

疯程

HTML 5/CSS 3/ JavaScript 讲义

李刚 编著

疯程源自梦想

技术成就辉煌

疯程源自梦想
技术成就辉煌



编程

HTML 5/CSS 3/ JavaScript 讲义

李刚 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

HTML 5 规范呼之欲出, 各浏览器厂商更早已摩拳擦掌, 纷纷推出支持 HTML 5 规范的浏览器: Firefox、Opera、Chrome 等浏览器早已很好地支持各种 HTML 5 规范, 微软也宣布 IE 9 将会更好地支持 CSS 3、SVG 和 HTML 5 等新规范, 这些迹象表明: HTML 5 时代到来了。

本书是一本全面介绍 HTML 5、CSS 3 和 JavaScript 前端开发技术的图书, 系统地介绍了 HTML 5 常用的元素和属性、HTML 5 的表单元素和属性、HTML 5 的绘图支持、HTML 5 的多媒体支持、CSS 3 的功能和用法、最前沿的变形与动画功能等。除此之外, 本书还系统地介绍了 JavaScript 编程知识, 包括 JavaScript 基本语法、DOM 编程, 以及 HTML 5 新增的本地存储、离线应用、JavaScript 多线程、客户端通信支持、WebSocket 编程等。

本书的定位是一本前端开发的“实战性”图书, 因此在介绍各知识点时并不是简单地停留在知识点层面阐述, 而是结合了大量实例来让读者对照练习、学以致用。本书最后还提供了一个基于 HTML 5 技术的小游戏: 疯狂俄罗斯方块。该游戏综合运用了 HTML 5 的绘图支持、客户端存储等技术, 具有较高的参考价值。本书提供了配套的答疑网站, 如果读者在阅读本书时遇到了技术问题, 可以登录疯狂 Java 联盟 (<http://www.crazyit.org>) 发帖, 笔者将会及时予以解答。

本书对 HTML 5、CSS 3、JavaScript 的介绍是“从零开始”的, 因此阅读本书并不需要额外的基础。对于刚刚从事前端开发的新人, 本书具有很好的学习价值; 对于有一定工作经验的前端工程师, 本书具有很高的参考价值。本书也可作为高校、培训机构的培训教材。

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

疯狂 HTML 5/CSS 3/JavaScript 讲义 / 李刚编著. —北京: 电子工业出版社, 2012.5
ISBN 978-7-121-16863-5

I. ①疯… II. ①李… III. ①超文本标记语言—程序设计②网页制作工具③JAVA 语言—程序设计
IV.①TP312②TP393.092

中国版本图书馆 CIP 数据核字 (2012) 第 078047 号

策划编辑: 张月萍

责任编辑: 葛 娜

印 刷: 北京中新伟业印刷有限公司
装 订:

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: 32 字数: 819 千字 彩插: 1

印 次: 2012 年 5 月第 1 次印刷

印 数: 4000 册 定价: 69.00 元 (含光盘 1 张)

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。

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

服务热线: (010) 88258888。



前 言

HTML 5 时代即将到来；Firefox、Opera、Chrome 等浏览器早已很好地支持各种 HTML 5 规范，微软从 IE 9 开始也能支持 HTML 5、CSS 3 和 SVG 等新规范。W3C 已经把发布 HTML 5 规范纳入了议事日程。这些事实表明：HTML 5 正在向大家招手。

笔者一直在自己的电脑上安装着 5 个浏览器：Internet Explorer、Firefox、Opera、Chrome 和 Safari，每当这些浏览器有新版本面世时，笔者总会在第一时间去下载并使用它们，总希望了解到这些浏览器对 HTML 5 的支持程度。现在很欣喜地看到，绝大部分浏览器都已对 HTML 5 提供了很好的支持。

对于所有的前端开发者来说，HTML 5 与 CSS 3 是一个长久的“期待”，因为以前的前端开发太痛苦了——①HTML 控件的功能过于单一，比如一个简单的日历选择框，前端开发者必须书写大量的 JavaScript 代码才能完成（或者直接用第三方提供的日历组件）；②各浏览器厂商对 HTML、JavaScript、CSS 的支持有差异，开发人员需要花费大量时间来原因解决前端开发所遇到的跨浏览器问题……对于所有从事企业级应用开发的程序员来说，前端开发又是无法回避的，有时候前端 UI 开发的时间占到整个项目开发时间的 40%（甚至更高），HTML 5 的到来可以让前端开发者更加轻松。

本书特点



本书是一本介绍 HTML 5、CSS 3 和 JavaScript 开发的实用图书。全书可分为 4 个部分。

第一部分：全面地介绍了 HTML 5 的全部标签，并且详细地介绍了各标签所支持的属性，并为各 HTML 标签、属性都提供了配套的示例页面，这些内容不仅可以作为学习 HTML 5 的教程，也可以作为日常开发的参考手册。

