



New
Riders

灵活Web设计

Flexible Web Design

Creating Liquid

and

Elastic Layouts With CSS

(加) Zoe Mickley Gillenwater 著
李静 等译



机械工业出版社
China Machine Press

灵活Web设计

Flexible Web Design

Creating Liquid

and

Elastic Layouts With CSS



(加) Zoe Mickley Gillenwater 著
李静 等译



机械工业出版社
China Machine Press

本书讲述如何应用可变或不固定布局及弹性布局来实现灵活设计，以满足用户的根据自己需求而调整浏览站点的窗口大小的要求。

全书共分为9章，内容包括：理解灵活布局、可变布局和弹性布局存在的挑战、设计灵活布局的方法、准备网页设计、创建可变布局结构、创建弹性布局结构、规范灵活性、设置文字间距、添加背景图像和颜色、创建灵活的图像。

本书适用于网页设计人员、网页设计爱好者。

Authorized translation from the English language edition entitled *Flexible Web Design: Creating Liquid and Elastic Layouts With CSS* (ISBN 978-0-321-55384-3) by Zoe Mickley Gillenwater, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2009 Zoe Mickley Gillenwater.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2009 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。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。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2009-2364

图书在版编目（CIP）数据

灵活Web设计/（加）吉伦沃特（Gillenwater, Z.）著；李静等译。—北京：机械工业出版社，2009.8

（UI设计丛书）

书名原文：Flexible Web Design: Creating Liquid and Elastic Layouts With CSS

ISBN 978-7-111-27294-6

I. 灵… II. ①吉… ②李… 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（2009）第083856号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李东震

北京京北印刷有限公司印刷

2009年9月第1版第1次印刷

186mm × 240mm · 18印张

标准书号：ISBN 978-7-111-27294-6

定价：45.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

前言

现在，拿起你手中正在阅读的这本书，调整它的大小，以便更易于阅读。

非常抱歉，我提出了一个不切实际的要求。但是，我保证，本书所包含的练习都是非常实用的，而非荒谬可笑的。因为我们接下来将要探讨的是Web设计，这不受纸张的物理限制。

尽管Web页面和纸张都可以呈现美观和令人印象深刻的设计，但是这两种方式各有利弊。成功的Web设计需要结合呈现介质的优势，其中的一个优势就是灵活性。这里所指的灵活性是广义上的，适用于各种查看情况。用户控制Web站点的经验将达到一定水平，最终会超过使用其他介质。他们开始选择（无论是自觉的还是自愿的），不考虑浏览Web页面的设备类型、屏幕和窗口的尺寸、文字以及字体的大小，也不考虑是浏览静态图像还是Flash动画。真正精明的用户甚至可以创建起自己的样式表对所浏览站点的视觉效果进行或大或小的调整。有些用户甚至不是“浏览”Web站点，取而代之的是，他们会去聆听或感受这些站点。

从狭义的角度来讲，Web媒介也是灵活的，即Web页面和内容的大小并不固定。默认情况下，可以调整Web页面的宽度和高度，以适应用户不同的文本大小和窗口宽度。这对使用div和CSS构建的Web页面和使用表格构建的Web页面而言都是一样的。

固定宽度的Web页面（由设计者设定了像素的页面）使得Web天生的灵活性无法发挥其作用。创建固定宽度的设计有一定的好处，这将在第1章介绍。但是固定宽度的替代选择会带来很多好处，而且其所适用的站点类型要远比你想象的多。

可变（或不固定）的站点是一种替代选择，该站点会根据用户窗口的大小进行调整，根据用户文本大小进行调整的弹性站点则又是另外一种替代选择。这两种设计类型都利用了Web天生的灵活性，并且，如果利用得当，将可以极大地改善用户体验——用户看到的所有页面都将与固定宽度的站点一样吸引人。

这就是本书的主要内容：为什么灵活设计是一个好东西，以及如何做好它。你或许会想，只要将这一灵活性融入Web中，那么设计灵活的或具有弹性的Web站点就是小菜一碟了。遗憾的是，事实并非如此。

尽管Web中融入了灵活性并且带来了很多好处，但是灵活的和具有弹性的Web站点依然很少。或许是因为这些网站在设计和开发阶段都有一些挑战，大多数Web设计师还不习惯。目前还没有太多直面这些挑战的通用方法，但是本书做到了。

