此文章已于 16:07:48 2016/1/20 重新发布到 Net | Linux | 安全

05. Web大前端时代之：HTML5+CSS3入门系列~H5 多媒体系

Web大前端时代之：HTML5+CSS3入门系列：<http://www.cnblogs.com/dunitian/p/5121725.html>

1.引 入

**概述**

音频文件或视频文件都可以看做是一个容器文件（类似于压缩的zip）

编解码器就是读取特定的容器格式，对其中的音频与视频轨进行解码，然后实现播放

**解码器**

解码器（decoder），是一种输入模拟视频信号并将它转换为数字信号格式，以进一步压缩和传输的硬件/软件设备

**视频格式**

**Ogg = 带有 Theora 视频编码和 Vorbis 音频编码的 Ogg 文件**

**MPEG4 = 带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件**

**WebM = 带有 VP8 视频编码和 Vorbis 音频编码的 WebM 文件**

**支持状况**

**audio的canPlayType可以检测浏览器支持的文件格式**

**在线检测（右击查看源码）**

**http://dnt.dkill.net/DNT/HTML5/demo/check.html**

**2.video**

**逆天测试仅供参考**

**测试浏览器：谷歌，360，火狐，Edge，IE ：mp4格式的基本上都支持了，wmv格式的基本上都不支持，avi格式的火狐不支持**

**案例**

1.**简单案例：**

**<video src="images/big.mp4" controls loop>你的浏览器不支持</video>**

**<video src="images/big.mp4" poster="http://dnt.dkill.net/Images/banner.jpg" controls loop></video>**

**说明：**

**loop 循环播放**

**controls 显示工具栏**

**<video src="images/big.mp4" controls>你的浏览器不支持</video>**



**poster 视频预览图**

**<video src="images/big.mp4" poster="http://dnt.dkill.net/Images/banner.jpg" controls loop></video>**

**autoplay 自动播放**

**<video src="images/big.mp4" controls autoplay loop>你的浏览器不支持</video>**

**preload 是否在页面加载后载入视频 如果设置了 autoplay 属性，则忽略该属性**

**属性值：**

**auto - 当页面加载后载入整个视频**

**meta - 当页面加载后只载入元数据 (有些是浏览器是 metadata)**

**none - 当页面加载后不载入视频**

**<video src="images/big.mp4" controls preload="auto">你的浏览器不支持</video>**

**2.多源案例**

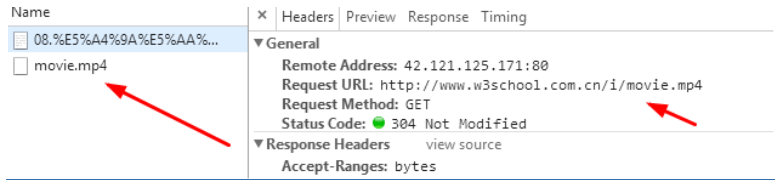
**浏览器自动选择第一个可识别的文件来播放**

**<video controls>**

**<source src="http://www.w3school.com.cn/i/movie.mp4" type="video/mp4" />**

**<source src="http://www.w3school.com.cn/i/movie.ogg" type="video/ogg"/>**

**</video>**



**3.小播放器**

**<video id="myvideo" src="../images/big.mp4" poster="http://dnt.dkill.net/Images/banner.jpg" controls loop>回家休息休息。。。</video>**

**<script type="text/javascript">**

**var musicObj = $("#myvideo");**

**//单击控制播放与否**

**musicObj.click(function () {**

**if (this.paused) {**

**this.play();**

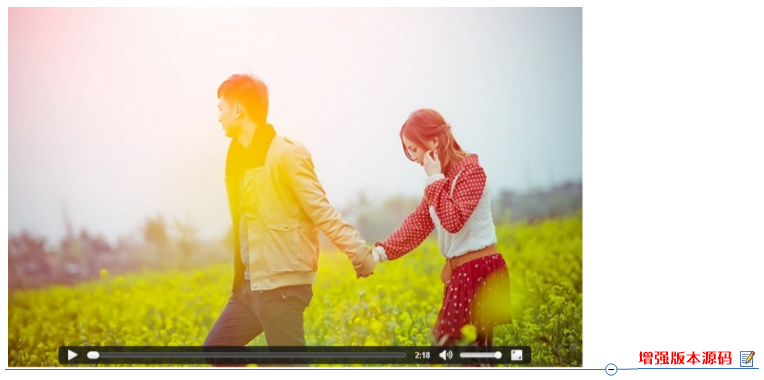
**} else {**

**this.pause();**

**}**

**});**

**</script>**

[](https://github.com/dunitian/LoTHTML5/tree/master/LoTVideo)

