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。

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基础知识点的理解。

- 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

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 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-Document 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

文档。而在使用 XHTML 1.0 的时候，还可以把文档标记为 HTML。

但是，这样做带来了很多问题。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 类型的文档，而规范又要求以 XML 类型来发送文档，这对于广大用户来说，是一件很痛苦的事。

所以说，XHTML 1.1 有点脱离实际，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 的错误处理模型确是这样的：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，浏览器也会崩溃，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，对于 Firefox 而言，遇到错误就停止解析，并且不呈现其他任何内容，这是严格按照 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 XHTML 2，但是这个版本并没有完成。从理论的角度来说，XHTML 2 是一个非常好的规范。如果所有人都同意使用的话，也一定是非常好的格式，只不过它还不够实际。

首先，XHTML 2 仍然使用 XML 错误处理模型，用户必须保证以 XML 类型发送文档；其次，XHTML 2 中有意不再向后兼容已有的 HTML 版本，甚至曾经讨论废除 img 元素，这对于每天都在做 Web 开发的人员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 XHTML 2 在理论上是多么完美的一种格式，却从未有机会付诸实践。之所以难以付诸实践，就是因为开发人员永远不会支持它，它向后不兼容。同样，浏览器厂商也不会支持它。

XHTML 1 和 XHTML 2 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，遇到错误就停止解析，这怎么能叫开放呢？

1.2.2　HTML5 开发动力

在 20 世纪末期，W3C 琢磨着改良 HTML 语言。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。他建议新的任务组可以跟 XHTML 2 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，因为他们觉得 XHTML 2 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，成立了 WHATWG(Web Hypertext Applications Technology Working Group，Web 超文本应用技术工作组)，在 HTML 的基础上开展工作，向其中添加新东西。

WHATWG 的工作不久就初见成效，而 W3C 的 XHTML 2 并没有实质性进展。于是，W3C 于 2007 年组建了 HTML5 工作组，在 WHATWG 工作成果的基础上继续开展工作，由伊恩 • 希克森担任 W3C HTML5 规范的编辑，同时兼任 WHATWG 的编辑，以方便新工作组开展工作。

这就是我们今天看到的局面：一种格式，两个版本。WHATWG 的网站上有这个规范，而 W3C 的网站上也有一份。但是，这两份成果也是有区别的。W3C 最终要制定一个具体的规范，而 WHATWG 还在不断地迭代，将开发一项简单的 HTML 或 Web 技术作为工作的核心目标。

1.3　编写第一个 HTML5 页面

尽管主流浏览器的最新版本都对 HTML5 提供了很好的支持，但 HTML5 毕竟是全新的，因此在执行 HTML5 页面之前，必须先搭建支持 HTML5 的浏览器环境，并检查浏览器是否支持 HTML 标记。

1.3.1　搭建上机练习环境

目前，Microsoft 的 IE(IE9+)浏览器，以及 Mozlilla 的 Firefox 与 Google 的 Chrome 浏览器等都可以很好地支持 HTML5。本书的示例主要运行在 Chrome 浏览器上。

1.3.2　检测浏览器是否支持

安装相应的浏览器以后，为了能进一步了解浏览器支持 HTML5 新标签的情况，还可以在引入新的标签前，通过编写 JavaScript 代码来检测浏览器是否支持该标签。

浏览器在加载 Web 页面时会构造一个文本对象模型(Document Object Model，DOM)，然后通过该文本对象模型来表示页面中的各个 HTML 元素，这些元素被表示为不同的 DOM 对象。全部的 DOM 对象都共享一些公共或特殊属性，如 HTML5 的某些特性。如果在支持该属性的浏览器中打开页面，就可以很快检测出这些 DOM 对象是否支持这些特性。