第二部分：详细地介绍了 CSS 3 的绝大部分常用选择器、属性，并为这些选择器、属性提供了示例，方便广大读者参考本书全面、系统地掌握 CSS 3 的功能和用法。这部分内容也可以作为前端开发者的参考手册。

第三部分：重点介绍了 JavaScript 编程的相关内容，包括 JavaScript 基础语法、JavaScript 函数、JavaScript 对象、DOM 编程、事件处理、本地存储、离线应用、多线程、跨文档消息通信、WebSocket 编程等知识，这些内容既覆盖了初学者的编程基础，也覆盖了 HTML 5 所支持的新功能，非常适合作为前端开发者的学习教程。

第四部分：综合运用了 HTML 5 的绘图支持、客户端存储、CSS 样式、JavaScript 编程等内容，开发了一个网页版的“疯狂俄罗斯方块”。这个小游戏既可让读者巩固前面所掌握的各种知识，也可让读者将所学理论运用到实际开发中。

需要说明的是，本书只是一本介绍 HTML 5、CSS 3、JavaScript 实际开发的图书，而不是一本关于所谓“设计思想”的书，不要指望学习本书能提高你所谓的“设计思想”，所以

奉劝那些希望提高思想的读者不要阅读本书。

本书所介绍的知识都很“浅显”，只要读者愿意坐下来、静心阅读本书，并把书中所有示例循序渐进地练习一遍，本书带给你的只是9个字：“看得懂、学得会、做得出”。本书并没有堆砌一堆“深奥”的新名词、一堆“高深”的思想，本书依然保持了“疯狂 Java 体系”的一贯风格：思路清晰，语言平实，操作步骤详细。

只要读者在阅读本书时遇到知识上的问题，都可以登录疯狂 Java 联盟 (<http://www.crazyit.org>) 与广大 Java 学习者交流，笔者也会通过该平台与大家一起交流、学习。

本书具有如下几个特点。

1. 知识全面，覆盖面广

本书全面介绍了 HTML 5、CSS 3、JavaScript 的各种相关知识，包括 HTML 5 增强的表单标签、绘图支持、多媒体支持、CSS 选择器、CSS 盒模型属性、CSS 变形和动画相关属性、离线应用、客户端存储、JavaScript 多线程、跨文档消息传递、WebSocket 等内容。本书全面覆盖了 W3C 官网上已发布的 HTML 5 新规范，也覆盖了部分即将发布的新规范。

2. 内容实际，实用性强

本书并不局限于枯燥的理论介绍，而是采用了“项目驱动”的方式来讲授知识点。无论是讲解 HTML 5 标签还是 CSS 3 选择器、属性的功能，几乎每个知识点都可找到对应的参考示例。本书最后还提供了“疯狂俄罗斯方块”案例，实用性很强。

3. 讲解详细，上手容易

本书保持了“疯狂 Java 体系”的一贯风格：思路清晰，语言平实，操作步骤详细。只要认真阅读本书，把书中所有示例循序渐进地练习一遍，并把本书最后一个案例独立完成，读者就可达到企业前端开发的要求。

创作感言



又一本新书将面世，我已经退去了看到新书面世的欣喜感，反倒油然而生一种惶恐：也许因为已出版了不少图书，不再产生新鲜的惊喜；也许因为大量读者的殷切期望，深恐自己的新书不能满足所有读者的期望。只能把自己十二分的热忱、十二分的努力交出来，希望众多殷殷期望的读者不会失望。

回想这几年，每年过年都在创作图书，也许这才是一种属于我的状态，一种我喜欢的状态，一种很单纯的状态，一种与浮嚣世界保持距离的状态。当家乡的炮竹声在除夕夜此起彼伏时，当广州的人群在街市上川流不息时，我宁愿、宁愿守着自己宁静的家，听着家人温馨的声音，看着屏幕上的字一个一个地出现，这就是我的生活方式。

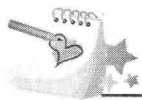
浮嚣世界，熙熙攘攘，世界是热闹的，以一己之身，行走于此滚滚尘世，往往很难心静如水。必得心无旁骛，方生冥冥之志。佛经亦云：凡所有相，皆是虚妄，若见诸相非相，则见如来，也许人生亦该如此。

本书写给谁看

本书是一本“从零学习”的 HTML 5、CSS 3、JavaScript 专业图书，阅读本书并不需要额外的基础。对于刚刚从事前端开发的新人，本书具有很好的学习价值；对于有一定工作经验的前端工程师，本书具有很高的参考价值。本书也可作为高校、培训机构的培训教材。由于本书是一本专业级的前端开发技术图书，对于那些只想简单了解 HTML、CSS 的业余人士，不推荐选择本书。



