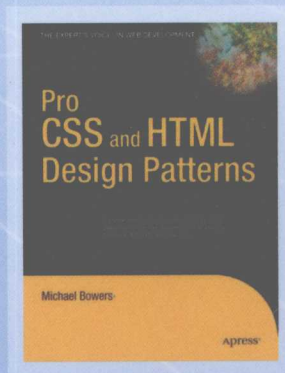


Pro CSS and HTML Design Patterns

精通CSS与HTML 设计模式

[美] Michael Bowers 著
刘申 朱瑜敏 鲁奇 译

- Amazon盛誉图书
- 业界专家十余年功力结晶，参透CSS与HTML
- 350多种可以立刻用于实战的设计模式
- Web设计与开发人员的必备参考书



人民邮电出版社
POSTS & TELECOM PRESS

TP393.092/1071

2008

TURING

图灵程序设计丛书

Web开发系列

Pro CSS and HTML Design Patterns

精通CSS与HTML 设计模式

[美] Michael Bowers 著

刘申 朱瑜敏 鲁奇 译

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

精通 CSS 与 HTML 设计模式 / (美) 鲍尔斯 (Bowers, M.)
著; 刘申, 朱瑜敏, 鲁奇译. —北京: 人民邮电出版社,
2008.9

(图灵程序设计丛书)

书名原文: Pro CSS and HTML Design Patterns

ISBN 978-7-115-18553-2

I. 精… II. ①鲍…②刘…③朱…④鲁… III. ①主页制
作—软件工具, CSS ②超文本标记语言, HTML—主页制作—
程序设计 IV. TP393.092 TP312

中国版本图书馆CIP数据核字 (2008) 第111350号

内 容 提 要

本书是一部非常实用的 CSS 与 HTML (XHTML) 解决方案手册。书中包含了 350 多种可以立即使用的设计模式 (涉及文本、背景、边框、图片、表格、布局等多方面), 并介绍了每种模式的原理和使用。每种设计模式、示例和源代码都经过了精心设计, 易于实现和使用。通过阅读此书, 可大大提高读者在 Web 设计和开发中的效率和创造力。

本书结构清晰, 示例丰富, 实践性强, 适用于所有 Web 开发和设计人员。

图灵程序设计丛书

精通CSS与HTML设计模式

◆ 著 [美] Michael Bowers

译 刘 申 朱瑜敏 鲁 奇

责任编辑 刘艳娟

◆ 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 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京铭成印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 32.25

字数: 759千字

印数: 1—4 000册

2008年9月第1版

2008年9月北京第1次印刷

著作权合同登记号 图字: 01-2007-4259号

ISBN 978-7-115-18553-2/TP

定价: 69.00元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Pro CSS and HTML Design Patterns* by Michael Bowers, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705.

Copyright © 2007 by Michael Bowers. Simplified Chinese-language edition copyright © 2008 Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

从事Web前端开发的人一定碰到过以下情况中的某一种：在Firefox中显示完全正常的页面，到IE下却发生了错乱；布局中某处一像素的偏差，调整过来反而会导致整个页面的变形；为表格赋予样式后，却完全达不到预期的效果；绝对定位元素跑到了相对定位的后面，而如果设置z-index，不同浏览器的解析方式还不一致，等等。HTML和CSS就是这么奇妙，看似简单的语言，在实际应用中，由于它们的不同排列组合，产生了千奇百怪的效果——这令很多开发者不知所措。如果你也正被它所困扰，那么本书就是你最好的选择了。

本书的英文名为*Pro CSS and HTML Design Patterns*——其中的“Patterns”说明了它所介绍的内容，在牛津的英英字典中对这个单词有这么一段解释：“a regular and intelligible form or sequence discernible in certain actions or situations”，瞧，人家解释得多精辟。而我们在前端开发中，最需要的也就是它所指的“Patterns”——能够在不同情况下告诉我们如何去做，以及会产生什么效果。本书共总结了350多种模式，这些模式都是从实际开发的不同情境中提炼出来的。所以，当你碰到某个问题的时候，你可以把它拆开来看，它一定会跟书中某一种或几种模式的组合相对应。当然，你不用（也不太可能）把这350多种模式都背下来，只要知道有哪些模式以及它们所对应的情境，当需要的时候再查阅本书就可以了。

