

2.6. ELEMENTOS MULTIMEDIA

Indudablemente no se puede concebir una página web hoy en día sin incluir en ella elementos visuales atractivos. En este sentido se conoce como multimedia el hecho de incorporar medios procedentes de diversos formatos.

2.6.1. Vídeo

Inserción de video:

```
<video src="ejemplo.mp4"></video>
```

Atributos:

atributo	uso
src	URL al vídeo que se desea mostrar
width	Anchura del vídeo en nuestra página
height	Altura del vídeo en nuestra página
autoplay	Usa el valor fijo <i>autoplay</i> (o no se indica valor alguno) para indicar que el vídeo se inicia automáticamente en cuanto se descargue
loop	Usa el valor fijo <i>loop</i> (o no se indica valor alguno) para indicar que el vídeo se ejecuta automáticamente una y otra vez
controls	Con valor <i>controls</i> (o no indicando valor alguno) indica que el navegador mostrará controles para que el usuario pueda controlar la reproducción del vídeo de reproducción (pausa, play,...)
preload	Indica al navegador una recomendación sobre cómo debemos realizar la descarga. Posibilidades: <ul style="list-style-type: none"> ▪ auto. El vídeo se descarga en cuanto se carga la página ▪ none. El vídeo no se descarga. Cuando el usuario pulse play, se descargará. ▪ metadata. Descarga los metadatos del vídeo, no el vídeo en sí
poster	Permite indicar la dirección URL a una imagen que se mostrará mientras el vídeo no se está reproduciendo. Si no se usa este atributo, se usa el primer fotograma del vídeo como póster.
muted	Atributo sin valor que marca que el vídeo se debe reproducir sin audio.

Ejemplo:

```
<video src="carrera.mp4" controls autoplay  
      poster="f1.jpg">  
</video>
```

Se mostrará el vídeo *carrera.mp4* de modo que se empezará a reproducir en cuanto se cargue. Antes de la carga del vídeo se mostrará la imagen *f1.jpg*, el vídeo se reproducirá inmediatamente (si que haga falta que el usuario pulse play).

Dentro del elemento **video**, podemos incorporar elementos **source** que hagan referencia a distintas fuentes de vídeo. **source** dispone de estos atributos:

Ejemplo:

```
<video autoplay="autoplay" controls="controls"  
poster="foto1.jpg" >  
  <source src="video.mp4"  
    type="video/mp4;codecs='avc1.42E01E, mp40a.40.2' " >  
  <source src="video.ogv"  
    type="video/ogg;codecs='theora, vorbis' " >  
  No se puede mostrar el vídeo en este navegador  
</video>
```

Dentro de **type**, el uso de codecs es opcional, ya que si el navegador no reconoce el formato no suele hacer caso a los **codecs** que se indiquen (aunque a veces les descarga). La idea es que si el primer formato no se reconoce (primer elemento **source**), se intenta el segundo y así sucesivamente. Si ninguno es reproducible por el navegador actual, éste mostrará la frase final tras el último **source**.

2.6.2. Audio

Insertar audio en una página web es muy similar a insertar vídeo. El elemento que lo hace posible, se llama precisamente audio. Sus atributos (que funcionan igual que con el elemento **video**) son:

atributo	valor
src	URL a la fuente de audio que estamos insertando
autoplay	El audio se reproducirá inmediatamente no se esperará a que el usuario pulse el botón Play
loop	El audio se repetirá continuamente
controls	Se muestran los controles de ese audio (play , rewind , pause , etc.)
preload	Recomendación sobre cómo debemos realizar la descarga. Posibilidades: <ul style="list-style-type: none"> ▪ auto. El audio se descarga en cuanto se carga la página ▪ none. El audio no se descarga. Cuando el usuario pulse play, se descargará. ▪ metadata. Descarga los metadatos del audio, no el audio en sí

Con el audio hay el mismo problema con la cuestión de los codecs y los formatos, por lo que también es habitual convertir el audio a distintos formatos y dar opciones dentro de la etiqueta audio gracias a la etiqueta **source**. Ejemplo:

```
<audio controls="controls" autoplay>
  <source src="audio1.ogg" type="audio/ogg">
  <source src="audio2.mp3" type="audio/mpeg">
  No se puede reproducir el archivo de audio
</audio>
```

2.6.3. Subtítulos

Se realiza mediante elementos de tipo **track**, los cuales se colocan dentro de la etiqueta **video**. Desde hace un par de años, casi todos los navegadores manejan los subtítulos, aunque todavía hay algunas carencias al respecto.