你将学到的和不会学到的

本书旨在告诉你：

译者序

设计是一个极度彰显个性的东西。不同人喜欢的、选择的产品各异。因此，“众口难调”几乎道出了所有设计师的心声。

为了尽量满足用户的使用和审美需求，能应需求而改变的灵活设计就成了理想设计的首选。灵活意味着多变、复杂；而多变和复杂预示着设计难度的提升；难度的提升又表现在设计师要考虑得更为周全。环环紧扣，这无疑给灵活设计蒙上了一层神秘的面纱。实际上，灵活设计并非如此神秘，它是有规律可循的。根据面向的用户群体不同，版式的不同，版面图像的丰富程度，以及文字所占比例的大小，资深的网页设计师们已经总结出了许多实用的设计规律。

人们常说成功的最好办法是站在巨人的肩膀上。因此，想最高效地制作出优秀网页的您，站在优秀设计师的肩膀上不失为上策。本书由经验丰富的Web设计师所撰写，集纳了很多实用的案例技巧，是想快速学习灵活网页设计的人们的不错选择。

本书的特点是，几乎每章都配有与本章内容紧密相关的案例练习，且所有案例均前后相互呼应，形成了一个整体。如果你想快速上手，完整地理解各种灵活布局的适用情况，针对不同情形选择最为实用的技法，那么不必多说，拿起本书，just do it……

本书由李静、贺倩、陈平锋、梁晓琴、李凌燕、贺强、吴启文、卢祖英、苏建忠、马睿倩和翁子扬翻译。由于译者水平有限，疏漏之处在所难免，望读者批评指正。

译者

2009年5月25日

- 每一种类型的灵活布局的好处。
- 如何为特定的内容、设计或用户选择合适的布局类型。
- 哪些视觉元素适合灵活设计，哪些应该避免。
- 如何在灵活性思维的指引下构建并切割图形设计版式。
- 灵活、弹性和混合布局中的 (X) HTML和CSS。
- 如何让背景和 content 在灵活布局容器中工作。

而你不会学到的内容是：固定宽度布局是“魔鬼”，如果你创建了一个固定宽度的布局，那么你就是个可怕的人或是黑客——并不是因为我的编辑让我把溢出的内容给剪切掉了，而只是因为这不是真的。

“固定宽度布局与灵活布局”的争论在Web设计社区里非常激烈，本书不会明确回答何者“更好”。我和很多设计师都认为，固定宽度布局和灵活布局拥有各自的优势，并各自适合于不同类型的网站。我的网站就使用了各种类型的布局，其中包括固定宽度布局。

本书仅为该争论中未过多涉及的一方——灵活布局——提供了更多详细的信息。如果你认为灵活布局适合你的站点（我希望本书能让你更明白灵活布局何时更有效），那么本书会教你如何做好它，并且不会很麻烦。

本书的读者对象

本书面向那些已经有创建网站经验的专业Web设计师们，它不会教你有关 (X) HTML或CSS的知识，要阅读本书，你至少需要在这两方面具备中等水平。我假定你已经有足够的CSS基础并能够使用它创建基本的布局，但你不必是CSS布局方面的专家。当然，你也不必拥有任何使用CSS创建灵活布局的经验。

即便你仍然计划以从事固定宽度的站点的设计为主，你仍可以使用本书中提供的很多设计理念来改善你的设计。由于Web具有用户控制属性（例如，如何让你的页面优雅地与不同的用户文本大小相匹配），固定宽度设计也会用到一定程度的灵活性，这就是CSS设计者应当了解的。

本书对从前那些习惯于基于表格的布局但又致力于创建纯CSS布局的平面设计师和Web设计师尤为有用。它将告诉你如何以CSS的思维方式来思考问题，从而让CSS设计更自然，你也不会再为创建页面时出现不合适的设计版式而感到苦恼了。

示例与练习文件

每一章中的Web设计技巧示例都可以在本书的网站 (www.flexiblewebbook.com) 上找到。你可以使用这些具有代表性的示例文件作为起始点来应用本书所展示的技巧，无须复制书中所有的代码。

绝大多数章节还包含了练习部分，你可以通过这部分内容在真实的页面中逐步练习本章前面讲到的一些技巧。通过整本书，你将经历从设计版式到完成整个页面的这一创建灵活Web站点的全过程。你可以登录www.flexiblewebbook.com下载这些练习的文件，也可以在阅读每个练习中的步骤时即时尝试。

