Web 开发技术丛书

HTML 5 与 CSS 3 权威指南
（第 3 版 · 上册）

陆凌牛　著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

HTML 5 与 CSS 3 权威指南（上册）/ 陆凌牛著．—3 版．—北京：机械工业出版社，2015.9
（Web 开发技术丛书）

ISBN 978-7-111-51443-5

I. H…　II. 陆…　III. ①超文本标记语言 – 程序设计 – 指南　②网页制作工具 – 指南
IV. ① TP312-62　② TP393.092-62

中国版本图书馆 CIP 数据核字（2015）第 213534 号

HTML 5 与 CSS 3 权威指南（第 3 版・上册）

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：姜　影　　责任校对：董纪丽
印　　刷：　　版　　次：2015 年 9 月第 3 版第 1 次印刷
开　　本：186mm×240mm　1/16　　印　　张：35.75
书　　号：ISBN 978-7-111-51443-5　　定　　价：89.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

Preface 前　言

为何写作本书

2014 年 10 月 28 日，W3C 的 HTML 工作组正式发布了 HTML 5 的正式推荐标准（W3C Recommendation），这一消息是 W3C 在美国圣克拉拉举行的 W3C 技术大会及顾问委员会会议（TPAC 2014）上宣布的。HTML 5 在这一版本中增加了支持 Web 应用开发者的许多新特性，以及更符合开发者使用习惯的新元素，并重点关注定义清晰的、一致的准则，以确保 Web 应用和内容在不同用户代理（浏览器）中的互操作性。HTML 5 是构建开放 Web 平台的核心。

2015 年 4 月 9 日，W3C 的 CSS 工作组发布 CSS 基本用户接口模块（CSS Basic User Interface Module Level 3，CSS3 UI）的标准工作草案。该文档描述了 CSS 3 中对 HTML、XML（包括 XHTML）进行样式处理所需的与用户界面相关的 CSS 选择器（selectors）、属性及属性值。它包含并扩展了在 CSS Level 2 及 Selector 规范中定义的与用户接口有关的特性。

HTML 5 带来了一组新的用户体验，如 Web 的音频和视频不再需要插件，通过 Canvas 更灵活地完成图像绘制，而不必考虑屏幕的分辨率，浏览器对可扩展矢量图（SVG）和数学标记语言（MathML）的本地支持，通过引入新的注释信息以增强对东亚文字呈现（Ruby）的支持，对富 Web 应用信息无障碍新特性的支持，等等。

前端技术将进入一个崭新的时代，至少已经开启了这扇门。

在这种局势下，学习 HTML 5 无疑成为 Web 开发者的一大重要任务，谁先学会 HTML 5，谁就掌握了开启未来 Web 平台的一把钥匙。因此，笔者希望借助此书帮助国内的 Web 开发者更好地学习 HTML 5 以及与之相伴的 CSS 3 技术，早日运用这些技术开发出一个具有现代水平的、在未来的 Web 平台上正常运行的 Web 网站或 Web 应用程序。

第 3 版与第 2 版的区别

自 2013 年上半年本书第 2 版出版以来，一直受到广大读者的欢迎，笔者在这里首先感谢广大读者的支持。自本书第 2 版出版之后，HTML 5 与 CSS 3 标准处于不断发展中，各主流浏览器也以最快的速度对 HTML 5 中各种最新公布的 API 提供了支持，其中包括各种新增元素、WebRTC 通信 API、鼠标指针锁定 API、JavaScript Promise 等。因此，本书以第 2 版的内容为基础，添加对 2013 上半年到 2015 年 5 月之间 HTML 5 中新增的各种元素及 API 的介绍，同时更新各主流浏览器对 CSS 3 的最新支持情况，以使读者能够学到 2015 年 5 月为止关于 HTML 5 与 CSS 3 标准的各种知识，了解各种最新浏览器对 HTML 5 与 CSS 3 标准的支持情况，能够早日通过这些新知识打造一个 HTML 5 时代功能强大的 Web 网站或 Web 应用程序。

具体来说，本书在第 2 版的基础上做出了如下主要修改：

- 第 2 章中新增部分 HTML 5 属性。
- 第 3 章中新增对 main 元素的介绍，移除第 2 版中对 hgroup 元素（HTML 5 标准中已废除该元素）的介绍。
- 第 4 章中新增对 dialog 元素的介绍。
- 第 5 章（第 2 版中第 6 章）中新增“使用 Path2D 对象绘制路径”和“图形、图像的组合与混合”的内容。
- 第 6 章（第 2 版中的第 15 章）中新增“对音频或视频添加字幕”内容。
- 第 10 章中新增“在 IndexedDB 数据库中保存 Blob 对象”内容。
- 新增第 12 章对 WebRTC 通信的介绍。
- 第 17 章（第 2 版中的第 16 章）中新增“鼠标指针锁定 API”、“requestAnimationFrame”、“Mutation Observer”、“JavaScript Promise”、“Beacon API”内容。
- 第 21 章中新增“使用 rem 单位定义字体大小”内容。
- 第 22 章中新增“创建盒内阴影”内容。
- 第 23 章中新增“新增的用于平铺背景图像的选项—space 与 round”和“使用渐变色背景”内容。
- 第 24 章中新增“使用 3D 变形功能”及“变形矩阵”内容。
- 第 26 章中根据 CSS 3 最新标准的内容重新编写“弹性盒布局”并新增“ calc 方法”内容。
- 第 28 章中新增“实现 CSS 3 中的滤镜特效”内容。

本书面向的读者

本书主要适合如下人群阅读：

- ❑ 具有一定基础的 Web 前端开发工程师。
- ❑ 具有一定美术功底的 Web 前端设计师和 UI 设计师。
- ❑ Web 项目的项目管理人员。
- ❑ 开设 Web 开发等相关专业的高等院校的师生和相关培训机构的学员及教师。

如何阅读本书

本书共分为上下两册。

上册对 HTML 5 中新增的语法、标记方法、元素、API，以及这些元素与 API 到目前为止受到了哪些浏览器支持等进行详细介绍。在对它们进行介绍的同时将其与 HTML 4 中的各种元素与功能进行对比，以帮助读者更好地理解为什么需要使用 HTML 5、使用 HTML 5 有什么好处、HTML 5 中究竟增加了哪些目前 HTML 4 不具备而在第三代 Web 平台上将会起到重要作用的功能与 API，以及这些功能与 API 的详细使用方法。

下册详细介绍了 CSS 3 中各种新增样式与属性，其中主要包括 CSS 3 中的各种选择器、文字与字体、背景与边框、各种盒模型、CSS 3 中的布局方式、CSS 3 中的变形与动画、CSS 3 中与媒体类型相关的一些样式与属性等。同时详细讲述了这些样式与属性到目前为止受到了哪些浏览器支持，以及针对不同浏览器应该怎样在样式代码中正确使用各种属性。最后详细讲解了两个实例，第一个实例展示了如何在一个用 HTML 5 语言编写而成的页面中综合运用 HTML 5 中新增的各种结构元素，如何对这些结构元素综合使用 CSS 3 样式；第二个实例展示了如何使用 HTML 5 中新增的表单元素以及操作本地数据库的功能来实现一个具有现代风格的 Web 应用程序，如何在这个由 HTML 5 语言编写而成的 Web 应用程序中综合使用 CSS 3 样式来完成页面的布局以及视觉效果的美化工作。

全书一共包含 389 个代码示例，每个代码示例都经过笔者上机实践，确保运行结果正确无误。每个代码示例的详细代码及其用到的脚本文件、各种资源文件都可在华章公司的官方网站（www.hzbook.com）的本书页面上下载，因为是用 HTML 5 编写而成的网页，所以可直接在各种浏览器中打开并查看运行结果（少量页面需要先建立网站，然后通过访问网站中该页面的方式进行查看；少量页面使用服务器端 PHP 脚本语言，可在 Apache 服务器中运行；少量页面使用服务器端 Node.js 脚本语言，可以通过安装运行 Node.js 来运行服务器并查看示例页面）。同时，对于 HTML 5 中的各种元素和各种 API，以及 CSS 3 中的各种属性与样式

受到了哪些浏览器的支持在书中都进行了详细介绍，读者可以针对不同的页面选择正确的浏览器来查看其正确的运行结果。

致谢

在本书的写作过程中，机械工业出版社华章公司的编辑杨福川先生和姜影女士给予了很大的帮助和支持，并提出了很多中肯的建议，在此表示感谢。同时，还要感谢机械工业出版社的所有编审人员为本书的出版所付出的辛勤劳动。本书的成功出版是大家共同努力的结果，谢谢他们。

另外，在本书的写作过程当中，由于时间及个人水平上的原因，有可能存在一些对 HTML 5 及 CSS 3 认识不全面或疏漏的地方，敬请读者批评指正，作者的联系 QQ 为 240824399，联系邮箱为 240824399@qq.com，谨以最真诚的心希望能与读者共同交流、共同成长。

Contents 目　录

下　册

第 1 章 Chapter 1

Web 时代的变迁

自从 2010 年 HTML 5 正式推出以来，它立刻受到了世界各大浏览器的热烈欢迎与支持。根据世界上各大 IT 界知名媒体评论，新的 Web 时代，HTML 5 的时代马上就要到来。本章重点介绍什么是 HTML 5，HTML 5 产生的时代背景，为什么 HTML 5 会如此深受业界欢迎，以及 HTML 能够解决什么问题。

学习内容：

- 初步了解什么是 HTML 5，HTML 5 与之前版本的 HTML 大致上有哪些区别。
- 了解世界各大知名浏览器目前的发展策略，为什么它们都不约而同地把支持 HTML 5 当成目前的工作重点，就连微软也把全面支持 HTML 5 作为新版 Internet Explorer 9（IE 9）浏览器的开发重点与主要宣传手段。
- 了解为什么说开发者今后可以放心大胆地使用 HTML 5 进行 Web 网站与 Web 应用程序的开发，HTML 5 被正式推广以后之前的 Web 网站与 Web 应用程序怎么办。
- 了解使用 HTML 5 到底可以解决哪些问题。

1.1 迎接新的 Web 时代

1.1.1 HTML 5 时代即将来临

自从 2010 年 HTML 5 正式推出以来，它就以一种惊人的速度被迅速推广着，就连微软也因此为下一代 IE 9 做了标准上的改进，使其能够支持 HTML 5。关于各主流浏览器对于 HTML 5 所表现出来的热烈欢迎、积极支持的详细情况，以及为什么 HTML 5 会如此受欢迎，

我们将在后面几节中详细介绍，这里，笔者要告诉大家的是，目前业界全体都步调一致地朝着 HTML 5 的方向迈进着，HTML 5 的时代马上就要到来了。

在全面介绍 HTML 5 的相关知识之前，我们先来认识一下 HTML 5 中的部分代码，对 HTML 5 有个初步的了解。

首先，我们来看一段 HTML 4 中常见的 JavaScript 代码，如代码清单 1-1 所示。

代码清单 1-1　HTML 4 中的 JavaScript 代码示例

```
<form>
<p><label>Username:<input name=search type="text" id="search"></label></p>
<script type="text/javascript">
       document.getElementById ('search').focus()
</script>
</form>
```

在 HTML 5 中，这段代码将会以怎样的形式出现呢？具体如代码清单 1-2 所示。

代码清单 1-2　用 HTML 5 实现代码清单 1-1 中的 JavaScript 代码

```
<form>
<p><label>Search:<input name=search autofocus></label></p>
</form>
```

我们来看一下在 HTML 4 中常见的一种页面结构，代码如代码清单 1-3 所示。

代码清单 1-3　div 标签示例（用 HTML 4 实现）

```
<div id="header">...</div>
<div id="nav">...</div>
<div class="article">
</div>
<div id="side-bar">...</div>
<div id="footer">...</div>
```

页面中有关该部分的结构示意图如图 1-1 所示。

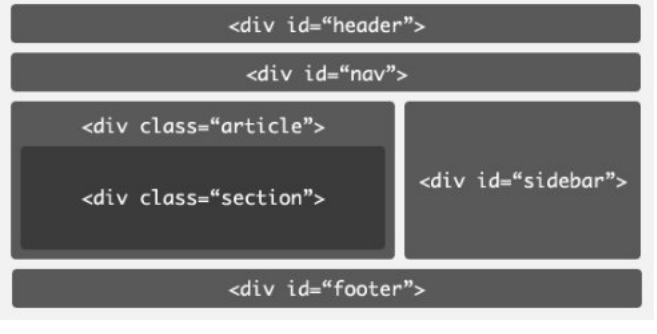


图 1-1　HTML 4 中的页面结构

那么，在 HTML 5 中，又会用怎样的页面代码来描述这种结构呢？具体如代码清单 1-4 所示。

代码清单 1-4　HTML 5 中的新型结构示例

```
<header>...</header>
<nav>...</nav>
<article>
</article>
<aside>...</aside>
<footer>...</footer>
```

