

Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with HTML5 and CSS3, Third Edition

无懈可击 的Web设计 (第3版)

——使用HTML 5和CSS 3提高网站的
灵活性与适应性

- Web设计大师Dan Cederholm力作最新版
- 畅销全球的Web标准设计经典著作
- 各大主流网站热评的CSS图书

[美] Dan Cederholm 著
马 跃 译

New
Riders



清华大学出版社

无懈可击的 Web 设计

——使用 HTML 5 和 CSS 3 提高网站的灵活性与适应性(第 3 版)

[美] Dan Cederholm 著
马 跃 译

清华大学出版社
北 京

Authorized translation from the English language edition, entitled *Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with HTML5 and CSS3*, Third Edition, 9780321808356 by Dan Cederholm, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2012.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by TSINGHUA UNIVERSITY PRESS, Copyright © 2012.

本书中文简体字版由 New Riders 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字：01-2011-8113

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

无懈可击的 Web 设计——使用 HTML 5 和 CSS 3 提高网站的灵活性与适应性(第 3 版)/

(美)西德霍姆(Cederholm, D.)著; 马跃 译. —北京: 清华大学出版社, 2012.5

书名原文: *Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with HTML5 and CSS3*, Third Edition

ISBN 978-7-302-28337-9

I. ①无… II. ①西… ②马… III. ①网页制作工具 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2012)第 044763 号

责任编辑: 王 军 韩宏志

装帧设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 张雪娇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185mm×230mm 印 张: 17.5

字 数: 339 千字

版 次: 2012 年 5 月第 3 版

印 次: 2012 年 5 月第 1 次印刷

印 数: 1~4000

定 价: 39.00 元

致 谢

感谢Michael Nolan在我和New Riders之间牵线，并坚定了我撰写本书的信念。同样，感谢Marjorie Baer在我刚开始撰写本书时给予的指导。

感谢Dave Roberts对本书进度的把握。

感谢Peachpit的每个人，他们都以某种方式参与了本书的创作。特别要感谢Rebecca Gulick管理本书的出版过程，保证了各个环节的通畅。

感谢本书第1版的技术编辑Ethan Marcotte，一名优秀的技术编辑。在与Ethan的合作过程中，我感到非常愉快，他的真知灼见可谓无价之宝。还要感谢本书第2版的技术编辑Virginia DeBolt和本书第3版的技术编辑Patrick H. Lauke。

感谢我的同事们，他们总是带给我启发和灵感，与我分享技术知识。正是基于下面这些人士(和其他许多未能在这里列出的人士)的辛勤付出，本书才得以面世：John Allsopp、Dan Benjamin、Holly Bergevin、Douglas Bowman、Andy Budd、Tantek Celik、Joe Clark、Andy Clarke、Mike Davidson、Todd Dominey、Jason Fried、John Gallant、Patrick Griffiths、Jon Hicks、Molly Holzschlag、Shaun Inman、Ryan Irelan、Richard Ishida、Roger Johansson、Jeremy Keith、Ian Lloyd、Drew McLellan、Eric Meyer、Cameron Moll、Matt Mullenweg、Mark Newhouse、Dunstan Orchard、Veerle Pieters、D. Keith Robinson、Richard Rutter、Jason Santa Maria、Christopher Schmitt、Dave Shea、Ryan Sims、Greg Storey、Jeffrey Veen、Josh Williams和Jeffrey Zeldman。

特别感谢Jon Hick创建了最初的3D CSS框模型演示(www.hicksdesign.co.uk/3dboxmodel/)。

感谢SimpleBits的读者和客户，是你们使我能全身心地投入到自己钟爱的事业中。

感谢Greater Beverly Adult Dodgeball联盟，我能够在每周二晚7:30有一个放松的机会。

感谢亲爱的父母，他们自始至终支持鼓励着我。感谢我哥哥Matt和他一家，以及我其他的亲戚和朋友们。

最后，一如既往地感谢我的贤内助Kerry。

最后，感谢您阅读本书。

前言

我必须承认这样一个事实：这世上并不存在什么完全无懈可击的网站。现在，在您合上这本书然后放回书架之前(希望您能多翻阅一下，谢谢)，请允许我解释一下。

如同警察穿上防弹衣保护自己一样，我们也可以采用一些方法来保护我们的Web设计方案。本书指导读者学会用多种策略来保护站点，使它们无懈可击：提高灵活性并能适应最恶劣的浏览环境。