本书可以作为一本实用的工具书，也可以作为一本很好的教学用书。因此本书既满足了有经验的Web开发者的需求，又非常适合前端初学者的入门。

本书由刘申、朱瑜敏、鲁奇共同翻译，刘申负责统稿。在这里要感谢FreeWheel的董彬，百度的李春平、咸立华、李国新等朋友在译者翻译期间所给予的大力帮助。最后要特别感谢宋薇对本书所提供的技术支持。由于本书较厚，外加翻译时间较紧，译文中难免有错漏之处，欢迎大家批评指正。

译者

2008年4月于哈工大

致 谢

我想感谢我的家人Teresa、Joshua和Sydney，正是因为他们的巨大牺牲，我才写出此书。我要特别感谢我的妻子Teresa。如果没有她不断的支持与鼓励，我是不可能写出这本书的。同样要感谢我的父母，是他们让我懂得了服务、努力工作和不断提高的价值。

我想对Apress团队卓有成效的工作表示诚挚的谢意：编辑Chris Mills、技术审稿人Paul Haines、项目经理Kylie Johnston、文字编辑Ami Knox、生产编辑Laura Esterman以及所有幕后的工作人员。

前言

本书是讲述如何使用CSS 2.1为HTML 4.01和XHTML 1.1添加样式的解决方案手册。书中包含了350多种可以立即使用的设计模式。每一种模式都是模块化的，并且可以进行自定义，可以将这些模式结合起来创建出无数种模式。

每一种设计模式都经过了严格的测试，并已证实可以应用于所有主要的网页浏览器，包括IE 7、IE 6、Firefox 2、Opera 9和Safari 2。

书中的全部内容均是可用和实用的。谁也不愿意浪费时间看那些用不上的东西。有了本书，你就不再需要为了能在多个浏览器中生效而使用各种奇技淫巧、没完没了地测试和不停地修改了。

使用一种设计模式很简单，将其复制粘贴到你的代码中然后修改一些值。本书将告诉你哪些值是可以修改的，对结果的影响如何，这样你就可以创建精确的样式，按需要进行布局，而无须担心它是否有效。

然而本书又不仅仅是一部解决方案手册。它系统地讲述了CSS的每一个可用的特征，并将这些特征与HTML结合创建出可复用的模式。每种模式都有一个直观的名称，便于查找、记忆和讨论。每种设计模式、例子和源代码都经过了精心设计，充分考虑了可访问性，运用了最佳实践。

你可以从头至尾通读本书，将它作为参考或用它寻找解决问题的方案。书中统一的版式会令你喜欢上它的，使用便是一种愉悦。

一般情况下书的左边页面是例子，右边是解释。每个例子都包含一个屏幕截图和所有相关的HTML和CSS代码，这样可以清楚地看到每种模式是如何工作的。每种模式的解释在例子的右侧书页上，你可以在阅读它如何工作的同时很方便地学习这个例子。

这种版式使本书的电子版也非常方便，因为你可以在同一屏幕中看到例子和它的解释。否则，就需要在页面间来回滚动，这对于电子书来说是很困难的。

每种设计模式都有一个名称，位于每页的顶端，便于查找、记忆和讨论。因为名称、屏幕截图、代码和解释都位于固定的位置，所以你可以迅速地翻查，以找到你所需的内容。

设计模式是通过主题（topic）组织的，并且与其他书不同的是，本书在具体应用上深入地讲解了所有实用的CSS规则。全部设计模式都是可访问的并且运用了最佳实践，因此本书值得通读，而且当你设计和编写代码时，可将本书放在手边作为一个极好的参考。

本书可以提高你在网页设计和开发中的效率和创造力。设计模式就像拼装玩具，可以用无数种方式将它们组装并创建出任何效果。它们也像工具箱中的工具，本书用上百种工具将你武装起来，使你可以迅速正确地解决问题。本书将展示如何可预见地创建模式——通过组合不同的可预见模式（predictable pattern），而不是局限于某一种解决方案。

适用读者

本书是为那些对CSS和HTML有所了解的人士编写的。它适用于以前阅读过CSS和HTML入门书的新手，也适用于曾经尝试过CSS但无法正确运用而放弃的设计者和开发者们，适用于想要提高CSS技术的专业人士，适用于所有希望快速完成设计的人——有了本书，他们就不用自己不断摸索，找出可以在所有浏览器中生效的方案了。

本书假设读者已经了解了编写CSS和HTML代码的基础知识。如果你只在像Dreamweaver或FrontPage这种所见即所得的设计软件中工作而从没处理过HTML或CSS代码，你可能会觉得本书中的代码有些难度。

