

慧与国际软件人才及产业基地慧据教育系列丛书

# Web 前端开发技术 —— HTML5+Ajax+jQuery

张 朋 编著

清华大学出版社

北 京

## 内 容 简 介

本书介绍了 Web 前端开发需要的各种技术, 如 HTML5、CSS3、JavaScript、jQuery、jQuery Mobile, 以及多媒体新特性、HTML5 浏览器兼容性测试、DOM 模型、Ajax 原理、JSON 格式等。本书是作者多年教学、项目经验的总结, 汇集了学生在学习 Web 前端技术时遇到的各种问题及解决方案, 在介绍范例代码的时候, 使用了较新的开发工具及在线编辑工具, 使读者能够高效地完成书中的案例。

本书适合 Web 前端技术初学者学习, 也适合软件培训机构采纳作为培训教材, 可作为应用类本科及高职高专学生 Web 课程的教材。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

### 图书在版编目(CIP)数据

Web 前端开发技术——HTML5+Ajax+jQuery / 张朋 编著. —北京: 清华大学出版社, 2017

(慧与国际软件人才及产业基地慧据教育系列丛书)

ISBN 978-7-302-44807-5

I. ①W… II. ①张… III. ①超文本标记语言—程序设计②计算机网络—程序设计③JAVA 语言—程序设计 IV. ①TP312②TP393.09

中国版本图书馆 CIP 数据核字(2016)第 192649 号

责任编辑: 王 军 李维杰

装帧设计: 牛静敏

责任校对: 成凤进

责任印制: 刘海龙

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者: 北京泽宇印刷有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 19.5 字 数: 487 千字

版 次: 2017 年 7 月第 1 版 印 次: 2017 年 7 月第 1 次印刷

定 价: 56.00 元

---

产品编号: 070784-01

# 前 言

惠普在 2015 年 11 月进行了战略拆分，正式拆分成两家独立企业，其中新成立“Hewlett Packard Enterprise (简称为 HPE)”，专注于企业级业务，核心业务就是领先的软件资产，由惠普领先的企业技术基础架构业务、软件业务和服务业务组成；另一家为 HP Inc，由惠普领先的个人系统业务和打印业务组成。

2016 年 2 月 1 日起，HPE 正式发布了新的品牌名称及品牌 LOGO。HPE 中文品牌名称“慧与”应运而生。“慧”象征智慧，和“惠”同样的发音联结着我们与过去的传承与发展。“与”寓意合作，表达我们与员工、合作伙伴、客户间携手共赢的关系。

慧与国际软件人才及产业基地开展实训业务三年来，在全国几十所高校设立了共建专业，并与全国几千家企业建立了人才合作关系，培养了近万名软件人才。其中软件开发方向开设了 Java EE、大数据、.NET、Android 等专业，处于行业领先水平。

近年来，随着互联网的兴起以及传统企业与互联网的结合，前端 Web 系统的复杂性不断增加，出现了许多专职的前端职位，比如高级前端开发工程师、前端架构师。Web 网站是一个从前到后的系统工程，从公司的角度看，实现这个系统需要的技术资源越来越多，包括系统设计、JavaScript、后端技术、UX、数据库。2010 年之后，后端技术更多迁移到云及大数据上，而其他技术大多涉及前端开发，以至于在 2010 年和 2014 年底出现了 HTML5 工程师需求的爆发式增长。

Web 前端方面的工作由原来的重视 Web 设计，发展到 Web 设计、Web 开发、Web 交互、Web 性能并重，这无疑对 Web 前端行业的从业者提出了巨大挑战。现在网上有许多介绍前端开发知识的文章，但是前端从业者发现找到适合自己的知识却不容易。本书从全局角度，尽量覆盖一名前端开发工程师所需的技能，构建企业所需的技能体系。

全书分四大部分，内容涉及 HTML5、CSS3、MongoDB、JavaScript、jQuery 和 jQuery Mobile。

其中每部分包含多个章节。本书是一本涵盖 Web 前端开发基础知识的教程，不仅包括从事 Web 前端开发需要掌握的理论知识和技术详解，还包括企业项目的实战，分析前端技术的使用场景及其在项目中的作用。

## 本书内容

全书共分为 10 章，主要内容如下：

第 1 章：介绍 HTML5 的发展历史、HTML5 的新特性及其背后的设计理念，详细介绍 HTML5 的新元素，包括 HTML5 新的结构元素、分组元素和语义元素、多媒体元素和 HTML5 表单。

第 2 章：介绍 CSS 的新特性、BootStrap 的作用及其使用方法。

第 3 章：介绍 JavaScript 基础内容以及使用 JavaScript 创建交互式网页，从 JavaScript 的基本语句到对象和函数的高级用法，都列举了实例，还介绍了浏览器的事件处理方法。

第 4 章：JavaScript 图形编程是 HTML5 最吸引人的特性之一，本章由浅入深地介绍了画布的各种概念和使用方法，还穿插了多个实例，帮助读者理解画布的具体开发流程。

第 5 章：介绍 MongoDB 的基本概念、增删改查基本操作，使读者能够马上入门和体验 MongoDB 的好处。

第 6 章：介绍 jQuery 的核心内容，使用 jQuery 能用最少的代码实现最多的功能，还介绍如何遍历 HTML 文档、处理事件，并对浏览器 DOM 模型进行了补充介绍。

第 7 章：介绍 Ajax 异步通信原理，并根据这个原理分析 jQuery 是如何实现这些异步通信的。

第 8 章：介绍流行的 jQuery 插件，讨论 jQuery 的扩展能力以及如何寻找插件。

第 9 章：介绍 Ajax 的具体应用，使用 jQuery Mobile 开发移动端应用。

第 10 章：介绍 jQuery 最常见的动画效果，结合使用 jQuery 的 DOM 操作、CSS 操作、动画特效和制作出绚丽的页面。

# 目 录

第 1 章 HTML5	1	1.5.2 使用<ol>元素创建有序列表	19
1.1 HTML5 介绍	1	1.6 HTML5 语义元素	20
1.1.1 HTML5 发展史	1	1.6.1 <ruby>元素	20
1.1.2 谁在开发 HTML5	2	1.6.2 <small>元素	21
1.1.3 HTML5 的主要新功能	2	1.6.3 <time>元素	21
1.2 HTML5 的优势	2	1.6.4 <details>元素	22
1.2.1 解决跨浏览器问题	2	1.6.5 <progress>元素	22
1.2.2 用户优先的原则	3	1.6.6 <datalist>元素	22
1.2.3 化繁为简的优势	4	1.6.7 <meter>元素	23
1.3 HTML5 基本语法	4	1.6.8 <mark>元素	23
1.3.1 新的 DOCTYPE	6	1.6.9 <figure>及<figcaption>元素	24
1.3.2 新的字符编码	7	1.7 HTML5 多媒体	24
1.3.3 引用链接和样式表	7	1.7.1 <audio>元素	24
1.3.4 HTML5 标签的向后兼容性	8	1.7.2 <video>元素	25
1.3.5 <base>元素	9	1.7.3 音频、视频容器和编解码器	26
1.4 HTML5 内容分组	9	1.7.4 跨浏览器问题	28
1.4.1 <header>元素	10	1.8 HTML5 表单介绍	28
1.4.2 <footer>元素	10	1.9 HTML5 新增的输入元素	29
1.4.3 <address>元素	10	1.10 HTML5 新增的表单属性	38
1.4.4 <article>元素	10	1.10.1 autocomplete 属性	38
1.4.5 <nav>元素	11	1.10.2 required 属性	38
1.4.6 <section>元素	11	1.10.3 multiple 属性	38
1.4.7 <hr>元素	12	1.10.4 autofocus 属性	38
1.4.8 <blockquote>元素	12	1.10.5 formnovalidate 属性	39
1.4.9 <aside>元素	12	1.10.6 placeholder 属性	39
1.4.10 <hgroup>元素	12	1.10.7 list 属性	40
1.4.11 <a>元素	12	1.10.8 pattern 属性	40
1.4.12 HTML5 的文档大纲	13	本章小结	44
1.5 HTML5 使用列表	18	本章练习	45
1.5.1 使用<ul>元素创建无序列表	18		