下面以加入画布标记为例，说明如何检测浏览器对 canvas 标签的支持。

【例 1-1】 在 Dreamweaver 中新建一个 HTML 页面，保存为 index.html，代码如下：

```
<!DOCTYPE html>
<html>
  <head>
      <meta charset="utf-8" />
      <title></title>
      <style type="text/css">
          #myCanvas {
              background: red;
              width: 200px;
              height: 100px;
          }
      </style>
  </head>
  <body>
      <canvas id="myCanvas">该浏览器不支持 HTML5 的画布标记！</canvas>
  </body>
</html>
```

在浏览器中执行页面文件 index.html，如果浏览器不支持 HTML5 的画布标记，将会显示“该浏览器不支持 HTML5 的画布标记！”；若支持，则显示图 1-1 所示效果。需要注意的是，虽然是同一个页面，但由于不同的浏览器对 HTML5 标记的支持情况也不同，因此显示的页面效果也各异。所以，在编写 HTML5 新标记时，有必要先检测浏览器是否支持该标记。

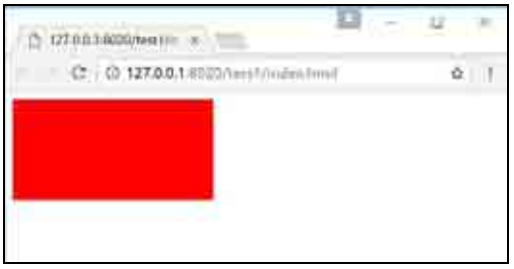

图 1-1　支持 HTML5 标记的浏览器的显示结果

1.3.3　使用 HTML5 编写简单的 Web 页面

与 HTML4 相比，HTML5 新增了很多新标签，整体页面结构也发生了很大变化。下面使用 HTML5 来编写一个 HTML 页面，保存为 index.html，在其中加入如下代码：

【例 1-2】　一个简单的 HTML5 页面。

```
<!DOCTYPE html>
<html>
  <head>
        <meta charset="UTF-8">
        <title>第一个 HTML5 页面</title>
  </head>
  <body>
        <p>Hello,World</p>
  </body>
</html>
```

该页面的运行效果如图 1-2 所示。

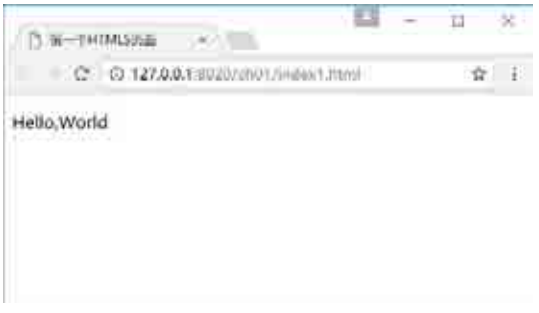

图 1-2　运行效果

通过短短几行代码就完成一个页面的开发，可见 HTML5 语法的简洁。下面逐句分析 HTML5 文档的组成。

第一行代码如下：

```
<!DOCTYPE html>
```

短短几个字符，甚至不包括版本号，就能告诉浏览器需要一个 doctype 来触发标准模式，简明扼要。接下来的代码：

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

这行代码说明了文档的字符编码，保证浏览器正确解析文档。HTML5 不区分字母、标记结束符的大小写以及属性是否加引号，下面的代码是等效的：

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

在主体<body>中，可以省略主体标记，直接编写需要显示的内容，即去掉<body>和</body>标签：

```
<!DOCTYPE html>
<html>
  <head>
        <meta charset="UTF-8">
        <title>第一个 HTML5 页面</title>
  </head>
        <p>Hello,World</p>
</html>
```

虽然在编写代码时可以省略<html>、<head>和<body>标记，但浏览器总能进行解析。

考虑到代码的可读性和可维护性，编写代码时，应该尽量增加这些在 HTML5 中可选的元素，从而最大限度地实现页面代码的简洁和完整。

1.4　HTML5 页面的特征

上一节介绍了一个 HTML5 页面的创建过程。下面通过一个较为完整的页面来介绍 HTML5 的页面特征。

1.4.1　使用 HTML5 的结构化元素

通过研究 Web 页面发现，如果使用一些带有语义性的标记，可以加快浏览器解释页面中元素的速度，如早期的<samp>、<var>元素；HTML5 继承了这些元素，并根据用户使用最为频繁的类名和 ID不断开发新的标记，因为这些标记能真正体现开发者真实意图所在。下面通过实例说明 HTML5 是如何使用这些全新的 HTML5 特征来结构化元素的。

