Bulletproof Web Design

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

- Web 设计大师最新力作
- 亚马逊五星级畅销书
- 提供基于Web标准的设计策略
- 全彩印刷，易学易用

(美) Dan Cederholm 著
常 可 译

New Riders

清华大学出版社

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

(美) Dan Cederholm 著
常可 译

清华大学出版社
北京

EISBN：0-321-34693-9
Bulletproof Web Design：Improving flexibility and protecting against worst-case scenarios with XHTML and CSS
Dan Cederholm

北京市版权局著作权合同登记号　图字：01-2006-1596

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/(美)西德霍姆(Cederholm，D.)著；常可 译. —北京：清华大学出版社，2006.11
书名原文：Bulletproof Web Design：Improving flexibility and protecting against worst-case scenarios with XHTML and CSS
ISBN 7-302-13030-2
Ⅰ.无… Ⅱ.①西…②常… Ⅲ.计算机网络—程序设计 Ⅳ.TP393.09

中国版本图书馆 CIP 数据核字(2006) 第 049564 号

出 版 者：清华大学出版社　　地　　址：北京清华大学学研大厦
　　http://www.tup.com.cn　　邮　　编：100084
　　社 总 机：010-62770175　　客户服务：010-62776969
组稿编辑：王　军　　文稿编辑：于　平
封面设计：康　博　　版式设计：康　博
印 刷 者：北京鑫丰华彩印有限公司　　装 订 者：三河市化甲屯小学装订二厂
发 行 者：新华书店总店北京发行所
开　　本：185×230　印张：16.5　字数：331 千字
版　　次：2006 年 11 月第 1 版　2006 年 11 月第1 次印刷
书　　号：ISBN 7-302-13030-2/TP·8270
印　　数：1～4000
定　　价：49.80 元

前　言

我首先要承认一个事实，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售书架之前，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。本书指导读者学会多种策略以保护站点—— 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具备可读性，我们就给自己的作品带来了一些与众不同的特色。这是一个不断进行的过程，并且当采用了 Web 标准相关技术，例如语义化的 XHTML 和 CSS，将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 的页面布局这一趋势保持了稳定增长，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在设计方面的强大能力，比如：更精简的代码，增强的可用性，并且更加容易维护。

但是，仅仅是用到了 CSS 和 XHTML 并不一定就意味着一切都变得更好。充分利用将页

注意：

我这里使用“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伸缩。

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和灵活性相关的话题——修改内容以及更新并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实际场景。

面核心内容和外观表现分离而获得的灵活性，您才能顺利地创建出适合每个Web用户的更好的设计方案。不过，这里的“灵活性”的确切含义是什么呢？

本书重点

在我开始思考这本书的主题的时候，我意识到有两项重要内容构成了一个高质量和吸引人的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。它是图片设计、颜色和排版的结合体。只需访问CSS Zen Garden网站(www.csszengarden.com)，就会显而易见地感觉到使用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，同时具备尽可能好的灵活性、适应性以及亲和力的网站了。

对诸如XHTML和CSS这样的Web标准的采用已经在高速增长，讨论如何以最佳的方式利用这些资源也就变得越来越重要。

本书结构

本书的每一章都讲述了一种无懈可击的设计指南。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。然后我们使用XHTML和CSS重构这个例子，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“构成一个整体”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。整个过程中，我都将解释为什么使用Web标准是有好处的，并会指出各章的这些指南如何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

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下面这样：

```
<!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，但是对于老旧的，被淘汰的浏览器(例如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的老旧浏览器提供一个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。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中包含的样式就对它们隐藏了。

```
<head>
  <title>Page Title</title>
```

注意：

这里我使用的是XHTML 1.0 Transitional DOCTYPE，但是您可以选择任何您喜欢的XHTML DOCTYPE。不知道DOCTYPE究竟是什么？不用着急，6.6.4节我会介绍更多这方面的内容。

```
    <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

本书中的许多地方我都将以它们的缩写形式来指代某种浏览器。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的列表：

