

重视大脑的学习指南

第二版
全面更新，涵盖HTML5

Head First HTML 与 CSS

只用一章开启
你的Web职业
生涯



创建基于标准的Web
页面的初学者指南



当心常见的
HTML和CSS
陷阱和问题



发现为什么你的朋友
对样式的理解
可能是错的

100个谜题和练习
帮你记住



避免验证错误的
尴尬

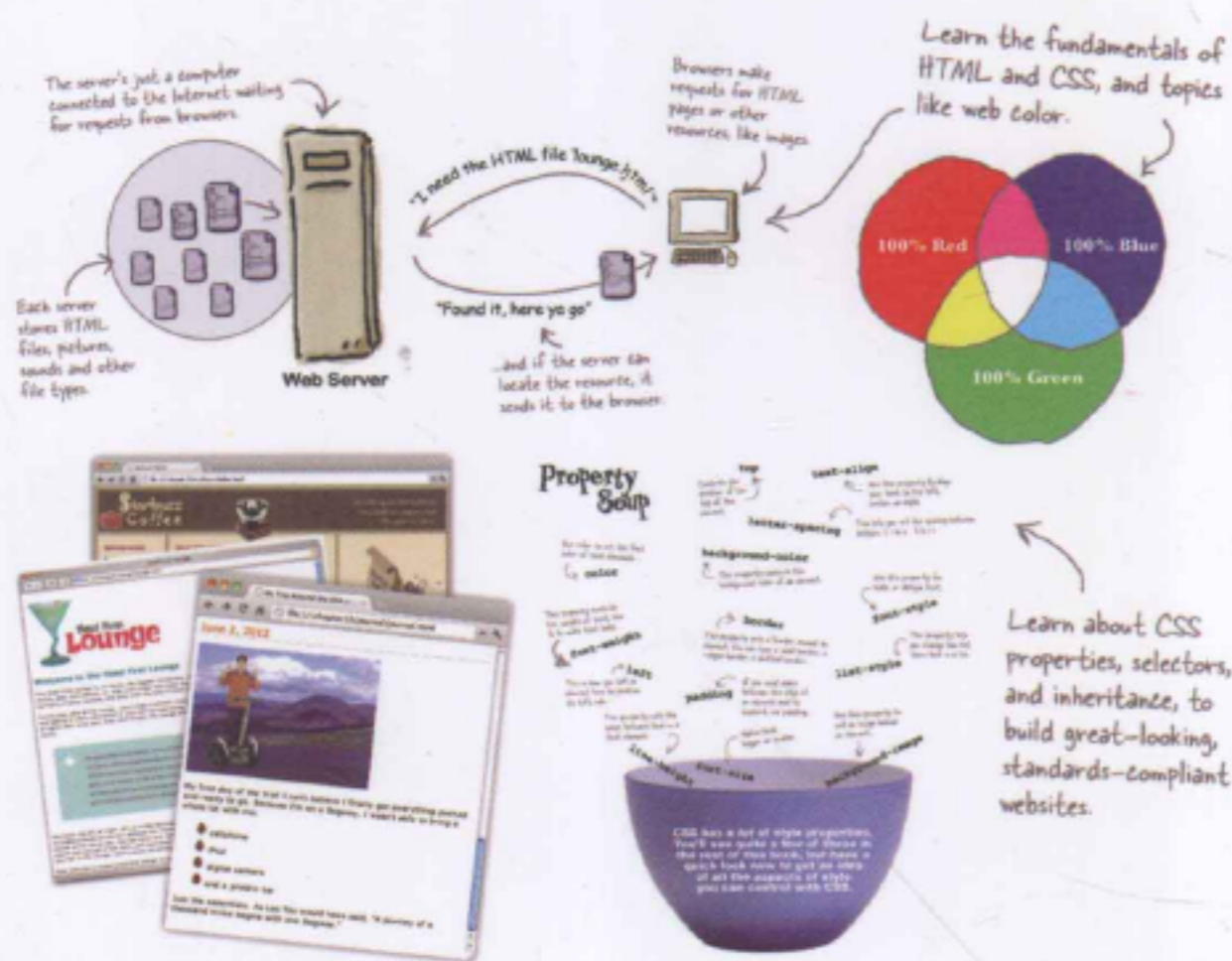


O'REILLY® 中国电力出版社

Elisabeth Robson & Eric Freeman 著
徐阳 丁小峰 等译

从这本书能学到什么?

是不是已经厌倦了那些深奥的HTML书?你可能在抱怨,只有成为专家之后才能读懂那些书。那么,找一本新修订的《Head First HTML与CSS(第二版)》吧,来真正学习HTML。你可能希望学会HTML和CSS来创建你想要的Web页面,从而能与朋友、家人、粉丝和狂热的顾客更有效地交流。你还希望使用最新的HTML5标准,能够保证随时间维护和扩展你的Web页面,使它们在所有浏览器和移动设备中都能正常工作。



为什么这本书如此与众不同?

在这本书中,你会学到创建Web页面的秘密,最重要的是,你会用一种不至于让你昏昏欲睡的方式来学习。如果你读过Head First书,应该知道这会是一本怎样的书:它会采用一种专门为你的大脑而设计的丰富格式娓娓道来。通过使用认知科学和学习理论的最新研究成果,这本书会向你的大脑输入HTML和CSS,并让它牢牢记住。



O'REILLY®

oreilly.com.cn
headfirstlabs.com

O'Reilly Media, Inc. 授权中国电力出版社出版

“《Head First HTML与CSS(第二版)》运用新颖的手法,介绍了Web页面标记和表现方面的前沿方法。它准确地把握了读者的困惑,并及时解决。丰富的图片和渐进的过程模拟了学习这个技术的最佳途径:你可以做一点改动,然后在浏览器上查看,了解每一个新特性的具体含义。”

——Danny Goodman,
《Dynamic HTML: The Definitive Guide》的作者

“这本书介绍的都是专业人士才了解的信息,不过采用了一种循循善诱、幽默诙谐的方式来讲解,绝对不会让你觉得这些东西学不会或者让你无所适从。”

——Christopher Schmitt,
《CSS Cookbook》与
《Professional CSS》的作者

ISBN 978-7-5123-4477-8



定价: 98.00元

Head First HTML与CSS

(第二版)

有没有一本HTML书假设你根本不知道元素、属性、验证、选择器和伪类为何物，一切都从头开始介绍？可能只是异想天开吧……



Elisabeth Robson
Eric Freeman 著
徐阳 丁小峰 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

Head First HTML与CSS: 第2版 / (美) 罗布森 (Robson, E.), (美) 弗里曼 (Freeman, E.) 著; 徐阳等译. —北京: 中国电力出版社, 2013.9

书名原文: Head First HTML and CSS, Second Edition

ISBN 978-7-5123-4477-8

I. ①H… II. ①罗… ②弗… ③徐… III. ①超文本标记语言—程序设计 ②网页制作工具 IV. ①TP312 ②TP393.092

中国版本图书馆CIP数据核字 (2013) 第108333号

北京市版权局著作权合同登记

图字: 01-2013-3086号

©2012 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2013. Authorized translation of the English edition, 2012 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2012。

简体中文版由中国电力出版社出版, 2013。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。


书 名/	Head First HTML与CSS (第二版)
书 号/	ISBN 978-7-5123-4477-8
责任编辑/	刘焯
封面设计/	Karen Montgomery, 张健
出版发行/	中国电力出版社
地 址/	北京市东城区北京站西街19号 (邮政编码100005)
印 刷/	航远印刷有限公司
开 本/	850毫米×980毫米 16开本 47.625印张 1021千字
版 次/	2013年9月第1版 2013年9月北京第1次印刷
印 数/	0001—3000册
定 价/	98.00元 (册)

敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪
本书如有印装质量问题, 我社发行部负责退换

版权专有 翻印必究

浏览器战争是什么？你
会在第6章找到答案。

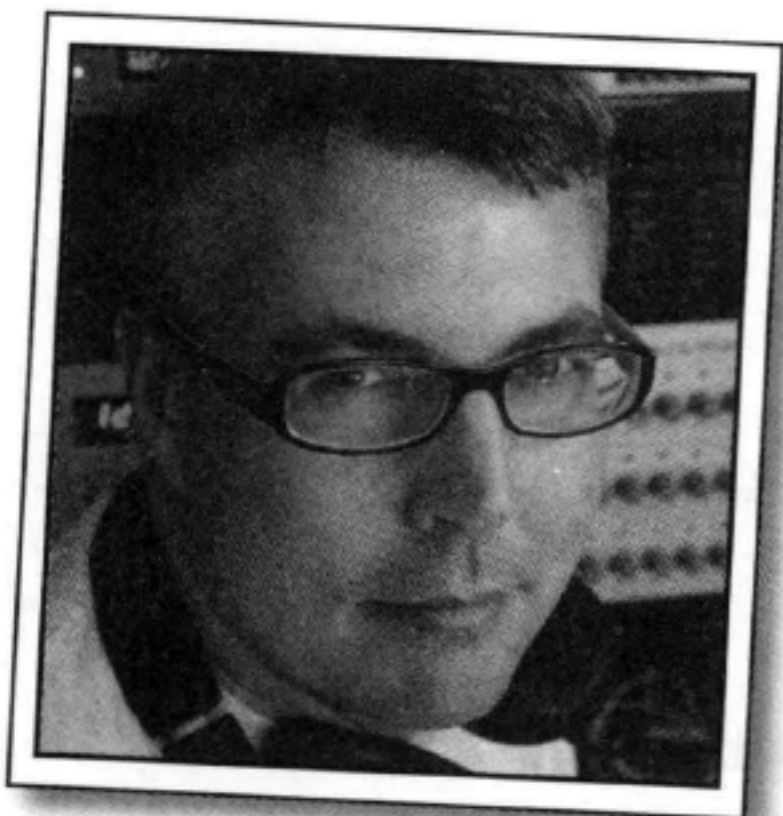


致W3C，感谢你们结束了浏览器战争，另外感谢你们的睿智，
是你们提倡结构（HTML）与表现（CSS）分离……

还要感谢你们让HTML和CSS这么复杂，正因如此，人们才需要
这样一本书来学习。

作者

《Head First HTML 与 CSS》的作者



←
Eric Freeman



↙ Elisabeth Robson

按照“Head First”系列合作者Kathy Sierra的说法，Eric是“少有的奇才之一，不仅语言流畅，实践经验丰富，在很多领域都表现非凡，他是嬉皮士高手、副总裁、工程师，而且是名副其实的智多星。”

在专业领域，Eric最近刚刚离开任职近十年的一家媒体公司，他在迪斯尼公司担任Disney Online & Disney.com的CTO。Eric现在把时间全部投入到WickedlySmart，这是他与Elisabeth共同创建的一家公司。

经过培训，Eric已经成为一位计算机科学家，在耶鲁大学攻读博士学位期间曾与业界杰出人物David Gelernter同窗。他的论文被认为对寻找桌面隐喻*的替代品有着深远影响，这也是活动流的首个实现（活动流是他与Gelernter博士提出的一个概念）。

在闲暇时间，Eric对音乐深深着迷；在iPhone App Store上可以找到Eric最近与音乐“先锋”Steve Roach合作的一个项目，名为Immersion Station。

Eric与妻子和小女儿居住在华盛顿州双桥岛。他女儿是Eric工作室的常客，她特别喜欢打开他的合成器和音效开关。Eric对儿童教育和营养也很关注，总在想方设法做出改善。

可以给Eric写邮件（eric@wickedlysmart.com），或者访问他的网站（http://ericfreeman.com）。

* 桌面隐喻是在用户界面中用人们熟悉的桌面上的物品来清楚地表现计算机可处理的能力。

Elisabeth是一位软件工程师、作家和培训师。从她作为耶鲁大学学生之日起就一直热衷于技术，她在耶鲁大学获得了计算机科学硕士学位，并设计了一个并发的可视化编程语言和软件体系结构。

Elisabeth从早期就一直从事Internet的工作，她合作创建了颇有声誉的网站：The Ada Project，这是最早帮助计算机科学领域的女性在线找工作和寻求指导信息的网站之一。

她目前是WickedlySmart的合作创始人，这是一个关注Web技术的在线教育项目，在这里她完成了有关的图书、文章、视频等。在此之前，作为O'Reilly Media的特殊项目主任（Director of Special Projects），Elisabeth曾经在各种技术专题发布过个人研讨文档和在线课程，这使她对创建学习体验来帮助人们理解技术越来越充满热情。在O'Reilly工作之前，Elisabeth主要在迪斯尼公司播洒她的仙女魔法粉，在那里，她带领数字媒体研发力量开展研究工作。

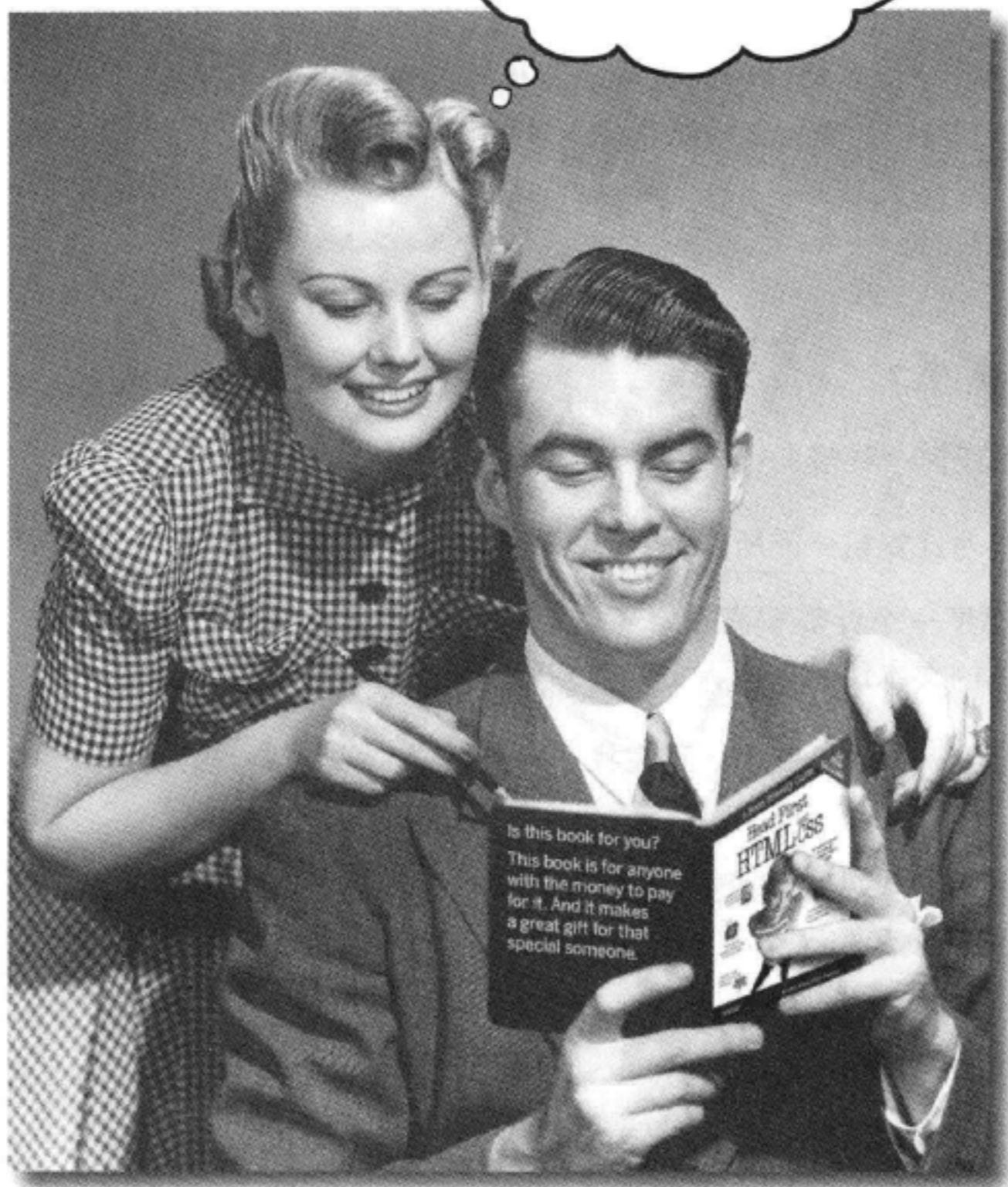
如果没有坐在计算机前，你就会发现Elisabeth总是带着她的相机在户外踏青、骑单车或者划皮艇，也可能在烹调素食大餐。

可以给她发Email（beth@wickedlysmart.com）或者访问她的博客（http://elisabethrobson.com）。

如何使用这本书

引子

真是无法相信，这些东西也能放在一本HTML书里！



有个问题真是听得我们耳朵都磨出茧了：“你们到底为什么要把这样一些东西放在一本HTML书里呢？”现在就来回答这个问题。

谁适合看这本书？

如果对下面的所有问题都能肯定地回答“是”：

- ① 你有一台安装了Web浏览器和文本编辑器的计算机吗？
- ② 你是不是想学习、理解、记住并且用最好的技术和最新的标准来创建Web应用？
- ③ 你是不是更喜欢一种轻松的氛围，就像在晚餐餐桌上交谈一样，而不愿意被动地听枯燥乏味的技术报告？

← 如果你的计算机是最近十年内的产品，答案就是“是”。

这正是你想要的书。

谁可能不适合看这本书？

如果满足下面任何一种情况：

- ① 你是不是对计算机一无所知？
不要求你是一个计算机高手，不过起码应该了解文件夹和文件、简单的文本编辑应用，另外要知道如何使用Web浏览器。
- ② 你是不是在开发Web应用，正在找一本参考书？
- ③ 你是不是对新鲜事物都畏首畏尾？只喜欢简单的样式，而不敢尝试把条纹和格子混在一起看看？你是不是觉得，如果加入了拟人化的HTML标记，这样一本书肯定不是一本正经八百的技术书？

那么，这本书将不适合你。



[来自市场的声音：只要有信用卡，都可以拥有这本书。]

我们知道你在想什么。

“这算一本正经八百的书吗？”

“这些图做什么用？”

“我真能这样学吗？”

我们也知道你的大脑正在想什么。

你的大脑总是渴求一些新奇的东西。它一直在搜寻、审视、期待着不寻常的事情发生。大脑的构造就是如此，正是这一点才让我们不至于墨守成规，能够与时俱进。

如今，真正面对老虎、被老虎吃掉的可能性很小，不过你的大脑还是在时刻提防着。你只是不知道而已。

我们每天都会遇到许多按部就班的事情，这些事情很普通，对于这样一些例行的事情或者平常的东西，你的大脑又是怎么处理的呢？它的做法很简单，就是不让这些平常的东西妨碍大脑真正的工作。那么什么是大脑真正的工作呢？这就是记住那些确实重要的事情。它不会费心地去记乏味的东西，就好像大脑里有一个筛子，这个筛子会筛掉“显然不重要”的东西，如果遇到的事情枯燥乏味，这些东西就无法通过这个筛子。

那么你的大脑怎么知道到底哪些东西重要呢？打个比方，假如你某一天外出旅行，突然一只大老虎跳到你面前，此时此刻，你的大脑还有身体会做何反应？

神经元会“点火”，情绪爆发，释放出一些化学物质。

好了，这样你的大脑就会知道……

这肯定很重要！可不能忘记了！

不过，假如你正待在家里或者坐在图书馆里，这里很安全、很舒适，肯定没有老虎。你正在刻苦学习，准备应付考试。也可能想学一些比较难的技术，你的老板认为掌握这种技术需要一周时间，最多不超过十天。

这就存在一个问题。你的大脑很想给你帮忙。它会努力地把这些显然不太重要的内容赶走，保证这些东西不去侵占本不算充足的脑力资源。这些资源最好还是用来记住那些确实重要的事情，比如大老虎，再比如遭遇火灾等。就像你的大脑会让你记住绝对不能再穿着短裤去滑雪。

没有一种简单的办法来告诉大脑：“嘿，大脑，真是谢谢你了，不过不管这本书多没意思，也不管现在我对它多么无动于衷，但我确实希望你能把这些东西记下来。”

你的大脑想着，这真的很重要。



噢，又是700多页干巴巴的文字，枯燥又乏味。

你的大脑认为，这些根本不值得去记。

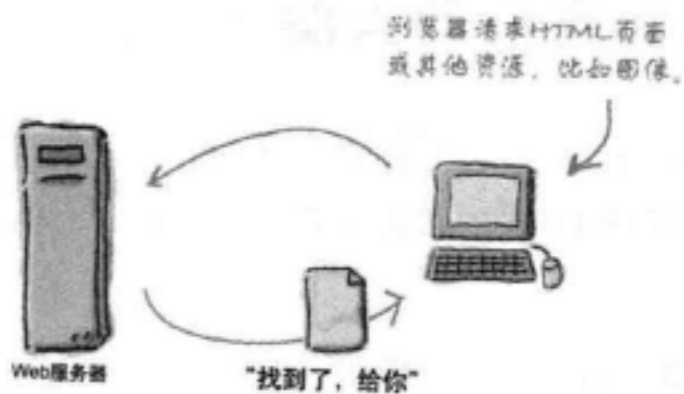


我们认为“Head First”的读者就是要学习的人。

那么，怎么学习呢？首先必须获得知识，然后保证自己确实不会忘记。这可不是填鸭式的硬塞。根据认知科学、神经生物学和教育心理学的最新研究结果，学习的途径相当丰富，绝非只是通过书本上的文字。我们很清楚怎么让你的大脑兴奋起来。

下面是一些Head First学习原则：

看得到。与单纯的文字相比，图片更能让人记得住，通过图片，学习效率会更高（对于记忆和传递型的学习，甚至能有多达89%的效率提升）。而且图片更能让人看懂。以往总是把图片放在一页的最下面，甚至放在另外的一页上，与此不同，如果把文字放在与之相关的图片内部，或者在图片的周围写上相关文字，学习者的能力就能得到多至两倍的提高，从而能更好地解决有关问题。



采用一种针对个人的交谈式风格。最新的研究表明，如果学习过程中采用一种第一人称的交谈方式直接向读者讲述有关内容，而不是用一种干巴巴的语调介绍，则学生在学习之后的考试中成绩会提高40%。正

确的做法是讲故事，而不是做报告。要用通俗的语言。另外不要太严肃。如果你面对着这样两个人，一个是你在餐会上结识的很有意思的朋友，另一个人学究气十足，喋喋不休地对你说教，那么在这两个人中，你会更注意哪一个呢？

忘了你的<body>元素确实很糟糕。



关于页面的信息都放在head元素里。

让学习的人想得更深。换句话说，除非你很积极地让神经元活动起来，否则你的头脑里什么也不会发生。必须引起读者的好奇，促进、要求，并鼓励读者去解决问题、得出结论、产生新的知识。为此，需要发出挑战，留下练习题和拓宽思路的问题，并要求读者完成一些实践活动，让左右脑都动起来，而且要利用到多种思维。

创建一个浴缸类来指定样式有意义吗？还是应该为整个浴室建立样式？



引起读者的注意，而且要让他一直保持注意。我们可能都有过这样的体验，“我真的想把这个学会，不过看过一页后实在是让我昏昏欲睡。”你的大脑注意的是那些不一般、有意思、有些奇怪、抢眼的、意料之外的东西。学习一项有难度的新技术并不一定枯燥。如果学习过程不乏味，你的大脑很快就能学

会。

影响读者的情绪。现在我们知道了，记忆能力很大程度上取决于所记的内容对我们的情绪有怎样的影响。如果是你关心的东西，就肯定记得住。如果让你感受到了什么，这些东西就会留在你的脑海中。不过，我们所说的可不是什么关于男孩与狗的伤心故事。这里所说的情绪是惊讶、好奇、觉得有趣、想知道“什么……”还有就是一种自豪感，如果你解决了一个难题，学会了所有人都觉得很难的东西，或者发现你了解的一些知识竟是那些自以为无所不能的傲慢家伙所不知道的，此时就会有一种自豪感油然而生。



元认知：有关思考的思考

如果你真的想学，而且想学得更快、更深，就应该注意你怎样才会专注起来，考虑自己是怎样思考的，并了解你的学习方法。

我们中间大多数人长这么大可能都没有上过有关元认知或学习理论的课程。我们想学习，但是很少有人教我们怎么来学习。

不过，这里可以做一个假设，如果你手上有这本书，你想学习项目管理，而且可能不想花太多时间。如果你想把这本书中读到的知识真正用起来，就需要记住你读到的所有内容。为此，必须理解这些内容。要想最大程度地利用这本书或其他任何一本书，或者掌握学习经验，就要让你的大脑负起责来，要求它记住这些内容。

怎么做到呢？技巧就在于要让你的大脑认为你学习的新东西确实很重要，对你的生活有很大影响。就像老虎出现在面前一样。如若不然，你将陷入旷日持久的拉锯战中，虽然你很想记住所学的新内容，但是你的大脑却会竭尽全力地把它们拒之门外。

那么究竟怎样才能让大脑把HTML&CSS看作是一只饥饿的老虎呢？

这有两条路，一条比较慢，很乏味。另一条路不仅更快，还更有效。慢方法就是大量地重复。你肯定知道，如果反反复复地看到同一个东西，即便再没有意思，你也能学会并记住。如果做了足够的重复，你的大脑就会说，“尽管看上去这对他说来好像不重要，不过，既然他这样一而再、再而三地看同一个东西，所以我假定这应该是重要的。”

更快的方法是尽一切可能让大脑活动起来，特别是开动大脑来完成不同类型的活动。如何做到这一点呢？上一页列出的学习原则正是一些主要的可取做法，而且经证实，它们确实有助于让你的大脑全力以赴。例如，研究表明，把文字放在所描述图片的中间（而不是放在这一页的别处，比如作为标题，或者放在正文中），这样会让你的大脑更多地考虑这些文字与图片之间有什么关系，而这就会让更多的神经元“点火”。让更多的神经元“点火”=你的大脑更有可能认为这些内容值得关注，而且很可能需要记下来。

交谈式风格也很有帮助，当人们意识到自己在与“别人”交谈时，往往会更专心，这是因为他们总想跟上谈话的思路，并能做出适当的发言。让人惊奇的是，大脑并不关心“交谈”的对象究竟是谁，即使你只是与一本书“交谈”，它也不会在乎！另一方面，如果写作风格很正统、干巴巴的，你的大脑就会觉得，这就像坐在一群人当中被动地听人做报告一样，很没意思，所以不必在意对方说的是什么，甚至可以打瞌睡。

不过，图片和交谈风格还只是开始而已，能做的还有很多。

我想知道怎么才能
骗过我的大脑，让它
记住这些东西……



我们是这么做的：

我们用了许多图，因为你的大脑更能接受看得见的东西，而不是纯文字。对你的大脑来说，一图胜千言。如果既有文字又有图片，我们会把文字放在图片当中，因为文字处在所描述的图片中间时，大脑的工作效率更高，倘若把这些描述文字作为标题，或者“淹没”在别处的大段文字中，就达不到这种效果了。

我们采用了重复手法，会用不同方式，采用不同类型的媒体，运用多种思维手段来介绍同一个东西，目的是让有关内容更有可能储存在你的大脑中，而且在大脑中多个区域都有容身之地。

我们会用你想不到的方式运用概念和图片，因为你的大脑喜欢新鲜玩艺。在提供图和思想时，至少会含着一些情绪因素，因为如果能产生情绪反应，你的大脑就会投入更大的注意。而这会让你感觉到这些东西更有可能被记住，其实这种感觉可能只是有点幽默，让人奇怪或者比较感兴趣而已。

我们采用了一种针对个人的交谈式风格，因为当你的大脑认为你在参与一个会谈，而不是被动地听一场演示汇报时，它就会更加关注。即使你实际上在读一本书，也就是说在与书“交谈”，而不是真正与人交谈，但这对你的大脑来说并没有什么分别。

在这本书里，我们加入了100多个实践活动，因为与单纯的阅读相比，如果能实际做点什么，你的大脑就会更乐于学习，更愿意去记。这些练习都是我们精心设计的，有一定的难度，但是确实能做出来，因为这是大多数人所希望的。

我们采用了多种学习模式，因为尽管你可能想循序渐进地学习，但是其他人可能希望先对整体有一个全面的认识，另外可能还有人只是想看看一个例子。不过，不管你想怎么学，要是同样的内容能以多种方式来表述，这对每一个人都会有好处。

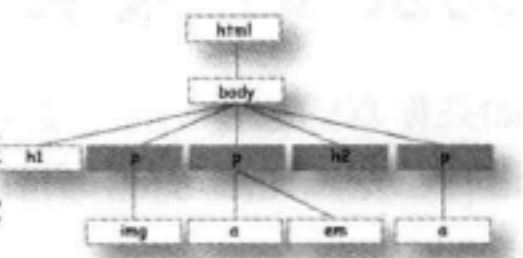
这里的内容不只是单单涉及左脑，也不只是让右脑有所动作，我们会让你的左右脑都动起来，因为你的大脑参与得越多，你就越有可能学会并记住，而且能更长时间地保持注意力。如果只有一半大脑在工作，通常意味着另一半有机会休息，这样你就能更有效率地学习更长时间。

我们会讲故事，留练习，从多种不同的角度来看同一个问题，这是因为，如果要求大脑做一些评价和判断，它就能更深入地学习。

我们会给出一些练习，还会问一些问题，这些问题往往没有直截了当的答案，通过克服这些挑战，你就能学得更好，因为让大脑真正做点什么的话，它就能学会并记住。想想吧，如果只是在体育馆里看着别人流汗，则对于保持你自己的体形肯定不会有什么帮助，正所谓临渊羡鱼，不如退而结网。不过另一方面，我们会竭尽所能不让你钻牛角尖，把劲用错了地方，而是能把功夫用在点子上。也就是说，你不会为弄明白一个难懂的例子而耽搁，也不会花太多时间去弄明白一段艰涩难懂而且通篇行话的文字，我们的描述也不会太过简洁而让人无从下手。

我们用了拟人手法。在故事中，在例子中，还有在图中，你都会看到人的出现，这是因为你本身是一个人，不错，这就是原因。如果和人打交道，相对于某件东西而言，你的大脑就会更为关注。

这里使用了一种80/20方法。我们相信你会成为一个一流的Web开发人员，这绝不会是你唯一的参考书。所以我们不打算面面俱到，涵盖所有内容，这里只提供你真正需要的东西。



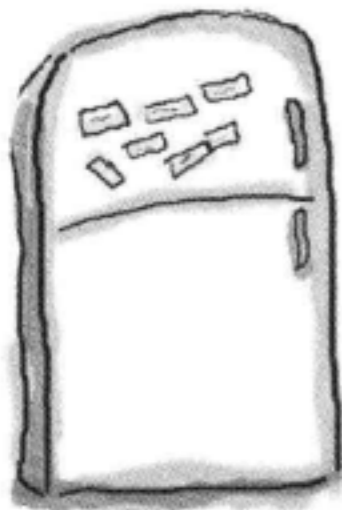
扮演浏览器



BULLET POINTS

猜谜游戏





把这一页撕下来，贴到你的冰箱上。

可以用下面的方法 让你的大脑就范

好了，我们该做的已经做了，剩下的就要看你自己的了。下面的提示可以作为一个起点，听一听你的大脑是怎么说的，弄清楚对你来说哪些做法可行，哪些做法不能奏效。要做些新的尝试。

① 慢一点。你理解的越多，需要记的就越少。

不要光是看看就行了。停下来，好好想一想。书中提出问题的時候，你不要直接去翻答案。可以假想真的有人在问你这个问题。你让大脑想得越深入，就越有可能学会并记住它。

② 做练习，自己记笔记。

我们留了练习，但是如果这些练习的解答也由我们一手包办，那和有人替你参加考试有什么分别？不要只是坐在那里看着练习发呆。拿出笔来，写一写画一画。大量研究都证实，学习过程中如果能实际动动手，将改善你的学习。

③ 阅读“没有傻问题”部分。

顾名思义。这些问题不是可有可无的旁注，它们绝对是核心内容的一部分！千万不要跳过去不看。

④ 上床睡觉之前不要再看别的书，至少不要看其他有难度的东西。

学习中有一部分是在你合上书之后完成的（特别是，要把学到的知识长久地记住，这往往无法在看书的过程中做到）。你的大脑也需要有自己的时间，这样才能再做一些处理。如果在这段处理时间内你又往大脑里灌输了新的知识，那么你刚才学的一些东西就会丢掉。

