

为君添增值!



## WEB前端基础 — HTML5和CSS3新特性快速参考

## 讲师：HELEN

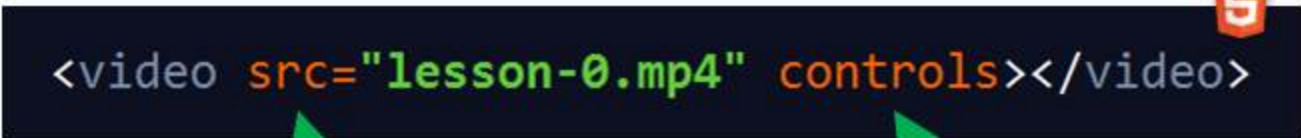




## 第七课

# HTML5视频播放

不需下载任何额外的浏览器视频播放插件  
完全由浏览器自身实现视频的解码和播放



```
<video src="lesson-0.mp4" controls></video>
```

The image shows a code block with a dark background and a red '5' icon in the top right corner. The code is `<video src="lesson-0.mp4" controls></video>`. Two green arrows point from the text labels below to the code: one points to `src="lesson-0.mp4"` and the other points to `controls`.

视频的URL地址

显示视频播放控件

为梦想增值!



# WEB前端基础 — HTML和CSS入门

51CTO学院

edu.51cto.com









src

controls

autoplay

preload

muted

loop

poster

width和height





```
<video src="lesson-0.mp4" controls autoplay></video>
```

The code is displayed in a dark-themed box. A green arrow points from the text below to the 'autoplay' attribute. A small orange icon with the number '5' is in the top right corner of the code box.

页面加载后自动播放视频



```
<video src="lesson-0.mp4" controls autoplay></video>
```

controls和autoplay是布尔属性：只有属性名，没有属性值。例如：

如果有autoplay属性，视频就会自动播放

如果没有autoplay属性，视频就不会自动播放



```
<video src="lesson-0.mp4" controls="controls" autoplay="autoplay"></video>
```

XHTML风格的代码语法更严格

属性必须由属性名和属性值两部分组成

属性值	详细说明
none	视频播放前，浏览器不会预先下载视频资源，当用户不点击播放按钮时，会节省带宽。
metadata	视频播放前，浏览器不会下载视频资源，但是会获取资源的元数据（视频大小、持续时间、视频格式、前几帧画面）
auto (默认值)	浏览器根据实际情况动态决定。例如：在wifi、3G、4G数据漫游等情况下，动态采用不同的加载方案。

这两种写法的意思相同

```
<video src="lesson-0.mp4" controls preload="auto"></video>
```

```
<video src="lesson-0.mp4" controls preload></video>
```

如果设置了autoplay属性，那么preload属性的值将会被忽略

是否默认静音播放

是否循环播放视频

```
<video src="lesson-0.mp4" controls muted loop></video>
```





cover.jpg 作为视频的封面



poster属性设置视频的封面

```
<video src="let-it-go.mp4" controls  
poster="cover.jpg"  
width="854" height="480">  
</video>
```

设置视频播放器的尺寸



视频容器

视频编码

音频编码

MP4容器

H.264视频编码

ACC音频编码

H.264由MPEG-LA授权  
文件扩展名为 **.mp4**

WebM容器

Vp8视频编码

Vorbis音频编码

WebM由google设计  
文件扩展名为 **.webm**

Ogg容器

Theora视频编码

Vorbis音频编码

Theora是一个开源编解码器  
文件扩展名为 **.ogv**



浏览器 视频	IE	FireFox	Google Chrome	Safari	Opera	IOS Safari	Opera Mini	Android Browser	Chrome for Android
MPEG-4(H.264)	√	√	√	√	√	√	×	√	√
WebM(Vp8)	×	√	√	×	√	×	×	×	√
Ogg(Theora)	×	√	√	×	√	×	×	×	×

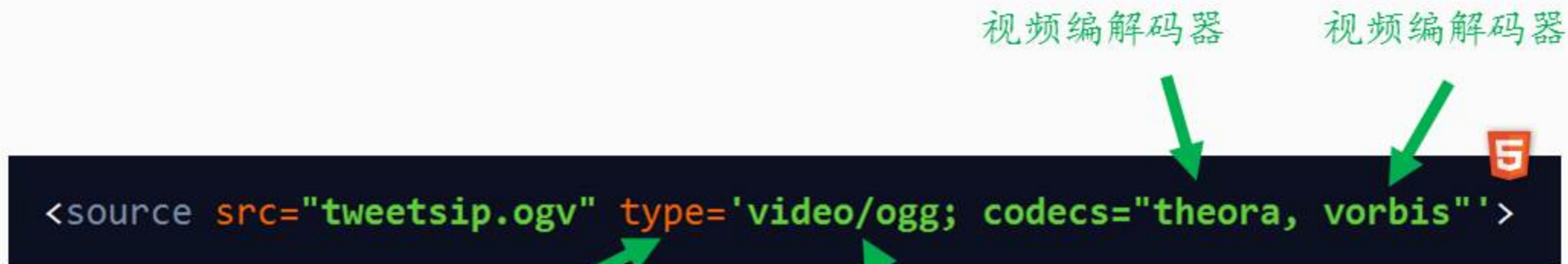
浏览器对视频格式的支持情况：<http://caniuse.com/#search=video>

浏览器从上到下查找source元素，直到找到它能播放的一种格式



```
<video controls>
  <source src="tweetsip.mp4">
  <source src="tweetsip.webm">
  <source src="tweetsip.ogv">
  <p>对不起，您的浏览器不支持video标签</p>
</video>
```

对于这里的每个source元素，浏览器都会加载视频文件的元数据，查看能不能播放这个视频，这个过程可能很耗费时间。



Type是一个可选的属性，这是向浏览器提供一个提示，帮助它确定能不能播放这种类型的文件。

这是视频文件的MIME类型，指定了容器的格式。



mp4的编解码器比另外两种更为复杂，因为H.264支持多种等级，不同的使用情况（如高带宽和低带宽）会有不同的编码。



```
<video controls>
  <source src="tweetsip.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
  <source src="tweetsip.webm" type='video/ogg; codecs="vp8, vorbis"'>
  <source src="tweetsip.ogv" type='video/ogg; codecs="theora, vorbis"'>
  <p>对不起，您的浏览器不支持video标签</p>
</video>
```



浏览器对video标签的支持情况：<http://caniuse.com/#search=video>

```
<video controls>
  <source src="tweetsip.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
  <source src="tweetsip.webm" type='video/ogg; codecs="vp8, vorbis"'>
  <source src="tweetsip.ogv" type='video/ogg; codecs="theora, vorbis"'>
  <embed src="tweetsip.swf" type="application/x-shockwave-flash"></embed>
</video>
```

embed标签或object标签用在不支持video标签  
的旧版本的浏览器中



# Thank You !

为梦想增值！

[edu.51cto.com](http://edu.51cto.com)