<b>第 2 章 CSS3 和 Bootstrap</b> .....	47	3.3.4 检测属性	89
2.1 CSS3 简介	47	3.3.5 JavaScript 数组	89
2.1.1 CSS3 边框	47	3.3.6 JavaScript 内置对象	91
2.1.2 CSS3 多列	48	3.3.7 浏览器对象	96
2.1.3 CSS3 Box-Sizing	49	<b>3.4 JavaScript 函数</b>	100
2.1.4 CSS3 背景	49	3.4.1 创建函数	100
2.1.5 CSS3 2D 转换	52	3.4.2 调用函数	101
2.1.6 CSS3 过渡	54	3.4.3 匿名函数	103
2.1.7 CSS3 动画	57	3.4.4 作用域和闭包	103
2.2 less 简介	58	<b>3.5 事件处理</b>	104
2.2.1 在客户端使用 less	59	3.5.1 浏览器事件介绍	104
2.2.2 混合	60	3.5.2 处理事件的两种方法	105
2.2.3 嵌套规则	61	3.5.3 事件对象	107
2.2.4 函数 & 运算	62	本章小结	110
2.3 Bootstrap 简介	62	本章练习	110
任务一：下载 Bootstrap 并查看其		<b>第 4 章 JavaScript 图形</b>	111
目录结构	63	4.1 走进 canvas 的世界	111
任务二：个人简历的制作	63	4.1.1 canvas 是什么	111
本章小结	70	4.1.2 在页面中放置 canvas 元素	111
本章练习	70	4.1.3 canvas 坐标	114
<b>第 3 章 JavaScript</b>	71	4.1.4 绘制带边框的矩形	114
3.1 JavaScript 简介	71	<b>4.2 绘制基本图形</b>	116
3.1.1 JavaScript 是什么	71	4.2.1 绘制线条	116
3.1.2 JavaScript 的作用	72	4.2.2 绘制圆形	116
3.1.3 在网页中插入 JavaScript 的		<b>4.3 绘制文本</b>	117
方法	72	4.3.1 文本设置	117
3.1.4 如何调试 JavaScript	74	4.3.2 文本的对齐方式	118
3.2 JavaScript 基础语法	75	<b>4.4 使用图像</b>	125
3.2.1 JavaScript 的变量	75	4.4.1 绘制图像	125
3.2.2 JavaScript 的 6 种数据类型	76	4.4.2 放大缩小图像	126
3.2.3 JavaScript 的运算符	79	4.4.3 平铺图像	127
3.2.4 JavaScript 流程控制	81	4.4.4 裁剪图像	129
3.3 JavaScript 对象	85	本章小结	133
3.3.1 创建对象	85	本章练习	133
3.3.2 属性的查询和设置	88		
3.3.3 删除属性	88		

第 5 章 MongoDB .....	135	6.5 jQuery 对 DOM 属性操作 .....	166
5.1 MongoDB 初探 .....	135	6.5.1 获取和设置元素属性 .....	166
5.2 MongoDB 的下载与安装 .....	136	6.5.2 获取和设置元素内容 .....	167
5.3 MongoDB 的基本概念 .....	138	6.5.3 获取和设置元素的 CSS 属性 .....	168
5.3.1 数据库 .....	138	6.6 jQuery 响应事件 .....	170
5.3.2 文档和集合 .....	138	6.6.1 绑定事件 .....	174
5.4 MongoDB 常用命令 .....	142	6.6.2 移除事件 .....	175
5.4.1 插入数据 .....	142	6.6.3 Event 实例 .....	175
5.4.2 查询数据 .....	142	6.6.4 触发事件 .....	176
5.4.3 删除数据 .....	144	6.6.5 自定义事件 .....	176
5.4.4 更新数据 .....	144	本章小结 .....	179
5.4.5 索引 .....	145	本章练习 .....	179
本章小结 .....	147	第 7 章 基于 jQuery 的 Ajax 技术 .....	181
本章练习 .....	148	7.1 Ajax 异步请求原理 .....	181
第 6 章 jQuery .....	149	7.1.1 Ajax 技术应用程序模型 .....	181
6.1 jQuery 入门 .....	149	7.1.2 使用原始的 Ajax 与服务 器通信 .....	182
6.1.1 jQuery 初体验 .....	149	7.2 jQuery 中载入文档 .....	185
6.1.2 jQuery 环境搭建 .....	150	7.2.1 load()方法载入文档 .....	185
6.1.3 jQuery 中的\$及其作用 .....	150	7.2.2 JSON 格式 .....	186
6.1.4 jQuery 对象和 DOM 对象 .....	153	7.2.3 比较 JSON 和 XML 数据 格式 .....	187
6.2 jQuery 选择器 .....	154	7.2.4 \$.getJSON()方法载入文档 .....	191
6.2.1 jQuery 基本选择器 .....	154	7.2.5 \$.getScript()方法载入文档 .....	194
6.2.2 jQuery 层次选择器 .....	155	7.3 jQuery 中请求服务器数据 .....	194
6.3 jQuery 过滤选择器 .....	155	7.3.1 \$.post()方法 .....	194
6.3.1 jQuery 基本过滤 .....	155	7.3.2 \$.get()方法 .....	195
6.3.2 jQuery 内容过滤 .....	156	7.3.3 表单的序列化方法 .....	196
6.3.3 jQuery 子元素过滤 .....	156	7.4 \$.ajax()方法 .....	199
6.3.4 jQuery 属性过滤 .....	157	7.4.1 详解\$.ajax()的细节 .....	199
6.3.5 jQuery 表单属性过滤 .....	158	7.4.2 \$.ajax()全局事件 .....	203
6.4 jQuery 操作 DOM 元素 .....	158	7.4.3 \$.ajax()全局设定 .....	205
6.4.1 元素的创建 .....	158	本章小结 .....	208
6.4.2 元素的插入 .....	159	本章练习 .....	208
6.4.3 元素的删除 .....	159		
6.4.4 元素的包裹 .....	160		
6.4.5 元素的替换和复制 .....	162		
6.4.6 遍历和筛选 DOM 元素 .....	164		

<b>第 8 章 jQuery 插件</b> .....	<b>209</b>		
8.1 jQuery 插件介绍.....	209		
8.1.1 通过全局函数创建插件.....	209		
8.1.2 通过\$.fn()方法创建插件.....	211		
8.1.3 让插件接收参数.....	212		
8.2 寻找插件.....	214		
8.2.1 搜索 jQuery 插件.....	214		
8.2.2 选择 jQuery 插件.....	215		
8.3 jQuery ImageArea Select 插件介绍.....	215		
8.3.1 jQuery ImageArea Select 下载及安装.....	215		
8.3.2 使用 jQuery ImageArea Select.....	216		
8.4 jQuery templating 插件介绍.....	221		
8.4.1 jQuery templating 下载及安装.....	221		
8.4.2 使用 jQuery templating.....	225		
8.5 jQuery UI 插件介绍.....	229		
8.5.1 jQuery UI 下载及安装.....	229		
8.5.2 使用 jQuery UI.....	231		
本章小结.....	241		
本章练习.....	241		
<b>第 9 章 使用 Ajax 和 jQuery Mobile</b> .....	<b>243</b>		
9.1 以 Ajax 方式加载数据.....	243		
9.1.1 加载 XML 数据.....	243		
9.1.2 加载 JSON 数据.....	245		
9.2 以 Ajax 方式提交数据.....	246		
9.2.1 无刷新的表单修改.....	246		
9.2.2 无刷新的表单新增.....	248		
9.2.3 无刷新的列表数据.....	251		
9.2.4 加载更多列表数据.....	252		
9.3 jQuery Mobile 介绍.....	254		
9.3.1 jQuery Mobile 特性.....	254		
9.3.2 jQuery Mobile 初始配置.....	256		
9.4 jQuery Mobile 页面和对话框.....	256		
9.4.1 页面基础.....	256		
9.4.2 创建多个页面.....	258		
9.4.3 将页面用作对话框.....	259		
9.5 jQuery Mobile 表单.....	260		
9.5.1 jQuery Mobile 表单输入.....	260		
9.5.2 jQuery Mobile 表单选择.....	262		
9.6 jQuery Mobile 列表.....	263		
9.6.1 jQuery Mobile 列表视图.....	263		
9.6.2 jQuery Mobile 列表内容.....	265		
9.7 jQuery Mobile 事件.....	266		
9.7.1 jQuery Mobile 页面事件.....	266		
9.7.2 jQuery Mobile 触屏事件.....	268		
9.7.3 jQuery Mobile 方向事件.....	270		
本章小结.....	271		
本章练习.....	271		
<b>第 10 章 jQuery 动画特效</b> .....	<b>273</b>		
10.1 jQuery 常用特效.....	273		
10.1.1 显示和隐藏特效.....	273		
10.1.2 淡入淡出特效.....	275		
10.1.3 滑动特效.....	277		
10.2 创建自定义动画.....	281		
10.2.1 基本自定义动画.....	281		
10.2.2 自定义动画的回调函数.....	283		
10.2.3 动画队列.....	283		
10.2.4 停止特效.....	286		
10.3 综合案例赏析.....	288		
本章小结.....	293		
本章练习.....	294		
<b>附录 A 服务端接口</b> .....	<b>295</b>		

# 第1章 HTML5

2014年10月28日，万维网联盟(W3C)宣布，历经8年的努力，HTML5标准规范最终制定完成，并已公开发布。此前的几年时间里，已经有很多开发者陆续使用了HTML5的部分技术，Firefox、Google Chrome、Opera、Safari 4+、Internet Explorer 9+都已支持HTML5。

HTML5是近十年来Web开发标准最巨大的飞跃。和以前的版本不同，HTML5并非仅用来表示Web内容，它的新使命是将Web带入一个成熟的应用平台。在HTML5平台上，视频、音频、图像、动画以及与计算机的交互都被标准化。HTML5在这个时间定稿，不早不晚，正是硬件性能更强、手机OS迭代速度下降的时期。随着HTML5标准的定稿，一切纷争将告一段落。现在，属于HTML5的时代到来了。

