

TURING

图灵程序设计丛书

Apress®

HTML5 Solutions

Essential Techniques for HTML5 Developers

HTML5 实战

[意] Marco Casario [比] Peter Elst

[美] Charles Brown [印度] Nathalie Wormser 著

[比] Cyril Hanquez

乔晶峰 孙爱红 译

- 实战、实例、实用
- 多位业内专家联袂奉献
- 参透80个问题，掌握HTML5



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

HTML5 Solutions

Essential Techniques for HTML5 Developers

HTML5 实战

[意] Marco Casario [比] Peter Elst
[美] Charles Brown [印度] Nathalie Wormser 著
[比] Cyril Hanquez
乔晶峰 孙爱红 译

人民邮电出版社

北 京

图书在版编目 (C I P) 数据

HTML5实战 / (意) 卡萨里奥等著 ; 乔晶峰, 孙爱红译. — 北京 : 人民邮电出版社, 2013. 1

(图灵程序设计丛书)

书名原文: HTML5 Solutions: Essential Techniques for HTML5 Developers

ISBN 978-7-115-29752-5

I. ①H… II. ①卡… ②乔… ③孙… III. ①超文本标记语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第253091号

内 容 提 要

本书秉承 Solutions 系列的一贯风格, 将 HTML5 新增的标签和 API 组织成 80 余个实战案例, 是一本集实用性和参考性于一体的实战手册。本书实战案例涉及 HTML5 结构化标记、语义元素、新增表单类型、音频及视频、Canvas 绘图、通信 API、WebSocket、Geolocation、本地存储及 HTML5 无障碍访问等, 几乎涵盖了 HTML5 的方方面面。

本书不仅适合刚刚入门的新手学习, 也适合作为开发人员的案头工具。

图灵程序设计丛书

HTML5实战

-
- ◆ 著 [意] Marco Casario [比] Peter Elst
[法] Charles Brown [印度] Nathalie Wormser
[比] Cyril Hanquez
- 译 乔晶峰 孙爱红
- 责任编辑 李松峰
- 执行编辑 王一枝
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 17.75
字数: 420千字 2013年1月第1版
印数: 1~3 500册 2013年1月河北第1次印刷
- 著作权合同登记号 图字: 01-2011-7470号
ISBN 978-7-115-29752-5
-

定价: 59.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

目 录

第 1 章 HTML5 页面结构	1	第 4 章 HTML5 表单	49
实战 1-1 创建 HTML5 的 DOCTYPE	1	认识新的输入类型	50
实战 1-2 在 HTML5 中声明字符编码	2	实战 4-1 使用 email 输入类型	50
实战 1-3 将文档分成不同区域	3	实战 4-2 使用 URL 输入类型	53
实战 1-4 使文档各部分可独立分发	4	实战 4-3 使用数字微调控件	55
实战 1-5 创建旁注	4	实战 4-4 用 range 输入类型添加滑动框 控件	58
实战 1-6 创建文头	5	实战 4-5 发送多个文件	61
实战 1-7 组合<h1>至<h6>元素	7	实战 4-6 使用 datalist 组件实现输入的 自动完成	63
实战 1-8 创建文脚	8	实战 4-7 验证表单控件	65
实战 1-9 在 HTML5 文档中创建导航	9	实战 4-8 利用正则表达式创建自定义 输入类型	69
实战 1-10 插入图表	11	实战 4-9 设置输入框的占位符文本	71
实战 1-11 浏览器兼容性	12	实战 4-10 创建日期和时间控件	72
本章小结	15	本章小结	75
第 2 章 HTML5 标记	16	第 5 章 HTML5 媒体元素: Audio 和 Video	76
实战 2-1 在 HTML5 中使用<hr>标签	16	实战 5-1 在 Web 页面中嵌入视频	78
实战 2-2 使用<iframe>标签	18	实战 5-2 检测浏览器是否支持<video>	80
实战 2-3 在页面中嵌入媒体	19	实战 5-3 定制视频控件	83
实战 2-4 使用<area>标签	21	实战 5-4 预加载视频	92
本章小结	23	实战 5-5 自定义视频进度条	93
第 3 章 HTML5 结构化语义元素	24	实战 5-6 使用多源<video>元素	99
认识微数据	24	实战 5-7 全屏播放视频	101
实战 3-1 使用 itemprop 和 itemscope 属性	26	实战 5-8 为视频添加遮罩	104
实战 3-2 创建自定义词汇表	31	实战 5-9 使用<audio>元素	107
实战 3-3 理解链接类型和关系	35	本章小结	108
实战 3-4 header 和 hgroup 元素	37		
实战 3-5 添加图片说明	41		
实战 3-6 添加旁注	43		
本章小结	48		

