



清华社“视频大讲堂”大系
网络开发视频大讲堂

1DVD



HTML5 + CSS3

精通

从入门到精通
配套视频讲解227节

●中小典型实例225个●综合实战案例31个●学习参考手册11部●实用网页模版83类

李东博◎编著



清华大学出版社



清华社“视频大讲堂”大系
网络开发视频大讲堂

HTML5+CSS3 从入门到精通

李东博 编著

清华大学出版社

北 京

内 容 简 介

《HTML5+CSS3 从入门到精通》(清华社“视频大讲堂”大系)通过基础知识+中小实例+综合案例的方式,讲述了用 HTML5+ CSS3 设计构建网站的必备知识,相对于权威指南、高级程序设计、开发指南同类图书,本书是一本适合快速入手的自学教程。内容有:创建 HTML5 文档,实战 HTML5 表单,实战 HTML5 绘画,HTML5 音频与视频,Web 存储,离线应用,Workers 多线程处理,Geolocation 地理位置等技术。CSS3 部分主要介绍了 CSS3 概述,CSS 选择器,文本、字体与颜色,背景和边框,2D 变形,设计动画,网页布局、用户界面以及 CSS3 的其他新特性。

《HTML5+CSS3 从入门到精通》内容涵盖了 HTML5+CSS3 的所有重要特性,通过大量实际案例对 HTML5+CSS3 的重要特性进行了详细讲解,内容全面丰富,易于理解,能够帮助读者提升实际应用技能。

本书内容翔实、结构清晰、循序渐进,基础知识与案例实战紧密结合,既可作为 HTML5+CSS3 初学者的入门教材,也适合作为中高级用户对新技术作进一步学习的参考用书。本书显著特色有:

1. 同步视频讲解,让学习更为直观高效。227 节大型高清同步视频讲解,先看视频再学习效率更高。
2. 海量精彩实例,用实例学更轻松快捷。225 个精彩实例,模仿练习是最快捷的学习方式。
3. 精选实战案例,为高薪就业牵线搭桥。31 个实战案例展示可为以后就业积累经验。
4. 完整学习套餐,为读者提供贴心服务。参考手册 11 部、网页模版 83 类、素材源程序,让学习更加方便。
5. 讲解通俗翔实,看得懂学得会才是硬道理。

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

HTML5+CSS3 从入门到精通/李东博编著. —北京:清华大学出版社,2013

(清华社“视频大讲堂”大系 网络开发视频大讲堂)

ISBN 978-7-302-30881-2

I. ①H… II. ①李… III. ①超文本标记语言-程序设计 ②网页制作工具 IV. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2012)第 291406 号

责任编辑:赵洛育

封面设计:李志伟

版式设计:文森时代

责任校对:张兴旺

责任印制:王静怡

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

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:203mm×260mm 印 张:35

字 数:983 千字

(附 1DVD,含配套视频、参考手册、网页模板、素材源程序等)

版 次:2013 年 6 月第 1 版

印 次:2013 年 6 月第 1 次印刷

印 数:1~5000

定 价:69.80 元

产品编号:044322-01



前言

Preface



互联网技术日新月异。2011 年以前,HTML5 和 CSS3 看起来还遥不可及,如今很多公司都已经开始运用这些技术了,Chrome、Safari、Firefox 和 Opera 等主流浏览器已经开始逐步实现对它们的支持。从前端开发技术看,互联网发展经历了三个阶段:第一阶段是 Web 1.0 的以内容为主的网络,主流技术是 HTML 和 CSS;第二阶段是 Web 2.0 的 Ajax 应用,热门技术是 JavaScript/DOM/异步数据请求;第三阶段是即将迎来的 HTML5+CSS3 技术,这两者相辅相成,使互联网又进入了一个崭新的时代。

HTML5+CSS3 奠定了打造下一代 Web 应用的基础,这两项技术可以让网站更易开发、更易维护、更具用户友好性。HTML5 被设计为跨平台的技术,只需要一个所有主流操作系统上都有的免费现代浏览器。最新版本的 Apple Safari、Google Chrome、Mozilla Firefox、Opera 以及 Microsoft Internet Explorer 都支持 HTML5 的许多特性。在 iPhone、iPad 及 Android 移动设备上预装的浏览器也都对 HTML5 提供了极好的支持。

本书系统地讲解了 HTML5 和 CSS3 的基础理论和实际运用技术,通过大量实例对 HTML5 和 CSS3 进行深入浅出的分析,着重讲解如何用 HTML5+CSS3 进行 Web 应用和网页布局。全书注重实际操作,使读者在学习技术的同时,掌握 Web 开发和设计的精髓,提高综合应用的能力。

本书特色

☑ 系统的基础知识

本书系统地讲解了 HTML5+CSS3 技术在网页设计中各种应用的知识,从为什么要用 HTML5 开始讲解,循序渐进,配合大量实例帮助读者奠定坚实的理论基础,做到知其所以然。

