Capítulo 8
Medios

8.1 Vídeo

Los vídeos son un método extremadamente eficaz de comunicación. Nadie puede negar la importancia de los vídeos en los sitios web y aplicaciones de hoy en día, y mucho menos aquellos que se encargan de desarrollar las tecnologías para la Web. Esta es la razón por la que HTML5 incluye un elemento con el único propósito de cargar y reproducir vídeos.

<video>—Este elemento inserta un vídeo en el documento.

El elemento **<video>** incluye los siguientes atributos para declarar el área que ocupa el vídeo y configurar el reproductor.

src—Esta atributo especifica la URL del vídeo a reproducir.

width—Este atributo determina el ancho del área del reproductor.

height—Este atributo determina la altura del área del reproductor.

controls—Este es un atributo booleano. Si está presente, el navegador muestra una interfaz para permitir al usuario controlar el vídeo.

autoplay—Este es un atributo booleano. Si está presente, el navegador reproduce el vídeo automáticamente tan pronto como puede.

loop—Este es un atributo booleano. Si está presente, el navegador reproduce el vídeo una y otra vez.

muted—Este es un atributo booleano. Si está presente, el audio se silencia.

poster—Este atributo especifica la URL de la imagen que se mostrará mientras el navegador espera que el vídeo se reproduzca.

preload—Este atributo determina si el navegador debería comenzar a cargar el vídeo antes de ser reproducido. Acepta tres valores: **none**, **metadata** o **auto**. El primer valor indica que el vídeo no se debería cargar y generalmente se utiliza para minimizar tráfico web. El segundo valor, **metadata**, recomienda al navegador descargar información acerca del recurso, como las dimensiones, la duración, el primer cuadro, etc. El tercer valor, **auto**, solicita al navegador que descargue el archivo tan pronto como sea posible (este es el valor por defecto).

El elemento **<video>** requiere etiquetas de apertura y cierre, y solo algunos parámetros para cumplir su función. La sintaxis es sencilla y solo es obligatorio el atributo **src**.

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

```
  <title>Reproductor de Video</title>
</head>
<body>
  <section>
    <video src="trailer.mp4">
    </video>
  </section>
</body>
</html>
```

Listado 8-1: *Cargando un vídeo con el elemento* `<video>`

El elemento **<video>** carga el vídeo especificado por el atributo **src** y reserva un área del tamaño del vídeo en el documento, pero el vídeo se reproduce. Tenemos que indicarle al navegador cuándo queremos que reproduzca el vídeo o facilitar las herramientas para dejar que el usuario decida. Existen dos atributos que podemos agregar al elemento para este propósito: **controls** y **autoplay**. El atributo **controls** indica al navegador que debería incluir sus propios elementos (botones y barras) para permitir al usuario controlar el vídeo, y el atributo **autoplay** solicita al navegador que comience a reproducir el vídeo tan pronto como pueda.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Reproductor de Video</title>
</head>
<body>
  <section>
    <video src="trailer.mp4" controls autoplay>
    </video>
  </section>
</body>
</html>
```

Listado 8-2: *Activando los controles por defecto*

Hágalo usted mismo: cree un nuevo archivo HTML con el documento del Listado 8-2. Descargue el vídeo trailer.mp4 desde nuestro sitio web. Abra el documento en su navegador. El navegador debería comenzar a reproducir el vídeo de inmediato y facilitar botones para controlarlo.

Por defecto, los navegadores determinan el tamaño de área del vídeo a partir de su tamaño original, pero podemos definir un tamaño personalizado con los atributos **width** y **height**. Estos atributos son como los atributos del elemento **<img>**, declaran las dimensiones del elemento en píxeles. Cuando están presentes, el tamaño del vídeo se ajusta para adaptarlo a estas dimensiones, pero no tienen el propósito de comprimir o expandir el vídeo. Podemos usarlos para limitar el área ocupada por el medio y preservar la coherencia en nuestro diseño.

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