⑤ 要喝水，而且要多喝点水。

能提供充足的液体，你的大脑才能有最佳表现。如果缺水（可能在你感觉到口渴之前就已经缺水了），学习能力就会下降。

⑥ 讲出来，而且要大声讲出来。

说话可以刺激大脑的另一部分。如果你想看懂什么，或者想更牢地记住它，就要大声地说出来。更好的办法是，大声地解释给别人听。这样你会学得更快，而且可能会有新的发现，这是以前光看不说的时候所没有的。

⑦ 听听你的大脑怎么说。

注意一下你的大脑是不是负担太重了。如果发现自己开始浮光掠影地翻看，或者刚看的東西就忘记了，这说明你该休息一会了。达到某个临界点时，如果还是一味地向大脑里塞，这对于加快学习速度根本没有帮助，甚至还可能影响正常的学习进程。

⑧ 要有点感觉。

你的大脑需要知道这是很重要的东西。要真正融入到书中的故事里。为书里的照片加上你自己的图题。你可能觉得一个笑话很蹩脚，不太让人满意，但这总比根本无动于衷要好。

⑨ 真正做些工作！

把这些知识应用到你的日常工作上去，利用你所学的方法完成项目决策。具体做些工作，你会得到书中练习和活动以外的经验。所需要的只是一支笔和一个要解决的问题——一个使用HTML和CSS可以妥善解决的问题。

重要说明

要把这看做是一个学习过程，而不要简单地把它看成是一本参考书。我们在安排内容的时候有意做了一些删减，只要是对有关内容的学习有妨碍，我们都毫不留情地把这些部分一律删掉。另外，第一次看这本书的时候，要从第一页从头看起，因为书中后面的部分会假定你已经看过而且学会了前面的内容。

我们先从基本HTML开始讲起，然后是基于标准的HTML5。

要编写基于标准的HTML，有很多技术细节需要了解，但是这些细节对于你学习基本HTML并没有帮助。我们的方法是先让你掌握HTML的基本概念（而不用担心那些细节），等你有了扎实的HTML基础后，再教你编写符合标准的HTML（HTML的最新版本是HTML5）。这有一个额外的好处：学习完基础之后，那些技术细节会更有意义。

开始使用CSS时，编写符合标准的HTML也很重要，所以我们强调在用CSS做具体工作之前，先要建立基于标准的HTML。

我们没有涵盖每一个HTML元素或属性，或者已创建的所有CSS属性。

有太多的HTML元素、太多的属性，还有太多的CSS属性。当然，它们都很有意思，不过我们的目标是写一本篇幅适量的书，起码重量不能超过读这本书的读者，所以这里没有面面俱到。我们的重点是核心HTML元素和对初学者最重要的CSS属性，确保你能真正地、深入地、扎实地了解如何使用以及何时使用这些内容。不管怎样，一旦读完这本《Head First HTML 与 CSS》，你就可以选择任何参考书，迅速补上我们没有谈到的元素和属性。

这本书提倡页面结构与页面表现的清晰分离。

如今，好的Web页面都使用HTML建立内容的结构，使用CSS提供样式和表现。20世纪90年代的Web页面通常使用另外一种模型，页面中HTML不仅用来指定结构，同时还用来建立样式。这本书会教你使用HTML建立结构，而用CSS指定样式，我们认为没有理由再教你那些过时的坏习惯。

建议你学习这本书时最好使用多种浏览器。

尽管我们会教你编写基于标准的HTML和CSS，但是你可能还会（而且有可能经常）发现不同的Web浏览器显示页面时会有一些细微的差别。所以，我们建议你至少选择两种现代浏览器，用它们分别测试你的页面。这会让你了解到不同浏览器之间的差异，另外还能获得一些经验，知道如何创建适用于多种不同浏览器的页面。

我们通常使用标记名表示元素名。

我们可能并不会说“a元素”或“‘a’元素”，而是使用一个标记名，如“<a>元素”。从理论上讲这可能并不对（因为<a>只是一个开始标记，而不是一个完整的元素），但是这样更有可读性，另外我们通常会在名字后面加上“元素”两个字来避免混淆。

书里的实践活动绝不是可有可无。

这里的练习和实践活动不是可有可无的装饰和摆设，它们也是这本书核心内容的一部分。其中有些练习和活动有助于记忆，有些能够帮助你理解，还有一些对于如何应用所学的知识很有帮助。千万不要把这些练习跳过不做。甚至填字游戏也很重要，它们可以帮助你有关概念植入你的大脑。不过更重要的是，它们可以让你的大脑有机会在不同的上下文考虑这些词汇和概念。

我们有意安排了许多重复，这些重复非常重要。

Head First系列图书有一个与众不同的地方，这就是，我们希望你确实实地掌握这些知识，另外希望在学完这本书之后你能记住学过了什么。大多数参考书都不太重视重复和回顾，但是由于这是一本有关学习的书，你会看到一些概念一而再、再而三地出现。

例子尽可能简洁。

有读者告诉我们，如果查了200行代码才能找到要理解的那两行代码，这是很让人郁闷的。这本书里大多数例子往往都开门见山，作为上下文的代码会尽可能的少，这样你就能一目了然地看到哪些东西是需要你学习的。别指望所有示例都非常健壮，要知道这里的代码甚至是不完整的。这些例子特意写得很简单，以方便你学习，而且它们的功能不一定完备。

所有示例代码和应用都已经在网上公布，以便你下载。所有代码都可以在<http://wickedlysmart.com/hfhtmlcss/>找到。

“Brain Power (头脑风暴)”练习没有答案。

有一些头脑风暴练习根本没有所谓正确的答案，另外一些练习中，头脑风暴实践活动的一部分学习过程就是让你确定：你的答案是否正确，以及在何种情况下正确。有些头脑风暴练习中，你会得到一些提示，为你指明正确的方向。

技术审校人员 (第一版)



Louise Barr



Joe Konior



Valentin Crettaz



Corey McGlone



Barney Marispini



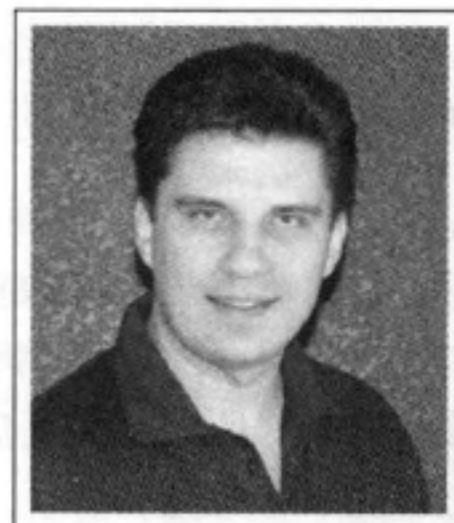
Pauline McNamara



Johannes de Jong



Marcus Green



Ike Van Atta



David O'Meara

埃菲尔铁塔

直言快语的极限审校团队队长

Pauline 获得“最严厉审校”奖

致我们的审校团队:

非常感谢我们的审校团队。Johannes de Jong 组织并领导了整个项目，他作为“系列之父”，着力保证所有工作顺利进行。Pauline McNamara，整个项目的“协调人”，负责联系各个方面，他也是第一个指出我们的例子有点过于“花哨”。整个团队的工作充分表明我们是多么需要他们的技术经验以及对细节的关注。Valentin Crettaz、Barney Marispini、Marcus Green、Ike Van Atta、David O'Meara、Joe Konior 和 Corey McGlone 在审校中一个纰漏都不放过，经过他们的努力，这本书更趋完美。你们真是太棒了！还要感谢 Corey 和 Pauline，督促我们纠正了太多过于正式（确切地讲，有些不合适）的措辞。还要特别感谢 JavaRanch 提供了那么多优秀的资源。

非常感谢 Louise Barr，最优秀的 Web 设计师，她让我们严谨地完成设计，同时正确地使用 HTML 和 CSS（尽管你可能对我们现在的设计还是颇有微辞）。

致谢（第一版）*

致更多技术审校：

同样非常感谢我们最尊敬的技术审校David Powers。对David我们真是又爱又恨，他让我们拼命工作，不过我们也得到了丰厚的回报。说真的，正是根据David的意见，我们对这本书做了大幅修改，现在这本书的技术含量绝对是之前的两倍。谢谢你，David。

致O'Reilly：

要向我们的编辑Brett McLaughlin致以最真诚的感谢，他牺牲了与家人共享天伦之乐的时间，为这本书扫清了所有障碍。Brett还为这本书的编辑相当费心费力（编辑Head First书可真不是件容易的事）。谢谢你，Brett，如果没有你，就不会有这本书。



Brett McLaughlin

最诚挚地感谢整个O'Reilly团队：Greg Corrin、Glenn Bisignani、Tony Artuso和Kyle Hart全力做市场推广，我们非常欣赏他们别出心裁的方式。感谢Ellie Volkhausen充满灵感的封面设计，有着一如既往的独特风格，还要感谢Karen Montgomery，她让这本书的封面有了生命力。与以往一样，还要感谢你，Colleen Gorman，谢谢你为文字编辑付出的艰辛劳动（还以此为乐）。如果没有Sue Willing和Claire Cloutier的努力，我们不可能完成这样一本色彩丰富的书。

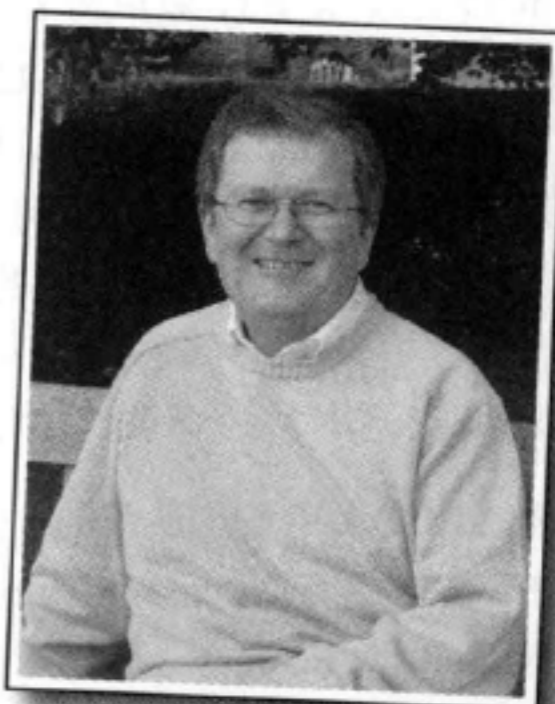
Head First书的致谢肯定少不了Mike Loukides，是他将Head First的概念变成了这样一个系列，另外要感谢Tim O'Reilly一如既往的支持。最后，还要感谢Mike Hendrickson，是他让我们加入Head First这个大家庭，而且坚信我们能够胜任这个工作。

致Kathy Sierra和Bert Bates：

最后，但绝不是不重要，非常感谢Kathy Sierra和Bert Bates，不仅是我们的合作伙伴，你们绝对是这个系列的“核心人物”。感谢你们对我们的信任甚至多于你们的孩子。再次希望我们没有辜负你们的期望。那三天的讨论为这本书指明了方向，希望很快还有这样的机会。噢，对了，下一次能不能打个电话给LTJ，让他来西雅图旅行一趟？

最让人尊敬的技术
审校

David Powers



别让柔和的毛衣欺骗了，这家伙相当强硬（当然是指技术方面）。

Bert Bates



Kathy Sierra



Kara

正在忙着完成《Head First Parelli》。

*之所以要感谢这么多人，这是因为我们发现了这样一条定律，书中致谢里提到的每个人都至少会买一本书，可能还会买好几本书，给亲戚和周围的所有人都送上一本。如果你希望我们在下一本书的致谢里提到你，而且你的家人很多的话，则可以写信给我们。

技术审校人员（第二版）

要是不认识高水平的HTML与CSS大师David Powers，没有他当这本书的技术审校来审查错漏，我们简直夜不能寐。事实上，这本书第一版出版之后，已经过去了这么多年，我们不得不请了一位私家侦探才找到他（说来话长，不过最后总算在他的地下HTML与CSS研究室找到了）。说正经的，这本书中的所有技术错误都要归咎于作者（也就是我们），不过，我们可以给你打保票，David为确保我们不出错绝对是竭尽全力。重申一句，David对这本书的要求真是相当苛刻。

我们要感谢技术审校团队的每一个人。Joe Konior再次加入了这一版的工作，还有Dawn Griffiths（Head First C的合作者），另外还有Shelley Powers（他绝对是一个丰富的HTML与CSS宝库，在Web开发方面有多年的经验）。再说一次，你们太棒了！你们的反馈那么细致深入，对我们大有帮助。谢谢你们。



David Powers

不穿粉红毛衣了，现在HTML与CSS水平更高超！



Dawn Griffiths



Joe Konior



Mike Hendrickson

致谢（第二版）

最需要感谢的是我们的主编Mike Hendrickson，是他在各个方面让这本书顺利得以出版（而不只是写书这一个环节），在整个过程中有他一直陪伴，更重要的是（这也是所有编辑做的最重大的事情），他充分信任我们能完成这个任务！谢谢你，Mike，没有你，我们的书都不可能问世。十多年来，你一直在支持我们，实在让我们感激不尽！

当然，出版一本书绝对是个大工程，正是O'Reilly这些才能非凡而又友好温和的人在幕后辛勤地工作，才有了这样一本书。我们要向整个O'Reilly团队致以最诚挚的感谢：Kristen Borg（无与伦比的制作编辑）、非凡的Rachel Monaghan（校对）、Ron Strauss（感谢你缜密的索引）、Rebecca Demarest（感谢你提供的插图）、Karen Montgomery（一流的封面设计者），最后但绝不是最不重要，还要感谢Louise Barr，她总能让我们的页面看起来更棒。



Louise Barr

Safari® 图书在线



Safari图书在线是一个按需提供资源的数字图书馆，从中可以很容易地搜索7500本技术书、参考书和视频，快速找到你想要的答案。

只需订阅，就可以从我们的在线图书馆阅读任何页面，观看任何视频。你可以在你的手机和移动设备上看书，能够在新书出版之前获得书目，还可以享有特权查看正在编写的书稿并向作者提出反馈意见。你可以剪切粘贴代码示例、整理最喜欢的图书、下载你需要的章节、为重要内容设置书签、创建笔记、打印页面，此外还有大量特性可以节省时间，让你从中受益。

O'Reilly Media已经将这本书（英文版）上传到Safari图书在线服务。要想通过数字方式全面访问这本书以及O'Reilly和其他出版商提供的其他类似图书，可以免费注册 <http://my.safaribooksonline.com>。

目录（概览）

引子	xxv
1 Web语言：认识HTML	1
2 认识HTML中的“HT”：深入了解超文本	43
3 Web页面建设：构建模块	77
4 Web镇之旅：连接起来	123
5 认识媒体：为你的页面增加图像	163
6 严肃的HTML：标准及其他	219
7 加一点样式：CSS入门	255
8 扩大你的词汇量：增加字体和颜色样式	311
9 与元素亲密接触：盒模型	361
10 高级Web建设：div与span	413
11 摆放元素：布局与定位	471
12 现代HTML：HTML5标记	545
13 建立表格：表格与更多列表	601
14 实现交互：HTML表单	645
附录：其他十大主题（我们没有谈到的）	697

目录

引子

你的大脑与HTML和CSS。你想学些新东西，但是你的大脑总是帮倒忙，它会努力让你记不住所学的东西。你的大脑在想：“最好留出空间来记住那些确实重要的事情，比如要避开哪个野生动物，还有裸体滑雪是不是不太好。”那么，如何让你的大脑就范，让它认为如果不知道HTML和CSS你将无法生存？

谁适合看这本书？	xxvi
元认知	xxix
我们是这么做的	xxx
让你的大脑就范	xxxii
技术审校人员（第一版）	xxxiv
致谢（第一版）	xxxv
技术审校人员（第二版）	xxxvi
致谢（第二版）	xxxvi

认识HTML

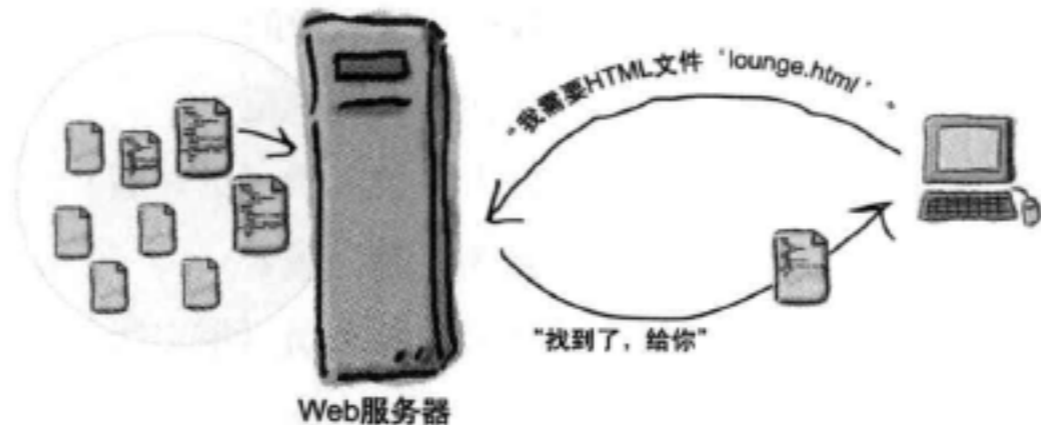
Web语言

要进入Web世界，在你面前只有一个障碍，就是要学会它的语言。超文本标记语言，或者简称为HTML。准备好了吗？我们要上几节语言课了。读完这一章之后，你不仅会了解一些基本的HTML元素，还能讲出带一点样式的HTML。说真的，等学完这本书，你就会像是从小在Web镇长大的一样，能自如地运用HTML。

别有压力，不过等你完成这个Web页面后，可能会有数以千计的人访问这个页面。所以它不仅要正确，而且看上去还要很漂亮！



Web让广播明星黯然失色	2
Web服务器能做什么？	3
你写的代码（HTML）	4
浏览器创建的页面	5
你在Starbuzz咖啡馆交好运了	9
创建Starbuzz Web页面	11
创建一个HTML文件(Mac系统)	12
创建一个HTML文件（Windows系统）	14
现在，再回到Starbuzz……	17
保存你的成果	18
在浏览器中打开你的Web页面	19
测试你的页面	20
完工了吗？	23
另一个测试	24
标记剖析	25
认识style元素	29
给Starbuzz网站加点样式……	30
测试样式……	31
练习答案	38



2

深入了解超文本

认识HTML中的“HT”

有人在说“超文本”？那是什么？哦，这就是Web的基础。第1章中我们简单认识了HTML，知道它是一种不错的标记语言（HTML中的“ML”），用来描述网页的结构。现在我们来了解HTML中的“HT”，也就是超文本（hypertext），有了它，我们就能摆脱单个页面的束缚，链接到其他页面。顺便我们还要认识一个强大的新元素——<a>元素，了解“相对”是多么美妙的东西。好了，系好你的安全带，准备学习一些超文本吧。



Head First休闲室，全新改良	44
创建新的休闲室	46
我们做了什么？	48
了解属性	51
组织	56
组织休闲室……	57
技术难点	58
规划你的路径……	60
修复那些损坏的图像……	66
练习答案	73

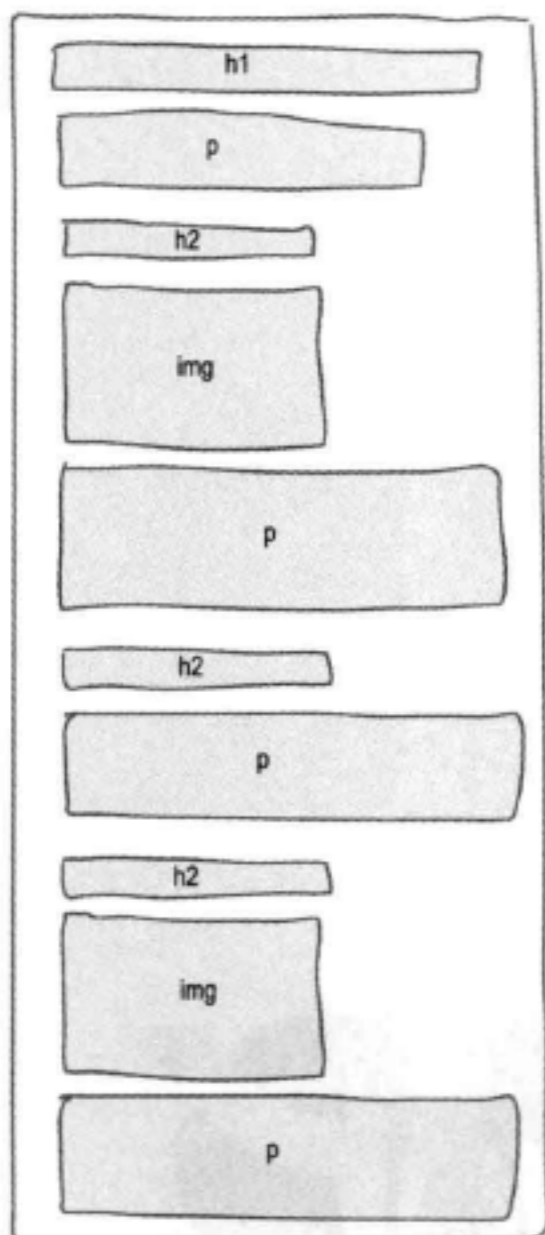


构建模块

3

Web页面建设

我听说这本书会教我具体创建Web页面？你已经学到不少了：标记、元素、链接、路径……不过，如果不学以致用，创建一些一流的Web页面，这些知识就毫无意义。这一章我们要把重点放在建设上：你要从概念到蓝图来设计Web页面，浇注地基，完成建设，甚至最后还可以加几笔润色。戴上安全帽，系好安全带，我们会增加一些新工具，让你掌握一些更深入的内部知识，有了这些知识，你也能成为这个行业的佼佼者。



从日志到网站，以12迈的速度出发	79
粗略的设计草图	80
从草图到略图	81
从略图到网页	82
测试Tony的Web页面	84
增加一些新元素	85
认识<q>元素	86
很长的引用	90
增加<blockquote>	91
<q>和<blockquote> 谜案的真相	94
再回到Tony的网站……	100
当然，也可以使用<p>元素创建列表……	101
两步轻松构建HTML列表	102
城市列表测试	104
把一个元素放在另一个元素中称为“嵌套”	107
要理解嵌套关系，画一个图	108
使用嵌套确保标记匹配	109
练习答案	117

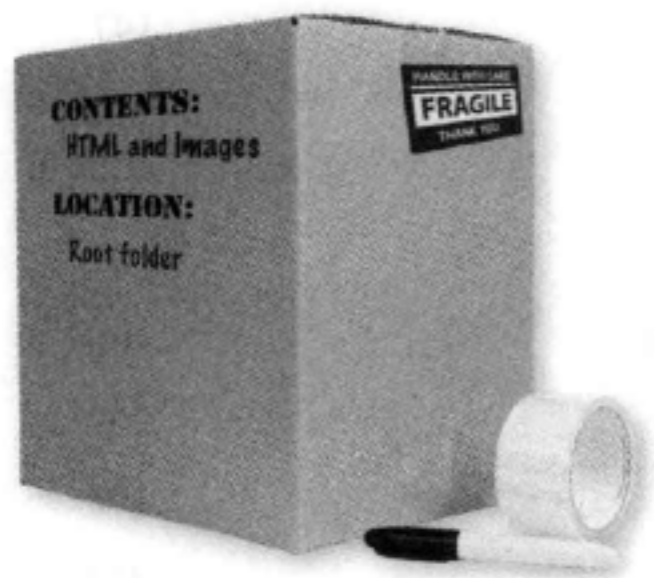
连接起来

4

Web镇之旅

Web页面是互联网给我们的最好的一道菜。到目前为止，你只是在自己的计算机上创建HTML页面，所链接的页面也都在自己的计算机上。接下来我们会有所改变。这一章会鼓励你把这些Web页面发布到互联网上，让你的朋友、粉丝和客户都能真正看到。通过破解 h, t, t, p, :, /, /, w, w, w 代码，我们还将揭开链接到其他页面的秘密。好了，收拾好行李，向下一站Web镇进发。

在Web上发布Starbuzz网站（或你自己的网站）	124
找一家托管公司	125
如何得到一个域名？	126
搬家	128
把你的文件复制到根文件夹	129
用两页尽可能讲清楚FTP	130
回到正题……	133
我们的主干道, USA	134
什么是HTTP？	135
什么是绝对路径？	136
默认页面如何工作	139
Earl需要你帮他确定URL	140
如何链接到其他网站？	142
链接到Caffeine Buzz	143
现在来试一试……	144
完善的Web页面	147
试一试标题……	148
链接到一个页面	149
使用id属性为<a>创建目标	150
如何用id链接到元素	151
链接到一个新窗口	155
使用target打开新窗口	156
练习答案	160

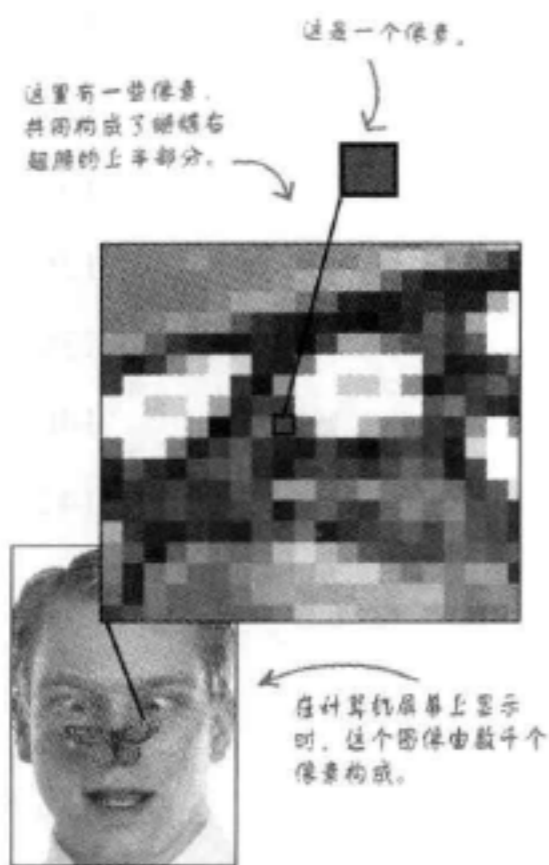


5

为你的页面增加图像

认识媒体

笑一笑，说“茄子”。说真的，应该是笑一笑，说“gif”、“jpg”或“png”。你为Web“冲印照片”时，就会用到它们。这一章中，我们将学习如何为你的页面添加第一个媒体类型：图像。你是不是拍摄了一些数码照片想要发到网上？没问题。是不是需要在页面上加一个logo？小事一桩。不过在深入学习这些内容之前，是不是需要正式介绍一下元素？真是抱歉，我们不是有意疏忽，只是一直没有找到“合适的机会”。作为补偿，下面用一整章专门讨论。学完这一章，你就会全面地了解使用元素及其属性的所有细节了。你还会亲眼看到这个小元素是怎样让浏览器做大量额外的工作来获取和显示图像的。



浏览器如何处理图像	164
图像是如何工作的	167
: 不再只是相对链接	171
一定要提供候选格式	173
调整图像大小	174
创建超级粉丝网站：myPod	175
哇！图像太大了	178
打开图像	182
调整图像大小	183
修复myPod HTML	188
myPod的更多照片	190
把缩略图变成链接	196
为照片创建单独的页面	197
那么，怎么链接到图像呢？	198
要使用什么格式？	203
透明，还是不透明？这是个问题……	204
等一下，Web页面背景色到底是什么？	206
查看有蒙版的logo	207
把logo增加到myPod Web页面	208
练习答案	213

6

标准及其他

严肃的HTML

关于HTML还要知道些什么？很不错，现在你已经掌握了HTML。实际上，是不是该转向CSS了？来学习如何让这些乏味的标记看起来更漂亮。在此之前，我们需要确保你已经掌握了足够的HTML知识，有能力应对激烈的竞争。不要误解我们的意思，没错，你一直在写一流的HTML，不过还需要了解额外的一些知识，才能真正建立“工业标准”的HTML。还要利用这个机会确保你使用的是最新、最棒的HTML标准，也就是HTML5。保证了这一点，就可以确保HTML在所有浏览器上（至少是你关心的浏览器上）能有更一致的显示，不仅如此，还可以在最新的iPhone、iPod之类的设备上（选择你最喜欢的设备）很好地显示。你的页面会加载得更快，可以很好地结合CSS，能够随着标准发展稳步走向未来。准备好了吗？这一章会让你从一个Web工匠变成真正的Web高手。



HTML简史	222
新的、改进的HTML5 doctype	227
HTML, 新的“活标准”	228
增加文档类型定义	229
测试doctype	230
认识W3C验证工具	233
验证Head First休闲室	234
唉呀，我们遇到一个问题……	235
修正错误	236
就快完成了……	237
增加一个<meta>指定字符编码	239
让验证工具（和很多浏览器）接受<meta>标记……	240
事不过三？	241
叫所有HTML专业人员来拿手册……	244
练习答案	251

CSS入门

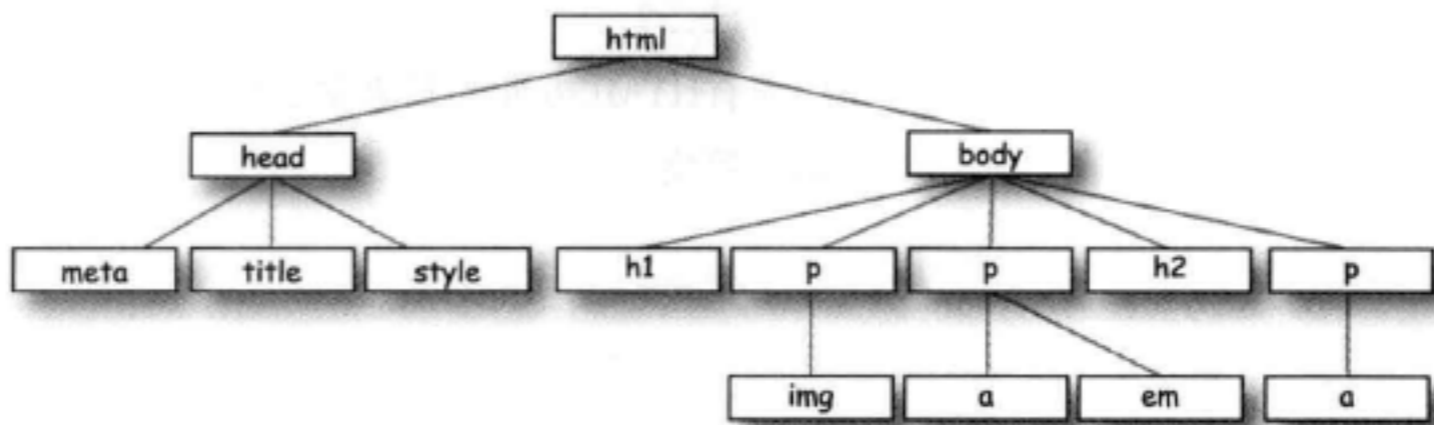
7

加一点样式

我听说这本书会介绍CSS。到目前为止，你一直在专心学习HTML来创建Web页面结构。不过可以看到，浏览器对样式有很多要求。当然，我们可以请时尚人士来帮忙，不过没有必要。利用CSS，你就能完全掌控页面的表现，通常甚至不用对HTML做任何改变。真的这么容易吗？嗯，你得学习一种新的语言。毕竟，Web镇是一个讲多种语言的小镇。读完这一章对CSS语言学习的介绍后，你就能胸有成竹地站在主干道两边与Web镇的人们交流了。



你已经离开堪萨斯	256
偶然听到的Web镇“交换空间”节目	258
结合HTML和CSS	259
把CSS放入HTML	261
为休闲室增加样式	262
再在欢迎消息下面加一条线	265
那么，选择器到底如何工作？	267
通过图解来研究选择器	270
为清凉饮料和路线说明页面加入休闲室页面的样式	273
来谈谈继承……	281
覆盖继承	284
把元素增加到greentea类	287
创建一个类选择器	288
更深入地研究类……	290
关于应用样式的世界上最简短最快捷的指南	292
练习答案	303



