

HTML5+CSS3

网页设计基础教程

- ◆ Web开发新时代
- ◆ HTML、XHTML、HTML5发展历程
- ◆ Web网站、移动网站、移动APP开发
- ◆ 表单、图形/图像、音频与视频
- ◆ 本地存储、离线应用开发
- ◆ 多线程、地理位置
- ◆ CSS3选择器、文本及修饰
- ◆ 背景和边框处理
- ◆ 变形处理、设计动画
- ◆ 网页布局



石磊 王维哲 主编
李娜 谢昆鹏 王鹏程 副主编

清华大学出版社

高等学校计算机应用规划教材

HTML5+CSS3 网页设计基础教程

石磊 王维哲 主 编

李娜 谢昆鹏 王鹏程 副主编



清华大学出版社

北 京

内 容 简 介

本书全面讲述了 HTML5+CSS3 网页设计基础知识体系。全书共分为 18 章, 主要内容包括: Web 开发概述、HTML/XHTML/HTML5 发展历程、HTML5 文档的创建、HTML5 表单的使用、图形/图像的绘制、音频与视频的播放与控制、本地存储体系、离线应用开发、Web Workers 多线程处理、CSS3 选择器、文本及修饰、背景和边框处理、变形与动画、网页布局等, 并且运用大量实例对各种关键技术进行深入浅出的分析。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实。本书面向期望学习 HTML 和 CSS 的 Web 开发人员, 适合作为高等院校相关专业的教材, 也适合从事网页设计制作和网站建设的人员学习。

本书的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

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 网页设计基础教程 / 石磊, 王维哲 主编. —北京: 清华大学出版社, 2018
(高等学校计算机应用规划教材)
ISBN 978-302-49091-3

I. ①H… II. ①石… ②王… III. ①超文本标记语言—程序设计—高等学校—教材 ②网页制作工具—高等学校—教材 ③JAVA 语言—程序设计—高等学校—教材 IV. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2017)第 300287 号

责任编辑: 胡辰浩 李维杰
封面设计: 孔祥峰
版式设计: 思创景点
责任校对: 牛艳敏
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

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 23.5 字 数: 587 千字

版 次: 2018 年 1 月第 1 版

印 次: 2018 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 58.00 元

产品编号: 073574-01

前 言

互联网技术日新月异。2011 年以前，HTML5 和 CSS3 看起来还遥不可及，如今对跨平台开发、离线 Web 应用的需求，使得 HTML5 取代 HTML4 技术从而成为 Web 前端主流技术，很多公司都已经开始运用 HTML5、CSS3 技术作为 Web 前端开发的主要工具，诸如 Chrome、Safari、Firefox 和 Opera 等主流浏览器已逐步完善对它们的支持。

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，使互联网又进入一个崭新的时代。

HTML5+CSS3 奠定了打造下一代 Web 应用的基础。HTML5 被设计为跨平台的技术，只需要一款主流浏览器即可运行，比如 PC 机上最新版本的 Apple Safari、Google Chrome、Mozilla Firefox、Opera、Microsoft Internet Explorer 等浏览器，都几乎完全支持 HTML5 的特性；iPhone、iPad 及 Android 等移动设备上的浏览器对 HTML5 的支持性也非常好。

本书系统地讲解 HTML5 和 CSS3 的基础理论和实际运用技术，书中辅以大量的实例进行讲解，其中着重讲解如何使用 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，提高综合应用能力。

本书特色

- 系统的基础知识

本书由浅入深地系统讲解 HTML5+CSS3 的基础知识及使用场景，从技术的产生原因、变迁，再到 HTML5 中的表现、使用场景、使用方法，循序渐进地进行介绍，并配合精选案例辅助读者加强对知识点的理解。

- 大量精选案例

本书为了向读者清晰地传达每一个知识点的含义，精选了大量案例，并且全书结合实际项目制作的需求来进行讲解，使读者能够真正做到学以致用。

- 深入剖析应用开发和布局

本书在介绍 CSS3 的时候，使用了相当大的篇幅重点介绍应用布局的方法和技巧，并配合经典布局案例来帮助读者理解项目中常用的布局技术，以做到活学活用。

本书从内容上可分为三部分，共 18 章，具体结构划分如下：

第一部分：HTML5 部分，包括第 1 章~第 10 章。这部分全面讲述 HTML5 和 CSS3 网页设计基础知识体系，主要内容包括：Web 开发概述、HTML/XHTML/HTML5 发展历程、HTML5 文档的创建、HTML5 表单的使用、图形/图像的绘制、音频和视频的播放与控制、

