TURING 图灵程序设计丛书 Web开发系列

Apress®

CSS Mastery

Advanced Web Standards Solutions Second Edition

精通CSS

高级Web标准解决方案（第2版）

Andy Budd
[英] Simon Collison 著
Cameron Moll

陈剑瓯 译

- Amazon第一CSS畅销书全新改版
- 令人叫绝的CSS技术汇总
- 涵盖CSS 3和HTML 5

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Web开发系列

CSS Mastery
Advanced Web Standards Solutions Second Edition

精通CSS
高级Web标准解决方案
（第2版）

Andy Budd
[英] Simon Collison 著
Cameron Moll
陈剑瓯 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

精通CSS：高级Web标准解决方案：第2版 /（英）巴德（Budd,A.），（英）科利森（Collison,S.），（英）莫尔（Moll,C.）著；陈剑瓯译. -- 北京：人民邮电出版社，2010.5
（图灵程序设计丛书）
书名原文：CSS Mastery: Advanced Web Standards Solutions Second Edition
ISBN 978-7-115-22673-0

Ⅰ. ①精… Ⅱ. ①巴… ②科… ③莫… ④陈… Ⅲ. ①主页制作－软件工具，CSS－程序设计 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2010)第050871号

内 容 提 要

本书汇集了最有用的CSS技术，介绍了CSS的基本概念和最佳实践，结合实例探讨了图像、链接和列表的操纵，还有表单设计、数据表格设计、纯CSS布局等核心CSS技术。此外，书中着眼于创建跨浏览器的技术，讨论了bug及其捕捉和修复技术，还将所有技术组合成两个精彩的实例，讲述这些技术的工作原理和实际用法。

本书适合具有HTML和CSS基础知识的读者阅读。

图灵程序设计丛书

精通CSS：高级Web标准解决方案（第2版）

◆ 著 [英] Andy Budd Simon Collison Cameron Moll
译 陈剑瓯
责任编辑 傅志红
执行编辑 谢灵芝

◆ 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
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：17.5
字数：413千字 2010年5月第1版
印数：1－4 000册 2010年5月北京第1次印刷

著作权合同登记号 图字：01-2009-7278号

ISBN 978-7-115-22673-0

定价：49.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223
反盗版热线：(010)67171154

序

在网页设计的精彩世界里，实现同一个目标有千百种方法，而且新的方法还在不断地出现。对于特定的问题没有唯一正确的解决方法，丰富的选择使我们这些网页设计人员受益良多，同时也困扰着我们。这些选择虽然能使网页设计变得生动有趣，但同时也会令人无所适从。本书将帮助你减少麻烦，理清头绪。

Andy Budd多年来一直在编写、设计和宣讲基于标准的网页设计，我们现在有幸在本书中亲眼目睹他以简洁清晰的方式讲授最重要的CSS技术。本书提供了一套网页设计人员不可或缺的解决方案、技巧和建议。

有些图书中仅仅提出一种实现某一目标的正确方法，我很不喜欢这样的做法，Andy所做的正好相反，他为各种任务提供了多种方法，如对链接应用样式，创建标签页式导航，使用节省时间的CSS 3解决方案，或创建固定、流式的、灵活的布局，这些都有多种途径加以实现，书中还针对如何消除使用CSS设计时出现的那些恼人的浏览器bug给出了许多提示。掌握了常见设计元素的这些时髦漂亮的设计方法，你就可以做出更明智的选择。

不只如此，Andy还邀请两位出色的设计人员将这些技术组合在一起，通过两个实例研究向我们展示这些基本技术如何组合在一起。长期以来，我一直推崇Cameron和Simon的作品，看了他们写的两个绝佳实例，讨论流式布局、无懈可击的布局和灵活多样的样式解决方案，真的是受益匪浅。

好了，现在请开始深入研究并逐步消化这千百种设计方法，祝你早日成为精通CSS的高手。

Dan Cederholm，经典著作*Web Standards Solutions*一书的作者

前　言

尽管CSS资源的数量越来越多，但是在CSS邮件列表上仍然总是看到有人问同样的问题：如何让设计居中？最好的圆角框技术是什么？如何创建三列布局？

如果你熟悉CSS设计社区，那么寻找解决方案时无非就是回想一下某篇文章或某种技术曾在哪个网站重点介绍过。但是，如果你是CSS的初学者，或者没有时间阅读所有博客，那么这些信息可能并不好找。