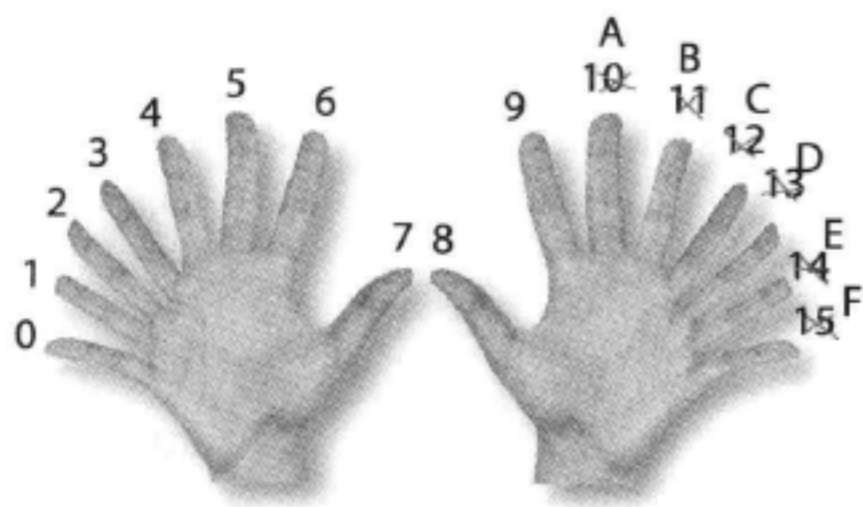
增加字体和颜色样式

8

扩大你的词汇量

你的CSS语言课程进行得很顺利。你已经掌握了CSS的基础知识，而且知道了如何创建CSS规则来选择和指定一个元素的样式。现在该扩大你的词汇量了，这意味着需要学习一些新的属性，了解它们能为你做什么。这一章中，我们会介绍影响文本显示的一些最常用的属性。为此，你需要对字体和颜色有所了解。你会发现，不必千篇一律地使用其他人都在用的字体，也不用按部就班地使用浏览器为段落和标题设置的默认字体大小和样式（真的很难看）。你还会看到除了让眼睛看着舒服外，还可以对颜色做很多其他设置。

从三万英尺的高空看文本和字体	312
字体系列到底是什么？	314
使用CSS指定字体系列	317
让Tony的旅行日志焕然一新	318
每个人都有不同的字体，我该如何处理？	321
Web字体如何工作	323
如何为页面增加Web字体……	325
调整字体大小	328
那么，我到底该如何指定字体大小呢？	330
下面修改Tony的Web页面中的字体大小	332
改变字体粗细	335
为字体增加风格	337
让Tony的引用有一点斜体风格	338
Web颜色如何工作？	340
如何指定Web颜色？来数数看有多少种方法……	343
十六进制码速查指南	346
如何找到Web颜色	348
再回到Tony的页面……	351
用不到一页的篇幅介绍	353
删除下划线……	354
练习答案	357



盒模型

9

与元素亲密接触

要推进Web建设，确实需要了解建筑材料。这一章中，我们会仔细研究我们的建筑材料：HTML元素。我们会把块元素和内联元素放在显微镜下，仔细观察它们的组成。你会看到，利用CSS，对于构造元素的各个方面几乎都可以控制。不过，我们并不会就此止步，你还会了解如何为元素提供唯一的身份。如果这些还不够，这一章还会告诉你什么时候要使用多个样式表，以及为什么要用多个样式表。好了，翻开下一页，与元素来个亲密接触吧。

休闲室要升级	362
先做一些简单的升级	364
查看新行高	366
准备全面翻新	367
仔细分析盒模型	368
对盒子能做哪些设置	370
创建guarantee样式	375
测试段落边框	376
保证段落的内边距、边框和外边距	377
增加一个背景图像	380
修正背景图像	383
如何只在左边增加内边距?	384
如何只在右边增加外边距?	385
边框简明指南	386
完善边框	389
休闲室页面中使用id	396
使用多个样式表	399
样式表，不再只面向浏览器……	400
直接在CSS中增加媒体查询	401
练习答案	407

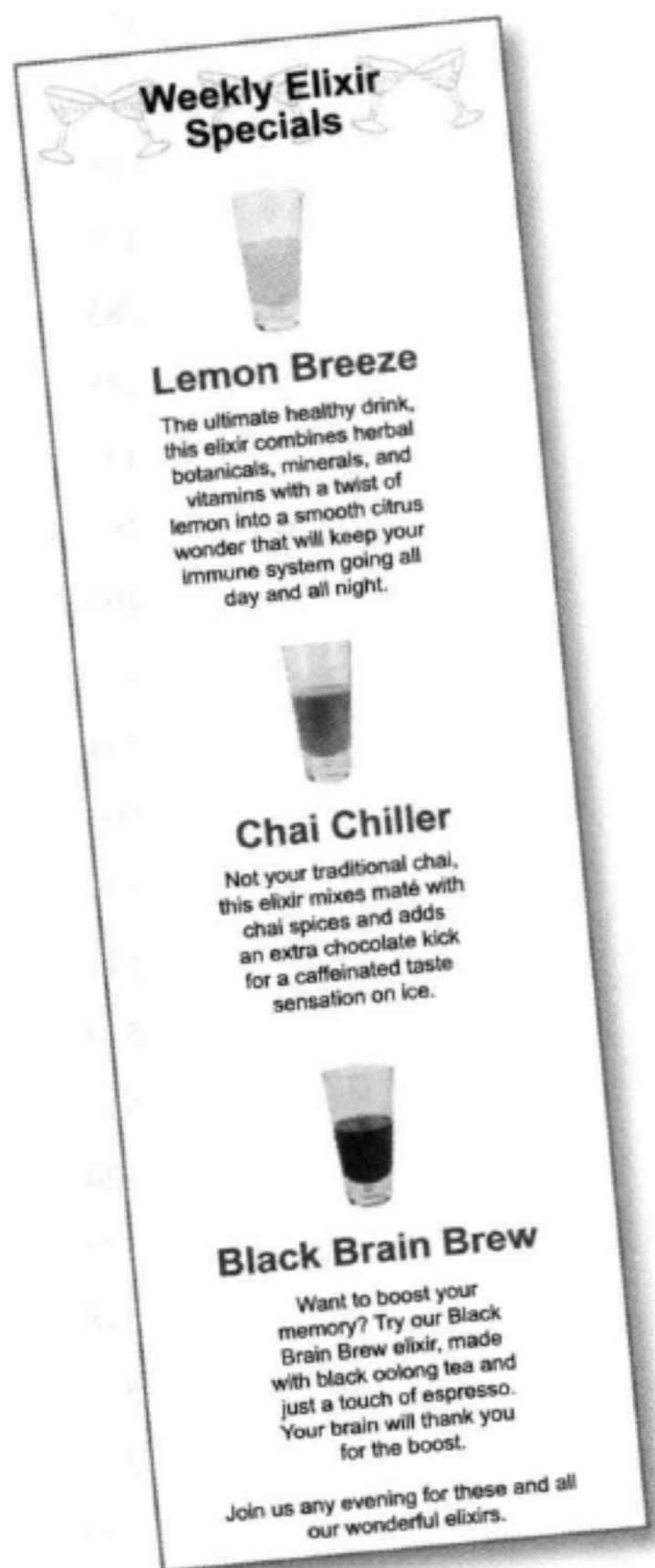


10

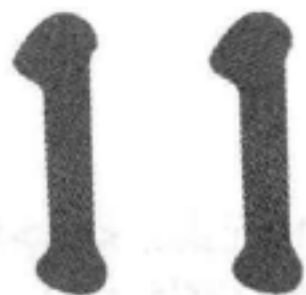
div与span

高级Web建设

该建个大工程了。这一章中，我们要介绍两个新的HTML元素：`<div>`和``。它们绝对不是简简单单的“小玩艺”，而是堪称举足轻重的钢铁支架。利用`<div>`和``，你能构建重要的支撑架构，一旦有了这些架构，就能用各种新颖、强大的方式为它们增加样式。我们已经注意到，你的CSS工具箱开始有点满了，所以很有必要为你展示一些捷径，让你能更容易地指定属性。这一章还会请到一些特殊的客人——伪类（pseudo-classes），利用它们你可以创建一些非常有趣的选择器（如果你觉得“伪类”这个名字很不错，想把它当做你的下一个乐队的名字，嘿嘿，太晚了，我们已经捷足先登）。



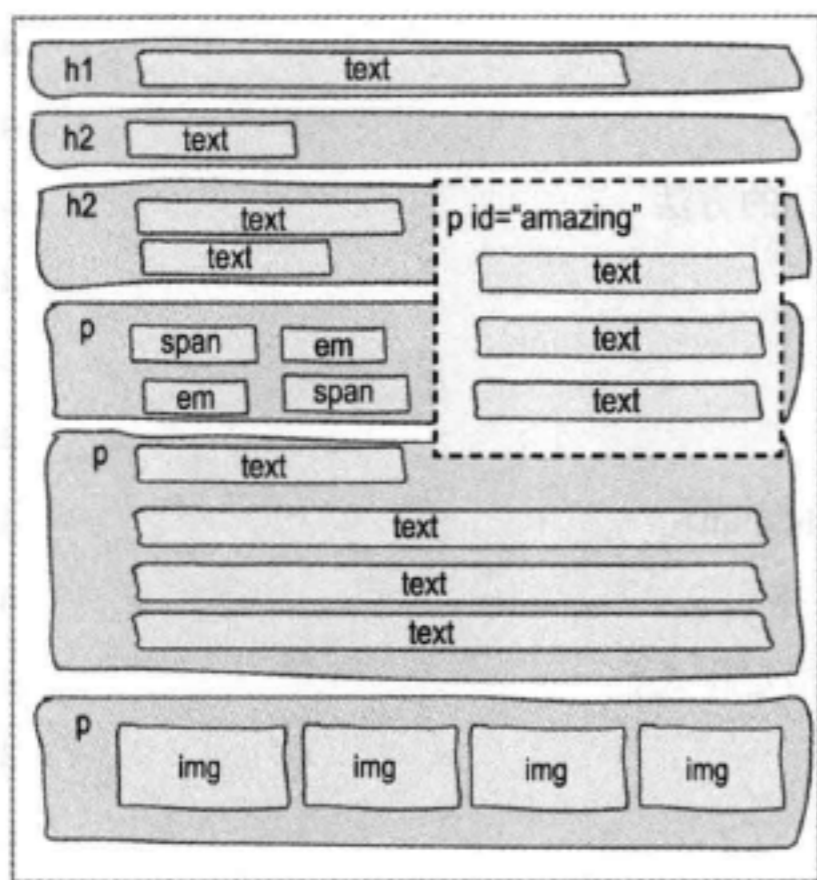
仔细观察饮料HTML	415
下面来研究如何将一个页面划分为逻辑区	417
增加边框	424
为饮料区增加一些真正的样式	425
处理elixir宽度	426
为elixirs增加基本样式	431
我们需要一种选择子孙的方法	437
改变饮料标题的颜色	439
修正行高	440
来点快捷方式	442
只需简单的3步来增加 <code></code>	448
<code><a></code> 元素和它的多重人格	452
如何根据元素的状态指定样式?	453
运用伪类	455
是不是该谈谈“层叠”了?	457
层叠	459
欢迎参加“我有多特定?”游戏	460
综合在一起	461
练习答案	467



布局与定位

摆放元素

让你的HTML元素学点新技巧。我们不再只是让那些HTML元素原地不动，它们该起来了，来帮我们创建一些有真正布局的页面。怎么做呢？嗯，你已经对<div>和结构元素相当熟悉，而且也知道了盒模型是如何工作的，是吧？现在就该充分利用这些知识来完成一些真正的设计。不，我们并不会大谈更多的背景和字体颜色，我们说的是使用多栏布局的那种成熟、专业的设计。在这一章中你可以把之前学到的知识汇集起来加以应用。



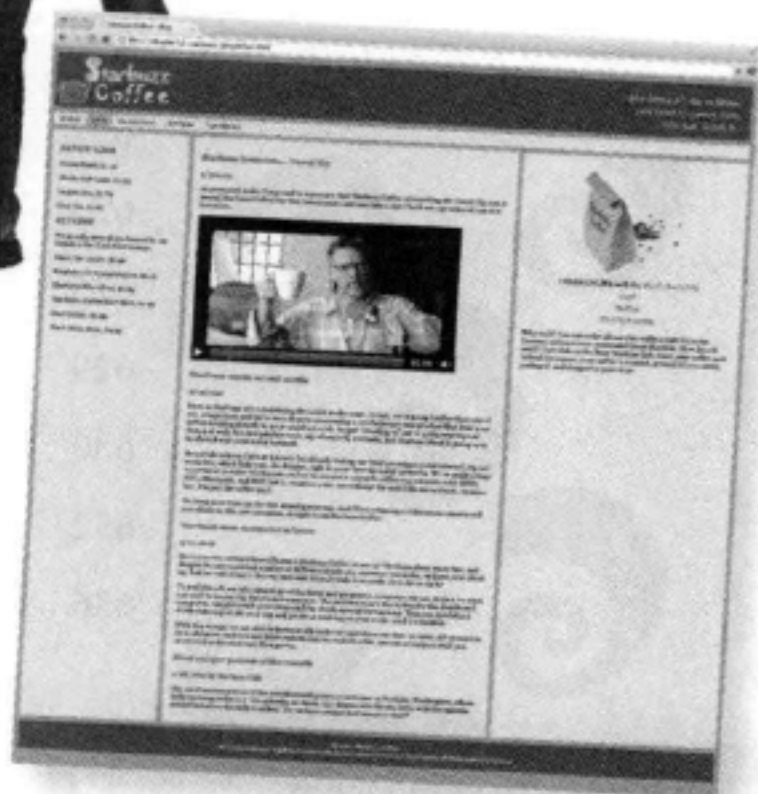
你做超级智力题了吗?	472
使用流	473
内联元素呢?	475
如何集成?	476
如何浮动元素	479
新的Starbuzz页面	483
把边栏移动到页眉下面	488
修正两栏问题	491
设置主内容区的外边距	492
解决重叠问题	495
右紧左松	498
流体与冻结设计	501
绝对定位如何工作	504
修改Starbuzz CSS	507
CSS表格显示如何工作	511
为表格显示增加HTML结构	513
这些间距是怎么回事?	517
页眉的问题	524
用float修正页眉图像	525
指定奖杯位置	528
固定定位如何工作?	531
使用负的left属性值	533
练习答案	539

12

HTML5 标记

现代HTML

你肯定听到过关于HTML5铺天盖地的宣传。另外，这本书读到这里已经篇幅不短，你可能在怀疑是不是书买错了。现在要明确重要的一点，目前为止你在这本书里学到的所有东西都是HTML，更确切地讲，这些都满足HTML5标准。不过，HTML5标准对HTML标记有一些补充，这些方面我们还没有谈到，这一章就来介绍这些内容。HTML5新增的这些特性中，大多都是演进发展而来，你会发现，如果努力完成了这本书中之前的所有工作，掌握这些新内容对你来说也会很轻松。当然也有一些创造性的新特性（比如视频），这些也会在这一章中讲到。下面我们就开始吧，好好看一看这些新的内容！



重新考虑HTML结构	546
更新Starbuzz HTML	551
如何为这些新元素更新CSS	554
建立博客页面的CSS	563
还需要为博客增加一个日期……	565
为博客增加<time>元素	566
如何增加更多<header>元素	568
这些首部到底怎么了？	570
页眉的最终测试	571
完成导航	574
谁需要GPS？测试导航	575
哇！看看这里的导航！	577
创建新博客条目	580
灯光，摄像，开拍……	581
<video>元素如何工作？	583
仔细检查video属性……	584
关于视频格式需要知道什么	586
视频格式竞争对手	587
如何处理所有这些格式……	589
如何更具体地指定视频格式	590
练习答案	597

13

表格与更多列表

建立表格

如果走路也像表格，说话也像表格……在生活中，总会有一些时候必须处理那些枯燥的表格数据。不论是创建一个页面表示公司去年的库存清单，还是需要为你收藏的玩偶建立一个编目表（放心，我们不会说出去的），你很清楚需要在HTML中建立这些表格，不过怎么做呢？嗯，你拣到便宜了。下单吧，只用一章，我们就能揭开表格的秘密，帮助你把自己的数据正确地放在HTML表格中。还不只这些，我们还会为每张订单提供我们的独家指南，指导你为HTML表格指定样式。另外，如果你现在行动，作为特殊奖励，我们提供指南教你指定HTML列表的样式。别犹豫了，快打电话吧！

如何用HTML建立表格?	603
用HTML创建一个表格	604
浏览器创建了什么?	605
表格剖析	606
增加一个标题	609
开始指定样式之前，先把表格放在Tony的页面上	611
折叠边框	616
来点颜色怎么样?	618
Tony有一个有趣的发现	620
再来看Tony的表格	621
如何让单元格跨多行	622
测试表格	624
天堂也有麻烦?	625
覆盖嵌套表格表头的CSS	629
对Tony网站的最后润色	630
如果需要一个定制标记呢?	632
练习答案	636



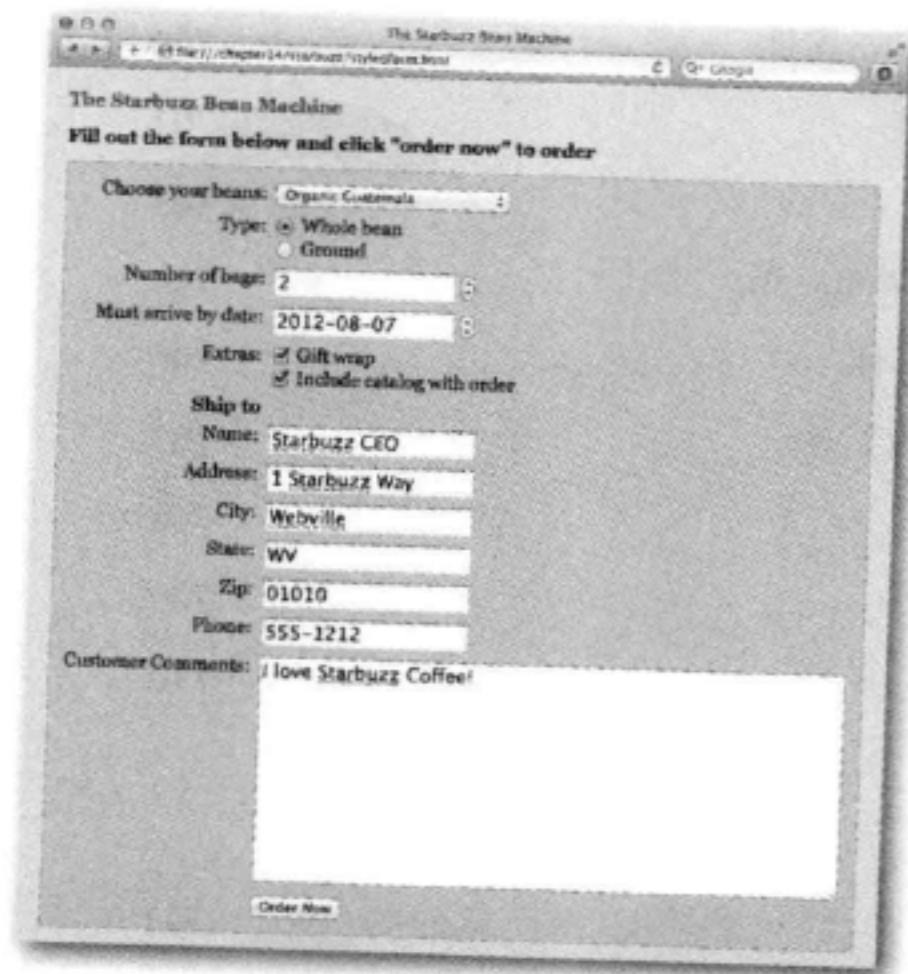
City	Date	Temperature	Altitude	Population	Giver Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5
Bountiful, UT	July 10th	91	4,226 ft	41,173	4/5
Last Chance, CO	July 23rd	102	4,780 ft	265	3/5
	August 9th	93			5/5
Truth or Consequences, NM	August 27th	98	4,242 ft	7,289	Tess 5/5
					Tony 4/5
Why, AZ	August 18th	104	860 ft	480	3/5

14

HTML表单

实现交互

到目前为止，你的所有Web通信都是单向的：只是从页面到访问者。嘿，如果访问者能有反馈就好了！这里就要引入HTML表单，一旦用表单来“武装”你的页面（当然还需要Web服务器的一点帮助），你的页面就能收集客户的反馈，得到网上订单，掌握在线游戏中的下一步，或者可以收集“hot or not”（热门与冷门）竞赛的选票。你将在这一章认识大量HTML元素，它们可以一起协作共同创建Web表单。你还会了解服务器在后台如何支持表单。我们甚至还会介绍如何建立表单的样式。



表单如何工作	646
你写的HTML代码	648
浏览器创建的页面	649
<form>元素如何工作	650
准备建立Bean Machine表单	660
增加<form>元素	661
表单元素名如何工作	662
将这些<input>元素放在HTML中	664
为表单增加更多输入元素	665
增加<select>元素	666
允许客户选择全豆咖啡还是研磨咖啡	668
试试单选按钮	669
使用更多输入类型	670
增加数字和日期输入类型	671
完成表单	672
增加复选框和文本区	673
GET的实际使用	679
将表单元素放入HTML结构实现表格显示布局	684
用CSS建立表单样式	686
关于可访问性	688
表单中还可以有哪些元素?	689
练习答案	693

15 附录：其他

十大主题（我们没有谈到的）

我们已经介绍了很多，这本书也快要结束了。我们会想你的，不过在你离开之前，还要再做一点准备，否则我们实在不放心让你贸然进入这个纷乱的世界。我们不可能把你需要知道的一切都放在这样短短的一章里。实际上，我们原先确实加入了有关HTML和CSS需要了解的所有内容（其他章节中没有谈到），只不过把字体缩小到0.00004。这样才能放得下，可惜没有人能看得清。所以，我们最后还是删去了大部分内容，只把最重要的部分保留在这个“十大主题”附录中。



#1 更多CSS选择器	698
#2 开发商特定的CSS属性	700
#3 CSS变换和过渡	701
#4 交互性	703
#5 HTML5 API和Web应用	704
#6 再来谈谈Web字体	706
#7 创建Web页面的工具	707
#8 XHTML5	708
#9 服务器端脚本	709
#10 音频	710

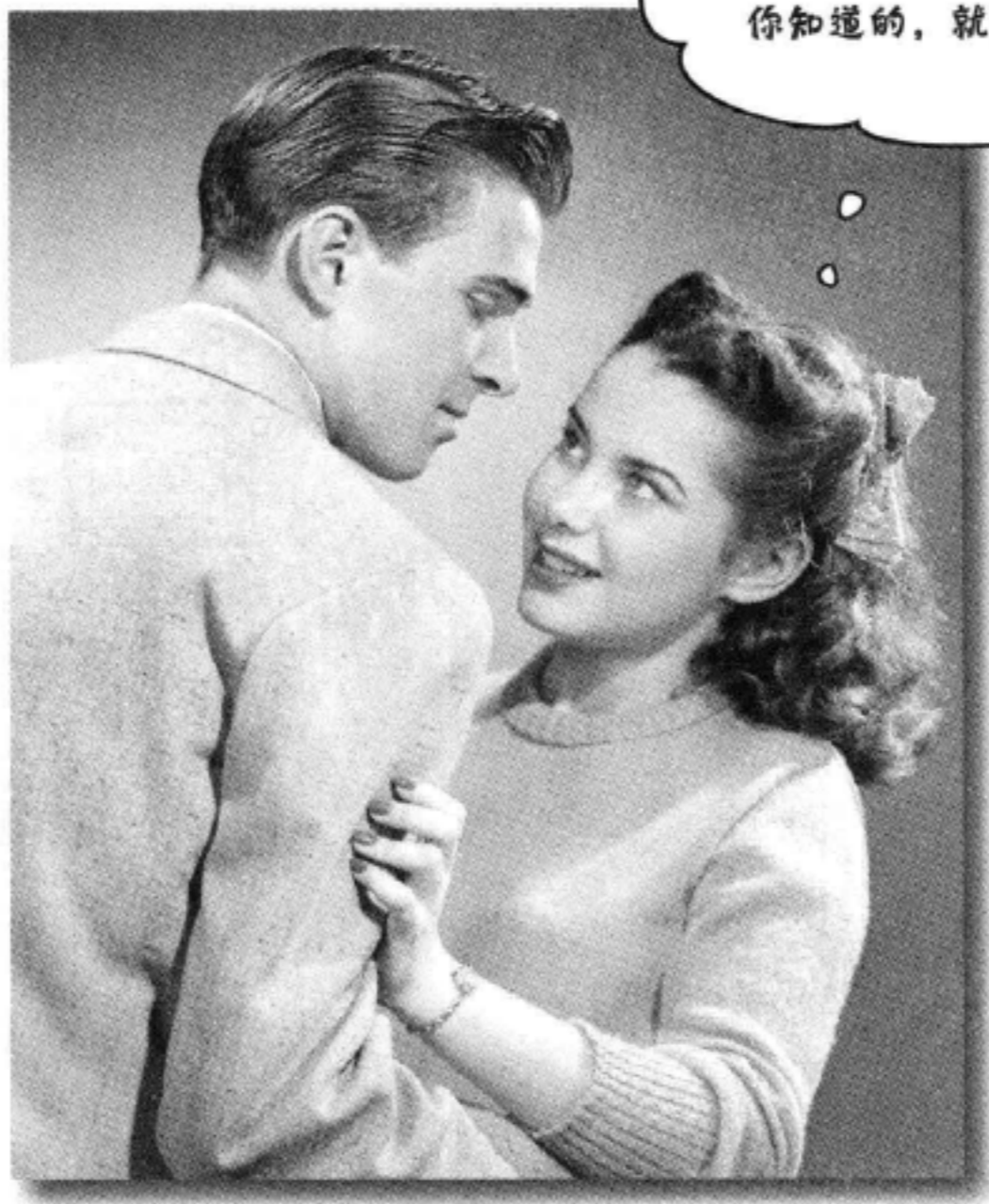
i 索引

711

1 认识HTML

Web语言

别那么着急……要认识我，你得先会讲通用语言。你知道的，就是HTML和CSS。



要进入Web世界，在你面前只有一个障碍，就是要学会它的语言。

超文本标记语言，或者简称为HTML。准备好了吗？我们要上几节语言课了。读完这一章之后，你不仅会了解一些基本的HTML元素，还能讲出带一点样式的HTML。说真的，等学完这本书，你就会像是从小在Web镇长大的一样，能自如地运用HTML。

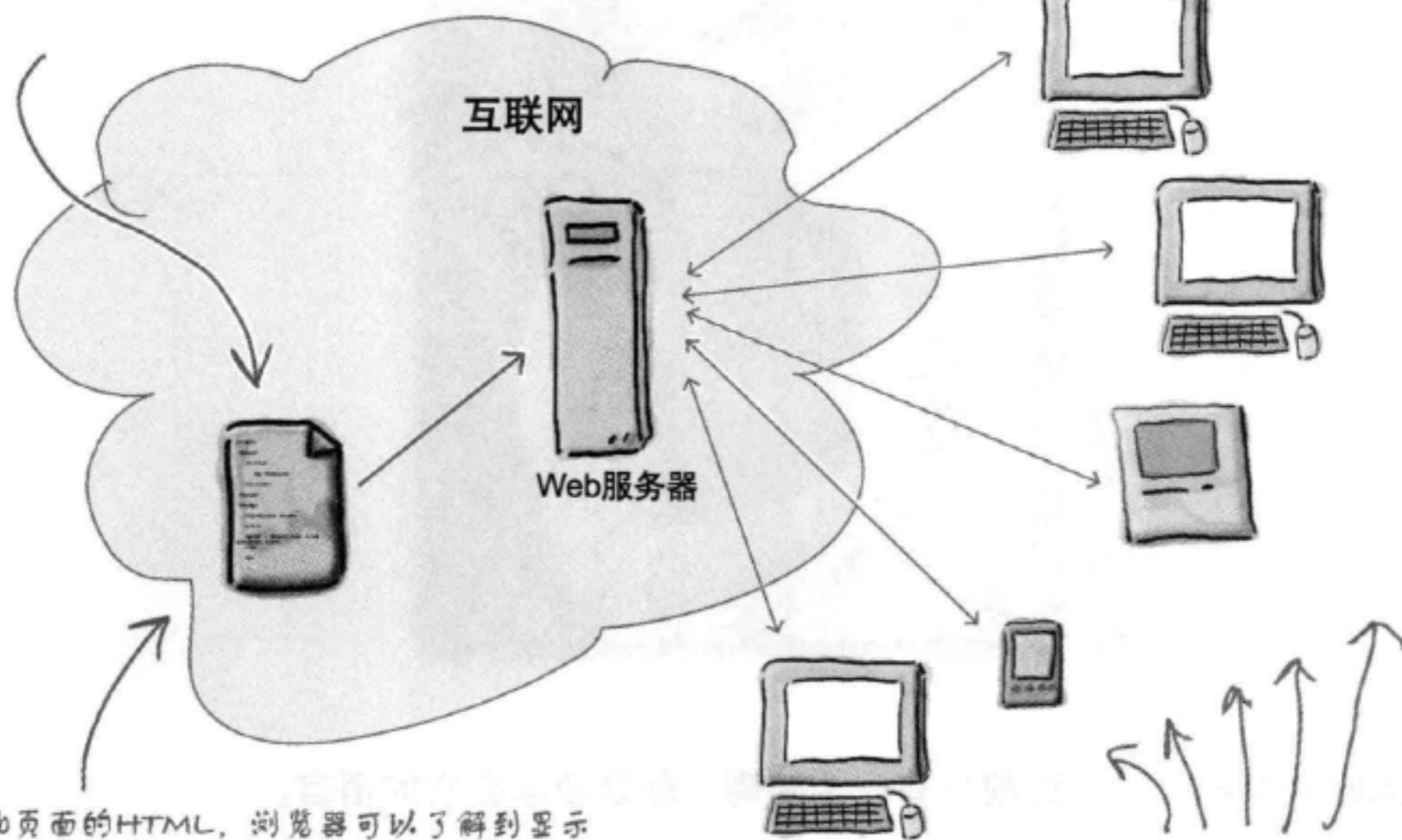
Web ~~电视~~让广播明星黯然失色

想表达某种想法？想出售某种商品？或者是不是需要一个有创意的销售渠道？进入Web世界吧，也许不用我们说你已经知道，Web早已经成为全球通用的交流方式。更棒的是，你能参与其中。

不过，如果你真的希望有效地使用Web，就必须对HTML有所了解，当然，还要对Web如何工作略知一二。下面先从宏观的角度来看看：

要建立Web页面，需要创建用超文本标记语言（HyperText Markup Language，简称为HTML）编写的文件，把它们放在一个Web服务器上（本书后面会介绍如何把你的文件放在一个服务器上。）

