

Bulletproof Web Design

无懈可击 的Web设计 (第2版)

——利用XHTML和CSS提高网站的
灵活性与适应性

- Web设计大师Dan Cederholm力作最新版
- 亚马逊五星级畅销书
- Web标准设计经典著作
- 各大主流网站热评的CSS图书
- 全彩印刷，易学易用

(美) Dan Cederholm 著
刘建宁 译



New
Riders

清华大学出版社

无懈可击

的Web设计

(第2版)

——利用XHTML和CSS提高网站的
灵活性与适应性

(美) Dan Cederholm 著
刘建宁 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 XHTML and CSS, Second Edition*, 9780321509024 by Dan Cederholm, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2008.

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 © 2009.

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: 01-2007-5027

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——利用 XHTML 和 CSS 提高网站的灵活性与适应性(第 2 版)/(美)西德霍姆(Cederholm, D.)著;刘建宁译. —北京:清华大学出版社, 2009.4

书名原文: *Bulletproof Web Design: Improving flexibility and protecting against worst-case scenarios with XHTML and CSS, Second Edition*

ISBN 978-7-302-19886-4

I. 无… II. ①西…②刘… III. ①超文本标记语言, XHTML—主页制作—程序设计②主页制作—软件工具, CSS

IV. TP312 TP393.092

中国版本图书馆 CIP 数据核字(2009) 第 045379 号

责任编辑: 王 军 李 阳

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 孟凡玉

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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: 北京鑫丰华彩印有限公司

装 订 者: 三河市漂源装订厂

经 销: 全国新华书店

开 本: 185×230 印 张: 18.5 字 数: 382 千字

版 次: 2009 年 4 月第 2 版 印 次: 2009 年 4 月第 1 次印刷

印 数: 1~4000

定 价: 59.90 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 026502-01

前言

我首先要承认这样一个事实：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(希望您能多翻一翻，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：获得增强的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，防弹衣并不能够 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可具备可读性，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 标准相关技术后，如语义化的 XHTML 和 CSS，将能够更加容易构建出外观吸引人并且具有良好适应性的设计方案。

近几年来，采用基于 CSS 的页面进行布局这一趋势保持着稳定增长，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。

但仅用到了 CSS 和 XHTML 并不一定就意



注意

我在这里使用“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短。

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。

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。

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适合每个 Web 用户的最佳设计方案。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 XHTML 和 CSS 完全可以实现吸引人的视觉设计效果，并且它们的应用已经相当普遍。

第二项(但是同等重要)是“无懈可击的实现方式”。这也是本书讨论的焦点：明智地选择使用 XHTML 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。

由于对诸如 XHTML 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。在各章节的开始我们会考察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 XHTML 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。在第 9 章中，我们会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——复习我们所学的知识，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——即使您刚刚才开始使用 XHTML 和 CSS。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。

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

XHTML 和 CSS 代码都假定位于一个现存 HTML 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 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" lang="en" xml:
lang="en">
<head>
    <title>Page Title</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
    <style type="text/css">
        ... example CSS goes here ...
    </style>
</head>

<body>
    ... example markup goes here ...
</body>
</html>
```

为了方便, 我们将 CSS 代码写在页面的<head>中, 但是对于旧的, 被淘汰的浏览器(如 Netscape Navigator 4.x), 这些 CSS 应该被隐去。由于这种隐藏很常见, 因此设计师会使用高级的 CSS 技术进行布局(如同整本书中我们所做的), 同时给无法处理 CSS 的老式浏览器提供一种没有样式但完全可读的文档表现方式。

针对老式浏览器隐藏 CSS 一般是通过使用 @import 方法引入外部样式表来实现的。例如, 如果将所有的样式写到一个名为 screen.css 的文件中, 就可以使用 @import 方法通过该文件的 URL 来引入外部样式表。因为像 Netscape 4.x 这样的老式浏览器并不识别 @import, 所以 screen.css 中包含的样式对它们而言就是隐藏的。



注意

这里我使用的是 XHTML 1.0 Transitional DOCTYPE, 但是您可以选择任何喜欢的 XHTML DOCTYPE。如果您不知道 DOCTYPE 究竟是什么? 那么不用着急, 6.7.5 节中将会介绍有关这方面的更多内容。