CSS有些方面（比如定位模型和特殊性）比较晦涩，即使是有经验的CSS开发人员也会遇到问题。这是因为大多数CSS开发人员都是靠自学的，他们从各种文章和别人的代码中学习经验，而没有全面理解CSS规范。这也不奇怪，因为CSS规范本身十分复杂，常常还自相矛盾，它的目标读者是浏览器厂商而不是网页开发人员。

此外，还得应付浏览器问题。浏览器的bug和不一致性是现代CSS开发人员面对的一个最大问题。不幸的是，许多bug都没有很好地记载，它们的修复方法基本上只是在开发人员之间口口相传。你知道自己必须以某种方式做某件事，否则在某种浏览器中就会出问题。但是，你记不住是在哪种浏览器中会出问题，也说不清为什么会出问题。

所以，我产生了写这么一本书的想法。这本书将最有用的CSS技术汇总在一起，集中介绍实际的浏览器问题，从而弥补人们欠缺的CSS知识。本书会帮助你加快学习CSS的进程，使你的编码技术很快达到CSS专家的水平。

读者对象

本书适合具有HTML和CSS基础知识的任何人[①]阅读。无论你是刚刚接触CSS设计，还是已经开发纯CSS站点好几年了，书中都有适合你的内容。如果你已经使用CSS一段时间了，但还没有达到专家级水平，那么你能够从本书获得最大的收益。本书为你提供了各种实用的建议和示例，可以帮助你精通现代CSS设计。

本书结构

本书前3章讨论基本的CSS概念和最佳实践，帮助你轻松地入门。你将学习如何建立代码结

① 如果你不具备这些基础知识，可以阅读人民邮电出版社出版的《HTML XHTML与CSS基础教程》（第6版）。——编者注

构和添加注释，了解CSS定位模型的细节以及浮动和清理的工作原理。你也许已经掌握了其中的许多内容，但是可能会发现自己有遗漏或理解不充分的地方。因此，前3章是不错的CSS入门材料，也可以帮助你重温已经知道的知识。

介绍了基本知识之后，后面5章讨论核心CSS技术，比如操纵图像、链接和列表、设计表单和数据表格，以及进行纯CSS布局。每一章都由浅入深，最后讨论比较复杂的示例。在这几章中，你将学习如何创建圆角框、带透明阴影的图像、标签页式导航条和交互式按钮。许多情况下，我会先展示传统技术，然后说明如何用CSS制作出同样的效果。如果你想研究本书中的示例，可以从www.cssmastery.com或www.friendsofed.com下载所有示例代码①。

浏览器bug是许多CSS开发人员最头疼的问题，所以本书中的所有示例都着眼于创建跨浏览器的技术。此外，本书还用一整章讨论bug和bug修复。在这一章中，你将全面学习bug捕捉技术，学会在bug作乱之前就发现并消灭它，甚至还会学习是什么造成了IE中许多看似毫无规律的CSS bug。

最后两章是真正的“大餐”。Simon Collison和Cameron Moll是两位最杰出的CSS设计人员，他们将本书讨论的各种技术组合成两个精彩的实例来研究。从而，你不但会学习这些技术的工作原理，而且会看到如何将它们用在实际项目中。

本书可以从头到尾地阅读，也可以放在计算机旁边作为参考资料，随时查阅提示、技巧和技术，决定权在你。

本书约定

本书使用了几个约定，需要注意。本书采用了以下术语。

- “HTML”指HTML和XHTML这两种语言。
- 除非特别声明，“CSS”是指CSS 2.1规范。
- “Windows的IE 6和更低版本”指Windows的IE 5.0～6.0。
- “现代浏览器”是指最新版的Firefox、Safari、Opera、IE 7以及IE 7以上版本。
- 本书中的所有HTML示例都应该嵌套在一个有效文档的<body>中，同时，CSS包含在外部样式表中。偶尔为了尽量简短，HTML和CSS放在了同一个代码示例中。但是在真实的文档中，这些代码需要放在各自的位置上才能正常工作。