第 6 章 HTML5 绘图 API.....	110	第 9 章 HTML5 WebSocket.....	199
实战 6-1 如何使用<canvas>元素的绘图 API.....	110	实战 9-1 检查浏览器是否支持 WebSocket.....	199
实战 6-2 使用路径和坐标.....	113	实战 9-2 建立 WebSocket 连接.....	201
实战 6-3 绘制图形：矩形和圆形.....	121	实战 9-3 处理 WebSocket 事件.....	204
实战 6-4 用纯色填充图形.....	124	实战 9-4 通过 WebSocket API 使用 WebSocket Server.....	207
实战 6-5 使用渐变色填充.....	128	本章小结.....	216
实战 6-6 在<canvas>中写字.....	132	第 10 章 HTML5 地理定位 API.....	217
实战 6-7 相对文字大小.....	135	认识地理定位 API.....	217
实战 6-8 输出 PNG 图片文件.....	138	实战 10-1 使用 navigator 对象.....	218
本章小结.....	142	实战 10-2 获取当前位置.....	219
第 7 章 HTML5 Canvas.....	143	实战 10-3 使用 position 对象.....	222
实战 7-1 了解画布 API.....	143	实战 10-4 处理定位错误.....	225
实战 7-2 检测是否支持<canvas>和绘制文本.....	150	实战 10-5 追踪用户的位置.....	227
实战 7-3 认识屏幕坐标系统和<canvas>变换.....	156	实战 10-6 使用 geo.js 开源库.....	230
实战 7-4 操作像素.....	160	本章小结.....	232
实战 7-5 应用阴影和模糊.....	166	第 11 章 HTML5 本地存储.....	233
实战 7-6 <canvas>的动画效果.....	170	实战 11-1 认识偶尔连接的应用.....	233
本章小结.....	176	实战 11-2 检查 HTML5 存储支持情况.....	236
第 8 章 HTML5 通信 API.....	177	实战 11-3 为页面声明清单文件.....	241
认识 postMessage API.....	177	实战 11-4 使用 ApplicationCache 对象.....	244
增强 postMessage 通信安全.....	178	实战 11-5 ApplicationCache 事件.....	246
实战 8-1 检查浏览器是否支持 postMessage API.....	179	实战 11-6 删除本地缓存.....	248
跨文档消息和 CORS.....	181	本章小结.....	252
实战 8-2 在 window 和<iframe>间传递消息.....	181	第 12 章 HTML5 无障碍访问.....	253
实战 8-3 用服务端事件技术编写实时 Web 应用.....	186	无障碍访问的四个原则.....	253
实战 8-4 用消息通道在不同的浏览上下文中运行代码.....	190	WCAG 的目的.....	254
实战 8-5 用 XMLHttpRequest Level 2 上传文件.....	192	实战 12-1 用<nav>元素创建锚点链接.....	255
实战 8-6 检查 XMLHttpRequest Level 2 跨源浏览器支持.....	195	实战 12-2 创建无障碍的表格数据.....	258
本章小结.....	198	实战 12-3 创建无障碍的表单.....	262
		实战 12-4 使用<video>元素并添加字幕和注解.....	265
		实战 12-5 使用 ARIA 项目.....	267
		本章小结.....	274

2004 年，来自 Apple、Opera 和 Mozilla 的一群开发者非常不看好当时 HTML 和 XHTML 的前景，他们为此成立了 Web 超文本应用技术工作小组（WHATWG），并于 2005 年发布了名为“Web Applications 1.0”的提案。2006 年，万维网联盟（W3C）决定停止开发 XHTML，转而正式支持 WHATWG。2007 年，W3C 重新发布了 HTML 规范，并命名为 HTML5。