2012-3-10



光盘说明

一、光盘内容



本光盘是《疯狂 HTML 5/CSS 3/JavaScript 讲义》一书的配书光盘，书中的代码按章、按节存放，比如第 3 章第 2 节所使用的代码放在 codes 文件夹的 03\2.2 文件夹下，依此类推。

书中每份源代码也给出与光盘源文件的对应关系，方便读者查找。



本光盘 codes 目录下有 19 个文件夹，其内容和含义说明如下。

(1) 01~19 文件夹名对应于《疯狂 HTML 5/CSS 3/JavaScript 讲义》中的章名，比如第 3 章所使用的代码放在 codes 文件夹的 03 文件夹下，依此类推。

(2) 光盘中有些文件夹下包含了 .classpath、.mymetadata、.project 等文件，它们是 Eclipse 项目文件，请不要删除。

二、运行环境



本书中的程序在以下环境中调试通过。

(1) 本书很多 HTML 5、CSS 3 代码必须要在高版本的浏览器中才能看到效果。建议读者安装 Opera 11.60、Firefox 10.0、Chrome 17.0 以上版本的浏览器。

(2) 本书少量程序需要使用 Java 编译、运行，建议安装 jdk-7u3-windows-i586.exe，安装完成后，添加 CLASSPATH 环境变量，该环境变量的值为：;%JAVA_HOME%/lib/tools.jar; %JAVA_HOME%/lib/dt.jar。

为了可以编译和运行 Java 程序，还应该在 PATH 环境变量中增加 %JAVA_HOME%/bin。其中 JAVA_HOME 代表 JDK（不是 JRE）的安装路径。

(3) 第 18 章的离线 Web 应用需要 Tomcat 服务器，建议安装 Apache 的 Tomcat 7.0.20，不要使用安装文件安装，而是采用解压缩的安装方式。安装 Tomcat 需要增加 JAVA_HOME 环境变量，环境变量值为 JDK 安装路径。

三、注意事项

(1) 对于 Web 应用，将该应用复制到 %TOMCAT_HOME%/webapps 路径下。启动 Tomcat 服务器，使用浏览器即可访问该应用。

(2) 在使用本光盘中的程序时，请将程序拷贝到硬盘上，并去除文件的只读属性。

四、技术支持

如果您在使用本光盘中遇到不懂的技术问题，可以登录如下网站与作者联系：
<http://www.crazyit.org>

目 录

CONTENTS

第一部分

第 1 章 HTML 5 简介	1	2.3.3 两个特殊功能的元素	40
1.1 HTML 历史与 HTML 5	2	2.4 HTML 5 头部和元信息	41
1.1.1 HTML 发展历史	2	2.5 HTML 5 新增的拖放 API	43
1.1.2 HTML 4.01 和 XHTML	3	2.5.1 启动拖动	43
1.1.3 HTML 和 XHTML 的文档类型 定义 (DTD)	4	2.5.2 接受“放”	44
1.1.4 从 XHTML 到 HTML 5	5	2.5.3 DataTransfer 对象	46
1.2 HTML 5 的优势	6	2.5.4 拖放行为	49
1.2.1 解决跨浏览器问题	6	2.5.5 改变拖放图标	50
1.2.2 部分代替了原来的 JavaScript	7	2.6 本章小结	51
1.2.3 更明确的语义支持	7	第 3 章 HTML 5 表单相关元素和属性	52
1.2.4 增强了 Web 应用程序的功能	8	3.1 HTML 原有的表单及 表单控件	53
1.3 HTML 5 的基本结构和语法 变化	8	3.1.1 表单元素	53
1.3.1 HTML 5 基本结构	8	3.1.2 使用 input 元素	54
1.3.2 HTML 5 的语法变化	9	3.1.3 使用 label 定义标签	57
1.4 本章小结	12	3.1.4 使用 button 定义按钮	58
第 2 章 HTML 5 的常用元素与属性	13	3.1.5 列表框和下拉菜单	58
2.1 HTML 5 保留的常用元素	14	3.1.6 使用 textarea 定义文本域	60
2.1.1 基本元素	14	3.2 HTML 5 新增的属性与元素	61
2.1.2 文本格式化元素	16	3.2.1 HTML 5 为表单控件新增的 属性	61
2.1.3 语义相关元素	18	3.2.2 功能丰富的 input 元素	65
2.1.4 超链接和锚点	20	3.2.3 HTML 5 新增的表单控件	68
2.1.5 列表相关元素	22	3.3 HTML 5 增强的文件上传域	69
2.1.6 图像相关元素	24	3.3.1 FileList 对象与 File 对象	69
2.1.7 表格相关元素	25	3.3.2 使用 FileReader 读取文件内容	71
2.1.8 框架相关元素	30	3.4 HTML 5 新增的客户端校验	75
2.2 HTML 5 新增的通用属性	31	3.4.1 使用校验属性执行校验	75
2.2.1 contentEditable 属性	31	3.4.2 调用 checkValidity 方法进行 校验	76
2.2.2 designMode 属性	32	3.4.3 自定义错误提示	77
2.2.3 hidden 属性	33	3.4.4 关闭校验	78
2.2.4 spellcheck 属性	33	3.5 本章小结	79
2.3 HTML 5 新增的常用元素	34	第 4 章 HTML 5 的绘图支持	80
2.3.1 文档结构元素	34	4.1 使用 canvas 元素	81
2.3.2 语义相关元素	39		