如果你喜欢通过例子学习，喜欢了解代码是如何工作的，并且对CSS和HTML有所了解，你肯定会爱上这本书的。

在第17章和第20章中，有7种设计模式使用了JavaScript。为了完全理解它们，你需要了解JavaScript的基础知识，但是使用这些模式不用精通JavaScript。最重要的是，你不需要知道任何有关JavaScript的知识来理解和使用书中的340多种设计模式，因为它们根本与JavaScript无关。

创新

本书包含了一些创新的概念、术语和方法。这些其实都不是新东西：那些技术已经内建于主要的浏览器中，概念也已隐含于CSS规范中，术语也是经常使用的。它们的创新之处在于定义和使用它们的方式，其中展示了CSS和HTML的巨大潜力。换言之，它们的创新之处在于它们简化了学习、理解和使用CSS和HTML的过程。这些理念改变了你对CSS和HTML的看法，使一切变得不同。而且书中许多设计模式都属于创新，因为它们展示了通过结合使用属性及元素，可以解决许多以前难以解决的问题。

6种盒模型

书中的一个创新是认为CSS有6种盒模型而不是一种。官方规定CSS只有一种盒模型，它定义了一套常用的属性和行为。单一的盒模型是很实用的概念，但是它过于简单。这么多年来，我下了不少苦功才发现，盒模型的属性根据其类型的不同会产生不同的效果。

这就是很多人在使用CSS时疲于应付的原因之一。盒模型虽然看起来简单，但是当使用它的属性，如width时，它只是在某些情况下起作用，有时候产生的效果与期望的不同。例如，属性width为块状盒模型设置内部宽度，但是对于表格盒模型来说，它设置的其实是边框的外宽度，而在内联盒模型中，它完全不起作用。

我把一个复杂的盒模型分成了6种简单的盒模型，并定义了每种盒模型的行为，而不是把各

种不同的行为都当作一个非常复杂的盒模型的特例。第4章列出了6种盒模型，它们分别是内联、内联块状、块状、表格、绝对和浮动盒模型。因为你会知道正在使用哪种盒模型，所以也就知道了每一种盒模型的属性将会产生的行为。

而且每种盒模型都定义了它自己的元素排列与定位方式。例如，内联盒模型水平方向排列元素并且跨行包裹元素。块状盒模型垂直方向排列元素。表格盒模型在行和列中排列其元素。浮动盒模型水平方向排列元素，包裹其下方的浮动盒模型并且令内联盒模型和表格盒模型产生偏离。绝对和固定盒模型不排列元素。相反地，它们会脱离排列的元素，并相对于离它们最近的祖先元素进行定位。

盒模型的范围

书中的另一创新是，一种盒模型有3种定义范围的方式：它可以被设定尺寸、包裹或拉伸（见第5章）。每种盒模型都需要通过不同的属性及属性值的结合来设定尺寸、包裹或拉伸。第5章至第9章中不同的设计模式展示了这些工作是如何进行的。这3个术语都不是官方的CSS术语，但是它们都暗含在CSS 2.1规范中，在它的规则中提到了“size”、“shrink-to-fit”和“stretch”^①（设定尺寸、收缩至适当大小和拉伸）。

确实，设定尺寸、包裹和拉伸不是新的观点。真正的创新之处在于本书明确定义了这3个术语，并且说明了它们是如何成为CSS的基本特征和CSS设计模式的核心。

盒模型的定位

另外一个创新之处是，关于一种盒模型相对于其容器和兄弟元素有3种定位方式的观点。具体来说，它可以缩进（或外凸）、与其兄弟元素偏移或者与其所在容器对齐和偏移（见第8章）。CSS 2.1规范中讲了很多关于偏移（offsetting）定位元素的内容，也讲了一点关于对齐（aligning）元素的内容（见CSS 2.1规范的第9章），然而它没有讲元素如何缩进（indented），尽管此行为暗含在它的规则之中。

缩进、偏移和对齐是不同的行为。例如，一个缩进的盒模型是被拉伸的，它的外边距收缩其宽度，而一个对齐的盒模型是设定尺寸或被包裹的，它的外边距并不收缩其宽度。对齐和缩进的盒模型都与它们的容器对齐，然而偏移的盒模型可以偏移它们的容器或兄弟元素。

缩进、偏移和对齐不同类型的盒模型需要不同的属性及属性值的组合。第8章和第9章的设计模式显示了这些工作是如何实现的。

① 在CSS 2.1规范中，术语“size”和“sized”（尺寸和设定尺寸）在第8、9、10、11、17章和第18章中出现了15次。这些术语都代表一个意思，即盒模型有大小。

