

工业和信息化人才培养工程系列丛书  
1+X证书制度试点培训用书

# Web 前端开发 **高级**

(下册)



工业和信息化部教育与考试中心 主编

 中国工信出版集团

 电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

工业和信息化人才培养工程系列丛书

1+X 证书制度试点培训用书

# Web 前端开发（高级） （下册）

工业和信息化部教育与考试中心 主编

RFID

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

面向职业院校和应用型本科院校开展 1+X 证书制度试点工作是落实《国家职业教育改革实施方案》的重要内容之一,为了便于 X 证书标准融入院校学历教育,工业和信息化部教育与考试中心组织编写了《Web 前端开发(高级)》教材。

本教材以《Web 前端开发职业技能等级标准》(高级)为编写依据,分上、下两册,包括前端高效开发框架技术与应用、移动 Web 设计与开发、性能优化与自动化技术三篇,分别对应《Web 前端开发职业技能等级标准》(高级)涉及的三门核心课程:“前端高效开发框架技术与应用”“移动 Web 设计与开发”“性能优化与自动化技术”。

本教材以模块化的结构组织各篇及其章节,以任务驱动的方式安排教材内容,选取移动 Web 典型应用作为教学案例。本教材可用于 1+X 证书制度试点工作中的 Web 前端开发职业技能等级证书教学和培训,也可以作为期望从事 Web 前端开发职业的应届毕业生和社会在职人员的自学参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

Web 前端开发:高级.下册 / 工业和信息化部教育与考试中心主编. —北京:电子工业出版社,2019.8  
(工业和信息化人才培养工程系列丛书)

1+X 证书制度试点培训用书

ISBN 978-7-121-36800-4

I. ①W… II. ①工… III. ①网页制作工具—教材 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 113243 号

责任编辑:胡辛征 特约编辑:田学清

印 刷:涿州市京南印刷厂

装 订:涿州市京南印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱

邮编:100036

开 本:787×1092 1/16 印张:15.75 字数:400 千字

版 次:2019 年 8 月第 1 版

印 次:2019 年 8 月第 1 次印刷

定 价:49.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254580, zuoya@phei.com.cn。

# 前 言



为积极响应《国家职业教育改革实施方案》，贯彻落实《关于深化产教融合的若干意见》《国家信息化发展战略纲要》的相关要求，应对新一轮科技革命和产业变革的挑战，促进人才培养供给侧和产业需求侧结构要素全方位融合，促进教育链、人才链与产业链、创新链有机衔接，推进人力资源供给侧结构性改革，深化产教融合、校企合作，健全多元化办学体制，完善职业教育和培训体系，着力培养高素质劳动者和技术技能人才。工业和信息化部教育与考试中心依据教育部《职业技能等级标准开发指南》中的相关要求，以客观反映现阶段行业的水平和对从业人员的要求为目标，在遵循有关技术规程的基础上，以专业活动为导向，以专业技能为核心，组织企业工程师、高职和本科院校的学术带头人共同开发了《Web 前端开发职业技能等级标准》。本教材以《Web 前端开发职业技能等级标准》中的职业素养和岗位技术技能为重点培养目标，以专业技能为模块，以工作任务为驱动进行组织编写，使读者对 Web 前端开发的技术体系有更系统、更清晰的认识。

随着新一轮科技革命与信息技术革命的到来，推动了产业结构调整与经济转型升级发展新业态的出现。在战略性新兴产业爆发式发展的同时，对新时代产业人才的培养提出了新的要求与挑战。据中国互联网络信息中心统计，截至 2018 年 12 月，我国网民规模达 8.29 亿人，手机网民规模达 8.17 亿人，网站数量达 523 万个，手机 App（移动应用程序）在架数量达 449 万款。在“互联网+”战略的引导下，Web 前端开发人员已经成为网站开发、手机 App 开发和人工智能终端设备界面开发的主要力量。企业增加门户网站的推广，从 PC 端到移动端，再到新显示技术、智能机器人、自动驾驶、智能穿戴设备、语言翻译、自动导航等新兴领域，全部需要应用 Web 前端开发技术。在智能制造等战略及新兴产业的高速发展中，出现了极为明显的人才短缺与发展不均衡现象。目前，软件开发行业的企业对 Web 前端开发工程师的需求量极大，全国总缺口每年近百万人。

