

# Usando a Tag de Áudio e Vídeo em HTML5

## Aplicações práticas com HTML5

Com a chegada do **HTML5 temos muitas novidades, umas conceituais outras mais visíveis**, entre estas podemos destacar a criação de três novas tags, `<video>`, `<audio>` e `<canvas>`. As duas primeiras serão abordadas aqui nesse artigo, saiba mais sobre a tag [`<canvas>`](#).

### Mais sobre as Tags `<audio>` e `<video>` no HTML

- [A tag `<audio>`](#)
- [TAG `<video>`](#)
- [Incorporando `<audio>` e `<video>` no HTML](#)

Para quem ainda não sabe, as tag's `<audio>` e `<video>` são tag's de media que foram criadas, entre outros motivos, para acabar com a dependência de plugin's para a reprodução de áudio e vídeo nas páginas. Dessa forma em sites que precisaríamos ter instalado o Flash, por exemplo o Youtube, não iremos mais, basta que sejam implementadas essas novas tag's e que tenhamos um navegador compatível com HTML5, o que não é incomum hoje em dia tendo em vista que todos os navegadores mais recente são. O IE8 foge dessa lista apenas o IE9 suporta o HTML5.

Como já sabemos, a simplicidade é uma das características do HTML5, para estas tag's isso não é diferente, pois agora passaremos a manipular áudio e vídeo praticamente como manipulamos a inserção de imagens.

Sabendo disso vejamos a seguir algumas características e demonstrações de ambas, de maneira a abordar o básico porém funcional. Vamos começar com a tag `<audio>`.

Como o nome já sugere esta tag serve para reproduzirmos uma mídia de áudio, vejamos:

### Listagem 1: Demonstração básica do uso da tag `<audio>`

```
<audio autoplay="autoplay" controls="controls">  
<source src="sua_musica.mp3" type="audio/mp3" />  
seu navegador não suporta HTML5  
</audio>
```

# Usando a Tag de Áudio e Vídeo em HTML5

Na Listagem 1 temos um exemplo básico do uso da tag `<audio>`, nela estamos setando `autoplay` e `controls`, que servem para tocar automaticamente e habilitar os controles do player, respectivamente. Será reproduzida a mídia `sua_musica.mp3` definida na tag `<source>` dentro da tag `<audio>` pela propriedade `"src"`. Veja a Imagem 1 como ficaria no navegador:



**Figura 1:** Player criado pela tag `<audio>`

## **Caso seu navegador não suporte o HTML5 você verá o seguinte: “seu navegador não suporta HTML5”**

Como é por natureza um componente HTML e não um plugin, podemos fazer algumas modificações no player via CSS, como por exemplo, cor, tamanho, float e position.

Pode ser feita também a customização por outros elementos HTML e por meio de Javascript podemos controlar propriedades como o `PLAY`, `PAUSE`, `VOLUME` e por CSS criamos nosso próprio player.

**Listagem 2:** Controle da tag áudio por Javascript.

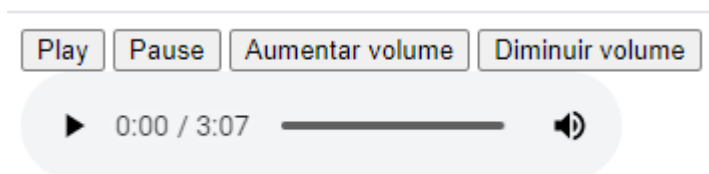
```
<!DOCTYPE HTML>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
  <title>HTML5 Vídeo</title>
</head>

<body>
  <div>
    <button onclick="document.getElementById('player').play()">Play</button>
    <button onclick="document.getElementById('player').pause()">Pause</button>
    <button onclick="document.getElementById('player').volume+=0.1">Aumentar
volume</button>
    <button onclick="document.getElementById('player').volume-=0.1">Diminuir vo-
lume</button>
  </div>

  <audio id="player" autoplay="autoplay" controls="controls">
    <source src="sua_musica.mp3" type="audio/mp3" />
    seu navegador não suporta HTML5
  </audio>
</body>
</html>
```

**O código acima resultaria mais ou menos como na Figura 2:**

# Usando a Tag de Áudio e Vídeo em HTML5



**Figura 2:** Controlando o áudio por Javascript

Este é apenas um exemplo básico, mas com um pouco conhecimento em javascript e CSS pode-se fazer um player muito bom. Dessa forma podemos desabilitar o atributo “controls”, para que tenhamos o controle apenas pelo nosso próprio player.

Os atributos da tag <audio>:

Atributo	Valor	Descrição
autoplay	autoplay	Define que o audio começará a ser tocado assim que ele estiver pronto.
controls	controls	Os controles serão mostrados.
loop	loop	Define que o audio começará a ser tocado novamente quando terminar.
preload	preload	Define que o audio será carregado enquanto a página é lida. Esse atributo é ignorado caso o atributo autoplay estiver definido.
src	url	URL do arquivo a ser tocado.

Formatos de áudio suportados:

Formato	IE 8+	Firefox 3.5+	Opera 10.5+	Chrome 3.0+	Safari 3.0+
Ogg Vorbis	Não	Sim	Sim	Sim	Não
MP3	Não	Não	Não	Sim	Sim
Wav	Não	Sim	Sim	Não	Sim

Agora veremos a TAG <video>.