```
<head>
  <meta charset="utf-8">
  <title>Reproductor de Video</title>
</head>
<body>
  <section>
    <video src="trailer.mp4" width="720" height="400" controls>
    </video>
  </section>
</body>
</html>
```

Listado 8-3: *Definiendo el área del vídeo*

Lo básico: si el vídeo se incorpora en un sitio web con diseño web adaptable, puede ignorar los atributos **width** y **height**, y declarar su tamaño por medio de CSS y media queries. El elemento **<video>** se puede adaptar al tamaño de su contenedor, como en el caso de las imágenes del Capítulo 5.

El elemento **<video>** incluye atributos adicionales que pueden resultar útiles en algunas aplicaciones. Por ejemplo, el atributo **preload** solicita al navegador que comience a descargar el vídeo tan pronto como pueda, de modo que cuando el usuario decide reproducirlo, la reproducción comienza de inmediato. También contamos con el atributo **loop** para reproducir el vídeo una y otra vez, y con el atributo **poster** para especificar una imagen que se mostrará en lugar del vídeo mientras este no se reproduce.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Reproductor de Video</title>
</head>
<body>
  <section>
    <video src="trailer.mp4" width="720" height="400" preload controls
loop poster="poster.jpg">
    </video>
  </section>
</body>
</html>
```

Listado 8-4: *Incluyendo una imagen que represente el vídeo*

El documento del Listado 8-4 carga el vídeo tan pronto como se carga el documento, reproduce el vídeo continuamente y muestra una imagen en lugar del vídeo mientras este no se reproduce. La Figura 8-1 muestra lo que vemos antes de pulsar el botón para iniciar la reproducción.

Hágalo usted mismo: actualice su documento con el código del Listado 8-4. Descargue el archivo poster.jpg desde nuestro sitio web y abra el documento en su navegador. Debería ver algo similar a la Figura 8-1.

Figura 8-1: *Poster*
© Derechos Reservados 2008, Blender Foundation/www.bigbuckbunny.org

Formatos de vídeo

En teoría, el elemento **<video>** por sí solo debería ser más que suficiente para cargar y reproducir un vídeo, pero el proceso es un poco más complicado en la vida real. Esto se debe a que, a pesar de que el elemento **<video>** y sus atributos son estándar, no existe un formato de vídeo estándar para la web. El problema es que algunos navegadores admiten un grupo de codificadores y otros no, y el codificador que se usa en el formato MP4 (el único que admiten navegadores destacados como Safari e Internet Explorer) se distribuye bajo licencia comercial.

Las opciones más comunes actualmente son OGG, MP4, y WebM. Estos formatos son contenedores de vídeo y audio. OGG contiene codificadores de vídeo Theora y audio Vorbis, MP4 contiene H.264 para vídeo y AAC para audio, y WebM usa VP8 para vídeo Vorbis para audio. Actualmente, OGG y WebM son compatibles con Mozilla Firefox, Google Chrome y Opera, mientras que MP4 funciona en Safari, Internet Explorer y Google Chrome.

HTML contempla este escenario e incluye un elemento adicional que funciona con el elemento **<video>** para definir las posibles fuentes del vídeo.

<source>—Este elemento define una fuente para un vídeo. Debe incluir el atributo **src** para indicar la URL del archivo.

Cuando necesitamos especificar múltiples fuentes para el mismo vídeo, tenemos que reemplazar el atributo **src** en el elemento **<video>** por elementos **<source>** entre las etiquetas, como en el siguiente ejemplo.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Reproductor de Video</title>
</head>

<body>
  <section>
    <video width="720" height="400" controls>
      <source src="trailer.mp4">
```

