivos .swf pudieras incorporar videos, pero no había un solo canal.

En el año 2005, nace Youtube y muchos de esos problemas se solucionan (o al menos aparece una opción más simple de incorporar video a la web).

Paralelamente, Adobe demostró que su esfuerzo para crear un sistema para reproducir vídeo iba a ser valorado y aprovecharía el dominio que su plugin de flash tenía en todos los navegadores del planeta.

En abril del 2010 una carta de Steve Jobs (CEO de Apple) sobre sus pensamientos de Flash arranca uno de esos nuevos ciclos donde una plataforma disruptiva que cada día ganaba mercado le daba la espalda al famoso plugin de Adobe en sus dispositivos móviles.

Apple apoyaba el desarrollo de HTML5 porque todo se podía resolver en temas de vídeo con un tag apropiado para ese fin y aprobado por el estándar.

Hoy volvemos a sobre una batalla de grandes empresas por el codec luego de que están todos de acuerdo que el navegador lleva una opción amigable para incluir un elemento de video, porque este elemento es tan importante como las imágenes.

Y podemos usar perfectamente el ejemplo de las imágenes para entender lo que pasa con el video.

Todos los navegadores permiten que en un documento HTML incluyas una etiqueta <img> a la cual debemos indicarle un atributo con la ruta a una imagen. Y la imagen puede ser un .jpg, un .gif, un .bmp o un .png (entre otros formatos). Dependerá de la capacidad del navegador de cada usuario para reconocer todos los contenidos de esta imagen, interpretarla y mostrarla. Con el video pasa lo mismo.

En esta nueva versión de HTML, para incorporar un archivo de video solo debemos contar con una etiqueta destinada a este fin: **<video>**

Esta etiqueta se encarga de dar soporte a contenido audiovisual que se interpretaría directamente en el navegador. En este espacio, se puede incluir tanto un clip como un streaming.

Estos son sus principales atributos:

- **src**: permite especificar la URL del video.
- **audio**: se trata de un atributo que nos permite indicar el estado del audio del video correspondiente. Y, si está disponible la opción, se puede asignar el valor determinado **muted**.
- **autoplay**: indica que el video se ejecutara automáticamente tras su descarga.
- **loop**: indica que el video se volverá a ejecutar después de terminar.
- **height**, **width**: tamaño en pantalla (la altura y anchura respectivamente en pixels o bien en porcentaje) del video.
- **controls**: indica si hay controles embebido del video. La interpretación final de este atributo va a depender mucho de la implementación de los navegadores.
- **poster**: nos brinda la posibilidad de indicar la URL de una imagen que represente al video, que se mostrará cuando el video no está disponible o no sea accesible.
- **preload**: se utiliza para indicar que el video cargue junto con la página y que esté listo para iniciarse.

Es posible especificar varios formatos alternativos para un mismo video para asegurarse de que el navegador es capaz de interpretar alguno de ellos. Para ello no hay que incluir el atributo @src y usar un elemento nuevo: **<sourse>**.

Este elemento acepta varios atributos: @src, para indicar la dirección de cada fichero; @type contiene el tipo MIME del fichero; y @codec contiene información adicional sobre la forma de interpretarlo.

Ejemplo de video básico (sin elementos de control)

```
<video src="tu_video.ogv">Tu navegador no soporta HTML5 </video>
```

Ejemplo de video con controles

```
<video controls src="http://devfiles.myopera.com/articles/2642/sintel-trailer.ogv">Tu navegador no soporta HTML5 </video>
```

Ejemplo de video con atributo @poster:

```
<video controls src="tu_video.ogv" poster="orang-utan.jpg" >Tu navegador no soporta HTML5 </video>
```

Esta imagen se mostrará cuando el navegador no tenga soporte para la etiqueta o cuando no reconozca la ruta o el formato del video.

