

# Styling media players with CSS3

The `<video>` and `<audio>` elements are HTML elements like any other, so CSS3 can be used for styling, including CSS3 transitions, animations and transforms. This was not possible with the Flash technology.

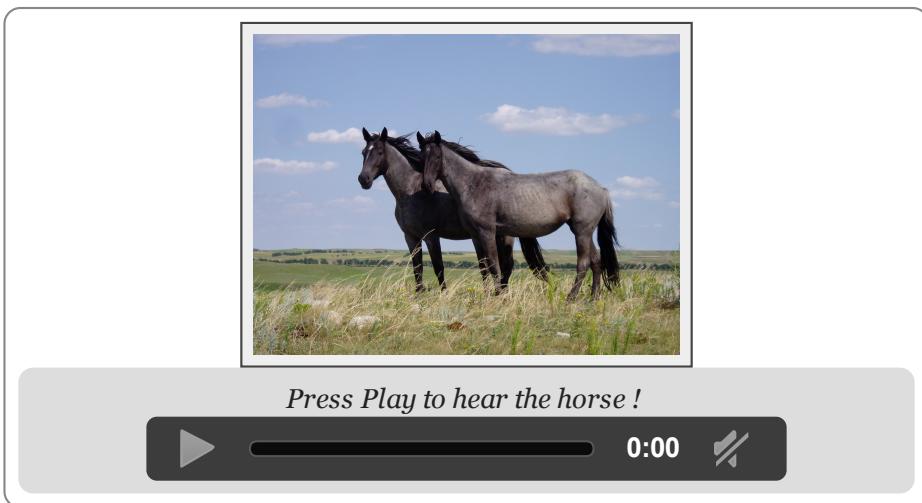
## AN EXAMPLE OF AN AUDIO PLAYER WITH SOME STYLE

### Add some decoration and styling

You can try this example [online at JS Bin](#).

To add some styling to the basic example we saw when we introduced the `<audio>` element, we just add a `<figure>` with two children: an `<img>` and a `<figcaption>`. Inside the `<figcaption>` we add the `<audio>` element from the previous example.

MOVE THE MOUSE POINTER OVER THIS PLAYER'S ELEMENTS!



HTML source code:

```
<figure id="figaudio1">
```

```
  
<figcaption id="figcaptionaudio1"> Press Play to hear the horse !  
<audio controls="controls">  
 <source src="https://dl.dropboxusercontent.com/u/1631516/horse.ogg"  
 type="audio/ogg" />  
 <source src="https://dl.dropboxusercontent.com/u/1631516/horse.mp3"  
 type="audio/mp3" />
```

Your browser does not support the audio element.

12. Download the audio/video in

```
<a href="https://dl.dropboxusercontent.com/u/1631516/horse.ogg">OGG</a>  
or <a href="https://dl.dropboxusercontent.com/u/1631516/horse.mp3">MP3</a>  
format.  
</audio>  
</figcaption>  
</figure>
```

CSS source code:

```
#figaudio1 {  
width : 420px;;  
text-align:center;  
padding : 6px;  
background : white;  
margin : 0 11px 0px 0;  
border :solid 1px #888888;  
border-radius : 8px ;  
}  
10.
```

```
#figcaptionaudio1 {  
font-size : .8em;
```

```

padding : 6px 8px;
background : #dddddd;
display :block;
text-align :center;
font-family : georgia, serif;
font-style : italic;
border-radius : 7px ;
20. }

#figaudio1 > img {
background : #eeeeee;
padding : 5px;
border : solid 1px #444444;
}

/* For audio and img transitions/animation */
audio, #figaudio1 > img {
30.   transition:all 0.5s;
}

#figaudio1 > img:hover {
  box-shadow: 15px 15px 20px rgba(0,0, 0, 0.4);
  transform: scale(1.05);
}

audio:hover, audio:focus, audio:active {
  box-shadow: 15px 15px 20px rgba(0,0, 0, 0.4);
40.   transform: scale(1.05);
}

```

## CHANGING THE SIZE OF A VIDEO ON THE FLY USING CSS3 TRANSFORMS

**Resizing and rotating a video as the mouse pointer comes over it**

See this [example online](#) (where you can modify the code on the fly) or just play the

following video, and move the mouse pointer in and out of the video while it's playing.



This example uses the pseudo CSS class `:hover` in order to track the `mouseover` event. On `mouseover`, it uses a CSS3 transition property that interpolates the changes in the scale and orientation of the video element (done using a `transform` CSS property).