☑ 大量的案例实战

本书设置大量应用实例,重点强调具体技术的灵活应用,并且全书结合了作者长期的网页设计制作和教学经验,使读者真正做到学以致用。

☑ 深入解剖 HTML5+CSS3 应用开发和布局

本书用相当多的篇幅重点介绍了用 HTML5+CSS3 进行应用开发和布局的方法和技巧,配合经典的布局案例,帮助读者掌握 HTML5+CSS3 最核心的应用技术。

☑ 精选综合实例

本书每章都会精选多个常见 Web 开发和设计的综合实例,帮助读者总结前面所学知识,综合应用各种技术、方法和技巧,提高读者综合应用的能力。

本书内容

本书分为两大部分,共 19 章,具体结构划分如下。

第一部分:HTML5 部分,包括第 1 章~第 10 章。这部分主要介绍了 HTML5 相关基础知识,包括 HTML5 概述、Web 开发历史、HTML5 文档结构异同与创建 HTML5 文档、设计 HTML5 表单、使



用 HTML5 绘画、HTML5 音频与视频、如何实现 Web 本地存储、如何实现 Web 离线应用、如何设计多线程应用、如何实现移动互联网中地图定位等技术。

第二部分：CSS3 部分，包括第 11 章～第 19 章。这部分主要讲解了 CSS3 的新特性和新用法，以实现在简单的代码中能够设计更加精彩的网页效果。主要内容包括 CSS3 概述，CSS 选择器，定义文本、字体与颜色，设计背景和边框，使用 2D 变形，设计动画，设计多列和流动网页布局，优化用户界面以及 CSS3 的其他新特性。



Note

本书读者

- ☑ 希望系统学习网页设计、网站制作的初学者、进阶者
- ☑ 从事网页设计制作和网站建设的专业人士
- ☑ 各大中专院校相关专业的老师、学生
- ☑ 相关培训机构的学员

本书约定

本书主要面向熟悉 HTML 和 CSS 的 Web 开发人员。初学者也可以从本书获益，读者还应该具备基本的 HTML、CSS、JavaScript 知识，我们会用这些知识来创建各种应用方案。

本书中上机练习的示例要用到 Firefox、Chrome 和 Opera 浏览器，因此，为了测试所有内容，读者需要安装上述所有类型的最新版本浏览器，因为各种浏览器的实现都稍有差异。

读者还需要安装 Internet Explorer 来测试自己的站点，以便确保示例的兼容性。如需针对不同版本的 Internet Explorer 测试示例，可以下载 IE Tester Windows 版，因为它可同时支持 IE6、IE7 和 IE8。对于非 Windows 用户，可以考虑使用 VirtualBox 或者 VMware 等虚拟机，或者使用 CrossBrowserTesting 和 MogoTest 等服务。

为了给读者提供更多的学习资源，同时弥补本书篇幅有限的遗憾，本书提供了很多参考链接，许多本书中无法详细介绍的问题都可以通过这些链接找到答案。因为这些链接地址会因时间而有所变动或调整，所以在此说明，这些链接地址仅供参考，无法保证这些地址都是长期有效的。

本书所列的插图可能会与读者实际环境中的操作界面有所差别，这可能是由于操作系统平台、浏览器版本等不同而引起的，在此特别说明，读者应该以实际情况为准。

为了帮助读者快速上手，在一般情况下，读者可以在程序和文档中自由使用本书中的示例代码。

关于我们

参与本书编写的人员包括咸建勋、奚晶、文菁、李静、钟世礼、李增辉、甘桂萍、刘燕、杨凡、李爱芝、余乐、孙宝良、余洪萍、谭贞军、孙爱荣、何子夜、赵美青、牛金鑫、孙玉静、左超红、蒋学军、邓才兵、袁江和李东博等。由于作者水平有限，书中疏漏和不足之处在所难免，欢迎读者朋友不吝赐教。广大读者如有好的建议、意见，或在学习本书时遇到疑难问题，可以联系我们，我们会尽快为您解答，服务邮箱为：design1993@163.com，liulm75@163.com。

编 者

目 录

Contents



第 1 章 Web 开发新时代.....1	2.3 HTML5 基础..... 38
1.1 HTML5 概述.....2	2.3.1 HTML5 语法..... 38
1.1.1 HTML5 新特性.....2	2.3.2 HTML5 元素.....40
1.1.2 HTML5 组织.....3	2.3.3 HTML5 增加及废除的属性.....46
1.1.3 HTML5 构成.....4	2.3.4 HTML5 全局属性.....48
1.2 HTML5 设计原理.....4	2.3.5 HTML5 其他功能..... 50
1.2.1 HTML 开发历程.....5	第 3 章 创建 HTML5 文档.....54
1.2.2 HTML5 开发动力.....6	 视频讲解: 1 小时 13 分钟
1.2.3 HTML5 设计理念.....7	3.1 认识 HTML5 文档结构..... 55
1.3 编写第一个 HTML5 页面.....14	3.2 HTML5 元素分类..... 58
1.3.1 搭建上机练习环境.....14	3.3 构建主体内容..... 59
1.3.2 检测浏览器是否支持.....14	3.3.1 标识文章.....59
1.3.3 使用 HTML5 编写简单的 Web 页面.....15	3.3.2 给内容分段.....61
1.4 HTML5 页面的特征.....17	3.3.3 设计导航信息.....64
1.4.1 使用 HTML5 结构化元素.....17	3.3.4 设计辅助信息.....66
1.4.2 使用 CSS 美化 HTML5 文档.....19	3.3.5 设计微格式.....67
第 2 章 从 HTML、XHTML 到 HTML521	3.3.6 添加发布日期.....68
 视频讲解: 2 小时	3.4 添加语义模块..... 69
2.1 HTML 基础.....22	3.4.1 添加标题块.....69
2.1.1 HTML 简介.....22	3.4.2 给标题分组.....70
2.1.2 HTML 特性.....22	3.4.3 添加脚注块.....70
2.1.3 HTML 结构.....23	3.4.4 添加联系信息.....71
2.1.4 HTML 语法.....23	3.5 综合实战: 使用 HTML5 设计博客 主页..... 72
2.1.5 HTML 标签.....25	3.5.1 设计大纲.....73
2.1.6 HTML 属性.....29	3.5.2 设计样式.....76
2.2 XHTML 基础.....32	第 4 章 实战 HTML5 表单.....79
2.2.1 XHTML 结构.....32	 视频讲解: 1 小时 08 分钟
2.2.2 XHTML 语法.....33	4.1 新增的 input 输入类型..... 80
2.2.3 XHTML 类型.....34	4.1.1 email 类型的应用..... 80
2.2.4 DTD 解析.....35	4.1.2 url 类型的应用.....82
2.2.5 命名空间.....37	



