



CEU

| *Centro de Estudios
Profesionales*

LENGUAJE DE MARCAS
1º DAW – 1º DAM

TEMA 21

Vídeo en HTML



Elemento <video>

El elemento <video> permite incrustar video fácilmente en un archivo html.

Para continuar con el tema vamos a crear una página denominada: Tema21.html:

```
<> tema21.html > html > body
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5      <meta charset="UTF-8">
6      <!-- ETIQUETA PARA HACER RESPONSIVE LA PÁGINA, ADAPTACIÓN DE ESCALAS -->
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Tema 21 html</title>
9      <!-- ICONO DE VENTANA -->
10     <link rel="icon" type="image/x-icon" href="imagenes/favicon.ico">
11     <!-- HOJA DE ESTILOS EXTERNA-->
12     <link rel="stylesheet" href="estilos/estilostema21.css">
13 </head>
14
15 <body>
16
17
18 </body>
```

La creamos con lo básico, creamos su hoja de estilo:

Estilostema21.css:

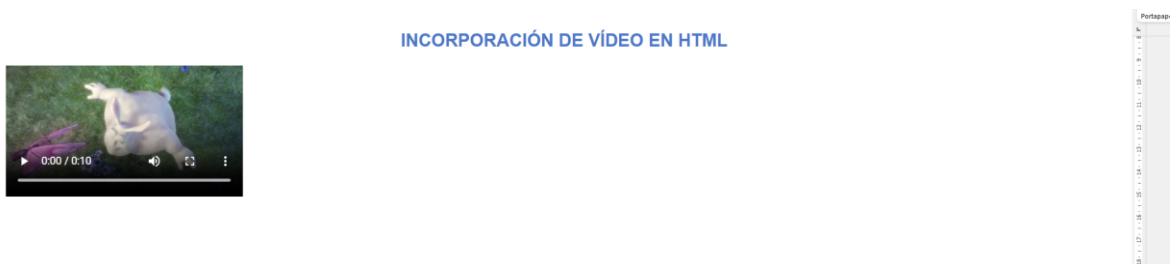


```
estilos > # estilostema20.css > ...
1  body {
2      font-family: Arial, sans-serif;
3      margin-left: 10px;
4      padding: 5px;
5  }
6
7  h2 {
8      color: #4d77ca;
9      text-align: center;
10 }
11
```

Vamos a escribir el siguiente código en nuestro archivo:

```
<body>
  <h2>INCORPORACIÓN DE VÍDEO EN HTML</h2>
  <video src="https://www.w3schools.com/html/mov_bbb.mp4" controls>
    <p>
      Tu navegador no soporta HTML5 video. Aquí está el
      <a href="https://www.w3schools.com/html/mov_bbb.mp4">enlace del video</a>.
    </p>
  </video>
</body>
```

Esto mostrará un reproductor de vídeo en nuestra página html:





En caso de que el navegador no pueda reproducir el vídeo, se ejecutará el código que está en el párrafo:

```
<p>  
  Tu navegador no soporta HTML5 video. Aquí está el  
  <a href="https://www.w3schools.com/html/mov_bbb.mp4">enlace del video</a>.  
</p>
```

Si no se puede reproducir, pues se le proporciona un enlace para abrir el vídeo en su origen.

Añadiendo subtítulos al elemento <video>

Para añadir subtítulos a los vídeos vamos a utilizar javascript, como ya hemos hecho en temas anteriores.

Lo primero es recoger el contenedor del vídeo por ID para poder operar con él. Una vez recogido, se crea una pista nueva “track” para subtítulos utilizando la api denominada “TextTrack”.

```
let textTrack = vid.addTextTrack("captions", "Español", "es");
```

captions: indica que la pista es de subtítulos.

Español: es el título de la pista.

