HTML5

音频 视频

目录

1 用法 3

2 重要HTML属性 4

3 source标签 5

4 常用事件 5

5 常用方法 6

6 常用API属性 6

7 自定义控制器 7

# 1 用法

<video src="audio/1.mp3"></audio>

<video src="./video/mv.mp4"></video>

注意：audio和video元素必须同时包含开始和结束标签，不能使用

不幸的是，和 HTML5 音频一样，涉及到视频的文件格式，Firefox 和 Safari/Chrome 的支持方式并不相同。因此，如果你想在这个时候使用 HTML5 视频，则需要创建三个视频版本。

.OGG

Firefox 能良好支持这种格式。你可以使用 VLC (媒体 -> 串流/保存) 实现视频的轻松转换。

.MP4

许多屏幕录制工具支持 MP4 格式的自动导出，你可以使用它们为 Safari 和 Chrome 浏览器生成指定格式的视频。

.FLV/.SWF

并非所有浏览器都支持 HTML5 视频，当然，考虑到兼容性，请确保添加一个退而求其次的 Flash 版本。

Video

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **格式** | **IE** | **Firefox** | **Opera** | **Chrome** | **Safari** |
| Ogg | No | 3.5+ | 10.5+ | 5.0+ | No |
| MPEG 4 | No | No | No | 5.0+ | 3.0+ |
| WebM | No | No | 10.6+ | 6.0+ | No |

Audio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **IE 8** | **Firefox 3.5** | **Opera 10.5** | **Chrome 3.0** | **Safari 3.0** |
| Ogg Vorbis |  | √ | √ | √ |  |
| MP3 |  |  |  | √ | √ |
| Wav |  | √ | √ |  | √ |

# 2 重要HTML属性

(1)width: 视频所需的宽度。默认情况下，浏览器会自动检测所提供的视频尺寸；  
(2)height: 视频所需高度；

　　(3)controls: 如果出现该属性，则向用户显示控件，比如播放按钮。每个浏览器中的播放控件都不太一样，但用途都一样，都可以控制开始和结束，跳到新位置和调节音量

(4)autoplay: 如果出现该属性，则视频在就绪后马上播放。如果不设置autoplay属性，必须是用户单击播放按钮才会播放音频文件。

(5)src: 用于指定[视频](http://www.wangqi.com/html/2010-11/c69.aspx)的地址。

(6)poster: 用于指定一张图片，在当前视频数据无效时显示(预览图)。视频数据无效可能是视频正在加载，可能是视频地址错误等等。

　　(7)loop: (循环播放)告诉浏览器在音频到达末尾时，再从头开始重新播放

　　(8)preload: auto、mete、none：告诉浏览器如何下载音频

auto：让浏览器下载整个文件，以便用户单击播放按钮时就能播放。当然，下载过程是后台进行的，网页访客不必等待下载完成，而且仍然可以随意查看网页。

meta：告诉浏览器先获取音频文件开头的数据块，从而足以确定一些基本信息(比如音频的总时长)

none：告诉浏览器不必预先下载。恰当地利用这些值，可以节省带宽。

如果没有设置preload属性，浏览器就自己决定是否预先下载了。对这一点，不同浏览器的处理方式也不一样。多数浏览器将auto作为默认值，但Firefox的默认值是metadata。不过，也请大家注意，这个preload属性也不是必须严格执行的规则，而只是你对浏览器的建议。根据具体情况，浏览器可以忽略你的设置。(有些旧版本浏览器根据不会在意preload属性。)

# 3 source标签

　　Source标签用于给媒体(因为audio标签同样可以包含此标签，所以这儿用媒体，而不是视频)指定多个可选择的文件地址，且只能在媒体标签没有使用src属性时使用。浏览器按source标签的顺序检测标签指定的视频是否能够播放(可能是视频格式不支持，视频不存在等等)，如果不能播放，换下一个。此方法多用于兼容不同的浏览器。Source标签本身不代表任何含义，不能单独出现。此标签包含src、type、media三个属性。

　　src属性：用于指定媒体的地址，和video标签的一样。

　　type属性：用于说明src属性指定媒体的类型，帮助浏览器在获取媒体前判断是否支持此类别的媒体格式。

　　media属性：用于说明媒体在何种媒介中使用，不设置时默认值为all，表示支持所有媒介。

# 4 常用事件

　　事件名称 ： 解释

oncanplay 当媒体可以播放的时候执行

oncanplaythrough 当媒体可以播放到最后的时候执行

ondurationchange 当媒体长度改变的时候执行

onemptied 当媒体资源元素变成空的时候执行

onended 当媒体已经到达最后的时候执行

onerror 当加载元素发生错误的时候执行

onloadeddata 当媒体数据加载完毕的时候执行

onloadedmetadata 当媒体元素的持续时间加载完毕的时候执行

onloadstart 当浏览器开始加载媒体数据的时候执行

onpause 当媒体数据暂停的时候执行

onplay 当媒体数据继续开始播放的时候执行

onplaying 当媒体数据已经开始播放的时候执行

onprogress 在浏览器正在获取媒体数据的时候执行

onreadystatechange 当ready-state发生改变的时候执行

onstalled 当获取元素数据发生错误的时候执行

onsuspend 当浏览器停止获取媒体数据的时候执行

ontimeupdate 当媒体改变其播放位置的时候执行

onvolumechange 当媒体音量大小发生改变的时候执行，包括无声

onwaiting 当媒体元素停止播放的时候

# 5 常用方法

　　方法名称 ：解释

　　play()：开始播放音频/视频

　　pause()：暂停当前播放的音频/视频

# 6 常用API属性

　　属性名称 ： 解释

　　duration：返回当前音频/视频的长度(以秒计)

　　paused：设置或返回音频/视频是否暂停

currentTime：设置或返回音频/视频中的当前播放位置(以秒计)

volume: 设置或返回音频/视频中的音量

　　ended：返回音频/视频的播放是否已结束

# 7 自定义控制器

1. 根据js控制video api
2. 根据css3伪元素操作

video::-webkit-media-controls-play-button {-webkit-appearance:none;background-color: red;}