4.2 绘图	82	4.6.1 位图裁剪	106
4.2.1 canvas 绘图基础:		4.6.2 像素处理	107
CanvasRenderingContext2D	82	4.7 输出位图	109
4.2.2 绘制几何图形	85	4.8 本章小结	111
4.2.3 绘制字符串	86	第 5 章 HTML 5 的多媒体支持	112
4.2.4 设置阴影	87	5.1 使用 audio 和 video 元素	113
4.2.5 使用路径	88	5.2 使用 JavaScript 脚本控制媒体	
4.2.6 绘制曲线	92	播放	116
4.2.7 绘制位图	94	5.2.1 HTMLAudioElement 与	
4.3 图形特效处理	95	HTMLVideoElement 支持的	
4.3.1 使用坐标变换	96	方法	116
4.3.2 坐标变换与路径结合使用	97	5.2.2 HTMLAudioElement 与	
4.3.3 使用矩阵变换	99	HTMLVideoElement 的属性	118
4.4 控制叠加风格	100	5.3 事件监听	119
4.5 控制填充风格	102	5.3.1 事件	120
4.5.1 线性渐变	102	5.3.2 监听器	121
4.5.2 圆形渐变	103	5.4 本章小结	122
4.5.3 位图填充	104		
4.6 位图处理	106		

第二部分

第 6 章 级联样式单与 CSS 选择器	123	6.4.1 内容相关的属性	142
6.1 样式单概述	124	6.4.2 插入图像	143
6.1.1 CSS 概述	124	6.4.3 只插入部分元素	144
6.1.2 CSS 的发展历史	125	6.4.4 配合 quotes 属性执行插入	145
6.2 CSS 样式单的基本使用	125	6.4.5 配合 counter-increment 属性	
6.2.1 引入外部样式文件	126	添加编号	146
6.2.2 导入外部样式单	127	6.4.6 使用自定义编号	147
6.2.3 使用内部 CSS 样式	128	6.4.7 添加多级编号	148
6.2.4 使用内联样式	129	6.5 CSS 3.0 新增的伪类选择器	150
6.3 CSS 选择器	131	6.5.1 结构性伪类选择器	150
6.3.1 元素选择器	131	6.5.2 UI 元素状态伪类选择器	158
6.3.2 属性选择器	132	6.5.3 浏览器专属的属性	162
6.3.3 ID 选择器	134	6.5.4 :not 和 :target	164
6.3.4 class 选择器	135	6.6 在脚本中修改显示样式	166
6.3.5 包含选择器	136	6.6.1 随机改变页面的背景色	166
6.3.6 子选择器	137	6.6.2 动态增加立体效果	167
6.3.7 CSS 3 新增的兄弟选择器	138	6.7 本章小结	168
6.3.8 选择器组合	139	第 7 章 字体与文本相关属性	169
6.4 伪元素选择器	140	7.1 字体相关属性	170