```
      <source src="trailer.ogg">
    </video>
  </section>
</body>
</html>
```

Listado 8-5: *Creando un reproductor de vídeo para varios navegadores*

En el Listado 8-5 el elemento **<video>** se expande. Ahora, entre las etiquetas del elemento hay dos elementos **<source>**. Estos elementos facilitan diferentes fuentes de vídeo para que el navegador elija. El navegador lee estos elementos y decide qué archivo se debería reproducir de acuerdo con los formatos que admite (en este caso, MP4 u OGG).

Hágalo usted mismo: cree un nuevo archivo HTML con el documento del Listado 8-5. Descargue los archivos trailer.mp4 y trailer.ogg desde nuestro sitio web. Abra el documento en su navegador. El vídeo se debería reproducir como siempre, pero ahora el navegador selecciona qué fuente usar.

IMPORTANTE: los navegadores requieren que los vídeos se envíen en el servidor con los correspondientes tipos MIME. Cada archivo tiene un tipo MIME asociado para indicar el formato de su contenido. Por ejemplo, el tipo MIME para un archivo HTML es **text/html**. Los servidores ya están configurados para la mayoría de los formatos de vídeo, pero normalmente no para nuevos formatos como OGG o WEBM. La forma de incluir este nuevo tipo MIME depende del tipo de servidor. Una manera sencilla de hacerlo es agregar una nueva línea al archivo **.htaccess**. La mayoría de los servidores incluyen este archivo de configuración en el directorio raíz de todo sitio web. La sintaxis correspondiente es **Addtype MIME/type extension** (por ejemplo, **AddType video/ogg ogg**).

8.2 Audio

El audio es un medio que no tiene la misma popularidad en la Web que los vídeos. Podemos filmar un vídeo con una cámara personal que será visto por millones de personas, pero lograr el mismo resultado con un archivo de audio sería casi imposible. Sin embargo, el audio aún se encuentra presente en la Web a través de shows de radio y podcasts. Por esta razón, HTML5 también ofrece un elemento para reproducir archivos de audio.

<audio>—Este elemento inserta audio en el documento.

Este elemento trabaja de la misma forma y comparte varios atributos con el elemento **<video>**.

src—Este atributo especifica la URL del archivo a reproducir.

controls—Este es un atributo booleano. Si está presente, activa la interfaz que facilita el navegador por defecto.

autoplay—Este es un atributo booleano. Si está presente, el navegador reproduce el audio automáticamente tan pronto como puede.

loop—Este es un atributo booleano. Si está presente, el navegador reproduce el audio una y otra vez.

preload—Este atributo determina si el navegador debe comenzar a cargar el archivo de audio antes de reproducirse. Acepta tres valores: **none**, **metadata** o **auto**. El primer valor indica que el audio no se debería cargar y generalmente se utiliza para minimizar tráfico web. El segundo valor, **metadata**, recomienda al navegador que descargue información acerca del recurso, como la duración del audio. El tercer valor, **auto**, solicita al navegador que descargue el archivo tan pronto como le sea posible (este es el valor por defecto).

La implementación del elemento **<audio>** en nuestro documento es muy similar a la del elemento **<video>**. Solo tenemos que especificar la fuente y ofrecer al usuario la posibilidad de iniciar la reproducción.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Reproductor de Audio</title>
</head>
<body>
  <section>
    <audio src="beach.mp3" controls>
    </audio>
  </section>
</body>
</html>
```

Listado 8-6: *Reproduciendo audio con el elemento* <audio>

Nuevamente tenemos que hablar de codificadores, y una vez más debemos decir que el código HTML del Listado 8-6 debería ser suficiente para reproducir un sonido en la Web, pero no lo es. MP3 se encuentra bajo licencia comercial, por lo que no es compatible con navegadores como Mozilla Firefox u Opera. Vorbis (el codificador de audio del contenedor OGG) es compatible con estos navegadores, pero no con Safari e Internet Explorer. Por lo tanto, una vez más, tenemos que usar el elemento **<source>** para facilitar al menos dos formatos para que el navegador pueda elegir.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Reproductor de Audio</title>
</head>
<body>
  <section>
    <audio id="medio" controls>
      <source src="beach.mp3">
      <source src="beach.ogg">
    </audio>
  </section>
```