最后，对于包含重复数据的HTML示例，我们不会列出每一行，而是适时地使用省略号表示部分代码。

① 本书示例代码也可从图灵网站www.turingbook.com本书网页免费注册下载。——编者注

致　谢

感谢所有直接或间接地帮助我们撰写本书的人。

感谢我在Clearleft的朋友和同事，他们在我撰写本书的过程中提供了鼓励和反馈意见。特别感谢Natalie Downe为本书贡献其广博的知识和经验。他的支持和指导是无价的，我到现在还没搞清楚他究竟是怎样挤出那么多时间的。

感谢Chris Mills在我开始动笔时就一直引导着我，帮助我将想法变成现实。感谢所有不知疲倦地帮助本书按时出版的Apress出版社的工作人员，他们的奉献精神和职业态度令人敬佩。

感谢我的同事一直以来与我分享他们的开发经验，这些知识在不断美化Web环境。如果没有下面各位的工作，本书是不可能完成的：Cameron Adams、John Allsopp、Pachel Andrew、Nathan Barley、Holly Bergevin、Mark Boulton、Douglas Bowman、The BritPack、Dan Cederholm、Tantek Çelik、Joe Clark、Andy Clarke、Simon Collison、Mike Davidson、Garrett Dimon、Derek Featherstone、Nick Fink、Patrick Griffiths、Jon Hicks、Molly E. Holzschlay、Shaun Inman、Roger Johansson、Jeremy Keith、Ian Lloyd、Ethan Marcotte、Drew McLellan、Eric Meyer、Cameron Moll、Dunstan Orchard、Veerle Pieters、D. Keith Robinson、Richard Rutter、Jason Santa Maria、Dave Shea、Jeffrey Veen、Russ Weakley、Simon Willison、Jeffrey Zeldman，等等。

感谢我的博客的所有读者和过去两年中我在各种会议、讨论会和培训活动中遇到的所有人，他们的意见和思想丰富了本书的内容。

最后，感谢你阅读本书，希望本书能够帮助你将CSS技能提升到新的层次。

——Andy Budd

首先，感谢你选购本书，希望它助你在日复一日的工作中改进作品质量。从事这一行业的人潜能无限，总能让我深受鼓舞，你肯定也不会例外。

我响应Andy的话，也要感谢那些改造和美化Web的重要人物，他们让今日的Web超越了以往任何时候。若干年以后，这些人必将深受推崇和爱戴，比肩那些将人类首次送上月球的科学家们。

特别感谢Aaron Barker（aaronbarker.net）帮助我实现了实例研究中的几个jQuery和AJAX实例。

最重要的是，我要向我美丽的夫人Suzanne和四个儿子Everest、Edison、Isaac和Hudson表示最深的感激之情。没有他们的爱、耐心和大力支持，我不可能做出这些成就。

——Cameron Moll

我要感谢我的同事和朋友Gregory Wood帮我构思并实现了“Climb the Mountains”。他的设计总能给我启发，我以后就要成为他那样的设计师。我还要感谢我在Erskine设计公司的所有同事，他们对我狂热地从事这本书的写作睁只眼闭只眼，承担了不少份外工作。非常感谢Simon Campbell、Jamie Pittock、Glen Swinfield、Phil Swan、Vicky Twycross和Angela Campbell。

最重要的是，我要借此机会感谢我的母亲，还有几位上一版出版后逝去的亲人，我的祖父和外祖父，尤其是我父亲。尽管他们已离我而去，但我依然在努力让他们为我自豪。

——Simon Collison

版 权 声 明

目　录

第1章

基础知识

人类天生就是一种好奇的动物，我们都很喜欢摆弄新鲜玩意儿。这不，最近我买了一台新的iMac，还没看说明书呢，自己就先把它鼓捣了一番。我们喜欢自己去琢磨，对新东西形成自己的看法。我们会自己先胡乱摸索一阵子，发觉不对劲了，才会去查阅手册。

学习CSS（层叠样式表）最好的一种方式是直接开始使用它。实际上，我认为大多数人学习Web编程的过程都是这样：先从博客上看到了一些出色的效果，于是通过查看源代码研究它们是如何实现的，然后就在自己的个人网站上大胆尝试。人们几乎不会先去读完整的CSS规范，这些规范能把任何人送入梦乡。

修改别人的代码是很好的起步方法，但是如果不小心的话，就可能误解重要的概念，或者给日后造成问题。这一点我很清楚，因为我犯过好几次了。本章将讲解一些基本的但常常被误解的概念，并讨论如何让HTML和CSS保持清晰且结构良好。