术语“shrink”和“shrink-to-fit”（收缩和收缩至适当大小）在CSS 2.1规范的第9章和第10章出现了9次。不同的盒模型可以包裹来适应其内容的观点隐含在规范的10.3.5～10.3.9节中以及17.5.2节中。

术语“stretch”和“stretched”（拉伸和被拉伸）在CSS 2.1规范的第9章和第16章中出现了4次。拉伸一盒模型到与其容器等宽的观点在下列引文中有提及（标注楷体），“很多盒模型的定位和尺寸是根据一种被称为包含块状元素的矩形盒子计算出来的”。（见规范的9.1.2节、9.3.1节和10.1节。）

列布局

另一个创新是发现、命名和阐述了内建于浏览器中的12种在表中进行列布局（见第16章）的自动技术。

所有的主要浏览器都包括这些强大的列布局特性。它们兼容于各主要的浏览器而且非常可靠。虽然不提倡用表格为页面布局，^①但是仍然需要布局表格数据（tabular data），这时就可以利用这些列布局让表格数据变得美观了。

流动布局

创新还包括流动布局（见第17章）。流动布局的概念并不新，但是创建它们的过程通常需要反复摸索。在第17章，我列出了4种简单的设计模式，你可以放心地用它们在所有主要浏览器中创建可预测的复杂流动布局。

这些设计模式，包括由外向内盒模型（Outside-in Box）、浮动区块（Floating Section）、浮动间隔（Float Divider）和流动布局（Fluid Layout），使用浮动和百分比宽度使其成为流动的，但这样做并没有产生使用这些技术通常遇到的问题，比如重叠的容器、浮动元素的错位以及百分比造成的浮动偏移。^②

流动布局设计模式运用表格的强大功能创建了列布局，却没有使用表格。这些布局优于表格，在显示器较窄时，会自动改变布局的宽度，并根据需要重新布局，把列变成行。

事件样式

还有一个创新是第17章中的事件样式JavaScript框架。这是一个简单的、功能强大的、开源的框架，它动态并交互地为文档赋予样式。它通过最新的最佳实践来保证HTML标记完全不受JavaScript代码的限制并且可访问，而且所有的赋予样式的工作都通过CSS完成。此外，使用这个框架，可以在JavaScript中使用与在CSS中同样的选择符来筛选元素。这大大地简化和统一了为动态HTML文档赋予样式和插入脚本的工作。

书中包含这个框架，展示了如何将JavaScript、CSS和HTML结合在一起，从而交互地使用样式。当然，如果你不想使用JavaScript，可以跳过第17章中的5个JavaScript设计模式和第20章中的两个JavaScript设计模式——剩下的340多种设计模式都没有用到JavaScript。

结合 HTML 和 CSS 创建设计模式

本书最后的也是影响最深远的创新是这样一个理念：将常见的HTML元素类型和CSS属性结合起来创建出新的设计模式。第2章定义了4种主要的HTML元素类型（结构化块状元素、终端块状元素、多目标块状元素和内联元素），并且第4章中将它们映射到6种盒模型（内联、内联块状、

① 使用表格布局会给视力残障用户带来可访问性的问题。不过，流动布局技术（见第17章）是完全可访问的，并且适应性比表格强很多。

② 当你浮动元素时，IE 6可能会产生一些bug。虽然大多数时间流动布局设计模式对于避免这些bug是有效的，但不幸的是，还没找到总能避开这些bug的办法。不过值得庆幸的是，IE 7修复了这些bug。

块状、表格、绝对和浮动)。

每一种设计模式都详细指出了它是如何应用于各种HTML元素类型的。换言之，设计模式不只是当使用特定元素时才有用，它还可以应用于HTML元素的所有等效类型。

例如，第18章中的浮动字母下沉(Floating Drop Cap)设计模式指定了一种使用块状和内联元素的模式，但它并没有指定必须使用哪些块状和内联元素(见代码清单1)。例如，可以用一个段落作为BLOCK元素，用一个span作为INLINE元素(见代码清单2)。或者可以用一个分区作为BLOCK元素，用作为INLINE元素等。

在一些特例中，设计模式可能指定一个实际的元素，如。在某个特定元素是最佳的、唯一的或极通用的解决方案时就会如此。即使在这些情况下，也可以将指定的元素换成同类型的另一元素，甚至可以使用另一种类型的元素，只要它能产生有效的XHTML并且你改变了它的盒模型使它兼容。(见第4章中的显示设计模式和盒模型设计模式，还有第11章中的块状化，第13章中的内联，第15章中的强制转换成表格、行和单元格。)