最初，人们估计最终版本的规范要到 2022 年才可能发布，但现在来看，需要重新考虑这个发布时间了。2009~2010 年，人们对 HTML5 的兴趣骤然爆发，涌现出大量支持 HTML5 的浏览器和设备。

第 1 章将介绍 HTML5 规范中新增加的结构，也会细数支持这些新结构的设备。

实战 1-1 创建 HTML5 的 DOCTYPE

由于 HTML 版本繁多，所以有必要告诉浏览器当前所使用的 HTML 版本号。浏览器是通过 DOCTYPE 来识别 HTML 版本号并正确显示要输出的内容的。

在本方案中，你将学到如何声明 HTML5 的 DOCTYPE。

1. 相关要点

传统 HTML 和 XHTML 文档中的 DOCTYPE 标签如下所示：

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

另外，还有一些变体形式。

而在 HTML5 中，该标签则简化成如下形式：

```
<!DOCTYPE html>
```

2. 实现方法

(1) 打开 HTML 编辑器或者其他文本编辑器。我们在本章使用 Dreamweaver CS5 来编辑示例代码。千万别使用文字处理软件，这类软件可能会插入一些 HTML 无法识别的多余字符。

(2) 有必要的话，新建一个 HTML 文档，起个名字然后保存到某个路径下。

假如你使用的 HTML 编辑器也是 Dreamweaver 之类的，那么代码可能如下所示：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Untitled Document</title>
  </head>
  <body>
  </body>
</html>
```

当然，即使实际代码和这个略有不同也无大碍，不必担心。

(3) 修改 DOCTYPE 标签为：

```
<!DOCTYPE html>
```

3. 专家提示

DOCTYPE 标签前面不要留有空白字符，因为它可能导致 HTML5 代码在浏览器中无法被正确地解析出来。

实战 1-2 在 HTML5 中声明字符编码

不同的语言要使用不同的字符集，<meta>标签声明使用何种字符集。大多数语言最常使用的字符集是 UTF-8。

在本方案中，你将学习如何在 HTML5 中正确地指定字符集。

1. 相关要点

HTML 文档的开头大多有如下一段代码：

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

在 HTML5 中，这段代码被简化为：

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

2. 实现方法

在实战 1-1 的<!DOCTYPE html>标签下面，键入以下代码即可：

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

3. 专家提示

多数场合都可使用 UTF-8，但很多开发者发现字符集 ISO-8859-1 使用起来更加灵活一些，而字符集 UTF-16 有时会产生错误的字符，甚至还会导致应用程序无法正常运行。

实战 1-3 将文档分成不同区域

1

在以前的 HTML 中,能把文档分成不同区域(section)的唯一方法是使用标签<div>。HTML5 则提供了一些新的选择。

在本方案中,你将学到如何使用 HTML5 的新标签创建不同的文档区域。在接下来的各方案中,我们还会讨论其他的结构化分区元素。

1. 相关要点

HTML 的<div>标签能将文档分成不同区域,但是通过<div>这个单词,无法清晰地理解文档各区域的含义。HTML5 提供了不少新的结构化元素,能把文档分成意义明确的区域。

首先介绍的是<section></section>标签。该元素表示文档任意的逻辑分区,例如可以表示产品描述、文档的章、讨论区等,其功能和<div>类似,但会以更具描述性和更与内容相关的方式来划分文档。

在 HTML5 中使用<section>就和使用<div>标签一样,可以使用 id 属性或者 class 属性等。这两种属性适用于所有 HTML5 元素,所以叫做全局属性。和 HTML 一样, id 的值必须是唯一的,而 class 则可以被多次使用,用于引用预定义好的脚本或者样式。

所有 HTML5 元素都有三种类型的属性,分别是:通用于所有元素的全局属性、只适用于某个元素的专用属性,以及事件处理器的内容属性(根据文档的内容而触发的属性)。我们将在本书中逐一讨论这些属性。

2. 实现方法

我们来创建一个关于制作奶油蛋糕的文档。下面的代码演示了<section></section>元素的典型用法。