一旦把文件放在Web服务器上，任何浏览器就能通过互联网得到你的Web页面。

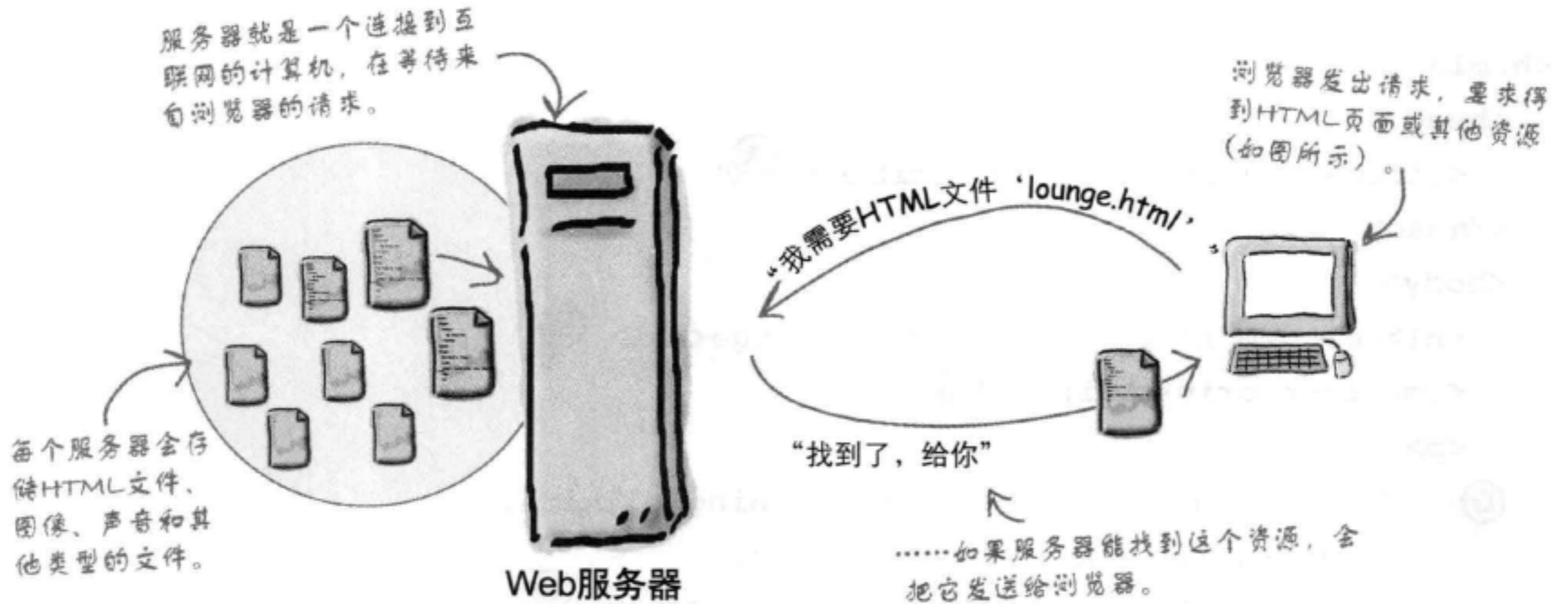


根据Web页面的HTML，浏览器可以了解到显示页面所需的全部信息。另外，如果编写得足够好，你的网页甚至在手机和移动设备上就能很好地显示，还允许有视觉缺陷的人在语音浏览器和屏幕放大器上查看。

大量PC和其他设备已经连接到互联网，它们都运行着Web浏览器。更重要的是，我们的朋友、家人、粉丝和潜在用户可能正在使用这些设备！

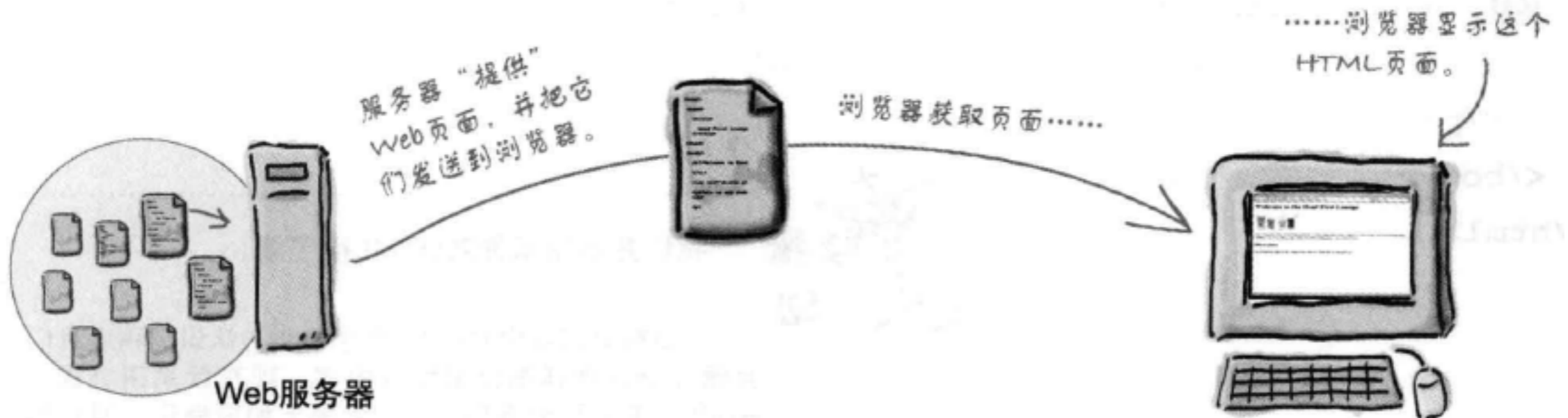
Web服务器能做什么？

Web服务器在互联网上有一份全天候的工作，夜以继日、不知疲倦地等待来自Web浏览器的请求。什么类型的请求？可能是请求Web页面、图像或声音，或者可能甚至是一个视频。服务器收到这些资源请求时，会查找所请求的资源，然后把找到的资源发回给浏览器。



Web浏览器能做什么？

你已经知道浏览器是如何工作的：你在网上冲浪，单击一个链接来访问某个页面。这个单击会导致浏览器向Web服务器请求一个HTML页面，获取这个页面，并在你的浏览器窗口中显示。



但是浏览器怎么知道如何显示一个页面呢？这里就要用到HTML了。HTML会告诉浏览器页面的所有内容和结构。下面来看这是如何做到的.....

你写的代码 (HTML)

如此说来,你已经知道HTML是浏览器显示页面的关键,但是HTML到底是什么样呢?它会做些什么?

下面来看一些HTML……假设你要创建一个Web页面,为Head First休闲室(Head First Lounge)做个宣传,这是当地大家常去的一个休闲场所,有美妙的音乐、清爽的饮料,还提供无线上网。下面是你编写的HTML代码:

```
<html>
  <head>
    <title>Head First Lounge</title> (A)
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1> (B)
     (C)
    <p>
(D) Join us any evening for refreshing elixirs,
      conversation and maybe a game or
      two of <em>Dance Dance Revolution</em>. (E)
      Wireless access is always provided;
      BYOWS (Bring your own Web server).
    </p>
    <h2>Directions</h2> (F)
    <p>
(G) You'll find us right in the center of
      downtown Webville. Come join us!
    </p>
  </body>
</html>
```



我们并不指望你对HTML很了解。

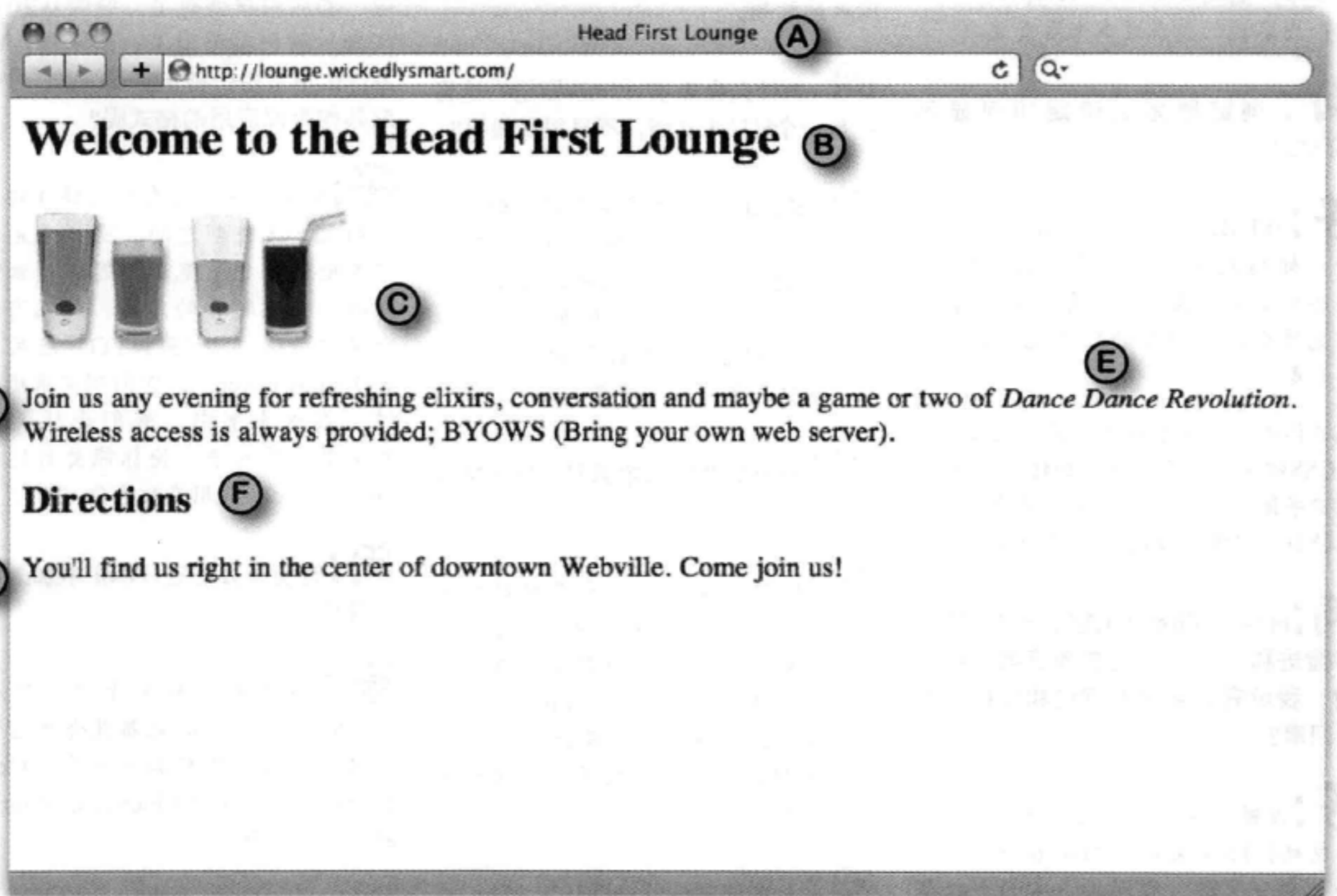
目前只需要你对HTML有个初步的认识,稍后我们会循序渐进地详细介绍所有内容。现在就来研究这个HTML,下一页会看到它在浏览器上如何表示。要特别注意每个字母标注,明确它们在浏览器中如何显示以及在哪里显示。

浏览器创建的页面

浏览器读到你的HTML时，它会翻译文本中的所有标记。标记就是尖括号括起来的词或字符，例如<head>、<p>、<h1>等。标记会告诉浏览器文本的结构和含义。所以并不是交给浏览器一大堆的文本，利用HTML，你可以用标记告诉浏览器哪些文本是标题，哪些文本是段落，哪些文本需要强调，或者图像需要放在什么位置。

下面来看浏览器如何翻译Head First休闲室中的标记：

注意HTML中的各个标记如何对应浏览器显示的内容。



there are no Dumb Questions

问:这么说, HTML就是一堆可以用来包围文本的标记, 是吗?

答:对初学者来说是这样的。要记住, HTML (HyperText Markup Language) 是超文本标记语言的缩写, 所以HTML提供了一种用标记“标示”文本的方法, 来告诉浏览器文本的结构是怎样的。不过, HTML还有另一个方面, 也就是超文本 (HyperText), 这个内容我们会在这本书后面谈到。

问:浏览器怎么确定如何显示HTML?

答:HTML会告诉浏览器文档的结构: 标题放在哪里, 段落放在哪里, 哪些文本需要强调等。有了这些信息, 浏览器会按照内置的默认规则显示各个元素。

不过你不必受制于默认设置。完全可以用CSS增加你自己的样式和格式化规则, 确定字体、颜色、大小和页面的很多其他特征。本章后面还会介绍CSS。

问:Head First休闲室的HTML有很多缩进和空格, 不过在浏览器上显示时, 我没有看到这些缩进和空格, 怎么回事?

答:没错, 这个问题问得好。浏览器会忽略HTML文档中的制表符、回车和大部分空格。实际上, 它们会根据你的标记来确定在哪里换行或分段。

那么, 既然浏览器会忽略这些空白符, 我们为什么还要插入自己的格式呢? 这是为了帮助我们在编辑HTML时能够更容易地读懂文档。随着HTML文档变得越来越复杂, 你会发现, 在某

些地方加一些空格、回车和制表符, 这会对提高HTML可读性很有帮助。

问:这么说, 共有两级标题, <h1>和子标题<h2>, 是吗?

答:实际上浏览器通常显示的标题一共有6级, 从<h1>到<h6>, 字体由大到小。除非你要创建一个非常复杂、庞大的文档, 否则一般不会用到<h3>以后的标题。

问:为什么会需要<html>标记? 这肯定是一个HTML文档, 不是很明显吗?

答:<html>标记告诉浏览器你的文档确实是一个HTML文件。如果没有加这个标记, 有些浏览器可以接受, 但有一些是不允许的, 本书后面介绍到“工业级强度HTML”时, 你会看到包含这个标记相当重要。

问:一个文件怎么才算是HTML文件呢?

答:HTML文件是一个简单的文本文件。与字处理文件不同, 其中没有嵌入特殊的格式。按照约定, 我们会在文件名末尾加一个“.html”, 让操作系统更清楚这个文件是什么。不过, 可以看到, 真正重要的是你在文件里放什么内容。

问:大家都在谈论HTML5。我们也会用HTML5吗? 如果是这样, 那么为什么不直接说“HTML-FIVE”, 而不是“HTML”呢?

答:你现在学习的是HTML, 实际上HTML5就是HTML的最新版本。HTML5最近很受关注, 这是因为它让

我们编写HTML的很多做法大大简化, 而且还提供了一些新的功能, 这些都会在这本书中介绍。另外, 通过它的JavaScript应用编程接口 (Application Programming Interfaces, API), 它还提供了一些高级特性。这些在《Head First HTML5 Programming》一书有详细介绍。

问:标记看起来很像, “所见即所得”的应用早就有了, 好像从20世纪70年代就已经出现了, 是不是? Web为什么不采用一种像Microsoft Word或其他类似应用的格式呢?


答:Web是基于没有任何特殊格式字符的文本文件创建的。这样才允许世界各地的各种浏览器都能够获取Web页面, 并理解它的内容。现在有很多所见即所得 (WYSIWYG) 应用 (比如Dreamweaver), 它们确实很棒。不过, 在这本书中, 我们会从最原始的文本文件入手, 使你能更好地理解Dreamweaver应用在后台做了什么。

问:有没有办法在HTML中加入自己的注释?

答:当然有, 如果你把注释放在<!--和-->之间, 浏览器就会把它们完全忽略。假设你想加入一个“Here's the beginning of the lounge content”注释, 可以这样写:

```
<!-- Here's the beginning  
of the lounge content -->
```

注意, 注释可以写为多行。要记住, “<!--”和“-->”之间放置的所有内容 (甚至HTML) 都会被浏览器忽略。



Sharpen your pencil

也许你没想到，实际你对HTML并不陌生……

下面再给出Head First休闲室的HTML。研究下面这些标记，看你能不能猜出它们会告诉浏览器关于内容的哪些信息。把你的答案写在右边的空行上，我们已经帮你填上了前两个。

```
<html>
```

```
  <head>
```

```
    <title>Head First Lounge</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Welcome to the Head First Lounge</h1>
```

```
    
```

```
    <p>
```

```
      Join us any evening for refreshing elixirs,  
      conversation and maybe a game or  
      two of <em>Dance Dance Revolution</em>.
```

```
      Wireless access is always provided;
```

```
      BYOWS (Bring your own Web server).
```

```
    </p>
```

```
    <h2>Directions</h2>
```

```
    <p>
```

```
      You'll find us right in the center of  
      downtown Webville. Come join us!
```

```
    </p>
```

```
  </body>
```

```
</html>
```

告诉浏览器HTML从这里开始。

.....
页面“head”开始（稍后介绍更多有关内容）。

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Sharpen your pencil
答案

```

<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing elixirs,
      conversation and maybe a game or
      two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided;
      BYOWS (Bring your own Web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of
      downtown Webville. Come join us!
    </p>
  </body>
</html>

```

告诉浏览器HTML从这里开始。

 页面“head”开始(稍后介绍更多有关内容)。

 为页面指定一个标题。

 head结束。

 页面主体(body)开始。

 告诉浏览器“Welcome to……”是一个标题。

 在这里放置图像“drinks.gif”。

 开始一个段落。

 强调Dance Dance Revolution。

 段落结束。

 告诉浏览器“Directions”是一个子标题。

 开始另一个段落。

 段落结束。

 页面主体结束。

 告诉浏览器HTML在这里结束。



你在Starbuzz咖啡馆交好运了

Starbuzz之所以名气很大，是因为它是发展最迅速的连锁咖啡店，随处都可以看到Starbuzz连锁店。也许你附近的街角有一家，街对面又能看到另一家。

实际上，Starbuzz发展得太快了，他们甚至没时间来建立自己的Web网站……你的好机会来了。很凑巧，你在Starbuzz买Chai Tea（印度香辣奶茶）时，刚好碰到了Starbuzz的CEO……

听说你懂点HTML。我们恰好需要建一个Web页面来推广我们Starbuzz的饮品。你能帮忙吗？

Starbuzz的CEO



BRAIN POWER

做出你的决定，
选出你最优先的事情（只能单选）：

- A. 给狗狗洗个澡。
- B. 实现收支平衡。
- C. 抓住这个机会，
从Starbuzz开启你崭新的
Web职业生涯。
- D. 与牙医约个时间。



太棒了！真高兴你能帮*
我们。这是我们在第一个
页面上发布的内容……

CEO在餐巾纸上潦草
地写了些东西，然后把
它递给你……



感谢你能助我们一臂之力！
我们只需要这个web页面显示
一些简单的内容（请看下面），
包括饮料名字、价格和介绍。

House Blend, \$1.49
A smooth, mild blend of coffees from
Mexico, Bolivia and Guatemala.

Mocha Cafe Latte, \$2.35
Espresso, steamed milk and chocolate
syrup.

Cappuccino, \$1.89
A mixture of espresso, steamed milk
and foam.

Chai Tea, \$1.85
A spicy drink made with black tea,
spices, milk and honey.

Sharpen your pencil



来看看餐巾纸上写了什么。你能确定它的大致结构吗？换句话说，有没有明显的标题？段落？有没有少点什么，比如页面标题？

在餐巾纸上加些标记（用铅笔），标出你看到的结构，再增加你认为缺少的内容。

答案可以在第1章最后找到。

* 如果你在上一页选择了A、B或者D，建议你在这本书捐给一个好一点的图书馆，或者冬天时拿来生火，也可以直接到Amazon上当二手书卖掉，换点钱回来。

创建Starbuzz web页面

当然，现在唯一的问题是你还没有具体创建过任何Web页面。不过，这正是你决定深入学习HTML的原因，不是吗？

别担心，以下就是后面几页要做的工作：

- ❶ 用你喜欢的文本编辑器创建一个HTML文件。
- ❷ 录入Starbuzz CEO在餐巾纸上写的菜单。
- ❸ 将文件保存为“index.html”。
- ❹ 在你喜欢的浏览器中打开“index.html”文件，然后等待见证奇迹的时刻吧。

别有压力，不过等你完成这个Web页面后，可能会有数以千计的人访问这个页面。所以它不仅要正确，而且看上去还要很漂亮！



创建一个HTML文件(Mac系统)

所有HTML文件都是文本文件。要创建一个文本文件，需要有一个能创建纯文本的应用，而不需要加入大量花哨的格式和特殊字符。你只要纯粹的纯文本。

这本书中，我们会使用Mac系统上的TextEdit。不过，如果你喜欢其他文本编辑器，则完全可以。另外，如果你运行的是Windows系统，则可以跳过这几页，直接看Windows系统的有关说明。

第1步:

导航到你的Applications文件夹。

TextEdit应用在Applications文件夹中。要找到这个文件夹，最容易的办法是从Finder（查找工具）的File菜单选择New Finder Window（新建查找窗口），然后在快捷方式中直接查找Applications（应用）。找到后，单击Applications。

第2步:

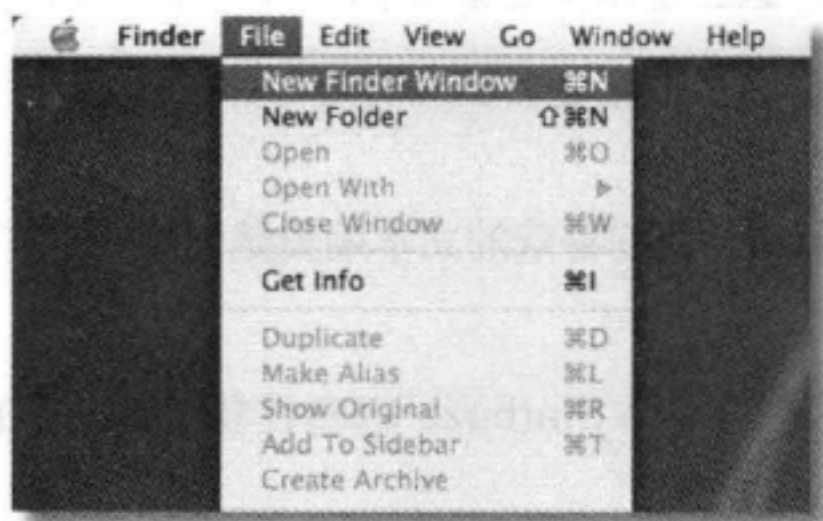
找到并运行TextEdit。

你的Applications文件夹中可能列出了很多应用，可以向下滚动，直到看到TextEdit。双击TextEdit图标，运行这个应用。

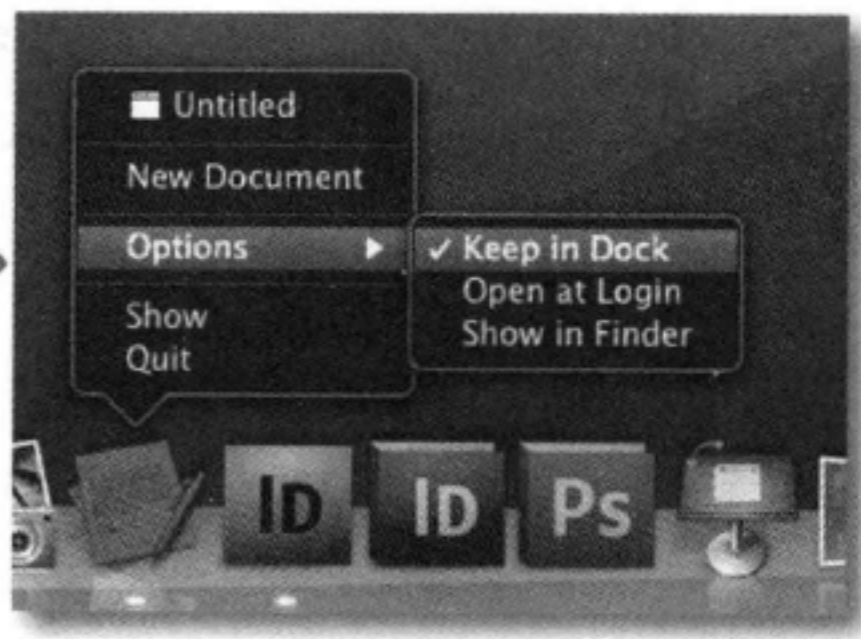
第3步（可选）:

让TextEdit一直在浮动条中。

如果你想更为便捷，则可以单击浮动条中的TextEdit图标（一旦TextEdit应用运行，这个图标就会出现）。显示一个弹出菜单时，选择Options（选项），然后选择“Keep in Dock”（保持在浮动条中）。这样一来，这个TextEdit图标就会一直出现在你的浮动条上，你就不用每次需要用到它时，都从Applications文件夹中查找了。



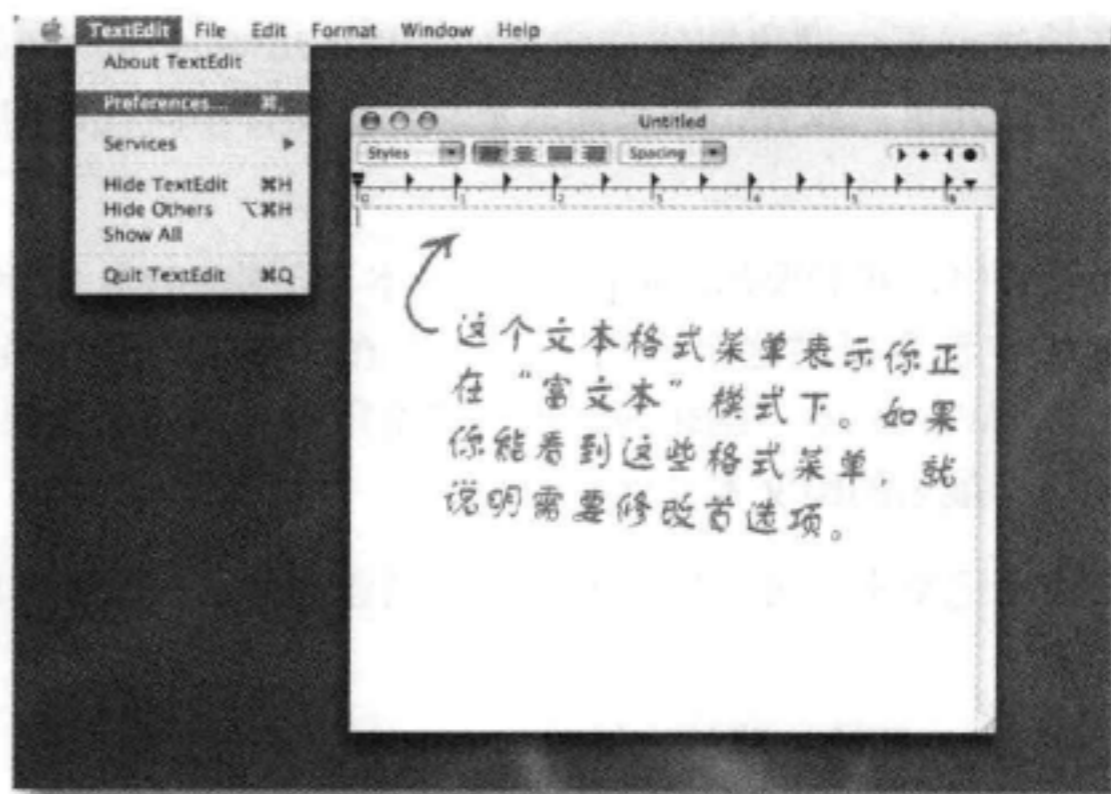
Finder快捷方式。



第4步:

修改TextEdit首选项。

默认地，TextEdit会采用“富文本”模式，这表明当你保存文件时，它会向你的文件增加它自己的格式和特殊字符，这可不是你想要的。所以，你需要修改TextEdit首选项，让TextEdit把文件保存为纯文本文件。为此，首先从TextEdit菜单选择菜单项Preferences（首选项）。

**第5步:**

将首选项设置为纯文本。

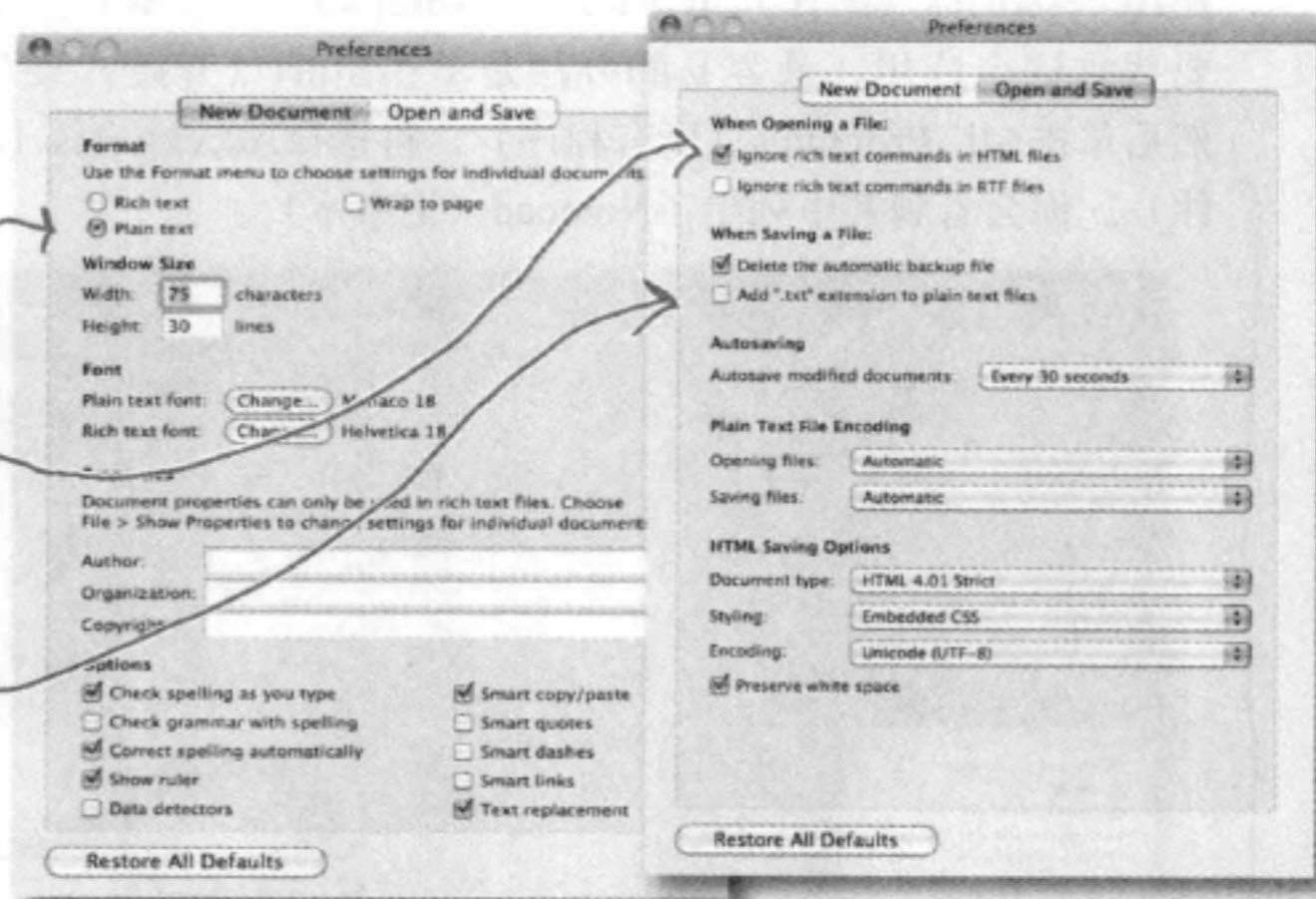
一旦看到首选项（Preferences）对话框，你就要做三件事。

首先，在New Document（新建文档）标签页中选择“Plain text”（纯文本）作为默认的编辑器模式。

然后，在“Open and Save”（打开并保存）标签页中，确保选中“Ignore rich text commands in HTML files”（忽略HTML文件中的富文本命令）。

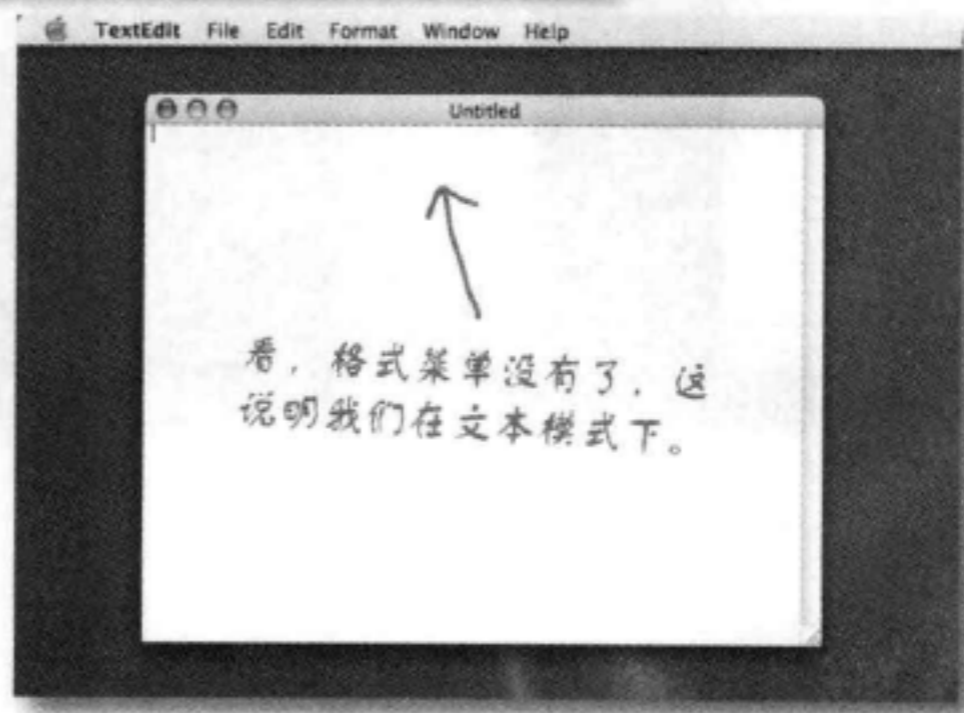
最后，确保未选中“Add .txt extension to plain text files”（为纯文本文件增加.txt扩展名）。