本章内容：

- 掌握HTML5的文档结构
- 掌握HTML5新增的语义化元素
- 使用HTML5中的多媒体元素
- 掌握HTML5表单元素

## 1.1 HTML5 介绍

### 1.1.1 HTML5 发展史

互联网发展到2005年前后，开始出现了一个变化，就是宽带互联。随着宽带的普及和计算机性能的增强，人们不再满足于单纯地通过互联网看新闻、收发邮件，消耗更高带宽的娱乐产品开始出现，就是流视频和网页游戏。HTML标准没有把握住产业的变化而及时演进，浏览器产品也未升级，这块新需求被浏览器插件满足了，那就是Flash。这个部署在亿万浏览器里的商业插件俨然成为事实标准。但乔布斯力挺HTML5，他坚持在iOS上不兼容Flash。为了能够支持新的Web应用，同时克服现有的缺点，HTML迫切需要添加新功能、制定新规范。

一小组人为了弥补HTML中最薄弱的环节，开始专门针对Web应用开发新功能，他们创立了HTML5规范。就是在那个时候开创了Web的第二个时代，Web 2.0这个新词也被发明出来，这主要表现在旧的静态网站逐渐让位于需要更多特性的动态网站和社交网站。2006年，W3C又重新介入HTML，并于2008年发布了HTML5的工作草案。到了2010年，因为HTML5能解决非常实际的问题，所以在规范还未定稿的情况下，各大浏览器厂家就已经按捺不住了，开始对旗下产品进行升级以支持HTML5的新功能。这样，得益于浏览器的实验性反馈，HTML5规范得到了持续完善，HTML5以这种方式迅速融入对Web平台的实质性

改进中。虽然 HTML5 已经定稿,但 HTML5 仍在完善中,并计划于 2016 年底前发布 HTML5.1 推荐标准。HTML5 正式版的网址是:

<http://www.w3.org/TR/2014/REC-html5-20141028/>

HTML5 可以是存在于手机桌面的图标,也可以来自超级 App(如微信朋友圈),以及搜索引擎、应用市场、广告联盟,为桌面和移动平台带来无缝衔接的丰富内容。HTML5 像血管一样连接一个个独立的 APP 器官并将血液输送到不同的 APP 中。

HTML5 的发展,有用户的需求在推动,也有技术开发者的需求在推动。学习 Objective-C 和 Java 比较费劲,既然会网页开发,为何不试试 HTML5 呢?它支持传统的桌面平台,又支持移动平台,包括 iOS、Android、BlackBerry、Windows Phone 等。这种跨平台的特性可以让所有平台共用一个 UI 代码库,节省几倍的开发时间。

### 1.1.2 谁在开发 HTML5

上面我们提到了一些开发 HTML5 的组织,下面进行详细介绍:

- **WHATWG:** 由来自 Apple、Mozilla、Google、Opera 等浏览器厂商的人组成,成立于 2004 年。WHATWG 开发 HTML 和 Web 应用 API,同时为各浏览器厂商以及其他有意向的组织提供开放式合作。
- **W3C:** W3C 下辖的 HTML 工作组目前负责发布 HTML5 规范。
- **IETF(Internet Engineering Task Force, 互联网工程任务组):** 这个任务组下辖 HTTP 等负责 Internet 协议的团队。HTML5 定义的一种新 API (WebSocket API)依赖于新的 WebSocket 协议,IETF 工作组正在开发这个协议。

### 1.1.3 HTML5 的主要新功能

- 语义化标记及新的特殊内容元素,比如 article、footer、header、nav、section
- 新的表单控件
- 用于绘画的 canvas 元素
- 用于媒介回放的 video 和 audio 元素
- 对本地离线存储的更好支持
- 地理位置
- 拖曳上传
- 多线程支持

## 1.2 HTML5 的优势

### 1.2.1 解决跨浏览器问题

一份互联网上对开发者的问卷调查表明:浏览器兼容性成了最大的问题,开发者对于浏

览器扩展规范能够统一、各大厂商浏览器兼容性能提升的愿望最为迫切。在某个 Web 浏览器上可以正常运行的 HTML/CSS/JavaScript 等 Web 程序，在另一个 Web 浏览器上就运行不正常了。如果用一句话来描述这个问题的原因，可以说是“规范不统一”。规范未实现标准化，是造成这个问题的主要原因。

作为新生儿的 HTML5 在各种 Web 浏览器上的兼容性也一直是 Web 开发人员的心头大患。HTML5 的使命是详细分析各 Web 浏览器所具有的功能，然后以此为基础，要求这些浏览器的所有内部功能都要符合一个通用标准。比如，Google 分析了上百万个页面，从中分析出了 Div 标签的通用 ID 名称，发现其重复量很大，很多开发人员使用 `id=header` 来标记页眉区域，那为何不直接添加一个 `<header>` 标签呢？

如今各大浏览器厂商都支持 HTML5 标准，可以访问 <https://html5test.com/> 查看自己的浏览器对 HTML5 的支持情况。那么以 HTML5 标准为基础来书写的程序在各浏览器上都能正常运行的可能性就大大提高了，这对于 Web 开发者和 Web 设计者都是一件令人可喜的事情。而且，今后开发者开发出来的 Web 功能只要符合通用标准，Web 浏览器也都是很愿意封装该功能的。

## 1.2.2 用户优先的原则

“用户即上帝”的宗旨也体现在 HTML5 中，用户怎么使用的，就怎么设计规范。HTML5 规范是基于用户优先准则编写的，这意味着在遇到无法解决的冲突时，规范会把用户放到第一位，其次是页面作者，再次是实现者(或浏览器)，接着是规范制定者(W3C/WHATWG)，最后才考虑理论的纯粹性。因此，HTML5 的绝大部分是实用的，只是有些情况下还不够完美。

例如，下面的代码在 HTML5 中是有效的：

```
<div id=wrapper>
<img SRC=logo.png alt=logo>
```

我们不推荐这么使用 HTML5，HTML5 的适应性保证了用户体验，因为一旦由于开发人员的原因造成页面错误而导致不能正常显示，那么受折磨的肯定是最终用户。

```
<video>
<source src="movie.mp4">
<source src="movie.ogv">
<object data="movie.swf">
<a href="movie.mp4">download</a>
</object>
</video>
```

上面的代码包含了 4 个不同的层次：

- 如果浏览器支持 video 元素，也支持 MP4，那么没什么好说的，用第一个视频。
- 如果浏览器支持 video 元素，支持 ogv，那么用第二个视频。
- 如果浏览器不支持 video 元素，那么就要试试 Flash 影片了。
- 浏览器不支持 video 元素，也不支持 Flash，还给出了下载链接。

HTML5 进行了优雅的降级，保证了最终用户的体验。

### 1.2.3 化繁为简的优势

HTML5 减少了对插件的依赖，HTML5 要的就是简单，避免不必要的复杂性，化繁为简无异于提供了更好的服务。HTML5 做了以下这些改进：

- 以浏览器原生能力替代复杂的 JavaScript 代码
- 新的简化的 doctype
- 新的简化的字符集声明
- 简单而强大的 HTML5 API

随后我们将详细讲解这些改进。为了实现所有的这些简化操作，HTML5 规范需要包括一系列定义明确的行为，任何歧义和含糊都可能延缓这一目标的实现。HTML5 规范已经变得非常大而且要求精确。

## 1.3 HTML5 基本语法

本书以人人微博网站为案例，在已有的基于 HTML4 的 Web 网页的基础上，修改为基于 HTML5 的 Web 网页。在学习项目案例之前，我们应该掌握 HTML4 和 CSS 的知识，熟悉 Tomcat 的部署和启动。项目需求如下：

人人微博是一种通过关注机制分享简短实时信息的广播式的社交网络平台，功能模块包括与用户相关的登录、注册、个人资料维护功能，与关系链相关的关注功能，与微博内容相关的发布微博、浏览微博功能。

- **用户登录：**用户可以使用手机号或邮箱登录微博，如图 1-1 所示。



图 1-1 微博首页

- **用户注册：**用户可以注册账号并设置昵称和性别，如图1-2所示。



图 1-2 用户注册

- **用户主页：**用户在个人主页上可以查看自己关注的好友，自己和好友发布的微博，并且可以实时发布微博，如图1-3所示。



图 1-3 用户主页

- **用户个人资料维护：**用户可以完善和修改自己的地址、头像、生日等个人信息，如图1-4所示。



图 1-4 微博个人信息

下面我们来运行案例代码：

- (1) 请打开本书源代码 RrwbHtml4.rar。
- (2) 将其解压到 Tomcat 目录的 Webapps 目录下，这里使用 Tomcat 默认的 8080 端口。
- (3) 运行 Tomcat 目录的 bin 目录下的 startup.bat。
- (4) 在浏览器的地址栏中输入 <http://localhost:8080/rrwb/>并回车，就能看到项目案例的主页了。

### 1.3.1 新的 DOCTYPE

还记得这段 HTML4 的 doctype 吗？

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

现在，HTML5 提供了一个很好的而且便于记忆的 doctype。

```
<!DOCTYPE html>
```

简化了吧，就这么多。我们只需要使用它，就可以告诉浏览器应处于标准兼容模式下。浏览器会根据 DOCTYPE 识别应该使用哪种显示模式，以及使用什么规则来验证页面。即使某个浏览器尚未实现 HTML5，页面仍然可以继续工作。

