

# <video>, <audio>

## ## 概述

`<video>`元素用来加载视频，是`HTMLVideoElement`对象的实例。`<audio>`元素用来加载音频，是`HTMLAudioElement`对象的实例。而`HTMLVideoElement`和`HTMLAudioElement`都继承了`HTMLMediaElement`，所以这两个 HTML 元素有许多共同的属性和方法，可以放在一起介绍。

理论上，这两个 HTML 元素直接用`src`属性指定媒体文件，就可以使用了。

```
```html
<audio src="background_music.mp3"/>
<video src="news.mov" width=320 height=240/>
```
```

注意，`<video>`元素有`width`属性和`height`属性，可以指定宽和高。`<audio>`元素没有这两个属性，因为它的播放器外形是浏览器给定的，不能指定。

实际上，不同的浏览器支持不同的媒体格式，我们不得不用`<source>`元素指定同一个媒体文件的不同格式。

```
```html
<audio id="music">
  <source src="music.mp3" type="audio/mpeg">
  <source src="music.ogg" type="audio/ogg; codec="vorbis">
</audio>
```
```

浏览器遇到支持的格式，就会忽略后面的格式。

这两个元素都有一个`controls`属性，只有打开这个属性，才会显示控制条。注意，`<audio>`元素如果不打开`controls`属性，根本不会显示，而是直接在背景播放。

## ## HTMLMediaElement 接口

`HTMLMediaElement`并没有对应的 HTML 元素，而是作为`<video>`和`<audio>`的基类，定义一些它们共同的属性和方法。

`HTMLMediaElement`接口有以下属性。

- `HTMLMediaElement.audioTracks`: 返回一个类似数组的对象，表示媒体文件包含的音轨。
- `HTMLMediaElement.autoplay`: 布尔值，表示媒体文件是否自动播放，对应 HTML 属性`autoplay`。

- `HTMLMediaElement.buffered`: 返回一个 `TimeRanges` 对象, 表示浏览器缓冲的内容。该对象的 `length` 属性返回缓存里面有多少段内容, `start(rangeId)` 方法返回指定的某段内容 (从0开始) 开始的时间点, `end()` 返回指定的某段内容结束的时间点。该属性只读。
- `HTMLMediaElement.controls`: 布尔值, 表示是否显示媒体文件的控制栏, 对应 HTML 属性 `controls`。
- `HTMLMediaElement.controlsList`: 返回一个类似数组的对象, 表示是否显示控制栏的某些控件。该对象包含三个可能的值: `nodownload`、`nofullscreen` 和 `noremoteplayback`。该属性只读。
- `HTMLMediaElement.crossOrigin`: 字符串, 表示跨域请求时是否附带用户信息 (比如 Cookie), 对应 HTML 属性 `crossorigin`。该属性只有两个可能的值: `anonymous` 和 `use-credentials`。
- `HTMLMediaElement.currentSrc`: 字符串, 表示当前正在播放的媒体文件的绝对路径。该属性只读。
- `HTMLMediaElement.currentTime`: 浮点数, 表示当前播放的时间点。
- `HTMLMediaElement.defaultMuted`: 布尔值, 表示默认是否关闭音量, 对应 HTML 属性 `muted`。
- `HTMLMediaElement.defaultPlaybackRate`: 浮点数, 表示默认的播放速率, 默认是1.0。
- `HTMLMediaElement.disableRemotePlayback`: 布尔值, 是否允许远程回放, 即远程回放的时候是否有工具栏。
- `HTMLMediaElement.duration`: 浮点数, 表示媒体文件的时间长度 (单位秒)。如果当前没有媒体文件, 该属性返回0。该属性只读。
- `HTMLMediaElement.ended`: 布尔值, 表示当前媒体文件是否已经播放结束。该属性只读。
- `HTMLMediaElement.error`: 返回最近一次报错的错误对象, 如果没有报错, 返回 `null`。
- `HTMLMediaElement.loop`: 布尔值, 表示媒体文件是否会循环播放, 对应 HTML 属性 `loop`。
- `HTMLMediaElement.muted`: 布尔值, 表示音量是否关闭。
- `HTMLMediaElement.networkState`: 当前网络状态, 共有四个可能的值。0表示没有数据; 1表示媒体元素处在激活状态, 但是还没开始下载; 2表示下载中; 3表示没有找到媒体文件。
- `HTMLMediaElement.paused`: 布尔值, 表示媒体文件是否处在暂停状态。该属性只读。
- `HTMLMediaElement.playbackRate`: 浮点数, 表示媒体文件的播放速度, 1.0是正常速度。如果是负数, 表示向后播放。
- `HTMLMediaElement.played`: 返回一个 `TimeRanges` 对象, 表示播放的媒体内容。该属性只读。
- `HTMLMediaElement.preload`: 字符串, 表示应该预加载哪些内容, 可能的值为 `none`、`metadata` 和 `auto`。
- `HTMLMediaElement.readyState`: 整数, 表示媒体文件的准备状态, 可能的值为0 (没有任何数据)、1 (已获取元数据)、2 (可播放当前帧, 但不足以播放多个帧)、3 (可以播放多帧, 至少为两帧)、4 (可以流畅播放)。该属性只读。
- `HTMLMediaElement.seekable`: 返回一个 `TimeRanges` 对象, 表示一个用户可以搜索的媒体内容范围。该属性只读。