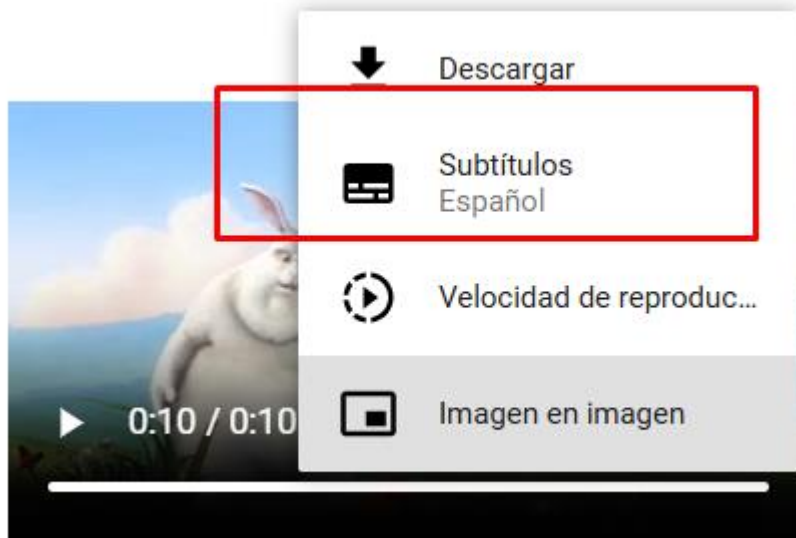
es: indica que el idioma de los subtítulos es español.

La siguiente propiedad hace visible los subtítulos.

```
textTrack.mode = "showing";
```



Si no tuviéramos esta propiedad en “showing” para activar los subtítulos, tendríamos que hacerlo desde las propiedades del vídeo en la propia página html:



Ahí podéis ver el nombre de la pista “Español”.

Por último, se añaden los subtítulos a la pista con la librería “VTTCue”:

```
// Añadir un subtítulo (cue) al text track  
let cue = new VTTCue(1, 4, "UNA MARIPOSA VA VOLANDO...");
```

Se crea una variable que almacena el texto a mostrar, los pármetros 1,4 indican los segundos en los que se muestra el texto: aparecería en el segundo 1 y desaparece en el segundo 4.

Vamos a probar ahora el siguiente código en nuestro vídeo:



```
<script>
  function myFunction() {
    let vid = document.getElementById("video1");

    // Añadir una pista de subtítulos (text track)
    let textTrack = vid.addTextTrack("captions", "Español", "es");
    textTrack.mode = "showing"; // Mostrar los subtítulos inmediatamente

    // Añadir un subtítulo (cue) al text track
    let cue = new VTTcue(1, 4, "UNA MARIPOSA VA VOLANDO....");
    textTrack.addCue(cue);
    cue = new VTTcue(5, 6, "CUANDO SE POSA EN EL SUELO...");
    textTrack.addCue(cue);
    cue = new VTTcue(7, 10, "Y MANZANAZO!!!!!!");
    textTrack.addCue(cue);
  }
  myFunction();
</script>
```

Para añadir los cue (subtítulos) al track que hemos creado, utilizamos el método “addCue”.

Podemos observar como van apareciendo los diferentes subtítulos que hemos ingresado en la pista.



Prueba a comentar la siguiente línea:

textTrack.mode = "showing";

Verás como no aparecen los subtítulos y tienes que activarlos como hemos comentado anteriormente.

Ahora vamos a añadir una nueva pista para otro idioma "Inglés":