### 1.3.2 新的字符编码

在 HTML4 中，使用 `meta` 元素的形式指定文件中的字符编码，如下所示：

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

但是，在 HTML5 中，它会像 `doctype` 一样非常简洁：

```
<meta charset="utf-8">
```

请记住，我们的页面需要字符编码和 `doctype` 声明。

### 1.3.3 引用链接和样式表

因为链接和样式表将成为我们所看到或创建的常见元素，所以这里快速介绍一下如何将它们包含于网页中，这很重要。本书稍后对 CSS 和 JavaScript 有更多的介绍。

在 HTML4 中，使用 `link` 给网页添加样式，如下所示：

```
<link rel="stylesheet" type="text/css" href="styles/changeinfo.css"/>
```

HTML5 对此进行了简化，如下所示：

```
<link rel="stylesheet" href="styles/changeinfo.css"/>
```

HTML5 向页面添加脚本则更加简单。在页面中添加一个 `<script>` 元素，然后添加 `src` 属性，指向需要使用的 JavaScript 文件的位置：

```
<script src="scripts/changeinfo.js"></script>
```

而 HTML4 添加脚本则需要指定 `type` 属性：

```
<script language="javascript" type="text/javascript" src="scripts/
changeinfo.js"></script>
```

`script` 元素新增了 `defer` 属性和 `async` 属性，正常情况下浏览器会按页面顺序加载 js 文件和渲染，`defer` 属性表示当前 js 文件会在页面渲染后加载。`async` 属性表示异步(并行)加载和执行当前 js 文件。例如：

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML5</title>
  <!-- 大小为 1.88 KB -->
  <script defer onload="alert('loaded A')" src="http://cdnjs.cloudflare.com
    /ajax/libs/html5shiv/3.6/html5shiv.min.js" ></script>
  <!-- 大小为 250 KB -->
  <script async onload="alert('loaded B')" src="http://apps.bdimg.com
    /libs/jquery/2.1.1/jquery.js" ></script>
</head>
```

```
<body>
  Hello
</body>
</html>
```

运行的结果为：先进行页面显示，后显示弹出框，经过测试弹出“loaded A”和“loaded B”的顺序不确定。这两个属性都能够实现以非阻塞的方式加载脚本，区别在于执行的时机不同。

### 1.3.4 HTML5 标签的向后兼容性

如何让 IE6~IE8 浏览器支持 HTML5 元素呢？使用一小段代码就行，将它引入页面的 head 部分的后面，看下面的例子：

```
<!--[if lt IE9]>
<script
src="http://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.6/html5shiv.min.js">
</script>
<![endif]-->
```

HTML5 shiv 的主要作用就是让 IE 正常识别 HTML5 标签，就目前而言，只要知道需要引入它即可。如果不这么做，在 IE8 和更早期版本中就会得到不可预测的结果。我们可以查阅 Paul Irish 的文章“The History of the HTML5 Shiv”来了解这段简短而重要的脚本的历史，网址是：

<http://paulirish.com/2011/the-history-of-the-html5-shiv/>

此外还有构建在 HTML5 shiv 之上的另一个库，它被称为 Modernizr。为保证向后兼容性，它的核心部分包含了 HTML5 shiv。书中的案例会使用相对较简单的 HTML5 Shiv 来确保对于 IE8 和更早版本的兼容性。

现在我们来修改人人微博网站的页面的 doctype、字符编码、链接和样式表并加入 HTML5 shiv，修改后所有的页面都有类似于下面的代码：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>人人微博</title>
<link rel="stylesheet" href="styles/index.css" />
<script src="scripts/index.js"></script>
<script src="scripts/move.js"></script>
<!--[if lt IE9]>
<script
src="http://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.6/html5shiv.min.js">
</script>
<![endif]-->
</head>
</html>
```

### 1.3.5 <base>元素

为页面上的所有链接规定默认基准 URL。浏览器随后将不再使用当前文档的 URL，而是使用通过 base 元素指定的 URL 作为基准。例如：

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML5</title>
  <base href="https://www.baidu.com/" target="_blank"/>
</head>
<body>
  <a href="img/bdlogo.png">logo</a>
</body>
</html>
```

当点击 logo 链接时，则会打开网址 <https://www.baidu.com/img/bdlogo.png>。

完成了这部分的学习后，我们已经准备好了继续学习页面结构，下面让我们学习 HTML5 的文档结构和内容分组。

## 1.4 HTML5 内容分组

在 HTML5 之前，大部分网页制作方法都是利用 class 或 id 属性来命名网站的页面结构。比如：

```
<body>
  <div id="header">
    <div id="logo"></div>
    <div id="navigation">
      <ul>
        <li><a href="#">链接</a></li>
      </ul>
    </div>
  </div>
  <div id="content"></div>
  <div id="footer"></div>
</body>
```

而 HTML5 追加了几个与页眉、页脚、内容区块等文档结构相关联的结构元素，对于微博网站来说，导航菜单、微博正文、微博的评论等每一个部分都可称为内容区块，新增的结构元素可以对内容进行更精确的分组。运用 HTML5 可以这样写：

```
<body>
  <header>
    <div id="logo"></div>
```

```
<nav>
  <ul>
    <li><a href="#">链接</a></li>
  </ul>
</nav>
</header>
<div id="content"></div>
<footer></footer>
</body>
```

HTML5 的新结构元素使得网页结构更加清晰、规范，使 HTML 更具语义化，语义化是指用合理的 HTML 标记及其特有的属性来格式化文档内容。通俗地讲，语义化就是对数据和信息进行处理，使得机器可以理解。这样做的好处有：

- **手持移动设备的无障碍阅读。**手持移动设备(如 PDA、智能手机等)可能对 CSS 的解析能力较弱，这时可能就需要更语义化的标签来体现页面的易读性。
- **盲人等一些障碍人士可以更好地阅读。**屏幕阅读器对不同标签所发出的声音是不同的，使用更语义化的标签能传达不同信息的重要性。
- **搜索引擎的友好收录。**虽然各大搜索引擎的排名规则不断地变化，但规则里的正则始终是要匹配标签的，如果能用更语义化的标签，搜索引擎就能够根据标签的使用来确定各关键字的权重。

#### 1.4.1 <header>元素

<header>元素可以用作网站的“题头”区域，也可以作为对其他内容(如<article>元素)的简要介绍。在稍后介绍的<article>元素中可以拥有它自己的题头，用来包含有关作者、发布日期以及文章标题等信息。

#### 1.4.2 <footer>元素

和<header>元素一样，它也可以视情况在同一个页面上多次出现。例如微博网站的页脚可以用它，我们在<footer>中加入团队简介、联系方式、版权信息等。

#### 1.4.3 <address>元素

<address>元素不只是用来呈现电子邮箱或真实地址，还用于标记<article>元素或整个文档的联络信息。还可用来展示与文档相关的联系人的所有联系信息。

#### 1.4.4 <article>元素

<article>元素是一个独立的内容块，它既可以独立存在，也可以被重用。可以想象一些常见的例子，比如一篇微博、一个论坛帖子、一段时尚评论、一段新闻稿或一个交互式小工具。最基本的经验是，如果一段内容可以被“聚合”，或者可以独立于网站其余环境进行分享，就应该使用<article>元素进行标记。比如一篇新闻：

```
<article>
```

```
<h1>Safari 5 released</h1>
<p>7 Jun 2010. Just after the announcement of the new iPhone 4 at WWDC,
Apple announced the release of Safari 5 for Windows and Mac.....</p>
</article>
```

### 1.4.5 <nav>元素

<nav>元素最常用的用途是作为网站的主导航。很多情况下，开发者都使用无序列表编写导航，它用于链接到网站内的其他页面，或是链接到一个页面的其他部分。HTML5 的编写方式为：

```
<nav class="header_l1ist">
<ul>
<li><a href="#">首页</a></li>
<li><a href="guangchang.html" target="_blank">广场</a></li>
<li><a href="#">微群</a></li>
<li><a href="#">应用</a></li>
<li><a href="#">游戏</a></li>
</ul>
</nav>
```

### 1.4.6 <section>元素

<section>元素定义文档中的节(区段)，比如章节、页眉、页脚或文档中的其他部分。<section>元素的使用方式和<div>元素的使用方式相似。不过，与<div>元素不一样的是，<section>元素具有语义含义，它是一组相关内容的组合。<section>元素中可以包含<article>元素，<article>元素也可以包含<section>元素，<article>元素是一个特殊的 section，它比 section 具有更明确的语义，它代表一个独立的、完整的相关内容块。一般来说，<article>元素会有标题部分(通常包含在<header>内)，有时也会包含<footer>元素。