7.1.1 添加阴影	173	10.2 盒模型和 display 属性	225
7.1.2 添加多个阴影	174	10.2.1 两种最基本的盒模型	225
7.1.3 使用 font-size-adjust 属性微调 字体大小	175	10.2.2 none 值	227
7.2 CSS 3 支持的颜色表示方法	176	10.2.3 inline-block 盒模型	228
7.3 文本相关属性	178	10.2.4 inline-table 盒模型	230
7.3.1 文本自动换行: word-break	180	10.2.5 表格相关的盒模型	231
7.3.2 长单词和 URL 地址换行	181	10.2.6 list-item 盒模型	232
7.4 CSS 3 新增的服务器字体	182	10.2.7 run-in 盒模型	233
7.4.1 使用服务器字体	183	10.3 对盒添加阴影	235
7.4.2 定义粗体、斜体字	184	10.3.1 使用 box-shadow 属性	235
7.4.3 优先使用客户端字体	185	10.3.2 对表格及单元格添加阴影	236
7.5 本章小结	186	10.4 CSS 3 的多栏布局	237
第 8 章 背景、边框和补丁相关属性	187	10.4.1 CSS 3 新增的分栏功能	237
8.1 背景相关属性	188	10.4.2 使用盒模型实现多栏布局	240
8.1.1 背景图片固定	190	10.5 本章小结	245
8.1.2 CSS 3 新增的背景相关属性	191	第 11 章 表格、列表相关属性及 media query	246
8.1.3 CSS 3 新增的多背景图片	195	11.1 表格相关属性	247
8.2 边框相关属性	196	11.2 列表相关属性	250
8.2.1 CSS 3 提供的渐变边框	199	11.3 控制光标的属性	253
8.2.2 CSS 3 提供的圆角边框	200	11.4 media query 功能	254
8.2.3 CSS 3 提供的图片边框	202	11.4.1 media query 语法	254
8.3 补丁相关属性	204	11.4.2 针对浏览器宽度调整布局	256
8.3.1 内补丁相关属性	204	11.5 本章小结	259
8.3.2 外补丁相关属性	205	第 12 章 变形与动画相关属性	260
8.4 本章小结	207	12.1 CSS 3 提供的变形支持	261
第 9 章 大小、定位、轮廓相关属性	208	12.1.1 4 种基本变形	262
9.1 大小相关属性	209	12.1.2 同时应用多种变形	263
9.1.1 CSS 3 新增的 box-sizing 属性	210	12.1.3 指定变形中心点	265
9.1.2 CSS 3 新增的 resize 属性	212	12.1.4 使用矩阵变换	266
9.2 定位相关属性	213	12.2 CSS 3 提供的 Transition 动画	268
9.3 轮廓相关属性	215	12.2.1 多个属性同时渐变	269
9.4 本章小结	217	12.2.2 指定动画速度	271
第 10 章 盒模型与布局相关属性	218	12.3 CSS 3 提供的 Animation 动画	272
10.1 布局相关属性	219	12.3.1 同时改变多个属性的动画	274
10.1.1 通过 float 属性实现多栏布局	221	12.3.2 鱼眼效果	275
10.1.2 使用 clear 属性实现换行	222	12.4 本章小结	277
10.1.3 使用 clip 属性控制裁剪	223		
10.1.4 控制组件的滚动条	224		

第三部分

第 13 章 JavaScript 语法详解	278
13.1 JavaScript 简介	279
13.1.1 运行 JavaScript	280
13.1.2 导入 JavaScript 文件	280
13.2 数据类型和变量	280
13.2.1 定义变量的方式	281
13.2.2 类型转换	282
13.2.3 变量	283
13.3 基本数据类型	286
13.3.1 数值类型	286
13.3.2 字符串类型	291
13.3.3 布尔类型	294
13.3.4 undefined 和 null	295
13.3.5 正则表达式	296
13.4 复合类型	298
13.4.1 对象	298
13.4.2 数组	299
13.4.3 函数	300
13.5 运算符	301
13.5.1 赋值运算符	302
13.5.2 算术运算符	302
13.5.3 位运算符	304
13.5.4 加强的赋值运算符	305
13.5.5 比较运算符	306
13.5.6 逻辑运算符	307
13.5.7 三目运算符	308
13.5.8 逗号运算符	309
13.5.9 void 运算符	310
13.5.10 typeof 和 instanceof 运算符	310
13.6 语句	311
13.6.1 语句块	312
13.6.2 空语句	312
13.6.3 异常抛出语句	312
13.6.4 异常捕捉语句	314
13.6.5 with 语句	315
13.7 流程控制	315
13.7.1 分支	316
13.7.2 while 循环	318
13.7.3 do while 循环	319
13.7.4 for 循环	319
13.7.5 for in 循环	320
13.7.6 break 和 continue	321
13.8 函数	323
13.8.1 定义函数的 3 种方式	324
13.8.2 递归函数	326
13.8.3 局部变量和局部函数	327
13.8.4 函数、方法、对象和类	329
13.8.5 函数的实例属性和类属性	332
13.8.6 调用函数的 3 种方式	334
13.8.7 函数的独立性	336
13.9 函数的参数处理	337
13.9.1 基本类型和复合类型的 参数传递	337
13.9.2 空参数	339
13.9.3 参数类型	340
13.10 使用对象	342
13.10.1 面向对象的概念	342
13.10.2 对象和关联数组	343
13.10.3 继承和 prototype	344
13.11 创建对象	347
13.11.1 使用 new 关键字调用构造器 创建对象	348
13.11.2 使用 Object 直接创建对象	348
13.11.3 使用 JSON 语法创建对象	350
13.12 本章小结	353
第 14 章 DOM 编程详解	354
14.1 DOM 模型概述	355
14.2 DOM 模型和 HTML 文档	356
14.2.1 HTML 元素之间的继承 关系	356
14.2.2 HTML 元素之间常见的包含 关系	357
14.3 访问 HTML 元素	358
14.3.1 根据 ID 访问 HTML 元素	358
14.3.2 利用节点关系访问 HTML 元素	359