就这么简单，现在单击左上角的红色按钮，关闭这个对话框。

**第6步:**

退出并重启。

现在退出TextEdit，从TextEdit菜单选择Quit（退出），然后重启应用。这一次，你会看到窗口最上面不再有复杂的文本格式菜单。你已经准备好了，现在可以创建HTML了。



创建一个HTML文件（Windows系统）

如果你在读这一页，则说明你肯定是一个Windows 7用户。如果不是，*或者Windows的其他版本。*则可以跳过这几页。或者，如果你只是想坐在后排不问任何问题，那你可以随意。

在Windows 7中，我们要用Notepad（记事本）创建HTML文件，每一个Windows版本都会提供这个应用，价格公道，易于使用。如果你另有自己喜欢的Windows 7编辑器，那也完全可以，只要能确保可以创建带“.html”扩展名的纯文本文件。

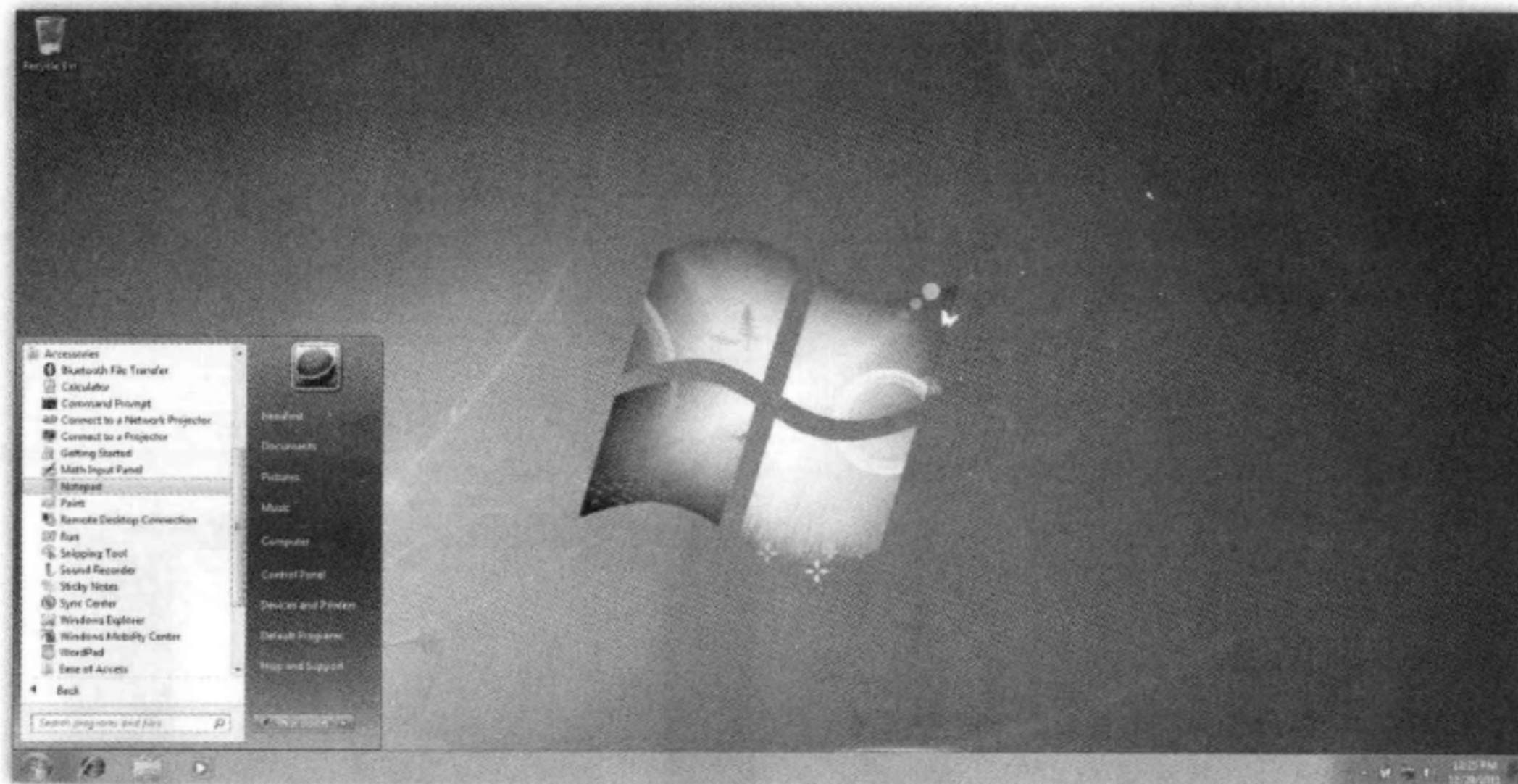
如果你在使用Windows的其他版本，也能找到这个记事本应用。

假设你要使用记事本，可以按下面的步骤创建你的第一个HTML文件。

第1步：

打开Start（开始）菜单，导航到Notepad（记事本）。

在Accessories（附件）里可以找到Notepad（记事本）应用。要找到这个应用，最容易的办法是单击Start（开始）菜单，然后单击All Programs（所有程序），再选择Accessories（附件）。你会看到其中列出了Notepad（记事本）。



第2步:

打开记事本。

一旦在Accessories（附件）文件夹找到记事本，单击这个图标，你就会看到一个空窗口，可以在这个窗口中输入HTML了。



不过建议最好完成这一步。

第3步: (可选)

不要隐藏已知文件类型的扩展名。

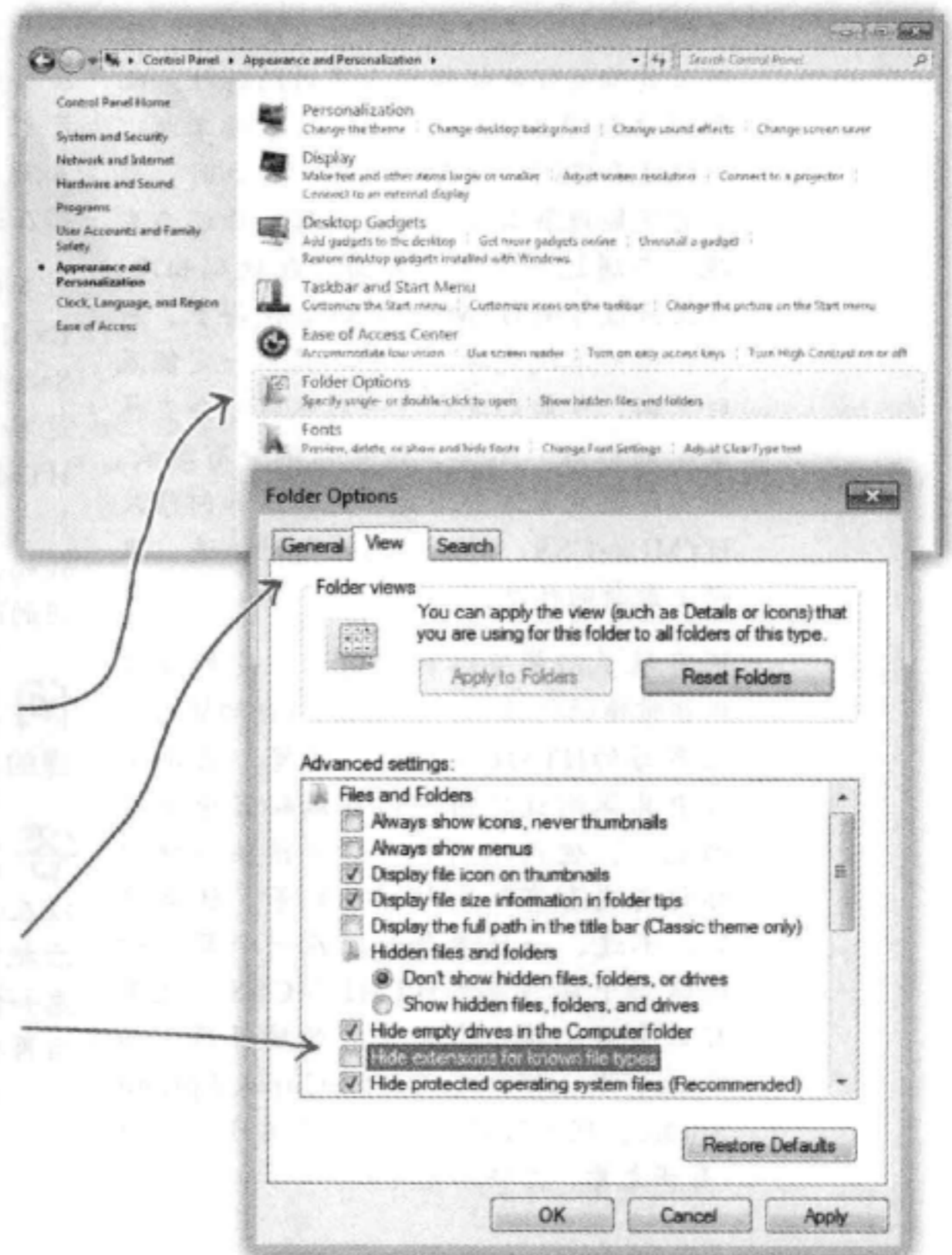
默认地，Windows File Explorer（文件资源管理器）会隐藏已知文件类型的文件扩展名。例如，一个名为“Irule.html”的文件在资源管理器中可能显示为“Irule”，而没有“.html”扩展名。

这可能会带来困惑，如果Windows显示这些扩展名，就不会那么让人糊涂了，所以下面改变你的文件夹选项，使文件扩展名可见。

首先，打开Folder Options（文件夹选项），单击Start（开始）按钮，再单击Control Panel（控制面板），接下来单击“Appearance and Personalization”（外观和个性化），然后单击Folder Options（文件夹选项）。

接下来，在View（查看）标签页中，在“Advanced settings”（高级设置）中向下滚动，直到看到“Hide extensions for known file types”（隐藏已知文件类型的扩展名），取消选中这个选项。

就这么简单。单击OK按钮保存首选项，现在就能在资源管理器中看到文件扩展名了。



there are no Dumb Questions

问：为什么我要使用简单的文本编辑器？难道不能用那些强大的工具创建Web页面吗？比如Dreamweaver和Expression Web？

答：你为什么读这本书？是因为你想了解创建Web页面使用的具体技术，对不对？尽管那些工具确实都非常棒，但是它们会越俎代庖，很多事情都帮你做，尽管你省事了，但是学不到多少东西。在你真正掌握了HTML和CSS之前，应该先好好学学这些内容，而不要完全依赖一个“能干的”工具。

不过，一旦你已经成为这方面的行家，这些工具确实就能提供很多不错的特性，比如语法检查和预览。如果你已经掌握了HTML和CSS，查看“代码”窗口时，则你肯定能理解其中的所有内容，你还会发现，与通过一个用户界面修改代码相比，直接修改原始HTML和CSS会快得多。另外，当标准改变时，那些工具不一定能及时更新，可能到发布下一版本时才会支持最新的标准。不过，这对你不会有影响，因为即使没有这个工具你也知道如何修改HTML和CSS，所以你总能与时俱进，跟得上最新的标准。

还有很多功能完备的编辑器，它们能提供非常棒的特性，如剪贴板（自动插入你经常写的HTML片段）、预览（在浏览器中具体测试之前可以在编辑器中直接预览）、使用不同颜色区分语法（使得标记与内容有不同的颜色）等，还有很多。不过，一旦学会如何在一个简单的编辑器中编写基本HTML和CSS，就很有必要再试试一个更强大的编辑器，如Coda、TextMate、CoffeeCup或Aptana Studio。现在有很多这样的工具可供选择（有些免费，有些不免费）。

问：我有编辑器了，不过我要用哪一个浏览器呢？有好多浏览器——Internet Explorer、Chrome、Firefox、Opera、Safari，怎么选啊？

答：简单的回答是，可以用你喜欢的任何浏览器。HTML和CSS是行业标准，这说明所有浏览器都会尽量以相同的方式支持HTML和CSS（只是为了得到最好的支持，要确保使用浏览器的最新版本）。

复杂的答案是，事实上，不同浏览器处理页面的方式存在细微的差别。如果你的用户可能用不同的浏览器访问你的页面，一定要在多个不同的浏览器中进行测试。有些页面在各种浏览器中看起来都一样，但有些可能并非如此。对HTML和CSS越深入，这些细微差别对你的影响就越大，我们将在这本书中讨论这些细小的差异。

大多数例子在所有主流浏览器（Internet Explorer、Chrome、Firefox、Opera和Safari）上都能正常工作（除非特别说明）。这些都是现代浏览器，提供了很好的HTML和CSS支持。作为一个Web开发人员，你可能需要在多个浏览器中测试你的代码，所以建议你下载并了解至少两个不同的浏览器！

问：这些文件是在我自己的计算机上创建的，怎么在网上浏览这些页面呢？

答：这是HTML的一个过人之处，你可以在自己的计算机上创建和测试文件，然后把它们发布到Web上。目前，我们只考虑如何创建文件以及文件中有些什么，稍后再介绍如何在Web上发布。

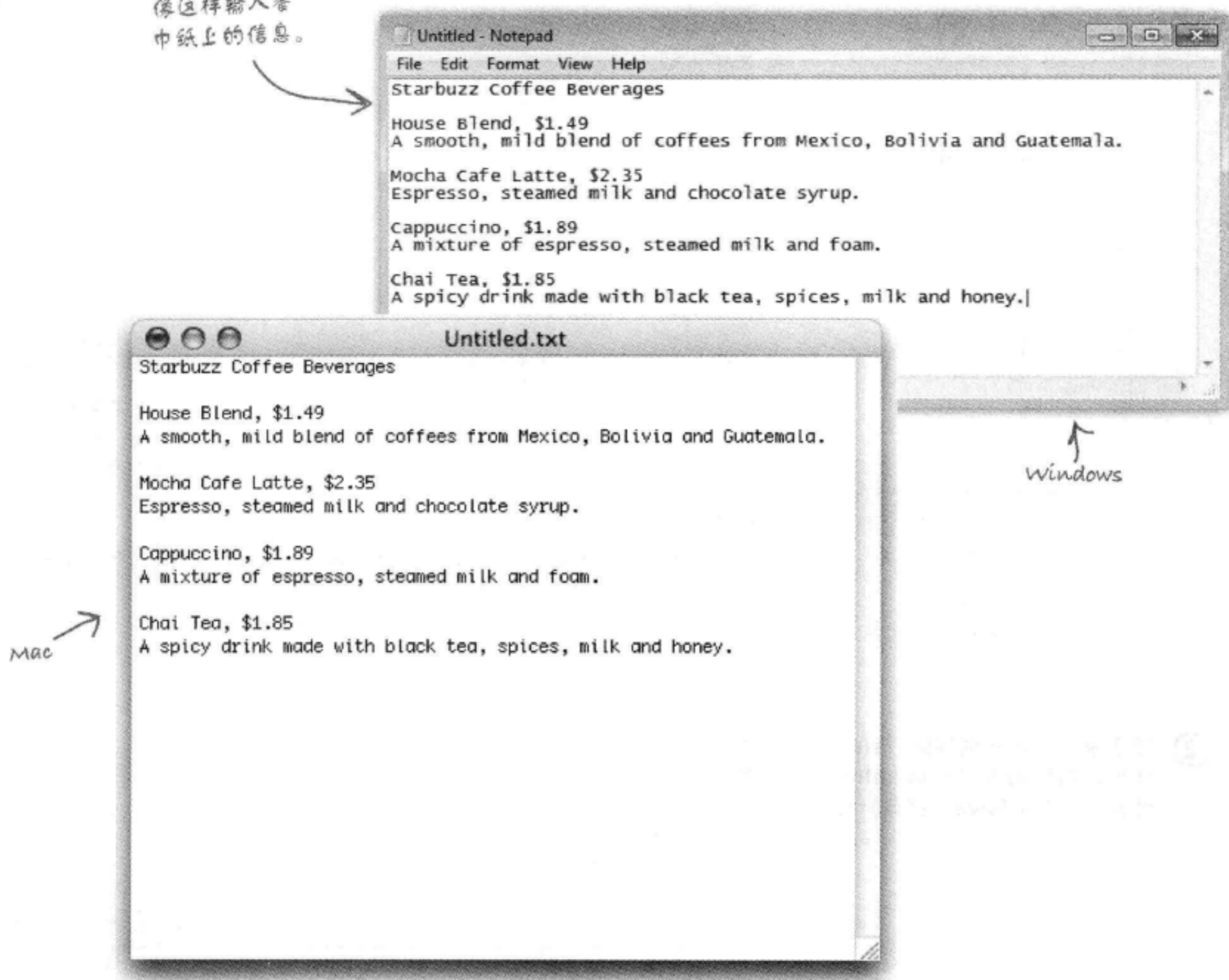


现在，再回到Starbuzz……

好了，你已经知道创建纯文本文件的基础知识，现在只需要在文本编辑器里放一些内容，保存文件，然后加载到你的浏览器里。

首先输入CEO在餐巾纸上写的那些饮料，这些饮料就是页面的内容。你要增加一些HTML标记，让这些内容有点结构，不过现在只需要输入基本内容。准备好了吧，首先在文件最上面增加“Starbuzz Coffee Beverages”。

像这样输入餐巾纸上的信息。

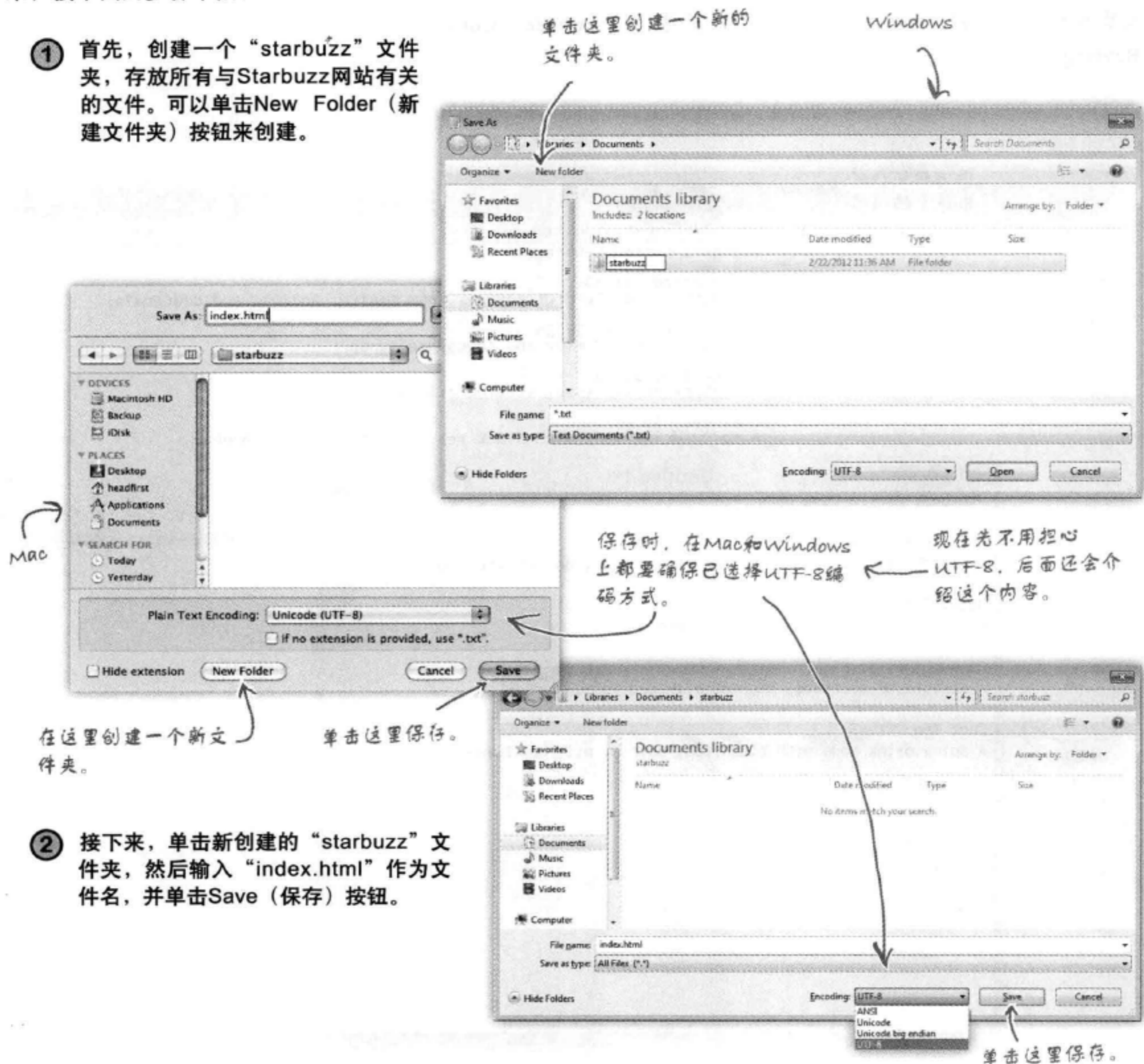


保存你的成果

一旦输入CEO在餐巾纸上写的饮料，接下来就要把你的成果保存到一个名为“index.html”的文件中。保存之前，可能需要创建一个文件夹，名为“starbuzz”，用来存放这个网站的文件。

首先，从File（文件）菜单选择Save（保存），你会看到一个Save As（另存为）对话框。接下来，按下面的步骤来做：

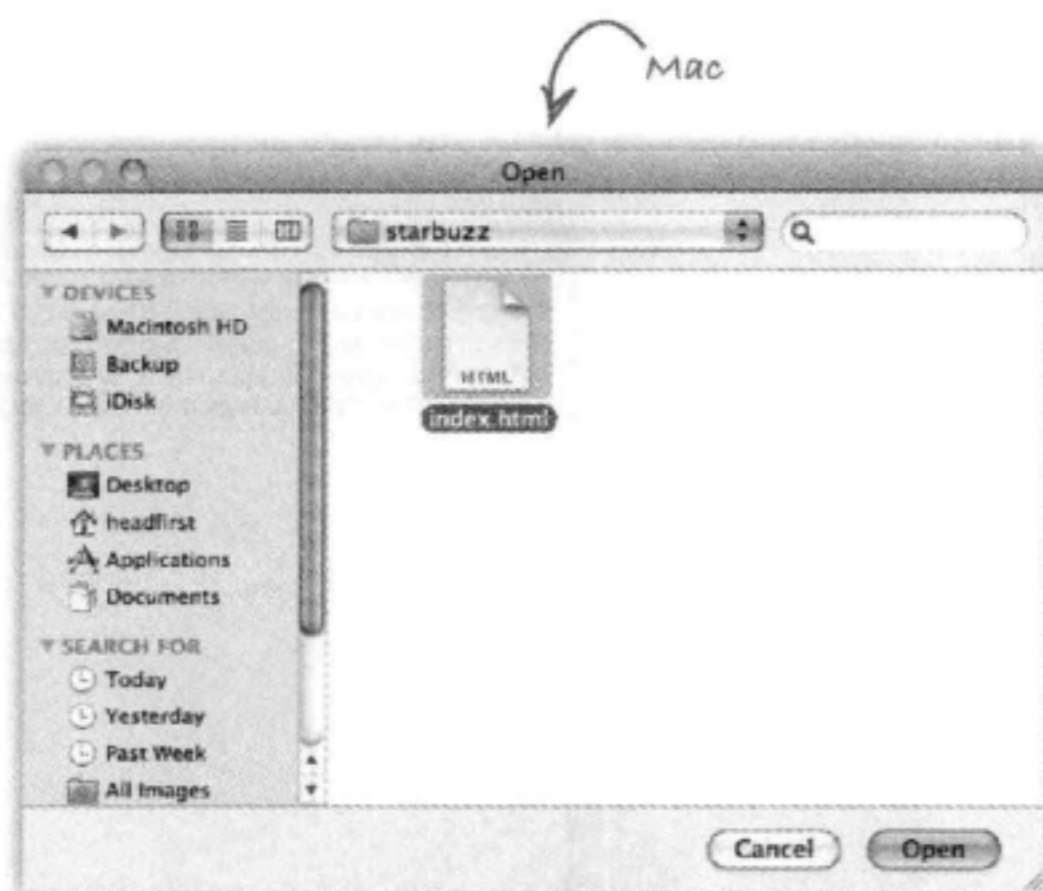
- 首先，创建一个“starbuzz”文件夹，存放所有与Starbuzz网站有关的文件。可以单击New Folder（新建文件夹）按钮来创建。



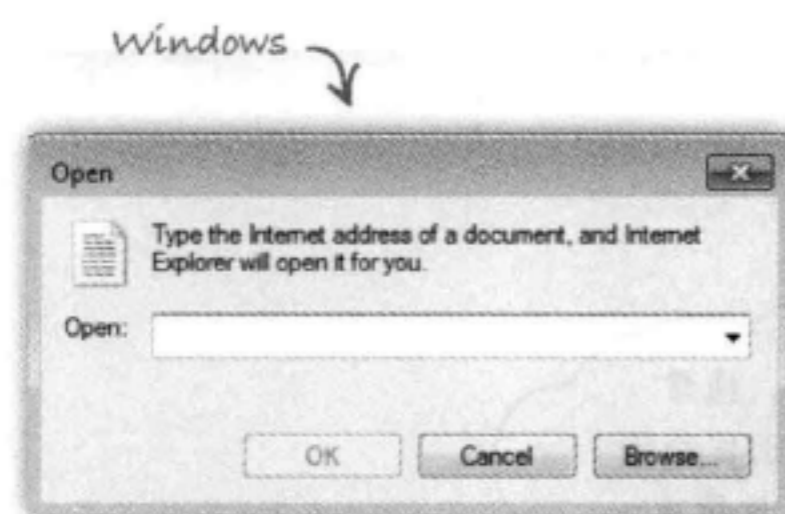
- 接下来，单击新创建的“starbuzz”文件夹，然后输入“index.html”作为文件名，并单击Save（保存）按钮。

在浏览器中打开你的Web页面

是不是准备打开你的第一个Web页面？可以用你喜欢的浏览器，从File（文件）菜单选择“Open File…”（打开文件……），如果使用Windows 7和Internet Explorer，则要选择“Open…”（打开……），导航到你的“index.html”文件。选择这个文件并单击Open（打开）按钮。

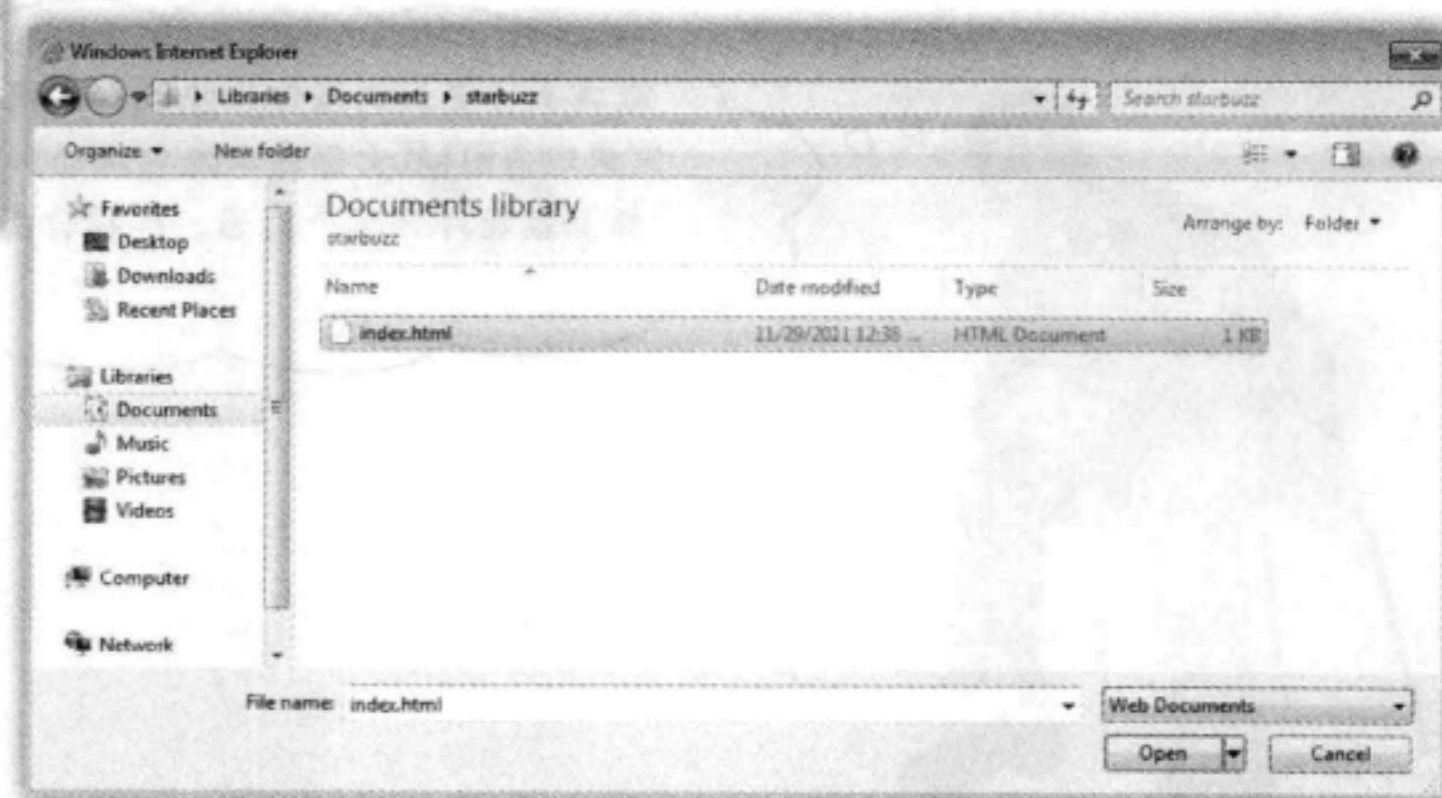


在Mac上，导航到你的文件，然后单击这个文件的图标选中它，再单击Open（打开）按钮。



在Windows Internet Explorer中，这个过程包括两步。首先，打开Open（打开）对话框。

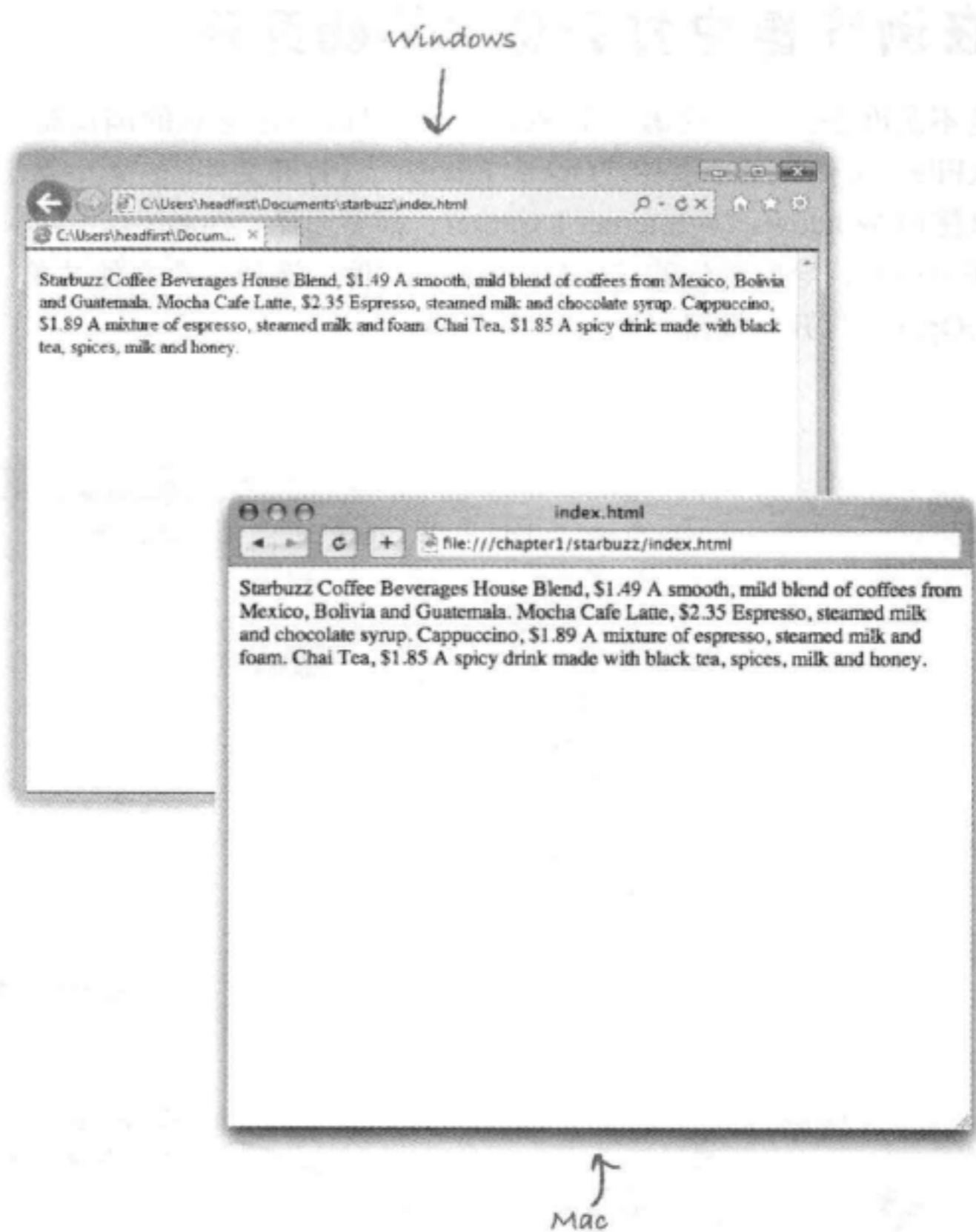
然后单击Browse（浏览）按钮，打开一个浏览对话框，再导航到保存文件的位置。



测试你的页面



成功了！你已经在浏览器中加载了页面，不过看起来有点……嗯……不太让人满意。这是因为，你目前所做的只是做一些最基本的工作，仅仅创建了一个页面，再在浏览器中查看，仅此而已。到目前为止，你只是输入了Web页面的内容。现在该HTML发挥作用了。利用HTML，你可以告诉浏览器页面的结构。什么是结构？你已经看到了，这是一种标记文本的方法，通过这种方法，浏览器就能知道什么是标题，哪些文本要作为段落，另外哪些文本是子标题等。一旦浏览器了解了结构，它就能用一种更有意义、更可读的方式显示你的页面。



取决于你使用的操作系统和浏览器，通常只需双击HTML文件或者把它拖动到浏览器图标上就能打开这个页面。这样会简单得多。



标记磁贴

好吧，现在来增加结构……

你的任务是为Starbuzz餐巾纸上的文本增加结构。使用这一页最下面的冰箱磁贴标记文本，指示哪些部分是标题、子标题和段落文本。我们已经帮你放好了几个磁贴。完成这个任务并不需要用到下面的全部磁贴，有些磁贴可能用不上。

`<h1>` Starbuzz Coffee Beverages `</h1>`

House Blend, \$1.49

A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala.