The corresponding HTML source code is:

```
<video id="w3devCampusVideo" autoplay controls>
  <source src=http://html5doctor.com/demos/video-canvas-
    magic/video.webm
    type=video/webm>
  <source src=http://html5doctor.com/demos/video-canvas-magic/video.ogg
    type=video/ogg>
  <source src=http://html5doctor.com/demos/video-canvas-magic/video.mp4
    type=video/mp4>
</video>
```

... and the CSS source code is as follows:

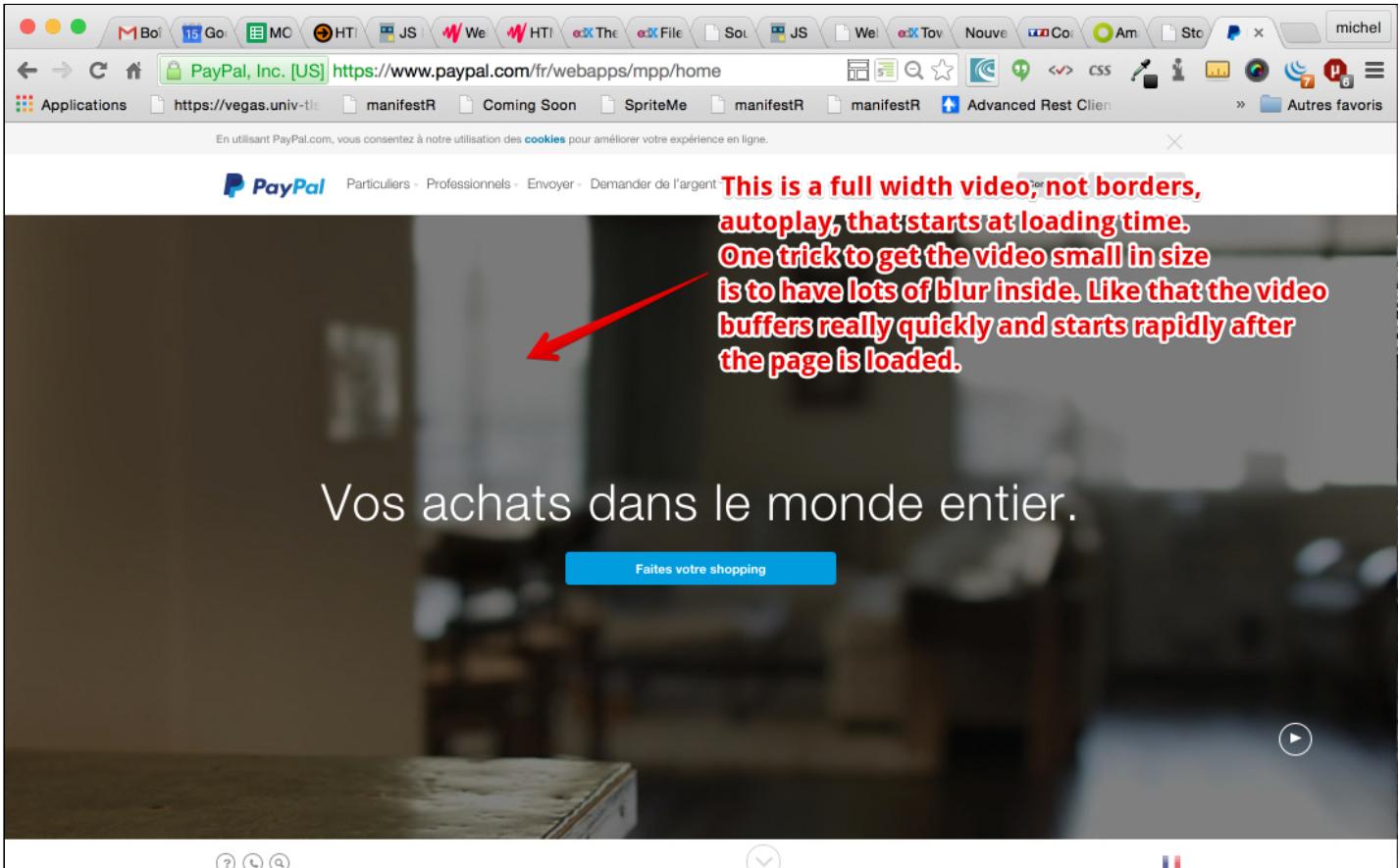
```
#w3devCampusVideo {
  width: 300px;
  transition: all 0.5s ease-in-out;
}

#w3devCampusVideo:hover {
  width:400px;
```

```
transform:rotate(-5deg);  
}
```

## Using CSS properties to adapt the video size to the size of the window without borders

This is a trendy way of displaying videos (look at the [PayPal Web site](https://www.paypal.com/fr/webapps/mpp/home) as an example).