代码清单1 浮动字母下沉设计模式

HTML

```
<BLOCK class="hanging-indent">
  <INLINE class="hanging-dropcap"> text </INLINE>
</BLOCK>
```

CSS

```
*.hanging-indent { padding-left:+VALUE; text-indent:-VALUE; margin-top:±VALUE; }
*.hanging-dropcap { position:relative; top:±VALUE; left:-VALUE; font-size:+SIZE;
  line-height:+SIZE; }
```

代码清单2 浮动字母下沉实例

HTML

```
<p class="hanging-indent">
  <span class="hanging-dropcap" >H</span>anging Dropcap.
</p>
```

CSS

```
*.hanging-indent { padding-left:50px; text-indent:-50px; margin-top:-25px; }
*.hanging-dropcap { position:relative; top:0.55em; left:-3px; font-size:60px;
  line-height:60px; }
```

约定

每一种设计模式都使用以下几种约定。

- 大写的标记都应当替换成实际的值(可以看到，在代码清单1中的大写标记在代码清单2中都换成了实际的值)。
- 大写的元素(element)标记应当被替换成你所选的元素。如果元素名称是小写的，就不应当被替换，除非你能确保替换后能产生同样的盒模型。下面列出了几种常用的元素占位符。

- **ELEMENT**代表任何一种元素。
 - **INLINE**代表内联元素。
 - **INLINE_TEXT**代表包含文本的内联元素，例如、或者<code>。
 - **BLOCK**代表块状元素。
 - **TERMINAL_BLOCK**代表终端块状元素。
 - **INLINE_BLOCK**代表内联块状元素。
 - **HEADING**代表<h1>、<h2>、<h3>、<h4>、<h5>和<h6>。
 - **PARENT**代表任何可作为有效父元素的元素。
 - **CHILD**代表任何可作为有效子元素的元素。
 - **LIST**代表任意一种列表元素，包括、和<dl>。
 - **LIST_ITEM**代表任意一种列表项元素，包括、<dd>和<dt>。
- 大写的选择符（selector）都应当被替换。如果一个选择符包含小写的文本，则表示该部分选择符不应当被改变，除非你把HTML模式也同时改变了，比如改变了类的名字。下面列出了几个常用的占位符。
- **SELECTOR {}**代表所有选择符。
 - **INLINE_SELECTOR {}**代表用于选择内联元素的选择符。
 - **INLINE_BLOCK_SELECTOR {}**代表用于选择内联块状元素的选择符。
 - **BLOCK_SELECTOR {}**代表用于选择块状元素的选择符。
 - **TERMINAL_BLOCK_SELECTOR {}**代表用于选择终端块状元素的选择符。
 - **SIZED_BLOCK_SELECTOR {}**代表用于选择设定尺寸的块状元素的选择符。
 - **TABLE_SELECTOR {}**代表用于选择表格元素的选择符。
 - **CELL_SELECTOR {}**代表用于选择表格单元格元素的选择符。
 - **PARENT_SELECTOR {}**代表用于选择设计模式中父元素的选择符。
 - **SIBLING_SELECTOR {}**代表用于选择设计模式中子元素的选择符。
 - **TYPE {}**代表用于选择某一种指定元素的选择符，例如h1或span。
 - ***.CLASS {}**代表通过指定的类名来选择元素的选择符。
 - **#ID {}**代表通过指定的ID来选择元素的选择符。
- 大写标记的值（value）都应当被替换。如果值中包含小写的文本，则该部分值是不应当被替换的。下面是常用的值标记。
- 一些值是字面值，不用被替换，例如0、-9999px、1px、1em、none、absolute、relative和auto。这些值总是小写的。
 - **+VALUE**代表大于等于零的非负度量值，例如0、10px或2em。
 - **-VALUE**代表小于等于零的非正度量值，例如0、-10px或-2em。
 - **±VALUE**代表任何度量值。
 - **VALUEem**代表一个em度量值。
 - **VALUEpx**代表一个px度量值。