No todos los navegadores soportaran el video en formato .ovg para ello podemos agregar una variación a nuestro código. Esto puede solucionarse agregando un segundo **<source>** para que si nuestro formato .ogv no es aceptado se use otro formato.

```
<video autoplay controls>
<source src="tu_video.ogv" type="video/ogg" />
<source src="tu_video.mp4" type="video/mp4" />
<p>Tu navegador no soporta HTML5</p>
 </video>
```

Y para los navegadores que no soportan html5 agregaremos el video en flash con .flv.

```
<video autoplay controls>
<source src="tu_video.ogv" type="video/ogg" />
<source src="tu_video.mp4" type="video/mp4" />
<object width="160" height="90" data="video.flv">
      <param name="movie" value="tu_video.flv">
      <embed src="tu_video.flv" width="160" height="90">
</object>
<p>Tu navegador no soporta HTML5 ni Flash</p>
 </video>
```

Video y Compatibilidad de Navegadores

Se deben abordar dos cuestiones muy importantes cuando se trata el video web con HTML5 y la compatibilidad. Una de ellas es, la de que navegador funciona con que formato de video.

En este punto, ir más allá de lo que se puede probar, es un poco arriesgado. Sin embargo, podemos examinar cuatro tipos diferentes de contendores de archivos y codecs y utilizarlos.

El formato del contenedor 3GP está relacionado con MPEG-4 pero, en realidad, es un formato H.263 y su principal aplicación es para dispositivos móviles como iphone.

Podemos ver la matriz de compatibilidad de los principales navegadores en los que se han llevado a cabo las pruebas de video.

http://caniuse.com/#feat=video

<AUDIO>

De la misma forma, la ejecución de sonidos se canalizaría a través de esta directica en lugar de requerir la inclusión de elementos externos.

El objetivo de esta etiqueta es permitir la carga y ejecución de archivos de audio sin requerir un plug-in de Flash, Silverlignt o Java.

El comité de estandarización W3C deja abierto a cada empresa que desarrolla navegadores los formatos que quieran soportar (así tenemos que algunos soportan mp3, wav, ogg, au)

Un ejemplo de disponer el elemento audio dentro de una página sería:

```
<audio src="sonido.ogg" autoplay controls loop></audio>
```

Estos son sus principales atributos:

- **src:** es un atributo único con la dirección del fichero de audio.

- **autoplay**: indica que el video se ejecutara automáticamente tras su descarga.
- **loop**: indica que el audio se volverá a ejecutar tras terminar, y el número de veces que va a hacerse.
- **controls**: indica si hay controles embebido del audio. Como en el caso del video, la interpretación va a depender mucho de la implementación de los navegadores.
- **preload**: permite indicar si el audio se cargará con la página. Puede tomar los valores: **auto**, **metadata** y **none**.

Como no hay un formato de audio universalmente adoptado por todos los navegadores el elemento audio nos permite agregarle distintas fuentes (igual que el video):

```
<audio controls autoplay loop>
        <source src="sonido.ogg">
        <source src="sonido.mp3">
        <source src="sonido.wav">
        <source src="sonido.au">
Tu navegador no soporta esta etiqueta
</audio>
```

El elemento source indica a través de la propiedad src la ubicación del archivo de audio respectivo.

El orden que disponemos estas fuentes es importante. Primero el navegador busca la primera fuente y verifica que puede reproducir dicho archivo, en caso negativo pasa a la siguiente fuente. Como vemos si queremos que cualquier navegador reciba un audio podemos inclusive hacer el aplique de Flash respectivo por si el navegador no implementa el elemento **AUDIO** o no soporta el formato de archivo.

Por el momento no hay un formato con soporte para todos los navegadores, dependerá del sitio que implementemos para ver si tiene sentido duplicar nuestros archivos con distintos formatos de audio.

<CANVAS>

Con el uso de la directiva <CANVAS> se pueden definir áreas de presentación independientes sin requerir argucias de programación en Javascript.

Cada “canvas” o “lienzo” definido es un área bidimensional en el que se puede dibujar figuras o reproducir imágenes. Pueden definirse varios canvas en una misma página y hacer que se alternen para estar en primer plano lo que permite obtener efectos visuales muy atractivos.

Con HTML pueden definirse varios canvas dentro de un documento con parámetros que permiten identificarlos (@name) y asignarles unas dimensiones (@width y @height).

A esta etiqueta le dedicaremos la unidad 3 de este módulo debido a las posibilidades de desarrollo con las que cuenta.

Resumen

En esta Unidad…

Trabajamos con los nuevos elementos multimediales de HTML5.

En la próxima Unidad…

Comenzaremos a trabajar con Tipografías, Íconos y Animación en CSS3.