14.3.3	访问表单控件	361	15.1.2	绑定 DOM 对象属性	401
14.3.4	访问列表框、下拉菜单的 选项	363	15.1.3	事件处理函数和关键字 this	403
14.3.5	访问表格子元素	364	15.1.4	使用返回值改变默认行为	405
14.4	修改 HTML 元素	365	15.1.5	在代码中触发事件	406
14.5	增加 HTML 元素	367	15.2	Internet Explorer 的事件 模型	407
14.5.1	创建或复制节点	367	15.2.1	使用 script for 绑定	408
14.5.2	添加节点	369	15.2.2	使用 attachEvent 方法执行 绑定	408
14.5.3	为列表框、下拉菜单添加 选项	369	15.2.3	访问事件对象	410
14.5.4	动态添加表格内容	371	15.2.4	事件冒泡	414
14.6	删除 HTML 元素	372	15.2.5	重定向事件	416
14.6.1	删除节点	373	15.2.6	取消事件默认行为	418
14.6.2	删除列表框、下拉菜单的 选项	374	15.2.7	捕获鼠标事件	419
14.6.3	删除表格的行或单元格	375	15.3	DOM 的事件模型	421
14.7	传统的 DHTML 模型	376	15.3.1	绑定事件处理器	422
14.8	使用 window 对象	378	15.3.2	访问事件对象	423
14.8.1	访问历史	381	15.3.3	事件传播	426
14.8.2	访问页面 URL	381	15.3.4	转发事件	431
14.8.3	客户机屏幕信息	382	15.3.5	取消事件的默认行为	432
14.8.4	弹出新窗口	383	15.4	本章小结	433
14.8.5	确认对话框和输入对话框	383	第 16 章	本地存储与离线应用	435
14.8.6	使用定时器	384	16.1	Web Storage	436
14.9	navigator 和地理位置	386	16.1.1	Storage 接口	437
14.9.1	HTML 5 新增的 geolocation 属性	387	16.1.2	使用 Storage 存储、读取 数据	438
14.9.2	获取地理位置	388	16.1.3	基于 Web Storage 的客户端 留言板	439
14.9.3	在 Google 地图上定位	389	16.1.4	存储结构化数据	441
14.10	使用 document 对象	391	16.2	离线应用	444
14.10.1	动态页面	392	16.2.1	离线应用与浏览器缓存的 区别	444
14.10.2	读写 Cookie	393	16.2.2	构建离线应用	444
14.11	HTML 5 新增的浏览器 分析	394	16.2.3	判断在线状态	446
14.11.1	分析时间性能	394	16.2.4	applicationCache 对象	447
14.11.2	分析导航行为	396	16.2.5	离线应用的事件与监听	449
14.12	本章小结	397	16.3	本章小结	451
第 15 章	事件处理机制	398	第 17 章	使用 Worker 创建多线程	452
15.1	基本事件模型	399	17.1	使用 Worker 创建多线程	453
15.1.1	绑定 HTML 元素属性	399			

17.2 与 Worker 线程进行数据交换	455	18.1.1 postMessage 与 onmessage	464
17.2.1 与 Worker 线程交换数据	456	18.1.2 跨文档消息传递示例	465
17.2.2 Worker 线程中可用的 API	458	18.1.3 发送复杂消息	467
17.3 Worker 线程嵌套	458	18.2 使用 WebSocket 与服务器通信	470
17.3.1 嵌套 Worker 线程	458	18.2.1 WebSocket 接口	470
17.3.2 子线程之间的数据交换	461	18.2.2 使用 WebSocket 进行通信	472
17.4 本章小结	462	18.2.3 基于 WebSocket 的多人实时聊天	475
第 18 章 客户端通信	463	18.3 本章小结	479
18.1 跨文档消息传递	464		

第四部分

第 19 章 HTML 5 的疯狂俄罗斯方块	480	19.4 实现游戏逻辑	490
19.1 俄罗斯方块简介	481	19.4.1 处理方块掉落	490
19.2 开发游戏界面	482	19.4.2 处理方块左移	494
19.2.1 开发界面布局	482	19.4.3 处理方块右移	496
19.2.2 开发游戏界面组件	483	19.4.4 处理方块旋转	497
19.3 俄罗斯方块的数据模型	485	19.4.5 初始化游戏状态	499
19.3.1 定义数据模型	485	19.5 本章小结	500
19.3.2 初始化游戏状态数据	486		

第 1 章

HTML 5 简介

本章要点

- ▶ HTML 的发展历史
- ▶ HTML 4.01 规范与 XHTML 规范
- ▶ HTML 与 DTD 语义约束
- ▶ HTML 5 出现的缘由
- ▶ HTML 5 的强大功能
- ▶ HTML 5 更明确的语义
- ▶ HTML 5 基本结构的变化
- ▶ HTML 5 的语法变化