【例 1-3】 本例将页面分成上、中、下 3 部分。上部用于显示导航；中部分为两个部分，左边设置菜单，右边显示文本内容；下部显示页面版权信息。

```
<!DOCTYPE html>
<html>
  <head>
        <meta charset="UTF-8">
        <title></title>
        <style type="text/css">
```

```
            #header,#siderLeft,#siderRight,#footer{
                border:solid 1px #666;
                padding: 10px;
                margin: 6px;
            }
            #header {width: 500px;}
            #siderLeft {
                float: left;
                width: 60px;
                height: 100px;
            }
            #siderRight{
                float: left;
                width: 400px;
                height: 100px;
            }
            #footer{
                clear: both;
                width: 500px;
            }
        </style>
    </head>
    <body>
        <div id="header">导航</div>
        <div id="siderLeft">菜单</div>
        <div id="siderRight">内容</div>
        <div id="footer">底部</div>
    </body>
</html>
```

运行以上代码，效果如图 1-3 所示。

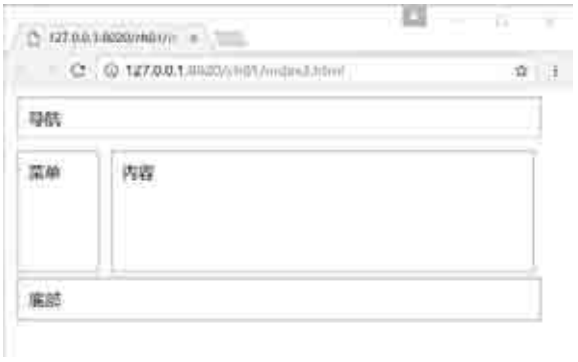

图 1-3　简单的网页布局

尽管上述代码没有任何错误，并且可以在 HTML5 环境中很好地运行，但该页面结构的很多部分对于浏览器来说都是未知的，这是因为浏览器是通过 ID 来定位元素的。因此，只要

开发者不同，就允许元素的 ID 各异，这会造成浏览器不能很好地表明元素在页面中的位置，必然影响页面解析的速度。幸好 HTML5 中新增的元素可以很快地定位某个标记，明确地表示页面中的位置。将上述代码修改成 HTML5 支持的页面代码，如下所示，运行代码后，显示的效果相同：

```
<!DOCTYPE html>
<html>
  <head>
        <meta charset="UTF-8">
        <title></title>
        <style type="text/css">
              header,nav,article,footer{
                    border: solid 1px #666666;
                    padding: 10px;
                    margin: 6px;
              }
              header {width: 500px;}
              nav {
                    float: left;
                    width: 60px;
                    height: 100px;
              }
              article {
                    float: left;
                    width: 400px;
                    height: 100px;
              }
              footer {
                    clear: both;
                    width: 500px;
              }
        </style>
  </head>
  <body>
        <header>导航</header>
        <nav>菜单</nav>
        <article>内容</article>
        <footer>底部说明</footer>
  </body>
</html>
```

虽然两段代码不一样，但在 Chrome 浏览器中实现的页面效果相同。从上述两段代码来看，使用 HTML5 新增元素创建的页面代码更加简单和高效。

可以看出，使用 等标记元素没有任何实现的意义，即浏览器不能根据标记的 ID 属性来推断这个标记的真正含义，因为 ID 号是可以变化的，不利于寻找。

而 HTML5 中的新增元素<header>可以明确告诉浏览器此处是页头，<nav>标记用于构建页面的导航，<article>标记用于构建页面内容的一部分，<footer>元素表明页面已到页脚或根元素部分，并且这些标记都可以重复使用，极大提高了开发者的工作效率。