页面中有关该部分的结构示意图如图 1-2 所示。

怎么样？看出区别来了吗？在第一个示例中，我们可以看见，在 HTML 4 中的一段 JavaScript 代码，在 HTML 5 中消失了，取而代之的是一个在 HTML 5 中新出现的属性。在第二个示例中，我们可以看见，在 HTML 4 中常见的用 div 来划分页面结构的方法，到了 HTML 5 中，也被一种 HTML 5 中新出现的标签给替代了。那么究竟为什么 HTML 5 要对 HTML 4 的脚本以及页面代码做这种修改呢？做这种修改的目的又是什么呢？

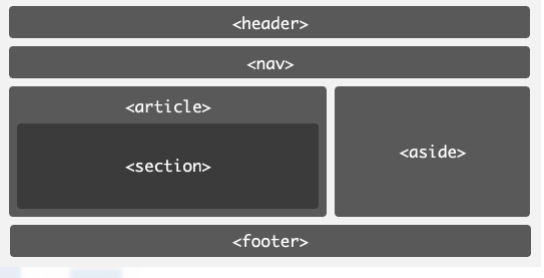


图 1-2　HTML 5 中的页面结构

1.1.2　HTML 5 的目标

HTML 5 的目标是为了能够创建更简单的 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，为下一代 Web 平台提供了许许多多新的功能。

那么让我们先来初步接触一下在 HTML 5 中究竟提供了哪些革命性的新功能。在第 2 章中，我们会针对这些功能做一个全面介绍。

首先，在 HTML 5 之前，有很多功能必须要使用 JavaScript 等脚本语言才能实现，譬如前面例子中提到，登录画面中经常使用的让文本框获得光标焦点的功能。如果使用 HTML 5，同样的功能只要使用元素的属性标签就可以实现了。这样的话，整个页面就变得非常清楚直观，容易理解。因此，Web 设计者可以非常放心大胆地使用这些 HTML 5 中新增的属性标签。由于 HTML 5 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，HTML 5 使页面结构也变得清楚明了。之前使用的 div 标签也不再使用了，

而是使用前面 HTML 5 示例中所提到的更加语义化的结构标签。这样的话，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让人一目了然。

虽然 HTML 5 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，使用它所进行的设计也是很简单的，因此，它深受 Web 设计者与 Web 开发者的欢迎。

1.2 HTML 5 深受欢迎的理由

1.2.1 世界知名浏览器厂商对 HTML 5 的支持

HTML 5 被说成是划时代也好，具有革命性也好，如果不能被业界承认并且大面积地推广使用，这些都是没有意义的。事实上，今后 HTML 5 被正式地、大规模地投入应用的可能性是相当高的。

通过对 Internet Explore、Google、Firefox、Safari、Opera 等主要的 Web 浏览器的发展策略的调查，发现它们都在支持 HTML 5 上采取了措施。

- 微软：2010 年 3 月 16 日，微软于拉斯维加斯市举行的 MIX10 技术大会上宣布已推出 IE9 浏览器开发者预览版。微软称，IE9 完成开发后，将更多支持 CSS 3、SVG 和 HTML 5 等互联网浏览通用标准。
- Google ：2010 年 2 月 19 日，谷歌 Gears 项目经理伊安 – 费特通过博客宣布，谷歌将放弃对 Gears 浏览器插件项目的支持，以此重点开发 HTML 5 项目。据费特表示，目前，在谷歌看来，Gears 面临的主要问题是，该应用与 HTML 5 的诸多创新非常相似，而且谷歌一直积极发展 HTML 5 项目。因此，只要谷歌不断以加强新网络标准的应用功能为工作重点，那么为 Gears 增加新功能就无太大意义了。目前，多种浏览器将会越来越多地为 GMail 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，这是谷歌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 HTML 5 新技术，包括全屏幕播放、HTML 5 视频、HTML 5 地理位置、HTML 5 切片元素、HTML 5 的可拖动属性、HTML 5 的形式验证、HTML 5 的 Ruby、HTML 5 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 Hakon Wium Lie 认为，HTML 5 与 CSS 3 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 HTML 5 相关产品，Web 的未来属于 HTML 5。
- 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。在该版本中的 Firefox 浏览器中进行了大幅改进，包括新的 HTML 5 语法分析器，以及支持更多 HTML 5 形式的控制等。从官方文档来看，Firefox 4 对 HTML 5 是完全级别的支持。目前包括在线视频、在线音频等多种应用都已在该版中实现。

以上证据表明，目前这些浏览器都纷纷地朝着支持 HTML 5、结合 HTML 5 的方向迈进着，因此 HTML 5 已经被广泛地推行开来了。为什么 HTML 5 会如此受欢迎，理由如 1.2.2 节和 1.2.3 节所示。

1.2.2　第一个理由：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。HTML 5 之前的情况是，由于各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而引起的 bug 就浪费了大量时间。而 HTML 5 的目标就是将 Web 带入一个成熟的应用平台，在 HTML 5 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 应用程序中使用着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。

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，正是在 W3C 开发及设计者，HTML 5 的重要人物 Ian Hickson。Ian Hickson 是 WHATWG（Web Hypertext Application Technology Working Group）开发团体的成员，担任 Web 标准规格的设计，现在是 W3C 的 HTML 5 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，让他们能够以此来证明自己是优秀的。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。图 1-3 为 Acid3 的一个测试图。

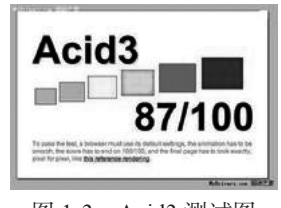


图 1-3　Acid3 测试图

1.2.3　第二个理由：Internet Explorer 8

Internet Explorer 也积极地朝着支持 HTML 5 的方向迈进着。Internet Explorer 对此十分重视。虽然它的使用者依然很多，但是由于最近被 Firefox 等其他 Web 浏览器抢去了很多市场份额，它很不甘心。于是继 Internet Explorer 7（IE 7）发表后不久，立刻推出了 Internet Explorer 8（IE 8）的 Release 版。

新推出的 IE 8 宣称遵从互联网通用标准。虽然其他的浏览器由于标榜遵从该标准而获得了很多市场份额，但是 Internet Explorer 肯定是要对此采取强有力的对策的。因此 Internet Explorer 把宣称遵从互联网通用标准看成了很重要的一件事，并且开始在 IE 8 里支持 HTML 5。

例如，HTML 5 中代替 Cookie 的 sessionStorage 功能与 globalStorage 功能在 IE 8 里都获得了支持。使用 Ajax 时如果点击返回按钮也可以真正让操作返回了（在 IE 7 中点击返回按钮，画面跳转到其他画面）。很多 Internet Explorer 自己独特的处理方法与特性，今后也会有所改变。

因为现在市场份额最高的 Internet Explorer 也在针对 HTML 5 做出积极对应，微软也对新的互联网通用标准表示了赞同和支持，所以可以说 HTML 5 在市场上大面积推广的势头是非常强的。

1.3 可以放心使用 HTML 5 的三个理由

Web 开发者最担心的是新技术推出时由于其不成熟所产生的问题。如果能够实现互联网通用标准，可以避免各浏览器之间的不统一，这一点已经被明确了，但是在朝着这方面前进的过程中会不会出现什么周折是令人担心的。

虽然 Web 开发者普遍认为有了 HTML 5 是比较好的，但是还是会很担心诸如“它在老版本的浏览器上也能正常运行吗？”，“会不会产生错误？”等各种问题。但是可以很高兴地告诉你，请放心，HTML 5 就像以前 CSS 刚开始普及时一样不会存在什么问题。

有三个理由证明可以放心使用 HTML 5：

- 兼容性：HTML 5 在老版本的浏览器上也可以正常运行。
- 实用性：HTML 5 内部并没有封装什么很复杂的、不切实际的功能，而只是封装了简单实用的功能。
- 非革命性的发展：HTML 5 的内部功能不是革命性的，只是发展性的。

以上三点就是所谓的“ HTML 设计原则”，HTML 5 也是以该设计原则为基本原则而开发出来的，各主流浏览器使用 HTML 5 的前提也就是要求 HTML 5 能够符合这些原则，今后也将以其为前提来支持 HTML 5。下面针对这些原则进行介绍。

首先是兼容性问题。虽然到了 HTML 5 时代，但并不代表现在用 HTML 4 创建出来的网站必须全部要重建，只会要求各 Web 浏览器今后能正常运行用 HTML 5 开发出来的功能。“非革命性的发展”这一点正是通过兼容性体现出来的。正是因为保障了兼容性才能让人毫不犹豫地用 HTML 5 来开发网站。

接着是实用性。实用性是指要求能够解决实际问题。HTML 5 内只封装了切实有用的功能，不封装复杂而没有实际意义的功能。

通过以上列举的 HTML 设计原则，尤其是与 HTML 4 相兼容的部分，基本上可以让人放下心来，大胆地使用 HTML 5。

1.4 HTML 5 要解决的三个问题

HTML 5 的出现，对于 Web 来说意义是非常重大的。因为它的意图是想要把目前 Web 上存在的各种问题一并解决掉，它是一个企图心比较强的 HTML 版本。

那么，到底 Web 上存在哪些问题，HTML 5 又打算怎么解决呢？

❑ Web 浏览器之间的兼容性很低。

首先要提到的就是，Web 浏览器之间的兼容性是非常低的。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 浏览器上就不正常了的事情是非常多的。

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”。规范不统一，没有被标准化，是这个问题的主要原因。

在 HTML 5 中，这个问题将得到解决。HTML 5 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，要求这些浏览器所有内部功能都要符合一个通用标准。

如果各浏览器都符合通用标准，然后以该标准为基础来书写程序，那么程序在各浏览器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。

❑ 文档结构不够明确。

第二个问题是，在之前的 HTML 版本中，文档的结构不够清晰、明确。例如，为了要表示“标题”，“正文”，之前一般都是用 元素。但是，严格说来， 不是一个能把文档结构表达得很清楚的元素，使用了过多的 要素的文章，阅读时不仔细研究，是很难看出文档结构的。而且，对于搜索引擎或屏幕阅读器等程序来说，过多使用了 div 元素，那么这些程序就连“从哪到哪算是重要的正文”，“这个 <ul> 要素是表示导航菜单，还是表示项目列表”等对于结构分析来说最基本的问题的答案也都不知道。

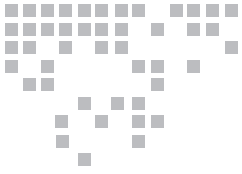
在 HTML 5 中，为了解决这个问题，追加了很多跟结构相关的元素。不仅如此，还结合了包括微格式、无障碍应用在内的各种各样的周边技术。

❑ Web 应用程序的功能受到了限制。

最后一个问题是，HTML 与 Web 应用程序的关系十分薄弱。Web 应用程序的特征是先从网络下载，然后忠实运行，因此应该对会威胁到用户安全的功能进行限制。

目前安全性的保障这方面已做到了，但对于 Web 应用程序来说，一直以来 HTML 真正所做出的贡献是很少的，譬如说就连上传文件的同时想选择一个以上的文件都做不到。

为了弥补这方面的不足，HTML 5 已经开始提供各种各样 Web 应用上的新 API，各浏览器也在快速地封装着这些 API，HTML 5 已经使富 Web 应用的实现变成了可能。



Chapter 2 第 2 章

HTML 5 与 HTML 4 的区别

HTML 5 以 HTML 4 为基础，对 HTML 4 进行了大量修改。本章从总体上概要介绍 HTML 5 对 HTML 4 进行了哪些修改，HTML 5 与 HTML 4 之间比较大的区别是什么地方。

学习内容：

- 掌握 HTML 5 与 HTML 4 在基本语法上的区别，这个基本语法区别包括 DOCTYPE 声明、内容类型（ContentType）、字符编码的指定方法、元素标记的省略、具有布尔值的属性、引号的省略等几个方面。
- 了解在 HTML 5 中新增的元素，删除了哪些 HTML 4 中的元素，以及为什么要删除这些元素，用什么元素或方法来取代这些被删除的元素。
- 了解在 HTML 5 中新增的属性，删除了哪些 HTML 4 中的属性，在 HTML 5 中用什么方法来取代这些被删除的属性。
- 掌握什么是全局属性，本章主要介绍几个常用全局属性，如 contentEditable 属性、designMode 属性、hidden 属性、spellcheck 属性以及 tabindex 属性。

2.1 语法的改变

2.1.1 HTML 5 的语法变化

与 HTML 4 相比，HTML 5 在语法上发生了很大的变化。可能很多人会因为“之前的 HTML 已经相当普及”，“如果改变基础语法，会产生什么影响”等想法而感到不安。

但是，HTML 5 中的语法变化，与其他开发语言中的语法变化在根本意义上有所不同。

它的变化，正是因为在 HTML 5 之前几乎没有符合标准规范的 Web 浏览器导致的。

HTML 的语法是在 SGML（Standard Generalized Markup Language）语言的基础上建立起来的。但是 SGML 语法非常复杂，要开发能够解析 SGML 语法的程序也很不容易，因此很多浏览器都不包含 SGML 的分析器。因此，虽然 HTML 基本上遵从 SGML 的语法，但是对于 HTML 的执行在各浏览器之间并没有一个统一的标准。