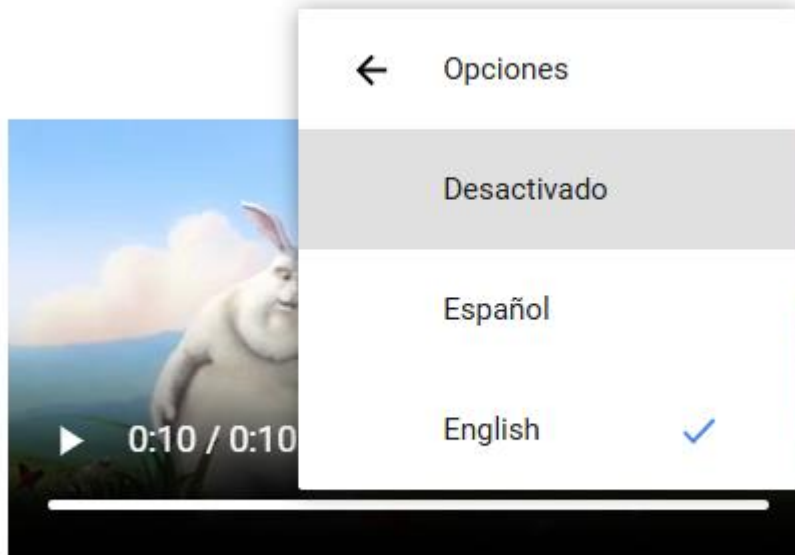
```
// En Inglés:
// Añadir una pista de subtítulos (text track)
let textTrackEN = vid.addTextTrack("captions", "English", "en");
textTrackEN.mode = "hidden"; // Mostrar los subtítulos inmediatamente

// Añadir un subtítulo (cue) al text track
let cueEN = new VTTcue(1, 4, "A BUTTERFLY IS FLYING....");
textTrackEN.addCue(cueEN);
cueEN = new VTTcue(5, 6, "WHEN IT LANCES ON THE GROUND...");
textTrackEN.addCue(cueEN);
cueEN = new VTTcue(7, 10, "AND AN APPLE-TAZO!!!!!!");
textTrackEN.addCue(cueEN);
```



Vemos que se crea otro track, esta vez le ponemos el título de “English” y el lenguaje es “en” de inglés.

Podremos seleccionarlo desde las propiedades del propio vídeo.



Hemos podido observar que el modo de reproducción del track inglés está oculto:

```
textTrackEN.mode = "hidden";
```

De esta forma sale visible por defecto el track de subtítulos en español.

Ahora vamos a ver cómo podemos cargar un fichero de subtítulos en vez de tenerlos que poner en el código.



Añadiendo subtítulos desde un fichero al elemento <video>

Para ello vamos a utilizar el tag “track”:

```
<body>
  <h2>INCORPORACIÓN DE VIDEO EN HTML</h2>
  <video id="video1" src="https://www.w3schools.com/html/mov_bbb.mp4" controls>
    <track src="media/subtitulos-es.vtt" kind="subtitles" srclang="es" label="Español" default>
    <track src="media/subtitulos-en.vtt" kind="subtitles" srclang="en" label="English">
  </video>
  <p>
    Tu navegador no soporta HTML5 video. Aquí está el
    <a href="https://www.w3schools.com/html/mov_bbb.mp4">enlace del video</a>.
  </p>
</body>
```

Podemos observar los parámetros:

src: localización del archivo de subtítulos.

kind: clase de archivo: “subtitles”.

srclang: idioma del archivo.

label: título del track.

default: es el track visible por defecto.

Para probar la carga de estos ficheros lo primero que vamos a hacer es comentar la llamada a la función de antes y segundo preparar esos archivos.



```
// Añadir un subtítulo (cue) al text track
let cueEN = new VTTCue(1, 4, "A BUTTERFLY IS FLYING...");
textTrackEN.addCue(cueEN);
cueEN = new VTTCue(5, 6, "WHEN IT LANCES ON THE GROUND...");
textTrackEN.addCue(cueEN);
cueEN = new VTTCue(7, 10, "AND AN APPLE-TAZO!!!!!!");
textTrackEN.addCue(cueEN);
}
// myFunction();
</script>
```

Una vez comentada la llamada a la función, hemos desactivado todo el proceso de creación en javascript de los tracks de subtítulos.