随着移动互联网技术的高速发展，网站在静态页面的基础上添加了各类桌面软件，网页不再只是承载单一的文字和图片，而是被要求具备炫酷的页面交互、跨终端的适配兼容功能，使用富媒体让网页的内容更加生动，从而让用户有更好的使用体验，这些都基于前端技术来实现，其中包括 HTML、CSS、HTML5、CSS3、AJAX、JavaScript、jQuery 等，使得无论是在开发难度上还是在开发方式上，都对前端开发人员提出了越来越高的要求。

本教材包括前端高效开发框架技术与应用、移动 Web 设计与开发、性能优化与自动化技术 3 个篇目 18 个章节。

**第一篇前端高效开发框架技术与应用。**主要讲述了渐进式框架 Vue 的应用、Express 服务器开发及组件化开发思想。其中包括第 1 章 Vue 基础，第 2 章 Vue 组件，第 3 章 Vue 工程化工具，第 4 章 Express 服务器开发，第 5 章 axios 网络交互，第 6 章 Vue 路由，第 7 章 Vuex 状态管理，第 8 章 Vue UI。

**第二篇移动 Web 设计与开发。**主要讲述了移动 Web 开发框架 jQuery Mobile 的引用，也包含多媒体、绘图、HTML5 新特性及 Less。其中包括第 9 章多媒体与绘图，第 10 章 HTML5 新特性，第 11 章 Less，第 12 章 jQuery Mobile。

**第三篇性能优化与自动化技术。**主要讲述前端代码优化、资源优化及模块化打包工具 webpack 的使用，以及 ES6 的基本语法。其中包括第 13 章 Web 前端开发概述，第 14 章 HTML 与 CSS 代码优化，第 15 章前端资源优化，第 16 章 JavaScript 代码优化，第 17 章 webpack 工具，第 18 章 ES6 基础。

本教材的编写与审校工作由严洁萍、陈慕菁完成，董旭依据《Web 前端开发职业技能等级标准》对全书做了内容统筹、章节结构设计和统稿。

由于编者水平有限，书中难免有不足之处，恳请读者不吝赐教并提出宝贵意见，相信读者的反馈将会为本教材再次修订提供良好的帮助。

# 目 录



第二篇 移动 Web 设计与开发	
第 9 章 多媒体与绘图 .....	2
9.1 多媒体 .....	3
9.1.1 图片格式 .....	3
9.1.2 音频格式 .....	4
9.1.3 视频格式 .....	5
9.2 HTML5 的多媒体支持 .....	6
9.2.1 <audio>和<video> .....	6
9.2.2 HTMLAudioElement 和 HTMLVideoElement .....	8
9.2.3 <audio>和<video>的事件 .....	12
9.3 HTML5 的绘图支持 .....	13
9.3.1 <canvas>元素 .....	13
9.3.2 绘制图形 .....	14
9.3.3 绘制几何图形 .....	17
9.3.4 绘制路径 .....	18
9.3.5 绘制字符串 .....	32
9.3.6 清除绘制内容 .....	38
9.3.7 绘制阴影 .....	39
9.3.8 绘制位图 .....	40
9.3.9 变形 .....	41
9.4 SVG .....	43
9.4.1 在 HTML5 中使用 SVG .....	43
9.4.2 SVG 的基本语法 .....	44
9.4.3 <svg>标签 .....	45
9.4.4 <svg>内部标签 .....	46
9.4.5 几何图形标签 .....	47
9.4.6 路径标签 .....	51
9.4.7 文字标签 .....	53
9.5 本章小结 .....	54
第 10 章 HTML5 新特性 .....	55
10.1 HTML5 新增元素 .....	56
10.2 HTML5 新增全局属性 .....	56
10.3 HTML5 废弃的元素 .....	56
10.4 HTML5 废弃的属性 .....	57
10.5 Web Storage .....	58
10.6 本章小结 .....	61
第 11 章 Less .....	62
11.1 Less 简介 .....	63
11.2 Less 的安装 .....	63
11.2.1 服务器端 .....	63
11.2.2 客户端 .....	63
11.3 Less 的使用 .....	64
11.3.1 变量 .....	64
11.3.2 嵌套 .....	67
11.3.3 混合 .....	67
11.3.4 继承 .....	71
11.3.5 函数 .....	72
11.3.6 导入 .....	73
11.3.7 其他 .....	74
11.4 本章小结 .....	74
第 12 章 jQuery Mobile .....	75
12.1 jQuery Mobile 的诞生 .....	76
12.2 jQuery Mobile 的安装 .....	76
12.3 jQuery Mobile 的使用 .....	77