```
<head>
  <title>Page Title</title>
  <meta http-equiv="Content-Type" content="text/html;
  charset=utf-8" />
  <style type="text/css">
    @import url("screen.css");
  </style>
</head>
```

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“IE5/Win”比使用“Internet Explorer version 5 for Windows”要轻松得多。以下是一些浏览器版本 / 系统平台缩写的约定：

- IE5/Win = Internet Explorer version 5.0 and 5.5 for Windows
- IE6/Win = Internet Explorer version 6 for Windows
- IE5/Mac = Internet Explorer version 5 for Macintosh

在描述各章节中例子的通常做法时，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毫无灵活性的页面。在一个表格中嵌套另一个表格，虽然可以使得精确地对齐图片和文字更加容易，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，这也将会成为维护人员的噩梦。

现在，有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	2
1.2 权衡我们的选择	5
1.2.1 长度单位	6
1.2.2 表示“相对大小”的关键字	6
1.2.3 百分比值	6
1.2.4 表示“绝对大小”的关键字	7
1.3 无懈可击的方法	7
1.3.1 关键字	7
1.3.2 放弃像素级别的精确度	8
1.3.3 需要解决的两个问题	9
1.3.4 简化的 Box Model Hack	11
1.4 为什么说它是无懈可击的	12
1.5 接下来的操作	12
1.5.1 设置基准值	12
1.5.2 使用百分比值来获取不同的尺寸	13
1.6 结合使用关键字和百分比值	16
1.6.1 设定一个中间的关键字基准值	16
1.6.2 谨慎使用嵌套百分比值	18
1.6.3 百分比值的一致性试验	20
1.7 通过 Em 实现灵活的文字	21
1.8 本章小结	23

第2章 可伸缩的导航栏	25
2.1 常见的方法	26
2.1.1 功能强大的选项卡	27
2.1.2 通常的翻转效果	28
2.2 为什么这样做不是无懈可击的	28
2.2.1 堆积如山的代码	28
2.2.2 不方便使用	29
2.2.3 可伸缩性的问题	29
2.2.4 缺乏灵活性	29
2.3 无懈可击的方法	29
2.3.1 无样式的导航列表	30
2.3.2 两幅小图片	31
2.3.3 应用样式	31
2.3.4 采用浮动来解决问题	32
2.3.5 为选项卡定形	33
2.3.6 对齐背景图片	34
2.3.7 增加底边	36
2.3.8 悬停变换	37
2.3.9 选中状态	38
2.4 为什么这样做是无懈可击的	38
2.5 通过 Em 来实现	39
2.6 其他示例	41
2.6.1 MOZILLA.ORG	41
2.6.2 斜杠	41
2.6.3 ESPN.COM 的搜索栏	42
2.7 本章小结	44
第3章 可扩展的行	45
3.1 常见的方法	46
3.2 为什么这样做不是无懈可击的	48
3.2.1 非必要的图片	48

3.2.2	固定的行高	48
3.2.3	臃肿的代码	49