关于“无懈可击”这个理念

在现实世界中，防弹衣并不能100%保证您不受伤害，但人们总是在不断地争取提升其防弹性能。毕竟，穿了防弹衣总是比不穿好。

这样的规则同样适用于Web设计以及本书中所描述的技术。通过增加网页的灵活性，以及采用必要的步骤来保证它在尽可能多的场景中都具备可读性，我们就给自己的作品赋予了一些与众不同的特色。这是一个循序渐进的过程，并且当采用了 Web标准相关技术(语义化的HTML和CSS)后，将能够更加容易构建出赏心悦目并且具有良好适应能力的设计方案。

近年来，采用基于CSS的页面进行布局已成了常规做法，学习如何正确并有效地运用CSS也变得越来越重要。其目标是充分利用这些技术的强大设计功能，比如：更精简的代码，增强的可用性，并且更加容易维护。

但用到了CSS和HTML并不一定就意味着一切都能变得更好。只有充分利用将页面核心内容和外观设计相分离而获得的灵活性，您才能顺利地创建出能够满足每个Web用户需要的最佳设计方案。不过，这里的“灵活性”的确切含义是什么呢？

本书重点内容

在开始思考本书的主题时，我意识到有两项重要内容构成了一个高质量且吸引人的

Web设计方案。一项是“可视组件”——每个人都能在最终页面上明显看到的部分。它是图像设计、颜色和排版的结合体。只需访问CSS Zen Garden网站(www.csszengarden.com), 您就会明显地感觉到使用 HTML和CSS完全可以实现引人入胜的视觉设计效果, 而且它们的应用已呈现出一派燎原之势。

第二项(但是同等重要)是“无懈可击的实现方式”。这也是本书讨论的重点: 明智地选择使用HTML和CSS创建网站, 可以享受到它们带来的全部好处。现在, 可以开始利用这些Web标准, 以及一些精妙的技巧, 创建出引人注目的, 同时具备尽可能高的灵活性、适应性以及亲和力的网站了。

对诸如HTML和CSS这样的Web标准的应用正在高速增长, 因此讨论如何以最佳方式利用这些资源也就变得越来越重要。

本书结构安排

本书的每一章都讲述了一种无懈可击的设计方法。每章开头都会列举一个网上现有的设计例子, 并会说明为什么那样做不是无懈可击的。然后使用 HTML 和CSS 对这个例子进行重构, 目标就是改善它的灵活性, 减少代码量。

这些例子大多数都是页面上的特定组件, 这样便于讨论如何将它们作为整体而设计得无懈可击。第9章将结合前面章节中讨论的全部技术, 创建一个完整的页面模板——回顾前面所学的知识, 并演示它们能怎样地结合在一起。

每章的范例都有一步步的详细过程, 能够让您轻松地学习——即使您是刚开始使用HTML和CSS的新手, 也可以很快上手。在整个过程中, 我都将解释为什么使用Web标准是有好处的, 并会指出如何使用各章的这些方法改善一个网站, 使其具备无懈可击的特性。



注意

这里使用“无懈可击”这个词来部分地表示“灵活性”——换句话说, 即设计方案能够轻松地适应各种不同的文字大小和内容数量, 能够随着这些变化自动扩展或缩小。

此外, 我们还可以(并且将会)从编辑、维护或者开发的角度探讨与灵活性相关的话题——修改内容, 更新及维护代码将会更加轻松, 并且不会影响设计效果。在准备页面的国际化版本时, 使用不同语言表达的内容长度千差万别, 此时保证页面设计效果不受影响十分重要, 对用户体验也会有很大的帮助。