**3.audio**

**学过video之后这个就小儿科了，简单介绍一下**

**基本用法**

**<audio src="http://dnt.dkill.net/DNT/music/棋子.mp3" controls></audio>**

**<audio controls>**

**<source src="http://www.dkill.net/DNT/music/棋子.ogg" type="audio/ogg" />**

**<source src="http://dnt.dkill.net/DNT/music/棋子.mp3" type="audio/mp3" />**

**</audio>**

4. **扩展内容**

**大纲**

**在所有属性中，只有 videoWidth 和 videoHeight 属性是立即可用的。在视频的元数据已加载后，其他属性才可用。**

**浏览器测试**

**代码和测试结果看备注（基本上都是满足的）**

**属性**

**只读**

**ended**

返回true代表播放结束

**paused**

**ture代表暂停播放，false代表正在播放**

**seeking**

返回用户是否正在音频/视频中进行查找

**true代表查找**

**duration**

**获取视频总时间（单位秒）**

**currentSrc**

返回媒体url

**readyState**

返回音频/视频当前的就绪状态

0 = HAVE\_NOTHING - 没有关于音频/视频是否就绪的信息

1 = HAVE\_METADATA - 关于音频/视频就绪的元数据

2 = HAVE\_CURRENT\_DATA - 关于当前播放位置的数

据是可用的，但没有足够的数据来播放下一帧/毫秒

**3** = HAVE\_FUTURE\_DATA - 当前及至少下一帧的数据是可用的

4 = HAVE\_ENOUGH\_DATA - 可用数据足以开始播放

n**etworkState**

**返**回音频/视频的当前网络状态

0 = NETWORK\_EMPTY - 音频/视频尚未初始化

1 = NETWORK\_IDLE - 音频/视频是活动的且已选取资源，但并未使用网络

2 = NETWORK\_LOADING - 浏览器正在下载数据

3 = NETWORK\_NO\_SOURCE - 未找到音频/视频来源

返**回对象**

**error**

**this.error.code**

**1** = MEDIA\_ERR\_ABORTED - 取回过程被用户中止

2 = MEDIA\_ERR\_NETWORK - 当下载时发生错误

3 = MEDIA\_ERR\_DECODE - 当解码时发生错误

4 = MEDIA\_ERR\_SRC\_NOT\_SUPPORTED - 不支持音频/视频

c**onsole.log(this.error.code);**

**played**

**获**得视频中以秒计的首段已播放的范围（部分）

属性值

length - 获得音频/视频中已播范围的数量

start(index) - 获得某个已播范围的开始位置

end(index) - 获得某个已播范围的结束位置

案**例**

**获取当前播放时间：this.played.end(0)**

**seekable**

**表**示音频/视频中的可寻址部分

属性值

length - 获得音频/视频中可寻址范围的数量

start(index) - 获得可寻址范围的开始位置

end(index) - 获得可寻址范围的结束位置

a**lert("Start: " + this.seekable.start(0) + " End: " + this.seekable.end(0));**

**buffered**

**返**回一个TimeRanges对象，确认浏览器已经缓存媒体文件

属性值

length - 获得音频/视频中已播范围的数量

start(index) - 获得某个已播范围的开始位置

end(index) - 获得某个已播范围的结束位置

a**lert("Start: " + this.buffered.start(0) + " End: " + this.buffered.end(0));**

**读写**

**src**

**src 属性设置或返回音频/视频的当前来源**

**loop**

**可以脚本控制播放，设置为true为循环播放**

**muted**

**可**以脚本控制播放，设置为true为静音

p**oster**

**设置视频封面地址**

**this.poster = 'http://dnt.dkill.net/Images/banner.jpg';**

**volume**

**设置或返回音频/视频的音量**

**规定音频/视频的当前音量。必须是介于 0.0 与 1.0 之间的数字。**

**preload**

**设**置或返回音频/视频是否应该在页面加载后进行加载

属性值

auto 指示一旦页面加载，则开始加载音频/视频。

metadata 指示当页面加载后仅加载音频/视频的元数据。

none 指示页面加载后不应加载音频/视频。

c**ontrols**

**可**以脚本控制显示工具栏，设置为true为显示

a**utoplay**

**可**以脚本控制播放，设置为true为自动播放

c**urrentTime**

**获取当前播放位置，返回时间（单位秒）**

**playbackRate**

**设**置或返回音频/视频播放的速度

设**置或返回音频/视频的默认播放速度**

**1**.0 正常速度

0.5 半速（更慢）

2.0 倍速（更快）

-1.0 向后，正常速度

-0.5 向后，半速

案**例**