3.3	无懈可击的方法	49
3.3.1	HTML 代码结构	49
3.3.2	标识出各部分	50
3.3.3	没有添加样式时的情形	51
3.3.4	设定背景	52
3.3.5	安排内容的位置	52
3.3.6	消失的背景	54
3.3.7	添加更多细节	55
3.3.8	四个圆角	58
3.3.9	文本和链接的细节	59
3.3.10	最后一步	61
3.3.11	针对 IE7 进行的修改	63
3.4	为什么这样做是无懈可击的	64
3.4.1	代码结构与设计效果的分离	64
3.4.2	不再有固定不变的高度	65
3.5	自适应扩展的另一个例子	66
3.5.1	HTML 代码	67
3.5.2	创建这两张图片	67
3.5.3	添加 CSS	68
3.5.4	自动扩展	70
3.6	本章小结	70
第 4 章	巧妙的浮动	73
4.1	常见的方法	75
4.2	为什么这么做不是无懈可击的	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	没有样式时的情形	81
4.3.5	为外围容器增加样式	81
4.3.6	标识图片	82
4.3.7	应用基本的样式	83
4.3.8	给图片定位	87
4.3.9	反向浮动	88
4.3.10	为任意长度的描述文字做准备	91
4.3.11	浮动自清除	92
4.3.12	尾声	96
4.3.13	切换浮动方向	99
4.3.14	表格效果	101
4.3.15	更换背景图片	104
4.3.16	其他清除浮动元素的方法	107
4.3.17	通过内容生成来轻松处理清除	108
4.4	为什么这样做是无懈可击的	112
4.5	本章小结	112

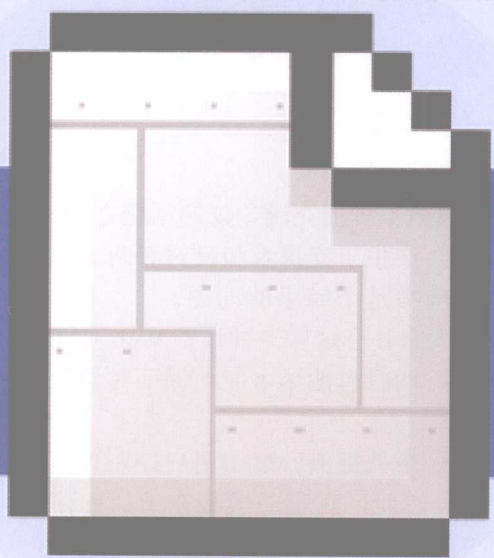
第 5 章 牢固的方框 113

5.1	常见的方法	114
5.2	为什么这样做不是无懈可击的	117
5.3	无懈可击的方法	118
5.3.1	HTML 代码结构	118
5.3.2	图片策略	119
5.3.3	应用样式	121
5.4	为什么这样做是无懈可击的	124
5.5	其他圆角实现技术	124
5.6	提示框	133
5.6.1	圆角提示	136
5.6.2	无懈可击的箭头	137
5.6.3	CSS 中的限制孕育了技术上的创新	138
5.7	本章小结	139

第 6 章 页面在缺失图片或 CSS 的情况下仍然易读	141
6.1 常见的方法	142
6.2 为什么这样做不是无懈可击的	145
6.3 无懈可击的方法	146
6.4 为什么这样做是无懈可击的	147
6.5 使用样式或禁用样式	150
6.5.1 10 秒钟可用性测试法	150
6.5.2 常见的方法	151
6.5.3 无懈可击的方法	152
6.6 Dig Dug 测试	154
6.7 无懈可击的工具	155
6.7.1 Favelets	155
6.7.2 Web Developer Extension	157
6.7.3 Web Accessibility 工具条	158
6.7.4 FIREBUB	158
6.7.5 把检验作为一种工具	159
6.8 本章小结	162
第 7 章 可转换的表格	165
7.1 常见的方法	166
7.2 为什么这样做不是无懈可击的	158
7.3 无懈可击的方法	169
7.3.1 HTML 代码结构	169
7.3.2 应用样式	175
7.4 为什么这样做是无懈可击的	191
7.5 本章小结	192
第 8 章 流动布局和弹性布局	195
8.1 常见的方法	196
8.2 为什么这样做不是无懈可击的	198
8.2.1 大量的代码	199

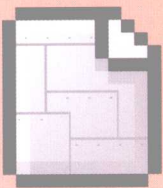
8.2.2	噩梦般的维护工作	199