Los subtítulos son textos que permiten traducir el contenido del vídeo. Por ejemplo, un vídeo podría estar grabado en inglés y los subtítulos nos enseñarían el texto traducido en español a tiempo real. Hay varios formatos de subtítulos, pero el estándar en la web es **WebVTT** (*Web Video Text Tracks*). Normalmente se utilizan archivos de texto cuya extensión es vtt basados en este estándar.

Ejemplo de archivo vtt:

```
WEBVTT
1
00:00:00.150 --> 00:00:01.300
¿Habéis visto a Sara?
2
00:00:01.540 --> 00:00:03.800
-No
-Yo tampoco la he visto
```

En este archivo la primera frase se muestra en el primer segundo del vídeo. En los dos segundos siguientes se mostrará el otro texto. En el navegador Chrome, y en algunos otros, se permite dar formato CSS al texto de los subtítulos.

El elemento track que permite asignar subtítulos a un vídeo posee los siguientes atributos:

atributo	uso
src	URL al archivo VTT o SRT que contiene los subtítulos
kind	Tipo de subtítulos. Posibilidades: <ul style="list-style-type: none"> ▪ subtitles. Subtítulos normales ▪ captions. Igual que el anterior pero con posibilidad de añadir sonidos, música y otros elementos más allá de la traducción del diálogo. El caso más utilizado este tipo de "tracks" son los audios especiales para personas con problemas de visión o textos para personas con problemas de audición. ▪ descriptions. Texto alternativo e independiente de la película. También muy utilizado para ayudar a interpretar el audio del vídeo a las personas sordas. ▪ chapters. Contiene los capítulos que permiten navegar de forma ágil por la película ▪ metadata. Metadatos de la película. No visibles para los usuarios, se usan con JavaScript.
label	Nombre de los subtítulos para reconocerlos en caso de haber indicado varios. Es el texto con el que aparece el nombre del idioma en el navegador al elegir un subtítulo u otro.
srclang	Código de idioma (oficial según la norma ISO BCP 47) de los subtítulos: por ejemplo es para español, en para inglés, fr francés, etc.
default	Indica los subtítulos que se deben de cargar por defecto si no es posible saber el idioma de preferencia del usuario.

Ejemplo:


```
<video autoplay="autoplay" controls="controls" poster="foto1.jpg" >
  <source src="video.mp4" type="video/mp4" >
  <source src="video.ogv" type="video/ogg" >
  <track src="spa.vtt" kind="subtitles" srclang="es"
    label="Español" >
  <track src="deu.vtt" kind="subtitles" srclang="de"
    label="Alemán" >
  El navegador no puede mostrar el vídeo
</video>
```

2.6.4. Canvas

En inglés **canvas** significa *lienzo*, y define muy bien para que sirve este elemento creado en la norma HTML5. Es uno de los componentes de HTML5 más famosos por el gran aporte que ha supuesto al dinamismo de las páginas web.

Mediante este elemento dispondremos de un área que podremos utilizar donde queramos para dibujar elementos gráficos mediante las instrucciones gráficas del lenguaje JavaScript. Eso ha permitido (gracias a la potencia de JavaScript) crear juegos, animaciones y elementos visuales atractivos en las páginas web.

Atributos:

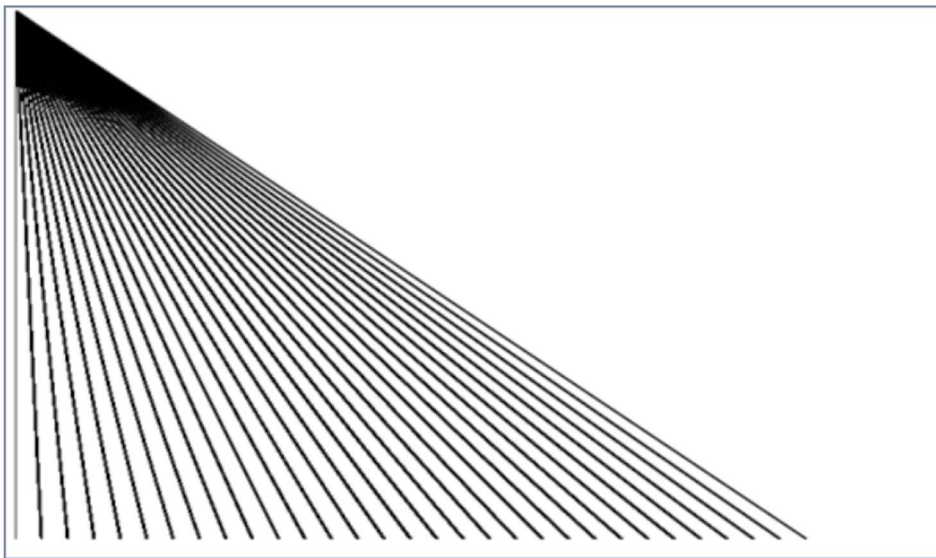
- **id**. Es el atributo que utilizan todos los elementos HTML para identificarlos. En el caso de canvas es casi obligatorio su uso para poder hacer referencia al mismo desde el lenguaje JavaScript.
- **width**. Anchura del lienzo (funciona igual que en el caso de **img**)
- **height**. Altura del lienzo.