- **VALUE%**代表一个百分比度量值。
- **VALUE_OR_PERCENT**代表一个度量值或百分比值。
- **WIDTH STYLE COLOR**代表多属性值，例如border所需要的多属性，我用大写标记来表示每一个值。
- **url("FILE.EXT")**代表一个背景图片，其中你需要把FILE.EXT替换成图片的URL。
- **CONSTANT**代表一个有效的常量值。例如，white-space支持3种常量值：normal、pre和nowrap。为了方便，我通常用大写字母配合下划线（放在两个值之间）列出全部有效的常量值，例如NORMAL_PRE_NOWRAP。
- **ABSOLUTE_FIXED**代表一组常量值，你可以从中任取其一。每个常量值用下划线分隔。position的值的完整列表包括static、relative、absolute和fixed。如果某个设计模式只应用absolute和fixed，那么在模式中会指定position:ABSOLUTE_FIXED。如果它可适用于全部4种值，那么它会指定position:STATIC_RELATIVE_ABSOLUTE_FIXED或者position:CONSTANT。
- **-(TAB_BOTTOM + EXTRA_BORDER + EXTRA_PADDING)**是一个公式的例子，你可以用一个计算好的值来替换它们。公式中的大写标记与设计模式中其他地方出现的标记是一样的。例如，如果指派TAB_BOTTOM为10px，EXTRA_BORDER为10px，EXTRA_PADDING为10px，那么应当用-30px来替换这个公式。

使用本书

可以通过本书来掌握CSS。当通读本书后，你的CSS水平便会提升到一个新的水平，同时，你还能发现蕴藏在每一种设计模式中有价值的闪光点。每一章的组织方式是，后面的内容都是建立在前面内容基础上的。另一方面，因为每一章和每种设计模式都是独立的，所以你可以按照任意顺序逐一阅读来掌握某一个具体的主题或技术。

可以把本书当作一本参考书。本书对所有可用的CSS属性进行了阐述，并用实例展示了它们的使用方法。更重要的是，许多属性当与其他属性进行组合时，其行为会发生变化。每一种设计模式对产生特定结果所需的属性组合进行了区分和讨论。这使得本书不仅仅是一本CSS属性参考书，而且是一本属性组合的参考书。

可以通过本书的实例来学习。既然书中所有的实例都遵循最佳实践，因此，你能够通过钻研它们来学习好的习惯和新的技术。为了使通过实例学习本书更容易，可通过网站www.cssDesignPatterns.com来查看所有相关的设计模式。这让你轻松地通过例子学会某种CSS属性或功能是如何在不同情境下发挥作用的。

可以把本书当作一本实用方案手册，让它来帮助你创建设计样式或解决难题。设计模式是通过主题来组织的，这样你便能够迅速地找到相关的解决方案。

我还为本书添加了一些功能，当你需要它的时候可以很容易地找到某种解决方案。你可通过目录、索引、各章大纲和设计模式名称来迅速查找属性、模式、问题和解决方案。因为每个实例的截图都位于页面的固定位置，所以你还可以通过快速翻阅本书查看截图来找到解决方案。我发现人工翻阅是一种非常简便、快速和有效的查找解决方案的办法。

配套网站

本书英文原版的配套网站是cssDesignPatterns.com，提供了原书示例的源代码和效果网页^①。

本书的结构

第1章到第3章探讨的是CSS和HTML的一些基础知识。

- 第1章展示了设计模式是如何简化CSS的运用的。这里我演示了如何将简单的设计模式组合成复杂和强大的模式。我也对CSS的语法和层叠排序进行了回顾。另外，我展示了几个图表，它们可令CSS的使用更简单：实用的CSS网站链接列表，单页CSS属性总结；一个4页的实用CSS属性、取值和选择符的列表，根据它们使用地方的不同来排列；度量单位和字体大小的图表；两个样式表实例，展示了在所有浏览器中对元素样式的标准化问题；一个12步的向导，用于解决CSS的有关的各种问题。
- 第2章介绍了HTML背后的设计模式。在本章，我展示了使用HTML（包括XHTML）的最佳实践。也对可用HTML创建的结构类型进行了探讨，其中包括结构化块状、终端块状、多目标块状和内联。还讲述了如何通过CSS选择符对ID和属性进行选择。
- 第3章介绍了关于CSS选择符和继承的设计模式。这里我演示了选择符是如何作为HTML和CSS的桥梁的。我列出了关于类型、类、ID、定位、分组、属性、伪元素、伪类和子类选择符的设计模式。我还对CSS的继承进行了讲解。

第4章到第6章对6种盒模型进行了介绍。它们展示了每个HTML元素在不同盒类型（不是在每种盒类型中都能被解析的）中解析方式的不同之处。它们也演示了同一个属性在不同盒模型中所产生的不同结果，并且在每个盒模型中，元素的排列顺序是存在差异的。

- 第4章介绍了6种盒模型：内联、内联块状、块状、表格、绝对和浮动。
- 第5章对3种度量盒子的方式进行了探讨：设定尺寸（sized）、包裹（shrinkwrapped）和拉伸（stretched）。
- 第6章介绍了每种盒模型的属性：外边距（margin）、边框（border）、内边距（padding）、背景（background）、溢出（overflow）、可见性（visibility）、分页（pagebreak）。