`<h2>` Mocha Cafe Latte, \$2.35 `</h2>`

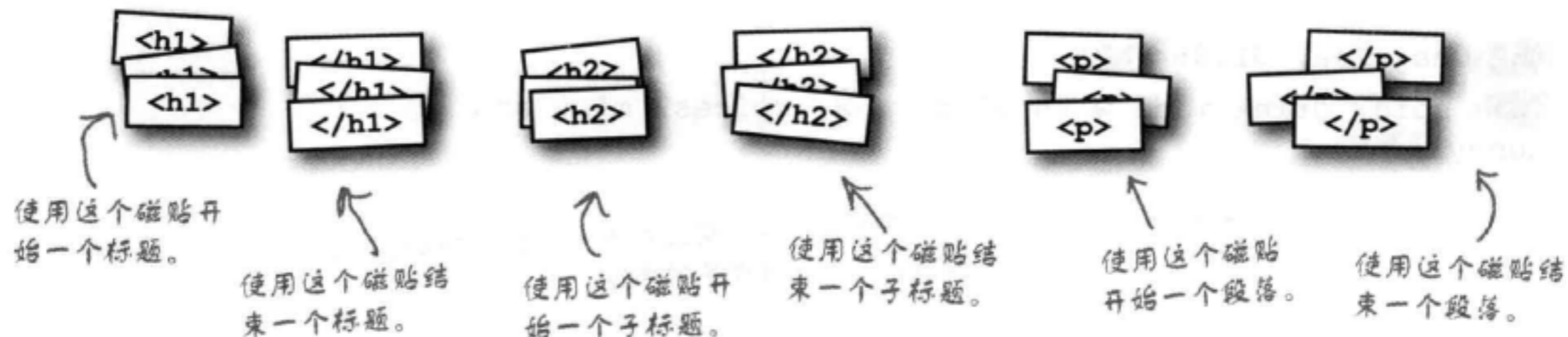
`<p>` Espresso, steamed milk and chocolate syrup. `</p>`

Cappuccino, \$1.89

A mixture of espresso, steamed milk and foam.

Chai Tea, \$1.85

A spicy drink made with black tea, spices, milk and honey.





祝贺你，你已经 写出你的第一个 HTML!

看起来与冰箱磁贴好像差不多，不过你确实已经用HTML对文本做了标记。只不过可以看到，我们通常会把这些磁贴称为标记（tag）。检查下面的标记，与上一页中的磁贴做个比较。

使用<h1>和</h1>标记来标记标题。这两个标记之间的所有文本就是标题的具体内容。

```
<h1>Starbuzz Coffee Beverages</h1>
```

```
<h2>House Blend, $1.49</h2>
```

```
<p>A smooth, mild blend of coffees from Mexico, Bolivia  
and Guatemala.</p>
```

<h2>和</h2>标记包围一个子标题。可以把<h2>标记看作是<h1>标题的子标题。

```
<h2>Mocha Cafe Latte, $2.35</h2>
```

```
<p>Espresso, steamed milk and chocolate syrup.</p>
```

<p>和</p>标记包围一个文本块，这是一个段落。其中可以有一个或多个句子。

```
<h2>Cappuccino, $1.89</h2>
```

```
<p>A mixture of espresso, steamed milk and foam.</p>
```

```
<h2>Chai Tea, $1.85</h2>
```

```
<p>A spicy drink made with black tea, spices, milk and  
honey.</p>
```

注意不用把匹配的标记放在同一行上。可以根据需要在它们之间放任意多的内容。

完工了吗?

你已经有了一个带标记的HTML文件，这就是一个Web页面吗？差不多吧。你已经见过<html>、<head>、<title>和<body>标记，现在只需要增加这些标记，让它变成一个一流的HTML页面……

首先，用<html>和</html>标记包围你的HTML。这会告诉浏览器文件的内容是HTML。

```
<html>
```

```
<head>
```

```
<title>Starbuzz Coffee</title>
```

```
</head>
```

```
<body>
```

```
<h1>Starbuzz Coffee Beverages</h1>
```

```
<h2>House Blend, $1.49</h2>
```

```
<p>A smooth, mild blend of coffees from Mexico,  
Bolivia and Guatemala.</p>
```

```
<h2>Mocha Cafe Latte, $2.35</h2>
```

```
<p>Espresso, steamed milk and chocolate syrup.</p>
```

```
<h2>Cappuccino, $1.89</h2>
```

```
<p>A mixture of espresso, steamed milk and foam.</p>
```

```
<h2>Chai Tea, $1.85</h2>
```

```
<p>A spicy drink made with black tea, spices,  
milk and honey.</p>
```

```
</body>
```

```
</html>
```

接下来增加<head>和</head>标记。首部(head)包含Web页面的有关信息，如页面标题。现在可以这样考虑：通过首部可以告诉浏览器关于Web页面的信息。

在head标记中放入title标记。title总出现在浏览器窗口的顶部。

首部包括<head>和</head>标记以及它们之间的所有内容。

页面主体包括<body>和</body>标记以及它们之间的所有内容。

编写HTML时要把首部和页面主体分开。

页面主体包含Web页面的所有内容和结构，这就是你在浏览器中看到的部分。

另一个测试

修改你的“index.html”文件，在文件中增加<head>、</head>、<title>、</title>、<body>和</body>标记。一旦完成，就保存所做的修改，在浏览器中重新加载这个文件。

重新加载index.html文件，可以再次选择Open File (打开文件) 菜单项，或者使用浏览器的重新加载按钮。

注意标题 (你在<head>元素中指定的内容) 会出现在这里。

现在看起来好一点了。浏览器会解释你的标记来提供页面显示，现在页面不仅有了结构，而且更可读。

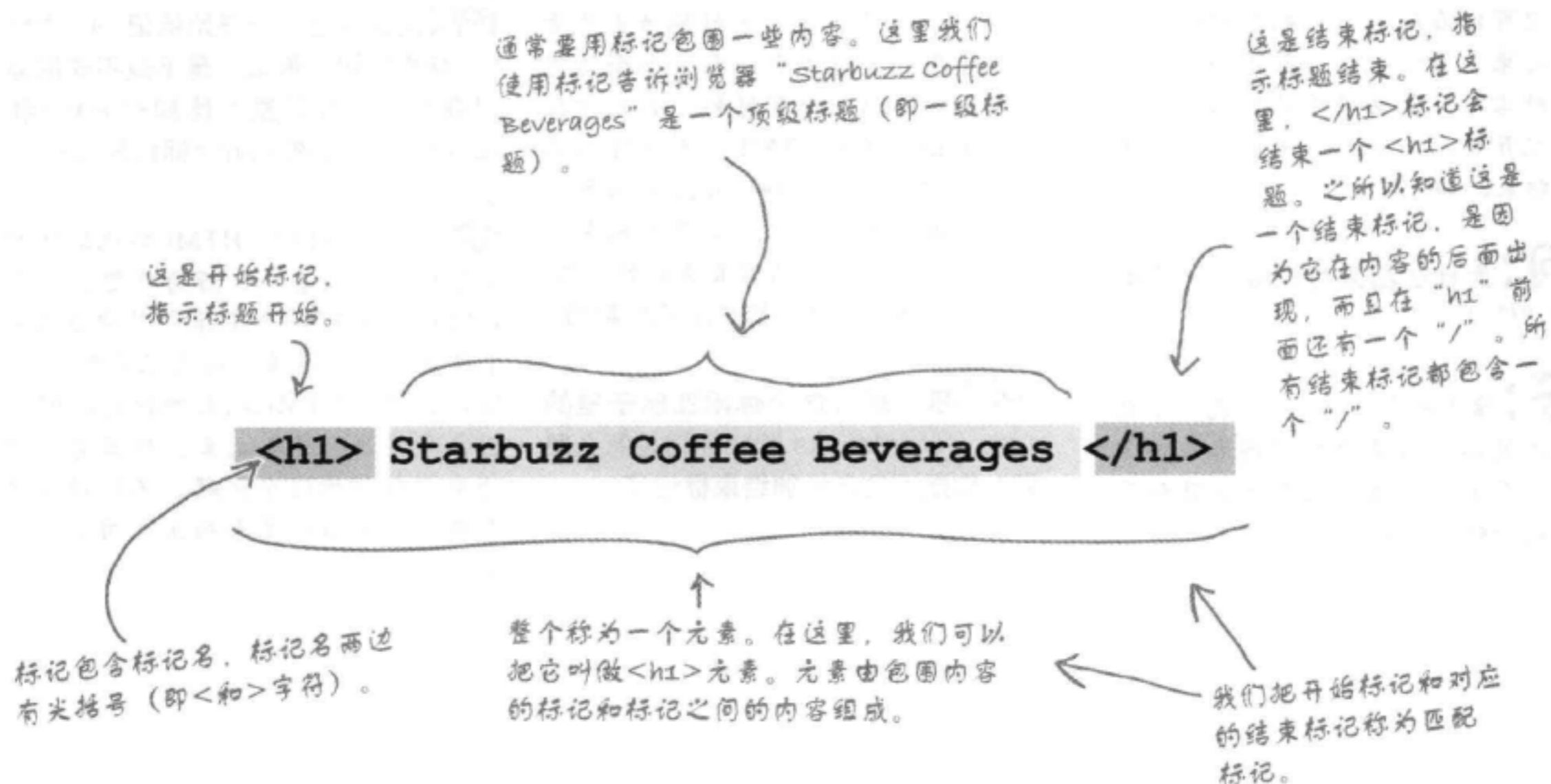


太棒了!



标记剖析

好的，你已经见过不少标记，下面在显微镜下看一看这些标记究竟是如何工作的。



要告诉浏览器页面的结构，需要用成对的标记包围页面内容。

要记住：

元素 = 开始标记 + 内容 + 结束标记

there are no Dumb Questions

问：这么说匹配标记不需要在同一行上？

答：没必要。记住，浏览器并不关心制表符、回车和大多数空格。所以，标记可以在同一行上的任何位置开始和结束，或者也可以在不同行上开始和结束。只是要确保必须以一个开始标记开始，如<h2>，并以一个结束标记结束，如</h2>。

问：为什么结束标记要有一个额外的“/”？

答：结束标记中的“/”是为了让你和浏览器知道某个特定内容在哪里结束。否则，结束标记看上去就和开始标记一样了，是不是？

问：我注意到有些页面的HTML中开始标记和结束标记并不一定匹配。

答：嗯，标记原本应该匹配的。一般来讲，即使你写了不正确的HTML，浏览器也能很好地猜出你原本的意思。不过你会看到，如今写完全无误的HTML大有好处。如果你担心写不出完美的HTML，则大可不必担心。把代码发布到Web服务器展示给全世界之前，有很多工具可以帮你验证代码。现在你只需要养成好习惯，保证开始标记与结束标记总是匹配的。

问：嗯，那么这个休闲室例子中的标记是怎么回事？你是不是忘记加结束标记了？

答：哦，你真厉害，一眼就发现了。有些元素会使用一种简写记法，只有一个标记。现在先不去管它，后面有一章会专门介绍这个内容。

问：元素就是一个开始标记 + 内容 + 结束标记，那么，是不是不能把标记嵌在其他标记里？比如<head>和<body>怎么能在<html>标记里呢？

答：这是可以的，HTML标记通常都会像这样“嵌套”。好好想想，一个HTML页面有一个主体，其中包含一个段落，依此类推，这是很自然的啊。所以，很多HTML元素的标记之间都包含有其他HTML元素。后面有几章还会详细讨论这个问题，不过现在只需要注意页面中元素相互之间是如何关联的。

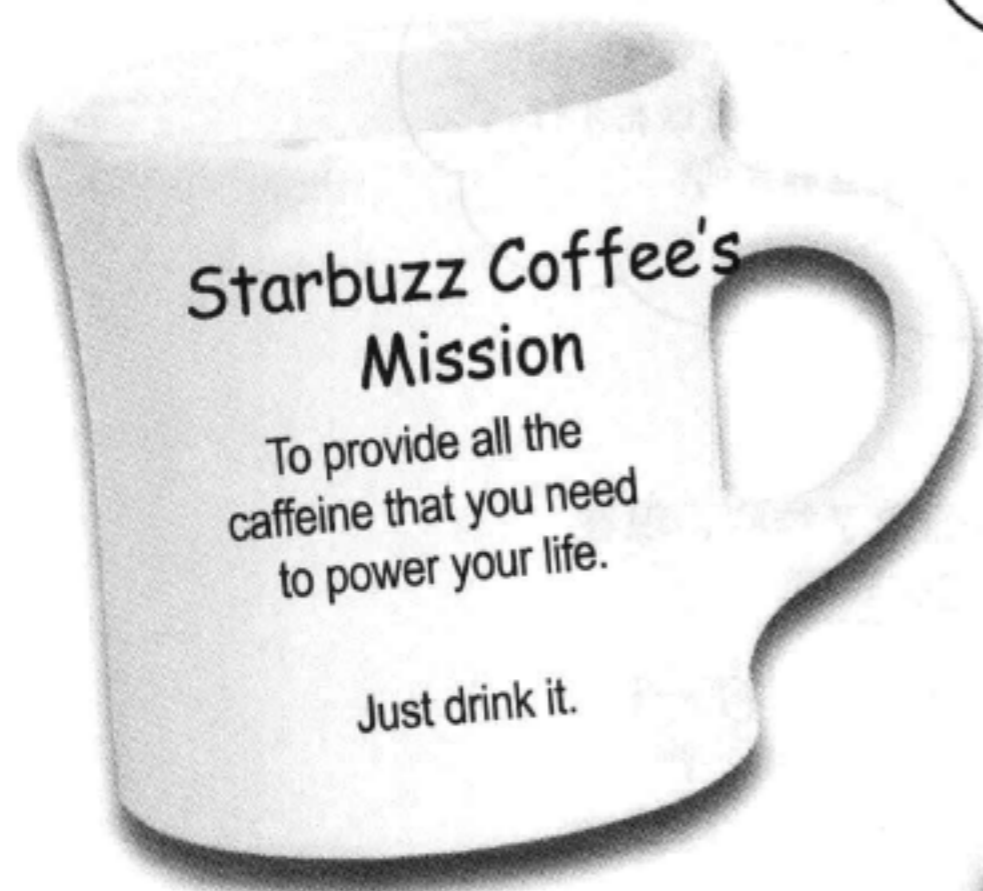


标记可能比你目前为止看到的更有趣一些。下面是一个段落标记，还增加了一点额外的内容。你觉得这会用来做什么？

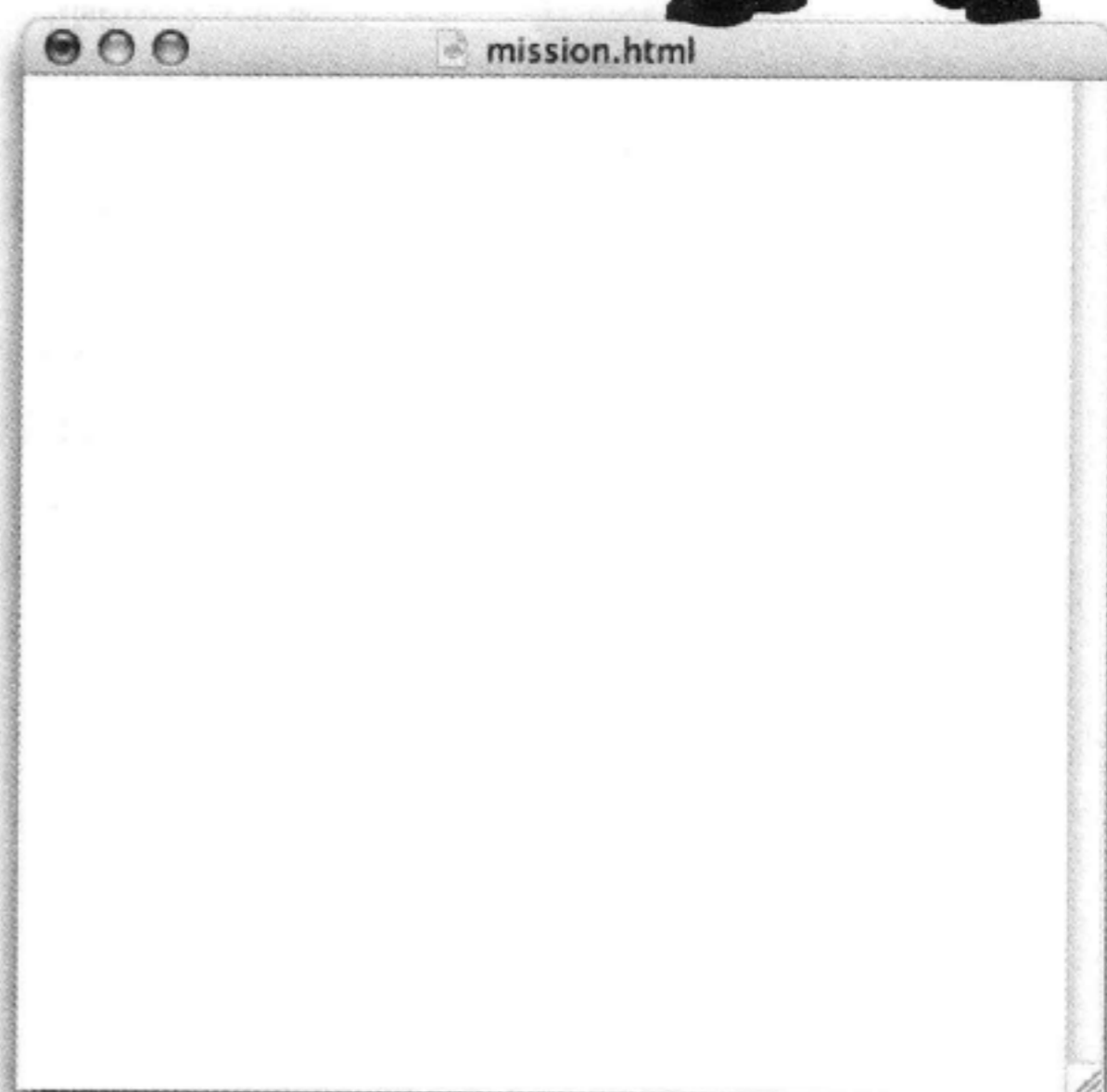
```
<p id="houseblend">A smooth, mild
blend of coffees from Mexico, Bolivia
and Guatemala.</p>
```



Exercise



噢,忘记提醒你了,我们
还要在页面上加上我们公司的
宗旨。可以从Starbuzz咖啡杯上找
到我们的宗旨,再为它创建一个页
面……



- ❶ 把新页面“mission.html”的HTML写在这里。
- ❷ 在一个文本编辑器里输入你的HTML,保存为“mission.html”,放在“index.html”文件所在的同一个文件夹下。
- ❸ 完成后,在浏览器里打开“mission.html”。
- ❹ 继续下一步之前,对照这一章最后给出的页面检查你的工作……



OK, 看起来你已经找到点门道了。现在主页面和宗旨页面都已经搞定。不过, 别忘了CEO说过, 网站看上去还要漂亮才行。你不觉得需要点样式吗?

好的。我们已经确定了结构, 现在来集中精力关注它的表现。

你已经知道了, HTML提供了一种方法来描述文件中内容的结构。浏览器显示HTML时, 它会使用它自己内置的默认样式来表现这个结构。不过, 如果完全依赖于浏览器, 那么显然你没办法得到“本月最佳设计”的称号。

这里CSS就要发挥作用了。利用CSS, 可以描述要如何表现你的内容。下面就来动手试试看, 让这个Starbuzz页面看起来更为美观(同时开启你的Web职业生涯)。

CSS是层叠样式表(Cascading Style Sheets)的缩写。稍后会详细介绍这些是什么意思, 不过现在只需要知道CSS提供了一种方法来告诉浏览器页面中的元素如何显示。

认识style元素

要增加样式，需要在页面中增加一个新的（和我们一起念）E-L-E-M-E-N-T，这就是<style>元素。下面再回到Starbuzz主页面，增加一些样式。试试看……

```
<html>
  <head>
    <title>Starbuzz Coffee</title>
    <style type="text/css">
    </style>
  </head>
  <body>
    <h1>Starbuzz Coffee Beverages</h1>

    <h2>House Blend, $1.49</h2>
    <p>A smooth, mild blend of coffees from Mexico, Bolivia and
    Guatemala.</p>

    <h2>Mocha Caffe Latte, $2.35</h2>
    <p>Espresso, steamed milk and chocolate syrup.</p>

    <h2>Cappuccino, $1.89</h2>
    <p>A mixture of espresso, steamed milk and milk foam.</p>

    <h2>Chai Tea, $1.85</h2>
    <p>A spicy drink made with black tea, spices, milk and honey.</p>
  </body>
</html>
```

<style>元素放在HTML的首部里。

与其他元素类似，<style>元素有一个开始标记<style>，还有一个结束标记</style>。

<style>标记还有一个（可选的）属性，名为type，它能告诉浏览器你在使用什么类型的样式。由于要使用CSS，所以可以指定“text/css”类型。

在这里定义页面的样式。

there are no Dumb Questions

问：元素可以有“属性”？这是什么意思？

答：通过属性，可以提供一个元素的附加信息。比如说，如果有一个<style>元素，属性允许你准确地指定是什么类型的样式。以后还会看到不同元素的更多属性：你只要记住一点，属性能提供元素的一些额外信息。

问：为什么必须指定样式类型(“text/css”)作为style元素的一个属性？难道还有其他类型的样式吗？

答：从前，HTML的设计者认为以后应该还会有其他样式，不过如今我们已经醒悟，事实表明，完全可以只使用<style>而不带类型属性，所有浏览器都知道你指的是CSS。很失望吧，让我们屏住呼吸，拭目以待<style type="50sKitsch">样式的到来吧。呵呵，开玩笑的。

给Starbuzz网站加点样式……

既然HTML首部里有了一个<style>元素，现在要做的就是提供一些CSS，让页面炫一些。你会发现，下面已经为你“烘制”好了一些CSS。看到☞成品标志时，就说明你看到的HTML和CSS可以原样输入。相信我们。后面还会学习这些标记是如何工作的，不过先看看它们能做什么。

现在先来看看这个CSS，再把它增加到“index.html”文件。输入这个CSS之后，保存文件。



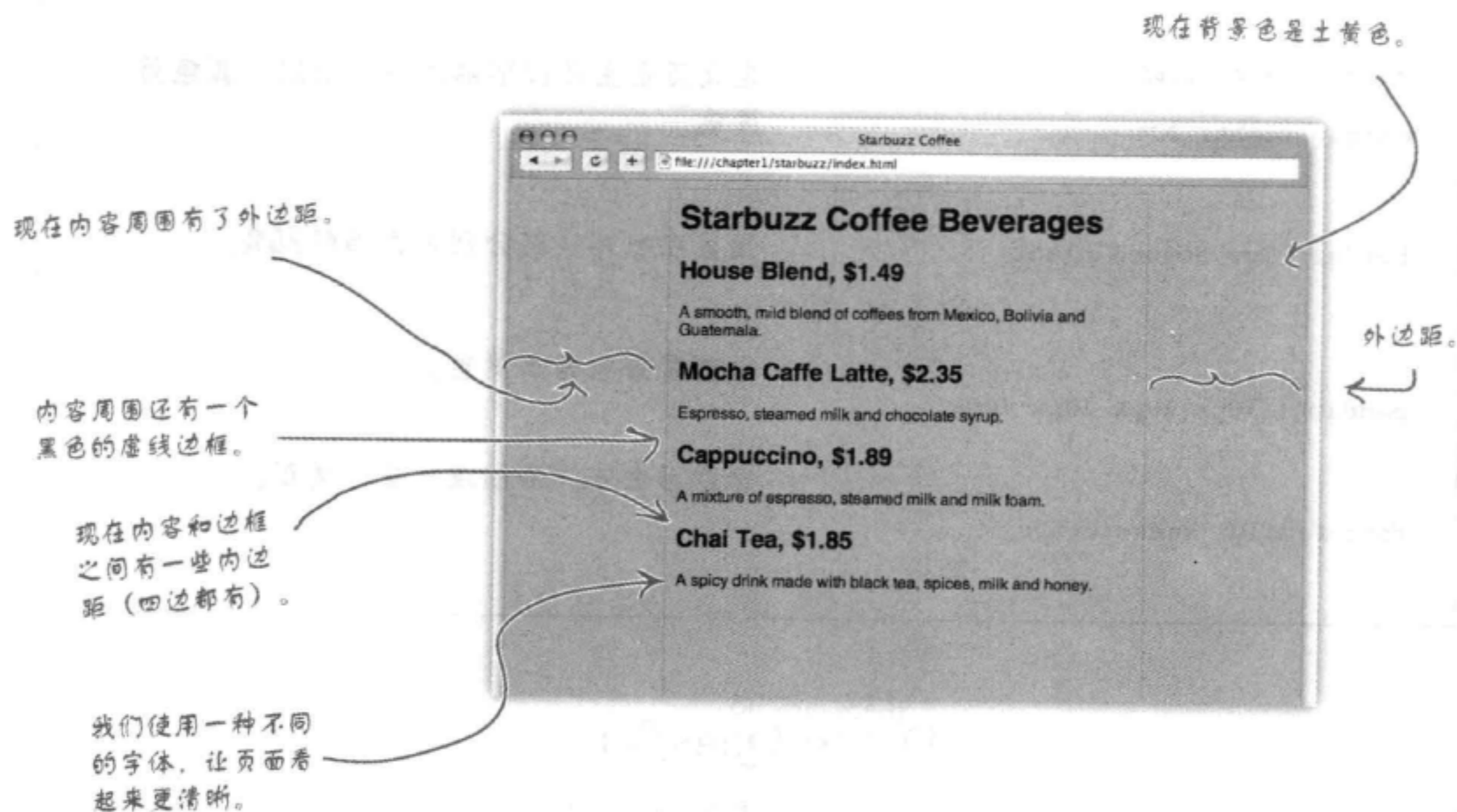
成品CSS

```
<html>
  <head>
    <title>Starbuzz Coffee</title>
    <style type="text/css">
      body {
        background-color: #d2b48c;
        margin-left: 20%;
        margin-right: 20%;
        border: 2px dotted black;
        padding: 10px 10px 10px 10px;
        font-family: sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Starbuzz Coffee Beverages</h1>
    <h2>House Blend, $1.49</h2>
    <p>A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala.</p>
    <h2>Mocha Caffe Latte, $2.35</h2>
    <p>Espresso, steamed milk and chocolate syrup.</p>
    <h2>Cappuccino, $1.89</h2>
    <p>A mixture of espresso, steamed milk and milk foam.</p>
    <h2>Chai Tea, $1.85</h2>
    <p>A spicy drink made with black tea, spices, milk and honey.</p>
  </body>
</html>
```

← CSS使用的语法与HTML完全不同。

测试样式……

又该做个测试了，下面重新加载你的“index.html”文件。这一次你会看到Starbuzz页面有了全新的面貌。



Watch it!

如果你用的浏览器是IE，则可能看不到这个边框。

Internet Explorer无法正确地显示页面主体周围的边框。要想看到这个边框，可以试着在Firefox、Chrome或Safari中加载这个页面。

哇！太棒了。现在我们可以营业了！



WHO DOES WHAT?