Note

4.1.3	number 类型的应用	83	5.3	绘制贝塞尔曲线	117
4.1.4	range 类型的应用	84	5.3.1	绘制二次方贝塞尔曲线	118
4.1.5	日期检出器类型的应用	85	5.3.2	绘制三次方贝塞尔曲线	119
4.1.6	search 类型的应用	90	5.4	图形的变换	120
4.1.7	tel 类型的应用	91	5.4.1	保存与恢复 Canvas 状态	120
4.1.8	color 类型的应用	92	5.4.2	移动坐标空间	121
4.2	新增的 input 属性	93	5.4.3	旋转坐标空间	123
4.2.1	新增的 autocomplete 属性	93	5.4.4	缩放图形	125
4.2.2	新增的 autofocus 属性	95	5.4.5	矩阵变换	126
4.2.3	新增的 form 属性	96	5.5	图形的组合与裁切	129
4.2.4	新增的表单重写属性	97	5.5.1	图形的组合	129
4.2.5	新增的 height 与 width 属性	98	5.5.2	裁切路径	132
4.2.6	新增的 list 属性	98	5.6	更多的颜色和样式选项	133
4.2.7	新增的 min、max 和 step 属性	99	5.6.1	应用不同的线型	133
4.2.8	新增的 multiple 属性	100	5.6.2	绘制线性渐变	138
4.2.9	新增的 pattern 属性	101	5.6.3	绘制径向渐变	139
4.2.10	新增的 placeholder 属性	101	5.6.4	绘制图案	140
4.2.11	新增的 required 属性	102	5.6.5	设置图形的透明度	141
4.3	新增的 form 元素	103	5.6.6	创建阴影	142
4.3.1	新增的 datalist 元素	103	5.7	绘制文字	144
4.3.2	新增的 keygen 元素	103	5.7.1	绘制填充文字	144
4.3.3	新增的 output 元素	104	5.7.2	文字相关属性	145
4.4	新增的 form 属性	105	5.7.3	绘制轮廓文字	145
4.4.1	新增的 autocomplete 属性	105	5.7.4	测量文字宽度	146
4.4.2	新增的 novalidate 属性	105	5.8	操作与使用图像	147
第 5 章	实战 HTML5 画布	106	5.8.1	向 Canvas 中引入图像	147
	视频讲解: 2 小时		5.8.2	改变图像大小	149
5.1	认识 HTML5 canvas 元素	107	5.8.3	创建图像切片	150
5.1.1	在页面中添加 canvas 元素	107	第 6 章	HTML5 音频与视频	152
5.1.2	Canvas 如何绘制图形	108		视频讲解: 50 分钟	
5.1.3	认识 Canvas 坐标	109	6.1	HTML5 多媒体技术概述	153
5.1.4	何时不用 Canvas	109	6.1.1	关于编解码器	153
5.1.5	如果浏览器不支持 Canvas	110	6.1.2	音频编解码器	153
5.1.6	检测浏览器支持	110	6.1.3	视频编解码器	154
5.2	绘制简单图形	111	6.2	浏览器支持概述	156
5.2.1	绘制直线	111	6.2.1	用 JavaScript 检测音频格式	157
5.2.2	绘制矩形	112		支持情况	157
5.2.3	绘制圆形	113	6.2.2	用 JavaScript 检测视频格式	158
5.2.4	绘制三角形	115		支持情况	158
5.2.5	清空画布	116	6.3	在 HTML5 中播放音频	159



6.3.1 认识 audio 元素	159
6.3.2 播放音频	160
6.4 在 HTML5 中播放视频	161
6.4.1 认识 video 元素	161
6.4.2 播放视频	162
6.5 音频与视频相关属性、方法与事件	164
6.5.1 音频与视频相关属性	164
6.5.2 音频与视频相关方法	167
6.5.3 音频与视频相关事件	168
6.6 综合实战	169
6.6.1 用脚本控制音乐播放	169
6.6.2 查看视频帧画面	170
第 7 章 Web 存储	175
 视频讲解: 1 小时 20 分钟	
7.1 认识 Web Storage	176
7.1.1 Cookie 存储机制的优缺点	176
7.1.2 为什么要用 Web Storage	176
7.1.3 Web Storage 的优缺点	177
7.1.4 浏览器支持概述	177
7.2 使用 Web Storage	178
7.2.1 检查浏览器的支持性	178
7.2.2 设置和获取数据	180
7.2.3 防止数据泄露	181
7.2.4 Web Storage 的其他用法	181
7.2.5 Web Storage 事件监测	182
7.2.6 实例 1: 设计网页皮肤	183
7.2.7 实例 2: 跟踪 localStorage 数据	184
7.2.8 实例 3: 设计计数器	186
7.2.9 综合应用: Web 应用项目实时跟踪	187
7.3 Web SQL 数据库	192
7.3.1 Web SQL 数据库概述	192
7.3.2 使用 Web SQL 数据库	193
7.3.3 实例 1: 创建简单的本地数据库	195
7.3.4 实例 2: 批量存储本地数据	198
7.3.5 综合应用: Web Storage 和 Web SQL 混合开发	199