12.3.1	页面.....	77	14.1.2	HTML5 新特性.....	151
12.3.2	过渡.....	80	14.1.3	HTML 代码优化及 写法规范.....	154
12.3.3	定位.....	81	14.2	CSS 优化.....	154
12.3.4	按钮.....	82	14.2.1	CSS3 新特性.....	155
12.3.5	图标.....	87	14.2.2	浏览器样式重置.....	157
12.3.6	导航栏.....	90	14.2.3	CSS 样式选择器与 优先级.....	158
12.3.7	折叠.....	93	14.2.4	CSS 去冗余.....	158
12.3.8	列布局.....	98	14.2.5	CSS 浏览器兼容性.....	158
12.3.9	列表.....	99	14.3	本章小结.....	159
12.4	jQuery Mobile 表单.....	106	第 15 章	前端资源优化.....	160
12.4.1	单选按钮.....	108	15.1	Sprite 拼合图.....	161
12.4.2	复选框.....	110	15.1.1	CSS Sprite 的原理.....	161
12.4.3	选择菜单.....	111	15.1.2	CSS Sprite 制作工具的 方式.....	163
12.4.4	范围滑块.....	114	15.2	代码压缩技术.....	164
12.4.5	切换开关.....	116	15.2.1	YUI Compressor.....	165
12.5	jQuery Mobile 主题.....	117	15.2.2	gzip.....	165
12.6	jQuery Mobile 实战.....	118	15.2.3	打包工具.....	166
12.7	jQuery Mobile 事件.....	120	15.3	预加载和懒加载技术.....	168
12.7.1	页面事件.....	120	15.3.1	预加载.....	168
12.7.2	触摸事件.....	121	15.3.2	懒加载.....	169
12.7.3	滚动事件.....	123	15.4	本章小结.....	171
12.7.4	方向事件.....	124	第 16 章	JavaScript 代码优化.....	172
12.8	网页设计平台差异性.....	126	16.1	JavaScript 代码可维护性.....	173
12.9	本章小结.....	128	16.1.1	代码与结构分离.....	173
第 3 篇	性能优化与自动化技术		16.1.2	样式与结构分离.....	175
第 13 章	Web 前端开发概述.....	130	16.1.3	数据与代码分离.....	176
13.1	Web 前端开发认知.....	131	16.2	JavaScript 代码可扩展性.....	177
13.1.1	Web 发展历程.....	131	16.3	JavaScript 代码可调试性.....	178
13.1.2	Web 前端开发技术.....	132	16.4	JavaScript DOM 优化.....	180
13.1.3	Web 前端开发常见问题.....	134	16.4.1	提升文件加载速度.....	180
13.2	Web 前端开发与调试工具.....	136	16.4.2	JavaScript DOM 操作优化.....	181
13.2.1	常用 Web 前端开发工具.....	136	16.4.3	JavaScript DOM 脚本 加载优化.....	184
13.2.2	常用 Web 前端调试工具.....	143	16.5	本章小结.....	185
13.3	本章小结.....	148			
第 14 章	HTML 与 CSS 代码优化.....	149			
14.1	HTML 优化.....	150			
14.1.1	网页文档结构规范.....	150			