Ahora vamos a crear los archivos, que deben de estar contenidos en el directorio “media”:

Creamos una carpeta denominada “media” en el directorio donde tenemos el archivo tema21.html. y en la misma creamos el siguiente archivo:

subtitulos-es.vtt (Nombre del archivo de subtítulos en español)

```
media > ≡ subtitulos-es.vtt
1  WEBVTT
2
3  00:00:00.000 --> 00:00:04.000
4  ¡Hola! Esto es un ejemplo de subtítulos en español.
5
6  00:00:04.000 --> 00:00:10.000
7  Puedes añadir más líneas dependiendo de la duración del video.
8
```



Importante que siempre lo primer que aparezca en el archivo sea:
WEBVTT

Podéis observar que los textos van por pares de valor, el tiempo de permanencia del subtítulo y debajo el propio subtítulo:

```
00:00:00.000 --> 00:00:04.000
```

El tiempo va en
horas:minutos:segundos:milisegundos

Ahora creamos nuestro archivo de subtítulos en inglés:

```
media > ≡ subtítulos-en.vtt
1  WEBVTT
2
3  00:00:00.000 --> 00:00:04.000
4  Hello! This is an example of English subtitles.
5
6  00:00:04.000 --> 00:00:10.000
7  You can add more lines depending on the time of the video.
8
```

Ahora podemos ejecutar la página.

Podemos observar que no se muestra nada, si mostramos el inspeccionador de elementos:



no disponible

Unsafe attempt to load URL file:///C:/Users/fgallego/Lenguaje%20de%20Marcas%201%C2%BA%20Daw/media/subtitulos-es.vtt from frame with URL file:///C:/Users/fgallego/Lenguaje%20de%20Marcas%201%C2%BA%20Daw/tema21.html. 'file:' URLs are treated as unique security origins.

Nos encontramos un problema de seguridad, desde el navegador no se deja acceder a un archivo de texto de la unidad C por propia seguridad. Este problema desaparece cuando tengamos la página instalada en un servidor. Pero mientras, tanto podemos hacer el siguiente truquito:

Vamos a desactivar la seguridad de Chrome, esto se puede hacer, abriendo un navegador Chrome con la seguridad deshabilitada.

Tenemos que buscar la ruta en la que se encuentra nuestro Chrome:

DESACTIVAR SEGURIDAD CHROME

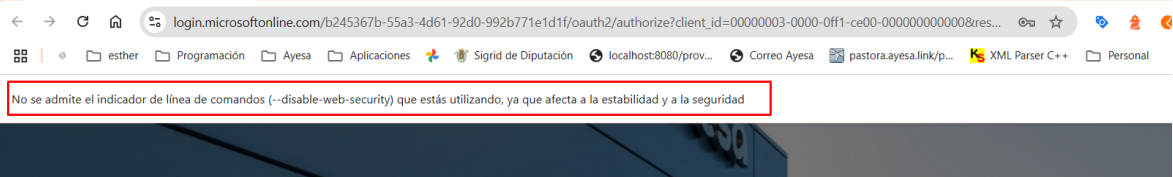
Buscar ruta

cd C:\Program Files (x86)\Google\Chrome\Application

Una vez encontrada ejecutar Chrome con los siguientes parámetros:

chrome.exe --disable-web-security --user-data-dir="C:\chrome_dev"

si se ha abierto correctamente el navegador con la seguridad deshabilitada, saldrá algo parecido a esto:





Ahora aquí si podemos probar nuestro archivo html para ver si carga los archivos de subtítulos.