第 8 章 离线应用

视频讲解: 23 分钟

8.1 HTML5 离线应用概述	208
8.1.1 为什么要学习 HTML5 离线应用	208
8.1.2 浏览器支持概述	209
8.2 HTML5 离线应用详解	210
8.2.1 解析 manifest 文件	210
8.2.2 搭建离线应用程序	213
8.2.3 检查浏览器是否支持	213
8.2.4 离线缓存更新实现	213
8.2.5 JavaScript 接口实现	214
8.2.6 离线存储事件监听	217
8.3 实战 1: 缓存首页	218
8.4 实战 2: 离线编辑内容	221
8.5 实战 3: 离线跟踪	225

第 9 章 Workers 多线程处理

视频讲解: 1 小时 01 分钟

9.1 认识 Web Workers	232
9.1.1 Web Workers 概述	232
9.1.2 浏览器支持概述	233
9.1.3 熟悉 Web Workers 成员	233
9.2 使用 Web Workers	234
9.2.1 检查浏览器支持性	234
9.2.2 创建 Web Workers	234
9.2.3 与 Web Workers 通信	235
9.2.4 使用 Web Workers 上机练习	237
9.3 案例实战	240
9.3.1 使用多线程实现后台运算	240
9.3.2 在后台过滤值	242
9.3.3 多任务并发处理	243
9.3.4 在多线程之间通信	246
9.3.5 使用线程技术计算 Fibonacci 数列	248
9.3.6 使用多线程绘图	249
9.4 综合应用: 模拟退火算法	253
9.4.1 认识模拟退火算法	253
9.4.2 编写应用主页面	254
9.4.3 编写 worker.js	256



Note

9.4.4 与 Web Workers 通信	257	11.3 CSS3 前景展望	301
第 10 章 Geolocation 地理位置	261	11.3.1 CSS3 的应用范围	302
10.1 位置信息概述	262	11.3.2 当前支持 CSS3 的浏览器	303
10.1.1 为什么要学习 Geolocation	262	11.4 案例实战：设计漂亮的表单	305
10.1.2 位置信息表示方式	262	第 12 章 CSS 选择器	310
10.1.3 位置信息来源	262	 视频讲解：34 分钟	
10.1.4 IP 定位	263	12.1 属性选择器	311
10.1.5 GPS 定位	263	12.1.1 认识属性选择器	311
10.1.6 Wi-Fi 定位	263	12.1.2 案例实战	312
10.1.7 手机定位	264	12.2 结构伪类选择器	314
10.1.8 自定义定位	264	12.2.1 认识结构伪类选择器	314
10.2 使用 Geolocation API	264	12.2.2 案例实战	315
10.2.1 检查浏览器支持性	264	12.3 UI 伪类选择器	321
10.2.2 获取当前地理位置	265	12.3.1 认识常用 UI 伪类选择器	321
10.2.3 监视位置信息	267	12.3.2 案例实战	322
10.2.4 停止获取位置信息	267	12.4 其他选择器	324
10.2.5 隐私保护	267	第 13 章 文本、字体与颜色	330
10.2.6 处理位置信息	267	 视频讲解：49 分钟	
10.2.7 使用 position 对象	268	13.1 设计文本阴影	331
10.3 案例实战	269	13.1.1 定义 text-shadow 属性	331
10.3.1 使用 Google 地图	269	13.1.2 应用阴影效果	333
10.3.2 跟踪行走速度	271	13.1.3 综合实战：设计黑客网站 首页	339
第 11 章 CSS3 概述	277	13.2 定义文本样式	341
 视频讲解：1 小时 01 分钟		13.2.1 文本样式简介	341
11.1 回顾 CSS	278	13.2.2 溢出文本	345
11.1.1 CSS 发展简史	278	13.2.3 文本换行	347
11.1.2 CSS 1.0 和 CSS 2.0 概述	278	13.3 设计颜色样式	353
11.1.3 CSS 与 DIV 标记之缘	285	13.3.1 使用 RGBA 颜色值	353
11.1.4 CSS 编码规范	288	13.3.2 使用 HSL 颜色值	355
11.2 了解 CSS3 新增特性	289	13.3.3 使用 HSLA 颜色值	359
11.2.1 属性选择器	289	13.3.4 定义 opacity 属性	360
11.2.2 RGBA 透明度	291	13.3.5 定义 transparent 颜色值	363
11.2.3 多栏布局	292	第 14 章 背景和边框	365
11.2.4 多背景图片	294	 视频讲解：1 小时 09 分钟	
11.2.5 字符串溢出	295	14.1 设计多色边框	366
11.2.6 块阴影与圆角阴影	296	14.1.1 用法详解	366
11.2.7 圆角	297		
11.2.8 边框图片	298		
11.2.9 形变	299		