没有哪个Web页面可以保证在所有浏览器中都看上去一样或以同样的方式工作，本书中的示例和练习文件也不例外。这些文件已经在各种主流浏览器（Internet Explorer、Firefox、Safari和Opera）的最新版本中测试通过。一些较老但仍然比较流行的版本（比如Internet Explorer 7和6）也可以正确地应用这些代码，不过比IE6更早的版本（包括适于Mac的Internet Explorer）则无法正确地应用该代码了，因为它们的市场份额太小。在这些浏览器中仍然可以查看到文件中的内容的，但看上去可能就不会那么漂亮了！

本书约定

注意本书使用的以下术语：

- (X) HTML是指HTML和XHTML语言。
- IE 6及更早版本的浏览器是指IE 5.0至IE 6.0的版本。
- IE 5.x指IE 5.0和IE 5.5。

除非特定说明外，本书中提及和所用到的CSS是基于CSS 2.1规范的。HTML 4.01 Strict用于标记示例，但所有示例也会适用于XHTML 1.1。

列出的所有CSS示例应置于一个外部样式表中或(X) HTML文档的head中，而所有(X) HTML示例则应置于文档的body元素中。偶尔，(X) HTML和CSS片段会在同一示例中显示，一个接一个，这是为了简洁的需要。不过，每一个片段都需要放在其对应的位置，从而正确的工作。

同样是为了简洁，一些代码示例中会用到省略号(...)以表明被删除或重复的部分。例如，下面这一段代码示例中的省略号就表明ul中的li元素未全部显示出来：

```
<ul>
  <li>Apples</li>
  <li>Bananas</li>
  ...
  <li>Watermelon</li>
</ul>
```

有些代码示例则会包含标有蓝色的字符或整行的代码。高亮部分表示自上次查看后，代码段的该部分内容有增加或被更改。

致 谢

感谢我的组稿编辑Wendy Sharp，他是第一个与我接洽的人。他帮我充实了本书的构思，在整个过程中都为我提供了指导，对另外一位作者也非常耐心。

感谢我的编辑Wendy Katz，她为本书提供了让人难以置信的、非常细致且深思熟虑的修订和建议，让这本书看上去更棒。她的支持和指导非常宝贵，能与她如此紧密地合作是一大乐事。

感谢我的技术编辑Sheri German，她不辞辛苦地核对所有代码示例和页面示例，并帮我查找疑难问题的答案。我特别要感谢的是她的热情。

感谢Peachpit/New Riders的生产、销售以及其他部门的全体员工，虽然我未能直接与他们接触，但正是有了他们，这本书才得以出版。

感谢我的文稿代理商David Fugate，他给予了我指导和支持。

感谢我的朋友和家人，他们为本书的示例站点提供了宠物照片：Karen Brock与她的小狗Shelby，Dean Gillenwater与他的小猫Alley和小狗Dutch，Sarah Grover与她的小猫Nomar，以及Laura Sandt与她的小狗Bailey。

感谢CSS专家Stephanie Sullivan和John Gallant，是他们在数年前给予了我撰写这本书的动力。从他们身上我学会了很多，很幸运能得到他们的支持和帮助。

感谢所有启发和教导我的CSS老师和Web设计师，他们是：Cameron Adams、Holly Bergevin、Doug Bowman、Andy Budd、Dan Cederholm、Ingo Chao、Andy Clarke、Bruno Fassino、Roger Johansson、Eric Meyer、Cameron Moll、Georg Sortun和Russ Weakley等。

感谢来自北卡罗莱纳州大学（UNC）公路安全研究中心的所有同事。在过去的数个月里，他们容忍了我的疲倦、心烦意乱和压抑。在此，尤其要感谢我的督导兼朋友——Katy Jones。

特别要感谢的是我的丈夫Cary和我们神秘的小宝宝，他们一直陪伴着我（噢，我的小宝还没出生）。我爱你们。Cary，你在面对一个压抑的作者和怀孕的女人时所表现出来的耐心让我倍感惭愧。非常感谢你的支持和鼓励，你还为我准备了一日三餐。我欠你好几顿晚餐，还有蛋糕。