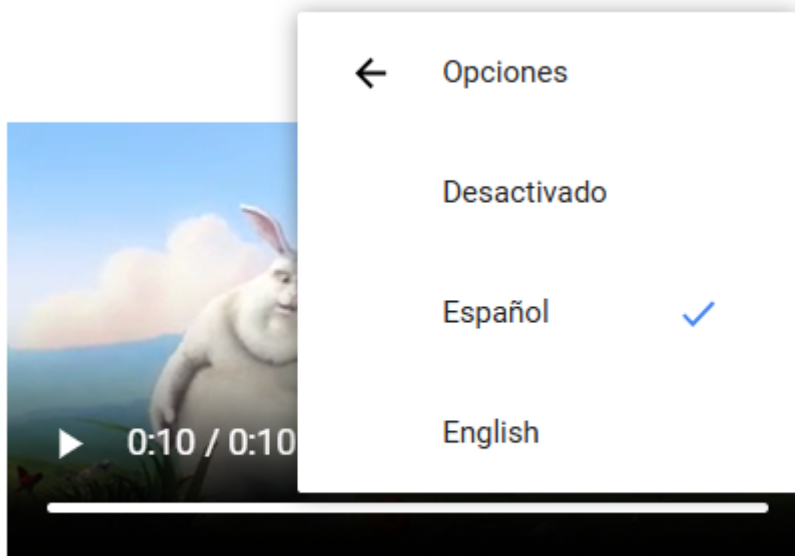
No se admite el indicador de línea de comandos (--disable-web-security) que estás utilizando, ya que afecta a la estabilidad y a la seguridad

INCORPORACIÓN DE VÍDEO EN HTML



Podemos observar que se ha quitado el error de seguridad de acceso y ahora si puede leer los archivos.

Recordemos que podemos cargar el archivo de inglés desde las propiedades del propio vídeo:





Vamos a ver a continuación como podemos reproducir el vídeo automáticamente sin necesidad de darle al play.

Atributo “Autoplay”:

```
<h2>INCORPORACIÓN DE VÍDEO EN HTML</h2>
<video id="video1" src="https://www.w3schools.com/html/mov_bbb.mp4" controls autoplay>
  <track src="media/subtitulos-es.vtt" kind="subtitles" srclang="es" label="Español" default>
  <track src="media/subtitulos-en.vtt" kind="subtitles" srclang="en" label="English">
  <p>
    Tu navegador no soporta HTML5 video. Aquí está el
    <a href="https://www.w3schools.com/html/mov_bbb.mp4">enlace del video</a>.
  </p>
</video>
```

Añadimos y probamos.

Puede pasar que el navegador este restringiendo el uso de autoplay, para que pueda funcionar, podemos añadir el atributo “muted”:

```
<video id="video1" src="https://www.w3schools.com/html/mov_bbb.mp4" controls autoplay muted>
```

Y ahora no debería haber problema en que se reprodujera automáticamente.

Ahora vamos a poder cambiar las propiedades de los textos de los subtítulos:

Añadiremos a nuestro archivo de hoja de estilos: estilostema21.css el siguiente estilo:



```
estilos > # estilostema21.css > video::cue
1  body {
2      font-family: Arial, sans-serif;
3      margin-left: 10px;
4      padding: 5px;
5  }
6
7  h2 {
8      color: #4d77ca;
9      text-align: center;
10 }
11
12 video::cue {
13     color: yellow; /* Color del texto */
14     background: black; /* Fondo del subtítulo */
15     font-size: 16px; /* Tamaño del texto */
16     font-weight: bold; /* Negrita */
17     text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); /* Sombra para mejor visibilidad */
18 }
```

Y probamos:

INCORPORACIÓN DE VÍDEO EN HTML



Ahora vamos a ver cómo podemos cambiar la posición de los subtítulos:

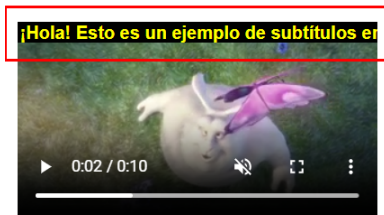
Si editamos los archivos de subtítulos:



```
media > ≡ subtítulos-es.vtt
1 WEBVTT
2
3 00:00:00.000 --> 00:00:04.000 line:0% align:center
4 ¡Hola! Esto es un ejemplo de subtítulos en español.
5
6 00:00:04.000 --> 00:00:10.000 line:0% align:center
7 Puedes añadir más líneas dependiendo de la duración del video.
8
```

Podemos observar que ahora los subtítulos saldrán arriba del todo y alineados al centro, podemos también poner alineación izquierda o derecha.