## TAG <video>

Aqui seguimos o mesmo conceito da <audio>, inclusive sua sintaxe é muito parecida. Vejamos o exemplo da Listagem 3.

**Listagem 3:** A tag <video>

```
<!DOCTYPE HTML>
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type">
<title>HTML5 Vídeo</title>
</head>

<body>
<video width="320" height="240" controls="controls">
  <source src="filme.mp4" type="video/mp4">
```

## Usando a Tag de Áudio e Vídeo em HTML5

```
<source src="filme.ogg" type="video/ogg">
Seu navegador não suporta HTML5.
</video>
</body>
</html>
```

Vejam que praticamente não há nada diferente do que já vimos, exceto as propriedades `width` e `height` que não tínhamos definido na tag `<audio>`. Mas para o vídeo é uma boa prática e extremamente importante, pois se não especificarmos o tamanho que a tag deve ocupar, o navegador pode se perder e causar um desastre no restante do layout.

Os formatos de vídeos suportados atualmente são:

Navegador	MP4	WebM	Ogg
Internet Explorer 9	SIM	NÃO	NÃO
Firefox 4.0	NÃO	SIM	SIM
Google Chrome 6	SIM	SIM	SIM
Apple Safari 5	SIM	NÃO	NÃO
Opera 10.6	NÃO	SIM	SIM

Da mesma forma como na tag `<audio>`, a tag `<video>` também pode ser manipulada por Javascript.

Na Listagem 4 pode ser observado o trecho de código Javascript escrito para controlar um vídeo.

### Listagem 4: Controlando o vídeo por Javascript

```
<script>
function playPause(){
  if (video.paused)
    video.play();
  else
    video.pause();
}

function grande(){
  /*altera a largura da tag, automaticamente a altura é alterada proporcionalmente */
  video.width=560;
}
```

# Usando a Tag de Áudio e Vídeo em HTML5

```
function pequeno(){  
/*altera a largura da tag, automaticamente a altura é alterada porporcionalmente */  
video.width=320;  
}  
  
function normal(){  
/*altera a largura da tag, automaticamente a altura é alterada porporcionalmente */  
video.width=420;  
}  
</script>
```

Bem pessoal fico por aqui, espero que tenham gostado e principalmente ficado curiosos para saber mais desse assunto, dessa independência de plugins.

## Incorporando mídia:

Incorporar mídia em documentos HTML é simples:

```
<video src="http://v2v.cc/~j/theora_testsuite/320x240.ogg" controls>  
Seu navegador não suporta o elemento <code>video</code>.  
</video>
```

Esse exemplo reproduz uma amostra de vídeo, com controles de reprodução, a partir do site Theora.

Aqui há um exemplo de áudio incorporado em um documento HTML

```
<audio src="/test/audio.ogg">  
<p>Seu navegador não suporta o elemento audio.</p>  
</audio>
```

O atributo src pode ser a URL do arquivo de áudio ou o caminho do arquivo no sistema local.

```
<audio src="audio.ogg" controls autoplay loop>  
<p>Seu navegador não suporta o elemento audio </p>  
</audio>
```

Esse exemplo de código usa atributos do elemento <audio>:

- controls : Mostra os controles padrão para o áudio na página.
- autoplay : Faz com que o áudio reproduza automaticamente.
- loop : Faz com que o áudio repita automaticamente.

```
<audio src="audio.mp3" preload="auto" controls></audio>
```

O atributo preload é usado em elementos audio para carregar arquivos grandes. Ele pode assumir 3 valores:

- "none" não carrega o arquivo

## Usando a Tag de Áudio e Vídeo em HTML5

- "auto" carrega o arquivo
- "metadata" carrega apenas os meta dados do arquivo

Vários arquivos podem ser especificados utilizando o elemento `<source>` para disponibilizar vídeo ou áudio codificados em formatos diferentes para navegadores diferentes. Por exemplo:

```
<video controls>
  <source src="foo.ogg" type="video/ogg">
  <source src="foo.mp4" type="video/mp4">
  Seu navegador não suporta o elemento <code>video</code>.
</video>
```

Isso reproduz o arquivo Ogg em navegadores que suportam o formato Ogg. Se o navegador não suportar Ogg, o navegador reproduz o arquivo MPEG-4.

Também é possível especificar os codecs que o arquivo de mídia requer; isso permite que o navegador faça escolhas mais inteligentes:

```
<video controls>
  <source src="foo.ogg" type="video/ogg; codecs=dirac, speex">
  Seu navegador não suporta o elemento <code>video</code>.
</video>
```

Aqui é especificado que o vídeo usa os codecs Dirac e Speex. Se o navegador suportar Ogg, mas não suportar os codecs especificados, o vídeo não será reproduzido.

Se o atributo `type` não estiver especificado, o tipo de mídia é obtido no servidor e é verificado se o navegador consegue reproduzi-lo; se ele não pode ser renderizado, o próximo `source` é verificado. Se nenhum dos elementos `source` pode ser utilizado, um evento `error` é enviado para o elemento `video`. Se o atributo `type` estiver especificado, ele é comparado aos tipos que o navegador consegue reproduzir, e se ele não for reconhecido, o servidor não é verificado; ao invés disso, o próximo `source` é verificado.

Fonte: <https://www.devmedia.com.br/html5-as-tags-audio-e-video/26018>