第7章到第9章对盒子的排列和定位进行了讲解。

- 第7章介绍了5种定位模型（静态、绝对、相对、固定和浮动），并将它们与6种盒模型联系起来。
- 第8章对盒子定位的3种方式进行了介绍：例如，可以对一个盒子缩进、外凸或相对它的兄弟元素进行偏移，或者参照它的容器来对齐与偏移。
- 第9章对第7、8章的模式进行了组合：这种组合会生成50多种对元素进行定位的设计模式（仅仅针对绝对和固定定位两种方式）。

第10章到第12章对内联盒模型的排列、样式、空白、文本对齐与对象进行了详细的阐述。

^① 因为书中示例本身也是阐述的一部分，中文版已将示例代码中的文字翻译，相应的源代码和效果网页请读者到图灵网站（www.turingbook.com）本书配套网页访问（需事先注册）。——编者注

□ 第10章介绍了为文本赋予样式的一些属性，其中还包括了隐藏文本的3种设计模式（这几种设计模式对视障用户也是可访问的）。

□ 第11章展示了为内联内容添加水平和垂直空白的方法。

□ 第12章讲述了如何对内联内容水平和垂直对齐。

第13章到第14章详细讲解了块状元素和图片的排列方式以及怎样为它们赋予样式。

□ 第13章介绍了块状元素的相关内容，它以探讨块状元素的结构含义和如何可视化地显示此含义开始，涵盖了列表、内联块状元素、重叠的外边距、内嵌块状元素、块状空白和边缘块状元素。

□ 第14章介绍了图片的相关内容，例如图片地图（image map）、半透明图片、用图片替换文本、内嵌图片（sprite）、阴影图片和圆角图片。

第15章和第16章深入地介绍了如何对表格以及单元格赋予样式和布局。

□ 第15章讲的是表格，包括表格选择符、重叠的边框（collapsed border）、隐藏单元格（hiding cell）、垂直对齐单元格内的内容以及把内联和块状元素当作表格来显示。

□ 第16章介绍了12种用于布局表格列的模式，包括自动包裹表格列、为它们设定尺寸、按比例分配等。

第17章对如何用浮动元素创建流动布局进行了详细的介绍。

□ 第17章展示了如何创建浮动布局，它可根据不同的设备、字体、宽度、缩放进行自动调整。其中也介绍了如何用JavaScript创建交互的布局。

第18章至第20章介绍了如何通过组合设计模式为同一个问题创建出多种解决方案。每种方案会满足不同的需求，它们也各有利弊。除了作为实用的解决方案外，它们也展示了如何通过组合模式来解决任何设计上的问题。

□ 第18章介绍了字母下沉。这里我通过7种不同的设计模式组合，对7种字母下沉进行了讲解。

□ 第19章对强调框和引用进行了讲解。本章展示了5种强调框和3种引用。

□ 第20章介绍了提示框。其中我列出了3种可交互的提示框和8种文本提示框（比如，提醒标识）。

下载代码

所有的代码都可到www.cssDesignPatterns.com下载。

你可以通过在www.apress.com上搜索*Pro CSS and HTML Design Patterns*，到它的页面上去下载相关代码^①。在本书的详细介绍页面上有一个实例代码的链接，它是一个ZIP文件。可以用WinZip这样的解压缩软件对其进行解压。

代码的使用

代码用文件夹组织，每章一个文件夹。为了使每章的文件夹便于浏览，文件夹的名字都由章号和章名构成。每章的文件夹中包含的是实例文件夹，分别对应这章的每种设计模式。

^① 也可到图灵网站本书配套网页免费注册下载。——编者注

所以，你能够很轻易地找到实例，每个实例文件夹拥有与设计模式相同的名字。这样便可通过搜索文件夹的名字，方便而快捷地找到你想要的设计模式。因为在每个实例名的HTML中都含有它所属设计模式的名称以及描述，所以可以通过在HTML文件中搜索相应的关键字来找到设计模式。也可以在CSS文件中进行搜索，例如，其中使用了某个特殊的CSS属性（比如display）。

为了可在多个浏览器中查看实例，我在根文件夹下放了一个index.html文件，它可链向所有的设计模式文件夹。同理，在每个文件夹下也包括了一个名为index.html的文件，它可指向在那个文件夹下的所有设计模式。这些导航页使我们能够迅速找到并浏览每章中的每种设计模式。