在这种情况下，各浏览器之间的互兼容性和互操作性在很大程度上取决于网站或网络应用程序的开发者在开发上所做的共同努力，而浏览器本身始终是存在缺陷的。

如上所述，在 HTML 5 中提高 Web 浏览器之间的兼容性是它的一个很大的目标，为了确保兼容性，就要有一个统一的标准。因此，在 HTML 5 中，围绕着这个 Web 标准，重新定义了一套在现有 HTML 的基础上修改而来的语法，以便各浏览器在运行 HTML 的时候能够符合一个通用标准。

因为关于 HTML 5 语法解析的算法也都提供了详细的记载，所以各 Web 浏览器的供应商可以把 HTML 5 分析器集中封装在自己的浏览器中。最新的 Firefox（默认为 4.0 以后的版本）与 WebKit 浏览器引擎中都迅速地封装了供 HTML 5 使用的分析器，IE（Internet Explorer）与 Opera 也在努力加快对 HTML 5 的支持——提高浏览器的兼容性指日可待。

接下来，让我们具体看一下在 HTML 5 中对语法进行了哪些改变。

2.1.2　HTML 5 中的标记方法

首先看一下 HTML 5 中的标记方法。

1. 内容类型（ContentType）

首先，HTML 5 的文件扩展符与内容类型保持不变。也就是说，扩展符仍然为“.html”或“.htm”，内容类型（ContentType）仍然为“text/html”。

2. DOCTYPE 声明

DOCTYPE 声明是 HTML 文件中必不可少的，它位于文件第一行。在 HTML 4 中，它的声明方法如下：

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

在 HTML 5 中，刻意不使用版本声明，一份文档将会适用所有版本的 HTML。HTML 5 中的 DOCTYPE 声明方法（不区分大小写）如下：

```
<!DOCTYPE html>
```

另外，当使用工具时，也可以在 DOCTYPE 声明方式中加入 SYSTEM 识别符，声明方法如下：

```
<!DOCTYPE HTML SYSTEM "about:legacy-compat">
```

在 HTML 5 中像这样的 DOCTYPE 声明方式是允许的（不区分大小写，引号不区分是单引号还是双引号）。

3. 指定字符编码

在 HTML 4 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">
```

在 HTML 5 中，可以使用对 <meta> 元素直接追加 charset 属性的方式来指定字符编码，如下所示：

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

两种方法都有效，可以继续使用前面一种方式（通过 content 元素的属性来指定），但是不能同时混合使用两种方式。在以前的网站代码中可能会存在下面代码所示的标记方式，但在 HTML 5 中，这种字符编码方式将被认为是错误的，这一点请注意。

```
<meta charset="UTF-8" http-equiv="Content-Type"
content="text/html;charset=UTF-8">
```

从 HTML 5 开始，对于文件的字符编码推荐使用 UTF-8。

2.1.3 HTML 5 确保的兼容性

HTML 5 的语法是为了保证与之前的 HTML 语法也能够达到最大程度的兼容而设计的。例如，符合“没有 <p> 的结束标记”的 HTML 代码随处可见，HTML 5 中并没有把这种情况作为错误来处理，而是允许存在这种情况，但明确规定了这种情况应该怎么处理。

那么，针对这个问题，我们从元素标记的省略、具有 boolean 值的属性、引号的省略这几方面来详细看一下在 HTML 5 中是如何确保与之前版本的 HTML 实现兼容的。

1. 可以省略标记的元素

在 HTML 5 中，元素的标记可以省略。具体来说，分为“不允许写结束标记”、“可以省略结束标记”和“开始标记和结束标记全部可以省略”三种类型。

接着，我们针对这三类情况列举一个元素清单，其中包括本书到现在为止还没有介绍的 HTML 5 中的新元素（关于这些新元素，2.2 节将进行介绍）。

- 不允许写结束标记的元素有：area、base、br、col、command、embed、hr、img、input、keygen、link、meta、param、source、track、wbr。
- 可以省略结束标记的元素有：li、dt、dd、p、rt、rp、optgroup、option、colgroup、thead、tbody、tfoot、tr、td、th。
- 可以省略全部标记的元素有：html、head、body、colgroup、tbody。

说明 “不允许写结束标记的元素”是指不允许使用开始标记与结束标记将元素括起来的形式，只允许使用“< 元素 />”的形式进行书写。例如“
...</br>”的书写方式是错误的，只允许“
”的书写形式。当然，HTML 5 之前的
 这种写法可以被沿用。

“可以省略全部标记的元素”是指该元素可以完全被省略。注意，即使标记被省略了，该元素还是以隐式的方式存在的。例如省略不写 body 元素时，在文档结构中它还是存在的，可以使用 document.body 访问。

2. 具有 boolean 值的属性

对于具有 boolean 值的属性，例如 disabled 与 readonly 等，当只写属性而不指定属性值时，表示属性值为 true，如果想要将属性值设为 false，则可以不使用该属性。另外，要想将属性值设定为 true 时，也可以将属性名设定为属性值，或将空字符串设定为属性值。

属性值的设定方法可以参考下面的代码示例：

```
<!-- 只写属性不写属性值代表属性为 true-->
<input type="checkbox" checked>
<!-- 不写属性代表属性为 false-->
<input type="checkbox">
<!-- 属性值 = 属性名，代表属性为 true-->
<input type="checkbox" checked="checked">
<!-- 属性值 = 空字符串，代表属性为 true-->
<input type="checkbox" checked="">
```

3. 省略引号

大家已经知道，在指定属性值的时候，属性值两边加引号时既可以用双引号，也可以用单引号。

HTML 5 在此基础上做了一些改进，当属性值不包括空字符串、“<”、“>”、“=”、单引号、双引号等字符时，属性值两边的引号可以省略。如下面的代码所示：

```
<!-- 请注意 type 的属性值两边的引号 -->
<input type="text">
<input type='text'>
<input type=text>
```

2.1.4　标记示例

现在，我们通过前面学到的 HTML 5 的语法知识来看一个关于 HTML 5 标记的示例。

代码清单 2-1 完全是用 HTML 5 写成的。其中省略了 <html>、<head>、<body> 等元素。可以通过这个示例复习一下 HTML 5 的 DOCTYPE 声明、用 <meta> 元素的 charset 属性指定字符编码、<p> 元素的结束标记的省略、使用 < 元素 /> 的方式来结束 <meta> 元素，以及

 元素等本节中所介绍的知识要点。

代码清单 2-1　HTML 5 标记示例

```
<!DOCTYPE html>
<meta charset="UTF-8">
<title>HTML 5 标记示例 </title>
<p> 这段代码是根据 HTML 5 语法
<br/> 编写出来的。
```

这段代码在 Firefox 4 浏览器中的运行结果如图 2-1 所示，另外，本书中如果没有特别说明使用什么浏览器，则使用的都是 Firefox 4 浏览器。

图 2-1　HTML 5 标记示例

2.2　新增的元素和废除的元素

本节将详细介绍 HTML 5 中新增和废除了哪些元素[㊀]。

2.2.1　新增的结构元素

在 HTML 5 中，新增以下与结构相关的元素。

（1）section 元素

section 元素表示页面中的一个内容区块，比如章节、页眉、页脚或页面中的其他部分。它可以与 h1、h2、h3、h4、h5、h6 等元素结合使用，标示文档结构。

HTML 5 中的代码示例：

```
<section>…</section>
```

HTML 4 中的代码示例：

```
<div>…</div>
```

（2）article 元素

article 元素表示页面中的一块与上下文不相关的独立内容，譬如博客中的一篇文章或报纸中的一篇文章。

HTML 5 中的代码示例：

```
<article>…</article>
```

HTML 4 中的代码示例：

㊀ 其他资料介绍的新增元素可能会比这里要多，这是因为 HTML 5 在最新发布的版本中又把这些本来想新增的元素删除了。

```
<div>…</div>
```

（3）aside 元素

aside 元素表示 article 元素的内容之外的、与 article 元素的内容相关的辅助信息。

HTML 5 中的代码示例：

```
<aside>…</aside>
```

HTML 4 中的代码示例：

```
<div>…</div>
```

（4）header 元素

header 元素表示页面中一个内容区块或整个页面的标题。

HTML 5 中的代码示例：

```
<header>…</header>
```

HTML 4 中的代码示例：

```
<div>…</div>
```

（5）footer 元素

footer 元素表示整个页面或页面中一个内容区块的脚注。一般来说，它会包含创作者的姓名、创作日期以及创作者联系信息。

HTML 5 中的代码示例：

```
<footer></footer>
```

HTML 4 中的代码示例：

```
<div>…</div>
```

（6）nav 元素

nav 元素表示页面中导航链接的部分。

HTML 5 中的代码示例：

```
<nav></nav>
```

HTML 4 中的代码示例：

```
<ul></ul>
```

（7）figure 元素

figure 元素表示一段独立的流内容，一般表示文档主体流内容中的一个独立单元。使用 figcaption 元素为 figure 元素组添加标题。

HTML 5 中的代码示例：

```
<figure>
<figcaption>PRC</figcaption>
<p>The People's Republic of China was born in 1949...</p>
</figure>
```

HTML 4 中的代码示例：

```
<dl>
<h1>PRC</h1>
<p>The People's Republic of China was born in 1949...</p>
</dl>
```

（8）main 元素

main 元素表示网页中的主要内容。主内容区域意指与网页标题或应用程序中本页面主要功能直接相关或进行扩展的内容。

HTML 5 中的代码示例：

```
<main>…</main>
```

HTML 4 中的代码示例：

```
<div>…</div>
```

2.2.2 新增的其他元素

除了结构元素外，在 HTML 5 中，还新增以下元素。

（1）video 元素

video 元素用于定义视频，比如电影片段或其他视频流。

HTML 5 中的代码示例：

```
<video src="movie.ogg" controls="controls">video 元素 </video>
```

HTML 4 中的代码示例：

```
<object type="video/ogg" data="movie.ogv">
       <param name="src" value="movie.ogv">
</object>
```

（2）audio 元素

audio 元素用于定义音频，比如音乐或其他音频流。

HTML 5 中的代码示例：

```
<audio src="someaudio.wav">audio 元素 </audio>
```

HTML 4 中的代码示例：

```
<object type="application/ogg" data="someaudio.wav">
```

```
    <param name="src" value="someaudio.wav">
</object>
```

（3）embed 元素

embed 元素用来插入各种多媒体，格式可以是 Midi、Wav、AIFF、AU、MP3 等。

HTML 5 中的代码示例：

```
<embed src="horse.wav" />
```

HTML 4 中的代码示例：

```
<object data="flash.swf"  type="application/x-shockwave-flash"></object>
```

（4）mark 元素

mark 元素主要用来在视觉上向用户呈现那些需要突出显示或高亮显示的文字。mark 元素的一个比较典型的应用就是在搜索结果中向用户高亮显示搜索关键词。

HTML 5 中的代码示例：

```
<mark></mark>
```

HTML 4 中的代码示例：

```
<span></span>
```

（5）progress 元素

progress 元素表示运行中的进程，可以使用 progress 元素来显示 JavaScript 中耗费时间的函数的进程。

HTML 5 中的代码示例：

```
<progress></progress>
```

这是 HTML 5 中的新增功能，故无法用 HTML 4 代码来实现。

（6）meter 元素

meter 元素表示度量衡，仅用于已知最大值和最小值的度量。必须定义度量的范围，既可以在元素的文本中，也可以在 min/max 属性中定义。

HTML 5 中的代码示例：

```
<meter></meter>
```

这是 HTML 5 中的新增功能，故无法用 HTML 4 代码来实现。

（7）time 元素

time 元素用于表示日期或时间，也可以同时表示两者。

HTML 5 中的代码示例：

```
<time></time>
```

HTML 4 中的代码示例：

```
<span></span>
```

（8）ruby 元素

ruby 元素表示 ruby 注释（中文注音或字符）。

HTML 5 中的代码示例：

```
<ruby>漢 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rp></rt></ruby>
```

这也是 HTML 5 中的新增功能。

（9）rt 元素

rt 元素表示字符（中文注音或字符）的解释或发音。

HTML 5 中的代码示例：

```
<ruby>漢 <rt> ㄏㄢˋ </rt></ruby>
```

这是 HTML 5 中的新增功能。

（10）rp 元素

rp 元素在 ruby 注释中使用，以定义不支持 ruby 元素的浏览器所显示的内容。

HTML 5 中的代码示例：

```
<ruby>漢 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rp></rt></ruby>
```

这是 HTML 5 中的新增功能。

（11）wbr 元素

wbr 元素表示软换行。wbr 元素与 br 元素的区别是：br 元素是此处必须换行，而 wbr 元素意思就是浏览器窗口或父级元素的宽度足够宽时（没必要换行时），不进行换行，而当宽度不够时，主动在此处进行换行。wbr 元素好像对字符型的语言用处比较大，但是对于中文，貌似没多大用处。

HTML 5 中的代码示例：

```
<p> To learn AJAX, you must be fami<wbr>liar with the XMLHttp<wbr>
Request Object. </p>
```

这是 HTML 5 中的新增功能。

（12）canvas 元素

canvas 元素表示图形，比如图表和其他图像。这个元素本身没有行为，仅提供一块画布，但把一个绘图 API 展现给客户端 JavaScript，以使脚本能够把想绘制的东西绘制到这块画布上。

HTML 5 中的代码示例：