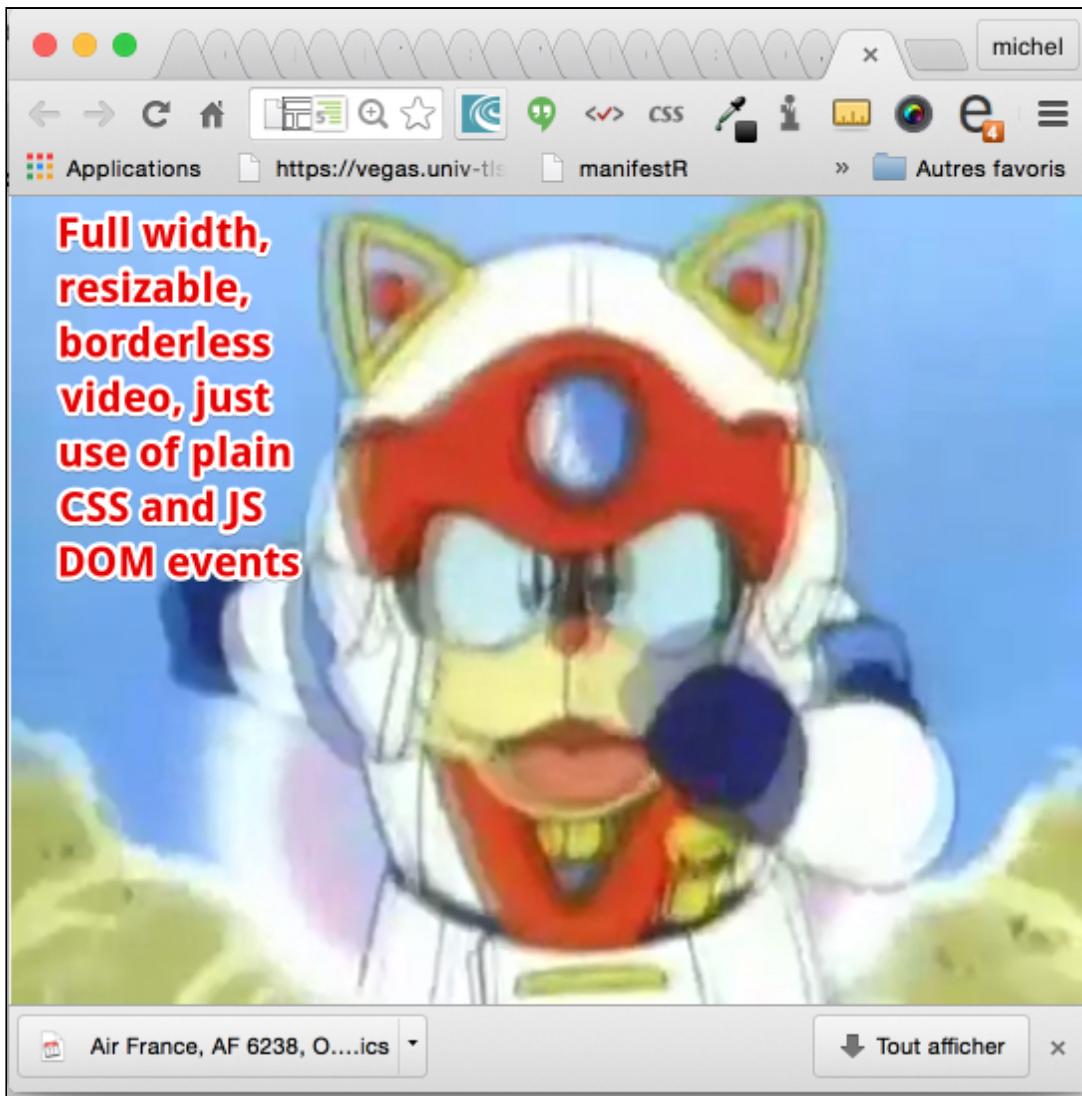


You will find below two examples that show how to do this trick. The first one is for a "regular" video, using the `<video>` and `<source>` elements. This technique can also be used on any YouTube embedded videos (see Example 2 below).

The interesting part is that we use a 100% standard (and really small and simple) JavaScript code here to handle the window `resize` events and we just set regular CSS properties `width` and `height` of the video element, to resize the video.

### Example 1: with a regular video

- Online at JS Bin



Here is the HTML code. It's really simple, just notice the `<body onload="init() ; ">` which calls the JavaScript `init()` function right after the page is loaded.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Full width video like PayPal site</title>
</head>
<body onload="init();">
  <video id="myVideo" autoplay>

```

```
<source
  src=http://html5doctor.com/demos/video-canvas-magic/video.webm
  type=video/webm>
12. <source
13.   src=http://html5doctor.com/demos/video-canvas-magic/video.ogg
14.   type=video/ogg>
<source
  src=http://html5doctor.com/demos/video-canvas-magic/video.mp4
  type=video/mp4>
</video>
</body>
```

Here is the CSS (remove margins, remove padding, hide parts that could overflow from the `<body>`):

```
body {
  margin:0;
  padding:0;
  overflow:hidden;
}
```

And now the JavaScript code:

```
var video;

function init() {
  // function called when the page is loaded
  video = document.querySelector("#myVideo");
  // For initial value
  video.width = window.innerWidth;
10.  video.height = window.innerHeight;
  // For dealing with window resize
  window.onresize = function() {
    video.width = window.innerWidth;
```

```
    video.height = window.innerHeight;  
};  
}  
}
```

## Example 2: with a YouTube video

- [Online at JS Bin](#)

The CSS and JavaScript codes for this example are exactly the same as in Example 1.