14.1.2 案例实战.....	368	16.2 3D 动画.....	428
14.2 设计边框背景.....	368	16.2.1 定义动画名称.....	429
14.2.1 用法详解.....	369	16.2.2 定义动画时间.....	429
14.2.2 案例实战.....	372	16.2.3 定义动画播放方式.....	429
14.3 设计圆角.....	375	16.2.4 定义动画延迟时间.....	429
14.3.1 用法详解.....	376	16.2.5 定义动画播放次数.....	430
14.3.2 案例实战: 设计椭圆图形.....	379	16.2.6 定义动画播放方向.....	430
14.4 设计阴影.....	380	16.2.7 案例实战: 设计图片翻转特效.....	430
14.4.1 用法详解.....	380	16.3 渐变效果.....	431
14.4.2 案例实战: 设计 Windows7 界面效果.....	385	16.3.1 设计 Webkit 渐变.....	432
14.5 设计背景.....	390	16.3.2 Webkit 案例实战.....	437
14.5.1 定义背景坐标.....	390	16.3.3 设计 Gecko 渐变.....	440
14.5.2 定义背景裁剪区域.....	392	16.3.4 Gecko 案例实战.....	446
14.5.3 定义背景图像大小.....	395	16.3.5 设计 IE 渐变.....	447
14.5.4 定义背景图像循环方式.....	396	16.3.6 设计 W3C 渐变.....	449
14.5.5 定义多背景图像.....	398	16.4 案例综合实战.....	449
第 15 章 2D 变形.....	400	16.4.1 设计礼品盒.....	450
视频讲解: 31 分钟		16.4.2 设计折叠面板.....	452
15.1 认识 transform.....	401	16.4.3 设计易拉罐.....	454
15.2 2D 变形.....	402	16.4.4 设计光盘滑动动画.....	457
15.2.1 旋转动画.....	403	16.4.5 设计下拉菜单.....	461
15.2.2 缩放动画.....	404	16.4.6 设计精致按钮.....	465
15.2.3 移动动画.....	406	第 17 章 网页布局.....	468
15.2.4 倾斜动画.....	408	视频讲解: 40 分钟	
15.2.5 变形动画.....	410	17.1 设计多列布局.....	469
15.2.6 案例实战: 设计涂鸦墙.....	412	17.2 设置多列显示样式.....	471
15.3 自定义变形.....	414	17.2.1 定义列宽.....	471
15.4 定义复杂变形.....	416	17.2.2 定义列数.....	472
第 16 章 设计动画.....	420	17.2.3 定义列间距.....	473
视频讲解: 1 小时 20 分钟		17.2.4 定义列边框样式.....	475
16.1 平滑过渡.....	421	17.2.5 定义跨列显示.....	476
16.1.1 定义过渡属性.....	421	17.2.6 定义列高度.....	478
16.1.2 定义过渡时间.....	422	17.2.7 定义打印列.....	480
16.1.3 定义过渡延迟时间.....	423	17.3 设计盒布局.....	481
16.1.4 定义过渡效果.....	424	17.4 设置盒布局格式.....	485
16.1.5 案例实战: 设计 Mac OS 导航器.....	426	17.4.1 定义自适应宽度.....	485
		17.4.2 定义列显示顺序.....	487
		17.4.3 定义列排列方向.....	489
		17.4.4 定义模块大小自适应.....	491



Note

17.4.5 消除空白	493	18.6 恢复默认样式	522
17.4.6 定义对齐方式	497	18.6.1 取消元素样式	522
17.5 综合实战: 设计多列网页	498	18.6.2 慎用 initial 的情况	524
第 18 章 用户界面	506	第 19 章 CSS3 其他新特性	527
视频讲解: 35 分钟		视频讲解: 40 分钟	
18.1 改变盒模型组成方式	507	19.1 溢出处理	528
18.2 调节元素尺寸	507	19.2 自定义字体类型	530
18.3 设计轮廓	509	19.2.1 使用 @font-face 规则	530
18.3.1 定义轮廓线	509	19.2.2 开放字体	532
18.3.2 定义轮廓线宽度	512	19.3 定义设备类型	532
18.3.3 定义轮廓线样式	512	19.3.1 认识 Media Queries 模块	532
18.3.4 定义轮廓线颜色	513	19.3.2 认识 @media 规则	533
18.3.5 定义轮廓线位移	513	19.3.3 使用 @media 规则	535
18.4 设计导航	516	19.3.4 在网站中应用 @media 规则	537
18.4.1 定义导航顺序	516	19.4 添加语音功能	543
18.4.2 定义方向键控制顺序	519	19.5 设计倒影	545
18.5 添加显示内容	521		

第1章

Web 开发新时代

从 2010 年开始, HTML5 和 CSS3 就一直是互联网技术中最受关注的两个话题。2010 年 MIX 10 大会上微软的工程师在介绍 IE9 时, 从前端技术的角度把互联网的发展分为 3 个阶段: 第一阶段是以 Web 1.0 为主的网络阶段, 前端主流技术是 HTML 和 CSS; 第二阶段是 Web 2.0 的 Ajax 应用阶段, 热门技术是 JavaScript/DOM/异步数据请求; 第三阶段是即将迎来的 HTML5+CSS3 阶段, 这两者相辅相成, 使互联网又进入了一个崭新的时代。本章将重点介绍 HTML5 语言基础, 为将来系统地学习 HTML5+CSS3 编程奠定扎实的基础。