此外，有些新增的 HTML5 元素还可以单独成为一个区域，如下所示：

```
<header>
    <article>
        <h1>内容 1</h1>
    </article>
</header>
<header>
    <article>
        <h2>内容 2</h2>
    </article>
</header>
```

在 HTML5 中，<article>可以创建一个新的节点，并且每个节点都可以有自己的单独元素，如<h1>和<h2>，这样不仅使内容区域各自分段、便于维护，而且代码简单，局部修改方便。

1.4.2　使用 CSS 美化 HTML5 文档

在支持 HTML5 新增元素的浏览器中，样式化各个新增元素变得十分简单，可以对任意一个元素应用 CSS，包括直接设置或引入 CSS 文件。需要说明的是，在默认情况下，CSS 默认元素的 display 属性值为 inline。因此，为了正确地显示设置的页面效果，需要将元素的 display 属性设置为 block。下面通过一个简单的示例说明这一点。

【例 1-4】 在页面中设置相关样式，显示一段文章的内容。

```
<head>
<meta charset="UTF-8">
<style type="text/css">
    article {display: block;}
    article header p {font-size: 13px;}
    article header h1 {font-size: 16px;}
    .p-date {font-size: 11px;}
</style>
</head>
<body>
    <article>
        <header>
            <h1><a>谷歌推出多项搜索功能 避谈</a></h1>
            <p class="p-date">日期：2017-04-02</p>
            <p>网易科技 4 月 2 日消息，据媒体报道，谷歌无人车研究有了新的进展……</p>
        </header>
    </article>
</body>
```

运行以上代码，效果如图 1-4 所示。

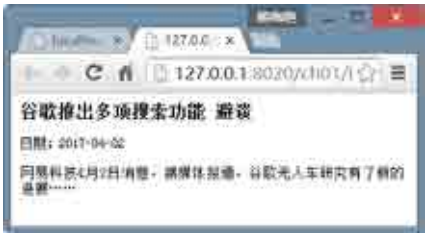

图 1-4　使用 CSS 美化 HTML5 页面

由于有些浏览器并不支持 HTML5 中新增的元素，如 IE8 或更早版本，其 CSS 只应用 IE 支持的那些元素；因此，为了能为新增的 HTML5 元素应用样式，可以在头部标记<head>中加入如下 JavaScript 代码，这样就可以应用样式了：

```
<script type="text/javascript">
        document.createElement('article');
        document.createElement('header');
</script>
```

考虑到各浏览器的兼容性不一样，可以对上述 JavaScript 代码进行优化，即使用条件语句包含该 JavaScript 代码，使浏览器只在不支持 HTML5 的情况下才执行这段脚本。

1.5　本章小结

本章是 HTML5 概述，从总体上向读者介绍 HTML5 的全貌。从总体上来说，HTML5 的出现与兴起并非偶然，这是业界专家与工程师们直面过去互联网技术的许多复杂问题，总结许多开发者在实际项目实践中经常遇到的问题、习惯性操作、解决方案的基础上，并根据当前技术发展的需要、设计原理等，基于 HTML 语言基础之上，制定出来的标准。本章首先对 HTML5 进行了概述，简单介绍了 HTML5 的目标、新特性、受欢迎的原因，以及 HTML5 文档的构成；其次，简单陈述了 HTML5 的设计原理；接着介绍了如何搭建编写和运行 HTML5 的环境，如何编写 HTML5 文档；最后介绍了 HTML5 页面的特征。希望通过本章的学习，读者能够对 HTML5 有一个总体的认识。

1.6　思考和练习

1. 了解一下 HTML 语言的历史变迁。
2. 简单描述 HTML5 的目标及特性。
3. 搭建上机练习环境，检测浏览器是否支持 HTML5。
4. 编写一个 HTML5 文档。