Ejemplo de uso de **canvas** (posee código JavaScript):

```
<canvas id="lienzo1" width="600" height="400">
<script type="text/javascript">
    var canvas=document.getElementById("lienzo1");
    var contexto=canvas.getContext("2d");
    contexto.lineWidth=2;
    for(i=0;i<=600;i+=20){
        contexto.moveTo(0,0);
        contexto.lineTo(i,400)
        contexto.stroke();
    }
</script>
```

Mediante la etiqueta script podemos colocar código en lenguaje JavaScript, desde ese código podemos utilizar el lienzo para dibujar.

El resultado del código anterior es la imagen:



2.6.5. SVG

En el tema anterior hemos visto que es posible insertar imágenes SVG (imágenes de tipo vectorial) a través del elemento **img** al igual que las imágenes JPEG o PNG por ejemplo. Sin embargo, las imágenes SVG están construidas en un dialecto de XML que cualquier navegador de hoy en día reconoce.

Es decir, podemos insertar el código SVG directamente en una página web.

Ejemplo:

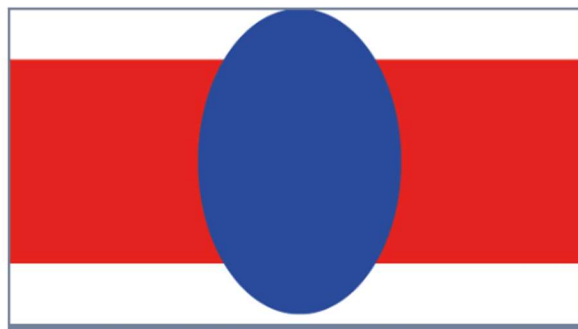
```
<svg xmlns="http://www.w3.org/2000/svg">
  ..etiquetas svg
</svg>
```

Dentro del elemento **svg** se colocan las etiquetas que permiten dibujar en el área del svg. En principio si no indicamos tamaño alguno, todo el área de la página web, se utiliza para dibujar en SVG, pero podemos indicar atributos **width** y **height** para indicar tamaños concretos.

Las etiquetas que se pueden utilizar dentro de **svg** pertenecen al lenguaje estándar SVG. Un ejemplo de ellas sería:

```
<h1>Dibujo SVG</h1>
<svg xmlns="http://www.w3c.org/2000/svg" height="400">
  <rect id="rec1" x="50" y="50" width="300"
    height="100" fill="red" >
  <ellipse id="elips1" cx="200" cy="100" rx="50"
    ry="75" fill="blue" >
</svg>
```

El resultado es:



2.6.6. Imágen

Las imágenes se colocan mediante la etiqueta **img**. El atributo **src** indica la URL (relativa o absoluta) a la imagen. Ejemplo:

```

```

img es una etiqueta sin cierre, la imagen se coloca directamente en la posición de esta etiqueta. Se la considera como texto al maquetarla. De modo que aparece entre el texto si ponemos la etiqueta entre el texto. Colocar de forma adecuada las imágenes respecto al texto es todo un arte que se consigue (una vez más) con ayuda de CSS.

atributo	significado
alt	Obligatorio. Indica un texto alternativo. Ese texto aparece cuando la imagen no se ha podido cargar (o durante la carga). También suele aparecer cuando arrimamos el cursor a la imagen a fin de informarnos sobre ella. Es un texto también tenido en cuenta por los buscadores a fin de identificar lo que muestra la imagen. Deberíamos tomarnos este atributo como obligatorio
width	Anchura de la imagen. No es aconsejable su uso, ya que si la ampliamos no se verá en buena calidad y si la reducimos estaremos cargando una imagen grande para luego mostrarla en pequeño; sería más inteligente reducirla primero con un editor de imágenes. En cualquier caso es importante utilizar este atributo (junto con height) para que el navegador sepa de antemano el tamaño de la imagen y así que prepare la página correctamente. De este modo si la imagen no se carga, al menos veremos el rectángulo que la misma ocuparía y la página no se desbarata.
height	Altura de la imagen. Tiene las mismas connotaciones que el atributo anterior.

HTML5 introdujo un nuevo elemento llamado **picture** para dar mayor versatilidad al uso de imágenes, que ya hemos visto anteriormente.