INCORPORACIÓN DE VÍDEO EN HTML



Esto es para cuando cargamos los subtítulos desde los archivos de texto.

Para mover los subtítulos desde los tracks que se crean desde javascript, tenemos los siguientes métodos:

Para probarlo, comenta el código en el que carga los archivos de subtítulos, luego vuelve a descomentar la llamada a la función y por último incorpora en el primer track de lenguaje español:



```
// Añadir un subtítulo (cue) al text track  
let cue = new VTTCue(1, 4, "UNA MARIPOSA VA VOLANDO...");  
textTrack.addCue(cue);  
cue.position = 1; // 10% desde la izquierda  
cue.line = 0; // 0% desde arriba hacia 100% en la parte más abajo  
cue.align = "left"; // Alineación izquierda
```

Verás como aparece el subtítulo de la siguiente forma:



INCORPORACIÓN DE VÍDEO EN HTML

Posición arriba del todo “line=0”, a la izquierda “position=1” y alineación izquierda con align=left.

Controles personalizados operar con <video> para vídeos

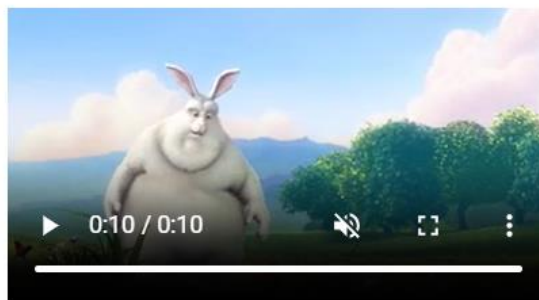
Vamos a añadir el siguiente código a nuestra página html para poner algunos controles:



```
<h2>INCORPORACIÓN DE VÍDEO EN HTML</h2>
<div style="text-align:center">
  <video id="video1" src="https://www.w3schools.com/html/mov_bbb.mp4" controls autoplay muted>
    <!-- <track src="media/subtitulos-es.vtt" kind="subtitles" srclang="es" label="Español" def
  <track src="media/subtitulos-en.vtt" kind="subtitles" srclang="en" label="English"> -->
    <p>
      Tu navegador no soporta HTML5 video. Aquí está el
      <a href="https://www.w3schools.com/html/mov_bbb.mp4">enlace del video</a>.
    </p>
  </video>
  <br>
  <h4>CONTROLES</h4>
  <button onclick="playPause()">Reproducir/Parar</button>
  <button onclick="makeBig()">Aumentar</button>
  <button onclick="makeSmall()">Reducir</button>
  <button onclick="makeNormal()">Normal</button>
</div>
```

Esto nos debería de salir así:

INCORPORACIÓN DE VÍDEO EN HTML



CONTROLES

Reproducir/Parar Aumentar Reducir Normal

Ahora vamos a poner el código javascript que hace estas funciones.

Sacamos esta línea de código a una variable global:



```
<script>

let vid = document.getElementById("video1");

function myFunction() {
  // Añadir una pista de subtítulos (text track)
  let textTrack = vid.addTextTrack("captions", "Español", "es");
  textTrack.mode = "showing"; // Mostrar los subtítulos inmediatamente
```

Porque la vamos a utilizar en nuestras funciones.

Añadimos el siguiente código a nuestro script:

```
function playPause() {
  if (vid.paused)
    vid.play();
  else
    vid.pause();
}

function makeBig() {
  vid.width = 560;
}

function makeSmall() {
  vid.width = 320;
}

function makeNormal() {
  vid.width = 420;
}
```



Ahora podéis probar los botones que hemos puesto antes pues llaman a estas funciones de javascript.