```
<canvas id="myCanvas" width="200" height="200"></canvas>
```

HTML 4 中的代码示例：

```
<object data="inc/hdr.svg" type="image/svg+xml" width="200" height="200">
</object>
```

（13）command 元素

command 元素表示命令按钮，比如单选按钮、复选框或按钮。

HTML 5 中的代码示例：

```
<command onclick=cut()" label="cut">
```

这是 HTML 5 中的新增功能。

（14）details 元素

details 元素表示用户要求得到并且可以得到的细节信息。它可以与 summary 元素配合使用。summary 元素提供标题或图例。标题是可见的，用户点击标题时，会显示细节信息。summary 元素应该是 details 元素的第一个子元素。

HTML 5 中的代码示例：

```
<details>
   <summary>HTML5</summary>
   This document teaches you everything you have to learn about HTML5.
</details>
```

这是 HTML 5 中的新增功能。

（15）datalist 元素

datalist 元素表示可选数据的列表，与 input 元素配合使用，可以制作出输入值的下拉列表。

HTML 5 中的代码示例：

```
<datalist></datalist>
```

这是 HTML 5 中的新增功能。

（16）datagrid 元素

datagrid 元素表示可选数据的列表，它以树形列表的形式显示。

HTML 5 中的代码示例：

```
<datagrid></datagrid>
```

这是 HTML 5 中的新增功能。

（17）keygen 元素

keygen 元素表示生成密钥。

HTML 5 中的代码示例：

```
<keygen>
```

这是 HTML 5 中的新增功能。

（18）output 元素

output 元素表示不同类型的输出，比如脚本的输出。

HTML 5 中的代码示例：

```
<output></output>
```

HTML 4 中的代码示例：

```
<span></span>
```

（19）source 元素

source 元素为媒介元素（比如 <video> 和 <audio>）定义媒介资源。

HTML 5 中的代码示例：

```
<source>
```

HTML 4 中的代码示例：

```
<param>
```

（20）menu 元素

menu 元素表示菜单列表。当希望列出表单控件时使用该标签。

HTML 5 中的代码示例：

```
<menu>
  <li><input type="checkbox" />Red</li>
  <li><input type="checkbox" />blue</li>
</menu>
```

在 HTML 4 中，menu 元素不被推荐使用。

（21）dialog 元素

dialog 元素表示对话框。

HTML 5 中的代码示例：

```
<dialog></dialog>
```

2.2.3 新增的 input 元素的类型

HTML 5 中新增很多 input 元素的类型，现列举如下：

- email：email 类型表示必须输入 e-mail 地址的文本输入框。
- url：url 类型表示必须输入 URL 地址的文本输入框。
- number：number 类型表示必须输入数值的文本输入框。

❑ range：range 类型表示必须输入一定范围内数字值的文本输入框。
❑ Date Pickers：HTML 5 拥有多个可供选取日期和时间的新型输入文本框。
- date：选取日、月、年。
- month：选取月、年。
- week：选取周和年。
- time：选取时间（小时和分钟）。
- datetime：选取时间、日、月、年（UTC 时间）。
- datetime-local：选取时间、日、月、年（本地时间）。

2.2.4　废除的元素

由于各种原因，在 HTML 5 中废除了很多元素，简单介绍如下。

1. 能使用 CSS 替代的元素

对于 basefont、big、center、font、s、strike、tt、u 等元素，由于它们的功能都是纯粹为画面展示服务的，而 HTML 5 中提倡把画面展示性功能统一放在 CSS 样式表中统一编辑，所以将这些元素废除，使用编辑 CSS、添加 CSS 样式表的方式进行替代。其中 font 元素允许由“所见即所得”的编辑器进行插入，s 元素、strike 元素可以由 del 元素进行替代，tt 元素可以由 CSS 的 font-family 属性进行替代。

2. 不再使用 frame 框架

对于 frameset 元素、frame 元素与 noframes 元素，由于 frame 框架对网页可用性存在负面影响，在 HTML 5 中已不支持 frame 框架，只支持 iframe 框架，或者由服务器方创建的由多个页面组成的复合页面的形式，同时将这三个元素废除。

3. 只有部分浏览器支持的元素

对于 applet、bgsound、blink、marquee 等元素，由于只有部分浏览器支持这些元素，特别是 bgsound 元素以及 marquee 元素，只被 Internet Explorer 所支持，所以在 HTML 5 中被废除。其中 applet 元素可由 embed 元素或 object 元素进行替代，bgsound 元素可由 audio 元素进行替代，marquee 可以由 JavaScript 编程的方式所替代。

4. 其他被废除的元素

其他被废除元素还有：
❑ 废除 rb 元素，使用 ruby 元素替代。
❑ 废除 acronym 元素，使用 abbr 元素替代。
❑ 废除 dir 元素，使用 ul 元素替代。
❑ 废除 isindex 元素，使用 form 元素与 input 元素相结合的方式替代。
❑ 废除 listing 元素，使用 pre 元素替代。

- 废除 xmp 元素，使用 code 元素替代。
- 废除 nextid 元素，使用 GUIDS 替代。
- 废除 plaintext 元素，使用 ″text/plian″ MIME 类型替代。

2.3 新增的属性和废除的属性

在 HTML 5 中，在增加和废除很多元素的同时，也增加和废除了很多属性，本节对于这些增加和废除的属性进行简单介绍[㊀]。

2.3.1 新增的属性

1. 表单相关的属性

新增的与表单相关的元素如下：

- 可以对 input（type=text）、select、textarea 与 button 元素指定 autofocus 属性。它以指定属性的方式让元素在画面打开时自动获得焦点。
- 可以对 input 元素（type=text）与 textarea 元素指定 placeholder 属性，它会对用户的输入进行提示，提示用户可以输入的内容。
- 可以对 input、output、select、textarea、button 与 fieldset 指定 form 属性，声明它属于哪个表单，然后将其放置在页面上任何位置，而不是表单之内。
- 可以对 input 元素（type=text）与 textarea 元素指定 required 属性。该属性表示在用户提交的时候进行检查，检查该元素内一定要有输入内容。
- 为 input 元素增加了几个新的属性：autocomplete、min、max、multiple、pattern 与 step。同时还有一个新的 list 元素与 datalist 元素配合使用。datalist 元素与 autocomlete 属性配合使用。multiple 属性允许在上传文件时一次上传多个文件。
- 为 input 元素与 button 元素增加了新属性 formaction、formenctype、formmethod、formnovalidate 与 formtarget，它们可以重载 form 元素的 action、enctype、method、novalidate 与 target 属性。为 fieldset 元素增加了 disabled 属性，用于把它的子元素设为 disabled（无效）状态。
- 为 input 元素、button 元素、form 元素增加了 novalidate 属性，该属性可以取消提交时进行的有关检查，表单可以被无条件提交。
- 为所有可使用标签（label 元素）的表单元素（包括非隐藏的 input 元素（type 属性值不等于 hidden)、button 元素、select 元素、textarea 元素、meter 元素、output 元素、progress 元素以及 keygen 元素）定义一个 labels 属性，属性值为一个 NodeList 对象，代表该元素所绑定的标签元素所构成的集合。

㊀ 其他资料介绍的新增属性可能会比本节要介绍的更多，这是因为 HTML 5 在最新发布的版本中又把这些本来想新增的属性删除了。

- 可以在标签（label 元素）内部放置一个表单元素，并且通过该标签的 control 属性访问该表单元素。
- 针对 input 元素与 textarea 元素，在 HTML 5 中增加 SelectionDirection 属性。当用户在这两个元素中用鼠标选取部分文字时，可以使用该属性来获取选取方向。当用户正向选取文字时，该属性值为“forward”，当用户反向选取文字时，该属性值为“backward”。当用户没有选取任何文字时，该属性值为“forward”。
- 对复选框（checkbox 元素）添加 indeterminate 属性，以说明复选框处于“尚未明确是否选取”状态。
- 对类型为 image 的 input 元素添加用于指定图片按钮中图片高度的 height 属性与图片宽度的 width 属性。
- 对 textarea 元素新增用于限定可输入文字个数的 maxlength 属性与用于指定表单提交时是否在文字换行处添加换行符的 wrap 属性。

2. 链接相关的属性

新增的与链接相关的属性如下：

- 为 a 与 area 元素增加了 media 属性、download 属性以及 ping 属性，其中 media 属性规定目标 URL 是为什么类型的媒介 / 设备进行优化的，download 属性用于让用户下载目标链接所指向的资源，而不是直接打开该目标链接，这些属性均只能在 href 属性存在时使用。
- 为 area 元素增加了 hreflang 属性与 rel 属性，以保持与 a 元素、link 元素的一致。
- 为 link 元素增加了新属性 sizes。该属性可以与 icon 元素结合使用（通过 rel 属性），该属性指定关联图标（icon 元素）的大小。
- 为 base 元素增加了 target 属性，主要目的是保持与 a 元素的一致性。

3. 其他属性

除了上面介绍的与表单和链接相关的属性外，HTML 5 还增加了下面的属性：

- 为 ol 元素增加 start 属性与 reversed 属性，其中 start 属性定义列表的开始编号，reversed 属性指定列表倒序显示。
- 为 meta 元素增加 charset 属性，因为这个属性已经得到广泛支持，而且为文档的字符编码的指定提供了一种比较良好的方式。
- 为 menu 元素增加了两个新的属性——type 与 label。label 属性为菜单定义一个可见的标注，type 属性让菜单可以以上下文菜单、工具条、与列表菜单三种形式出现。
- 为 style 元素增加 scoped 属性，用来规定样式的作用范围，譬如只对页面上某个树起作用。
- 为 script 元素增加 async 属性，它定义脚本是否异步执行。
- 为 html 元素增加属性 manifest，开发离线 Web 应用程序时它与 API 结合使用，定义

一个 URL，在这个 URL 上描述文档的缓存信息。

- 为 iframe 元素增加三个属性——sandbox、seamless 与 srcdoc，用来提高页面安全性，防止不信任的 Web 页面执行某些操作。

2.3.2 废除的属性

HTML 4 中的一些属性在 HTML 5 中不再使用，而是采用其他属性或其他方案进行替代，具体如表 2-1 所示。

表 2-1 HTML 5 中废除的属性

HTML 4 中使用的属性	使用该属性的元素	HTML 5 中的替代方案
rev	link，a	rel
charset	link，a	在被链接的资源中使用 HTTP Content-type 头元素
shape、coords	a	使用 area 元素代替 a 元素
longdesc	img，iframe	使用 a 元素链接到较长描述
target	link	多余属性，被省略
nohref	area	多余属性，被省略
profile	head	多余属性，被省略
version	html	多余属性，被省略
name	img	id
scheme	meta	只为某个表单域使用 scheme
archive、classid、codebase、codetype、declare、standby	object	使用 data 与 type 属性类调用插件。需要使用这些属性来设置参数时，使用 param 属性
valuetype、type	param	使用 name 与 value 属性，不声明值的 MIME 类型
axis、abbr	td，th	使用以明确简洁的文字开头，后跟详述文字的形式。可以对更详细内容使用 title 属性来使单元格的内容变得简短
scope	td	在被链接的资源中使用 HTTP Content-type 头元素
align	caption，input，legend，div，h1，h2，h3，h4，h5，h6，p	使用 CSS 样式表进行替代
alink、link、text，vlink、background、bgcolor	body	使用 CSS 样式表进行替代
align、bgcolor、border、cellpadding、cellspacing、frame、rules、width	table	使用 CSS 样式表进行替代
align、char、charoff、height、nowrap、valign	tbody，thead，tfoot	使用 CSS 样式表进行替代
align、bgcolor、char、charoff、height、nowrap、valign、width	td，th	使用 CSS 样式表进行替代
align、bgcolor、char、charoff、valign	tr	使用 CSS 样式表进行替代

（续）

HTML 4 中使用的属性	使用该属性的元素	HTML 5 中的替代方案
align、char、charoff、valign、width	col，colgroup	使用 CSS 样式表进行替代
align、border、hspace、vspace	object	使用 CSS 样式表进行替代
clear	br	使用 CSS 样式表进行替代
compact、type	ol，ul，li	使用 CSS 样式表进行替代
compact	dl	使用 CSS 样式表进行替代
compact	menu	使用 CSS 样式表进行替代
width	pre	使用 CSS 样式表进行替代
align、hspace、vspace	img	使用 CSS 样式表进行替代
align、noshade、size、width	hr	使用 CSS 样式表进行替代
align、frameborder、scrolling、marginheight、marginwidth	iframe	使用 CSS 样式表进行替代
autosubmit	menu	

2.4　全局属性

在 HTML 5 中，新增了一个“全局属性”的概念。所谓全局属性，是指可以对任何元素都使用的属性，本节将详细介绍几种常用的全局属性。

2.4.1　contentEditable 属性

contentEditable 是由微软开发的，被其他浏览器反编译并投入应用的一个全局属性。该属性的主要功能是允许用户编辑元素中的内容，所以该元素必须是可以获得鼠标焦点的元素，而且在点击鼠标后要向用户提供一个插入符号，提示用户该元素中的内容允许编辑。contentEditable 属性是一个布尔值属性，可以被指定为 true 或 false。

除此之外，该属性还有个隐藏的 inherit（继承）状态，当属性值为 true 时，元素被指定为允许编辑；当属性值为 false 时，元素被指定为不允许编辑；当未指定 true 或 false 时，则由 inherit 状态来决定，如果元素的父元素是可编辑的，则该元素就是可编辑的。

另外，除了 contentEditable 属性外，元素还具有一个 isContentEditable 属性，当元素可编辑时，该属性值为 true；当元素不可编辑时，该属性值为 false。

代码清单 2-2 中给出了一个使用 contentEditable 属性的示例，当列表元素被加上 contentEditable 属性后，该元素就变成可编辑的了，读者可自行在浏览器中对该示例进行试验。

代码清单 2-2　contentEditable 属性示例

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
```

```
<title>conentEditalbe 属性示例 </title>
</head>
<h2> 可编辑列表 </h2>
<ul contentEditable="true">
<li> 列表元素 1</li>
<li> 列表元素 2</li>
<li> 列表元素 3</li>
</ul>
```