8.2.3	并非最佳的内容顺序	199
8.3	无懈可击的方法	200
8.3.1	HTML 代码结构	200
8.3.2	创建栏: 浮动与定位	201
8.3.3	应用样式	203
8.3.4	gutter	207
8.3.5	栏的留白	210
8.3.6	设置宽度的最大和最小值	215
8.3.7	滑动人造栏	218
8.3.8	三栏布局	222
8.4	为什么这样做是无懈可击的	230
8.5	基于 Em 的布局	230
8.5.1	一个弹性布局的例子	231
8.5.2	HTML 代码	232
8.5.3	CSS	234
8.5.4	一致性是最理想的	237
8.5.5	注意滚动条	237
8.6	本章小结	238
第 9 章	构成一个整体	239
9.1	目标	240
9.2	为什么这样做是无懈可击的	241
9.2.1	流动的布局	241
9.2.2	灵活的文字	242
9.2.3	即使没有图片和(或)CSS, 页面也没问题	243
9.2.4	国际化	245
9.3	构建过程	245
9.3.1	HTML 代码结构	246
9.3.2	基本的样式	248
9.3.3	布局结构	248

9.3.4 侧边栏背景	249
9.3.5 页头	251
9.3.6 信息行	254
9.3.7 gutter	256
9.3.8 内容栏	257
9.3.9 侧边栏	266
9.3.10 页脚	272
9.4 针对 IE 的 CSS 调整	273
9.4.1 hack 管理	274
9.4.2 页脚补丁	275
9.4.3 自清除问题的补丁	275
9.5 本章小结	276
网友评论精选	277



1

灵活的文字



使用关键字和百分比来设定页面文字的大小，以允许用户控制并实现最大程度的灵活性

在 Web 设计中，“如何设定页面文字的大小”是为数不多的充满争议的话题之一。它使初学者感到困惑，甚至引发了设计者之间的争论，产生了支持者和反对者两大阵营。这样说也许夸张了点，不过对许多人而言，它确实是个棘手的问题。

我无意要当争论的调停人，在本章中，我只想和大家分享这样一种设定 Web 页面中文字大小的策略：它具有高度的灵活性并易于实现，同时也能尽可能地保证对设计细节的全权控制。

灵活设置页面文字的大小，是本书中各个范例的一个关键点。赋予用户控制页面文字大小的能力，也就是提供了更好的页面可读性。然而，真正的挑战在于：如何在保持灵活性的同时仍能在文字周围进行精确巧妙的细节设计。本书中包含大量的示例，在您读完本书之后，应该就能很好地应对这类挑战了。

首先，让我们来看看目前常见的设计方式，并分析这些设计方式为什么缺乏灵活性。

1.1 常见的方法

关于文字大小设定的常见方法，这里以一个实际站点为例：eyeglasses.com。显然，如果这个网站注重页面可读性以及用户可控性，那么将能更加利于视力不佳的用户进行浏览(见图 1-1)。

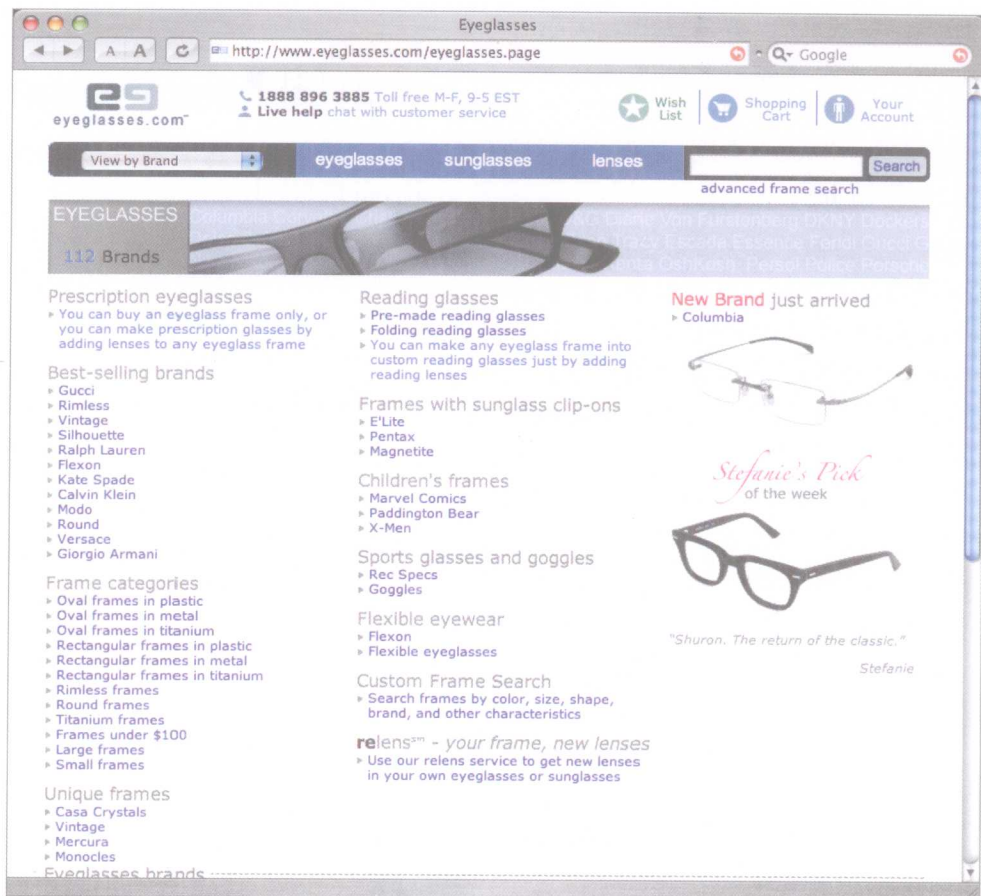


图 1-1 我们以 eyeglasses.com 网站为例来演示设定页面文字大小的常见做法
(本图截取自 2004 年 12 月)

eyeglasses.com 网站很好地使用了 CSS，其设计很精美。和许多其他站点一样，它在页面的 `<body>` 元素中使用 `font-size` 属性为整页的文字设定了一个以像素为单位的基准。

```
body {
    font-size: 11px;
}
```

通过在 `<body>` 元素中为页面上的文字设定一个基准(本例中是 11px)，设计师可以确保整个页面中所有的文字都将是 11 个像素那么大(除非被后继的属性值覆盖)。使用像素作为单位的好处在于，不论使用哪种浏览器或者设备来查看页面，文字看起来大小