第 17 章	webpack 工具	186	第 18 章	ES6 基础	217
17.1	Web 前端安全性	187	18.1	ECMAScript 概述	218
17.1.1	常见安全性问题	187	18.2	Symbol 数据类型	218
17.1.2	安全性解决方案	191	18.3	let 和 const	219
17.2	npm 及模块化	192	18.3.1	let	219
17.2.1	npm 安装配置	192	18.3.2	const	220
17.2.2	npm 基本指令	193	18.4	变量的解构赋值	221
17.2.3	package.json 文件	196	18.4.1	默认值	221
17.2.4	node 模块化	198	18.4.2	解构赋值分类	221
17.3	webpack 概述	198	18.5	Set 与 Map	223
17.4	webpack 安装与配置	200	18.5.1	声明	223
17.4.1	安装 webpack	200	18.5.2	操作方法	224
17.4.2	webpack 配置详解	201	18.5.3	遍历方法	228
17.5	webpack 常用 Loader	206	18.6	箭头函数	231
17.5.1	babel-loader 编译 ES6	206	18.7	ES6 相对于 ES5 扩展	233
17.5.2	less-loader 处理 less 文件	207	18.7.1	函数的扩展	233
17.5.3	css-loader 与 style-loader 打包 CSS	209	18.7.2	对象的扩展	234
17.5.4	file-loader 与 url-loader 引入图片	210	18.7.3	数组的扩展	234
17.6	webpack 常用 Plugin	212	18.8	ES6 高级操作	235
17.6.1	HtmlWebpackPlugin 插件	213	18.8.1	Promise 对象	235
17.6.2	ExtractTextWebpackPlugin 插件	213	18.8.2	Iterator	236
17.6.3	其他 Plugin	214	18.8.3	Generator	237
17.7	本章小结	215	18.8.4	Class	238
			18.9	本章小结	239
			附录	Web 前端命名与格式规范	240