本地存储体系概述与应用、离线应用程序开发、Web Workers 多线程处理、移动互联网中的地图定位等技术。

第二部分：CSS3 部分，包括第 11 章～第 17 章。这部分主要讲解 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，设计多栏布局、盒子布局和弹性盒布局等知识。

第三部分为第 18 章。第 18 章是综合实例，通过典型的企业网站建设流程和手机阅读器的开发，使读者能够熟悉实际项目的开发流程，掌握系统使用 HTML5+CSS3 技术来开发项目的方法。

本教程内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实。每一章的引言部分概述该章的内容及学习目标。在每一章的正文中，结合所讲述的关键技术和难点，穿插了大量极富实用价值的示例。每一章末尾都安排了有针对性的思考题和练习题，思考题有助于读者巩固所学的基本概念，练习题有助于培养读者的实际动手能力、增强对基本概念的理解和实际应用能力。

本书面向期望学习 HTML 和 CSS 的 Web 开发人员，适合作为高等院校相关专业的教材，也适合从事网页设计制作和网站建设的人员学习。

除封面署名的作者外，参加本书编写的人员还有周爱萍、屈文斌、万鑫、张春辉、梅泉滔、杨永好、郑梦成、孙红胜、何玉华、李文静、冯波、马协隆、马金帅、张晓晗、张梦甜、李亮等。由于作者水平有限，本书难免有不足之处，欢迎广大读者批评指正。我们的信箱是 huchenhao@263.net，电话是 010-62796045。

本书的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

作 者

2017 年 12 月

目 录

第 1 章 Web 开发新时代	1	2.3.1 HTML5 语法	21
1.1 HTML5 概述	1	2.3.2 HTML5 元素	22
1.1.1 HTML5 的目标	1	2.4 新增和废除的属性	25
1.1.2 HTML5 新特性	2	2.4.1 新增的属性	25
1.1.3 HTML5 深受欢迎的原因	3	2.4.2 废除的属性	27
1.1.4 HTML5 的构成	4	2.5 全局属性	28
1.2 HTML5 设计原理	5	2.5.1 contentEditable 属性	28
1.2.1 HTML 的历史变迁	5	2.5.2 designMode 属性	29
1.2.2 HTML5 开发动力	6	2.5.3 hidden 属性	29
1.3 编写第一个 HTML5 页面	7	2.5.4 spellcheck 属性	29
1.3.1 搭建上机练习环境	7	2.5.5 tabIndex 属性	29
1.3.2 检测浏览器是否支持	7	2.6 新增的事件	29
1.3.3 使用 HTML5 编写简单的 Web 页面	8	2.7 本章小结	30
1.4 HTML5 页面的特征	9	2.8 思考和练习	30
1.4.1 使用 HTML5 的结构化元素	9	第 3 章 创建 HTML5 文档	31
1.4.2 使用 CSS 美化 HTML5 文档	12	3.1 认识 HTML5 文档结构	31
1.5 本章小结	13	3.2 HTML5 元素分类	33
1.6 思考和练习	13	3.3 构建主体内容	34
第 2 章 HTML、XHTML、HTML5	14	3.3.1 标识文章: article 元素	34
2.1 HTML 基础	14	3.3.2 给内容分块: section 元素	36
2.1.1 HTML 简介	14	3.3.3 设计导航信息: nav 元素	37
2.1.2 HTML 结构	15	3.3.4 设计辅助信息: aside 元素	39
2.1.3 HTML 语法	15	3.3.5 设计微格式: time 元素	40
2.2 XHTML 基础	17	3.3.6 添加发布日期: pubdate 属性	41
2.2.1 XHTML 结构	17	3.4 添加语义模块	41
2.2.2 XHTML 语法	18	3.4.1 添加标题块: header 元素	41
2.2.3 XHTML 类型	18	3.4.2 给标题分组: hgroup 元素	42
2.2.4 DTD 解析	19	3.4.3 添加脚注块: footer 元素	43
2.2.5 命名空间	21	3.4.4 添加联系信息: address 元素	44
2.3 HTML5 基础	21	3.5 本章小结	44
		3.6 思考和练习	45