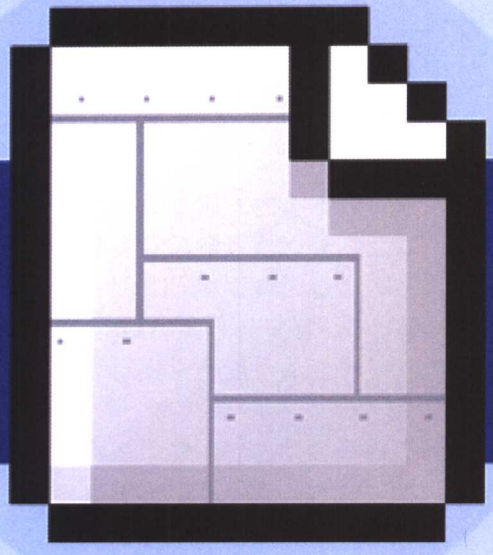
- IE5/Win = Internet Explorer version 5.0 and 5.5 for Windows
- IE6/Win = Internet Explorer version 6 for Windows
- IE5/Mac = Internet Explorer version 5 for Macintosh

在描述各章节中例子的通常做法时，我经常会提到“嵌套表格”以及“空白GIF占位图片”。这指的是在构建网站的传统方式中，会用表格创建每个像素都很完美但是毫无灵活性的页面。在一个表格中嵌套另一个表格，使得精确地对齐图片和文字更加容易，但是页面代码却堆积如山，并且存在大量可用性方面的问题。

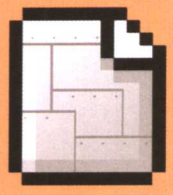
术语“空白GIF占位图片”指的是将单张透明的GIF图片拉伸为各种尺寸，以便创建整页中的间隔区域、分栏以及分割线。一个并非无懈可击的网站的HTML代码中会包含很多这样的垃圾，使得页面臃肿，也将成为维护人员的噩梦。

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



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 设计领域中，“如何设定页面文字的大小”是为数不多的充满争议的著名话题之一。从历史上看，它以让初学者困惑，引发争论以及划分支持者和反对者阵营而著称。这么说也许夸张了点，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 页面的文字大小的策略：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。

让文字大小的设定具有灵活性，是本章中各个范例的核心指导思想。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，应该就能轻松应付这类挑战了。

让我们首先来看看目前常见的设计方式，并分析为什么这样做缺乏灵活性。

1.1 常见的方法

关于文字大小设定的通常方法，这里可以举一个实际站点为例：eyeglasses.com。显然，如果这个网站注重页面可读性以及用户的可控性，那么浏览该网站的用户便更能因此受益(因为他们应该大都视力不佳)(见图 1-1)。