这段代码运行后的结果如图 2-2 所示。

图 2-2　可编辑列表示例

在编辑完元素中的内容后，要想保存其中内容，只能把该元素的 innerHTML 发送到服务器端进行保存，因为改变元素内容后该元素的 innerHTML 内容也会随之改变，目前还没有特别的 API 来保存编辑后元素中的内容。

最后，在这里列举一下支持 contentEditable 属性的元素：defaults、A、ABBR、ACRONYM、ADDRESS、B、BDO、BIG、BLOCKQUOTE、BODY、BUTTON、CENTER、CITE、CODE、CUSTOM、DD、DEL、DFN、DIR、DIV、DL、DT、EM、FIELDSET、FONT、FORM、hn、I、INPUT type=button、INPUT type=password、INPUT type=radio、INPUT type=reset、INPUT type=submit、INPUT type=text、INS、ISINDEX、KBD、LABEL。

2.4.2　designMode 属性

designMode 属性用来指定整个页面是否可编辑，当页面可编辑时，页面中任何支持上文所述的 contentEditable 属性的元素都变成了可编辑状态。designMode 属性只能在 JavaScript 脚本中被编辑修改。该属性有两个值——“ on”与“ off”。当该属性值被指定为“ on”时，页面可编辑；当该属性值被指定为“ off”时，页面不可编辑。使用 JavaScript 脚本来指定 designMode 属性的方法如下所示：

```
document.designMode="on"
```

针对 designMode 属性，各浏览器的支持情况也各不相同：

- IE 8：出于安全考虑，不允许使用 designMode 属性让页面进入编辑状态。
- IE 9：允许使用 designMode 属性让页面进入编辑状态。
- Chrome 3 和 Safari：使用内嵌 frame 的方式，该内嵌 frame 是可编辑的。
- Firefox 和 Opera：允许使用 designMode 属性让页面进入编辑状态。

2.4.3　hidden 属性

在 HTML 5 中，所有的元素都允许使用一个 hidden 属性。该属性类似于 input 元素中的 hidden 元素，功能是通知浏览器不渲染该元素，使该元素处于不可见状态。但是元素中的内容还是被浏览器创建的，也就是说页面装载后允许使用 JavaScript 脚本将该属性取消，取消后该元素变为可见状态，同时元素中的内容也即时显示出来。hidden 属性是一个布尔值的属性，当设为 true 时，元素处于不可见状态；当设为 false 时，元素处于可见状态。

2.4.4　spellcheck 属性

spellcheck 属性是 HTML 5 针对 input 元素（type=text）与 textarea 这两个文本输入框提供的一个新的属性，它的功能对用户输入的文本内容进行拼写和语法检查。spellcheck 属性是一个布尔值的属性，具有 true 或 false 两种值。但是它在书写时有一个特殊的地方，就是必须明确声明属性值为 true 或 false，书写方法如下所示：

```
<!—以下两种书写方法正确 -->
<textarea spellcheck="true" >
<input type=text spellcheck=false>
<!—以下书写方法为错误 -->
<textarea spellcheck >
```

需要注意的是，如果元素的 readOnly 属性或 disabled 属性设为 true，则不执行拼写检查。

目前除了 IE 之外，Firefox、Chrome、Safari、Opera 等浏览器都对该属性提供了支持。

图 2-3 为 Opera 浏览器中 spellcheck 属性的表现形式。

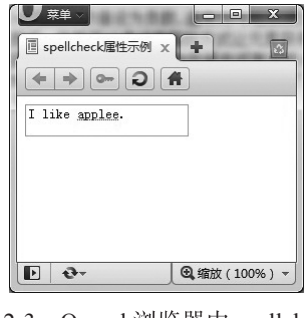


图 2-3　Operal 浏览器中 spellcheck 的属性示例

2.4.5　tabindex 属性

tabindex 是开发中的一个基本概念，当不断敲击 Tab 键让窗口或页面中的控件获得焦点，当对窗口或页面中的所有控件进行遍历的时候，每一个控件的 tabindex 表示该控件是第几个被访问到的。

过去这个属性在编辑网页时是非常有用的，但如今控件的遍历顺序是由元素在页面上所处位置决定的，所以就不再需要了。

但是 tabindex 还有另外一个作用，在默认情况下，只有链接元素与表单元素可以通过按

键获得焦点。如果对其他元素使用 tabindex 属性后，也能让该元素获得焦点，那么当脚本中执行 focus() 语句的时候，就可以让该元素获得焦点。但这样做会产生一个副作用：该元素也可以通过按 Tab 键获得焦点，而这有时可能不是开发者想要的结果。

把元素的 tabindex 值设为负数（通常为 –1）后就可以解决这个问题。tabindex 值为负数后，仍然可以通过编程的方式让元素获得焦点，但按下 Tab 键时该元素就不能获得焦点，这在复杂的页面中或复杂的 Web 应用程序中是十分有用的。在 HTML 4 中，–1 是一个无用的属性值，但到了 HTML 5 中，通过巧妙运用可以让该属性值获得极大应用。

2.5 新增的事件

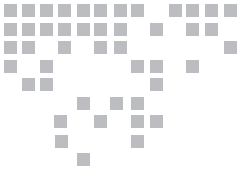
HTML 5 中对页面、表单、键盘元素新增了各种事件，如表 2-2 所示（在 HTML 5 的新增 API 中使用到的部分事件将在介绍这些 API 时介绍）。

表 2-2 HTML 5 中对页面、表单、键盘元素新增的各种事件

元素或对象	事件	触发时机	代码示例
window 对象 body 元素	beforeprint	即将开始打印之前触发	<body onbeforeprint="alert(' 即将打印 ');">
	afterprint	打印完毕时触发	<body onafterprint=" alert(' 打印完毕 ');">
	resize	浏览器窗口大小发生改变时触发	<body onresize="alert(' 用户调整页面尺寸 ');">
	error	页面加载出错时触发	<body onerror=" alert(' 页面加载出错 ');">
	offline	页面变为离线状态时触发	<body onoffline=" alert(' 页面变为离线状态 ');">
	online	页面变为在线状态时触发	<body ononline=" alert(' 页面变为在线状态 ');">
	pageshow	页面加载时触发，类似于 load 事件，区别在于 load 事件在页面第一次加载时触发，而 pageshow 事件在每一次加载时触发，即从网页缓存中读取页面时只触发 pageshow 事件，不触发 load 事件	<body onpageshow=" alert(' 页面被加载 ');">
	beforeunload	当前页面被关闭时触发，该事件通知浏览器显示一个用于询问用户是否确实离开本页面的确认窗口，可以设置该窗口中的提示文字 在 Firefox 浏览器的确认窗口中只显示浏览器给定的默认询问信息，开发者不能对其进行修改	<body onbeforeunload="return ' 是否确实需要离开本页面？ ';">
	hashchange	当页面 URL 地址字符串中的哈希部分（# 后面的部分）发生改变时触发	<body onhashchange="alert('URL 的 hash 部分发生改变 ');"> 测试链接

（续）

元素或对象	事件	触发时机	代码示例
任何元素	mousewheel	当用户鼠标指针悬停在元素上并滚动鼠标滚轮时触发	<body onmousewheel="alert(' 用户滚动鼠标滚轮 ');">
任何容器元素	scroll	当元素滚动条被滚动时触发	<body onscoll="alert(' 元素被滚动 ');">
input 元素 textarea 元素	input	当用户修改文本框中内容时触发，input 事件与 change 事件的区别为 input 事件在元素尚未失去焦点时已触发，change 事件只在元素失去焦点时触发	<input type="text" oninput="alert(' 元素内容被修改 ');">
form 元素	reset	当用户按下表单元素中的 type 类型为 reset 的 input 元素或 JavaScript 脚本代码中执行表单对象的 reset 方法时触发	<form id="form1" onreset="alert(' 表单被重置 ');">



Chapter 3 第 3 章

HTML 5 的结构

在 HTML 5 对 HTML 4 所做的各方面修改中，一个比较重大的修改就是为了使文档结构更加清晰明确，容易阅读，增加了很多新的结构元素。本章详细介绍这些新增的结构元素，涉及它们的定义、使用方法以及使用示例，最后再介绍一下在 HTML 5 中，究竟怎样将这些新增的结构元素结合使用。

学习内容：

- 掌握 HTML 5 中新增的主体结构元素的定义以及使用方法、使用场合。新增的主体结构元素包括 article 元素、section 元素、nav 元素以及 aside 元素。
- 掌握 HTML 5 中新增的非主体结构元素的定义以及使用方法、使用场合。新增的非主体结构元素包括 header 元素、hgroup 元素、footer 元素以及 address 元素。
- 掌握在 HTML 5 中应该怎样结合运用这些新增结构元素来合理编排页面总体布局，掌握什么是显式编排，什么是隐式编排，HTML 5 分析器是按什么原则来分析页面结构的，以及怎样对这些新增元素使用 CSS 样式。

3.1 新增的主体结构元素

在 HTML 5 中，为了使文档的结构更加清晰明确，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。需要说明的是，本章所讲的内容区块是指将 HTML 页面按逻辑进行分割后的单位。例如，对书籍来说，章、节都可以称为内容区块；对博客网站来说，导航菜单、文章正文、文章的评论等每一个部分都可称为内容区块。

接下来将详细讲解 HTML 5 在页面的主体结构方面新增加的结构元素。

3.1.1　article 元素

article 元素代表文档、页面或应用程序中独立的、完整的、可以独自被外部引用的内容。它可以是一篇博客或报章杂志中的文章、一篇论坛帖子、一段用户评论或一个独立的插件，或者其他任何独立的内容。

除了内容部分，一个 article 元素通常有它自己的标题（一般放在一个 header 元素里面），有时还有自己的脚注。

现在，以博客为例来看一段关于 article 元素的代码示例，如代码清单 3-1 所示。

代码清单 3-1　article 元素示例

```
<article>
    <header>
        <h1> 苹果 </h1>
        <p> 发表日期 : <time pubdate="pubdate">2010/10/09</time></p>
    </header>
    <p><b> 苹果 </b> ，植物类水果，多次花果 ...(" 苹果 " 文章正文 )</p>
    <footer>
        <p><small> 著作权归 *** 公司所有。</small></p>
    </footer>
</article>
```

这个示例展示的是一篇讲述苹果的博客文章，在 header 元素中嵌入文章的标题部分，在这部分中，文章的标题“苹果”被镶嵌在 h1 元素中，文章的发表日期镶嵌在 p 元素中。在标题下部的 p 元素中，嵌入一大段该博客文章的正文，在结尾处的 footer 元素中，嵌入文章的著作权作为脚注。整个示例的内容相对比较独立、完整，因此，对这部分内容使用 article 元素。

article 元素是可以嵌套使用的，内层的内容在原则上需要与外层内容相关联。例如，一篇博客文章中，针对该文章的评论就可以使用嵌套 article 元素的方式；用来呈现评论的 article 元素被包含在表示整体内容的 article 元素里面。

接下来看一个关于 article 元素嵌套的代码示例，如代码清单 3-2 所示。

代码清单 3-2　article 元素嵌套示例

```
<article>
    <header>
        <h1> 苹果 </h1>
        <p> 发表日期 :
                <time pubdate datetime="2010/10/09">2010/10/09</time>
        </p>
    </header>
    <p><b> 苹果 </b> ，植物类水果，多次花果 ...(" 苹果 " 文章正文 )</p>
    <section>
        <h2> 评论 </h2>
        <article>
```

```
            <header>
                <h3> 发表者：陆凌牛 </h3>
                <p>
                    <time pubdate datetime="2010-10-10T19:10-08:00">
                        1 小时前
                    </time>
                </p>
            </header>
            <p> 我喜欢苹果，我最喜爱的品种是红富士。</p>
        </article>
        <article>
            <header>
                <h3> 发表者：张玉 </h3>
                <p>
                    <time pubdate datetime="2010-10-10T19:15-08:00">
                        1 小时前
                    </time>
                </p>
            </header>
            <p> 苹果？我不喜欢，我喜欢吃橘子。</p>
        </article>
    </section>
</article>
```