第4章 HTML5 表单	46	4.7 思考和练习	69
4.1 新增的表单属性	46	第5章 图形/图像的绘制	70
4.1.1 autocomplete 属性	46	5.1 canvas 元素基础	70
4.1.2 novalidate 属性	47	5.1.1 添加 canvas 元素	70
4.2 新增的表单元素	47	5.1.2 检测浏览器是否支持	71
4.2.1 datalist 元素	47	5.1.3 使用 canvas 元素绘制图形	71
4.2.2 keygen 元素	48	5.1.4 canvas 坐标系	73
4.2.3 output 元素	49	5.2 绘制简单图形	73
4.3 新增的输入类型	49	5.2.1 绘制直线	73
4.3.1 email 类型	50	5.2.2 绘制矩形	74
4.3.2 url 类型	51	5.2.3 绘制弧线与圆形	75
4.3.3 number 类型	51	5.2.4 绘制三角形	78
4.3.4 Date Pickers 类型	52	5.2.5 清空画布	78
4.3.5 search 类型	56	5.3 绘制贝塞尔曲线	79
4.3.6 tel 类型	57	5.3.1 二次贝塞尔曲线	79
4.3.7 color 类型	57	5.3.2 三次贝塞尔曲线	80
4.4 新增的输入属性	58	5.4 绘制变形图形	81
4.4.1 form 属性	58	5.4.1 保存与恢复 canvas 状态	81
4.4.2 formaction 属性	59	5.4.2 移动坐标空间	82
4.4.3 formmethod 属性	59	5.4.3 旋转坐标空间	84
4.4.4 formenctype 属性	59	5.4.4 缩放图形	86
4.4.5 formtarget 属性	60	5.4.5 矩阵变换	86
4.4.6 autofocus 属性	60	5.5 丰富图形效果	89
4.4.7 required 属性	60	5.5.1 应用不同的线型	89
4.4.8 labels 属性	61	5.5.2 绘制线性渐变	90
4.4.9 control 属性	62	5.5.3 绘制径向渐变	91
4.4.10 placeholder 属性	63	5.5.4 绘制图案	92
4.4.11 list 属性	63	5.5.5 设置图形的透明度	93
4.4.12 文本框的 pattern 属性	64	5.5.6 创建阴影	93
4.4.13 selectionDirection 属性	64	5.6 图像处理	95
4.4.14 复选框的 indeterminate 属性	65	5.6.1 裁剪图像	95
4.4.15 height 与 width 属性	66	5.6.2 像素处理	96
4.4.16 maxlength 和 wrap 属性	67	5.6.3 组合图形	98
4.5 表单验证	67	5.6.4 混合图像	100
4.5.1 自动验证	67	5.7 绘制文字	101
4.5.2 取消验证	68	5.7.1 绘制填充文字	101
4.5.3 显式验证	68	5.7.2 文字相关属性	101
4.6 本章小结	69	5.7.3 绘制轮廓文字	102

