

Controlando el ERROR en los elementos de audio y vídeo:

Vamos a crear una función que captará el error en los elementos de audio o vídeo (evento **onerror**), y en función de cuál sea el motivo del mismo, mostrará una alerta u otra. Tendremos que indicar, que cuando suceda dicho evento se ejecute la función que vamos a crear:

```
34 <body>
35 <h1>Video en HTML5</h1>
36 <video width="435" height="435" src="Video.ogv" controls onerror="fail(event)">
37 </video>
38 </body>
```

```
34 <body>
35 <h1>Audio en HTML5</h1>
36 <audio controls src="Audio.ogg" controls onerror="fail(event)">
37 </audio>
38 </body>
```

En los dos elementos definimos que la función que se debe ejecutar cuando se active el evento **onerror** sea la función *fail()*, esta función es idéntica en ambos casos:

```
5 <script>
6 function fail(e)
7 {
8     switch (e.target.error.code)
9     {
10
11         case e.target.error.MEDIA_ERR_ABORTED:
12             alert('Has abortado la reproduccion');
13             break;
14
15         case e.target.error.MEDIA_ERR_NETWORK:
16             alert('Error de red');
17             break;
18
19         case e.target.error.MEDIA_ERR_DECODE:
20             alert('Problema de corrupcion');
21             break;
22
23         case e.target.error.MEDIA_ERR_SRC_NOT_SUPPORTED:
24             alert('Formato no soportado o archivo no encontrado');
25             break;
26         default:
27             alert('Ocurrio un problema desconocido');
28             break;
29     }
30 }
31 </script>
```



Anotación

En el momento que utilizamos las etiquetas de <VIDEO> o <AUDIO> para reproducir elementos multimedia, los navegadores, si son capaces de reproducir dicho elemento, lo reproducen, y sino muestran el elemento multimedia pero sin posibilidad de reproducción. Si estamos controlando el error, nos dirán porqué no se reproduce. Pero, **¿qué pasa con los navegadores más antiguos que no soportan estas etiquetas?**

Tenemos la opción de poner un texto que aparecerá en los casos en que el navegador no sea capaz de interpretar las etiquetas:

```
6 <body>
7 <h1>Video en HTML5</h1>
8 <video width="300" height="300" controls>
9   <source src="Video.ogv" type="video/ogg"></source>
10  <source src="Video.mp4" type="video/mp4"></source>
11  <source src="Video.webm" type="video/webm"></source>
12  Tu navegador no soporta la etiqueta de VIDEO
13 </video>
14 </body>
```

Y también, tenemos la opción de colocar un elemento *Flash* para que en caso de que no soporte estas etiquetas de HTML5, por lo menos reproduzca el archivo mediante este formato:

```
7 <body>
8 <video width="435" height="435" controls>
9   <source src="Video.ogv" type="video/ogg"></source>
10  <source src="Video.mp4" type="video/mp4"></source>
11  <source src="Video.webm" type="video/webm"></source>
12
13  <object type="application/x-shockwave-flash"
14    width="435" height="435" data="player.swf?Video.mp4">
15
16    <param name="movie" value="player.swf?Video.mp4">
17    <a href="Video.mp4">Descarga el video</a>
18  </object>
19 </video>
20 </body>
```

Por lo tanto, **si el navegador no puede interpretar las etiquetas**, en el primer caso mostrará ese texto alternativo, y en el segundo caso, lo intentará utilizando un plugin de *Flash* y si tampoco soporta *Flash*, mostrará un enlace para poder descargar el archivo.

Aunque en el ejemplo aparezca con la etiqueta <VIDEO>, en el caso de un elemento de <AUDIO> funcionará de la misma manera.