```
<article>
  <hgroup>
    <h1>苹果</h1>
    <h2>品尝美味的苹果!</h2>
  </hgroup>
  <p>苹果的品种有很多。</p>
  <section>
    <h1>红苹果、芳香</h1>
    <p>这种红苹果在超市里非常常见。</p>
  </section>
  <section>
    <h1>绿苹果、较酸</h1>
    <p>绿苹果可以用来制作果汁。</p>
  </section>
</article>
```

### 1.4.7 <hr>元素

<hr>元素用于在页面中创建一条横跨页面的横线，分隔页面中不适于使用新标题的独立区域。HTML5 中不再支持 align、noshade、size 以及 width 属性。

### 1.4.8 <blockquote>元素

<blockquote>与</blockquote>之间的所有文本都会从常规文本中分离出来，经常会在左右两边进行缩进(增加外边距)，而且有时会使用斜体，还可以使用 cite 属性指明引用内容的来源的 HTTP 地址。

### 1.4.9 <aside>元素

<aside>元素用来表示当前页面或文章的附属信息部分，它可以包含与当前页面或主要内容相关的引用、侧边栏、广告、导航条，以及其他类似的有别于主要内容的部分。<aside>元素主要有以下两种使用方法：

- 被包含在<article>元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的相关资料、名词解释等。
- 在<article>元素之外使用作为页面或站点全局的附属信息部分。最典型的是侧边栏，其中的内容可以是友情链接、博客中的其他文章列表、广告单元等。

### 1.4.10 <hgroup>元素

<hgroup>元素用于组织具有一些逻辑联系的多级标题，例如子标题、标题信息、可选标题等内容，可以被包含在<section>、<article>等元素中。

### 1.4.11 <a>元素

HTML5 中的<a>元素可以包含段落、列表、表格等块级元素。下面的例子演示的整个广告块被包含在<a>元素中：

```
<aside class="advertising">
  <h1>广告</h1>
  <a href="">
    <section>
      <h1>岁末清仓，省时、省力!</h1>
      <p>热卖!</p>
      <p>只要 9.9 元</p>
    </section>
  </a>
  <a href="">
    <section>
      <h1>团购年货</h1>
      <p>1 元起</p>
      <p>Go!</p>
    </section>
  </a>
</aside>
```

## 1.4.12 HTML5 的文档大纲

就像 Word 中的大纲视图，HTML 也有自己的大纲。比如，新闻网站中文章的结构与报纸上文章的结构类似。每一篇文章包含标题、文本段落以及一些图片(有时可能以视频代替图片)。相同点非常明显，而唯一的不同就是，在报纸上可以在一个版面上容纳多篇故事，而在网页上每篇故事则倾向于独占一个页面。HTML4 的大纲是由<h1>、<h2>、<h3>、<h4>、<h5>、<h6>完成的，由于整个 HTML 文档是一个独立的元素，因此应该只拥有一个总体摘要，在摘要树的顶端是唯一的<h1>元素。要了解 HTML5 的文档大纲，需要知道主体结构元素和非主体结构元素的区别以及 HTML5 的新大纲算法。

### 1. 结构元素和非结构元素的区别

上面学习了 HTML5 的新元素，新增的主体结构元素包括<article>元素、<section>元素、<nav>元素及<aside>元素。主体结构元素能够表示内容区块，例如对于书籍来说，章、节都可以称为内容区块。对于微博网站来说，导航菜单、微博正文、微博的评论等每一个部分都可称为内容区块。新增的非主体结构元素包括<header>元素、<hgroup>元素、<footer>元素及<address>元素，非主体结构元素表示逻辑结构或附加信息。

### 2. HTML5 的新大纲算法

由于加入新的内容分组元素，HTML5 的文档结构比大量使用<div>元素的 HTML4 的文档结构清晰、明确了很多。在微博网站中，会有多篇包含于同一个 HTML 文档中的独立文章。这些文章都可以拥有自己的逻辑摘要，并独立于页面的其他部分。基于这个原因，HTML5 引入了一种能力，即可以在同一页面中加入多个<h1>元素，并将其用于页面中不同的分区元素内。大纲算法允许用户代理从一个 Web 页面生成一个信息结构目录，让用户对页面有一个快速概览。类似书籍、PDF、帮助文档等，都有一个清晰的目录结构，用户能方便地定位所需内容。一个结构良好的大纲，不仅是对搜索引擎的优化，更是为借助于屏幕阅读器浏览网页的失明人士或弱视力用户提供了巨大的帮助。如果想要查看这段代码的大纲，可以试试 Geoffrey Sneddon 做的大纲工具：

<https://gsnedders.html5.org/outliner/>

我们来验证如下 HTML5 代码：

```
<div>
  <h6>名著推荐</h6>
  <section>
    <h1>国内名著</h1>
    <article>
      <h1>红楼梦</h1>
      <p>红楼梦讲的是..</p>
    </article>
    <article>
      <h3>边城</h3>
```

```
<p>边城讲的是..</p>
</article>
</section>
<section>
  <h5>国外名著</h5>
  <article>
    <h6>战争与和平</h6>
    <p>战争与和平讲的是..</p>
  </article>
  <article>
    <h1>红与黑</h1>
    <p>红与黑讲的是..</p>
  </article>
</section>
</div>
```

生成的文档大纲如图 1-5 所示。

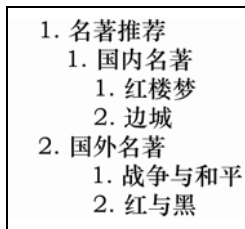


图 1-5 生成的文档大纲

为什么 h1 到 h6 的层级在这里没有表现出来？原因是此时大纲是由节点元素生成的，而非标题元素。HTML5 的新标签 section、article、aside、nav 会生成显性节点(explicit sections)，每个显性节点内部又有自己的标题结构。这也就是为什么 HTML5 允许多个 h1 存在的原因。不过，在全部浏览器、屏幕阅读器都完美支持 HTML5 之前，建议还是同时考虑标题结构，优雅降级。上面的结构可以改成这样：

```
<div>
  <h1>名著推荐</h1>
  <section>
    <h2>国内名著</h2>
    <article>
      <h3>红楼梦</h3>
      <p>红楼梦讲的是..</p>
    </article>
    <article>
      <h3>边城</h3>
      <p>边城讲的是..</p>
    </article>
  </section>
</div>
```

```

<h2>国外名著</h2>
<article>
  <h3>战争与和平</h3>
  <p>战争与和平讲的是..</p>
</article>
<article>
  <h3>红与黑</h3>
  <p>红与黑讲的是..</p>
</article>
</section>
<div>

```

我们可以得出结论：显性节点(section 等)可以包含隐性节点(h1~h6)，反之则不可以。有时会出现未命名节点(untitled sections)，比如：

```

<nav class="header_l1ist">
  <ul>
    <li><a href="#">首页</a></li>
    <li><a href="guangchang.html" target="_blank">广场</a></li>
    <li><a href="#">微群</a></li>
    <li><a href="#">应用</a></li>
    <li><a href="#">游戏</a></li>
  </ul>
</nav>
<footer>
  <h3>我是 footer 可是为什么我成为了文档标题啊 </h3>
</footer>

```

生成的文档大纲如图 1-6 所示。

```

1. 我是footer可是为什么我成为了文档标题啊
1. Untitled Section

```

图 1-6 生成的文档大纲

因为 nav 里并没有包含任何标题元素，<nav>元素会生成一个 untitled section，这不是错误，也不会被认为是好的 HTML5 结构。但是 section、article 还是建议给予适当的标题。为什么<footer>元素的标题成了文档标题呢？因为文档标题是文档中第一个非节点元素里的第一个标题元素。

### 1) 实现微博首页

掌握了 HTML5 的文档结构、结构元素之后，就可以用这些基础知识来搭建一个语义清晰、结构分明的 HTML5 网站了。下面介绍如何利用 HTML5 的各种结构元素搭建微博首页。微博首页的主体结构如图 1-7 所示。

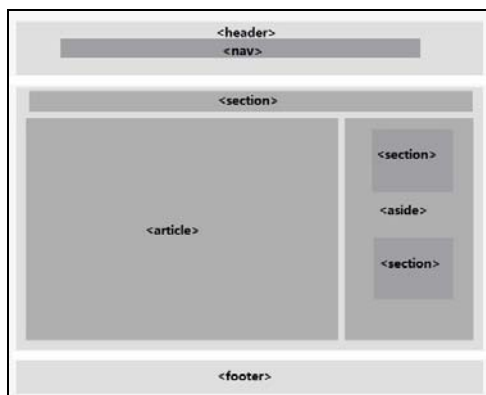


图 1-7 微博首页的主体结构

该页面主要分为 4 个部分：第 1 部分为网页头部，显示该微博网站的网站标题、网站导航链接、搜索框；第 2 部分为网页宣传栏和登录框，显示醒目的内容；第 3 部分由微博文章、达人推荐、微博应用组成，是该微博网站首页上的主要内容；第 4 部分为页面底部的版权信息和地址信息。

(1) 前面介绍过，<header>元素是一种具有引导和导航作用的结构元素，通常需要放置一个标题。在首页上，一般将标题与整个网站的导航链接作为整体网页的标题放置在<header>元素中。如前所述，<nav>元素是一个可以作为页面导航的链接组，其中的导航元素链接到其他页面或当前页面的其他部分。这里我们使用图片 logo 作为标题，并把整个网站的导航链接放在该<nav>元素中。该部分的结构代码如下所示：