最后，我们还站在浏览环境的角度讨论了灵活性。设计方案会如何影响网站内容和功能的完整性？我们必须确保所做的一切能够适应各种各样的设备和软件(可以是Web浏览器、屏幕阅读器或者移动设备)。在设计时考虑到灵活性意味着页面可被更多的设备和软件读取并解释。

本书范例的上下文环境

本书所有的范例都建立在一个基本的页面结构之上。换句话说，每章中出现的HTML和CSS代码都假定位于一个现有HTML 5文档的<body>和</body>标签之间。

例如，本书范例基于的基本框架可以是如下形式：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Page Title</title>
  <meta charset="utf-8" />
  <style type="text/css">
    ... example CSS goes here ...
  </style>
</head>
<body>
  ... example markup goes here ...
</body>
</html>
```

虽然为了方便，我们将CSS代码写在页面的<head>中，但是可以(而且可能应该)把它们写到单独的文件中。

使用HTML 5

本书中的所有示例都假设在HTML 5文档中编写，而且整本书会用到HTML 5中新增的一些对象。只需使用前面显示的简单的DOCTYPE，任何人都可以立即开始使用HTML 5。

然而，需要执行两个小步骤来确保所有浏览器都能识别并应用新的HTML 5元素。IE8及更低版本不能识别完全在HTML 5中引入的新元素。让这些浏览器知道新元素的最简单方法是使用Remy Sharp编写的一个简单的JavaScript shim: <http://remysharp.com/2009/01/07/html5-enabling-script/>。

将下面显示的代码添加到文档的<head>中，就可以为IE8及更低版本从Google加载该工具，从而可以在IE中使用这些新的HTML 5元素：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Page Title</title>
  <meta charset="utf-8" />
  <style type="text/css">
    ... example CSS goes here ...
  </style>
  <!-- enable HTML5 elements in IE7+8 -->
  <!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/
    → html5.js"></script>
  <![endif]-->
</head>
```

重置样式

浏览器通常会默认应用外边距、内边距和其他一些样式，为了消除它们的影响，本书将经常用到一个重置样式表：它们是一组CSS规则，为设置页面的样式建立了一个统一的基础。重置样式位于其他所有样式之前，如果没有使用外部样式表，它们位于主样式表的顶部，否则将先于其他样式表连接进来。

推荐使用Eric Meyer创建的重置样式表，他会根据最新的情况及时进行更新：

<http://meyerweb.com/eric/tools/css/reset/>

使用HTML 5时，需要知道一些较早的浏览器不支持HTML 5中的新元素，为它们将这些新元素声明为display:block;很重要，否则，在设置它们的样式时可能会遇到一些奇怪的问题。

下面所列为编写本书时，Eric最新的重置样式表，其中突出显示的部分是与HTML 5有关的代码：

```
/* http://meyerweb.com/eric/tools/css/reset/
   v2.0 | 20110126
   License: none (public domain)
*/
```