## 第二篇

# 移动 Web 设计与开发

# 第9章

## 多媒体与绘图



### 学习任务

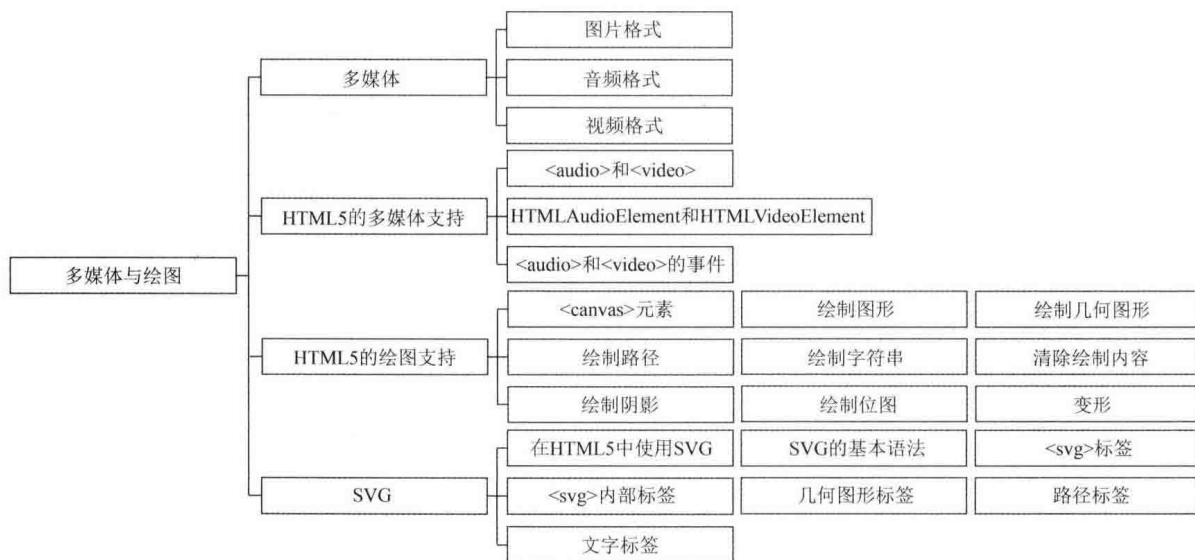
【任务 1】了解多媒体的分类。

【任务 2】掌握 HTML5 多媒体标签<audio>和<video>。

【任务 3】掌握 HTML5 绘图标签<canvas>和<svg>。



### 学习路线



## 9.1 多媒体

多媒体 (Multimedia) 是多种媒体的综合, 一般包括文本、音频、图像、动画、视频等媒体形式。多媒体以多种方式存在。在 Internet 上, 多媒体是超媒体 (Hypermedia) 系统中的一个子集, 而超媒体系统是使用超链接 (Hyperlink) 构成的全球信息系统, 全球信息系统是在 Internet 上使用 TCP/IP 和 UDP/IP 的应用系统。我们接触的多媒体主要有 4 种: 文本、图像、音频、视频。

第一批 Web 浏览器只支持文本, 甚至是单一颜色、单一字体的文本。随着 Web 的发展, Web 浏览器也逐渐支持了多种颜色、字体、文字样式、图像等, 即第二批 Web 浏览器。目前, Web 浏览器已经支持多种媒体格式, 包括音频、视频等。

多媒体元素 (比如音频或视频等) 一般存储在媒体文件中, 我们可以通过媒体文件的后缀名来确认它们是何种多媒体文件。

### 9.1.1 图片格式

图片格式是计算机存储图片的格式, 常见的图片格式包括如下几个。

- BMP 格式。

BMP 的后缀名是 “.bmp”。BMP 是一种与硬件设备无关的图像文件格式, 它采用的存储格式是位映射存储格式, 除图像深度可选以外, 不进行任何压缩, 因此, BMP 文件所占用的空间很大。

- JPEG (Joint Photographic Expert Group) 格式。

JPEG 的后缀名是 “.jpg” 或 “.jpeg”。JPEG 格式是目前最常用的图片格式之一。它是一种有损压缩, 能够将图像压缩到很小, 压缩过程中图像的重复部分或不重要的信息会丢失, 因此在高压压缩比例情况下, 很容易造成图像数据的损失。尽管这种压缩存在损失, 但由于占用空间较小, 其还是成为了互联网主流的图片格式之一。

- GIF (Graphics Interchange Format) 格式。

GIF 的后缀名是 “.gif”, 它是由 CompuServe 公司开发的图像文件格式。它是一种基于 LZW 算法的连续色调的无损压缩格式, 其压缩率一般在 50% 左右。GIF 图像最大的特点是一个文件可以存多幅彩色图像, 把这些彩色图像逐帧显示可以构成一个简单的动画, 所以它也成为了互联网主流的图片格式之一。

- PNG (Portable Network Graphics) 格式。

PNG 的后缀名是 “.png”。它支持 Alpha 通道透明度, 通常用于程序开发中的贴图, 因此也成为了互联网主流的图片格式之一。

## 9.1.2 音频格式

音频格式即音乐格式，计算机播放音频文件是对声音进行数、模转换的过程，音频格式最大带宽是 20kHz，这是因为人耳所能听到的声音的频率范围是 20Hz~20kHz。常见的音频格式包括如下几个。

- MIDI 格式。

MIDI 的后缀名是 “.mid” 或 “.midi”。MIDI 是一种在电子音乐设备（比如合成器与 PC 声卡）之间传送音乐信息的格式。MIDI 文件并不是一段录制好的声音，而是记录声音的信息，然后再告诉声卡如何再现音乐的一组指令。MIDI 格式的优点是，由于它只包含指令（音符），所以 MIDI 文件可以非常小，每存 1 分钟的音频大约只用 5~10KB 的空间。MIDI 的缺点是，它无法记录声音（仅能记录音符），也就是说它不能存储歌曲，仅能存储曲调。所以 MIDI 格式主要用于原始乐器作品、流行歌曲的表演、游戏或软件音轨等。