尽管只是简单地看一眼CSS，你应该已经能猜出它能做什么了。把样式定义和相应的功能连连看。

```
background-color: #d2b48c;
```

定义文本使用的字体。

```
margin-left: 20%;
```

定义页面主体周围的边框是虚线，颜色为黑色。

```
margin-right: 20%;
```

设置左右外边距分别占页面的20%。

```
border: 2px dotted black;
```

设置背景色为土黄色。

```
padding: 10px 10px 10px 10px;
```

在页面主体周围创建一些内边距。

```
font-family: sans-serif;
```

there are no Dumb Questions

问：CSS看上去与HTML完全是两种不同的语言。为什么要用两种语言？这样一来，我就得学更多东西了，是吗？

答：你说的没错，HTML和CSS确实是完全不同的语言，不过这是因为它们的工作截然不同。就像你不能用英语算账，也不能用数学写诗一样，你不会用CSS来创建结构，或者使用HTML创建样式，因为这并不是当初设计它们的初衷。尽管这意味着你得学习两种语言，但你会发现，由于每个语言各有其擅长的方面，与试图使

用一种语言兼顾这两方面的工作相比，实际上学习两种语言让它们各司其职反而更为容易。

问：#d2b48c看起来可不像是个颜色。#d2b48c怎么会是“土黄色”呢？

答：用CSS指定颜色有很多不同的方式。最常用的一种方式称为“十六进制码”，#d2b48c就是一个十六进制码。这实际上就是土黄色。现在知道就行了，稍后还会告诉你#d2b48c为什么是一种颜色。

问：为什么CSS规则前面有一个“body”？这是什么意思？

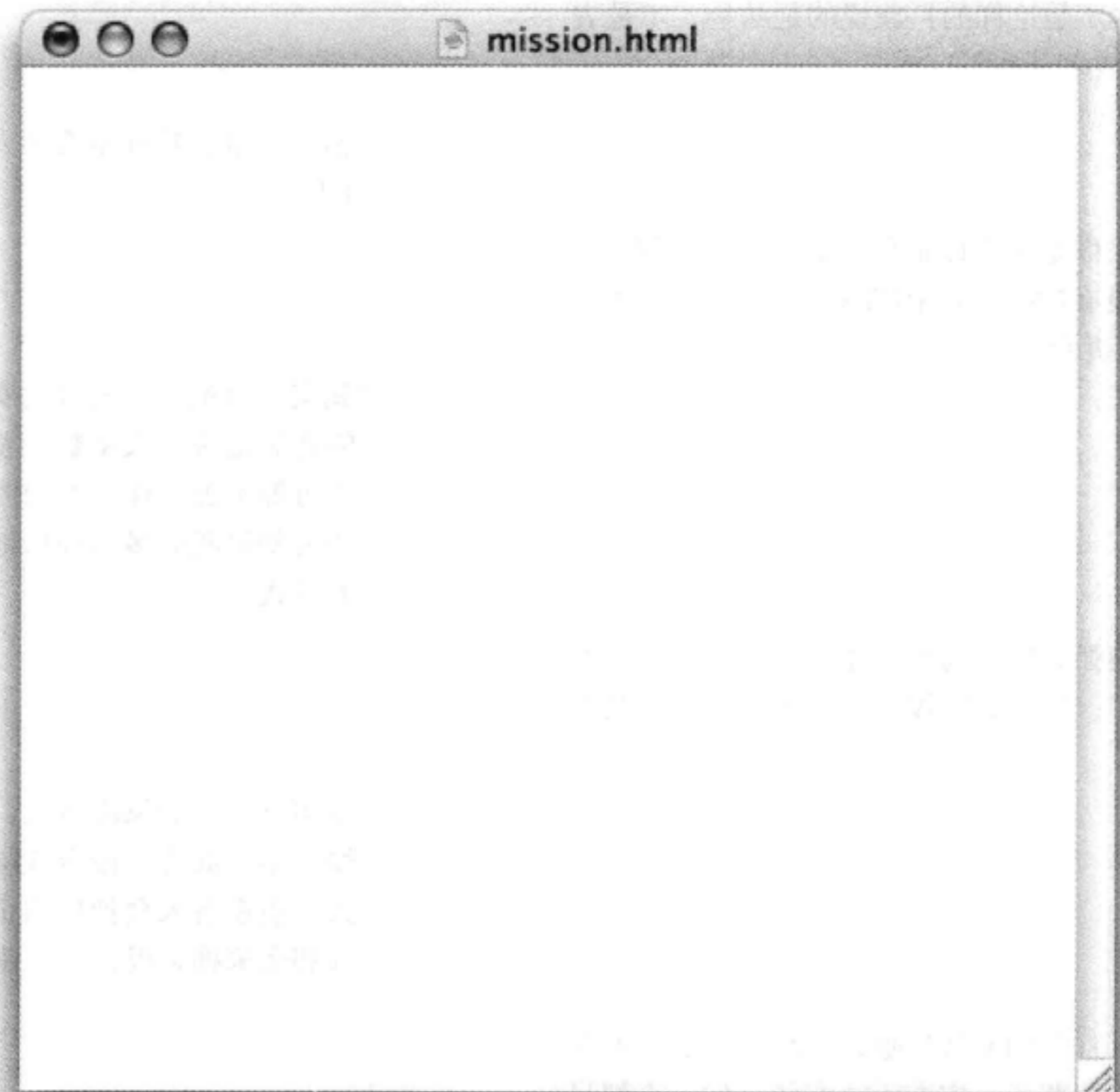
答：CSS中的“body”表示“{”和“}”之间的所有CSS要应用于HTML <body>元素中的内容。所以，将字体设置为sans-serif时，就是说页面主体中的默认字体是sans-serif。

稍后还会更详细地介绍CSS如何工作，请继续读下去。很快你就会发现自己能更熟悉地应用这些规则，这会让你做出漂亮的设计。

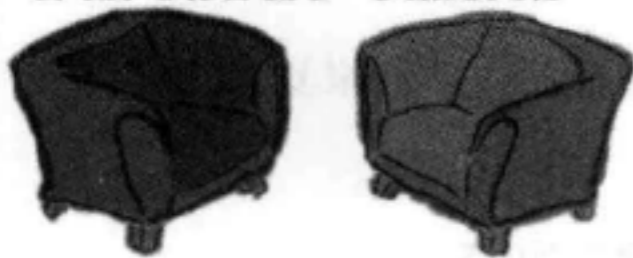


既然已经在Starbuzz“index.html”页面中加了一点样式，下面来更新你的“mission.html”页面，让它有同样的样式。

- ❶ 把“mission.html”页面的HTML写在下面，然后增加新CSS。
- ❷ 更新你的“mission.html”文件，包含新增加的CSS。
- ❸ 完成后，在浏览器中重新加载“mission.html”。
- ❹ 确保你的页面看起来与本章最后给出的宗旨页面是一样的。



Fireside Chats



今晚话题: HTML与CSS谈谈内容和样式

HTML

你好啊, CSS。很高兴你能来, 因为我一直想澄清大家对我们的一些误解。

很多人认为我的标记就是告诉浏览器要如何显示内容。根本不是这样的! 我说的是结构, 不是表现。

嗯, 看到人们是怎么以讹传讹的了吧; 不管怎么说, 即使没有CSS, 只用HTML也完全可以得到一个精美的页面。

喂, 我也很强大啊。为内容建立结构可比让它好看重要多了。样式太肤浅了, 内容的结构才是重中之重。

嗨, 真是个自负的家伙! 嗯, 我想从你这儿不能指望有什么收获了, 实在无法交流, 你一直都只是想用你鼓吹的样式来表现你的个性。

CSS

有吗? 什么误解?

嘿, 没错。我可不希望人们把我的功劳算到你头上!

“精美”可能有点言过其实了吧, 你不觉得吗? 我的意思是说, 大多数浏览器显示纯HTML时看起来可都不怎么样。人们需要了解CSS是多么强大, 还要知道我能多么轻松地给他们的网页增加漂亮的样式。

现实点吧! 如果没有我, 则网页会是多么乏味枯燥。不仅如此, 没有我就没有办法为页面指定样式, 也没有人会把你的页面当回事。一切看上去都那么笨拙呆板、不专业。

HTML

没错。实际上，我们是完全不同的语言，这很好，因为我可不希望你们样式设计人员弄乱我的结构元素。

对，对我来说，很显然每次看到CSS就像是外星人的语言。

你要是这么说，肯定会有上百万的Web开发人员反驳你。我的语法很简洁，与内容配合得很好。

嘿，听说过结束标记吗？

刚注意到，不管你到哪里，我都会用<style>标记把你包围起来。祝你逃跑愉快！

CSS

个性？好的设计和布局对页面的可读性和可用性会有非常大的影响。另外，你应该高兴，正是有了我灵活的样式规则，才允许设计人员对你的元素做各种各样有趣的处理，而不必把你的结构搅得一塌糊涂。

别担心，我们在两个完全不相关的世界里。

是啊，那么HTML也能算是一种语言吗？谁见过这么拙劣的东西，还有所有那些标记？

来看看CSS吧。多么高雅，多么简单，不会<在所有东西><周围>都加上那些笨拙的尖括号。<看><我><也><能><像><HTML先生><那样><说话><,><快><看><我><!>

哈！看着吧……因为，猜不到吧？我真的能逃掉……

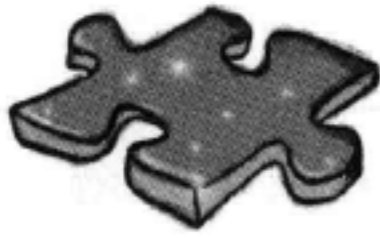
↑ 敬请关注！

现在我们不仅有香浓的House Blend咖啡，还有一个网页向所有顾客介绍我们的咖啡。做得好。关于将来我还有一些更好的想法。另外，你能想一想怎么把这些页面发布到互联网上，让其他人都看到，能做到吗？



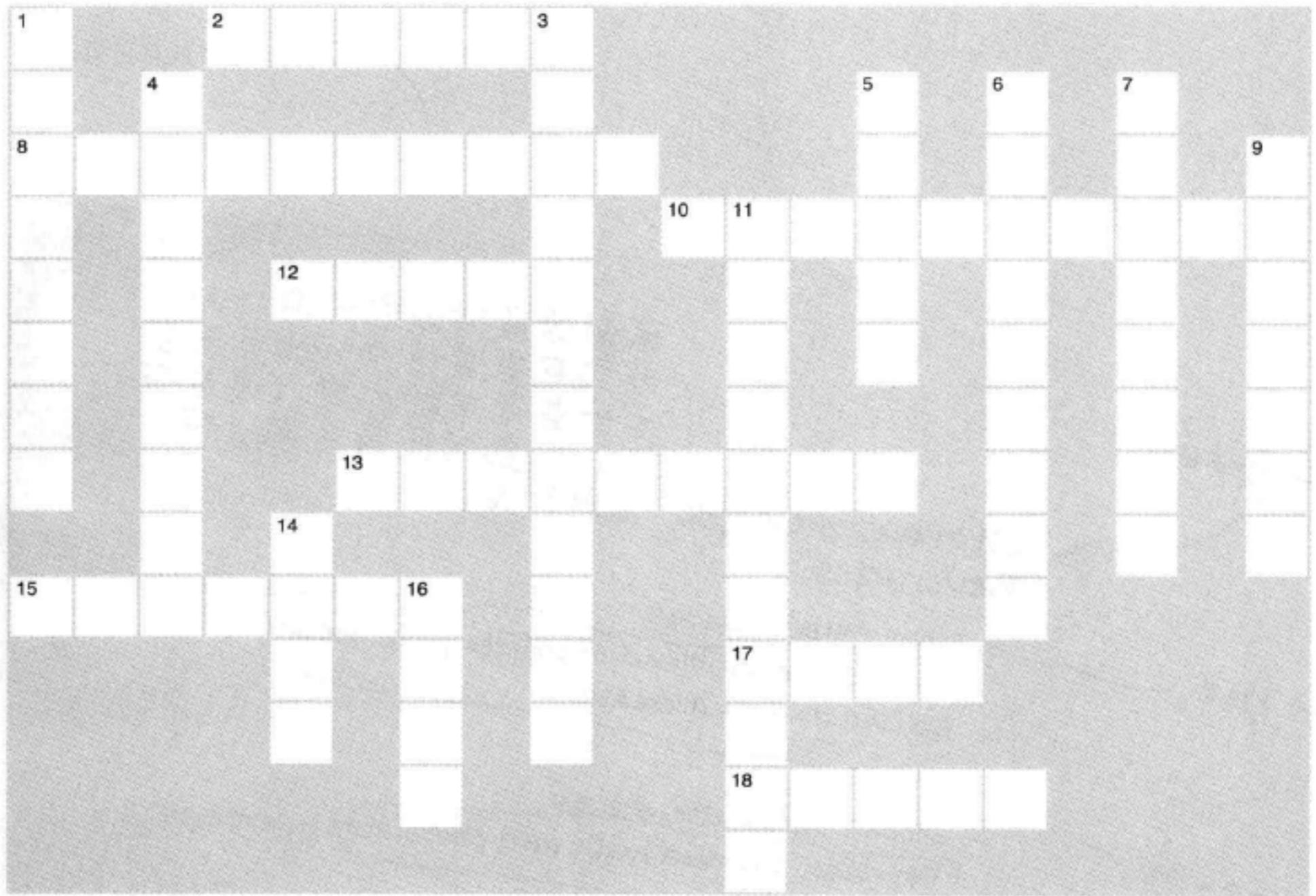
BULLET POINTS

- HTML和CSS是我们用来创建网页的语言。
- Web服务器存储并提供由HTML和CSS创建的网页。浏览器获取页面，并根据HTML和CSS显示网页的内容。
- HTML是超文本标记语言（HyperText Markup Language）的缩写，用来建立网页的结构。
- CSS是层叠样式表（Cascading Style Sheets）的缩写，用来控制HTML的表现。
- 通过HTML，我们利用标记来标示内容提供结构。我们把匹配标记以及它们包围的内容称为元素。
- 元素由3部分组成：一个开始标记、内容和一个结束标记。不过有些元素（比如- 开始标记可以有属性。我们已经见过一个属性：type。
- 结束标记在左尖括号后面、标记名前面有一个“/”，以明确这是结束标记。
- 所有页面都要有一个<html>元素，其中要有一个<head>元素和一个<body>元素。
- 网页的信息放在<head>元素里。
- <body>元素里的内容就是你将浏览器里看到的东西。
- 大多数空白符（制表符、回车、空格）都会被浏览器忽略，不过可以利用空白符让你的HTML（对你）更有可读性。
- 可以在<style>元素中写CSS规则，为HTML网页增加CSS。<style>元素总要放在<head>元素里。
- 可以使用CSS在HTML中指定元素的特性。



HTML填字游戏

该坐下来让你的左脑活动活动了。这是一个标准的填字游戏，所有答案都在这一章里出现过。



横向

2. HTML中的"M"。
8. 标记可以有这些来提供附加信息。
10. 浏览器会忽略它们。
12. 通过这个元素来定义表现。
13. <p>元素的作用。
15. 两个标记及内容。
17. 页面上所看到的。
18. 我们强调了Dance _____ Revolution。

纵向

1. 它们共有6个。
3. 要控制_____时会使用CSS。
4. 标记内容来提供_____。
5. 对于每个页面总是出现在浏览器顶部。
6. 我们希望有的一种样式类型。
7. 开启你的Web职业生涯的公司。
9. 唯一可用的样式类型。
11. HTML中总是将它们分离。
14. 关于网页的信息。
16. 开始和结束。

Sharpen your pencil Solution

在餐巾纸上加些标记（用铅笔），标出你看到的结构，再增加你认为缺少的内容。

增加一个页面标题。

一个子标题。

另一个子标题。

更多子标题。

感谢你能助我们一臂之力！我们只需要这个web页面显示一些简单的内容（请看下面），包括饮料名字、价格和介绍。

这些不放在网页上。

Starbuzz Coffee Beverages

- House Blend, \$1.49
A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala.
- Mocha Cafe Latte, \$2.35
Espresso, steamed milk and chocolate syrup.
- Cappuccino, \$1.89
A mixture of espresso, steamed milk and foam.
- Chai Tea, \$1.85
A spicy drink made with black tea, spices, milk and honey.



标记磁贴答案

你的任务是为Starbuzz餐巾纸上的文本增加结构。使用这一页最下面的冰箱磁贴标记文本，指示哪些部分是标题、子标题和段落文本。下面是我们的答案。

`<h1>` Starbuzz Coffee Beverages `</h1>`

`<h2>` House Blend, \$1.49 `</h2>`

`<p>` A smooth, mild blend of coffees from Mexico, Bolivia and Guatemala. `</p>`

`<h2>` Mocha Cafe Latte, \$2.35 `</h2>`

`<p>` Espresso, steamed milk and chocolate syrup. `</p>`

`<h2>` Cappuccino, \$1.89 `</h2>`

`<p>` A mixture of espresso, steamed milk and foam. `</p>`

`<h2>` Chai Tea, \$1.85 `</h2>`

`<p>` A spicy drink made with black tea, spices, milk and honey. `</p>`

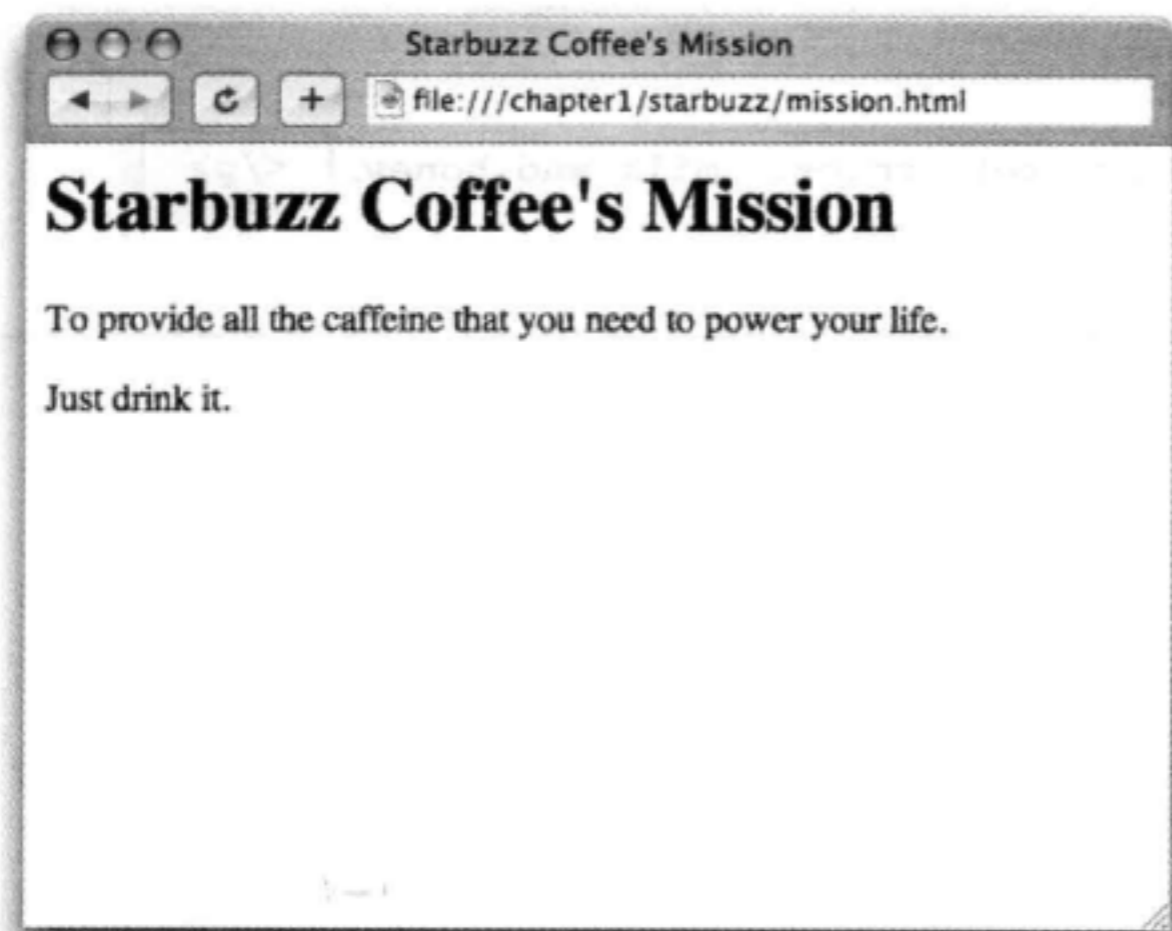


没用上的磁贴。



```
mission.html

<html>
  <head>
    <title>Starbuzz Coffee's Mission</title>
  </head>
  <body>
    <h1>Starbuzz Coffee's Mission</h1>
    <p>To provide all the caffeine that you need to
    power your life.</p>
    <p>Just drink it.</p>
  </body>
</html>
```



↑
这是宗旨页面的HTML
源代码。

←
这是浏览器中显示的HTML。



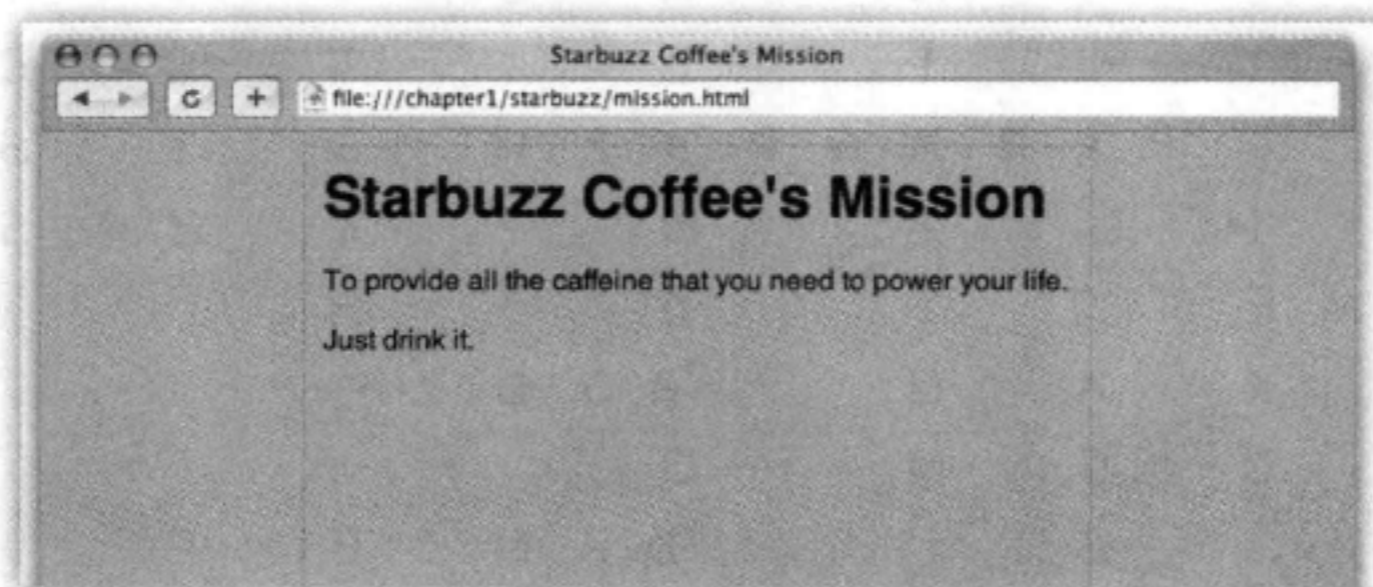
Exercise Solution

这里是宗旨页面中的
CSS。



```
mission.html
<html>
  <head>
    <title>Starbuzz Coffee's Mission</title>
    <style type="text/css" >
      body {
        background-color: #d2b48c;
        margin-left: 20%;
        margin-right: 20%;
        border: 2px dotted black;
        padding: 10px 10px 10px 10px;
        font-family: sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Starbuzz Coffee's Mission</h1>
    <p>To provide all the caffeine that you need to power
    your life.</p>
    <p>Just drink it.</p>
  </body>
```

现在样式与Starbuzz
主页面一致了。





★ WHO DOES WHAT? ★

尽管只是简单地看一眼CSS，你应该已经能猜出它能做什么了。把样式定义和相应的功能连连看。

background-color: #d2b48c;

定义文本使用的字体。

margin-left: 20%;
margin-right: 20%;

定义页面主体周围的边框是虚线，颜色为黑色。

border: 2px dotted black;

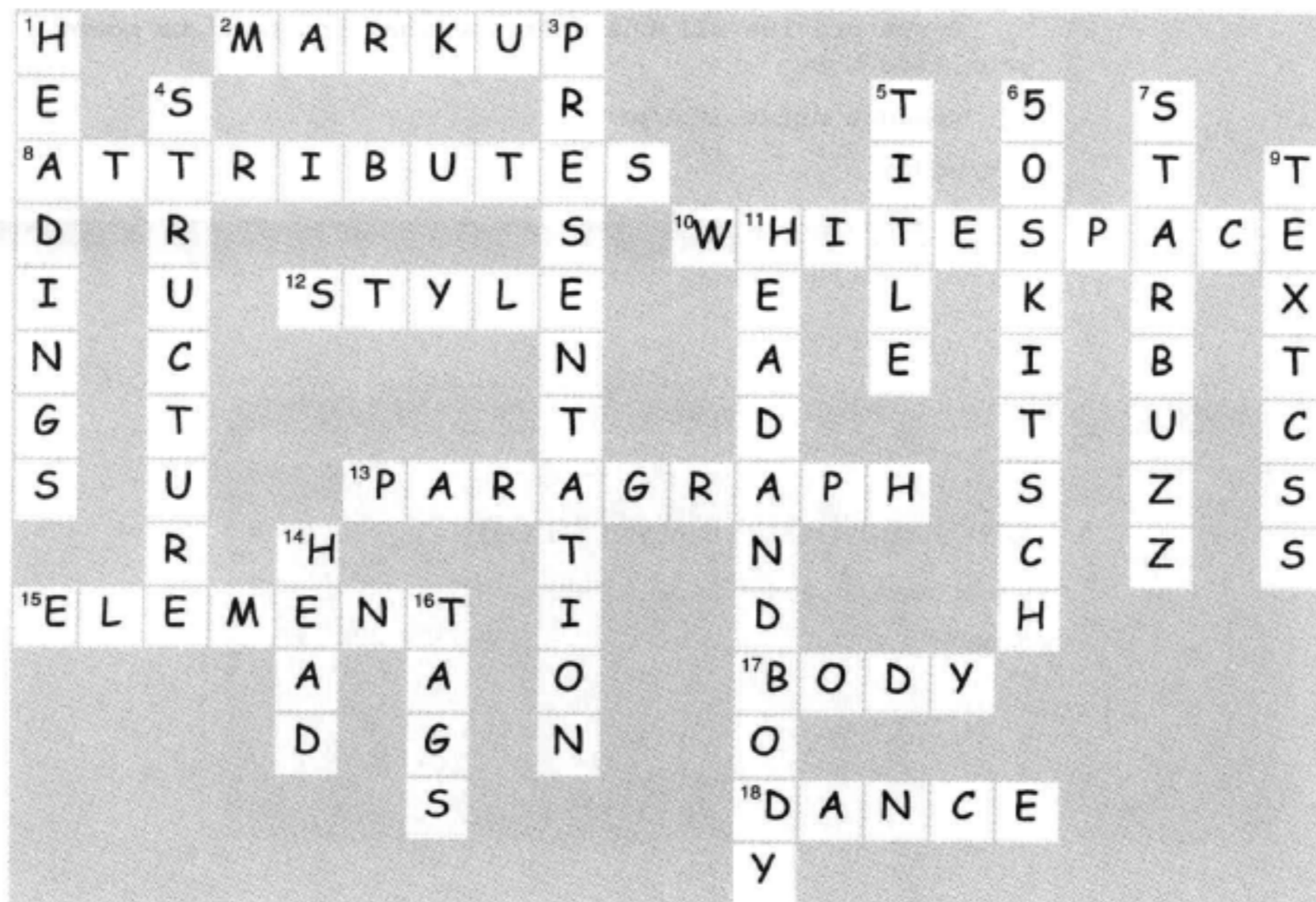
设置左右外边距分别占页面的20%。

padding: 10px 10px 10px 10px;

设置背景色为土黄色。

font-family: sans-serif;

在页面主体周围创建一些内边距。

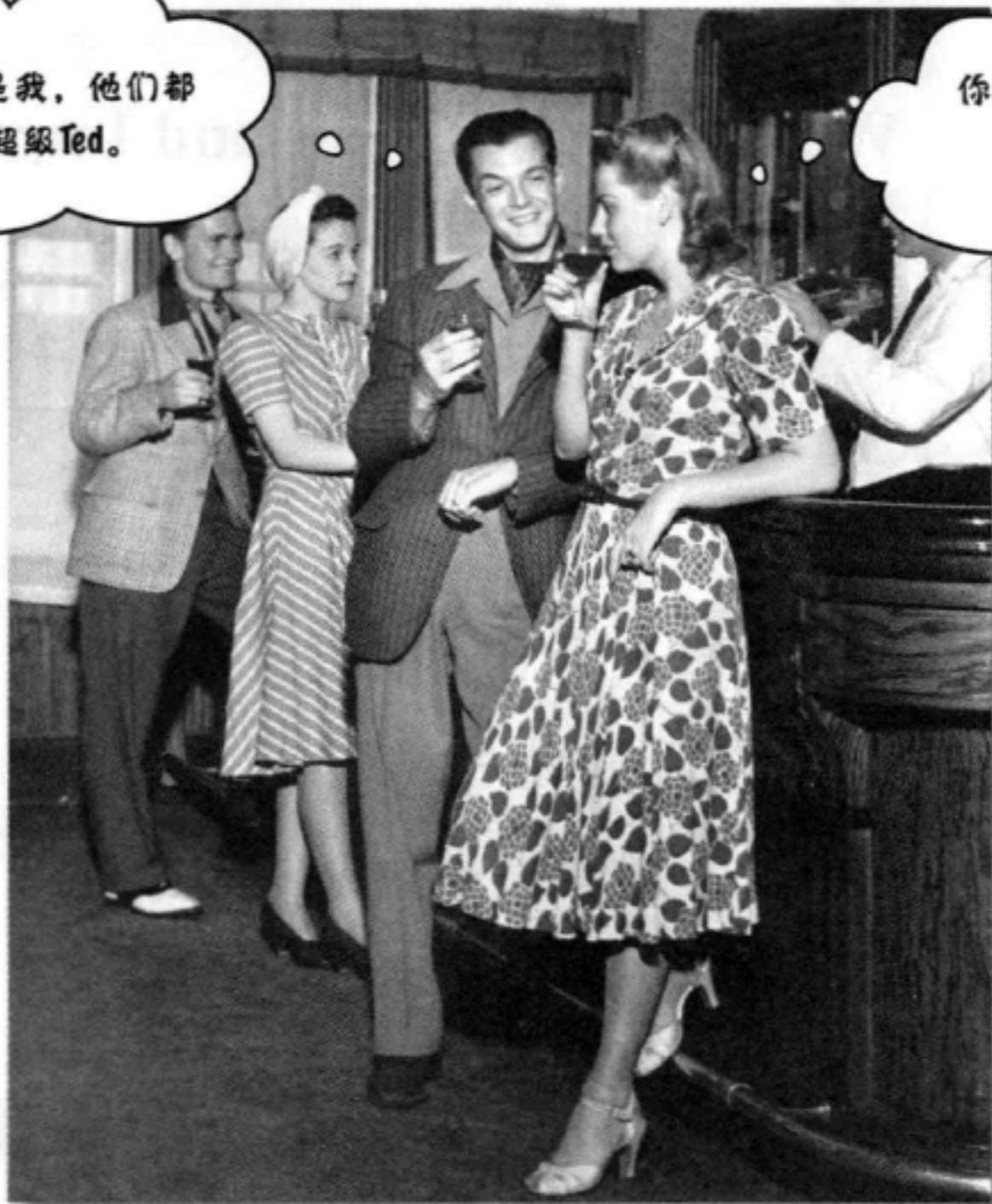


2 深入了解超文本

★ 认识HTML中的“HT” ★

没错，就是我，他们都叫我超级Ted。

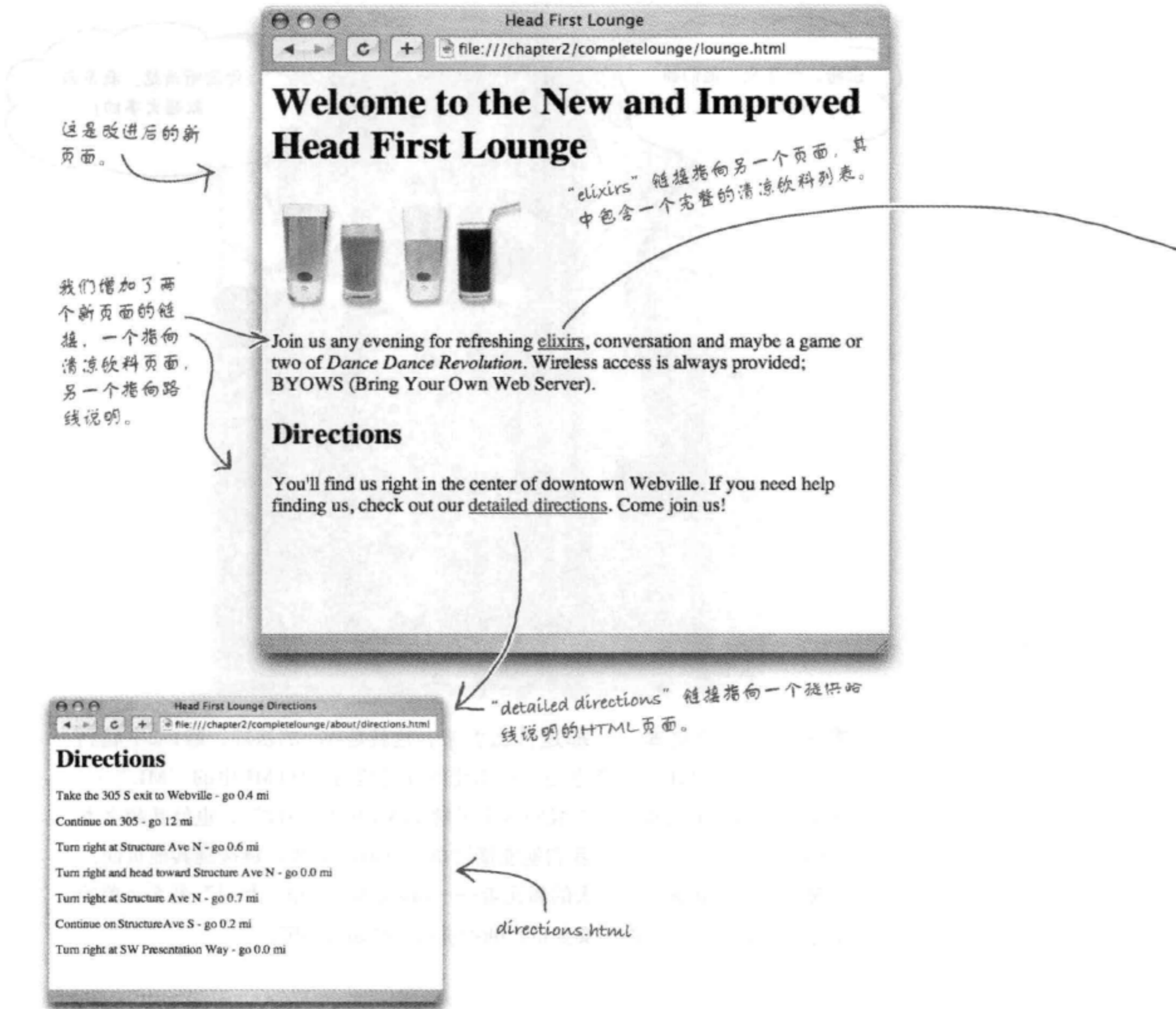
你没听清楚。我是来找超文本的！



有人在说“超文本”？那是什么？哦，这就是Web的基础。第1章中我们简单认识了HTML，知道它是一种不错的标记语言（HTML中的“ML”），用来描述网页的结构。现在我们来了解HTML中的“HT”，也就是超文本（hypertext），有了它，我们就能摆脱单个页面的束缚，链接到其他页面。顺便我们还要认识一个强大的新元素——<a>元素，了解“相对”是多么美妙的东西。好了，系好你的安全带，准备学习一些超文本吧。