- `HTMLMediaElement.seeking`: 布尔值，表示媒体文件是否正在寻找新位置。该属性只读。
- `HTMLMediaElement.src`: 布尔值，表示媒体文件的 URL，对应 HTML 属性 `src`。
- `HTMLMediaElement.srcObject`: 返回 `src` 属性对应的媒体文件资源，可能是 `MediaStream`、`MediaSource`、`Blob` 或 `File` 对象。直接指定这个属性，就可以播放媒体文件。
- `HTMLMediaElement.textTracks`: 返回一个类似数组的对象，包含所有文本轨道。该属性只读。
- `HTMLMediaElement.videoTracks`: 返回一个类似数组的对象，包含多有视频轨道。该属性只读。
- `HTMLMediaElement.volume`: 浮点数，表示音量。0.0 表示静音，1.0 表示最大音量。

`HTMLMediaElement` 接口有如下方法。

- `HTMLMediaElement.addTextTrack()`: 添加文本轨道（比如字幕）到媒体文件。
- `HTMLMediaElement.captureStream()`: 返回一个 `MediaStream` 对象，用来捕获当前媒体文件的流内容。
- `HTMLMediaElement.canPlayType()`: 该方法接受一个 MIME 字符串作为参数，用来判断这种类型的媒体文件是否可以播放。该方法返回一个字符串，有三种可能的值，`probably` 表示似乎可播放，`maybe` 表示无法在不播放的情况下判断是否可播放，空字符串表示无法播放。
- `HTMLMediaElement.fastSeek()`: 该方法接受一个浮点数作为参数，表示指定的时间（单位秒）。该方法将媒体文件移动到指定时间。
- `HTMLMediaElement.load()`: 重新加载媒体文件。
- `HTMLMediaElement.pause()`: 暂停播放。该方法没有返回值。
- `HTMLMediaElement.play()`: 开始播放。该方法返回一个 `Promise` 对象。

下面是 `play()` 方法的一个例子。

```
```javascript
var myVideo = document.getElementById('myVideoElement');

myVideo
  .play()
  .then(() => {
    console.log('playing');
  })
  .catch((error) => {
    console.log(error);
  });
```
```

## ## HTMLVideoElement 接口

`HTMLVideoElement` 接口代表了 `<video>` 元素。这个接口继承了 `HTMLMediaElement` 接口，并且有一些自己的属性和方法。

HTMLVideoElement 接口的属性。

- HTMLVideoElement.height: 字符串, 表示视频播放区域的高度 (单位像素), 对应 HTML 属性`height`。
- HTMLVideoElement.width: 字符串, 表示视频播放区域的宽度 (单位像素), 对应 HTML 属性`width`。
- HTMLVideoElement.videoHeight: 该属性只读, 返回一个整数, 表示视频文件自身的高度 (单位像素)。
- HTMLVideoElement.videoWidth: 该属性只读, 返回一个整数, 表示视频文件自身的宽度 (单位像素)。
- HTMLVideoElement.poster: 字符串, 表示一个图像文件的 URL, 用来在无法获取视频文件时替代显示, 对应 HTML 属性`poster`。

HTMLVideoElement 接口的方法。

- HTMLVideoElement.getVideoPlaybackQuality(): 返回一个对象, 包含了当前视频回访的一些数据。

## ## HTMLAudioElement 接口

`HTMLAudioElement`接口代表了`<audio>`元素。

该接口继承了`HTMLMediaElement`, 但是没有定义自己的属性和方法。浏览器原生提供一个`Audio()`构造函数, 返回的就是`HTMLAudioElement`实例。

```
```javascript
var song = new Audio([URLString]);
```
```

`Audio()`构造函数接受一个字符串作为参数, 表示媒体文件的 URL。如果省略这个参数, 可以稍后通过`src`属性指定。

生成`HTMLAudioElement`实例以后, 不用插入 DOM, 可以直接用`play()`方法在背景播放。

```
```javascript
var a = new Audio();
if (a.canPlayType('audio/wav')) {
  a.src = 'soundeffect.wav';
  a.play();
}
```
```

## ## 事件

`<video>`和`<audio>`元素有以下事件。

- loadstart: 开始加载媒体文件时触发。
- progress: 媒体文件加载过程中触发, 大概是每秒触发2到8次。
- loadedmetadata: 媒体文件元数据加载成功时触发。
- loadeddata: 当前播放位置加载成功后触发。
- canplay: 已经加载了足够的数据, 可以开始播放时触发, 后面可能还会请求数据。
- canplaythrough: 已经加载了足够的数据, 可以一直播放时触发, 后面不需要继续请求数据。
- suspend: 已经缓冲了足够的数据, 暂时停止下载时触发。
- stalled: 尝试加载数据, 但是没有数据返回时触发。
- play: 调用`play()`方法时或自动播放启动时触发。如果已经加载了足够的数据, 这个事件后面会紧跟`playing`事件, 否则会触发`waiting`事件。
- waiting: 由于没有足够的缓存数据, 无法播放或播放停止时触发。一旦缓冲数据足够开始播放, 后面就会紧跟`playing`事件。
- playing: 媒体开始播放时触发。
- timeupdate: `currentTime`属性变化时触发, 每秒可能触发4到60次。
- pause: 调用`pause()`方法、播放暂停时触发。
- seeking: 脚本或者用户要求播放某个没有缓冲的位置, 播放停止开始加载数据时触发。此时, `seeking`属性返回`true`。
- seeked: `seeking`属性变回`false`时触发。
- ended: 媒体文件播放完毕时触发。
- durationchange: `duration`属性变化时触发。
- volumechange: 音量变回或者静音时触发。
- ratechange: 播放速度或默认的播放速度变化时触发。
- abort: 停止加载媒体文件时触发, 通常是用户主动要求停止下载。
- error: 网络或其他原因导致媒体文件无法加载时触发。
- emptied: 由于`error`或`abort`事件导致`networkState`属性变成无法获取数据时触发。