```
<header id="global_header">
  <div class="header">
    <div class="header_logo"><h1> </h1> </div>
    <nav class="header_Llist">
      </nav>
  </div>
</header>
```

我们使用了一个小技巧，可以指定<img>元素的 alt 属性，并使用<h1>元素包含<img>元素，这样 alt 的属性就会作为标题显示在文档大纲中，由于<h1>元素的默认边距太宽会使得<img>元素显示在外部，我们在 index.css 中添加如下代码：

```
h1,h2,h3,h4,h5,h6{padding:0; margin:0}
```

(2) 将网页宣传栏和登录框修改为<section>元素：

```
<section class="banner">
  <div id="show"></div>
  <div id="msg"> </div>
  <div id="usrlogin"> </div>
</section>
```

(3) 实现微博网站首页的主要内容部分，左侧为微博文章列表，每一篇文章都可以包含

自己独立的内容，我们使用<article>元素。右侧是侧边栏，可以将微博达人、下载微博应用的链接等不是主体内容的其他附属内容放在<aside>元素中。由于微博达人、下载微博应用是不同的内容，可以使用<section>元素表示不同的节、不同的内容。

代码结构如下：

```
<div id="content">
<div id="saying">
<div id="sayframe">
<article id="weibo1" class="weibo"></article>
<article id="weibo2" class="weibo"></article>
<article id="weibo3" class="weibo"></article>
</div>
</div>
</div>
<aside>
<section class="fav_people"> </section>
<section> </section>
</aside>
```

(4) 实现页面底部信息，包含版权信息和联系信息，这里使用<address>元素完成联系信息。代码如下：

```
<footer id="footer">
Copyright 人人微博 All rights reserved.
<address>联系我们(QQ):12345678</address>
</footer>
```

至此，整个首页的改版就完成了，整个页面看起来和原来没什么区别，但是它已经是全新的 HTML5 页面了，也可以通过大纲工具查看微博首页的文档大纲。

## 2) 实现微博个人主页

我们再来修改微博个人主页为 HTML5 版本。该页面主要分为 4 个部分：第 1 部分为网页头部，显示该微博网站的网站标题、网站导航链接、搜索框，与微博首页是相同的；第 2 部分是左边栏，包括我的信息、每日记事、我的关注；第 3 部分包括发布微博、最新微博列表；第 4 部分为页面底部的版权信息和地址信息，与微博首页是相同的。

(1) 按照 index.html 头部代码修改微博个人主页的头部。

(2) 在左边栏中，可以用<aside>元素来显示。将我的信息、每日记事、我的关注放在侧边栏中，结构代码如下所示：

```
<aside class="cont_Left">
  <section class="cont_Leftone">
    <div class="cont_threetop">
      <h3><span class="cont_tretop1 cont_two_n">我的信息</span></h3>
    </div>
  </section>
  <section class="cont_Lefttwo">
    <div class="cont_threetop">
```

```

        <h3><span class="cont_tretop1 cont_two_n">每日记事</span></h3>
    </div>
</section>
<section class="cont_Leftthree">
    <div class="cont_threetop">
        <h3><span class="cont_tretop1 cont_two_n">我的关注</span></h3>
    </div>
</section>
</aside>

```

(3) 第 3 部分微博列表使用<article>元素来显示每一篇微博文章。下面给出部分代码:

```

<article class="cont_down4_b">
    <span class="fo">张三: </span>
    世界上最美丽的天坑，阿曼 Bimmah 天坑，当地人则说，它是月亮上掉下来的一块碎片砸进地里造成的。
    [图片]
</article>

```

(4) 按照 index.html 底部代码修改微博个人主页的底部。

## 1.5 HTML5 使用列表

### 1.5.1 使用<ul>元素创建无序列表

我们经常在网页上看到不同的文字列表和图片列表，比如图 1-8 所示的无序列表。



图 1-8 无序列表

这是使用<ul>(即“unordered list”，无序列表之意)元素创建的。在 HTML4 中，<ul>元素的 compact 和 type 属性是不推荐使用的。在 HTML5 中，不再支持这两个属性。代码如下:

```

<ul>
    <li><a href="#">热搜榜首页</a></li>
    <li><a href="#">实时热搜榜</a></li>
    <li><a href="#">综合热搜榜</a></li>
    <li><a href="#">时事热搜榜</a></li>
    <li><a href="#">影视热搜榜</a></li>

```

```
<li><a href="#">财经热搜榜</a></li>
<li><a href="#">名人热搜榜</a></li>
</ul>
```

## 1.5.2 使用<ol>元素创建有序列表

有时候我们希望给列表加上序号，以表示有序的列表，只需要将每一个列表项嵌套于<ol>元素内。比如下面的例子：

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

在 HTML4 中不赞成使用 start 和 type 属性，但是在 HTML5 中支持。我们来看下面的例子：

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

运行效果如图 1-9 所示。

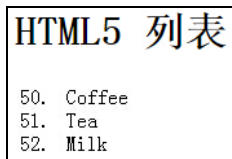


图 1-9 有序列表的运行效果

type 属性让我们可以使用数字(1、2、3)、字母(A、B、C)或罗马数字(i、ii、iii)标识序号。表 1-1 给出了可用选项。

表 1-1 textarea 元素属性

属性值	类型	描述
1	decimal	小数(默认)
a	lower-alpha	小写拉丁字母
A	upper-alpha	大写拉丁字母
i	lower-roman	小写罗马数字
I	upper-roman	大写罗马数字

HTML5 中新增了 reverse 属性，让我们可以反转有序列表的顺序。例如：

```
<ol start="4" reversed type="A">
  <li>Coffee</li>
```

```
<li>Tea</li>
<li>Milk</li>
</ol>
```

运行效果如图 1-10 所示。

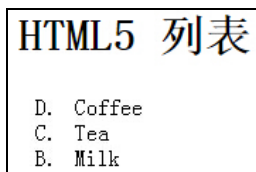


图 1-10 有序列表反转的运行效果

## 实现导航

我们可以使用列表实现顶部的导航栏，代码如下：

```
<nav class="header_l1ist">
  <ul>
    <li><a href="index.html" target="_self">首页</a></li>
    <li><a href="guangchang.html" target="_self">广场</a></li>
    <li><a href="game.html" target="_blank">游戏</a></li>
  </ul>
</nav>
```

## 1.6 HTML5 语义元素

### 1.6.1 <ruby>元素

<ruby>元素用来显示包含文字的注音。ruby 原本是印刷语言，指放于表意文字上方或右边的拼音或批注，广泛应用于日文和中文。<ruby>元素内可以包含<rt>元素，<rt>元素用来显示要加上注音标注的文字，也就是用来显示标注的注音符号，<rt>元素的内容则可放置<ruby>元素外的任何内联元素。例如：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
</head>
<body>
  <ruby>
    前端开发 <rt>qian duan kai fa</rt>
  </ruby>
</body>
</html>
```

运行效果如图 1-11 所示。

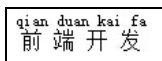


图 1-11 <ruby>元素的运行效果

## 1.6.2 <small>元素

<small>元素代表小打印体，如版权信息、条款或法律信息。缩小字体应该由 CSS 来完成。

```
<small>Copyright 人人微博 All rights reserved.</small>
```

## 1.6.3 <time>元素

<time>元素表示内容是日期类型，可以是各种格式的时间、时区偏移字符串、持续时间字符串，它有 `datetime` 属性，用来表示以计算机可识别的格式记录相同的日期和时间。例如：

```
<p> 会议开始于<time>20:00</time>./p>
```

如省略 `datetime` 属性，则<time>元素的内容必须符合 `datetime` 的有效格式，如表 1-2 所示。

表 1-2 time 的有效格式

格 式	示 例
有效的日期字符串	<time>2015-1-1</time>
有效的“月”日期字符串	<time>2015-1</time>
有效的“月日”日期字符串	<time>1-1</time>
有效的本地日期和时间字符串	<time>2015-1-1T11:11</time>
有效的时间字符串	<time>11:11</time>
有效的时区偏移字符串	<time>+0000</time> <time>-0800</time>
有效的全球日期和时间字符串	<time>2015-1-1T14:54+0000</time> <time>2015-1-1T06:54-0800</time> <time>2015-1-1T06:54:39-0800</time>
有效的周字符串	<time>2015-W1</time>
有效的非负整数年份字符串	<time>2015</time>
有效的持续时间字符串	<time>2h 3m 38s</time>
有效的日期字符串	<time>2015-1-1</time>

下面的例子以人类可读的形式展现有效日期并提供计算机可以识别的日期格式：

```
<p>我们见面的日期是 <time datetime="2009-09-23">星期二</time>.</p>  
<time datetime="2015-1-1">New Year's Day 2013</time>
```

## 1.6.4 <details>元素

<details>元素定义元素细节，用户可以查看，也可以通过点击进行隐藏。请看下面的例子：

```
<details>
<summary>HTML5</summary>
<p>用户优先</p>
<p>化繁为简</p>
<p>新的 DOCTYPE</p>
</details>
```