在本章中，你将学习以下内容：

- 设计代码的结构；
- 有意义的文档的重要性；
- 命名约定；
- 什么时候使用ID和类名；
- 微格式；
- HTML和CSS的不同版本；
- 文档类型、DOCTYPE切换和浏览器模式。

1.1　设计代码的结构

大多数人不关心建筑物的地基。但是，如果没有坚固的地基，建筑物的主体也就不会存在了。虽然本书讨论的是高级的CSS技术，但是如果没有结构良好且有效的HTML文档，那么我们要做的许多事情都是不可能实现的（至少实现起来非常困难）。

在本节中，你将明白为什么结构良好且有意义的HTML文档在基于标准的CSS开发中非常重要，还将学习如何丰富文档的意义，从而让自己的开发工作更轻松。

1.1.1　标记简史

早期的Web仅仅是一系列相互链接的研究文档，使用HTML添加基本的格式和结构。但是，随着万维网的流行，HTML开始用来表现页面。人们结合使用字体和粗体标签来创建所需的视觉效果，而不只是用标题元素突出显示页面的标题。表格成了一种布局工具而不是显示数据的方式，人们使用块引用（blockquote）来添加空白而不是表示引用。Web很快就含义不清，成了字体和表格标签的大杂烩。Web设计者把这样的标记称为“标签汤”（见图1-1）。

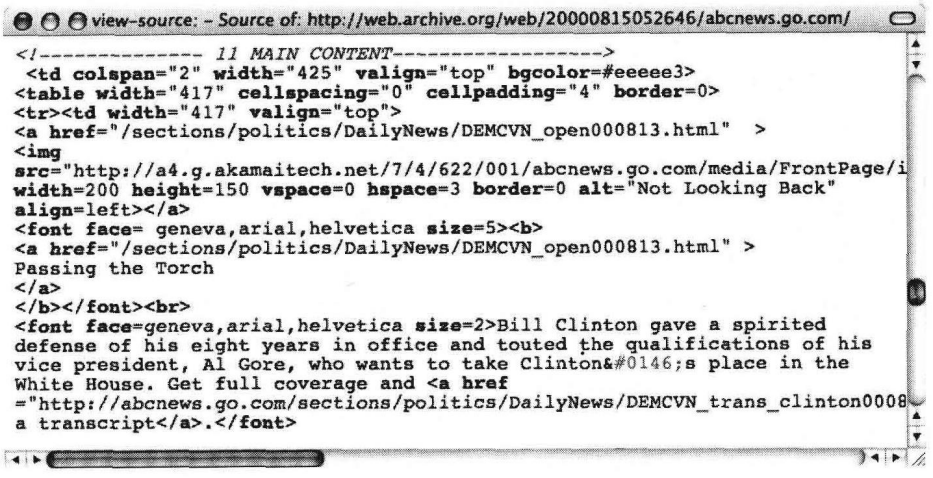

图1-1　abcnews.com（2000年8月14日）上的新闻头条页面的标记，它使用表格进行布局，对标题使用大的粗体字。代码结构不明，很难理解

网页变得越来越具表现力，代码却变得越来越难以理解和维护了。WYSIWYG（所见即所得）编辑器让设计者可以摆脱这些复杂性，它宣称可以提供全新的图形布局环境。遗憾的是，这些工具并没有使事情简化，反而添加了它们自己的复杂标记。使用FrontPage或Dreamweaver等编辑器能够通过简单的鼠标操作构建复杂的表格布局，但是嵌套的表格和“分隔线GIF”把代码弄得非常混乱（见图1-2）。更糟糕的是，这些布局极其脆弱，很容易被破坏。因为标记中有许多无意义的代码，很容易意外删除错误的标签，整个布局就可能会崩溃。另外，由于代码的复杂性，要找

到bug几乎是不可能的，这时从头编写页面往往比寻找bug更容易。如果是大型网站，情况会更复杂。因为网站的表现锁定到了各个页面，所以即使是最小的全站修改，也必须进行细致的“搜索并替换”。我曾经由于轻率地执行“搜索并替换”，导致弄坏了不止一个网站。因此，页面模板很快就不同步了，即使是简单的修改，也需要手工编辑网站上的每个页面。