5.7.4 获取文字宽度	102	7.3.3 创建一个简单的数据库	131
5.7.5 文字绘制实战	102	7.3.4 综合应用——点评功能	132
5.8 本章小结	103	7.4 本章小结	138
5.9 思考和练习	103	7.5 思考和练习	138
第 6 章 音频与视频	104	第 8 章 离线应用程序	139
6.1 HTML5 多媒体技术概述	104	8.1 离线 Web 应用程序详解	139
6.1.1 关于编解码器	105	8.1.1 本地缓存技术产生的原因	139
6.1.2 音频编解码器	105	8.1.2 本地缓存概述	139
6.1.3 视频编解码器	106	8.1.3 本地缓存与浏览器网页缓存的 区别	140
6.2 浏览器音视频支持检测	107	8.1.4 浏览器支持检测	141
6.3 audio 与 video 元素	109	8.2 HTML5 离线应用详解	141
6.3.1 audio 元素	109	8.2.1 Web 服务器配置	141
6.3.2 video 元素	111	8.2.2 manifest 文件结构与含义	142
6.4 综合实战	112	8.2.3 搭建离线应用程序	143
6.4.1 用脚本控制音乐播放	112	8.2.4 离线应用中浏览器和服务器的 交互过程	143
6.4.2 用脚本控制视频播放	114	8.3 applicationCache 对象	145
6.5 为音频或视频添加字幕	115	8.3.1 swapCache 方法	145
6.5.1 track 元素的基础知识	115	8.3.2 applicationCache 对象的事件	146
6.5.2 track 元素的各种属性	116	8.4 缓存网站的首页	149
6.5.3 WebVTT 文件	117	8.4.1 新建 HTML5 页面	149
6.6 本章小结	120	8.4.2 添加.htaccess 支持	149
6.7 思考和练习	121	8.4.3 创建 manifest 文件	150
第 7 章 本地存储	122	8.4.4 关联 manifest 文件到 HTML5 页面	150
7.1 Web 存储	122	8.4.5 测试离线应用	150
7.1.1 Cookie 存储机制的优缺点	122	8.5 本章小结	151
7.1.2 为什么要用 Web 存储	122	8.6 思考和练习	151
7.1.3 Web 存储的优缺点	123	第 9 章 Web Workers 多线程处理	152
7.2 使用 Web 存储	123	9.1 认识 Web Workers	152
7.2.1 检查浏览器的支持性	123	9.1.1 HTML4 处理长耗时操作的 问题	152
7.2.2 设置和获取数据	124	9.1.2 HTML5 针对长耗时操作的 解决方法	152
7.2.3 Web 存储的其他操作	124	9.1.3 Web Workers 的使用示例	153
7.2.4 监测 Web 存储事件	125	9.1.4 Web Workers 的使用场合	155
7.2.5 制作简单的网页皮肤	127		
7.2.6 网站人气值和在线人数统计	128		
7.3 本地数据库	129		
7.3.1 本地数据库的基本概念	129		
7.3.2 用 executeSql 执行查询	129		

9.2 使用 Web Workers	155	第 11 章 CSS3 概述	183
9.2.1 检查浏览器支持情况	155	11.1 CSS 的历史变迁	183
9.2.2 与线程进行数据交互	156	11.1.1 CSS 产生的原因	183
9.3 线程的嵌套	158	11.1.2 CSS 的发展历史	184
9.3.1 单层嵌套	158	11.1.3 Hello CSS World	185
9.3.2 在多个子线程中进行数据交互	160	11.1.4 为文档应用 CSS 的方式	187
9.4 线程中可用的变量、函数与类	161	11.2 了解 CSS3 新增特性	188
9.5 共享线程	162	11.2.1 CSS3 选择器	188
9.5.1 基础知识	162	11.2.2 引用服务器端字体	189
9.5.2 与共享线程通信	163	11.2.3 换行处理	190
9.6 线程的工作原理	163	11.2.4 文字渲染	191
9.6.1 线程事件处理模型	163	11.2.5 多栏布局	191
9.6.2 线程的应用范围和作用域	164	11.2.6 边框和颜色	191
9.6.3 线程的生命周期	164	11.2.7 渐变效果	192
9.7 综合实战	165	11.2.8 阴影和反射效果	193
9.7.1 使用线程做后台数值计算	165	11.2.9 背景效果	194
9.7.2 使用共享线程处理多用户并发连接	167	11.2.10 盒子模型	195
9.7.3 HTML5 线程代理	168	11.2.11 过渡、形变与动画	197
9.8 本章小结	171	11.3 CSS3 兼容性速查	199
9.9 思考和练习	171	11.4 本章小结	199
第 10 章 Geolocation 地理位置	172	11.5 思考和练习	200
10.1 Geolocation API 的基本知识	172	第 12 章 CSS3 选择器	201
10.1.1 位置信息的表示方式	172	12.1 选择器的用法	201
10.1.2 位置信息的来源	173	12.2 属性选择器	202
10.2 使用 Geolocation API	174	12.2.1 CSS2 定义的属性选择器	202
10.2.1 检测浏览器支持情况	174	12.2.2 CSS3 定义的属性选择器	202
10.2.2 获取当前地理位置	174	12.2.3 案例实战	203
10.2.3 持续监视位置信息	176	12.3 结构伪类选择器	205
10.2.4 停止获取位置信息	176	12.3.1 CSS 中的伪类选择器及伪元素	205
10.2.5 隐私保护	176	12.3.2 root、not、empty 和 target	205
10.2.6 处理位置信息	177	12.3.3 first-child、last-child、nth-child(n) 和 nth-last-child(n)	207
10.2.7 position 对象	177	12.3.4 first-of-type 和 last-of-type	209
10.3 使用百度地图	178	12.3.5 nth-of-type(n) 和 nth-last-of-type(n)	210
10.4 本章小结	182		
10.5 思考和练习	182		