目 录

译者序	
前言	
致谢	
第1章 理解灵活布局	1
1.1 布局类型	2
1.1.1 固定宽度：固定像素	2
1.1.2 可变布局或不固定布局： 根据视口调整	4
1.1.3 弹性：与文本大小相适	7
1.1.4 混合布局	10
1.2 可变布局和弹性布局带来的挑战	12
1.2.1 摆脱网格束缚：进行设计 非舒适区域	12
1.2.2 告别像素的完美性	13
1.2.3 让设计看上去很大或是很小	14
1.2.4 处理文本溢出的问题	14
1.2.5 处理水平滚动条的问题	15
1.2.6 在弹性框中适配固定宽度的内容	15
1.2.7 更长的设计时间和测试时间	16
1.2.8 浏览器缺陷	16
1.3 为页面选择合适的布局类型	16
1.3.1 固定宽度布局	17
1.3.2 可变布局	18
1.3.3 弹性布局	19
第2章 设计灵活布局的方法	20
2.1 灵活布局的设计原理	21
2.1.1 避免：包含文本的内容采用 固定高度	21
2.1.2 避免：不规则形状	24
2.1.3 避免：与不能扩展的图像 相匹配的文字	26
2.1.4 避免：固定宽度和全宽度内容	33
2.1.5 规划：超出原始尺寸的图像	40
2.1.6 规划：设计位于最大宽度 之外的白色区域	41
2.1.7 规划：折成多行的并排放置的项	43
2.2 设计前和设计后：我们的非 兼容设计和固定的设计	48
第3章 准备网页设计	53
3.1 创建图像版式	54
3.1.1 设置画布尺寸	54
3.1.2 使用图层和页面来管理版式	55
3.1.3 非破坏性编辑技术	58
3.2 针对灵活设计对图形进行切片 处理	60
3.2.1 为切片创建多个单独的文件	61
3.2.2 切分方框	64
3.3 站点创建练习：为构建站点准备 Beechwood Animal Shelter版式	65
3.3.1 在版式中导航	65
3.3.2 为页眉创建切片	66
3.3.3 为主页创建切片	68

3.3.4	为步骤页创建切片	71	5.3.1	准备页面	139
3.3.5	导出图像	72	5.3.2	添加基础样式	141
第4章	创建可变布局结构	74	5.3.3	创建栏	142
4.1	为CSS布局准备页面	75	第6章	规范灵活性	147
4.1.1	通用页面的创建步骤	75	6.1	创建混合布局	148
4.1.2	为本章实例准备页面	76	6.1.1	固定侧边栏搭配可变居中栏位	148
4.2	使用浮动创建可变栏	79	6.1.2	带弹性居中栏的固定侧边栏	153
4.2.1	使用浮动创建栏位并且匹配侧边缘	80	6.2	添加最小和最大宽度	165
4.2.2	通过浮动所有对象创建栏	91	6.2.1	保护你的用户、设计和内容	165
4.2.3	使用浮动和负边缘创建栏位	95	6.2.2	选择min-width和max-width值	166
4.3	禁止布局横跨整个视口	100	6.2.3	使用min-width和max-width属性	168
4.3.1	更改栏位宽度	101	6.2.4	IE 6和更早版本的浏览器 中的不兼容	177
4.3.2	指定侧边缘	103	6.3	限制灵活性的其他方式	178
4.3.3	为包装器指定百分比宽度	108	6.3.1	使用overflow属性	178
4.4	站点创建练习：创建Shelter的 内页布局结构	109	6.3.2	让栏位下移	181
4.4.1	准备页面	110	6.4	站点创建练习：限制Shelter的 页面的灵活性	182
4.4.2	添加基本样式	111	6.4.1	修改Programs页面	183
4.4.3	创建双栏布局	112	6.4.2	修改主页面	184
4.4.4	减少双栏的宽度	115	第7章	设置文字间距	187
第5章	创建弹性布局结构	117	7.1	匹配度量单位	188
5.1	将尺寸换成em	118	7.1.1	可变布局间隔	188
5.1.1	em的定义	118	7.1.2	弹性布局间隔	197
5.1.2	嵌套em度量	118	7.2	混合度量单位	198
5.1.3	有选择性地采用em度量	119	7.2.1	为宽度较窄的div添加边缘和填充	199
5.2	使用浮动创建弹性栏	121	7.2.2	为内容而非div添加边缘和填充	200
5.2.1	复习：本章实例	122	7.2.3	添加缓冲div	202
5.2.2	使用浮动创建栏位与匹配侧边缘	122	7.3	站点创建练习：为主页和 内页添加间隔	205
5.2.3	通过浮动所有元素来创建栏	135	7.3.1	为页眉添加间隔	205
5.3	站点创建练习：创建Shelter的 主页布局结构	139			