<summary>元素从属于<details>元素，用鼠标点击<summary>元素中的内容文字时，<details>元素中的其他所有从属元素将展开或收缩。<details>元素的 open 属性定义了 details 是否可见，添加该属性后，在画面打开时细节区域则会处于展开状态。上述代码的运行效果如图 1-12 所示。



图 1-12 <details>元素的运行效果

## 1.6.5 <progress>元素

<progress>元素表示进度条，它的属性包括 value 和 max，value 表示当前进度，max 表示需要完成的进度。value 属性和 max 属性只能指定为有效的浮点数。value 属性的值必须大于 0，且小于等于 max 属性。例如：

```
<progress value="70" max="100">70 %</progress>
```

运行效果如图 1-13 所示。



图 1-13 <progress>元素的运行效果

## 1.6.6 <datalist>元素

<datalist>元素定义选项列表。datalist 及其选项不会被显示出来，它仅是合法的输入值列表。比如：

```
<input list="cars" />
<datalist id="cars">
  <option value="BMW"/>
  <option value="Ford"/>
```

```
<option value="Volvo" />
</datalist>
```

运行效果如图 1-14 所示。

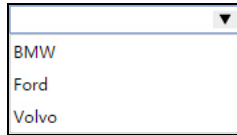


图 1-14 <datalist>元素的运行效果

### 1.6.7 <meter>元素

<meter>元素表示已知范围的度量衡，比如磁盘用量、调查结果等。常用属性有：

- **max**: 规定范围的最大值，默认是 1。
- **min**: 规定范围的最小值，默认是 0。
- **high**: 规定被视为高的值的范围。
- **low**: 规定被视为低的值的范围。
- **value**: 必需的属性，规定度量的当前值。

比如下面的例子：

```
<p><meter high=".8" value=".5"></meter></p>
<p>如果 value 的值大于 max 的值，value 的值将变为 max 的值<meter value="5"></meter></p>
<p>如果 value 的值大于 high 的值，以黄色警告显示值的度量<meter value=".9" high=".8">
  </meter></p>
```

这段代码的运行效果如图 1-15 所示。

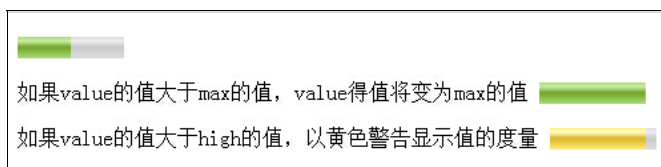


图 1-15 <meter>元素的运行效果

### 1.6.8 <mark>元素

<mark>元素表示带记号的文本(高亮显示文本)，其初衷是将注意力吸引到文档中标记的部分，类似于纸质书中使用高亮笔标注内容。比如：

```
<p>我们的任务是熟悉<mark>前端开发!</mark></p>
```

这段代码的运行效果如图 1-16 所示。

我们的任务是熟悉前端开发!

图 1-16 <mark>元素的运行效果

## 1.6.9 <figure>及<figcaption>元素

<figure>元素用来表示网页上一块独立的内容，将其从网页上移除后不会对网页上的其他内容产生任何影响。可包含插图、统计图、代码块等。它包含一个可选的<figcaption>元素，用来定义<figure>元素的标题，<figcaption>元素应该被置于<figure>元素的第一个或最后一个子元素的位置，<figure>元素内最多只允许放置一个<figcaption>元素。比如：

```
<p> 泰山日出是岱顶奇观之一，也是泰山的重要标志</p>
<figure>
  <figcaption>泰山日出</figcaption>
  
</figure>
```

这段代码的运行效果如图 1-17 所示。



图 1-17 <figure>及<figcaption>元素的运行效果

## 1.7 HTML5 多媒体

HTML5 为网页带来了全新的标准原生视频和音频渲染规范，并增加了<audio>和<video>元素。<audio>和<video>元素的出现让 HTML5 的媒体应用多了新选择，开发人员不必使用插件就能播放音频和视频。本章先介绍一些<audio>和<video>元素的基础使用方法，然后考虑一些常见的编码和跨浏览器问题。

### 1.7.1 <audio>元素

<audio>元素也是 HTML5 的新元素，允许开发人员在其网站上嵌入音乐，下面是一个<audio>示例：

```
<audio src="/test/audio.ogg">
<p>你的浏览器不支持 audio 元素.</p>
</audio>
```

万一某些用户的浏览器不支持<audio>元素，那么他们会得到一个提示。还可以将音频源文件的位置放在元素之间：

```
<audio>
<source src="music.ogg">
</audio>
```

下面我们介绍<audio>元素的属性。HTML5 中的<audio>元素和<video>元素有很多相同之处，两者都支持的操作有播放、暂停、静音/消除静音、加载等，因此通用的动作形成了<audio>元素和<video>元素的通用属性。表 1-3 给出了通用媒体属性。

表 1-3 通用媒体属性

属 性	解 释
autoplay	将媒体文件设置为创建后自动播放，或者查询是否已设置为 autoplay
loop	如果媒体文件播放完毕后能重新播放则返回 true，或者将媒体文件设置为循环播放(或不循环播放)
currentTime	以秒为单位返回从开始播放到现在所用的时间。在播放过程中，设置 currentTime 来进行搜索，并定位到媒体文件的特定位置
controls	显示或隐藏用户控制界面，或者查询用户控制界面当前是否可见
volume	在 0.0 到 1.0 之间设置音量的相对值，或者查询当前音量的相对值
muted	为音频文件设置静音或消除静音，或者检测当前是否为静音
autobuffer	通知播放器在媒体文件开始播放前，是否进行缓冲加载。如果媒体文件已设置为 autoplay，则忽略此特性
preload	如果出现该属性，则视频在页面加载时进行加载，并预备播放。如果媒体文件已设置为 autoplay，则忽略此特性

此外，还有一些只读属性，比如 duration、paused、ended、startTime 等。

### 1.7.2 <video>元素

先用一段简单的代码来显示视频：

```
<video src="http://v2v.cc/~j/theora_testsuite/320x240.ogg" controls autoplay>
  Your browser does not support the <code>video</code> element.
</video>
```

这里声明了 autoplay 属性，这样一来，页面加载完成后，视频马上会被自动播放。运行效果如图 1-18 所示。

<video>元素还有额外的属性，如表 1-4 所示。

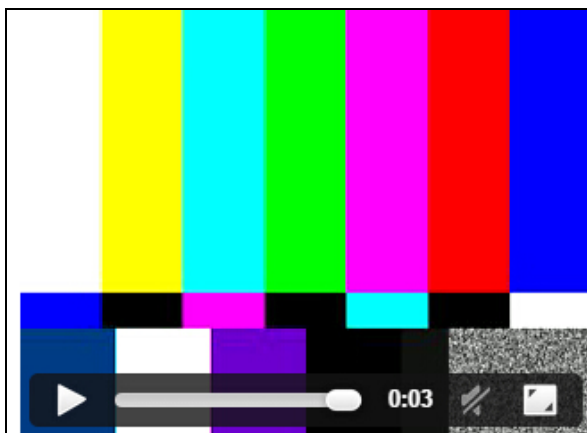


图 1-18 <video>元素的运行效果

表 1-4 <video>元素的属性

属 性	描 述
poster	在视频加载完成之前，代表视频内容的图片的 URL 地址。该特性不仅可读，而且可以修改，以便更换图片
width height	读取或设置显示尺寸。如果设置的宽度与视频本身的大小不匹配，则可能导致居中显示或出现黑色条状区域
videoWidth videoHeight	只读，返回视频的固有或自适应的宽度和高度

### 1.7.3 音频、视频容器和编解码器

HTML5 遇到的最大挑战是视频编码问题，这需要了解音频、视频容器和编解码器。主流视频容器支持如下视频格式：

- Audio Video Interleave(.avi)
- Flash video(.flv)
- MPEG 4(.mp4)
- Matroska(.mkv)
- Ogg(.ogv)

不论是音频文件还是视频文件，实际上都只是容器文件，这类似于压缩了一组文件的 ZIP 文件。视频文件(视频容器)包含音频轨道、视频轨道和其他一些元数据。视频播放时，音频轨道和视频轨道是绑定在一起的。元数据部分包含该视频的封面、标题、子标题、字幕等相关信息。

下面是一些音频编解码器：

- AAC
- MPEG-3
- Ogg Vorbis

下面是一些视频编解码器：

- MPEG-4(H.264)
- VP8(WebM)
- Ogg Theora

主要的挑战是 MPEG-4 编码，它要求任何使用该产品的厂商购买授权许可，Apple、Microsoft 支持具有专利保护的 MPET-4 格式，Google、Opera、Mozilla 支持免费、开源或者诸如 WebM 或 OGG/Theora 等版权免费的视频格式。没有任何一种编解码器可以被所有浏览器厂商接受并在其产品中提供支持，如图 1-19 所示。

BROWSER	H264	H265	VP8	VP9	AAC	MP3	VORBIS	OPUS
Chrome for Desktop	30+	-	30+	30+	30+	30+	30+	33+
Internet Explorer for Windows	9+	-	-	-	9+	9+	-	-
Firefox for Desktop	22+ <sup>1</sup>	-	20+	28+	22+ <sup>1</sup>	22+ <sup>1</sup>	20+	20+
Safari for Mac	3+	-	-	-	3+	3+	-	-
Opera for Desktop	-	-	11+	16+	-	-	11+	12+
Safari for iOS	3+	-	-	-	3+	3+	-	-
Stock Browser for Android	2.3+	-	4.0+ <sup>2</sup>	-	2.3+	2.3+	4.0+ <sup>3</sup>	-
Chrome for Android	30+	- <sup>4</sup>	30+ <sup>2</sup>	-	30+	30+	30+	-
Internet Explorer for WinPhone	7.5+	-	-	-	7.5+	7.5+	-	-
Firefox for Android	20+	-	20+ <sup>2</sup>	20+ <sup>2</sup>	20+	-	20+	20+

1. Firefox supports H264/AAC/MP3 on OSX as of version 35.  
 2. Software decoding only, impacting performance and battery life.  
 3. `canPlayType()` not reporting on Vorbis, but audio is playing  
 4. Android Lollipop contains a software decoder that makes H265 play in Chrome.

