

## html5 标签

**<video>**:Html5 提供的播放视频的标签

**<video src="resource/video/test.mp4" autoplay controls loop muted ></video>**

src:资源地址

controls:该属性定义是显示还是隐藏用户控制界面

**<audio>**:Html5 提供的播放音频的标签

**<audio src="resource/audio/test.mp3" controls ></audio>**

src:资源地址

controls:该属性定义是显示还是隐藏用户控制界面

**<source>**

视频:

type='video/webm; codecs="vp8, vorbis"'

type='video/ogg; codecs="theora, vorbis"'

type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'

音频:

type='audio/ogg; codecs="vorbis"'

type='audio/aac; codecs="aac"'

type='audio/mpeg'

## ###video 标签的属性

**width** : 视频显示区域的宽度, 单位是 CSS 像素

**height** : 视频展示区域的高度, 单位是 CSS 像素

**poster** : 一个海报帧的 URL, 用于在用户播放或者跳帧之前展示

**src** : 要嵌到页面的视频的 URL

**controls** : 显示或隐藏用户控制界面

**autoplay** : 媒体是否自动播放

**loop** : 媒体是否循环播放

**muted** : 是否静音

**preload** : 该属性旨在告诉浏览器作者认为达到最佳的用户体验的方式是什么

**none** : 提示作者认为用户不需要查看该视频, 服务器也想要最小化访问流量;  
换句话说就是提示浏览器该视频不需要缓存。

**metadata** : 提示尽管作者认为用户不需要查看该视频,  
不过抓取元数据 (比如: 长度) 还是很合理的。

**auto**: 用户需要这个视频优先加载; 换句话说就是提示: 如果需要的话,  
可以下载整个视频, 即使用户并不一定会用它。  
空字符串: 也就代指 auto 值。

### ###audio 标签的属性

src            音频的位置  
controls    是否有控制台  
autoplay    是否页面加载了就播放  
loop  
muted  
preload

### ###音视频 js 相关属性

currentTime : 开始到播放现在所用的时间(可读写)  
duration    : 媒体总时间(只读)  
muted       : 是否静音(可读写,相比于 volume 优先级要高)  
volume      : 0.0-1.0 的音量相对值(可读写)  
paused      : 媒体是否暂停(只读)  
ended       : 媒体是否播放完毕(只读)  
error       : 媒体发生错误的时候,返回错误代码 (只读)  
currentSrc   : 以字符串的形式返回媒体地址(只读)

视频多的部分:

poster : 视频播放前的预览图片(读写)  
width、height : 设置视频的尺寸(读写)之前设置了才能读,不然就是 0,  
videoWidth、 videoHeight : 视频的实际尺寸(只读)

### ###音视频 js 相关函数

play() : 媒体播放  
pause() : 媒体暂停  
load() : 重新加载媒体

## ###js 相关事件

### 视频:

**abort** 在播放被终止时触发,例如, 当播放中的视频重新开始播放时会触发这个事件。

**canplay** 在媒体数据已经有足够的数据（至少播放数帧）可供播放时触发。这个事件对应 **CAN\_PLAY** 的 **readyState**。

**canplaythrough** 在媒体的 **readyState** 变为 **CAN\_PLAY\_THROUGH** 时触发, 表明媒体可以在保持当前的下载速度的情况下不被中断地播放完毕。注意: 手动设置 **currentTime** 会使得 **firefox** 触发一次

**canplaythrough** 事件, 其他浏览器或许不会如此。

**durationchange** 元信息已载入或已改变, 表明媒体的长度发生了改变。例如, 在媒体已被加载足够的长度从而得知总长度时会触发这个事件。

**emptied** 媒体被清空（初始化）时触发。

**ended** 播放结束时触发。

**error** 在发生错误时触发。元素的 **error** 属性会包含更多信息。参阅 **Error handling** 获得详细信息。

**loadeddata** 媒体的第一帧已经加载完毕。

**loadedmetadata** 媒体的元数据已经加载完毕, 现在所有的属性包含了它们应有的有效信息。

**loadstart** 在媒体开始加载时触发。

**mozaudioavailable** 当音频数据缓存并交给音频层处理时

**pause** 播放暂停时触发。

**play** 在媒体回放被暂停后再次开始时触发。即, 在一次暂停事件后恢复媒体回放。

**playing** 在媒体开始播放时触发（不论是初次播放、在暂停后恢复、或是在结束后重新开始）。

**progress** 告知媒体相关部分的下载进度时周期性地触发。有关媒体当前已下载总计的信息可以在元素的 **buffered** 属性中获取到。

**ratechange** 在回放速率变化时触发。

**seeked** 在跳跃操作完成时触发。

**seeking** 在跳跃操作开始时触发。

**stalled** 在尝试获取媒体数据, 但数据不可用时触发。

**suspend** 在媒体资源加载终止时触发, 这可能是因为下载已完成或因为其他原因暂停。

**timeupdate** 元素的 **currentTime** 属性表示的时间已经改变。

**volumechange** 在音频音量改变时触发（既可以是 **volume** 属性改变, 也可以是 **muted** 属性改变）。

**waiting** 在一个待执行的操作（如回放）因等待另一个操作（如跳跃或下载）被延迟时触发

## 音频:

**abort** 在播放被终止时触发,例如, 当播放中的视频重新开始播放时会触发这个事件。

**canplay** 在媒体数据已经有足够的数据（至少播放数帧）可供播放时触发。这个事件对应 **CAN\_PLAY** 的 **readyState**。

**canplaythrough** 在媒体的 **readyState** 变为 **CAN\_PLAY\_THROUGH** 时触发, 表明媒体可以在保持当前的下载速度的情况下不被中断地播放完毕。注意: 手动设置 **currentTime** 会使得 **firefox** 触发一次 **canplaythrough** 事件, 其他浏览器或许不会如此。

**durationchange** 元信息已载入或已改变, 表明媒体的长度发生了改变。例如, 在媒体已被加载足够的长度从而得知总长度时会触发这个事件。

**emptied** 媒体被清空（初始化）时触发。

**ended** 播放结束时触发。

**error** 在发生错误时触发。元素的 **error** 属性会包含更多信息。参阅 **Error handling** 获得详细信息。

**loadeddata** 媒体的第一帧已经加载完毕。

**loadedmetadata** 媒体的元数据已经加载完毕, 现在所有的属性包含了它们应有的有效信息。

**loadstart** 在媒体开始加载时触发。

**mozaudioavailable** 当音频数据缓存并交给音频层处理时

**pause** 播放暂停时触发。

**play** 在媒体回放被暂停后再次开始时触发。即, 在一次暂停事件后恢复媒体回放。

**playing** 在媒体开始播放时触发（不论是初次播放、在暂停后恢复、或是在结束后重新开始）。

**progress** 告知媒体相关部分的下载进度时周期性地触发。有关媒体当前已下载总计的信息可以在元素的 **buffered** 属性中获取到。

**ratechange** 在回放速率变化时触发。

**seeked** 在跳跃操作完成时触发。

**seeking** 在跳跃操作开始时触发。

**stalled** 在尝试获取媒体数据, 但数据不可用时触发。

**suspend** 在媒体资源加载终止时触发, 这可能是因为下载已完成或因为其他原因暂停。

**timeupdate** 元素的 **currentTime** 属性表示的时间已经改变。

**volumechange** 在音频音量改变时触发（既可以是 **volume** 属性改变, 也可以是 **muted** 属性改变）。

**waiting** 在一个待执行的操作（如回放）因等待另一个操作（如跳跃或下载）被延迟时触发