

HTML5 HTML5视频与音频

十二、HTML5视频与音频

音频 - audio

关于音频的格式

ogg

mp3

wav

acc

使用

属性

[HTML5 Audio/Video 方法](#)

[HTML5 Audio/Video 属性](#)

[HTML5 Audio/Video 事件](#)

[利用AudioContext绘制](#)

十二、HTML5视频与音频

简单介绍

HTML5 对视频和音频特性规范文档的制定是被讨论最多的。除浏览器自带支持的明显好处外，评论点集中在浏览器提供商对音频/视频格式 的不同选择。如果你准备使用HTML5的

音频: ogg (ogg, oga), mp3, wav, AAC

视频: ogg (ogv), H.264 (mp4)

另外你还需要留意一下 Google 的 VP8 视频解码，这个将被作为一个开源格式来结束（格式选择的）纷争。HTML5 提供的一个解决方案是，让你可以指定多个不同格式的源文件，以便于用户浏览器选择它认识的文件。对于 < IE9 和旧浏览器，你将需要一个折衷的解决方案。

当你第一次尝试 HTML5 的音频/视频，你可能会想知道这些可能对你有帮助的东东：

你的服务器必须支持你打算提供的音频/视频的 MIME 格式。你将需要检查一下其在本地服务器上是否被支持。

如果你使用 Safari 来检测 HTML5 的音频/视频支持，需要安装 QuickTime Player，没有QuickTime Player 的 Safari 什么都不能播放。

音频 - audio

关于音频的格式

ogg

Ogg全称应该是OGGVobis(oggVorbis)是一种新的音频压缩格式，类似于MP3等的音乐格式。Ogg是完全免费、开放和没有专利限制的。OggVorbis文件的扩展名是.OGG。Ogg文件格式可以不断地进行大小和音质的改良，而不影响旧有的编码器或播放器。

mp3

MP3是一种音频压缩技术，其全称是 动态影像专家压缩标准音频层面3 (Moving Picture Experts Group Audio Layer III)，简称为MP3。它被设计用来大幅度地降低音频数据量。利用 MPEG Audio Layer 3 的技术，将音乐以1:10 甚至 1:12 的压缩率，压缩成容量较小的文件，而 对于大多数用户来说重放的音质与最初的不压缩音频相比没有明显的下降。它是在1991年由位于德国埃尔朗根的研究组织Fraunhofer-Gesellschaft 的一组工程师发明和标准化的。用MP3形式存储的音乐就叫做MP3音乐，能播放MP3音乐的机器就叫作MP3播放器。

wav

WAV为微软公司(Microsoft)开发的一种声音文件格式，它符合RIFF(Resource Interchange File Format)文件规范，用于保存Windows平台的音频信息资源，被Windows平台及其应用程序所广泛支持，该格式也支持MSADPCM，CCITT A LAW等多种压缩运算法，支持多种音频数字，取样频率和声道，标准格式化的WAV文件和CD格式一样，也是44.1K的取样频率，16位量化数字，因此在声音文件质量和CD相差无几！WAV打开工具是WINDOWS 的媒体播放器。

acc

AAC (Advanced Audio Coding)，中文称为“高级音频编码”，出现于1997年，基于 MPEG-2的音频编码技术。由Fraunhofer IIS、杜比实验室、AT&T、Sony (索尼) 等公司共同开发，目的是取代MP3格式。2000年，MPEG-4标准出现后，AAC 重新集成了其特性，加入了SBR技术和PS技术，为了区别于传统的 MPEG-2 AAC 又称为 MPEG-4 AAC。

使用

1. `<audio controls autoplay loop src="audio/after the afterparty.mp3">`
2. `<!-- <source src="audio/after the afterparty.mp3" type=""> -->`
3. `</audio>`

属性

HTML5 Audio/Video 方法

`addTextTrack()`：向音频/视频添加新的文本轨道
`canPlayType()`：检测浏览器是否能播放指定的音频/视频类型
`load()`：重新加载音频/视频元素
`play()`：开始播放音频/视频
`pause()`：暂停当前播放的音频/视频

HTML5 Audio/Video 属性

`audioTracks`：返回表示可用音轨的 `AudioTrackList` 对象
`autoplay`：设置或返回是否在加载完成后随即播放音频/视频
`buffered`：返回表示音频/视频已缓冲部分的 `TimeRanges` 对象
`controller`：返回表示音频/视频当前媒体控制器的 `MediaController` 对象
`controls`：设置或返回音频/视频是否显示控件（比如播放/暂停等）
`crossOrigin`：设置或返回音频/视频的 `CORS` 设置
`currentSrc`：返回当前音频/视频的 `URL`
`currentTime`：设置或返回音频/视频中的当前播放位置（以秒计）
`defaultMuted`：设置或返回音频/视频默认是否静音
`defaultPlaybackRate`：设置或返回音频/视频的默认播放速度
`duration`：返回当前音频/视频的长度（以秒计）
`ended`：返回音频/视频的播放是否已结束
`error`：返回表示音频/视频错误状态的 `MediaError` 对象
`loop`：设置或返回音频/视频是否应在结束时重新播放
`mediaGroup`：设置或返回音频/视频所属的组合（用于连接多个音频/视频元素）
`muted`：设置或返回音频/视频是否静音
`networkState`：返回音频/视频的当前网络状态
`paused`：设置或返回音频/视频是否暂停
`playbackRate`：设置或返回音频/视频播放的速度
`played`：返回表示音频/视频已播放部分的 `TimeRanges` 对象
`preload`：设置或返回音频/视频是否应该在页面加载后进行加载
`readyState`：返回音频/视频当前的就绪状态
`seekable`：返回表示音频/视频可寻址部分的 `TimeRanges` 对象
`seeking`：返回用户是否正在音频/视频中进行查找
`src`：设置或返回音频/视频元素的当前来源
`startDate`：返回表示当前时间偏移的 `Date` 对象
`textTracks`：返回表示可用文本轨道的 `TextTrackList` 对象
`videoTracks`：返回表示可用视频轨道的 `VideoTrackList` 对象
`volume`：设置或返回音频/视频的音量`

HTML5 Audio/Video 事件

abort：当音频/视频的加载已放弃时
canplay：当浏览器可以播放音频/视频时
canplaythrough：当浏览器可在不因缓冲而停顿的情况下进行播放时
durationchange：当音频/视频的时长已更改时
emptied：当目前的播放列表为空时
ended：当目前的播放列表已结束时
error：当在音频/视频加载期间发生错误时
loadeddata：当浏览器已加载音频/视频的当前帧时
loadedmetadata：当浏览器已加载音频/视频的元数据时
loadstart：当浏览器开始查找音频/视频时
pause：当音频/视频已暂停时
play：当音频/视频已开始或不再暂停时
playing：当音频/视频在已因缓冲而暂停或停止后已就绪时
progress：当浏览器正在下载音频/视频时
ratechange：当音频/视频的播放速度已更改时
seeked：当用户已移动/跳跃到音频/视频中的新位置时
seeking：当用户开始移动/跳跃到音频/视频中的新位置时
stalled：当浏览器尝试获取媒体数据，但数据不可用时
suspend：当浏览器刻意不获取媒体数据时
timeupdate：当目前的播放位置已更改时
volumechange：当音量已更改时
waiting：当视频由于需要缓冲下一帧而停止

注意： 为了兼容性。一般使用source标签加载多个音频

利用AudioContext绘制