这个示例中的内容比代码清单 3-1 中的内容更加完整，它添加了文章读者的评论内容，整体内容还是比较独立、完整的，因此对其使用 article 元素。具体来说，该部分内容又分为几部分，把文章标题放在 header 元素中，把文章正文放在 header 元素后面的 p 元素中，然后用 section 元素把正文与评论部分进行区分。后面马上会介绍 section 元素，它是一个分块元素，用来对页面中的内容进行分块，在 section 元素中嵌入评论的内容，评论中有几个人的评论，每个人的评论部分相对来说又是比较独立、完整的，因此对它们都使用一个 article 元素，在评论的 article 元素中，又可以分为标题与评论内容部分，分别放在 header 元素与 p 元素中。

关于示例中提到的 time 元素与 pubdate 属性，可查看本节结尾处关于 time 元素与 pubdate 属性的说明。

另外，article 元素也可以用来表示插件，它的作用是使插件看起来好像内嵌在页面中一样。以下为它的一个示例，如代码清单 3-3 所示。

代码清单 3-3　用 article 元素表示插件

```
<article>
    <h1>My Fruit Spinner</h1>
    <object>
        <param name="allowFullScreen" value="true">
        <embed src="#" width="600" height="395"></embed>
    </object>
</article>
```

3.1.2　section 元素

section 元素用来对网站或应用程序中页面上的内容进行分块，一个 section 元素通常由内容及其标题组成。但 section 元素并非一个普通的容器元素；当一个容器需要直接定义样式或通过脚本定义行为时，推荐使用 div 元素而非 section。

我们可以这样理解：section 元素中的内容可以单独存储到数据库中或输出到 word 文档中。以下为它的一个示例（注意，标题部分位于它的内部，而不是它的前面），如代码清单 3-4 所示。

代码清单 3-4　section 元素示例

```
<section>
    <h1> 苹果 </h1>
    <p><b> 苹果 </b> ，植物类水果，多次花果 ...(" 苹果 " 文章正文 )</p>
</section>
```

通常不推荐为那些没有标题的内容使用 section，可以使用 HTML 5 轮廓工具来检查页面中是否有不包含标题部分的 section，HTML 5 轮廓工具的网址为“ http://gsnedders.html5.org/outliner/”，如果使用该工具进行检查后，会看见对于某个 section 的说明中有“ untitiled section”（没有标题的 section）文字，这个 section 就有可能使用不当（但是 nav 元素或 aside 元素没有标题是合理的）。

section 元素的作用是对页面上的内容进行分块，或者说对文章进行分段，不要将它与表示“有着自己的完整的、独立的内容”的 article 元素混淆。

下面来看两个 article 元素与 section 元素结合使用的示例，希望能够帮助你更好地理解 article 元素与 section 元素的区别。

首先来看一个带有 section 元素的 article 元素的示例，如代码清单 3-5 所示。

代码清单 3-5　带有 section 元素的 article 元素示例

```
<article>
    <h1> 苹果 </h1>
    <p><b> 苹果 </b> ，植物类水果，多次花果 ...</p>
    <section>
        <h2> 红富士 </h2>
        <p> 红富士是从普通富士的芽（枝）变中选育出的着色系富的统称 ...</p>
    </section>
    <section>
        <h2> 国光 </h2>
        <p> 国光苹果品，又名小国光、万寿。原产美国，1600 年发现的偶然实生苗 ...</p>
    </section>
</article>
```

代码清单 3-5 中的内容首先是一段独立的、完整的内容，因此使用 article 元素。该内容是一篇关于苹果的文章，该文章分为 3 段，每一段都有一个独立的标题，因此使用了两

个 section 元素。记住，对文章分段的工作也是使用 section 元素完成的。这里有人会问，为什么没有对第一段使用 section 元素，其实是可以使用 section 元素的，但是由于这里的结构比较清晰，分析器是可以识别第一段内容在一个 section 元素里的，所以也可以将第一个 section 元素略去不写，但是如果第一个 section 元素里还要包含子 section 元素或子 article 元素，那么必须写明第一个 section 元素。

接下来看一个包含 article 元素的 section 元素的示例，如代码清单 3-6 所示。

代码清单 3-6　包含 article 元素的 section 元素示例

```
<section>
    <h1> 水果 </h1>
    <article>
        <h2> 苹果 </h2>
        <p> 苹果，植物类水果，多次花果 ...</p>
    </article>
    <article>
        <h2> 橘子 </h2>
        <p> 橘子，是芸香科柑橘属的一种水果 ...</p>
    </article>
    <article>
        <h2> 香蕉 </h2>
        <p> 香蕉，属于芭蕉科芭蕉属植物，又指其果实，热带地区广泛栽培食用 ...</p>
    </article>
</section>
```

这个示例比前面的示例复杂了一些，首先，它是一篇文章中的一段，因此没有使用 article 元素。但是，在这一段中，可以分为几块独立的内容，因此嵌入了几个独立的 article 元素。

看到这里，你可能又糊涂了，这两个元素可以互换使用吗？它们的区别到底是什么呢？事实上，在 HTML 5 中，article 元素可以看成是一种特殊种类的 section 元素，它比 section 元素更强调独立性。section 元素强调分段或分块，而 article 强调独立性，具体来说，如果一块内容相对来说比较独立、完整的时候，应该使用 article 元素，但是如果想将一块内容分成几段的时候，应该使用 section 元素进行分段。另外，在 HTML 5 中，div 元素变成了一种容器，当使用 CSS 样式的时候，可以对这个容器进行一个总体的 CSS 样式的套用。

另外再补充一点，在 HTML 5 中，可以将所有页面的从属部分，譬如导航条、菜单、版权说明等包含在一个统一的页面中，以便统一使用 CSS 样式进行装饰。

最后，关于 section 元素的使用禁忌总结如下：

- ❑ 不要将 section 元素用作设置样式的页面容器，因为那是 div 元素的工作。
- ❑ 如果 article 元素、aside 元素或 nav 元素更符合状况，不要使用 section 元素。
- ❑ 不要为没有标题的内容区块使用 section 元素。

3.1.3　nav 元素

nav 元素是一个可以用来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。并不是所有的链接组都要放进 nav 元素，只需要将主要的、基本的链接组放进 nav 元素即可。例如，在页脚中通常会有一组链接，包括服务条款、首页、版权声明等，这时使用 footer 元素是最恰当的。一个页面中可以拥有多个 nav 元素，作为页面整体或不同部分的导航。

接着让我们来看一个 nav 元素的使用示例，在这个示例中，一个页面由几个部分组成，每个部分都带有链接，但只将最主要的链接放入 nav 元素中，如代码清单 3-7 所示。

代码清单 3-7　nav 元素示例

```
<body>
<h1> 技术资料 </h1>
<nav>
    <ul>
        <li><a href="/"> 主页 </a></li>
        <li><a href="/events"> 开发文档 </a></li>
        ...more...
    </ul>
</nav>
<article>
   <header>
        <h1>HTML 5 与 CSS 3 的历史 </h1>
        <nav>
            <ul>
                <li><a href="#HTML5">HTML 5 的历史 </a></li>
                <li><a href="#CSS3">CSS 3 的历史 </a></li>
                ...more...
            </ul>
        </nav>
    </header>
    <section id="HTML 5">
        <h1>HTML 5 的历史 </h1>
        <p> 讲述 HTML 5 的历史的正文 </p>
    </section>
    <section id="CSS 3">
        <h1>CSS 3 的历史 </h1>
        <p> 讲述 CSS 3 的历史的正文 </p>
    </section>
    ...more...
    <footer>
        <p>
            <a href="?edit"> 编辑 </a> |
            <a href="?delete"> 删除 </a> |
            <a href="?rename"> 重命名 </a>
        </p>
    </footer>
```

```
</article>
<footer>
    <p><small> 版权所有：陆凌牛 </small></p>
</footer>
</body>
```

在这个例子中，第一个 nav 元素用于页面导航，将页面跳转到其他页面上（跳转到网站主页或开发文档目录页面）；第二个 nav 元素被放置在 article 元素中，用来实现在这篇文章中的两个组成部分的页内导航。

具体来说，nav 元素可以用在以下场合：

❑ 传统导航条

现在主流网站上都有不同层级的导航条，用来将当前画面跳转到网站的其他主要页面。可以用 nav 元素实现导航条。

❑ 侧边栏导航

现在主流博客网站及商品网站上都有侧边栏导航，用来将页面从当前文章或当前商品跳转到其他文章或其他商品页面。可以用 nav 元素实现侧边栏导航。

❑ 页内导航

可以用 nav 元素实现页内导航，用来在本页面几个主要组成部分之间跳转。

❑ 翻页操作

nav 元素可以用在多个页面的前后页或博客网站的前后篇文章的滚动中。

除此之外，nav 元素也可以用在其他所有你觉得重要的、基本的导航链接组中。

注意，在 HTML 5 中不要用 menu 元素代替 nav 元素。过去很多 Web 应用程序的开发人员喜欢用 menu 元素进行导航，有必要再次强调的是，menu 元素是被用在一系列发出命令的菜单上的，是一种交互性的元素，或者更确切地说是使用在 Web 应用程序中的。

3.1.4 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。

1）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参考资料、名词解释等。

这部分代码如代码清单 3-8 所示。

代码清单 3-8　文章内部的 aside 元素示例

```
<!DOCTYPE html>
<head>
<meta charset="utf-8">
<title>aside 元素示例 </title>
```

```
</head>
<body>
<header>
    <h1>F# 入门 </h1>
</header>
<article>
    <h1> 第四节　词法闭包 </h1>
    <p>lambda 表达式可以创建词法闭包 ...( 文章正文 )</p>
    <aside>
        <!-- 因为这个 aside 元素被放置在一个 article 元素内部，
        所以分析器将这个 aside 元素的内容理解成是和 article 元素的内容相关联的。 -->
        <h1> 名词解释 </h1>
        <dl>
            <dt>F#</dt>
            <dd>F# 为 .Net2010 中引入的新型函数型编程语言 </dd>
        </dl>
        <dl>
            <dt> 词法闭包 </dt>
            <dd> 词法闭包是指，将创建 lambda 表达式时的环境保存起来 ...( 详细解释 )</dd>
       </dl>
    </aside>
</article>
</body>
```

程序运行结果如图 3-1 所示。

图 3-1　aside 元素示例

这是笔者博客网页中的一篇文章，网页的标题放在 header 元素中，在 header 元素的后面将所有关于文章的部分放在一个 article 元素中，将文章的正文部分放在一个 p 元素中，但是该文章还有一个名词解释的附属部分，用来解释该文章中的一些名词，因此，在 p 元素的下部又放置了一个 aside 元素，用来存放名词解释部分的内容。

2）在 article 之外元素使用，作为页面或站点全局的附属信息部分。最典型的形式是侧边栏，其中的内容可以是友情链接、博客中其他文章列表或广告单元等。

下面这个示例为标准博客网页中一个侧边栏的示例，示例中的“IT 新技术”为博客的名称，如代码清单 3-9 所示。

代码清单 3-9　侧边栏示例

```
<aside>
    <nav>
        <h2> 评论 </h2>
        <ul>
            <li>
            <a href="http://blog.sina.com.cn/1683">erway</a>    10-24 14:25
            </li>
            <li>
            <a href="http://blog.sina.com.cn/u/1345"> 太阳雨 </a>   10-22 23:48<br/>
            <a href="http://blog.sina.com.cn/s/blog_6a9kv8f.html#comment">
                顶，拜读一下老牛的文章
            </a>
            </li>
            <li>
            <a href="http://blog.sina.com.cn/u/1259295385"> 新浪官博 </a>
            08-12 08:50<br/>
            <a href="#"> 恭喜！您已经成功开通了博客 </a>
            </li>
        </ul>
    </nav>
</aside>
```

如果对这部分也加上 CSS 样式，在浏览器中的显示效果如图 3-2 所示。

图 3-2　用 aside 元素实现的侧边栏示例

该示例为一个典型的博客网站中的侧边栏部分，因此被放在 aside 元素中，但是该侧边栏又具有导航的作用，因此被放置在 nav 元素中，该侧边栏的标题是“评论”，被放在 h2 元素中，在标题之后使用一个 ul 列表，用来存放具体的导航链接。

3.1.5　time 元素与微格式

首先来说一下微格式，它是一种利用 HTML 的 class 属性来对网页添加诸如新闻事件发生的日期和时间、个人电话号码、企业邮箱之类的附加信息的方法。

微格式并不是在 HTML 5 之后才有的，在 HTML 5 之前它就和 HTML 结合使用了，但是使用过程中在日期和时间的机器编码上出现了一些问题，编码过程中会产生一些歧义。HTML 5 增加了一种新的元素来无歧义地、明确地对机器编码日期和时间，并且以让人易读的方式来展现。这个元素就是 time 元素。

time 元素代表 24 小时中的某个时刻或某个日期，表示时刻时允许带时差。它可以定义很多格式的日期和时间，如下所示。

```
<time datetime="2010-11-13">2010 年 11 月 13 日 </time>
<time datetime="2010-11-13">11 月 13 日 </time>
<time datetime="2010-11-13"> 我的生日 </time>
<time datetime="2010-11-13T20:00"> 我生日的晚上 8 点 </time>
<time datetime="2010-11-13T20:00Z"> 我生日的晚上 8 点 </time>
<time datetime="2010-11-13T20:00+09:00"> 我生日的晚上 8 点的美国时间 </time>
```