7.3.2	为页脚添加间隔	210	8.3.1	调整页眉的背景和颜色	242
7.3.3	处理边缘塌陷	211	8.3.2	调整Programs页面背景	245
7.3.4	为Programs页面内容添加间隔	212	8.3.3	调整主页背景	250
7.3.5	为主页内容添加间隔	215	第9章 创建灵活的图像	252	
第8章 添加背景图像和颜色	221	9.1 动态更改图像的屏幕区域	253		
8.1 混合背景图像	222	9.1.1 与布局同时缩放的前景图像	253		
8.1.1 创建渐变	222	9.1.2 隐藏和显示部分图像	255		
8.1.2 创建圆角	225	9.1.3 创建变化的合成图像	260		
8.1.3 创建曲线边缘	229	9.2 创建灵活的图像组	262		
8.2 创建等高栏	232	9.2.1 趣味缩览图列表	263		
8.2.1 可变-固定宽度混合 布局的伪列布局	233	9.2.2 缩览图库	267		
8.2.2 可变布局的伪列布局	237	9.3 站点创建练习：为主页添加 灵活的图像	272		
8.2.3 弹性布局的伪列布局	241	9.3.1 添加可变裁切的横幅广告图像	272		
8.3 站点创建练习：为主页和 内页添加背景	242	9.3.2 创建趣味缩览图列表	273		

第 1 章

理解灵活布局

对不同的人而言，“灵活布局 (flexible layout)”这一术语的意义也各不相同。因此，在继续设计和创建网页前请确保你打开的页面与本书所指的是同一页面。本章将学习如何定义前言中讲述的各种主要布局类型（固定宽度、可变宽度、弹性宽度以及混合宽度）的特性。我们将关注每种类型的优缺点，以便你可以针对特定的站点选择适宜的类型。

1.1 布局类型

我们可以根据网页布局的宽度设置将其分为3种类型：固定宽度、可变（或不固定）宽度和弹性宽度。还可以通过混合不同的度量单位将这些布局合并为混合布局，设计的每一个分栏都可以使用不同的单位。这4种布局类型中的任何一种都可以使用任意数量的分栏或者美化主题，布局类型直接地决定了浏览设备如何在用户面前展现布局的宽度。

1.1.1 固定宽度：固定像素

固定宽度布局（fixed-width layout）是以前最常见和最常采用的设计，之所以说是以前的，是因为本书将讲解几种替代此布局的设计。固定宽度设计的整体布局宽度以像素为单位来设置，具体值由设计者决定。通常，设计者基于最常见的屏幕分辨率（例如，800×600和1024×768）来选择宽度。

固定宽度设计是固定不变的，它们的尺寸不会随着用户的设置而变化，如图1-1所示。这样一来，即便是存在不同的用户设置，你也可以创建一个包含丰富图像的网页，并且能让它看上去紧凑一致。如果你对站点的目标用户进行了调查，那么你就能设计出符合大多数用户浏览器窗口的布局，并且可以确保将文本行之间的宽度设置为一个便于阅读的理想值。