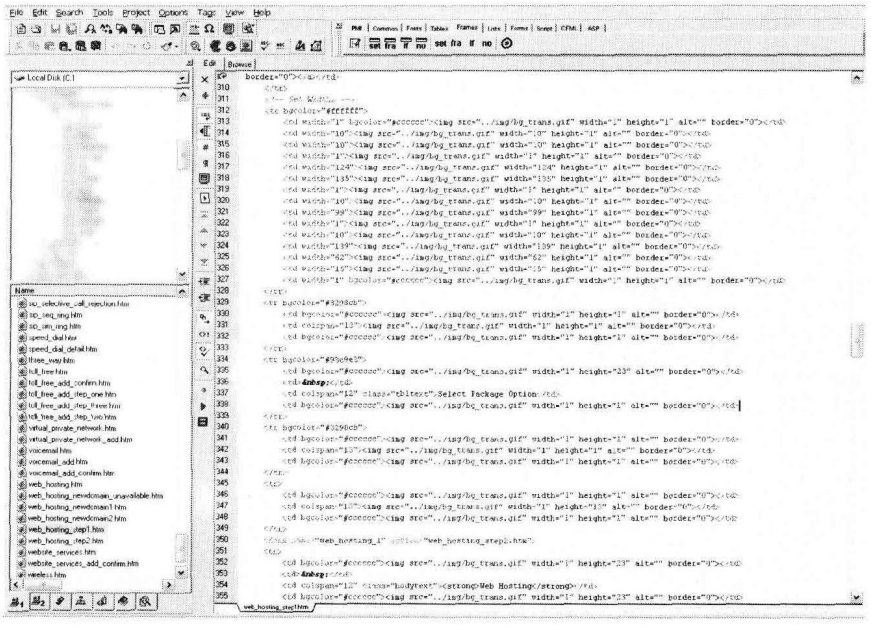

图1-2 使用大量分隔线GIF的基于表格的复杂布局（由Jeff L.提供）

表格本来不适合用来实现布局，所以David Siegel发明了一种聪明的解决方法。为了避免表格水平或垂直收缩，Siegel建议使用1像素的透明GIF。把这些隐藏的图像放在它们自己的表格单元格中，然后垂直或水平伸缩它们，这样就可以人为地控制单元格的最小宽度，从而保护布局不被破坏。这种图像也称为“shim GIF”，因为Dreamweaver中使用这个文件名。在基于表格的老式布局中常常会看到这种图像。好在这种做法已经过时了，现在代码中通常没有这些混乱的表现性元素。

HTML并没有被看做简单的标记语言，反而得到了复杂、混乱和容易出错的坏名声。因此，许多人害怕直接编写代码，这导致人们过分依赖于可视化编辑器，造成整整一代设计者不知道如何编写代码。

千禧年之际，Web设计业简直是一团糟，必须采取措施了。

就在这种背景下，CSS出现了。有了CSS，就可以控制页面的外观，并且将文档的表现部分与内容分隔开。现在，只需在一个地方进行站点修改，修改就会贯彻到整个系统。可以去掉表现标签（比如字体标签），而且可以使用CSS而不是表格来控制布局。标记返朴归真，人们又开始

对底层代码感兴趣了。

文档又有意义了。浏览器的默认样式可以被覆盖，所以可以将某些内容标为标题，而不需要为它指定大号的、加粗的、难看的字体。可以创建列表，而这些列表不一定要用一系列项目符号来表示，可以使用没有关联样式的块引用。开发人员开始按照HTML元素的原义使用它们，无需管它们的外观（见图1-3）。