12.3.6	only-child 选择器	211	13.4.4	RGBA 模式	233
12.4	UI 元素状态伪类选择器	212	13.4.5	HSL 模式	233
12.4.1	UI 元素状态伪类选择器的 语法	212	13.4.6	HSLA 模式	234
12.4.2	E:hover、E:active 和 E:focus	213	13.5	本章小结	234
12.4.3	E:enabled 与 E:disabled	214	13.6	思考和练习	234
12.4.4	E:read-only 与 E:read-write	214	第 14 章	背景和边框	235
12.4.5	E:checked、E:default 和 E:indeterminate	215	14.1	设计多色边框	235
12.4.6	E::selection	216	14.1.1	用法详解	235
12.4.7	E:invalid 与 E:valid	217	14.1.2	案例实战	236
12.4.8	E:required 与 E:optional	218	14.2	设计边框背景	237
12.4.9	E:in-range 与 E:out-of- range	219	14.2.1	border-image 属性	237
12.5	本章小结	219	14.2.2	border-image 绘制原理 简述	238
12.6	思考和练习	220	14.3	设计圆角	238
第 13 章	CSS3 文本属性	221	14.3.1	border-radius 属性	238
13.1	CSS3 文本属性概述	221	14.3.2	border-radius 属性的 4 种 写法	239
13.2	设计文本阴影	221	14.3.3	用 border-radius 属性画实心 半圆和实心圆	240
13.2.1	text-shadow 属性的使用 方法	222	14.4	设计阴影	242
13.2.2	一般文字阴影效果	222	14.4.1	box-shadow 属性	242
13.2.3	文字凹凸效果	223	14.4.2	box-shadow 兼容性处理	242
13.2.4	为文本指定多个阴影	224	14.4.3	案例实战	243
13.3	设置文本样式	224	14.5	设计背景	245
13.3.1	text-stroke 属性	225	14.5.1	background-image 属性	246
13.3.2	文本溢出	225	14.5.2	background-position 属性	246
13.3.3	强制换行——word-wrap 属性	227	14.5.3	background-size 属性	247
13.3.4	嵌入字体——@font-face	228	14.5.4	background-origin 属性	249
13.3.5	字体尺寸——font-size- adjust 属性	229	14.5.5	background-repeat 属性	250
13.4	颜色模式	232	14.5.6	background-clip 属性	250
13.4.1	关键字	232	14.5.7	background-attachment 属性	252
13.4.2	十六进制	233	14.6	本章小结	252
13.4.3	RGB 模式	233	14.7	思考和练习	252
			第 15 章	变形处理	253
			15.1	认识 transform 属性	253
			15.2	2D 变形	254