- CD 格式。

CD 的后缀名是 “.cda”。CD 格式是 CD 音乐光盘中的文件格式，它是标准的 44.1kHz 的采样频率，16 位量化位数，音质比较高。在计算机上看到的 CDA 文件都是 44 字节长，这是因为 CDA 文件只是一个索引信息，并不包含声音的信息，声音的信息通常需要专业的声音采样工具来进行采集转换。

- WAVE 格式。

WAVE 的后缀名是 “.wav”。WAVE 文件格式是由微软和 IBM 联合开发的用于音频数字存储的标准，它采用 RIFF 文件格式结构，非常接近于 AIFF 和 IFF 格式。WAVE 文件数据块包含以脉冲编码调制（PCM）格式表示的样本。在 Windows 平台下，基于 PCM 编码的 WAVE 格式是被支持得最好的音频格式，所有音频软件都能完美支持，而且本身可以达到较高的音质要求，缺点是因为音质高，所以需要占用较大的存储空间。

- AU 格式。

AU 的后缀名是 “.au”。AU 格式类似 WAVE 格式，是为 UNIX 操作系统开发的一种音频格式。

- AIFF 格式。

AIFF 的后缀名是 “.aiff”。AIFF 格式类似 WAVE 格式，是苹果公司开发的用于音频数字存储的标准。

- MP3（Moving Picture Experts Group Audio Layer III）格式。

MP3 的后缀名是 “.mp3”。MP3 格式是一种音频压缩技术。它丢弃掉脉冲编码调制音频数据中对人类听觉不重要的数据，从而实现了较小的文件大小。

- RealAudio 格式。

RealAudio 的后缀名是“.rm”或“.ram”。RealAudio 格式是由 Real Media 针对 Internet 开发的。该格式也支持视频。该格式支持低带宽条件下的音频流（在线音乐、网络音乐）。由于是低带宽优先的，所以质量常会降低。

- WMA（Windows Media Audio）格式。

WMA 的后缀名是“.wma”。WMA 格式的质量优于 MP3，兼容大多数播放器。WMA 文件可作为连续的数据流来传输。

### 9.1.3 视频格式

视频格式是视频播放软件为了能够播放视频文件而赋予视频文件的一种识别符号。视频格式可以分为适合在本地播放的本地影像视频和适合在网络中播放的网络流媒体影像视频两大类。常见的视频格式包括如下几个。

- AVI（Audio Video Interleave）格式。

AVI 的后缀名是“.avi”。AVI 格式是由微软开发的。所有运行 Windows 操作系统的计算机都支持 AVI 格式。它是 Internet 上很常见的格式，但在非 Windows 操作系统的计算机上支持性就不是很好了。

- WMV（Windows Media Video）格式。

WMV 的后缀名是“.wmv”。WMV 格式是由微软开发的。WMV 在 Internet 上很常见，但是如果未安装额外的（免费）组件，则无法播放 WMV 电影。它和 AVI 格式一样，在非 Windows 操作系统的计算机上支持性就不是很好了。

- MPEG（Moving Pictures Expert Group）格式。

MPEG 的后缀名是“.mpeg”。MPEG 格式是 Internet 上流行的格式。它是跨平台的，得到了所有流行的浏览器的支持。

- QuickTime 格式。

QuickTime 的后缀名是“.mov”。QuickTime 格式是由苹果公司开发的，是 Internet 上常见的格式，但是 QuickTime 格式的电影不能在未安装额外组件的 Windows 操作系统的计算机上播放。

- RealVideo 格式。

RealVideo 的后缀名是“.rm”或“.ram”。RealVideo 格式是由 Real Media 针对 Internet 开发的。该格式支持低带宽条件下（在线视频、网络视频）的视频流。由于是低带宽优先的，所以质量常会降低。