```
<section id="mixing">
  <h2>The proper way to mix ingredients</h2>
  <p>When using a stand-mixer, it is important that you do not over-mix the
ingredients...</p>
</section>
<section id="baking">
  <h2>Proper baking techniques</h2>
  <p> It is important that you bake your cheesecake using a lot of moisture in the
oven...</p>
</section>
```

3. 专家提示

<section></section>元素以及接下来将要提到的其他结构化元素并不是为了要替代 HTML 的<div>标签。如果要把文档分成不同的逻辑区域,可以使用<section></section>元素或者其他任何一种结构化元素。而如果仅仅是为了格式化而划分文档的话,<div>标签则是更合适的选择。

实战 1-4 使文档各部分可独立分发

使页面中的部分或全部内容可独立分发已经变得越来越重要。例如，论坛中的一个讨论、一篇博客、一条读者评论等都将可能是分发或者聚合的对象。

在本方案中，我们将讨论 HTML5 的新元素<article></article>，有了它，实现这些功能比传统的 HTML 要容易得多。

1. 相关要点

当然，这个结构化标签的目的并不仅仅是要成为另一个划分文档的方法。文档中通常存在这样一些区域，我们希望其内容相对独立于其他内容，并且可以对外部分发，此标签就是用于这一目的。

<article></article>元素是独立的，它可以有自己的<section>和更细分的内容。

任何内容，只要用<article></article>元素包围起来，就能独立分发了。

2. 实现方法

像下面这样修改一下实战 1-3 中的例子，奶油蛋糕的制作方法就是可独立分发的了。

```
<article>
  <section id="mixing">
    <h2>The proper way to mix ingredients</h2>
    <p>When using a stand-mixer, it is important that you do not over mix the
ingredients...</p>
  </section>
  <section id="baking">
    <h2>Proper baking techniques</h2>
    <p>It is important that you bake your cheesecake using a lot of moisture in the
oven...</p>
  </section>
</article>
```

3. 专家提示

可以视<article></article>元素为一个独立的文档，而不是某个大文档的一部分。这样，在分发的时候，它就是完全可读且可理解的独立内容了。

实战 1-5 创建旁注

在传统的 HTML 中，当需要在侧边栏创建一个旁注的时候，我们会使用<div>标签，并用样式表（CSS）确定其显示位置。HTML5 中的结构化元素<aside></aside>简化了这一过程。和<section>元素一样，用它来分区文档具有更好的可描述性。

你将在本方案中学到如何使用<aside></aside>元素。

1. 相关要点

你肯定经常创建位于侧边栏的类似旁注的东西,图 1-1 说明了如何使用`<aside></aside>`标签实现这一功能。

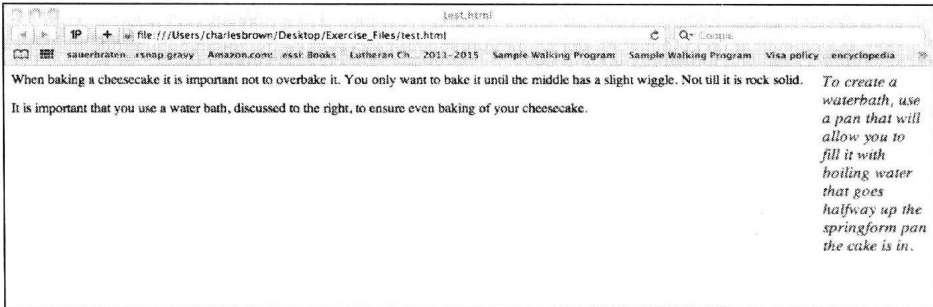


图 1-1 使用`<aside></aside>`元素

当然,也可以使用`<div>`元素来实现。不过,使用`<aside></aside>`元素,结构化描述看起来会更有意义。

2. 实现方法

前一小节演示的例子,其完整的实现代码如下:

```
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
  To create a water bath, use a pan that will allow you to fill it with boiling water that
  goes halfway up the springform pan in which the cake is placed.
</aside>
<p>
  When baking a cheesecake, it is important not to over bake it. You only want to bake it
  until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>
  It is important that you use a water bath, discussed at the right, to ensure even baking
  of your cheesecake.
</p>
```

3. 专家提示

`<aside></aside>`元素所处的位置很关键。在上例中,作为主文档的一部分,它位于主文档内。如果希望侧边栏是专用于某一特定区域的话,则必须把它放在该区域的内部。在使用`<article></article>`元素时这一点尤为重要,要让侧边栏与其相关内容共同发布。

实战 1-6 创建文头

使用结构化元素`<header></header>`来创建文档或区域的文头。该元素内还可以包含`<h1>`至`<h6>`元素;当然,在本章稍后你将会看到,这几个元素最好是包含在`<hgroup></hgroup>`元素内。

文头中也可放置品牌标志和区域内文章的目录等。

你在本方案中会看到使用<header></header>元素的例子。

1. 相关要点

为文档和区域提供介绍的简单有效方法就是使用<header></header>元素。我们在例子中添加<header></header>元素后的效果如图 1-2 所示。

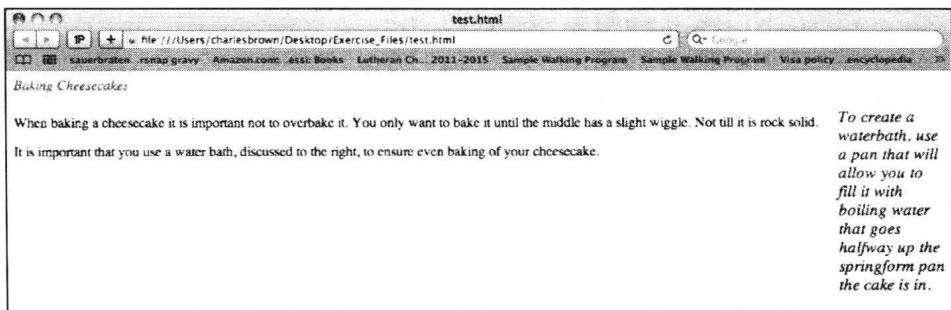


图 1-2 使用<header></header>元素

注意，本例中用<hr>元素添加了一条水平线，但这不是必须的。

2. 实现方法

下面的代码可以创建图 1-2 中所示的文档。

```
<header>
  <span style="color:red;font-style:italic;">
    Baking Cheesecakes</span>
  <hr>
</header>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
  To create a water bath, use a pan that will allow you to fill it with boiling water that
  goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
  When baking a cheesecake, it is important not to over bake it. You only want to bake it
  until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>
  It is important that you use a water bath, discussed at the right, to ensure even baking
  of your cheesecake.
</p>
```

对文档中多数其他的结构化元素，都可以用<header></header>元素包含简介。

3. 专家提示

不可以在<footer>、<address>以及其他<header>元素内使用<header></header>元素，否则将会导致无法正确地显示文档。

实战 1-7 组合<h1>至<h6>元素

有时, 需要把<h1>至<h6>组合在一起。例如, 用<h1>元素创建某一区域的标题, 然后在其下方用<h2>元素创建一个副标题。HTML5 新增的结构化元素<hgroup>可以把<h1>和<h2>元素组合在一起。

你将在本方案中看到使用<hgroup></hgroup>元素的例子。

1. 相关要点

有了结构化元素<hgroup></hgroup>后, 在有可选标题和副标题的情况下, 就可以把它们(<h1>到<h6>)组合在一起了。

在实战 1-6 的例子中添加<hgroup>后的效果如图 1-3 所示。

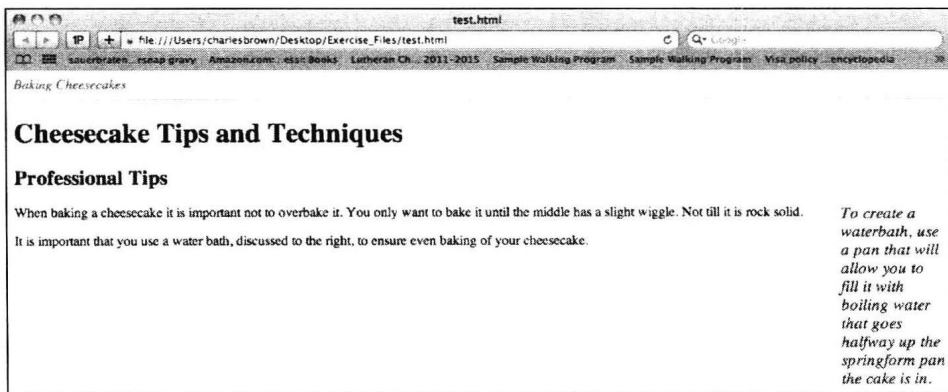


图 1-3 使用<hgroup></hgroup>元素

本例中, <h1>和<h2>标题被组合在<hgroup>内。

2. 实现方法

图 1-3 的例子所对应的代码如下:

```
<header>
  <span style="color:red;font-style:italic;">
    Baking Cheesecakes
  </span>
</header>
<hr>
<hgroup draggable="true">
  <h1>Cheesecake Tips and Techniques</h1>
  <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
  To create a water bath, use a pan that will allow you to fill it with boiling water that
  goes halfway up the spring form pan in which the cake is placed.
</aside>
```

```
<p>
    When baking a cheesecake, it is important not to over bake it. You only want to bake it
    until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
```

注意 我们把<hgroup>元素放在了<header></header>元素的后面，这并不是必须的，但在实践中建议这么做。HTML5 文档的任何区域中都可以使用<hgroup>元素。

3. 专家提示

对于构建良好的 HTML5 文档，把各种标题和副标题组合起来的最佳方法之一就是使用<hgroup></hgroup>元素。这在使用了<article></article>元素的时候尤其如此，因为它能保证标题及其副标题可以作为一个整体。

实战 1-8 创建文脚

顾名思义，<footer></footer>元素创建 HTML5 文档的收尾信息，这也是文档的一个结构化部分。文脚可以包含版权信息、作者信息、引用、隐私策略等。

在本方案中，你将学到<footer></footer>元素的使用。

1. 相关要点

可以使用<footer></footer>元素为 HTML5 文档或文档内任意区域创建文脚。在实战 1-7 的代码中添加<footer>元素后，结果如图 1-4 所示。

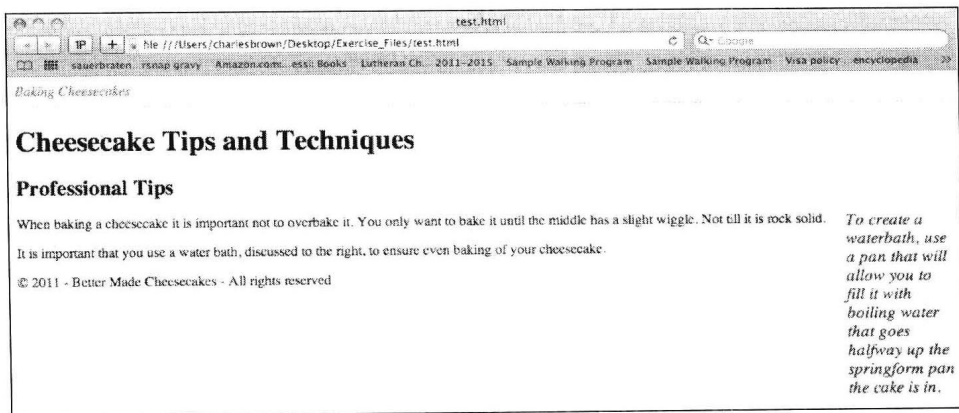


图 1-4 使用<footer></footer>元素

版权符号“©”以及其他权利相关的信息都显示在文脚中。

2. 实现方法

图 1-4 对应的代码为：