15.2.1	旋转	254	16.3.2	线性渐变在 Mozilla 下的应用	291
15.2.2	缩放	254	16.3.3	线性渐变在 Opera 下的应用	292
15.2.3	移动	255	16.3.4	线性渐变在 IE 下的应用	292
15.2.4	扭曲	256	16.4	案例综合实战	292
15.2.5	复杂变形	257	16.4.1	设计级联菜单	292
15.3	3D 变形	260	16.4.2	设计实用按钮	297
15.3.1	3D 位移	261	16.5	本章小结	299
15.3.2	3D 旋转	264	16.6	思考和练习	299
15.3.3	3D 缩放	267	第 17 章	网页布局	300
15.3.4	3D 变形兼容性	268	17.1	多栏布局	300
15.3.5	多重变形	269	17.1.1	设置列宽和列数	300
15.4	变形矩阵	275	17.1.2	设置列间距	303
15.4.1	矩阵概述	276	17.1.3	设置列边框	303
15.4.2	变形与坐标系统	276	17.1.4	设置跨列标题	304
15.4.3	2D 矩阵变形	276	17.1.5	统一列高	305
15.4.4	3D 矩阵变形	278	17.2	盒布局	305
15.4.5	使用矩阵实现多重变形	279	17.2.1	CSS 盒子模型	305
15.5	本章小结	280	17.2.2	使用盒布局	306
15.6	思考和练习	280	17.2.3	盒布局和多栏布局的区别	309
第 16 章	设计动画	282	17.3	弹性盒布局	309
16.1	过渡动画	282	17.3.1	对多个元素使用 flex 属性	309
16.1.1	定义过渡属性	283	17.3.2	设置元素的显示顺序	310
16.1.2	定义过渡时间	284	17.3.3	设置元素的排列方向	312
16.1.3	定义过渡延迟时间	285	17.3.4	定义宽高自适应	312
16.1.4	定义过渡效果	286	17.3.5	消除空白	314
16.2	3D 动画	288	17.3.6	灵活使用 flex 属性	315
16.2.1	定义动画名称	288	17.3.7	控制换行方向	320
16.2.2	定义动画时间	288	17.4	弹性盒布局的布局原理	321
16.2.3	定义动画播放方式	288	17.4.1	弹性盒布局概述	321
16.2.4	定义动画延迟时间	288	17.4.2	justify-content 属性	322
16.2.5	定义动画播放次数	289	17.4.3	align-items 属性	323
16.2.6	定义动画播放方向	289	17.5	本章小结	323
16.2.7	控制播放状态	289	17.6	思考和练习	323
16.2.8	翻转的图片	289	第 18 章	综合实例	324
16.3	渐变效果	290	18.1	前端应用开发的现状与趋势	324
16.3.1	线性渐变在 WebKit 下的应用	290			

18.1.1	HTML5 应用现状	324	18.3.1	组织网页结构	327
18.1.2	HTML5 行业发展趋势	325	18.3.2	构建网页标题	329
18.2	网站开发流程	325	18.3.3	构建侧边栏	331
18.2.1	确定建站目标	325	18.3.4	构建主体内容	332
18.2.2	进行需求分析	325	18.3.5	构建版权信息	339
18.2.3	绘制网站原型	326	18.4	手机阅读器	339
18.2.4	系统整理所需资料	326	18.4.1	使用到的技术	339
18.2.5	与网站设计美工确定布局和 风格	326	18.4.2	HTML 页面代码分析	340
18.2.6	程序员完成网站功能实现	326	18.4.3	CSS3 样式代码分析	343
18.2.7	网站上线测试	326	18.4.4	JavaScript 脚本代码分析	352
18.2.8	网站推广	327	18.5	本章小结	361
18.3	企业网站	327	参考文献		362

第1章 Web开发新时代

HTML5 自 2010 年推出以来,受到各大浏览器厂商的支持和广大开发人员的喜爱。2010 年,微软 IE9 预览版在 MIX10 大会上首次公开亮相,工程师在介绍时,从前端角度将 Web 发展历程分为三个阶段:第一个阶段为 Web 1.0,主流技术是 HTML 和 CSS;第二阶段为 Web 2.0,主流技术为 Ajax 应用,如 JavaScript/DOM/异步数据请求;第三阶段则是即将到来的 HTML5+CSS3 阶段。2014 年 10 月 29 日,万维网联盟宣布,经过几乎 8 年的艰辛努力,HTML5 标准规范终于最终制定完成并公开发布,这宣告 Web 开发正式进入第三个阶段。

本章学习目标:

- 了解什么是 HTML5,以及 HTML5 相比之前版本的 HTML 有哪些区别
- 了解世界各大知名浏览器目前的发展策略,以及为什么它们都不约而同地把支持 HTML5 当成目前的工作重点,就连微软也把全面支持 HTML5 作为 IE 浏览器的开发重点与主要宣传手段
- 了解为什么开发者今后可以大胆地使用 HTML5 进行 Web 网站与 Web 应用程序的开发,以及 HTML5 被正式推广以后,之前的 Web 网站与 Web 应用程序怎么办
- 了解 HTML5 到底可以解决哪些问题