在编码时机器读到的部分在 datetime 属性里，而元素的开始标记与结束标记中间的部分显示在网页上。datetime 属性中日期与时间之间要用“ T”文字分隔，“ T”表示时间。注意倒数第二行，时间加上 Z 字母表示对机器编码时使用 UTC 标准时间，倒数第一行则加上了时差，表示向机器编码另一地区时间，如果是编码本地时间，则不需要添加时差。

3.1.6　pubdate 属性

putdate 属性是一个可选的 boolean 值的属性，它可以被应用到 article 元素中的 time 元素上，意思是 time 元素代表了文章（artilce 元素的内容）或整个网页的发布日期，putdate 属性的具体使用方法如代码清单 3-10 所示。

代码清单 3-10　putdate 与 time 结合使用（一）

```
<article>
    <header>
        <h1> 苹果 </h1>
        <p> 发布日期
            <time datetime="2010-10-29" pubdate>2010 年 10 月 29 日 </time>
        </p>
    </header>
   <p> 苹果，植物类水果，多次花果 ...(" 苹果 " 文章正文 )</p>
  ...
</article>
```

你也许会疑惑为什么需要用到 pubdate 属性，为什么不能认为 time 元素就直接表示了文章或网页的发布日期呢？来看代码清单 3-11。

代码清单 3-11　putdate 与 time 结合使用（二）

```
<article>
    <header>
        <h1> 关于 <time datetime=2010-10-29>10 月 29 日 </time> 的舞会通知 </h1>
        <p> 发布日期：
            <time datetime=2010-10-11 pubdate>2010 年 10 月 11 日 </time>
        </p>
    </header>
    <p> 大家好：我是法律系 3 年级学生代表，.....（关于舞会的通知 )</p>
</article>
```

这个示例中有两个 time 元素，分别定义了两个日期——一个是舞会日期，另一个是通知发布日期。由于都使用了 time 元素，所以需要使用 pubdate 属性表明哪个 time 元素代表通知的发布日期。

3.2 新增的非主体结构元素

除了以上几个主要的结构元素之外，HTML 5 内还增加了一些表示逻辑结构或附加信息的非主体结构元素。

3.2.1 header 元素

header 元素是一种具有引导和导航作用的结构元素，通常用来放置整个页面或页面内的一个内容区块的标题，但也可以包含其他内容，例如数据表格、搜索表单或相关的 LOGO 图片。

很明显，整个页面的标题应该放在页面的开头，我们可以用如下形式书写页面的标题。

```
<header><h1> 页面标题 </h1></header>
```

需要强调的一点是，一个网页内并不限制只能有一个 header 元素，可以拥有多个，可以为每个内容区块加一个 header 元素，如代码清单 3-12 所示。

代码清单 3-12　多个 header 元素示例

```
<header>
    <h1> 网页标题 </h1>
</header>
<article>
    <header>
        <h1> 文章标题 </h1>
    </header>
    <p> 文章正文 </p>
</article>
```

在 HTML 5 中，一个 header 元素通常包括至少一个 heading 元素（h1 ~ h6），也可以包括我们后面将要讨论的 hgroup 元素，也可以包括其他元素（譬如 table 或 form），根据最新的 W3C HTML 5 标准，还可以包括 nav 元素。

最后，让我们看一下博客网页中 header 元素的一个应用示例，示例中 header 元素处于页面顶部，详见代码清单 3-13。

代码清单 3-13　博客网页中 header 元素示例

```
<header>
<hgroup>
<h1>IT 新技术 </h1>
```

```
<a href="http://blog.sina.com.cn/itnewtech">
      http://blog.sina.com.cn/itnewtech
</a>
<a href="#">[ 订阅 ]</a>
<a href="#">[ 手机订阅 ]</a>
</hgroup>
<nav>
<ul>
<li> 首页 </li>
<li><a href="http://blog.sina.com.cn/articlelist1.html"> 博文目录 </a></li>
<li><a href="http://photo.blog.sina.com.cn/itnewtechl"> 图片 </a></li>
<li><a href="http://photo.blog.sina.com.cn/itnewtech"> 关于我 </a></li>
</nav>
</header>
```

如果对这段代码使用 CSS 样式，显示界面如图 3-3 所示。

图 3-3　博客网页中 header 元素示例

3.2.2　footer 元素

footer 元素可以作为其上层父级内容区块或一个根区块的脚注。footer 通常包括其相关区块的脚注信息，如作者、相关阅读链接以及版权信息等。

在 HTML 5 出现之前，我们使用下面的方式编写页脚，如代码清单 3-14 所示。

代码清单 3-14　HTML 5 之前的页脚示例

```
<div id="footer">
    <ul>
        <li> 版权信息 </li>
        <li> 站点地图 </li>
        <li> 联系方式 </li>
    </ul>
<div>
```

但是 HTML 5 出现之后，这种方式将不再使用，而是使用更加语义化的 footer 元素来替代，如代码清单 3-15 所示。

代码清单 3-15　footer 元素示例

```
<footer>
    <ul>
```

```
            <li> 版权信息 </li>
            <li> 站点地图 </li>
            <li> 联系方式 </li>
        </ul>
    </footer>
```

与 header 元素一样，一个页面中也不限制只可以用一个 footer 元素。同时，可以为 article 元素或 section 元素添加 footer 元素，来看下面两个示例。

代码清单 3-16 为一个在 article 元素中添加 footer 元素的示例。

代码清单 3-16　在 article 元素中添加 footer 元素

```
<article>
    文章内容
    <footer>
        文章的脚注
    </footer>
</article>
```

代码清单 3-17 为一个在 section 元素中添加 footer 元素的示例。

代码清单 3-17　在 section 元素中添加 footer 元素

```
<section>
    分段内容
    <footer>
        分段内容的脚注
    </footer>
</section>
```


3.2.3　address 元素

address 元素用来在文档中呈现联系信息，包括文档作者或文档维护者的名字、文档作者或文档维护者的网站链接、电子邮箱、真实地址、电话号码等。address 应该不只是用来呈现电子邮箱或真实地址，还可以用来展示跟文档相关的联系人的所有联系信息。譬如，代码清单 3-18 中展示了一些博客中某篇文章评论者的名字及其在博客中的网址链接。

代码清单 3-18　address 元素示例

```
<address>
    <a href=http://blog.sina.com.cn/itnewtech> 陆凌牛 </a>
    <a href=http://blog.sina.com.cn/zhangyu> 张玉 </a>
    <a href=http://blog.sina.com.cn/baiquanli> 白权立 </a>
</address>
```

下面通过代码清单 3-19 来查看如何结合使用 footer 元素、time 元素与 address 元素。

代码清单 3-19　footer、time 与 address 结合使用

```
<footer>
    <div>
        <address>
            <a title=" 文章作者: 陆凌牛 "href="http://blog.sina.com.cn/itnewtech"> 陆凌牛 </a>
        </address>
        发表于 <time datetime="2010-10-04">2010 年 10 月 4 日 </time>
    </div>
</footer>
```

在这个示例中，把博客文章的作者、博客的主页链接作为作者信息放在 address 元素中，把文章发表日期放在 time 元素中，把这个 address 元素与 time 元素中的总体内容作为脚注信息放在 footer 元素中。

3.2.4　main 元素

main 元素表示网页中的主要内容。主内容区域指与网页标题或应用程序中本页面主要功能直接相关或进行扩展的内容。该区域应该为每一个网页中所特有的内容，不能包含整个网站的导航条、版权信息、网站 LOGO、公共搜索表单等整个网站内部的共同内容。

每个网页内部只能放置一个 main 元素。不能将 main 元素放置在任何 article、aside、footer、header 或 nav 元素内部。

注意，由于 main 元素不对页面内容进行分区或分块，所以不会对下文所要描述的网页大纲产生任何影响。

main 元素的一个使用示例如代码清单 3-20 所示。

代码清单 3-20　main 元素的使用示例

```
<!DOCTYPE html>
<html>
<head>
<title>2022 年夏季 毕业典礼 </title>
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="courses.html"> 课程 </a></li>
                <li><a href="fees.html"> 学费 </a></li>
                <li><a> 毕业 </a></li>
            </ul>
        </nav>
    </header>
    <main>
        <h1> 毕业 </h1>
        <nav>
```

```
            <ul>
                <li><a href="#ceremony"> 典礼 </a></li>
                <li><a href="#graduates"> 毕业生 </a></li>
                <li><a href="#awards"> 表彰 </a></li>
            </ul>
        </nav>

        <H2 id="ceremony"> 典礼 </H2>
        <p> 入场仪式 </p>
        <p> 毕业生代表发言 </p>
        <p> 学生会主席发言 </p>
        <p> 颁发毕业证书 </p>
        <p> 校长总结发言 </p>

        <h2 id="graduates"> 毕业生 </h2>
        <ul>
            <li>Eileen Williams</li>
            <li>Andy Maseyk</li>
            <li>Blanca Sainz Garcia</li>
            <li>Clara Faulkner</li>
            <li>Gez Lemon</li>
            <li>Eloisa Faulkner</li>
        </ul>

        <h2 id="awards"> 表彰 </h2>
        <ul>
            <li>Clara Faulkner</li>
            <li>Eloisa Faulkner</li>
            <li>Blanca Sainz Garcia</li>
        </ul>
    </main>
    <footer> Copyright 2012 常州市玉凌软件 </footer>

</body>
</html>
```

3.3 HTML 5 中网页结构

前面两节详细介绍了在 HTML 5 中具体新增了哪些结构元素以及这些元素的定义和使用方法。在 HTML 5 中，可以使用这些结构元素来编排一份网页大纲，通过该网页大纲，我们可以一目了然地知道网页中具有哪些内容，网页中以什么样的结构形式来组织这些内容。

3.3.1 HTML 5 中的大纲

在 Word 文档中，一份文档的大纲如下所示：

1．HTML 5 的基础知识
1．1　HTML 5 概述
（第 1 章中第 1 节的正文）
1．1．1　HTML 5 是什么？
（第 1 章中第 1 节第 1 小节的正文）
1．1．2　HTML 5 中的新增 API
（第 1 章中第 1 节第 2 小节的正文）

HTML 5 网页文档中的大纲也基本如此，只不过使用不同的结构元素进行表达而已。换句话说，在 HTML 5 中，使用各种结构元素所描述出来的整个网页的层次结构，就是该网页的大纲。因此，在组织这份大纲的时候，不能使用 div 元素，因为 div 元素只能当作容器，用在需要对网页中某个局部使用整体样式时。

许多工具可以对 HTML 5 的网页文档进行分析，然后将该文挡中的大纲以可视化的形式展现出来。http://gsnedders.html5.org/outliner/ 网站中就有一个文档大纲分析器，可以针对代码清单 3-21 中所示文档进行分析，分析结果如图 3-4 所示。

代码清单 3-21　大纲分析工具测试用代码

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
<title> 大纲分析 </title>
</head>
<section>
    <h1>HTML5 的基础知识 </h1>
    <section>
        <h2>HTML5 概述 </h2>
        ( 正文 )
        <section>
            <h3>HTML5 是什么？ </h3>
            ( 正文 )
        </section>
        <section>
            <h3>HTML5 中的新增 API</h3>
            ( 正文 )
        </section>
    </section>
    <section>
        <h2>HTML5 快速入门 </h2>
        ( 正文 )
        <section>
            <h3>HTML 与 XHTML</h3>
```

```
            （正文）
        </section>
    </section>
</section>
```

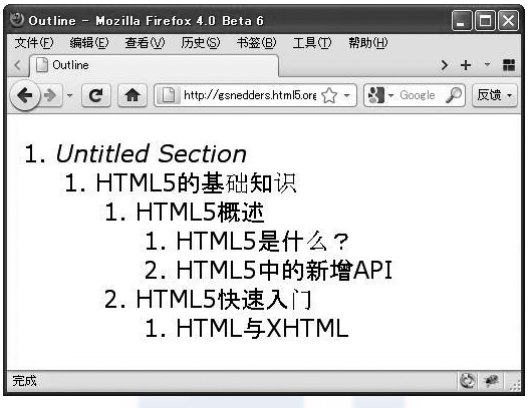


图 3-4　在线大纲分析器的分析结果

图 3-4 中出现“1.Untitled Section”，是由于该网页文档中第一个元素即为 section 元素，缺乏整个网页标题元素。加入标题元素（<h1> 元素），将代码清单 3-21 中的代码修改为代码清单 3-22 所示的代码，分析出来的大纲如图 3-5 所示。

代码清单 3-22　添加了 header 元素后的大纲分析工具测试用代码

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
<title> 大纲分析 </title>
</head>
 <h1>HTML5 的基础知识 </h1>
<section>
    <h2>HTML5 概述 </h2>
    （正文）
    <section>
        <h3>HTML5 是什么？ </h3>
        （正文）
    </section>
    <section>
        <h3>HTML5 中的新增 API</h3>
        （正文）
    </section>
</section>
<section>
    <h2>HTML5 快速入门 </h2>
```

```
    （正文）
    <section>
        <h3>HTML 与 XHTML</h3>
        （正文）
    </section>
</section>
```