**<video id="myvideo"></ video>**

**$(function () {**

**var videoObj = $('#myvideo');**

**var This = videoObj.get(0);**

**This.controls = true;**

**This.poster = 'http://dnt.dkill.net/Images/banner.jpg';**

**This.src = 'images/big.mp4';**

**});**

**defaultPlaybackRate**

**专业酱油12年**

设置或返回音频**/视频的默认播放速度**

**1.0 正常速**度

0.5 半速（更慢）

2.0 倍速（更快）

-1.0 向后，正常速度

-0.5 向后，半速

不会影响当前播放速度

方法

addText**Track()**

**向音频/视频添**加新的文本轨道

语法

audio|v**ideo.addTextTrack(kind,label,language)**

**kind**

**规定文本轨道的类型。**

**"subtitles"**

**"caption"**

**"descriptions"**

**"chapters"**

**"metadata"**

**label**

**字符串值，为文**本轨道规定标签。用于为用户对文本轨道进行标识。

languag**e**

**双字母语言代码**，规定文本轨道的语言。

中文

zh

英文

en

案例

canPlay**Type()**

**检测浏览器是否**能播放指定的音频/视频类型

案例

http://**dnt.dkill.net/DNT/HTML5/demo/check.html**

**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**

**事件**

**audio（v**ideo）媒体事件

事件+描述

abort 当**音频/视频的加载已放弃时触发。**

**canplay 当浏览器可以开始播放音频/视频时触发。**

**canplaythrough 当浏览器可在不因缓冲而停顿的情况下进行播放时触发。**

**durationchange 当音频/视频的时长已更改时触发。**

**emptied 当目前的播放列表为空时触发。**

**ended 当目前的播放列表已结束时触发。**

**error 当在音频/视频加载期间发生错误时触发。**

**loadeddata 当浏览器已加载音频/视频的当前帧时触发。**

**loadedmetadata 当浏览器已加载音频/视频的元数据时触发。**

**loadstart 当浏览器开始查找音频/视频时触发。**

**pause 当音频/视频已暂停时触发。**

**play 当音频/视频已开始或不再暂停时触发。**

**playing 当音频/视频在因缓冲而暂停或停止后已就绪时触发。**

**progress 当浏览器正在下载音频/视频时触发。**

**ratechange 当音频/视频的播放速度已更改时触发。**

**seeked 当用户已移动/跳跃到音频/视频中的新位置时触发。**

**seeking 当用户开始移动/跳跃到音频/视频中的新位置时触发。**

**stalled 当浏览器尝试获取媒体数据，但数据不可用时触发。**

**suspend 当浏览器刻意不获取媒体数据时触发。**

**timeupdate 当目前的播放位置已更改时触发。**

**volumechange 当音量已更改时触发。**

**waiting 当视频由于需要缓冲下一帧而停止时触发。**

**扩**

**当音频/视频处于加载过程中时，会依次发生以下事件**

**loadstart**

**durationchange**

**loadedmetadata**

**loadeddata**

**progress**

**canplay**

**canplaythrough**

**使用方法**

**常见两种：**

**addEventListener**

**$(function () {**

**var videoObj = $('#myvideo');**

**var This = videoObj.get(0);**

**This.controls = true;**

**This.poster = 'http://dnt.dkill.net/Images/banner.jpg';**

**This.src = 'images/big.mp4';**

**This.addEventListener('pause', function () {**

**alert('视频这么精彩你丫还敢暂停？');**

**});**

**});**

**document.getElementById('myvideo').addEventListener('pause', function () {**

**alert('视频这么精彩你丫还敢暂停？');**

**});**

**属性赋值**

**$(function () {**

**var videoObj = $('#myvideo');**

**var This = videoObj.get(0);**

**This.controls = true;**

**This.poster = 'http://dnt.dkill.net/Images/banner.jpg';**

**This.src = 'images/big.mp4';**

**This.onpause = myPause;**

**});**

**function myPause() {**

**alert('视频这么精彩你丫还敢暂停？')**

**}**