```
<header>
  <span style="color:red;font-style:italic;">
    Baking Cheesecakes
  </span>
  <hr>
</header>
<hgroup draggable="true">
  <h1>Cheesecake Tips and Techniques</h1>
  <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
  To create a water bath, use a pan that will allow you to fill it with boiling water that
  goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
  When baking a cheesecake, it is important not to over bake it. You only want to bake it
  until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
<footer> &copy; 2011 - Better Made Cheesecakes - All rights reserved </footer>
```

<footer></footer>元素可用于整个 HTML5 文档，也可以像本例这样用于文档内任意一个结构化区域。

3. 专家提示

<footer>元素不能放在<header>元素或者其他<footer>元素内，否则可能导致文档无法正确显示。

实战 1-9 在 HTML5 文档中创建导航

多数网站都有导航链接。无论是超链接还是按钮形式的，它们通常都被<div>元素与文档的其他内容区分开来。同样，<div>元素只表示它是文档的一部分，并不能表示某个区域是专用于导航的。而在 HTML5 中，可以使用标签来表明某个区域就是用于导航的。

在本节，你将学习到 HTML5 新引入的结构化元素<nav></nav>。

1. 相关要点

结构化元素<nav></nav>创建一个容器，用于放置整个 HTML5 文档或者文档内任意区域的导航。

结果如图 1-5 所示。

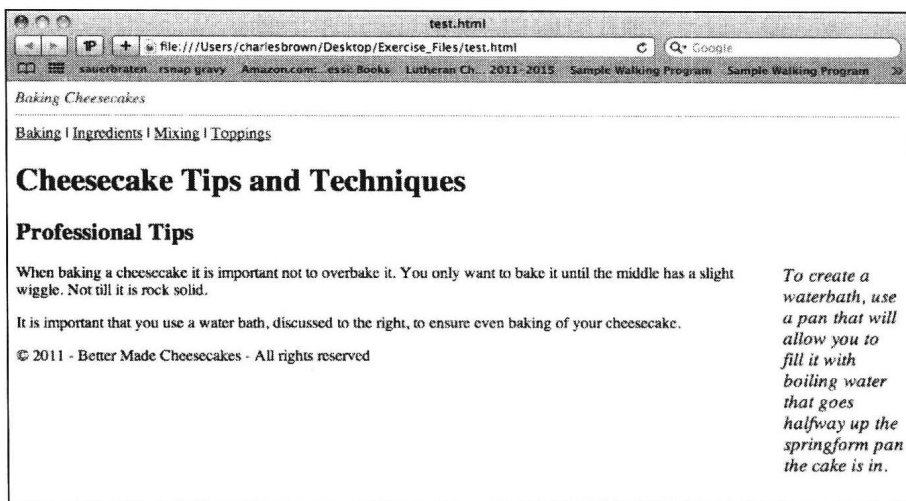


图 1-5 使用<nav></nav>元素

有了专门用于导航区域的标签，辨别哪些是导航也就容易多了。

2. 实现方法

图 1-5 对应的代码如下。其中的导航链接仅为演示所用，不具有实际功能。