每个实例文件夹下包含了所有让实例运行的文件。这使得在你自己的项目中使用这些实例也变得轻而易举：只要把整个文件夹复制过去，然后修改就可以了。你不用为关联和包括其他文件夹下的文件而担心。

实例文件夹下最重要的两个文件是example.html和page.css。example.html包含了例子中的XHTML代码。page.css是实例中的主要样式表。

每个实例还使用了一个名为site.css的CSS文件。它包含了一些非必要的字体和头部规则，从而令书中的实例的显示效果保持一致。

在某些例外的情况下，还使用了另一个CSS文件用于修正IE中的bug和非标准的行为。^① ie6.css包含了用于修正IE 6中问题的规则。ie7.css则用于修正IE 7中的问题。ie67.css是用于同时对付IE 6和IE 7的。这些文件中的规则都会覆盖掉page.css中的规则。

那7个JavaScript实例使用了5个JavaScript文件。在第17章的事件样式设计模式中有对它们的阐述。page.js是其中最重要的文件，因为它包含了例子中用到的JavaScript代码。剩下的JavaScript文件都是一些开源的函数库。

最后，每个实例文件夹中还包含了例子中用到的所有图片文件。

勘误表

你可以到www.cssDesignPatterns.com和www.apress.com上本书的详细介绍页面上浏览勘误表^②。

如果你在本书中发现了任何错误，并能告知于我，我将不胜感激。请用电子邮件把问题发到support@apress.com和support@cssDesignPatterns.com。^③

联系作者

可以通过电子邮件mike@cssDesignPatterns.com联系我。期待你的评论、建议和提问。

① 350多种设计模式中只有25个多用了文件。其中文件大多数只包含一条简单的规则，例如

{zoom:1;}

。尽管IE 6中存在很多bug、古怪和非标准的行为，但是我仅仅需要在25个设计模式中考虑相应的解决方案。这是因为，我非常仔细地设计本书中的模式，以避免那些问题。只有当我找不到任何可替代方案的时候，我才允许模式中存在例外。我不得不对数百种设计模式逐一进行测试，以找到不存在任何意外情况的模式。最后，因为IE 7修正了IE 6中的大部分bug，所以25个例外中只有4个适用于IE 7。

② 可在图灵网站本书配套网页上提交和访问本书勘误。——编者注

③ 如中文版中有任何问题，可发邮件到contact@turingbook.com。——编者注

目 录

第 1 章 设计模式：轻松搞定 CSS!	1	2.7 结构化块状元素	46
1.1 设计模式——精心组织的解决方案	2	2.8 终端块状元素	48
1.2 使用设计模式	2	2.9 多目标块状元素	50
1.3 应用样式表	7	2.10 内联元素	52
1.4 CSS 语法	8	2.11 类和 ID 属性	54
1.4.1 CSS 语法细节	9	2.12 HTML 空白	56
1.4.2 在 CSS 中使用空白	10	第 3 章 CSS 选择符与继承	59
1.4.3 使用属性值	11	3.1 本章大纲	59
1.5 使用层叠排序	14	3.2 类型、类和 ID 选择符	60
1.6 简化层叠	17	3.3 定位和群组选择符	62
1.7 CSS 和 HTML 链接	18	3.4 属性选择符	64
1.8 CSS 属性	19	3.5 伪类元素选择符	66
1.9 CSS 的属性和值：常见的	20	3.6 伪类选择符	68
1.10 CSS 的属性和值：内容	21	3.7 子类选择符	70
1.11 CSS 的属性和值：布局	22	3.8 继承	72
1.12 CSS 的属性和值：专有的	23	3.9 视觉继承	74
1.13 选择符	23	第 4 章 盒模型	77
1.14 变动的度量单位	24	4.1 本章大纲	77
1.15 固定的度量单位	24	4.2 显示	78
1.16 在 96dpi 下不同单位之间的比率	25	4.3 盒模型	80
1.17 在 96dpi 下常用的 font-size 值	25	4.4 内联盒模型	82
1.18 CSS 疑难解答	26	4.5 内联块状盒模型	84
1.19 使样式表规范化	28	4.6 块状盒模型	86
第 2 章 HTML 设计模式	31	4.7 表格盒模型	88
2.1 本章大纲	31	4.8 绝对定位盒模型	90
2.2 HTML 结构	32	4.9 浮动盒模型	92
2.3 XHTML	38	第 5 章 盒模型的尺寸	95
2.4 DOCTYPE	40	5.1 本章大纲	95
2.5 头部元素	42	5.2 宽度	96
2.6 条件样式表	44		