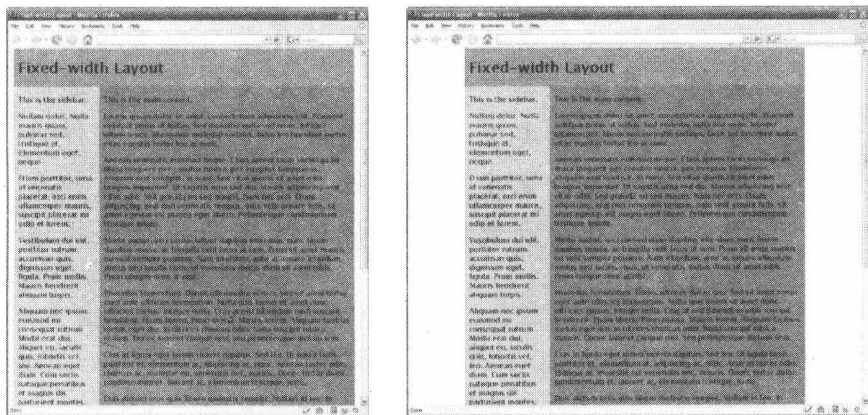


图1-1 同一固定宽度的布局在不同大小的浏览器窗口中显示的结果

固定宽度设计并非弊病

本章花了大量篇幅对固定宽度设计进行阐述，并极力“颂扬”了可变设计和弹性设计。这仅仅是因为这就是本书的宗旨：推广和教授灵活布局技术以及它们必须提供要使用这些技术所必需的条件。要强调的是，我并不认为固定宽度设计是“错误的”，它们在特定的情形下是非常适合的，我们将在本章最后对此进行详述。

1. 屏幕分辨率并不等同于浏览器窗口大小

固定宽度布局的最大问题在于，它们本质上取决于你估计何种宽度会最适合大多数用户。即便是你的网页统计软件可以告知每个用户屏幕分辨率——即便是你正在制作一个内联网并且确定只会使用单一的分辨率——屏幕的分辨率也并非总是能够与浏览器窗口相匹配。有一些人不会使用最大化的浏览器窗口进行浏览（不能否认，这是很少的一部分人，但是随着显示器和分辨率大小的增加，这类人的数量也在增加）。另外，还有一些人会使用占用了可用宽度数百像素的浏览器工具条。

固定宽度设计带来的后果往往是，相对于浏览者的窗口而言，某些浏览者看到的设计不是过宽（迫使他们不停地拖动水平滚动条）就是过窄（使得布局的一侧或者两侧留下大量空白）。根据我对用户所做的测试，很多人几乎都对浏览器中“被浪费的空间”或者费劲地拖动水平滚动条表示苦恼不堪！

对浏览器窗口大小的调查

2006年10月，Thomas Baekdal在其网站www.baekdal.com上发表了一篇非常有趣的报告，《Actual Browser Sizes》。他搜集了5个网站3个月内的有关屏幕分辨率和浏览器窗口尺寸的数据。报告显示，尽管绝大多数被测试用户都将他们的浏览器设置为最大值（或者至少是接近最大值），但是还是有很大一部分人并未这样做。例如，使用了1024×768分辨率的用户（迄今为止这是最为普遍的）80%的机率会将其浏览器设置为最大值。Thomas的结论是，“为了支持95%的访问者，你需要一个符合776像素×424像素的设计”——尽管他发现只有5%的用户使用800×600的屏幕。要想查看该报告全文，可访问以下网址：<http://baekdal.com/reports/actual-browser-sizes>。

2. 并非所有人都使用16像素的文本

如果你知道所处理文本的大小，那么你可以选择一个固定的宽度来优化每行的字符数或者行长以提高易读性。印刷设计师们一直都在做这一工作；Robert Bringhurst的名作《印刷样式元素》（The Elements of Typographic Style）推荐的行长为45至75个字符，这是基于数年对印刷文本易读性的调查研究所得出的结论。根据对屏幕显示文本易读性的最新研究表明，75到100个字符之内，行长越长，阅读速度越快（尽管很多受测试用户表示他们更喜欢短一些的文本）。

不过，在网络中，我们无法知道用户的文本大小。浏览器默认值为16像素，绝大多数用户会对其文本使用该默认值。然而，有些用户还是会更改该默认值，或者将用户样式表设置成格式文本以便更易阅读。甚至，如果文本的某一页读起来很费劲（即便是像素可调的文本，Internet Explorer 6以及更早版本的浏览器中的文本除外），那些对该文本使用默认值的用户

也会增加每页文本的大小。因此，如果你为16像素大小的文本进行行长优化，你或许是在为绝大多数访问者进行易读性优化工作，但并不是为所有的访问者。不要误解我的意思：为绝大多数人设计是一件好事。只是不要错误地认为“绝大多数”等同于“所有”。

另一个问题是，这些对行长的研究并未考虑到会促使特定群体更倾向于更短或是更长行长的各种不利条件。尽管我认同设计师的工作多半是围绕用户需求并以最有利于用户的方式而展开（因为用户自己并不知道什么是最好的），但是有时候还是需要让用户自己而不是我们，去更好地了解什么才能最佳地满足他们的需求。至少，我们可以根据自己认为的能够帮助绝大多数用户的考虑来优化设计，不过要告知用户为了更好地满足其需求而需适应我们的设计的可能性——这是印刷以及更为固定的媒介无法获得的一大优势。