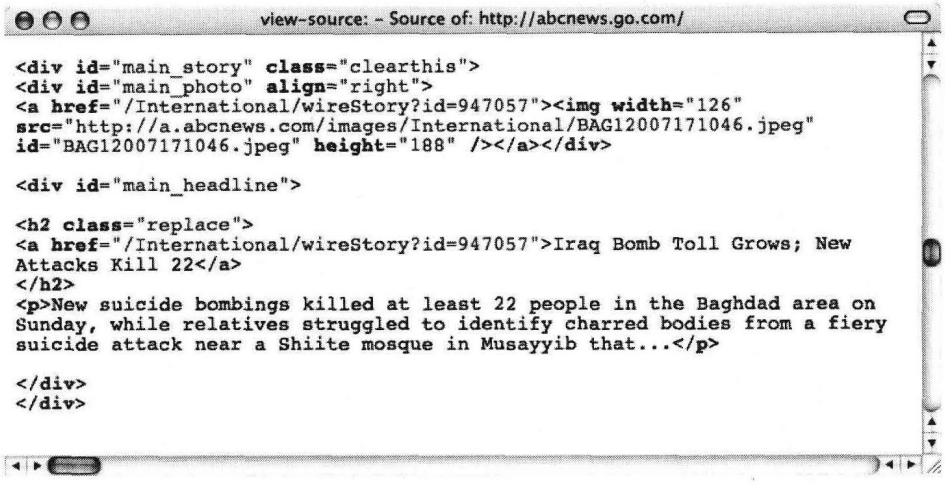

图1-3　今年早些时候abcnews.com的新闻头条页面的标记，它具有良好的结构，容易理解。虽然它仍然包含一些表现标记，但是与图1-1中的代码相比有了显著的改进

1. 意义的重要性

有意义的标记为开发人员提供了几个重要的好处。与表现性的页面相比，有意义的页面更容易处理。例如，假设需要修改页面中的一个引用，如果这个引用加上了正确的标记，那么很容易搜索代码，找到第一个块引用元素。但是，如果这个引用只是另一个段落元素标签，就很难寻找了。再举一个更复杂但不太现实的例子，假设你需要在主页中添加一栏，只需把新内容放在右边，然后在CSS中更新宽度。要想在基于表格的布局中完成相同的任务，就需要在表格中添加一列，修改colspan设置，修改所有单元格的宽度，修改所有shim GIF的宽度。实际上，为了完成这个简单的修改，必须修改整个页面结构。

除了人之外，程序和其他设备也可以理解有意义的标记（也称为语义标记）。例如，搜索引擎可以识别出标题（因为它被包围在h1标签中）并予以重视。屏幕阅读器的用户可以依靠标题进行页面导航。

对于本书来说，更重要的是，有意义的标记可以简便地将元素调整为你所需的样式。它在文档中添加结构并且创建一个底层框架。可以直接设置元素的样式，而不需要添加其他标识符，因此避免了不必要的代码膨胀。

HTML包含丰富的有意义元素，比如：

- h1、h2等；
- ul、ol和dl；
- strong和em；
- blockquote和cite；
- abbr、acronym和code；
- fieldset、legend和label；
- caption、thead、tbody和tfoot。

因此，如果元素有恰当的含义，就应该使用。

几年来，在博客、邮件列表和开发人员论坛上，对于使用CSS还是表格有许多争论。出现这些争论常常是由于一部分开发人员习惯了基于表格的方法，他们不愿意学习新的技能。我可以理解他们的想法，因为基于CSS的布局最初看起来的确很难，尤其是当前的方法似乎仍然可行时，他们更不愿意接受新东西。但是，CSS的好处非常多，包括代码更少、下载更快和更容易维护等。大多数专业开发人员已经认识到Web标准的好处，而且大多数组织都不愿意按老方法做。因此，如果你仍然在使用基于表格的布局，会越来越难找到工作。好在这些老习惯正在被摒弃，新一代开发人员已经很难忍受麻烦的表格布局了。

2. ID和类名

有意义的元素会提供很好的基础，但是可用元素并不全面。HTML 4是作为简单的文档标记语言创建的，而不是界面语言。因此，没有用于内容区域或导航栏等的专用元素。虽然可以使用XML创建自己的元素，但是由于它太复杂，这在目前还不太现实。

HTML 5能为开发人员提供更丰富的元素，有望解决其中一部分问题。这包括header、nav、article、section和footer等结构性元素，以及data inputs和menu元素等新的UI特性。为了准备迎接HTML 5，许多开发人员已经开始采用这些名称作为ID和类名的命名约定。

次优的解决方案是使用现有的元素，并且通过添加ID或类名给它们赋予额外的意义。这会在文档中添加额外的结构，并给样式提供有用的“钩子”（hook）。因此，可以建立一个简单的链接列表，并且给它分配ID nav，从而创建出定制的导航元素：

```
<ul id="nav">
  <li><a href="/home/">Home</a></li>
  <li><a href="/about/">About Us</a></li>
  <li><a href="/contact/">Contact</a></li>
</ul>
```

ID用于标识页面上的特定元素（比如站点导航），而且必须是唯一的。ID也可以用来标识持久的结构性元素，例如主导航或内容区域。ID还可以用来标识一次性元素，例如某个链接或表单