Note

1.1 HTML5 概述

2004 年成立的 Web 超文本应用技术工作组 (WHATWG) 创立了 HTML5 规范, 同时开始专门针对 Web 应用开发新的功能。2006 年, W3C 介入 HTML5 的开发, 并于 2008 年发布了 HTML5 的工作草案。2009 年, W3C 停止了对 XHTML2 的更新。2010 年 HTML5 开始用于解决实际问题。这时各大浏览器厂商开始对旗下产品进行升级以支持 HTML5 的新功能, 因此, HTML5 规范也得到了持续性的完善。

1.1.1 HTML5 新特性

HTML5 是基于各种全新的理念进行设计的, 这些设计理念体现了对 Web 应用的可能性和可行性的新认识, 下面简单介绍 HTML5 语言的特征和优势, 以便提高读者自学 HTML5 的动力, 明确学习目标。

1. 兼容性

考虑到互联网上 HTML 文档已经存在二十多年了, 因此支持所有现存 HTML 文档是非常重要的。HTML5 不是颠覆性的革新, 它的核心理念就是要保持与过去技术的兼容和过渡。一旦浏览器不支持 HTML5 的某项功能, 针对该功能的备选行为就会悄悄运行。

2. 合理性

HTML5 新增加的元素都是对现有网页和用户习惯进行跟踪、分析和概括而推出的。例如, Google 分析了上百万的页面, 从中分析出了 DIV 标签的通用 ID 名称, 并且发现其重复量很大, 如很多开发人员使用 `<div id="header">` 来标记页眉区域, 为了解决实际问题, HTML5 就直接添加一个 `<header>` 标签。也就是说, HTML5 新增的很多元素、属性或者功能都是根据现实互联网中已经存在的各种应用进行技术精炼, 而不是在实验室中理想化地虚构新功能。

3. 效率

HTML5 规范是基于用户优先准则编写的, 其宗旨是用户即上帝, 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, 只是在有些情况下还不够完美。例如, 下面的几种代码写法在 HTML5 中都能被识别。

```
id="prohtml5"  
id=prohtml5  
ID="prohtml5"
```

当然, 上面几种写法比较混乱, 不够严谨, 但是从用户开发角度考虑, 用户不在乎代码怎么写, 根据个人习惯书写反而提高了代码编写效率。当然, 我们并不提倡初学者一开始写代码就这样随意、不严谨。

4. 安全性

为保证安全性, HTML5 规范中引入了一种新的基于来源的安全模型, 该模型不仅易用, 而且各



种不同的 API 都可通用。这个安全模型可以不需要借助于任何所谓聪明、有创意却不安全的 hack 就能跨域进行安全对话。

5. 分离

在清晰分离表现与内容方面, HTML5 迈出了很大一步。HTML5 在所有可能的地方都努力进行了分离, 包括 HTML 和 CSS。实际上, HTML5 规范已经不支持老版本 HTML 的大部分表现功能了。

6. 简化

HTML5 要的就是简单, 避免不必要的复杂性。为了尽可能简化, HTML5 做了以下改进:

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

7. 通用性

通用访问的原则可以分成 3 个概念。

- ☑ 可访问性: 出于对残障用户的考虑, HTML5 与 WAI (Web 可访问性倡议) 和 ARIA (可访问的富 Internet 应用) 做到了紧密结合, WAI-ARIA 中以屏幕阅读器为基础的元素已经被添加到 HTML 中。
- ☑ 媒体中立: 如果可能的话, HTML5 的功能在所有不同的设备和平台上应该都能正常运行。
- ☑ 支持所有语种: 如新的<ruby>元素支持在东亚页面排版中会用到的 Ruby 注释。

8. 无插件

在传统 Web 应用中, 很多功能只能通过插件或者复杂的 hack 来实现, 但在 HTML5 中提供了对这些功能的原生支持。插件的方式存在很多问题:

- ☑ 插件安装可能失败。
- ☑ 插件可以被禁用或屏蔽 (如 Flash 插件)。
- ☑ 插件自身会成为被攻击的对象。
- ☑ 插件不容易与 HTML 文档的其他部分集成, 因为存在插件边界、剪裁和透明度问题。

以 HTML5 中的 canvas 元素为例, 以前在 HTML4 的页面中较难画出对角线, 而有了 canvas 元素就可以很轻易地实现了。基于 HTML5 的各类 API 的优秀设计, 可以轻松地对它们进行组合应用。例如, 从 video 元素中抓取的帧可以显示在 canvas 中, 用户单击 canvas 即可播放这帧对应的视频文件。

1.1.2 HTML5 组织

HTML5 开发主要由下面 3 个组织负责和实施。

- ☑ WHATWG: 由来自 Apple、Mozilla、Google 和 Opera 等浏览器厂商的人员组成, 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 (因特网工程任务组): 这个任务组下辖 HTTP 等, 是负责开发 Internet 协议的团队。HTML5 定义的一种新 API (WebSocket API) 所依赖的 WebSocket 协议, 正由 IETF 工作组负责开发。