A continuación, vamos a ver más controles sobre el vídeo:

```
<h4>CONTROLES</h4>
<button onclick="playPause()">Reproducir/Parar</button>
<button onclick="makeBig()">Aumentar</button>
<button onclick="makeSmall()">Reducir</button>
<button onclick="makeNormal()">Normal</button>
<br><br>
<button onclick="retroceder()">Retroceder</button>
<button onclick="avanzar()">Avanzar</button>
<button onclick="fullScreen()">Pantalla Completa</button>
</div>
```

Ahora ponemos el código script:

```
function retroceder() {
  vid.currentTime-=1;
}

function avanzar() {
  vid.currentTime+=1;
}

function fullScreen() {
  vid.requestFullscreen();
}
```



Podéis probarlos.

INCORPORACIÓN DE VÍDEO EN HTML



CONTROLES



Ahora vamos a añadir un botón para cambiar la velocidad del vídeo:

Con **playbackRate**

Ejemplo:

```
let video = document.getElementById("miVideo");  
video.playbackRate = 1.5; // 1.5x más rápido
```

Los valores recomendados son:

- 0.5: mitad de velocidad.
- 1: normal
- 1.5: 1.5x más rápido



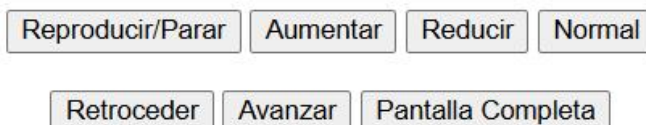
- 2: el doble de velocidad.

Como ejercicio el alumno deberá de crear una fila más de botones debajo de la última fila de controles, que hagan estas acciones:

INCORPORACIÓN DE VÍDEO EN HTML



CONTROLES



VELOCIDAD VÍDEO



Para poder realizar estas pruebas, podemos usar la propiedad del elemento vídeo que hace que se repita indefinidamente: “loop”:

```
<h2>INCORPORACIÓN DE VÍDEO EN HTML</h2>
<div style="text-align:center">
  <video id="video1" src="https://www.w3schools.com/html/mov_bbb.mp4" controls loop autoplay mute
    <!-- <track src="media/subtitulos-es.vtt" kind="subtitles" srclang="es" label="Español" defau
    <track src="media/subtitulos-en.vtt" kind="subtitles" srclang="en" label="English"> -->
```



Una vez probado todo, podéis quitar el atributo loop, ahora vamos a detectar eventos.

Vamos a comprobar cuando ocurre un evento en el vídeo.

Agreguemos el siguiente código a nuestro script:

```
<script>

let vid = document.getElementById("video1");

vid.addEventListener("ended", function () {
  alert("¡El video ha terminado!");
});

function myFunction() {
  // Añadir una pista de subtítulos (text track)
  let textTrack = vid.addTextTrack("captions", "Español", "es");
  textTrack.mode = "showing"; // Mostrar los subtítulos inmedi
```

Ahora cuando el vídeo termine saldrá el mensaje de alerta que hemos puesto en la función.



Otros eventos que se pueden capturar son:

- play: se ha iniciado.
- pause: se ha pausado.
- seeked: se ha cambiado la posición de la línea de tiempo

Como ejercicio el alumno debe poner un mensaje para cada uno de estos eventos.