HTML 5 的时代已经到来，W3C 组织开始积极制订 HTML 5 规范，虽然距离 W3C 组织发布 HTML 5 规范还有一段时间，但各大浏览器厂商已经迫不及待地开始支持 HTML 5 功能了。Firefox、Opera、Chrome、Safari 各种主流浏览器都已经开始支持 HTML 5 功能。

HTML 5 时代的到来，对所有的前端开发人员来说是一种福音，HTML 5 致力于解决跨浏览器问题，可以部分取代原来的 JavaScript；HTML 5 致力于把浏览器变成一个前端程序执行环境，而不是简单的视图呈现工具。借助于 HTML 5，前端开发人员可以花费更少的时间，开发出功能更加强大的人机界面。

1.1 HTML 历史与 HTML 5

HTML 的全称是 Hyper Text Markup Language（超文本标记语言），它是互联网上应用最广泛的标记语言。不要把 HTML 语言和 Java、C 等编程语言混淆起来（把 HTML 想得很复杂），HTML 只是一种标记语言，简单地说，HTML 文件就是普通文本+HTML 标记（很多地方也称为 HTML 标签），而不同的 HTML 标记能表示不同的效果。

HTML 的发展历史“比较复杂”，因为它实在太“随意”了，而负责解析 HTML 的浏览器又太“宽容”了，以至于到了“写一份对的 HTML 文档很容易，写一份错的 HTML 文档很难”的程度。而且不同浏览器之间又存在一些差异，因此导致 HTML 给人的感觉比较混乱。

1.1.1 HTML 发展历史

从 HTML 面世开始，它就显得“很不正规”：1991 年年底推出 HTML，但最早的 HTML 并没有任何严格的定义。直到 1993 年，IETF（Internet Engineering Task Force，互联网工程工作小组）才开始发布 HTML 规范的草案。在 HTML 语言的发展历史中，大致经历了如下发展历史：

- HTML（第 1 版）：1993 年 6 月由互联网工程工作小组发布的 HTML 工作草案。
- HTML 2.0：1995 年 11 月作为 RFC 1866 发布。
- HTML 3.2：1996 年 1 月 14 日由 W3C 组织发布，是 HTML 文档第一个被广泛使用的标准。
- HTML 4.0：1997 年 12 月 18 日由 W3C 组织发布，也是 W3C 推荐标准。
- HTML 4.01：1999 年 12 月 24 日由 W3C 组织发布，是 HTML 文档另一个重要的、广泛使用的标准。
- XHTML 1.0：发布于 2000 年 1 月 26 日，是 W3C 组织推荐标准，后来经过修订于 2002 年 8 月 1 日重新发布。

在 HTML 3.2 之前，HTML 的发展极为混乱，各软件厂商经常自行增加 HTML 标记，而各浏览器厂商为了保持最好的兼容性，总是尽力支持各种 HTML 标记。在 HTML 发展历史中，最广为人知的是 HTML 3.2 和 HTML 4.01。

在早期的 HTML 发展历史中，由于 HTML 从未执行严格的规范，而且各浏览器对各种错误的 HTML 极为宽容，这就导致了 HTML 显得极为混乱。例如有如下页面。

程序清单：codes\01\1.1\qs.html

```
<ol>
  <li>疯狂 Java 讲义
  <li>轻量级 Java EE 企业应用实战
  <li>疯狂 Android 讲义
</ol>
```

对于上面的 HTML 页面代码，它并不是一份完全规范的 HTML 页面，但随便使用任何浏览器来浏览它，基本上都可以看到一个“有序列表”的效果，如图 1.1 所示。

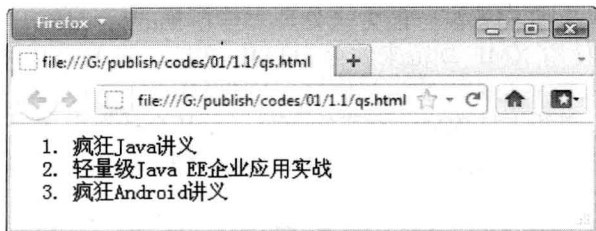


图 1.1 使用浏览器查看 HTML 文档

从图 1.1 可以看出，、等 HTML 标记在浏览器中就可以呈现出特定效果——这就是 HTML 文档的作用：通过在文本文件中嵌入 HTML 标记，这些标记告诉浏览器如何显示页面，从而使 HTML 文件呈现出更丰富的表现效果。



提示：

当修改了 HTML 文档内容后，浏览器并不会自动更新该文档的显示，我们必须用浏览器重新打开该文档，或者单击浏览器的“刷新”按钮来重新加载该文档，这样浏览器才会显示 HTML 文档的最新改变。