图 1-19 浏览器与视频编码

HTML5 规范本来打算指定编解码器的计划实施起来困难重重，最终还是放弃了对编解码器的要求，并支持主要的三种格式：MP4、WebM、Ogg/Ogv。我们要做的只能是熟悉各种浏览器的支持情况，针对不同的浏览器环境对媒体文件进行重新编码。这里推荐使用 `mirovideoconverter`，这是一款非常易用且完全免费的软件，它支持多种不同视频类型之间的转换，地址为 <http://www.mirovideoconverter.com/>。

我们可以在 `type` 属性中设置多媒体类型和多媒体的编码，比如：

```
<source src="foo.mp4" type="video/mp4; codecs=avc1.42E01E, mp4a.40.2">
<source src="foo.ogv" type="video/ogg; codecs=dirac, speex">
```

如果我们明确知道浏览器支持某种类型，那么明确写清楚这种类型显然更明智。否则，最好是省略 `type` 属性，让浏览器自己检测编码。因为如果 `type` 属性中指定的类型与源文件不匹配，浏览器可能会拒绝播放。

### 1.7.4 跨浏览器问题

我们肯定希望所有主流浏览器都能够播放音频或视频，我们可以准备多个视频和音频源，浏览器可以从这么多的来源中自动选择，下面通过使用多个<source>元素，在<video>元素中提供多种格式的视频文件。代码如下：

```
<video controls>
  <source src="somevideo.webm" type="video/webm">
  <source src="somevideo.mp4" type="video/mp4">
  <source src="somevideo.ogv" type="video/ogg">
</video>
```

对于不支持 HTML5 多媒体元素的旧浏览器，可以添加额外的向后兼容内容，我们可以使用 Flash 播放器提供音频或视频，比如：

```
<video src="video.ogv" controls>
  <object data="flvplayer.swf" type="application/x-shockwave-flash">
    <param value="flvplayer.swf" name="movie"/>
  </object>
</video>
```

下面我们在首页实现视频微博的功能，使用<video>元素在首页播放一段视频，代码如下：

```
<section id="play_weibo">
  <h3>视频微博</h3>
  <div id="pic_weibo">
    <video width="320" height="240" controls poster="images/pic_weibo.png">
      <source src="video/big.ogv" type="video/ogg">
      <source src="video/big.mp4" type="video/mp4">
    </video>
  </div>
  <div id="mobile"></div>
</section>
```

## 1.8 HTML5 表单介绍

HTML5 表单并没有完全修改以前的表单规范，HTML5 表单规范更加注重对现有的 HTML 表单功能进行改进，以使其包含更多控件类型。HTML5 表单仍然使用<form>元素作为容器，我们可以在其中设置基本的提交特性。之前所讲的表单控件，如文本框、单选按钮、复选框等，都兼容原来的使用方法。当用户提交页面时，表单仍然用于向服务器发送表单中控件的值，我们仍可放心地使用 JavaScript 操作表单控件或者编写自己的处理函数。

<form>元素除了带有 action、method、target 属性外，还可以带有以下属性：novalidate、autocomplete、enctype。

novalidate 属性是一个布尔值，用来指定表单在提交时是否应该进行校验。如果该属性存

在，则浏览器不会在提交表单前校验表单。例如：

```
<form action="example.jsp" novalidate>
```

`autocomplete` 属性指明浏览器是否应该自动填写表单值。将之设置为 `off` 指明浏览器不应该自动填写任何内容，默认值为 `on`。例如：

```
<form action="example.jsp" autocomplete="off" >
```

`enctype` 属性用于指定浏览器在将数据发送到服务器之前如何对其进行编码，浏览器通常支持三种类型的编码方式：`application/x-www-form-urlencoded` 作为默认值是大多数表单使用的标准方法，因为一些字符，如空格、加号以及一些非字母数字字符不能直接发送到服务器，在使用此属性值时，这些字符会被其他可用的字符替代；`multipart/form-data` 属性值允许数据分成几个部分传送，每一个连续部分以其在表单中出现的顺序对应一个表单控件，该方式通常被用于用户需要向服务器上传文件、图片时，每一个部分可以有一个属于它自己的可选的 `content-type` 头部信息，用来指定所属表单控件的数据类型；`text/plain` 以纯文本形式不加修改地将数据发送至服务器。

## 1.9 HTML5 新增的输入元素

在继续学习之前，我们看一下在人人微博网站中的哪些地方用到了表单？有注册表单、登录表单、个人信息表单、微博发表表单。可见人人微博的核心功能都是通过表单实现的，表单的创建不仅要符合用户的操作需求和业务需求，还要考虑与服务器端数据的交互。以个人信息页面(如图 1-20 所示)为例，用户的有些需求使用 HTML4 开发会很费劲而且是重复性的工作，而 HTML5 表单可以利用新的输入元素和属性满足这些需求。



The image shows a screenshot of the Weibo user profile page. The page title is "人人微博" (Weibo). The main content area is titled "基本信息 (\*为必填)" (Basic Information). Below the title, there is a sub-header "以下信息将显示在个人资料页，方便大家了解你" (The following information will be displayed on your profile page to help others get to know you). The form contains several input fields: "登录名: l@126.com" (Login name), "\*昵称:" (Nickname), "真实姓名:" (Real name), "\*所在地:" (Location) with dropdown menus for "山东" (Shandong) and "淄博" (Zibo), "\*性别:" (Gender) with radio buttons for "男" (Male) and "女" (Female), "生日:" (Date of birth) with dropdown menus for "1992" (Year), "1" (Month), and "08" (Day), "博客地址:" (Blog address), "QQ:" (QQ number), and "MSN:" (MSN address). A green "保存" (Save) button is located at the bottom of the form.

图 1-20 HTML4 个人信息页面

本节将介绍新的 HTML5 输入类型以及一些新属性，它们极大地提升了<input>元素的输入体验，特别是在移动设备上，使得用户更容易填写网页表单。旧浏览器会将任何未知类型的<input>元素按 text 类型对待。这意味着即使新输入元素和新属性不可用，我们也仍然能够收集数据。简而言之，浏览器会忽略任何它不理解的属性，因此尽管该属性可能尚未得到支持，即使用户使用了诸如 IE6 的浏览器，也仍然可以安全地使用这些属性。

HTML5 新增的专用文本输入类型包括：search、tel、url、email、number、color、range、日期和时间输入类型。

- **search**: 搜索专用字段。与所有表单元素一样，大部分浏览器只能将搜索字段显示为一个标准的文本字段。
- **tel**: 电话号码专用字段。目前，国际上没有统一的电话号码格式，因此为 tel 设置类型属性似乎不那么有用——我们必须在智能手机或平板电脑上进行测试才有效。
- **url**: 统一资源定位专用字段。它的查验的严格程度可能超出我们的想象，“www.example.com”会遭受拒绝(如图 1-21 所示)，因为它不是一个有效的 URL。它要求一个以 http://等开头的完全合格的 URL。



图 1-21 Chrome 拒绝一个未加前导 http://的 URL

- **email**: email 专用字段。我们设置输入文本类型为 email，如果输入内容包含非法字符或没有包含一个@标记，浏览器将阻止表单的提交。在移动设备上，当用户单击该字段时，移动设备将显示相应的键盘。默认情况下，它能够阻止用户在该字段中填入多个 email 地址。除非我们在<input>标签中设置如下所示的 multiple 属性，否则只允许使用一个 email 地址：

```
<input type="email" name="email" id="email" multiple>
```

- **number**: 数值专用字段。创建一个适用于数值输入的字段。最新版本的主流桌面浏览器均显示一个与图 2-22 类似的能够翻动数字的输入框。单击输入框右侧的向上箭头(或者按上方向键)能够增大字段中的数字，而单击向下箭头(或者按下方向键)能够减小字段中的数字。在移动设备上，单击该字段将显示一个数字键盘，但我们也可以切换到数字字母输入方式。

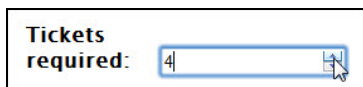


图 1-22 数字翻动输入框使得更便于输入数值

- **color**: 颜色选择器专用字段。在 Firefox、Chrome 浏览器中的显示效果如图 1-23 所示。