- Flash 格式。

Flash 的后缀名是 “.swf” 或 “flv”。Flash 格式是由 Macromedia 公司开发的，Flash 格式需要额外的组件来播放。

- Mpeg-4 格式。

Mpeg-4 的后缀名是 “.mp4”。Mpeg-4 是一种针对 Internet 的新格式，具有占用空间小、清晰度较好等优点。

## 9.2 HTML5 的多媒体支持

### 9.2.1 <audio>和<video>

在 HTML5 规范出现之前，Web 页面访问音频和视频等多媒体主要是通过 Flash、Activex 插件、Silverlight 等实现的，其中 Flash 是主流方式。但是随着互联网的不断发展，尤其是进入移动互联网时代以后，Flash 渐渐被 HTML5 取代，主要原因是 Flash 经常出现漏洞，安全性令人担忧，而且性能方面较差，对设备的电池消耗比较大，等等。Flash 天生就是为 PC 而生的，无法适应移动时代的特点，所以被各大厂商逐渐抛弃。在 2011 年，拥有 Flash 的 Adobe 公司做出了一个令人惊讶的决定：宣布停止对 Android 移动设备 Flash 播放器的开发工作。目前移动互联网的多媒体支持主要有音频支持和视频支持两部分，如下所示。

- audio: 定义声音播放器，比如音乐或其他音频流，支持 MP3、WAV、Ogg 三种格式。
- video: 定义视频播放器，比如电影片段或其他视频流，支持 MP4、WebM、Ogg 三种格式。

这两个标签在使用上大同小异，基本语法如下：

```
<audio src="****" controls="controls">
当前浏览器不支持 audio
</audio>
<video width="800" height="" src="****" controls="controls">
当前浏览器不支持 video
</video>
```

其中 src 设置多媒体文件的路径，controls 设置是否使用播放控件，如果在标签里写了 controls="controls"，那么网页会显示 audio 自带的播放控件，如果没写就不会显示。开始标签和结束标签中间的内容是供不支持<audio>或<video>标签的浏览器显示的。

示例代码如下：

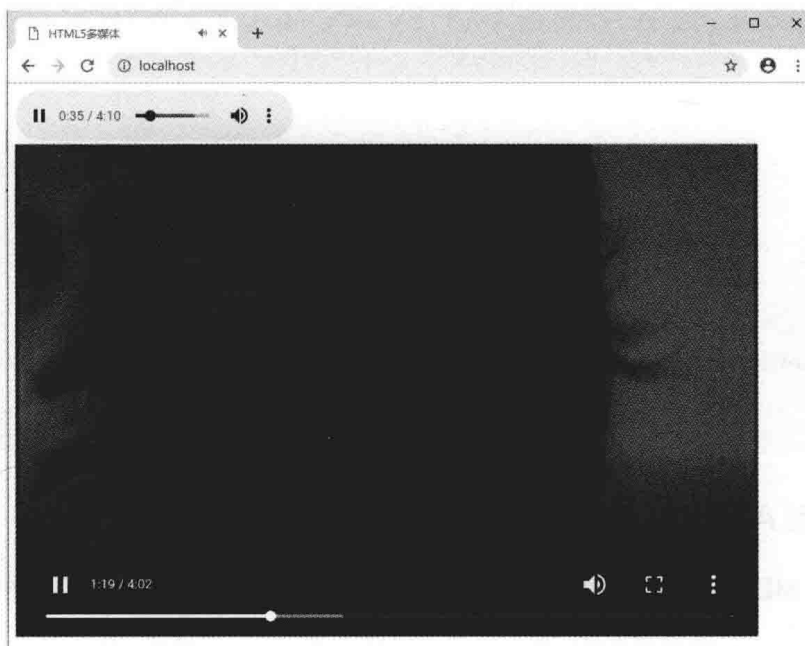
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML5 多媒体</title>
```