1.1.2 HTML 4.01 和 XHTML

XHTML 的全称是 (eXtensible Hyper Text Markup Language, 扩展的超文本标记语言)，XHTML 和 HTML 4.01 具有很好的兼容性，而且 XHTML 是更严格、更纯净的 HTML 代码。前面已经讲过了，由于 HTML 已经发展到一种极度混乱的程度，所以 W3C 组织制订了 XHTML，它的目标是逐步取代原有的 HTML。简单地说，XHTML 就是最新版本的 HTML 规范。

我们习惯上认为 HTML 也是一种结构化文档，但实际上 HTML 的语法非常自由、宽容（主要是各浏览器纵容的结果），所以才有如下 HTML 代码。

程序清单：codes\01\1.1\bad.html

```
<html>
<head>
<title>混乱的 HTML 文档</title>
<body>
<h1>混乱的 HTML 文档
```

上面代码中 4 个粗体字标签都没有正确结束，这显然违背了结构化文档的规则，但使用浏览器来浏览这份文档时，依然可以看到浏览效果——这就是 HTML 不规范的地方。而 XHTML 致力于消除这种不规范，XHTML 要求 HTML 文档首先必须是一份 XML 文档。

XML 文档是一种结构化文档，它有如下 4 条基本规则：

- 整个文档有且仅有一个根元素。
- 每个元素都由开始标签和结束标签组成（例如<a>和就是开始标签和结束标签），除非使用空元素语法（例如
就是空元素语法）。
- 元素与元素之间应该合理嵌套。例如<a>疯狂 Java 讲义，可以很明确地看出<b.../>元素是<a.../>元素的子元素，这就是合理嵌套；但<a>疯狂 Java 讲义这种写法就比较混乱，也就是所谓的不合理嵌套。

元素的属性必须有属性值，而且属性值应该用引号（单引号和双引号都可以）引起来。通常，计算机里的浏览器可以对付各种不规范的 HTML 文档，但现在很多浏览器运行在移动电话和手持设备上，它们就没有能力来处理那些糟糕的标记语言。

为此，W3C 建议使用 XML 规范来约束 HTML 文档，将 HTML 和 XML 的长处加以结合，从而得到现在和未来都能使用的标记语言：XHTML。

XHTML 可以被所有的支持 XML 的设备读取，在其余的浏览器升级至支持 XML 之前，XHTML 强制 HTML 文档具有更加良好的结构，保证这些文档可以被所有的浏览器解释。

1.1.3 HTML 和 XHTML 的文档类型定义 (DTD)

表面上看，HTML 和 XHTML 显得杂乱无章，但实际上 W3C (World Wide Web Consortium, 万维网联盟，制订 Web 标准的权威组织) 为 HTML 和 XHTML 制订了严格的语义约束。W3C 组织使用 DTD (Document Type Definition, 文档类型定义) 来定义 HTML 和 XHTML 的语义约束，包括 HTML 文档中可以出现哪些元素，各元素支持哪些属性等。

打开 HTML 4.01 的 DTD 文档 (地址: <http://www.w3.org/TR/html401/loose.dtd>)，在该文档中可以看到如下片段：

```
<!ELEMENT BODY O O (%flow;)* +(INS|DEL) -- document body -->
<!ATTLIST BODY
  %attrs;                                -- %coreattrs, %i18n, %events --
  onload      %Script; #IMPLIED -- the document has been loaded --
  onunload    %Script; #IMPLIED -- the document has been removed --
  background  %URI;    #IMPLIED -- texture tile for document
                                     background --
  %bodycolors;                            -- bgcolor, text, link, vlink, alink --
>
```

上面的 DTD 片段定义了 BODY (全部大写) 元素可以支持 %attrs 指定的各种通用属性；除此之外，BODY 元素还可以指定 onload、onunload、background、bgcolor、text、link、vlink、alink 这些属性。



提示：

对 HTML 有一定熟悉的人可能经常看到 BODY 标签 (也叫 BODY 标记) 的说法，在 HTML 语言中，经常会发生把元素称为标签的情况。但实际上按标准说法，应该称为 BODY 元素。比如上面的 DTD 片段使用了 ELEMENT 来定义 BODY 元素。

BODY 元素能接受的子元素则由 %flow 来决定，它是一个参数实体引用，这个参数实体的定义如下：

```
<!ENTITY %flow "%block; | %inline;">
```

其中 %block 也是一个参数实体引用，它代表换行的“块模型”的 HTML 元素，它的定义如下：

```
<!ENTITY %block
  "P | %heading; | %list; | %preformatted; | DL | DIV | CENTER |
  NOSCRIPT | NOFRAMES | BLOCKQUOTE | FORM | ISINDEX | HR |
  TABLE | FIELDSET | ADDRESS">
```

其中 %inline 也是一个参数实体引用，它代表不换行的“内联”HTML 元素，它的定义如下：

```
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">
```