

# Crear Reproductor De Audio Personalizado Con MediaElement.js

Existen una gran cantidad de **proyectos que pueden ser personalizados con HTML5**. Los nuevos elementos de HTML5 permiten simplificar las cosas, que en el pasado era necesario agregar con CSS para hacer que funcionara. Dos de los buenos ejemplos son los **elementos de audio y vídeo**. Aunque el elemento de audio y vídeo tienen un aspecto genérico, se puede personalizar usando un **plugin de jQuery llamado MediaElement.js**.

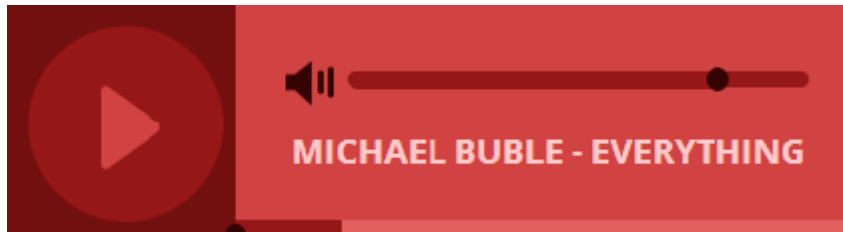
En este paso a paso, te mostraremos como **desarrollar un reproductor de audio** utilizando los kits de herramientas de interfaz de usuario realizadas por Rudolph Musngi, usando el **estilo mejorado con CSS3** y las funcionalidades que brinda [MediaElement.js](#).

## Recursos que necesitas

- Herramientas UI de Rudolph Musngi
- Librería jQuery Biblioteca
- MediaElement.js
- Algo de tiempo y paciencia

# Reproductor de audio personalizado

Al final, tendras el siguiente reproductor de audio:



## Paso 1 – Descargar biblioteca y MediaElement.js jQuery

Para hacer que todo funcione correctamente, es necesario descargar, primero, la [Biblioteca jQuery](#) o utilizar la [librería alojada en Google](#). También es necesario el **script MediaElement.js** y aojarlo en su lugar preferido. Dentro de la carpeta de MediaElement.js extraído, necesitaras los siguientes archivos:

- MediaElement-y-player.min.js
- flashmediaelement.swf
- silverlightmediaelement.xap

## Paso 2 – Agregar los vínculos necesarios de los archivos

Para empezar, es necesario agregar los siguientes enlaces al head del HTML. Esto incluye los enlaces a los archivos CSS,

librería jQuery y MediaElement.js.

```
[html]
```

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<meta charset="utf-8">
<title>Creating a Custom HTML5 Audio and Video Element UI with
MediaElement.js</title>
<link href="css/normalize.css" rel="stylesheet"
type="text/css">
<link href="css/style.css" rel="stylesheet" type="text/css">
<link href="css/audio.css" rel="stylesheet" type="text/css">
<script src="js/jquery-1.11.1.min.js"
type="text/javascript"></script>
<script src="js/mediaelement-and-player.min.js"
type="text/javascript"></script>
</head>
```

```
[/html]
```

## Paso 3 – EL HTML

Ahora, es necesario crear el div que contendrá el elemento de audio. Vamos a darle una clase y dentro de él, poner un elemento de párrafo para el título y el propio elemento de audio.

```
[html]
```

```
<div class="audio-player">
<p>Michael Buble – Everything</p>
<audio id="audio-player" src="restheces/Michael Buble –
Everything.mp3" type="audio/mp3" controls="controls"></audio>
</div>
```

```
[/html]
```

# Paso 4 – El Script

Ahora que ya había establecido el código HTML, es necesario agregar el código de abajo antes de la final de la etiqueta.

Observe que ha seleccionado la clase `audio-player` para envolver los elementos de audio. Estos son los parámetros básicos establecidos; puedes revisar la [documentación mediaelement.js](#) para aprender más sobre las opciones disponibles.

```
[js]
$(document).ready(function() {
$('#audio-player').mediaelementplayer({
alwaysShowControls: true,
features: ['playpause', 'volume', 'progress'],
audioVolume: 'horizontal',
audioWidth: 364,
audioHeight: 70
});
});

[/js]
```

# Paso 5 – El CSS

A continuación, configure los estilos del reproductor de audio personalizado. En primer lugar, añadir algunos estilos de reinicio para asegurarse de que no hay otros estilos extra en cada navegador.

```
[css]
.audio-player, .audio-player div, .audio-player p, .audio-player
a, .audio-player button {
margin: 0;
```

```
padding: 0;
border: none;
outline: none;
}[/css]
```

Entonces, el estilo del contenedor jugador y agregar el ancho de altura sobre el mismo y el estilo de la etiqueta de párrafo por el título.

```
[css]
```

```
.audio-player {
position: relative;
margin: 0 auto;
background: #d14242;
width: 426px;
height: 120px;
}
```

```
.audio-player p {
position: absolute;
text-align: center;
text-transform: uppercase;
left: 143px;
top: 62px;
font-family: 'Open Sans', sans-serif;
font-weight: 700;
font-size: 17px;
color: #fcc;
font-weight: 700;
}
```

```
[/css]
```

Luego, este será el estilo en el botón play-pausa. Esto será una forma de la imagen de sprites png y manipulado usando el background-position.