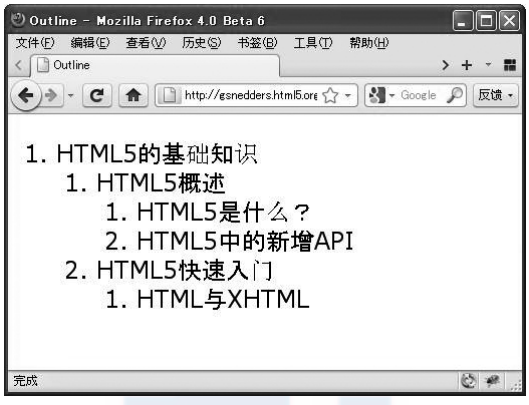


图 3-5　加入网页标题后的页面大纲

看到这里，你也许会问，如果光加一个 <h1> 元素，就可以分析出标题来，那么还需要 header 元素干什么？答案是 h1 元素一般用来显示文字标题。事实上，在要定义为网页标题的整个内容中，往往并不只是显示一段文字而已，其中还包括了大量的导航条、企业 LOGO 图片、广告 flash 等，这些内容都可以放在一个 header 元素中，作为整个网页标题的内容，而标题文字为 h1 元素中的文字，在大纲中显示该标题文字。但是，这里要说明的是，header 元素本身的作用不是被用来生成大纲的，大纲是依靠 header 元素中的 h1 ~ h6 元素来生成的，如使用代码清单 3-23 中的代码生成的大纲如图 3-6 所示。

代码清单 3-23　在企业网站中使用图片来显示企业名称

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
<title> 企业网站 </title>
</head>
 <header>
    <img src="title.jpg" alt=" 用来显示企业名称的图片 ">
</header>
<section>
    <h2> 企业描述 </h2>
    （正文）
</section>
```

图 3-6　header 元素本身并没有用来生成大纲

这里会产生这样一个问题，在很多企业网站（或其他网站）中，企业的标题并不是以文字来显示的，而是为了视觉效果，放在图片中显示的。难道这种情况就不能生成大纲了吗？笔者认为，这里有个小技巧，在 header 元素中，使用如下代码，既可以用图片来显示企业名称，又可以生成大纲。

```
<header>
    <h1><img src="title.jpg" alt="企业名称"></h1>
</header>
```

在 header 元素中使用这段代码后，生成的大纲如图 3-7 所示。

图 3-7　在 header 元素中使用图片来生成大纲

与 header 元素相同，footer 元素中如果没有标题元素（h1 ~ h6 元素）也不用于生成大纲。

在代码清单 3-21 或代码清单 3-23 中的代码底部追加如下代码，生成的大纲将不会有任何变化。

```
<footer>
```

```
    版权所有：陆凌牛
</footer>
```

另外，对 nav 元素与 aside 元素来说，如果不在元素内部加入标题元素（h1 ~ h6 元素），生成大纲时会在该元素所在位置处添加一个“Untitled Section”的说明文字，但是根据 HTML 5 的开发文档记载，nav 元素的作用为存放一组链接组，aside 元素的作用为表示当前页面或文章的附属信息部分，允许不在这两个元素中添加标题，也就是说，大纲中存在对这两个元素的内容为“Untitled Section”的说明文字是合理的。

在代码清单 3-21 的代码底部添加如下代码，生成的大纲如图 3-8 所示。

```
<nav>
    <ul>
        <li><a href="#">链接测试 1</a></li>
        <li><a href="#">链接测试 2</a></li>
    </ul>
</nav>
<aside>
    侧边栏中的内容
</aside>
```

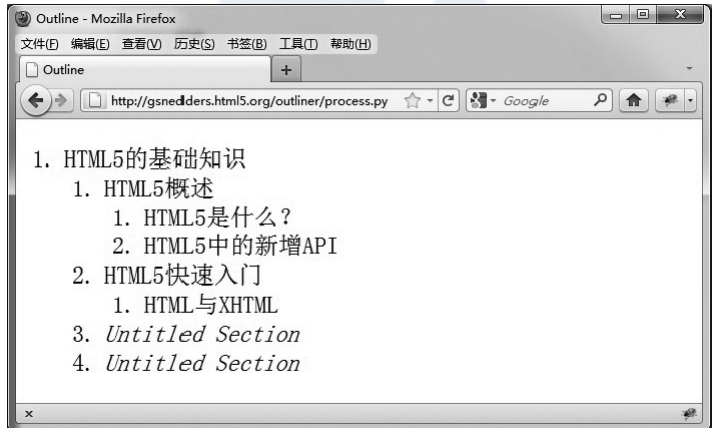


图 3-8　在文档的底部添加 nav 元素与 aside 元素

另外，在 HTML 5 中，body 元素、blockquote 元素、fieldset 元素、td 元素、details 元素及 figure 元素称为节根（sectioning root）元素。这些元素的共同特征是拥有自己独立的大纲，并且这些元素内的 section 元素、article 元素、标题元素（h1 ~ h6 元素）、nav 元素以及 aside 元素等，只用于生成其父元素的大纲时，而不用于生成父元素的上层祖先元素的大纲时。

在代码清单 3-24 中，blockquote 元素内部有一个 h1 元素，正是因为这个 h1 元素是位于节根元素 blockquote 元素内部的，所以在针对 blockquote 元素的父元素 body 元素生成页面大纲时，该 h1 元素并没有显示在大纲中，如图 3-9 所示。

代码清单 3-24　针对 body 元素生成大纲时节根元素中的子元素不起作用

```
<!DOCTYPE html>
<meta charset="UTF-8">
<body>
<h1> 网页标题 </h1>
<blockquote>
<h1> 节根元素内部标题 </h1>
</blockquote>
</body>
```

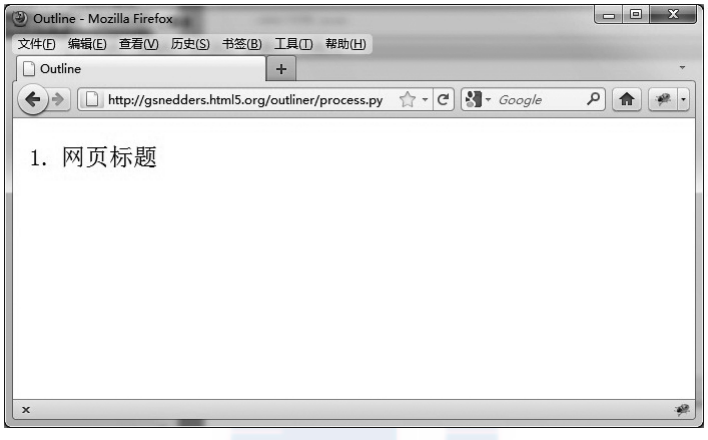


图 3-9　针对 body 元素生成大纲时节根元素中的子元素不起作用

3.3.2　大纲的编排规则

关于内容区块的编排，可以分为显式编排与隐式编排两种方式。

1. 显式编排内容区块

显式编排是指明确使用 section 等元素创建文档结构，在每个内容区块内使用标题（h1 ~ h6、hgroup 等），如代码清单 3-25 所示。

代码清单 3-25　显式编排内容区块示例

```
<body>
    <h1> 网页级内容区块标题 </h1>
    <p> 网页级内容区块的正文 </p>
    <section>
        <h2>section 级内容区块的标题 </h2>
        <p>section 级内容区块的正文 </p>
    </section>
</body>
```

2. 隐式编排内容区块

所谓隐式编排，是指不明确使用 section 等元素，而是根据页面中所书写的各级标题

（h1 ~ h6、hgroup）等自动创建各级内容区块。因为 HTML 5 分析器只要看到书写了某个级别的标题，就会判断存在相对应的内容区块。代码清单 3-26 为一个隐式编排内容区块的示例。

代码清单 3-26　隐式编排内容区块示例

```
<body>
    <h1> 网页级内容区块标题 </h1>
    <p> 网页级内容区块的正文 </p>
    <!-- 分析器根据 h2 等元素判断生成内容区块 -->
    <h2>section 级内容区块的标题 </h2>
    <p>section 级内容区块的正文 </p>
</body>
```

将这两种编排方式进行对比，很明显，显式编排更加清晰明确、易读。

3. 标题分级

不同标题之间是有级别的，h1 的级别最高，h6 的级别最低。隐式编排时按如下规则自动生成内容区块。

- ❑ 如果新出现的标题比上一个标题级别低，生成下级内容区块。
- ❑ 如果新出现的标题比上一个标题级别高或级别相等，生成新的内容区块。

第一条规则的示例与前面一样，现在我们来看关于第二条规则的示例，如代码清单 3-27 所示。

代码清单 3-27　第二条规则示例

```
<body>
<section>
    <h2>section 级别的内容区块的标题 </h2>
    <p>section 级别的内容区块的正文 </p>
    <!—因为下面的标题级别比上一个标题级别高，所以自动创建新的内容区块 -->
    <h1> 新的 section 级别的内容区块的标题 </h1>
    <p> 新的 section 级别的内容区块的正文 </p>
</section>
</body>
```

如果把上一个示例改成显式编排，代码如代码清单 3-28 所示。

代码清单 3-28　第二条规则的显式编排示例

```
<body>
<section>
    <h2>section 级别的内容区块的标题 </h2>
    <p>section 级别的内容区块的正文 </p>
</section>
<section>
    <h1> 新的 section 级别的内容区块的标题 </h1>
    <p> 新的 section 级别的内容区块的正文 </p>
```

```
</section>
</body>
```

因为隐式编排容易让自动生成的整个文档结构与所想要的文档结构不一样，而且也容易引起文档结构混乱，所以尽量使用显式编排。

4. 不同的内容区块可以使用相同级别的标题

另外，不同的内容区块可以使用相同级别的标题。例如，父内容区块与子内容区块可以使用相同级别的标题 h1。这样做的好处是，每个级别的标题都可以单独设计，如果既需要“整个网页的标题”，又需要“文章的标题”（譬如书写文档时），这样做将会带来很大的便利性，如同代码清单 3-29 所示。

代码清单 3-29　不同的内容区块可以使用相同级别的标题

```
<body>
<h1> 网页的标题 </h1>
<article>
    <header>
        <hgroup>
            <h1> 文章标题 </h1>
            <h2> 文章子标题 </h2>
        </hgroup>
        <p> 文章正文 </p>
    </header>
</article>
</body>
```

5. 网页编排示例

基于以上讲解过的知识点，我们来看应该怎样编排网页的内容。代码清单 3-30 为一个标准博客网页的示例，这个示例具备一个标准博客网页的基本要素，只缺少为了使用样式而补充添加的 div 元素。

代码清单 3-30　网页编排示例

```
<!DOCTYPE html>
<head>
  <title> 网页编排示例 </title>
  <meta charset="UTF-8">
</head>
<body>
<!-- 网页标题 -->
<header>
    <h1> 网页标题 </h1>
    <!-- 网站导航链接 -->
    <nav>
        <ul>
            <li><a href="index.html"> 首页 </a></li>
```

```
            <li><a href="help.html"> 帮助 </a></li>
        </ul>
    </nav>
</header>
<!-- 文章正文 -->
<article>
    <hgroup>
        <h1> 文章主标题 </h1>
        <h2> 文章子标题 </h2>
    </hgroup>
    <p> 文章正文 </p>
    <!-- 文章评论 -->
    <section class="comments">
        <article>
            <h1> 评论标题 </h1>
            <p> 评论正文 </p>
        </article>
    </section>
</article>
<!-- 版权信息 -->
<footer>
    <small> 版权所有：陆凌牛 </small>
</footer>
</body>
```

这个示例使用了嵌套 artilce 元素的方式，将关于评论的 article 元素嵌套在主 article 元素中，在 HTML 5 中，推荐使用这种嵌套 article 元素的方式。

3.3.3　对新的结构元素使用样式

因为很多浏览器尚未对 HTML 5 中新增的结构元素提供支持，我们无法知道客户端使用的浏览器是否支持这些元素，所以需要使用 CSS 追加如下声明，目的是通知浏览器页面中使用的 HTML 5 中新增元素都是以块方式显示的，如下所示。

```
//追加 block 声明
article, aside, dialog, figure, footer, header, legend, nav, section, main
{   display: block; }
//正常使用样式
nav{float;left;width:20%;}
article{float:right;width:79%;}
```

另外，Internet Explorer 8 及之前的浏览器不支持用 CSS 的方法来使用这些尚未支持的结构元素，为了在 Internet Explorer 浏览器中也能正常使用这些结构元素，需要使用 JavaScript 脚本，如下所示。

```
//在脚本中创建元素
<script>
document.createElement("header");
```

```
document.createElement("nav");
document.createElement("article");
document.createElement("footer");
document.createElement("main");
</script>
<style>
// 正常使用样式
nav{float;left;width:20%;}
article{float:right;width:79%;}
</style>
```

尽管这段 JavaScript 脚本在其他浏览器中是不需要的，但它不会对这些浏览器造成什么不良影响。另外，到了 Internet Explorer 9 之后，这段脚本就不需要了。