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
```

在框架中添加了HTML 5 JS shim和重置样式后，就可以向其中添加本书的例子了。每个例子都假设这些框架已经存在，这样我们就可以集中精力来处理要点，而不必每次都重复劳动。

本书中的常用术语

对于本书中的常用术语，在许多地方我都将以它们的缩写形式来指代。例如，使用“IE6/Win”比使用“Windows系统中的Internet Explorer version 6”要简便得多。以下是本书用到的一些表示浏览器的术语：

- IE = Windows系统中的Internet Explorer
- WebKit = Safari和Chrome使用的浏览器引擎
- Mozilla = Firefox使用的浏览器引擎

在描述各章节中例子的传统做法时，我经常提到“嵌套表格”以及“空白GIF占位图片(spacer GIF shim)”。这指的是在传统的构建网站的方式中，会用表格来创建每个像素都很完美但不够灵活的页面。在一个表格中嵌套另一个表格，虽然便于精确地对齐图片和文字，但页面代码却会堆积如山，并且存在大量可用性方面的问题。

术语“空白GIF占位图片”指的是将单张透明的GIF图片拉伸为各种尺寸，以便创建整页中的间隔区域、分栏以及分割线。一个并非无懈可击的网站的HTML代码中通常会包含很多这样的垃圾，使得页面变得臃肿不堪，这也将会成为维护人员的噩梦。

现在，可以采用更好的方法来实现同样的视觉效果：使用简洁的、富有语法意义的HTML代码和CSS。通过使用这些基于Web标准的技术，可以创建出既具有吸引人的效果，同时又具有灵活性，能够适应任何情况的设计方案。这就是“无懈可击的Web设计”。

目 录

第1章 灵活的文字	1
1.1 常见的方法	3
1.2 为什么这样设计不是无懈可击的	4
1.3 权衡我们的选择	6
1.3.1 长度单位	6
1.3.2 表示“相对大小”的关键字	6
1.3.3 百分比值	7
1.3.4 表示“绝对大小”的关键字	7
1.4 无懈可击的方法	7
1.4.1 关键字	8
1.4.2 放弃像素级别的精确度	8
1.5 为什么这样设计是无懈可击的	9
1.6 接下来的操作	9
1.6.1 设置基准值	9
1.6.2 使用百分比值来获取不同的尺寸	10
1.7 结合使用关键字和百分比值	13
1.7.1 设定一个中间的关键字基准值	13
1.7.2 慎用嵌套百分比值	15
1.7.3 百分比值的一致性试验	17
1.8 通过em实现灵活的文字	18
1.9 rem单位	19
1.10 本章小结	21
第2章 可伸缩的导航栏	23
2.1 常见的方法	24
2.1.1 功能强大的选项卡	25

2.1.2 通常的翻转效果	26
2.2 为什么这样设计不是无懈可击的	26
2.2.1 堆积如山的代码	26
2.2.2 不方便使用	27
2.2.3 可伸缩性的问题	27
2.2.4 不够灵活	27
2.3 无懈可击的方法	27
2.3.1 无样式的导航列表	29
2.3.2 两幅小图片	29
2.3.3 应用样式	30
2.3.4 采用浮动来解决问题	31
2.3.5 为选项卡定形	32
2.3.6 对齐背景图片	32
2.3.7 增加底边	34
2.3.8 悬停变换	35
2.3.9 选中状态	35
2.4 为什么这样设计是无懈可击的	36
2.5 使用CSS 3渐变而不使用图片实现	37
2.6 通过em来实现	40
2.7 其他示例	42
2.7.1 MOZILLA.ORG	42
2.7.2 斜杠	43
2.7.3 ESPN.COM的搜索栏	43
2.8 本章小结	45
第3章 可扩展的行	47
3.1 常见的方法	48
3.2 为什么这样设计不是无懈可击的	50
3.2.1 非必要的图片	50
3.2.2 固定的行高	50
3.2.3 臃肿的代码	51
3.3 无懈可击的方法	51
3.3.1 HTML 代码结构	51

3.3.2	标识出各部分	52
3.3.3	没有添加样式时的情形	53
3.3.4	添加背景	54
3.3.5	安排内容的位置	54
3.3.6	消失的背景	55
3.3.7	添加更多细节	56
3.3.8	四个圆角	58
3.3.9	文本和链接的细节	60
3.3.10	最后一步	62
3.3.11	针对IE7进行的修改	63
3.4	为什么这样设计是无懈可击的	64
3.4.1	代码结构与设计效果的分离	65
3.4.2	不再有固定不变的高度	65
3.5	通过border-radius实现	66
3.6	另一个自适应扩展例子	68