1.1 HTML5 概述

2004 年成立的 Web 超文本应用技术工作组(Web Hypertext Application Technology Working Group, 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 HTML5 的工作草案。2009 年,W3C 停止对 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 规范得到持续的完善。2014 年 10 月 29 日,HTML5 规范终于最终制定完成并公开发布。

1.1.1 HTML5 的目标

HTML5 的目标是创建更简单的 Web 程序,书写出更简洁的 HTML 代码。例如,为了使 Web 应用程序的开发变得更容易,提供了很多 API;为了使 HTML 变得更简洁,开发出了新的属性、新的元素,等等。总体来说,HTML5 为下一代 Web 平台提供了许许多多新的功能。

HTML5 提供了以下革命性的新功能:

首先,在 HTML5 之前,有很多功能必须使用 JavaScript 等脚本语言才能实现,譬如在登录页面中经常使用的让文本框获得光标焦点的功能。如果使用 HTML5,同样的功能只要使用元素的属性标签即可实现。这样的话,整个页面就变得非常清楚、直观且容易理解。因此,Web 设计者可以非常放心大胆地使用 HTML5 中这些新增的属性标签。由于 HTML5 中

提供了大量的这种可以替代脚本的属性标签,使得开发出来的界面语言也变得更加简洁易懂。

不但如此,HTML5 使页面结构变得清楚明了。之前使用的 `div` 标签也不再使用了,而是使用 HTML5 提供的更加语义化的结构标签。这样书写出来的界面结构显得非常清晰,各部位要展示什么内容也一目了然。

虽然 HTML5 宣称的立场是“非革命性的发展”,但是它所带来的功能是让人渴望的,使用它进行设计也是简单的,因此深受 Web 设计者和 Web 开发者的欢迎。

1.1.2 HTML5 新特性

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 不是颠覆性的创新,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进行。

2. 合理性

HTML5 新增加的元素都是经过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 分析了上百万个页面,从中分析出 `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 直接添加了一个 `<header>` 标签。也就是说,HTML5 新增的很多元素、属性或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,宗旨是“用户即上帝”,这意味着在遇到无法解决的冲突时,HTML5 规范会把用户放在第一位,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 的绝大部分功能是实用的,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能提高代码编写效率。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,HTML5 规范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 API 都可通用。这个安全模型不需要借助任何所谓聪明、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 规范已经不支持旧版 HTML 的大部分表现功能了。

6. 简单

HTML5 要的就是简单, 避免不必要的复杂性。为了尽可能简单, 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

通用访问的原则可以分成如下 3 个概念:

- 可访问性: 出于对残疾人士的考虑, HTML5 与 WAI(Web Accessibility Initiative, Web 可访问性倡议)和 ARIA(Accessible Rich Internet Application, 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: 例如, 新的<body>元素支持在东亚地区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 的 canvas 为例, 以前在 HTML4 页面中较难画出对角线, 而有了 canvas 元素就可以轻易地实现了。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 即可播放与该帧对应的视频文件。

1.1.3 HTML5 深受欢迎的原因

1. 世界知名浏览器厂商对 HTML5 的支持

HTML5 被说成划时代也好, 具有革命性也好, 如果不被业界承认并且大范围地推广使用, 这些都没有意义。事实上, 今后 HTML5 被正式、大规模地投入应用的可能性是相当高的。

通过对 Internet Explorer、Google、Firefox、Safari、Opera 等主流 Web 浏览器的发展策略的调查, 发现它们都在支持 HTML5 上采取了措施。

- 微软: 2010 年 3 月 16 日, 微软于美国拉斯维加斯举行的 MIX10 技术大会上宣布推出 IE9 浏览器开发者预览版。微软称, IE9 完成开发后, 会更多地支持 CSS3、SVG 和 HTML5 等互联网浏览通用标准。
- Google: 2010 年 2 月 19 日, Google Gears 项目经理伊安·费特通过博客宣布, Google 将放弃对 Gear 浏览器插件项目的支持, 以此重点开发 HTML5 项目。据费特表示,

目前,在 Google 看来,Gears 面临的主要问题是,该应用与 HTML5 的诸多创新非常相似,而且 Google 一直在积极发展 HTML5 项目。因此,只要 Google 不断以加强新网络标准的应用功能为工作重点,那么为 Gears 增加新功能的意义就不大了。目前,多种浏览器将会越来越多地为 Gmail 以及其他服务提供更多脱机功能方面的支持,因此 Gears 面临的需求也在日益下降,这是 Google 做出上述调整的重要原因。