```
[css]
```

```
.mejs-controls .mejs-button button {  
position: absolute;  
display: block;  
width: 73px;  
height: 71px;  
cursor: pointer;  
background: #961717;  
}
```

```
.mejs-controls .mejs-play button,.mejs-controls .mejs-pause  
button {  
width: 115px;  
height: 120px;  
background: #961717 url(../img/playerbuttons.png) 0 0 no-  
repeat;  
}
```

```
.mejs-controls .mejs-pause button {  
background-position: 0 -120px;  
}
```

[/css]

Para el botón Mute y Activar, utilice una imagen de sprites png y manipular el background-position para mostrar cada parte.

[css]

```
.mejs-controls .mejs-mute button,.mejs-controls .mejs-unmute  
button {  
width: 24px;  
height: 23px;  
background: transparent url(../img/mute-unmute.png) 0 0;  
top: 28px;  
left: 140px;  
}
```

```
.mejs-controls .mejs-unmute button {
```

```
background-position: 0 -23px;
}
```

```
[/css]
```

Ahora, agregue los siguientes estilos para el control deslizante de volumen. Ten en cuenta que en lugar de utilizar una imagen para el icono de círculo, CSS3 se utilizó en lugar de utilizar la propiedad border-radius.

```
[css]
```

```
.mejs-controls div.mejs-horizontal-volume-slider {
position: absolute;
top: 33px;
right: 23px;
cursor: pointer;
}
```

```
.mejs-controls .mejs-horizontal-volume-slider .mejs-
horizontal-volume-total {
width: 232px;
height: 8px;
background: #961717;
-webkit-border-radius: 4px;
-moz-border-radius: 4px;
border-radius: 4px;
}
```

```
.mejs-controls .mejs-horizontal-volume-slider .mejs-
horizontal-volume-current {
position: absolute;
width: 0;
height: 8px;
background: #961717;
top: 1px;
left: 1px;
-webkit-border-radius: 8px;
-moz-border-radius: 8px;
```

```
border-radius: 8px;
}
```

```
[/css]
```

Los códigos CSS a continuación configuran el color de fondo y dimensión del tiempo cargado y la hora actuales.

```
[css]
```

```
.mejs-controls div.mejs-time-rail {
width: 380px;
}
```

```
.mejs-controls .mejs-time-rail span {
position: absolute;
display: block;
width: 380px;
height: 12px;
top: 40px;
left: 55px;
cursor: pointer;
-webkit-border-radius: 0 0 2px 2px;
-moz-border-radius: 0 0 2px 2px;
border-radius: 0 0 2px 2px;
}
```

```
.mejs-controls .mejs-time-rail .mejs-time-total {
background: #e26060;
width: 311px !important;
height: 12px;
margin-top: 68px;
margin-left: 60px;
}
```

```
.mejs-controls .mejs-time-rail .mejs-time-loaded {
top: 0;
left: 0;
width: 0;
```



```
background: #961717;
}
```

```
.mejs-controls .mejs-time-rail .mejs-time-current {
top: 0;
left: 0;
width: 0;
background: #961717;
}
```

```
[/css]
```

Luego, se sigue personalizando el control del volumen.

```
[css]
```

```
.mejs-controls .mejs-time-rail .mejs-time-handle {
position: absolute;
display: block;
width: 10px;
height: 10px;
top: 2px;
background: #330202;
-webkit-border-radius: 50%;
-moz-border-radius: 50%;
border-radius: 50%;
left: 11px;
}
```

```
.mejs-controls .mejs-horizontal-volume-slider .mejs-
horizontal-volume-handle {
position: absolute;
display: block;
width: 12px;
height: 12px;
-webkit-border-radius: 50%;
top: -2px;
background: #330202;
}
```

```
[/css]
```

Por último, añadir los estilos para la descripción Tiempo Barra de progreso. Esto aparecerá cuando se pasa el puntero sobre la barra de progreso.

```
[css]
```

```
.mejs-controls .mejs-time-rail .mejs-time-float {  
position: absolute;  
z-index: 9999;  
color: fff;  
width: 33px;  
height: 23px;  
top: -26px;  
display: none;  
margin-left: -17px;  
}
```

```
.mejs-controls .mejs-time-rail .mejs-time-float-current {  
width: 33px;  
display: block;  
left: 0;  
top: 4px;  
font-family: 'Open Sans', sans-serif;  
font-size: 10px;  
font-weight: 700;  
color: #f89494;  
text-align: center;  
z-index: 9999;  
background: #bf3737;  
padding: 2px 2px 5px;  
}
```

```
.mejs-controls .mejs-time-rail .mejs-time-float-current:after  
{  
content: "";  
position: absolute;
```

```
width: 0;
height: 0;
border-width: 10px;
border-style: solid;
border-color: #bf3737 transparent transparent;
top: 15px;
left: 8px;
}
```

```
[/css]
```

Y eso es todo! Como puedes ver, es mas trabajo de CSS que de otra cosa.