1.1.2 可变布局或不固定布局：根据视口调整

注意 视口（viewport）是描述用户设备中页面可见区域的通用术语。相比于窗口（window），人们更愿意使用视口，因为毕竟并不是所有设备都使用窗口（例如，手机）。

可变布局（liquid layout），又名不固定布局（fluid layout），会基于用户视口的大小而更改宽度。使用CSS创建的可变布局可能设定了或者未设定宽度。如果不设定宽度，则无论用户视口有多大，它们都会填满该视口，如图1-2所示。

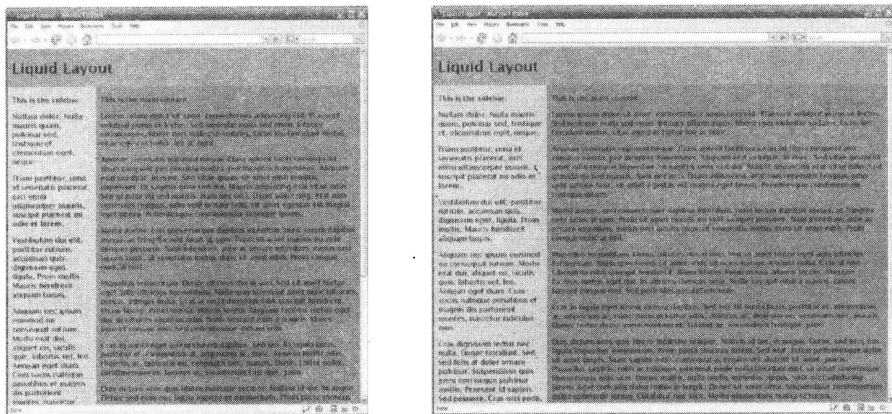


图1-2 同样的可变布局会基于浏览器窗口大小而调整宽度

如果设计师对可变布局设定了宽度，那么该宽度将会以百分比计量，而非像素。百分比是指它所占视口的比例。

1. 发挥屏幕区域的优势

当改变可变布局大小时，该布局中的所有内容（通常还包括背景图像）不得不在页面上进行移动以填满页面空间。只要内容可以折行，这一灵活性便可以防止水平滚动条出现，并

充分利用用户设备上的可用屏幕区域（图1-3）。

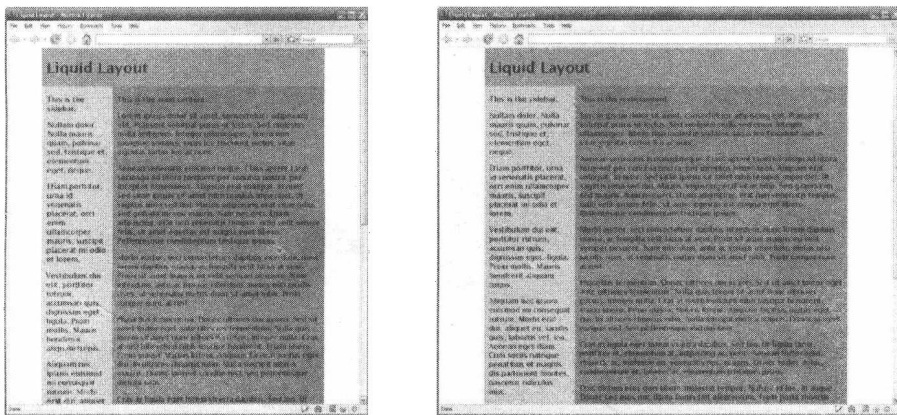


图1-3 可变布局往往会根据浏览器窗口的大小来调整尺寸——只有在你设定了一个较小的百分比宽度时，它才不会占满浏览器窗口的整个宽度

一旦内容无法进行折行，鉴于图像和其他内容的固定宽度，水平滚动条最终还是会出现，不过这只会最窄屏幕或者拥有非常大的固定宽度内容的站点上才会出现。使用可变布局取代固定宽度的布局，用户错过那些被水平滚动条隐藏的重要内容的可能性变小。另外，对那些拥有大型视口的用户而言，更多的内容将一次性呈现在其眼前，减少了他们拖动滚动条的操作。