```
<header>
  <span style="color:red;font-style:italic;">
    Baking Cheesecakes
  </span>
  <hr>
</header>
<nav>
  <a href="/Baking/" target="_blank">Baking</a> |
  <a href="/ingredients/" target="_blank">Ingredients</a> |
  <a href="/mixing/" target="_blank">Mixing</a> |
  <a href="/toppings/" target="_blank">Toppings</a>
</nav>
<hgroup draggable="true">
  <h1>Cheesecake Tips and Techniques</h1>
  <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
  To create a water bath, use a pan that will allow you to fill it with boiling water that
  goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
  When baking a cheesecake, it is important not to over bake it. You only want to bake it
  until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
your cheesecake.</p>
<footer> &copy; 2011 - Better Made Cheesecakes - All rights reserved </footer>
```

可以看出，文档的导航区域也有了易于辨别的专属标签。如本例所示，该标签既可以用于

HTML5 文档，也可以用于文档的任意区域。

3. 专家提示

除了图 1-5 中所示的超链接外，还可以在<nav></nav>元素中放置任意用于辅助导航的内容。

另外，建议不要在<footer>元素内使用<nav></nav>元素。虽没有严格禁止，但这样做可能导致页面显示不正确。

实战 1-10 插入图表

我们经常需要在网页中插入图片、图解、图表等。在此之前，开发人员还只能使用元素。现在，在 HTML5 中可以使用新的<figure></figure>元素来插入图表。

在本方案中，你将看到使用<figure></figure>元素的例子。

1. 相关要点

结构化元素<figure></figure>用于在 HTML5 文档或文档的任意部分插入图解类的元素。如图 1-6 所示。

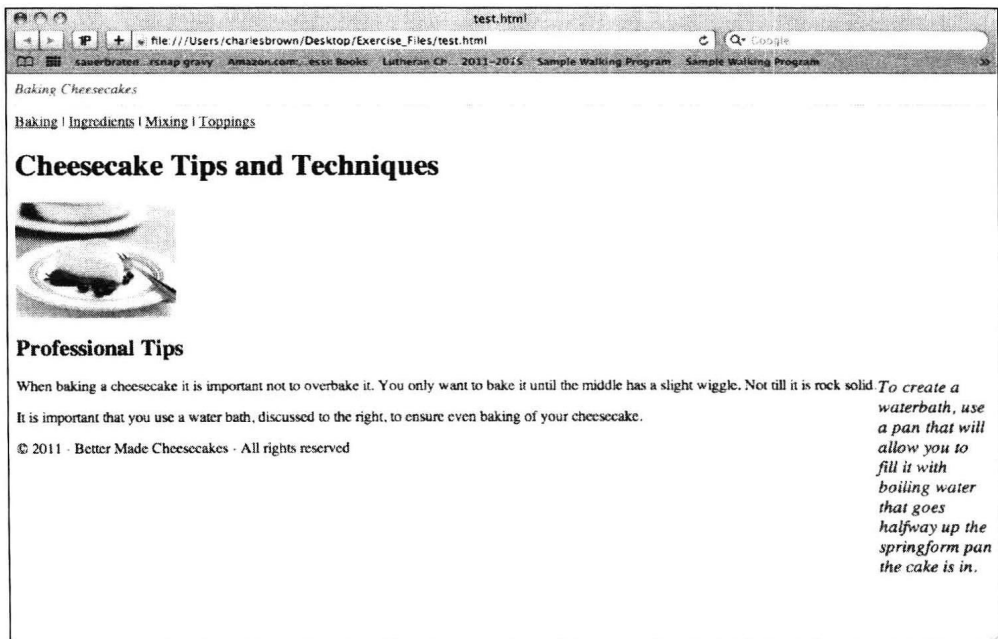


图 1-6 使用<figure></figure>元素

现在，文档的图解元素有了易于辨别的专属标签，它可用于文档的任何区域。

2. 实现方法

图 1-6 所示结果可通过如下代码实现。同样，其中的导航链接仅为了演示，不具有实际功能。

```
<header>
  <span style="color:red;font-style:italic;">
    Baking Cheesecakes
  </span>
  <hr>
</header>
<nav>
  <a href="/Baking/" target="_blank">Baking</a> |
  <a href="/ingredients/" target="_blank">Ingredients</a> |
  <a href="/mixing/" target="_blank">Mixing</a> |
  <a href="/toppings/" target="_blank">Toppings</a>
</nav>
<hgroup draggable="true">
  <h1>Cheesecake Tips and Techniques</h1>
  <figure>
    
  </figure>
  <h2>Professional Tips</h2>
</hgroup>
<aside style="font-size:larger;font-style:italic;color:blue;float:right;width:120px;">
  To create a water bath, use a pan that will allow you to fill it with boiling water that
  goes halfway up the spring form pan in which the cake is placed.
</aside>
<p>
  When baking a cheesecake, it is important not to over bake it. You only want to bake it
  until the middle has a slight wiggle, not until it is rock solid.
</p>
<p>It is important that you use a water bath, discussed at the right, to ensure even baking of
  your cheesecake.</p>
<footer> &copy; 2011 - Better Made Cheesecakes - All rights reserved </footer>
```

由于使用了专用标签<figure></figure>，很容易就能判断出图片放在什么地方了。

3. 专家提示

除了<figure>元素，HTML5 还增加了另一个与之密切相关的新元素<figcaption></figcaption>（图题）。该元素内嵌在<figure>元素内使用，用法如下所示：

```
<figure>
  
  <figcaption>One of our many cheesecakes</figcaption>
</figure>
```

如果<figcaption>放在了图片的下面，如本例所示，图题会显示在图片的右边；如果放在上面，图题就会显示在左边。

实战 1-11 浏览器兼容性

本章所用的截图取自 Safari 5.0.3 和 Mozilla Firefox 3.6.13。虽说 HTML5 可在所有主流浏览器上运行，但各浏览器对它的兼容性并不完全一致。