```

</head>
<body>
  <audio src="song.mp3" controls="controls">
    当前浏览器不支持 audio
  </audio>
  <br />
  <video width="800" height="" src="mv.mp4" controls="controls">
    当前浏览器不支持 video
  </video>
</body>
</html>

```

运行结果如下图所示。



<audio>和<video>标签还支持一些其他属性，如下表所示。

属性名	说明
autoplay	如果设置为 autoplay，表示音频和视频加载完成后会自动播放，默认不设置，通常需要配合 muted 属性来使用
loop	如果设置为 loop，表示音频和视频播放完成后会再次重复播放，默认不设置
muted	如果设置为 muted，表示音频输出为静音
preload	如果设置为 auto，表示预加载音频和视频
	如果设置为 metadata，表示预加载音频和视频的元数据，如大小、时间、第一帧等
	如果设置为 none，表示不执行预加载
poster	只对<video>有效，设置视频加载完成播放前显示的图片，属性值为图片 URL
width	只对<video>有效，设置视频播放器的宽度
height	只对<video>有效，设置视频播放器的高度

考虑到各浏览器对音频和视频的支持不相同，通常会给音频和视频指定多个媒体源，

让浏览器自动加载最合适的媒体源，HTML5 提供了<source>元素来设置多个媒体源。<source>元素需要指定两个属性，src 属性设置音频和视频的 URL，type 属性设置音频和视频的 MIME 类型。示例代码如下：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML5 多媒体</title>
  </head>
  <body>
    <audio controls="controls">
      <source src="song.mp3" type="audio/mp3" />
      <source src="song.wav" type="audio/x-wav" />
      <source src="song.ogg" type="audio/ogg" />
      当前浏览器不支持 audio 元素
    </audio>
    <video controls="controls">
      <source src="mv.mp4" type="video/mp4" />
      <source src="mv.webm" type="video/webm" />
      <source src="mv.ogv" type="video/ogg" />
      当前浏览器不支持 video 元素
    </video>
  </body>
</html>
```

## 9.2.2 HTMLAudioElement 和 HTMLVideoElement

除了在 HTML（HyperText Markup Language，超文本标记语言）页面中使用<audio>元素和<video>元素来播放音频和视频，更多的时候需要使用 JavaScript 脚本来控制<audio>元素和<video>元素，在 JavaScript 中获取<audio>元素的对象为 HTMLAudioElement，获取<video>元素的对象为 HTMLVideoElement。

HTMLAudioElement 和 HTMLVideoElement 支持的方法如下表所示。

方法名	说明
play()	播放音频和视频
pause()	暂停音频和视频
load()	重新加载音频和视频
canPlayType()	判断支持的 type 类型，属性值可以是 MIME 字符串、codecs 属性，也可以是 MIME 字符串并带 codecs 属性。它的返回值有以下 3 种：probably，表示该浏览器支持播放这种格式的音频和视频；maybe，表示该浏览器可能支持播放这种格式的音频和视频；空字符串，表示该浏览器不支持播放这种格式的音频和视频

有了上述方法，就可以通过 JavaScript 来控制<audio>元素和<video>元素了，示例代码如下：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML5 多媒体</title>
  </head>

  <body>
    <audio id="audiol" controls="controls">
      <source src="song.mp3" type="audio/mp3" />
      <source src="song.wav" type="audio/x-wav" />
      <source src="song.ogg" type="audio/ogg" />
    </audio>
    <br>
    <button type="button" onclick="playMusic()">播放</button>
    <button type="button" onclick="pauseMusic()">暂停</button>
    <button type="button" onclick="loadMusic()">重新加载</button>
    <button type="button" onclick="showCanPlay()">显示支持的音频格式
  </button>
    <p id="text"></p>
  </body>
  <script type="text/javascript">
    var player = document.getElementById("audiol");

    function playMusic() {
      player.play();
    }

    function pauseMusic() {
      player.pause();
    }

    function loadMusic() {
      player.load();
    }

    function showCanPlay() {
      var str = player.canPlayType("audio/mp3") + " MP3 格式<br />" +
player.canPlayType("audio/x-wav") + " WAV 格式<br />" + player.canPlayType
("audio/ogg") + " OGG 格式<br />";
      document.getElementById("text").innerHTML = str;
    }
  </script>
</html>
```

运行结果如下图所示。