- 苹果:2010年6月7日,苹果在开发者大会的会后发布了 Safari 5,这款浏览器支持 10 个以上的 HTML5 新技术,包括全屏幕播放、HTML5 视频、HTML5 地理位置、HTML5 切片元素、HTML5 的可拖动属性、HTML5 的形式验证、HTML5 的 Ruby、HTML5 的 Ajax 历史和 WebSocket 字幕。
- Opera:2010年5月5日,Opera 软件公司首席技术官 Hakon Wium Lie 先生在访华之际,接受了中国软件资讯网等少数几家媒体的采访。号称“CSS 之父”的他认为,HTML5 和 CSS3 将是全球互联网发展的未来趋势,目前包括 Opera 在内的诸多浏览器厂商,纷纷在研发 HTML5 相关产品,Web 的未来属于 HTML5。
- Mozilla:2010年7月,Mozilla 基金会发布了即将推出的 Firefox 4 浏览器的第一个早期测试版,并在该版本的 Firefox 浏览器中进行了大幅改进,包括新的 HTML5 语法分析器,以及支持更多 HTML5 形式的控制等。从官方文档来看,Firefox 4 对 HTML5 提供完全级别的支持。目前包括在线视频、在线音频等多种应用都已在该版本中实现。

以上证据表明,目前这些浏览器都纷纷朝着支持 HTML5、结合 HTML5 的方向在迈进,因此 HTML5 已经被广泛推行开来。

2. 时代的要求

现在的时代已经迫切地要求有一个统一的互联网通用标准。在 HTML5 发布之前的情况是,由于各大浏览器之间的不统一,光是修改 Web 浏览器之间的由于兼容性引起的 Bug 就浪费了大量时间。而 HTML5 的目标就是将 Web 带入一个成熟的应用平台,在 HTML5 平台上,视频、音频、图像、动画以及同电脑的交互都被标准化。

关于 Web 浏览器,网页标准计划小组设计并推出了 Acid3 测试,它是针对网页浏览器及设计软件之标准相容性的一项测试。它针对 Web 应用程序中使用的动态内容进行检查,测试焦点主要集中在 ECMAScript、DOM Level 3、Media Queries 和 data:URL 上。

Acid3 测试推出后,各大浏览器都认真接受了它的测试并希望能够获得比较高的分数。这个测试的设计者,正是 W3C 的开发及设计者,HTML5 的重要人物 Ian Hickson。Ian Hickson 是 WHATWG 开发团队的成员,负责 Web 标准的设计,现在是 W3C 的 HTML5 工作组的负责人之一。

Ian Hickson 设计 Acid3 测试的意图,是给声称“让开发者能够什么都不必担心,可以放心大胆地进行开发”的各大 Web 浏览器提供一个机会,让它们能够以此来证明自己是优秀的。针对 Acid3 的宣传是很重要的,要想扩大 Web 浏览器的市场份额,宣称遵从它所依赖的标准是最有效的宣传方法。

1.1.4 HTML5 的构成

HTML5 主要包括下面这些功能:Canvas(2D 和 3D)、Channel 消息传递、Cross-Documnt 消息传送、Geolocation、MathML、Microdata、Server-Send Events、Scalable Vector

Graphics(SVG)、WebSocket API 及协议、Web Origin Concept、Web Storage、Web SQL Database、Web Workers、XMLHttpRequest Level 2。

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利。
- 默认设置智能化。

1.2.1 HTML 的历史变迁

HTML 最早从 2.0 版开始，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 制定的，从第三个版本开始，W3C 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。问题是到了版本 4.01 的时候，W3C 的认识发生倒退，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 和 HTML 4.01 其实是一样的。唯一不同的是 XHTML 1.0 要求使用 XML 语法。

从规范本身的内容来看，本质是相同的，不同之处在于编码风格，因为浏览器读取符合 HTML 4.01、HTML 3.2 或 XHTML 1.0 规范的网页都没有问题。对于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+HTML 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，属性值也全部添加引号。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，唯一的变化就是把文档标记为 XML