Note



1.1.3 HTML5 构成

HTML5 主要包括下面这些功能。

- ☒ Canvas (2D 和 3D)
- ☒ Channel 消息传送
- ☒ Cross-document 消息传送
- ☒ Geolocation
- ☒ MathML
- ☒ Microdata
- ☒ Server-Sent Events
- ☒ Scalable Vector Graphics (SVG)
- ☒ WebSocket API 及协议
- ☒ Web Origin Concept
- ☒ Web Storage
- ☒ Web SQL database
- ☒ Web Workers
- ☒ XMLHttpRequest Level 2

HTML5 发展的速度非常快, 因此不用担心浏览器的支持问题。读者可以访问 www.caniuse.com, 该网站按照浏览器的版本提供了详尽的 HTML5 的功能支持情况。如果通过浏览器访问 www.html5test.com, 该网站会直接显示用户浏览器对 HTML5 规范的支持情况。另外, 还可以使用 Modernizr (JavaScript 库) 进行特性检测, 它提供了非常先进的 HTML5 和 CSS3 检测功能。建议读者使用 Modernizr 检测当前浏览器是否支持某些特性。

1.2 HTML5 设计原理

设计原理是 Web 发展背后的驱动力, 也是通过 HTML5 反映出来的某种思维方式。软件就像所有技术一样, 具有天然的独裁性。代码必然会反映作者的选择、偏见和期望。任何开放的标准, 都应该追求以下几点:

- ☒ 简化最常见的任务, 让不常见的任务不至于太麻烦。
- ☒ 只为 80% 设计。
- ☒ 给内容创建者最大的权力。
- ☒ 默认设置智能化。

在制定设计原理时, 很多人用了很多时间却抓不住重点, 因为他们想取悦所有人。关键在于我们不是要取悦所有人, 而是要明确哪些人是最重要的。

意大利经济学家帕累托提出“世界上 20% 的人口拥有 80% 的财富”。这个比例又暗合了自然界各个领域的幂律分布现象。总之, 无论是编写软件, 还是制造什么东西, 都是一样的, 即 20% 的努力可以触及 80% 的用例, 最后 20% 的用例则需要付出 80% 甚至更多的努力。因此, 有时候据此确定只为 80% 设计是很合理的, 因为我们知道为此只要付出 20% 的努力即可。



1.2.1 HTML 开发历程

HTML 最早是从 2.0 版开始的, 实际上并没有 HTML 1.0 版官方规范。HTML Tags 文档可以算作 HTML 的第一个版本, 但它却不是一个正式的版本。第一个正式版本 HTML 2.0 也不是出自 W3C 之手, 而是由 IETF (Internet Engineering Task Force, 因特网工程任务组) 制定的, 从第三个版本开始, W3C (World Wide Web Consortium, 万维网联盟) 开始接手并负责后续版本的制定工作。

20 世纪 90 年代, HTML 有过几次快速的发展。众所周知, 那时构建网站是一项十分复杂的工程, 浏览器大战曾令人头疼不已, 市场竞争的结果就是各家浏览器里都塞满了各种专有的特性, 都试图在专有特性上胜人一筹。当时的混乱程度不堪回首, HTML 还重不重要, 或者它作为 Web 格式的前景如何, 谁都说不清楚。

从 1997 年到 1999 年, HTML 的版本从 3.2 到 4.0, 再到 4.01, 经历了非常快的发展。问题是到了 4.01 的时候, W3C 的认识发生了倒退, W3C 并没有停止开发这门语言, 只不过他们对 HTML 不再感兴趣了。在 HTML 4.01 之后, W3C 提出了 XHTML 1.0 的概念。虽然听起来完全不同, 但 XHTML 1.0 与 HTML 4.01 其实是一样的。

这两个规范的内容是一样的, 词汇表是一样的, 所有的元素是一样, 所有的属性也都是一样的, 唯一不同的就是 XHTML 1.0 要求使用 XML 语法。也就是说, 所有属性都必须使用小写字母, 所有元素也必须使用小写字母, 所有属性值都必须加引号, 所有的标签都必须有结束标签, 对 `img` 和 `br` 等孤标签, 要使用自结束标签。

从规范本身的内容来看, 本质是相同的, 不同之处就是编码风格, 因为浏览器读取符合 HTML 4.01、HTML 3.2 或者 XHTML 1.0 规范的网页都没有问题, 对浏览器来说这些网页都是一样的, 都会生成相同的 DOM 树, 只不过用户更喜欢 XHTML 1.0, 因为不少人认同它比较严格的编码风格。

到了 2000 年, Web 标准项目 (Web Standards Project) 的活动开展得如火如荼, 开发人员对浏览器里包含的那些乱七八糟的专有特性已经忍无可忍了。当时 CSS 有了长足的发展, 而且与 XHTML 1.0 的结合也很紧密, CSS+XHTML 1.0 可以算是最佳实践了。虽然 HTML 4.01 与 XHTML 1.0 没有本质上的区别, 但是大部分开发人员接受了这种组合。专业的开发人员能做到元素全部小写, 属性全部小写, 属性值也全部加引号。由于专业人员起到了模范带头作用, 越来越多的人也都开始支持这种语法。

XHTML 1.0 之后是 XHTML 1.1, 只是小数点后面的数字变成了 1, 而且从词汇表的角度看, 规范本身没有什么新内容, 元素、属性也都相同, 唯一的变化就是必须把文档标记为 XML 文档。而在使用 XHTML 1.0 的时候, 还可以把文档标记为 HTML。