A continuación, vamos a añadir efectos de vídeo:

Para ello vamos a poner nuevos controles en nuestro body:

```
<h4>VELOCIDAD VÍDEO</h4>
<button onclick="velocidad(0.5)">Lenta</button>
<button onclick="velocidad(1)">Normal</button>
<button onclick="velocidad(1.5)">Rápida</button>
<button onclick="velocidad(2)">SuperRápida</button>
<h4>EFECTOS VÍDEO</h4>
<select id="filter">
  <option value="none">Ninguno</option>
  <option value="grayscale(100%)">Escala de grises</option>
  <option value="sepia(100%)">Sepia</option>
  <option value="invert(100%)">Invertido</option>
  <option value="blur(5px)">Desenfoque</option>
</select>
</div>
```

Ahora vamos a crear un listener de javascript para escuchar cualquier cambio en este campo lista:

En primer lugar guardamos en una variable la referencia al campo lista:

```
<script>

  let vid = document.getElementById("video1");
  const filterSelect = document.getElementById("filter");

  vid.addEventListener("ended", function () {
    alert("El video ha terminado!");
  });
```



Y ahora ponemos el listener:

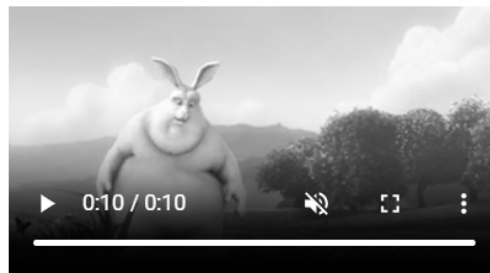
```
function velocidad(velocidad) {  
  vid.playbackRate = velocidad;  
}  
  
// Evento cuando se cambia de opción en la lista de filtros:  
filterSelect.addEventListener("change", () => {  
  vid.style.filter = filterSelect.value;  
});  
  
myFunction();  
  
</script>
```

Cuando se produzca un cambio en la lista se ejecuta esta función y dentro de ella lo que hacemos es cambiar el estilo de filtro con el valor que hemos puesto en las diferentes opciones del campo lista.

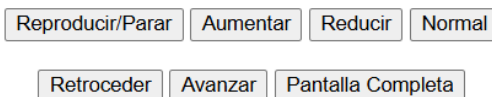
La salida por pantalla será:



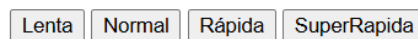
INCORPORACIÓN DE VÍDEO EN HTML



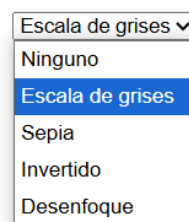
CONTROLES



VELOCIDAD VÍDEO



EFECTOS VÍDEO



POR ÚLTIMO, VAMOS A VER CÓMO PODEMOS CAPTURAR FOTOGRAFÍAS DEL VÍDEO EN UN CANVAS:



Captura de fotogramas de vídeo en canvas

<video>

Vamos a añadir el siguiente código a nuestra página html para añadir un canvas y el control de captura:

```
<option value="invert(100%)">Invertido</option>
<option value="blur(5px)">Desenfoque</option>
</select>
<br>
<h4>CAPTURA DE FOTOGRAMAS</h4>
<!-- CAPTURA DE FOTOGRAMAS DE VIDEO EN CANVAS -->
<button id="capture">Capturar Imagen</button>
<br>
<canvas id="canvas" width="640" height="360" style="border: 1px solid black"></canvas>
</div>

<script>
```

Se ha añadido el canvas y el botón de captura.

Ahora tenemos que poner el código javascript que recoja el evento click del botón de captura y también recoger la referencia del canvas para poder obtener un fotograma del vídeo:



```
// Captura de fotogramas:  
// Se recoge el canvas:  
const canvas = document.getElementById("canvas");  
// Se crea el contexto:  
const ctx = canvas.getContext("2d");  
// Se recoge la referencia al botón de captura:  
const captureButton = document.getElementById("capture");  
  
// Se crea un listener para escuchar el click del botón:  
captureButton.addEventListener("click", () => {  
  //Capturamos el fotograma del video:  
  ctx.drawImage(vid, 0, 0, canvas.width, canvas.height);  
});  
  
myFunction();
```

El método que realiza la captura es: **“drawImage”**. La captura se recoge con las coordenadas de un rectángulo, en este caso se ha recogido la imagen completa de la imagen del canvas.