3.6.1	HTML 代码	69
3.6.2	创建两幅图片	69
3.6.3	添加 CSS	70
3.6.4	自动扩展	71
3.7	本章小结	72
第4章	巧妙的浮动效果	73
4.1	常见的方法	75
4.2	为什么这样设计不是无懈可击的	76
4.3	无懈可击的方法	77
4.3.1	对HTML代码无止境的抉择	77
4.3.2	使用定义列表	78
4.3.3	HTML代码结构	79
4.3.4	没有添加样式时的情形	80
4.3.5	为外围容器添加样式	80
4.3.6	标识图片	81
4.3.7	应用基本的样式	82
4.3.8	给图片定位	86

4.3.9 反向浮动	86
4.3.10 为任意长度的说明文字做准备	89
4.3.11 浮动自清除	90
4.3.12 尾声	92
4.3.13 切换浮动方向	94
4.3.14 表格效果	96
4.3.15 更换背景图片	99
4.3.16 应用box-shadow	101
4.3.17 其他清除浮动元素的方法	103
4.3.18 通过生成的内容进行清除	104
4.4 为什么这样设计是无懈可击的	107
4.5 本章小结	108
第5章 牢固的方框	109
5.1 常见的方法	111
5.2 为什么这样设计不是无懈可击的	113
5.3 无懈可击的方法	114
5.3.1 HTML 代码结构	115
5.3.2 图片策略	115
5.3.3 应用样式	117
5.4 为什么这样设计是无懈可击的	119
5.5 通过CSS 3实现	120
5.6 其他圆角实现技术	124
5.7 框提示	132
5.7.1 单圆角	132
5.7.2 圆角提示	135
5.7.3 无懈可击的箭头	136
5.7.4 CSS 中的限制孕育了技术上的创新	137
5.8 本章小结	138
第6章 页面在缺失图片或CSS的情况下仍然易读	139
6.1 常见的方法	140
6.2 为什么这样设计不是无懈可击的	143

6.3	无懈可击的方法	144
6.4	为什么这样设计是无懈可击的	146
6.5	使用样式或禁用样式	148
6.6	常见的方法	149
6.7	无懈可击的方法	150
6.8	Dig Dug 测试	152
6.9	无懈可击的工具	153
6.9.1	Favelet	154
6.9.2	Web Developer Extension	156
6.9.3	Web Accessibility 工具栏	158
6.9.4	Firebug	158
6.9.5	将验证作为一种工具	159
6.10	本章小结	162
第7章	可转换的表格	163
7.1	常见的方法	164
7.2	为什么这样设计不是无懈可击的	166
7.3	无懈可击的方法	167
7.3.1	HTML 代码结构	168
7.3.2	应用样式	174
7.4	为什么这样设计是无懈可击的	186
7.5	本章小结	187
第8章	流动布局和弹性布局	189
8.1	常见的方法	190
8.2	为什么这样设计不是无懈可击的	192
8.2.1	大量的代码	192
8.2.2	噩梦般的维护工作	193
8.2.3	并非最佳的内容顺序	193
8.3	无懈可击的方法	194
8.3.1	HTML 代码结构	194
8.3.2	创建栏：浮动与定位	195
8.3.3	应用样式	197

8.3.4	gutter	201
8.3.5	栏的内边距	203
8.3.6	设置宽度的最大值和最小值	209
8.3.7	滑动人造栏	213
8.3.8	三栏布局	216
8.4	为什么这样设计是无懈可击的	223
8.5	基于em的布局	224
8.5.1	一个弹性布局的例子	224
8.5.2	HTML 代码	226
8.5.3	CSS	228
8.5.4	一致性是最理想的	230
8.5.5	注意滚动条	230
8.6	本章小结	231
第9章	构成一个整体	233
9.1	目标	234
9.2	为什么这样设计是无懈可击的	235
9.2.1	流动的布局	236
9.2.2	灵活的文字	237
9.2.3	即使没有图片和CSS也可以使用页面	238
9.2.4	国际化	240
9.3	构建过程	240
9.3.1	HTML 代码结构	241
9.3.2	基本样式	242
9.3.3	布局结构	242
9.3.4	灵活的网格	244
9.3.5	设置max-width	244
9.3.6	页头	247
9.3.7	灵活的图片	249
9.3.8	侧边栏	251
9.3.9	CSS 3的多栏布局	255
9.3.10	媒体查询的魔力	256
9.4	本章小结	264

第 1 章

灵活的文字