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

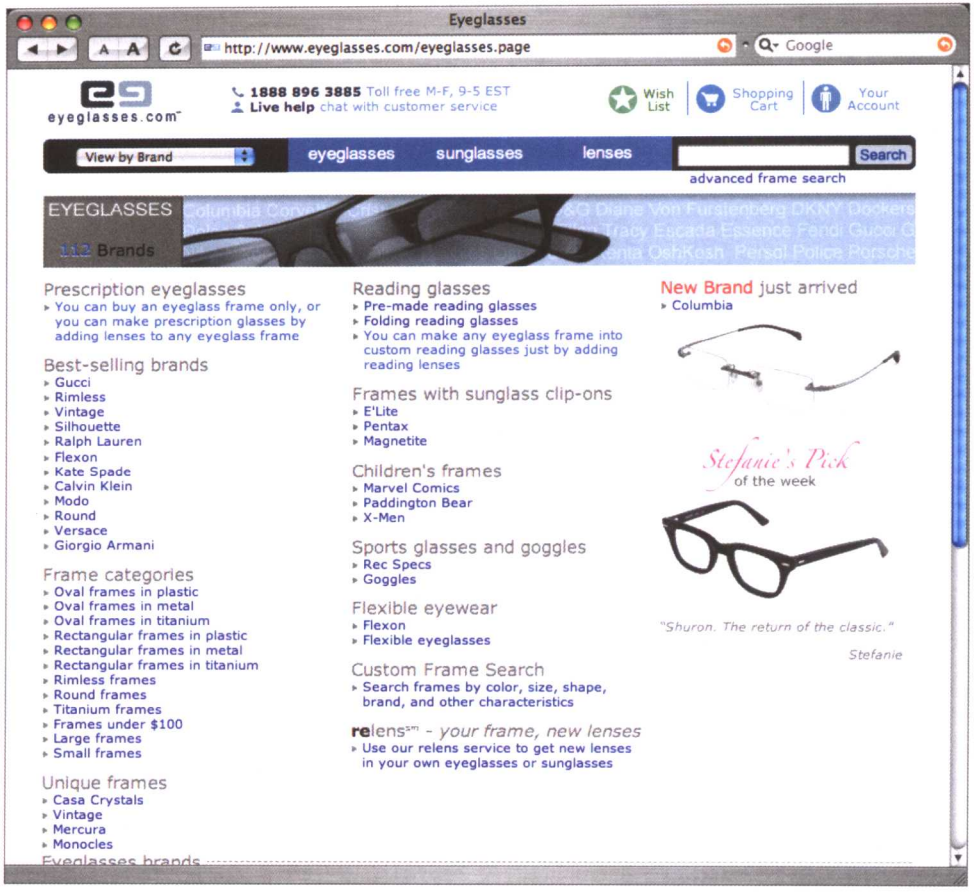


图 1-1　我们以 eyeglasses.com 网站为例来演示设定页面文字大小的常见做法(本图截取自 2004 年 12 月)

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

通过在<body>元素上为页面上的文字设定一个基准大小(本例中是11px)，设计师可以确保在整个页面中(除非在别的对象中被后继的属性值覆盖)的所有文字都将是 11 个像素那么大。使用像素作为单位的好处在于，不管使用哪种浏览器或者设备来查看页

面，文字看起来都几乎一样大。这种一致的，能被事先确定的大小单位正是设计师们所乐意采用的。那些要求页面达到像素级别的完美精确性的人，更是会用像素取代其他一切大小单位。然而，这么做还是有一个小问题。

为什么这样做不是无懈可击的

使用像素来设定文字的大小，能让设计师精确控制各种字体的显示尺寸。但是，目前最流行的Web浏览器(至少在写作本书时)：Windows平台下的Internet Explorer(IE/Win)的广大用户，会面临一个问题。

注意：

我们这里绝非在嘲弄eyeglasses.com网站的设计人员。事实上，这是一个设计得非常好的站点，采用了结构化的XHTML和CSS来实现——这也是我选它为例的原因之一。在这方面它是值得赞赏的。这里只是选取它作为一个参考范例而已。

Web浏览器通常会给用户提供一些方法，让他们可以随意改变页面上文字的大小，以取代设计师原本的设定。这是个很好的功能，对于那些视力较差的用户而言尤为实用！这些用户可以通过浏览器的菜单选择让文字大一号显示(图1-2)。

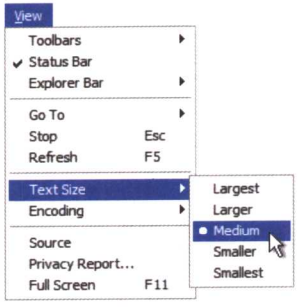


图1-2 IE6/Win的“字体大小”菜单可以让用户随意加大或减小页面文字的大小

听上去很不错吧？但问题在于，当以像素为单位设定了文字的大小后，Windows平台下的Internet Explorer的用户便无法再任意地改变这个大小了。因此，虽然使用像素可以让设计师非常精确地控制尺寸，但同时也禁止了IE/Win用户按照他们的意愿调整文字大小。并且令人不满意的是，很多其他浏览器的用户却可以安全地调整字体大小，不管文字大小如何被设定。

对于视力较差的用户，能够随意调整文字大小应该是他们的基本需求，此外，用户可能还想通过这种方法来让页面内容更易于阅读。而作为设计师，我们很自然地想要充分地控制整个页面的显示效果。这里确实产生了矛盾。但是，如果设计师将这种控