但是, 这样做就带来很多问题: 首先, 把文档标记为 XML 后, IE 浏览器不能处理。当然, IE9 及其以上版本是可以处理的。作为全球领先的浏览器, IE 无法处理接收到的 XML 文档类型的文档, 而规范又要求以 XML 文档类型来发送文档, 这对于广大用户来说, 是一件很痛苦的事情。

所以说 XHTML 1.1 有点脱离现实, 而用户不想把文档以 XML 格式发送给那些能够理解 XML 的浏览器, 则是因为 XML 的错误处理模型。XML 的语法, 无论是属性小写, 元素小写, 还是始终要给属性值加引号, 这些都没有问题, 但 XML 的错误处理模型却是这样的: 如果解析器遇到错误, 停止解析。如果把 XHTML 1.1 标记为 XML 文档类型, 假设用 Firefox 打开这个文档, 而文档中有一个符号没有正确编码, 就算整个页面中只有这一处错误, 浏览器也会死掉, 用户将看不到任何网页内容。根据 XML 规范, 这样处理是正确的, 对 Firefox 而言, 遇到错误就停止解析, 并且不呈现其他任何内容是严格按照 XML 规范处理的。因为它不是 HTML, HTML 根本没有错误处理模型, 但根据 XML 规范, 这样做没错。这就是为什么人们不会把文档标记为 XML 的另一个原因。



接下来,新的版本是 XHTML2,但是这个版本并没有完成。从理论的角度来说,XHTML2 是一个非常好的规范。如果所有人都同意使用的话,也一定是一个非常好的格式。只不过,它还不够实际。

首先,XHTML2 仍然使用 XML 错误处理模型,用户必须保证以 XML 文档类型发送文档;其次,XHTML2 中有意不再向后兼容已有的 HTML 的各个版本,甚至曾经讨论过废除 `img` 元素,这对每天都在做 Web 开发的人来说确实有点难以接受,理论上分析,使用 `object` 元素可能会更好。

因此,无论 XHTML2 在理论上是多么完美的一种格式,却从未有机会付诸实践。之所以难以将其付诸实践,就是因为开发人员永远不会支持它,它不向后兼容。同样,浏览器厂商也不会支持它,必须要保证浏览器向后兼容。

为什么 XHTML 1.1 没有像 XML 那样得到真正广泛地应用?为什么 XHTML2 从未落到实处?因为它违反了一条设计原理,这条设计原理就是著名的伯斯塔法则:发送时要保守,接收时要开放。

接收的时候要开放,而这也正是 Web 得以构建的基础。开发浏览器的人必须敞开胸怀,接收所有发送给浏览器的东西,因为它们过去一直都在接收那些不够标准的东西,Web 上的很多文档都不规范,但那正是 Web 发展的动力。从某种角度讲,Web 走的正是一条混沌发展之路,虽然混沌,却非常美丽诱人。在 Web 上,格式不规范的文档随处可见,如果所有人都能够写出精准的 XML,所有文档的格式都十分正确,那当然好,可是那不现实。

作为专业人士,在发送文档的时候应该保守一些,尽量采用最佳实践,尽量确保文档格式良好,但从浏览器的角度说,它们必须以开放的姿态去接收任何文档。

XHTML 1.1 和 XHTML2 都使用 XML 错误处理模型,但这个错误处理模型太苛刻了,它不符合“接收时开放”这个法则,遇到一个错误就停止解析怎么能叫开放呢?我们只能说它与伯斯塔法则是对立的。

1.2.2 HTML5 开发动力

在 20 世纪末期,W3C 琢磨着改良 HTML 语言,“HTML 也许还可以更长寿一点,只要把我们放在 XHTML 上的时间和精力拿出一部分来,就可以提升一下 HTML 中的表单,可以让 HTML 更接近编程语言,就可以让它更上一层楼。”

于是,在 2004 年 W3C 成员内部的一次研讨会上,Opera 公司的代表伊恩·希克森(Ian Hickson)提出了一个扩展和改进 HTML 的建议。他建议新任务组可以跟 XHTML2 并行,但是在已有 HTML 的基础上开展工作,目标是对 HTML 进行扩展。但是 W3C 投票表示反对,因为 HTML 已经死了,XHTML2 才是未来的方向。然后,Opera、Apple 等浏览器厂商以及其他一些成员脱离了 W3C,他们成立了 WHATWG (Web Hypertext Applications Technology Working Group, Web 超文本应用技术工作组),这就为 HTML5 将来的命运埋下了伏笔。

WHATWG 决定完全脱离 W3C,在 HTML 的基础上开展工作,向其中添加一些新东西。这个工作组的成员里有浏览器厂商,因此他们可以保证实现各种新奇、实用的点子。结果,大家不断提出一些好点子,并且逐一做到了浏览器中。

WHATWG 的工作效率很高,不久就初见成效。在此期间,W3C 的 XHTML2 没有什么实质性的进展。在 2006 年,蒂姆·伯纳斯·李(Tim Berners-Lee)写了一篇博客反思 HTML 的发展历史,“你们知道吗?我们错了。我们错在企图一夜之间就让 Web 跨入 XML 时代,我们的想法太不切实际了,是的,也许我们应该重新组建 HTML 工作组了。”

W3C 在 2007 年组建了 HTML5 工作组。这个工作组面临的第一个问题,毫无疑问就是“我们是从头开始做起呢,还是在 2004 年成立的那个叫 WHATWG 的工作组既有成果的基础上开始工作呢?”



Note