2. 尊重用户的偏好

使用可变布局，你可以不用去猜测应该为用户设计什么样的作品，相反你可以让用户自行选择适合他们需求的最佳页面宽度。你的主页面上不再需要类似“最佳浏览器分辨率为1024×768”的声明。即便是用户可以更改分辨率以满足此要求，他也不大可能这样做来迎合你的网站。用户设置分辨率有一个原因：由于设备条件约束而别无选择，或者该分辨率是他所想要的或认为是最有用的（例如，一些用户一直使用800×600的分辨率，原因在于这样可以让所有的东西看上去更大，便于阅读）。有了可变布局，你无须再对此存有担忧。可变布局适用的查看情形更为广泛，它十分尊重用户查看网页的偏好。

3. 提高可读性

注意 事实上，有关可变布局的争论之一在于，文本在大的浏览窗口中显示的行长过长，降低了可读性。当可变布局不适用时或者在特定的用户环境下时，便会出现这个问题。我们将在本章后面的内容中细谈该问题。

水平滚动条是可读性不共戴天的敌人。频繁地左右滚动来阅读数行文本，这毕竟不是让人愉悦的阅读经历。有了可变布局，水平滚动条几乎不再可能出现，因为用户可以对窗口大

小进行调整，从而生成多行文本来找到最舒适的状态以便阅读和理解。对行长的偏好会因年龄、身体状况和浏览设备的不同而有所不同，因此，相对于设置一个固定的宽度而言，保持宽度的可调节性可以让更多的人高效地阅读你的内容。

便携式设备情形

虽然可变布局增加了尺寸的范围，使得网站看上去舒服且适用，你仍需要为手机和PDA等便携式设备设置一个独立的样式表。要让同一设计在200像素宽度和在1200像素宽度中均适用是不可能的（除非你打算显示旧学院派的、纯文本的外观）。

4. 增加残障人士对内容的可访问性

行长对身体残疾的用户顺利阅读和理解文本显得更加重要。如果某一用户因视觉有损伤而需要将文本设置得很大，那么他会倾向于使用一个窗口非常大的浏览器，这样就可以在一行里显示更多单词。如果在一个狭窄的、固定宽度的布局里，一篇大的文本或许只能够允许一行仅显示2个或3个单词，这就使得阅读起来很费劲而导致垂直滚动条过长。

其他类型的视觉缺陷者或许会用到屏幕放大软件，其每次只会向用户显示窗口被高度放大的一小部分。使用屏幕放大器的人或许倾向于将窗口设置得非常窄，这样文本每一行整体宽度就会与其所在的被放大的屏幕的狭小区域相符，而无需来回滚动水平滚动条查看放大的内容了。

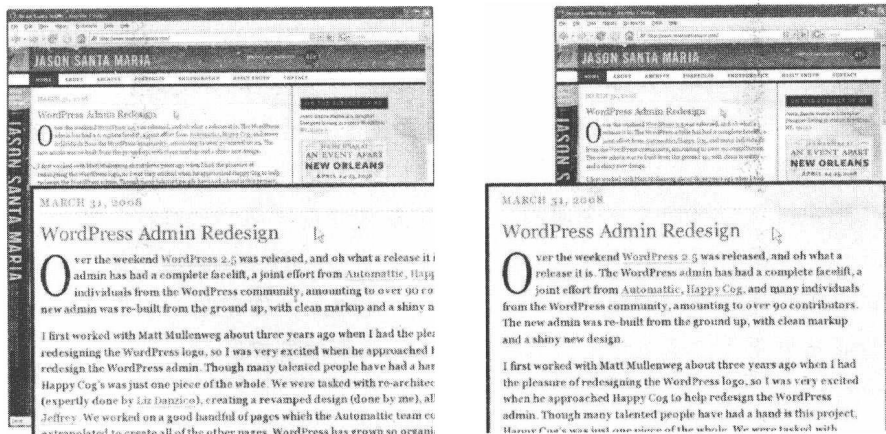


图1-4 如果浏览窗口很宽，那么屏幕放大软件或许无法在放大的布局内显示文本的整个行长，因此用户不得来回拖动以便阅读文本（左图）。如果浏览器窗口狭窄，文本的宽度也会相应地调整，整个文本行则可以与放大的布局（右图）相符，从而让阅读变得更为快捷

行长对理解也有影响。例如，很多遭遇阅读障碍的人会发现，短的行长更有助于阅读和理解。