Head First休闲室，全新改良

还记得Head First休闲室吗？那是一个不错的网站，不过，如果顾客能看到一个清凉饮料列表就更好了，不是吗？甚至还可以更棒，我们应该为顾客提供一些具体的路线说明，让他们能找到我们的休闲室。





这个页面列出一些清凉健康饮料。可以先来一杯。



创建这个全新改良版的休闲室需要3步……

下面来改造原先的Head First休闲室页面，让它链接到这两个新页面。

- 1 第1步很容易，因为我们已经为你创建了“directions.html”和“elixir.html”文件。可以在本书的源代码文件  成品 (<http://wickedlysmart.com/hfhtmlcss>) 中找到这两个文件。
- 2 接下来，需要编辑“lounge.html”文件，加入链接到“directions.html”和“elixir.html”所需的HTML。
- 3 最后，对这些页面做个测试，试试看新链接是否正常。完成后，我们再坐下来好好研究它是如何工作的。

翻开下一页，我们开始吧……

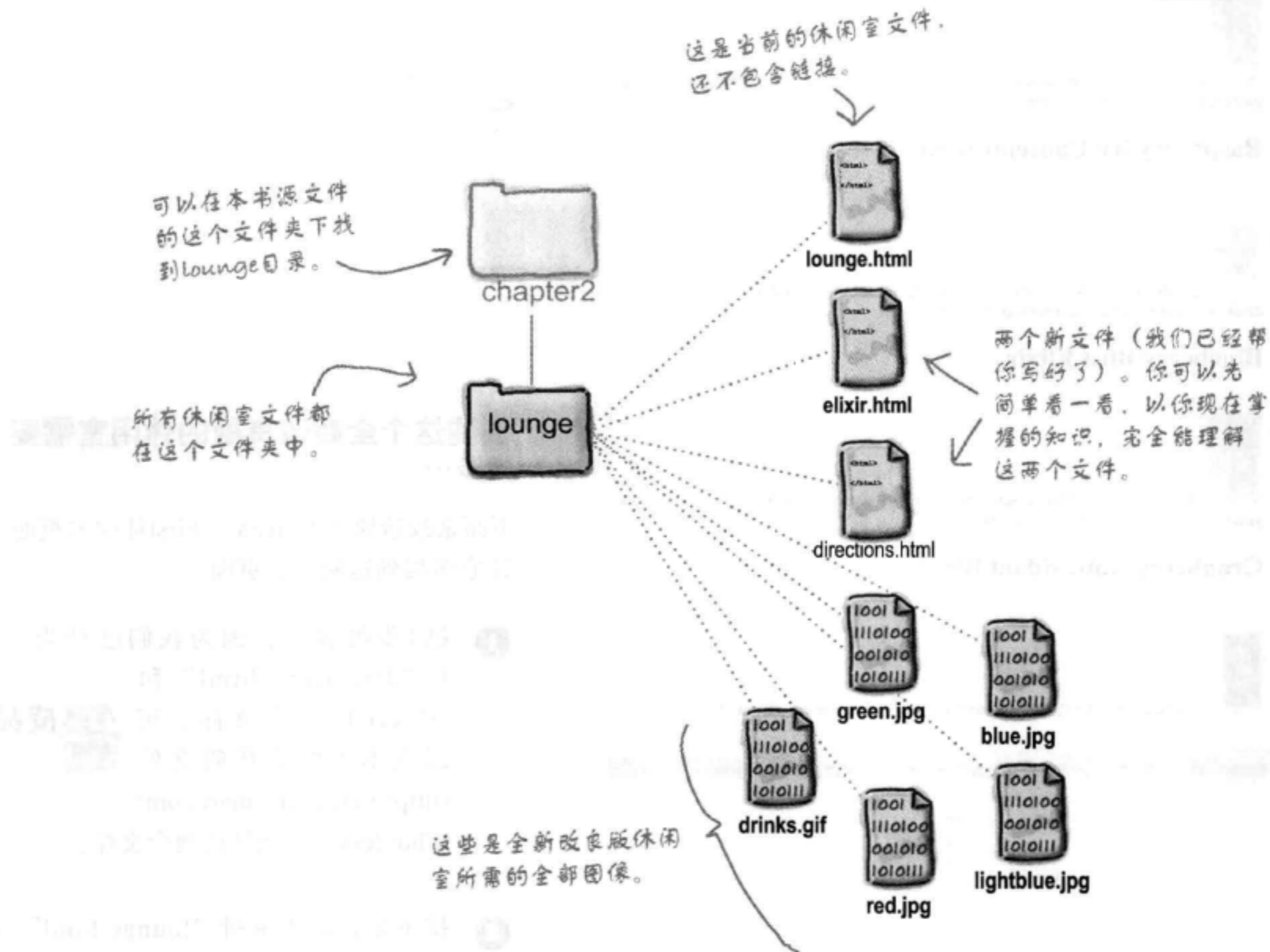
elixir.html



创建新的休闲室

1 获取源文件。

从<http://wickedlysmart.com/hfhtmlcss>获取源文件。下载之后，查看文件夹“chapter2/lounge”，你会发现“lounge.html”、“elixir.html”和“directions.html”文件（还有一大堆图像文件）。



Head First休闲室还在发展壮大，你觉得把网站的所有文件都放在一个目录下能很好地组织这个网站吗？有没有别的办法？

2 编辑lounge.html。

在编辑器中打开“lounge.html”。增加下面突出显示的新文本和HTML。输入这些代码，下一页我们再来分析这是如何工作的。

```

<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for
      refreshing <a href="elixir.html">elixirs</a>,
      conversation and maybe a game or two of
      <em>Dance Dance Revolution</em>.
      Wireless access is always provided;
      BYOWS (Bring your own web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown Webville.
      If you need help finding us, check out
      our <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>

```

在标题中增加文本“New and Improved”。

在这里增加清凉饮料链接的HTML。

要创建链接，需要使用<a>元素，稍后我们会讨论这个元素是如何工作的……

这里需要增加一些文本，指引顾客访问新增的路线说明。

在这里增加指向这些说明的链接，同样的，还是要使用一个<a>元素。

3 保存lounge.html，再做个测试。

完成上述修改后，保存文件“lounge.html”，再在浏览器中打开。可以做下面的尝试：

- ① 单击清凉饮料链接，会显示新的清凉饮料页面。
- ② 单击浏览器的“后退”按钮，会再次显示“lounge.html”。
- ③ 单击路线说明链接，会显示新的说明页面。

在幕后

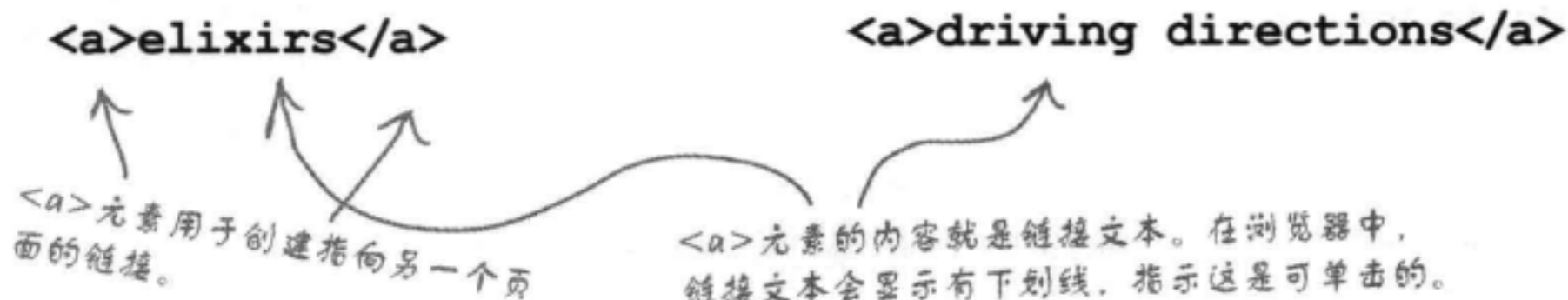


Ok, 我已经加载了新的休闲室页面, 单击了链接, 一切都工作正常。不过我还想搞清楚这些HTML是如何工作的。



我们做了什么?

- ① 下面一步一步来创建HTML链接。首先, 需要把链接文本放在元素中, 就像下面这样:



- ② 既然有了链接文本, 现在需要增加一些HTML告诉浏览器这个链接指向哪里:

`elixirs`

要通过href属性来指定链接的目标文件。

`driving directions`

对于这个链接, 浏览器会显示文本“elixirs”, 单击这个文本时, 用户将被带往“elixir.html”页面。

对于这个链接, 浏览器会显示“driving directions”链接, 单击时, 用户会进入“directions.html”页面。

浏览器做了什么？

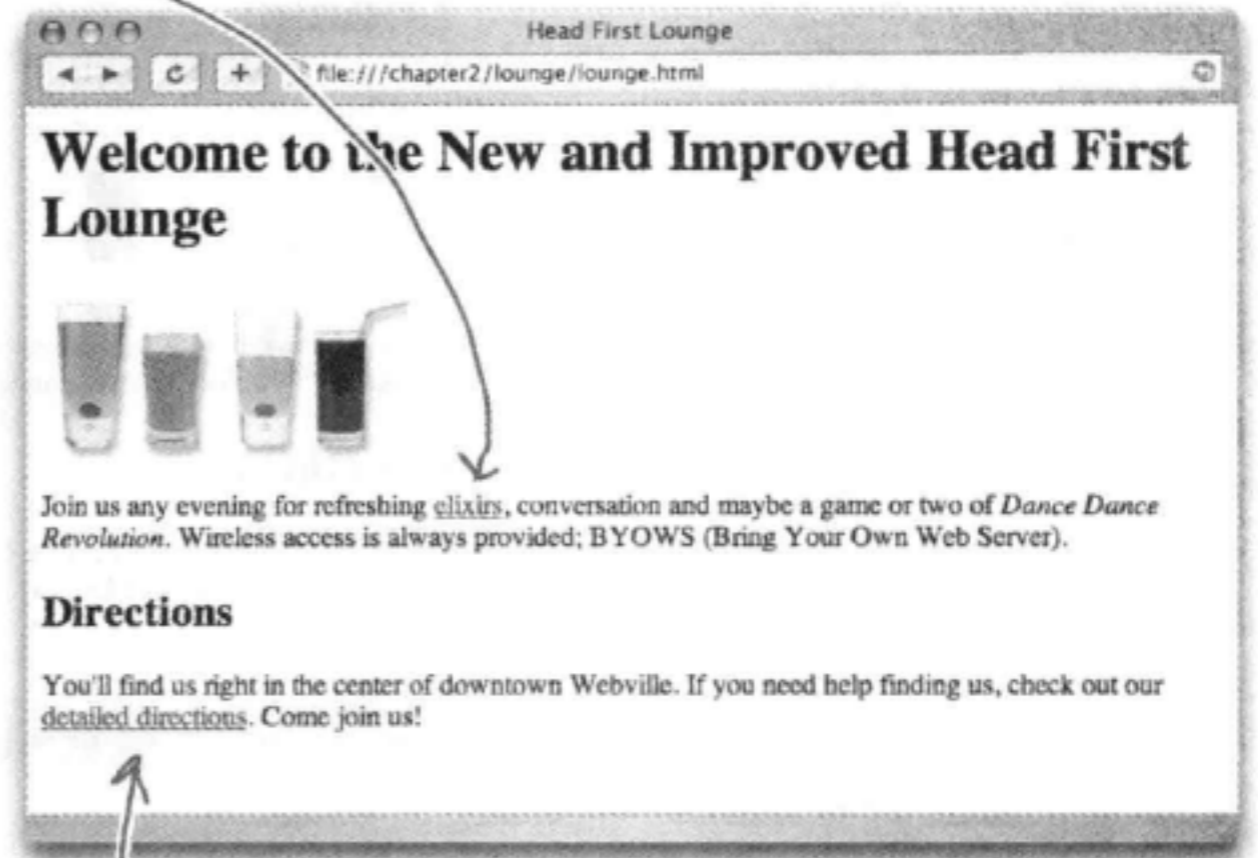
在幕后



- ① 首先，浏览器会显示页面，如果遇到一个[<a>](#)元素，则会取这个元素的内容，把它显示为一个可单击的链接。

```
<a href="elixir.html">elixirs</a>
```

“elixirs”和“detailed directions”都放在开始和结束[<a>](#)标记之间，所以在网页上它们会显示为可单击的文本。



```
<a href="directions.html">detailed directions</a>
```

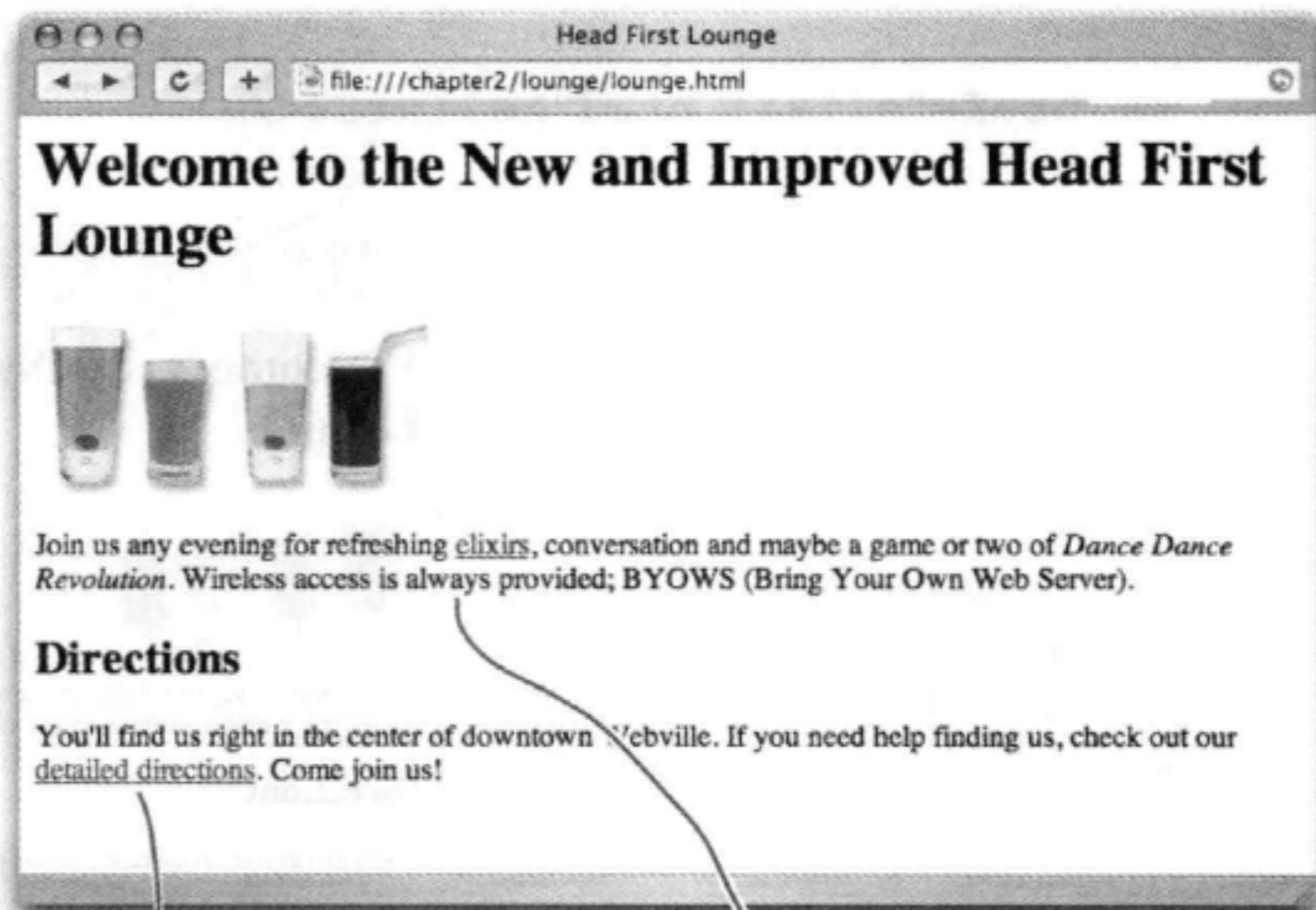
使用[<a>](#)元素创建一个超文本链接，链接到另一个Web页面。

[<a>](#)元素的内容会成为Web页面中可单击的文本。

[href](#)属性告诉浏览器链接的目标文件。



- ② 接下来，用户单击一个链接时，浏览器使用“href”属性来确定这个链接指向哪个页面。



用户单击
“elixirs” 链接或……

……单击 “detailed
directions”。

单击 “detailed directions” 时，浏览器获取 href 属性的值，在这里就是 “directions.html” ……

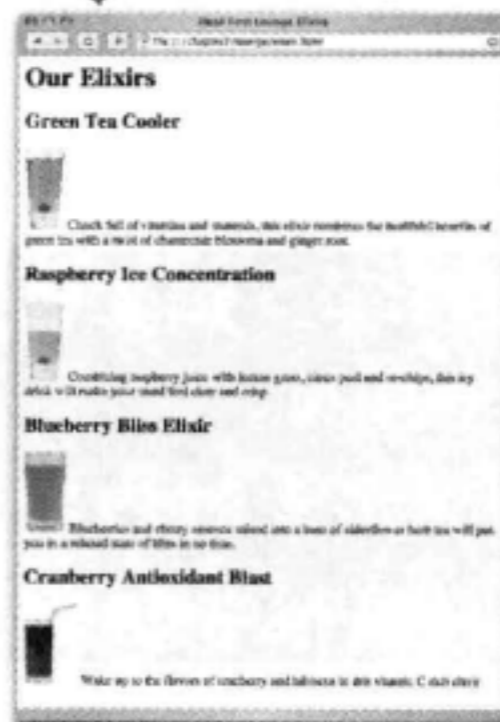
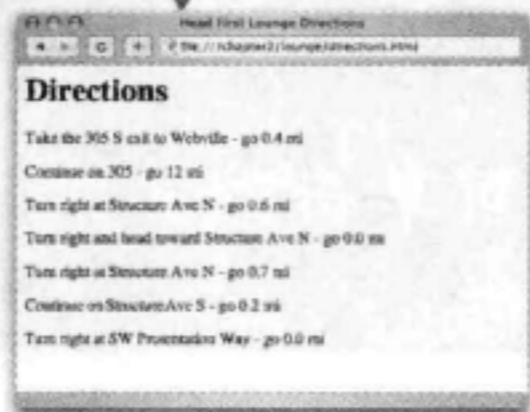
如果单击 “elixirs”，则浏览器会得到 href 值 “elixir.html” ……

```
<a href="elixir.html">elixirs</a>
```

……并显示 “elixir.html” 页面。

```
<a href="directions.html">detailed directions</a>
```

……并加载 “directions.html”。



了解属性

利用属性 (Attributes)，可以指定一个元素的附加信息。尽管我们还没有详细分析属性，不过之前已经见过属性的几个例子：



```
<style type="text/css">
```

type属性指定我们使用哪一种样式语言，这里就是CSS。

```
<a href="irule.html">
```

href属性告诉我们一个超链接的目标文件。

```

```

src属性指定一个img标记显示的图像的文件名。

下面给出一个例子，以便你更好地了解属性如何工作：

如果<car>是元素呢？

如果<car>是一个元素，很自然地，你可能想写类似这样的标记：

```
<car>My Red Mini</car>
```

没有任何属性，我们能提供的仅仅是一个描述性的车名。

不过这个<car>元素只能提供一个描述性的车名，它没有告诉我们汽车的品牌、型号、是不是敞篷车，以及我们可能想知道的很多其他细节。所以，如果<car>确实是一个元素，则我们可能会像下面这样使用属性：

```
<car make="Mini" model="Cooper" convertible="no">My Red Mini</car>
```

不过有了属性，就可以用各种信息定制这个元素。

是不是好一些了？现在这个标记采用一种方便、易写的形式提供了更多信息。

安全第一



属性的写法都是一样的：首先是属性名，后面是一个等于号，然后是用双引号括起来的属性值。

你可能看到Web上有一些不严谨的HTML没有加双引号，不过你可不能偷懒。如果不严谨，可能导致你以后会遇到很多问题（本书后面就会看到）。

这样做（最佳实践）

```
<a href="top10.html">Great Movies</a>
```



不要这样做

```
<a href=top10.html>Great Movies</a>
```

属性值周围没有双引号

there are no Dumb Questions

问:我能为一个HTML元素造新属性吗?

答:Web浏览器只认识每个元素的一组预定义的属性。如果你造了新属性,则浏览器不知道该怎么处理,另外在本书后面你会看到,这样做很可能给你带来麻烦。浏览器能识别一个元素或属性时,我们会说浏览器“支持”这个元素或属性。应该只使用浏览器支持的那些属性。

尽管如此,关于编写Web应用(这是《Head First HTML5 Programming》讨论的主题),现在HTML5已经支持定制数据属性,允许你为新属性构造定制的属性名。

问:谁来决定“支持”哪些属性?

答:有一些标准委员会专门考虑HTML的元素和属性。这些委员会由没其他事可干的这样一群人组成,他们不计得失、不计报酬,无私地投入时间和精力来确保建立一个通用的HTML路线图,使得所有组织都能使用这样一个标准实现他们的浏览器。

问:我怎么知道哪些属性和元素得到支持呢?另外,所有属性都能应用于任何元素吗?

答:一个给定元素只能使用某些特定的属性。可以这样来考虑:你不会对元素<toaster>使用属性“convertible”吧,是不是?也就是说,你只希望使用元素支持的那些有意义的属性。

通过这本书的介绍,你会逐步了解哪些元素支持哪些属性。学完这本书之后,还有很多不错的参考书可以帮你复习,如《HTML & XHTML: The Definitive Guide》(O'Reilly)。





属性闪亮登场

本周访谈：
href属性的自白

Head First: 欢迎你, href。很荣幸能采访像你这么有名气的属性。

href: 谢谢。我也很高兴来这里,可以摆脱所有链接。对一个属性来说,这实在是让我精疲力尽了。每次有人单击一个链接时,猜猜看是谁告诉浏览器下一步该去哪里?那就是我啊。

Head First: 真是感谢你能在百忙中抽出时间。能不能从头说说……属性到底能做什么?

href: 当然可以。嗯,属性可以用来定制一个元素。用标记包围一些内容很容易,比如“Sign up now!”。我们会这么做: `<a>Sign up now!`。不过,如果没有我,就没有href属性,你就没办法告诉元素链接的目标文件是什么。

Head First: 不太明白……

href: ……不过,有了属性,你就能提供元素的附加信息。就我而言,就是链接要指向哪里: `Sign up now!`。这说明,这个标签为“Sign up now!”的元素要链接到“signup.html”页面。当今世界还有许许多多其他属性,不过在元素中只能用我来告诉它要指向哪里。

Head First: 噢。那么我还得问一句,希望你别介意,这个名字是怎么来的? href? 它有什么特别的意思吗?

href: 这是互联网家族的一个古老名字。它的含义是“超文本引用”(hypertextreference),不过我的朋友都叫我小名“href”。

Head First: 那又是什么?

href: 超文本引用就是互联网或你的计算机上的一个资源的别称。通常这个资源就是一个Web页面,不过我也可以指向PDF文档……实际上我可以指向各种各样的资源。

Head First: 有意思。我们的读者目前为止看到的链接都只是指向他们自己的页面,怎么链接到Web上的其他页面和资源呢?

href: 嘿,我得回去工作了,没有我,整个Web该成一团乱麻了。再说了,教他们如何链接不该是你的工作吗?

Head First: 好吧,好吧,没错,我们马上去做……非常感谢你接受采访, href。



Exercise

你已经创建了从“lounge.html”到“elixir.html”和“directions.html”的链接,现在再反过来。下面是“elixir.html”的HTML。请在清凉饮料页面的最下面增加一个标为“Back to the Lounge”的链接,指回到“lounge.html”。

```
<html>
  <head>
    <title>Head First Lounge Elixirs</title>
  </head>
  <body>
    <h1>Our Elixirs</h1>

    <h2>Green Tea Cooler</h2>
    <p>
      
      Chock full of vitamins and minerals, this elixir
      combines the healthful benefits of green tea with
      a twist of chamomile blossoms and ginger root.
    </p>
    <h2>Raspberry Ice Concentration</h2>
    <p>
      
      Combining raspberry juice with lemon grass,
      citrus peel and rosehips, this icy drink
      will make your mind feel clear and crisp.
    </p>
    <h2>Blueberry Bliss Elixir</h2>
    <p>
      
      Blueberries and cherry essence mixed into a base
      of elderflower herb tea will put you in a relaxed
      state of bliss in no time.
    </p>
    <h2>Cranberry Antioxidant Blast</h2>
    <p>
      
      Wake up to the flavors of cranberry and hibiscus
      in this vitamin C rich elixir.
    </p>

    </body>
  </html>
```

你的新HTML放在这里。 →

完成后, 再在“directions.html”中做同样的处理。

CONSTRUCTION
ZONE

BEGINS

我们需要你的帮助来构造和解构元素。希望你新学的元素知识能有帮助。下面每一行分别给出了标签、目标文件和完整的元素。请填入缺少的信息。我们已经帮你完成了第一行。

标签	目标文件	HTML中要写的代码
Hot or Not?	hot.html	<code>Hot or Not?</code>
Resume	cv.html	
	candy.html	<code>Eye Candy</code>
See my mini	mini-cooper.html	
let's play		<code>.....</code>

there are no Dumb Questions

问：我见过很多页面上还可以单击图像而不只是文本。能用元素来实现吗？

答：可以，如果把一个元素放在标记之间，这个图像就会像文本一样可单击。我们不打算花好几章来深入讨论图像，不过要知道，图像确实也可以作为链接。

问：这么说任何东西只要放在标记之间就会成为可单击的，是吗？比如说，一个段落？

答：确实可以把一个

元素放在元素中来链接整个段落。我们大多会在元素中使用文本和图像（或者二者同时使用），不过，如果你需要链接一个

或

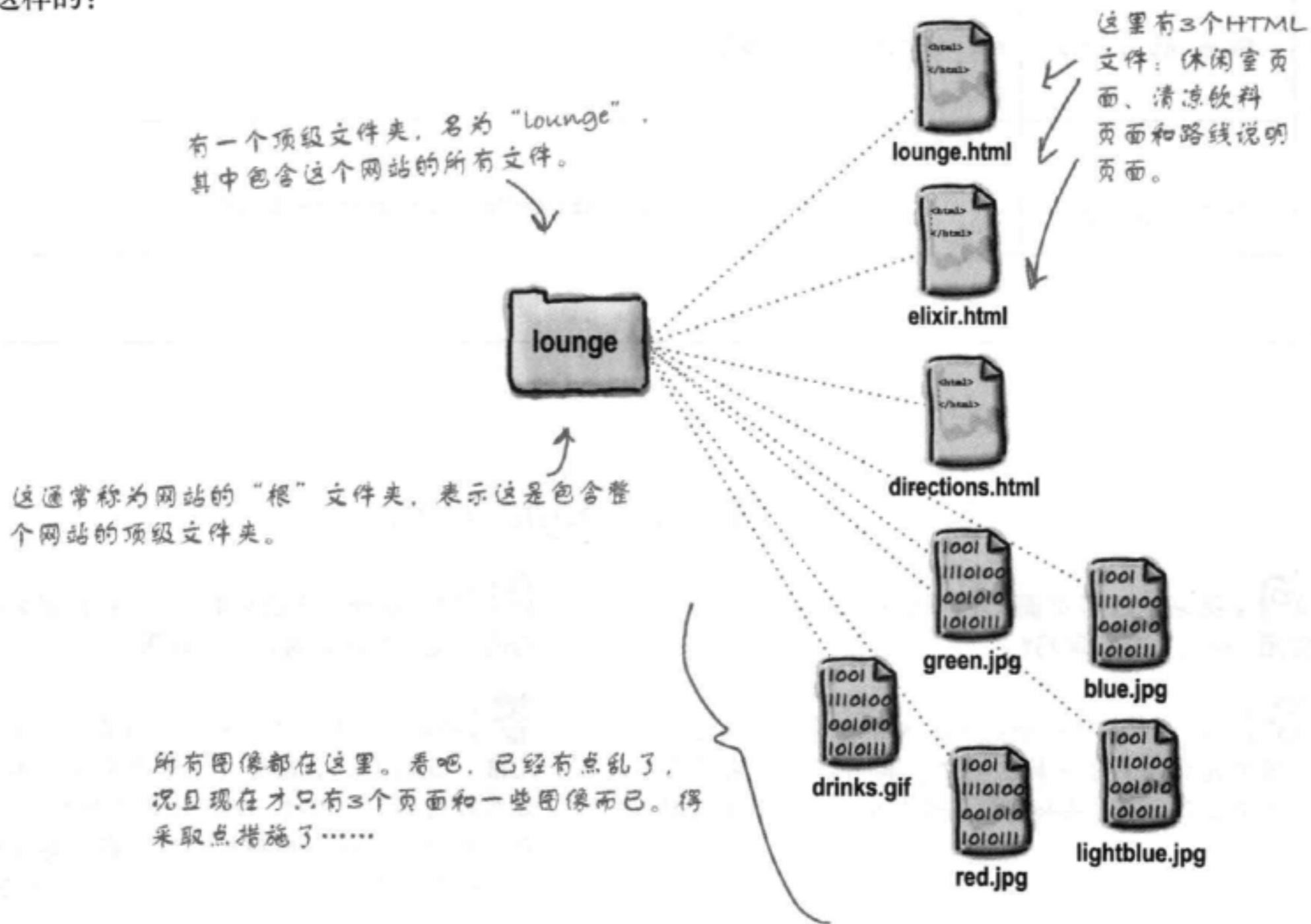
元素，则完全可以。哪些标记可以嵌在其他标记里，这完全是另一个话题了，不过不用担心，很快就会谈到这个内容。

你的Head First休闲室
网站真是很有成效。加上那
些诱人的清凉饮料和路线说明，顾客们频频光顾，
访问网站的人也越来越多。现在我们有计划全方位
扩展休闲室的线上内容。



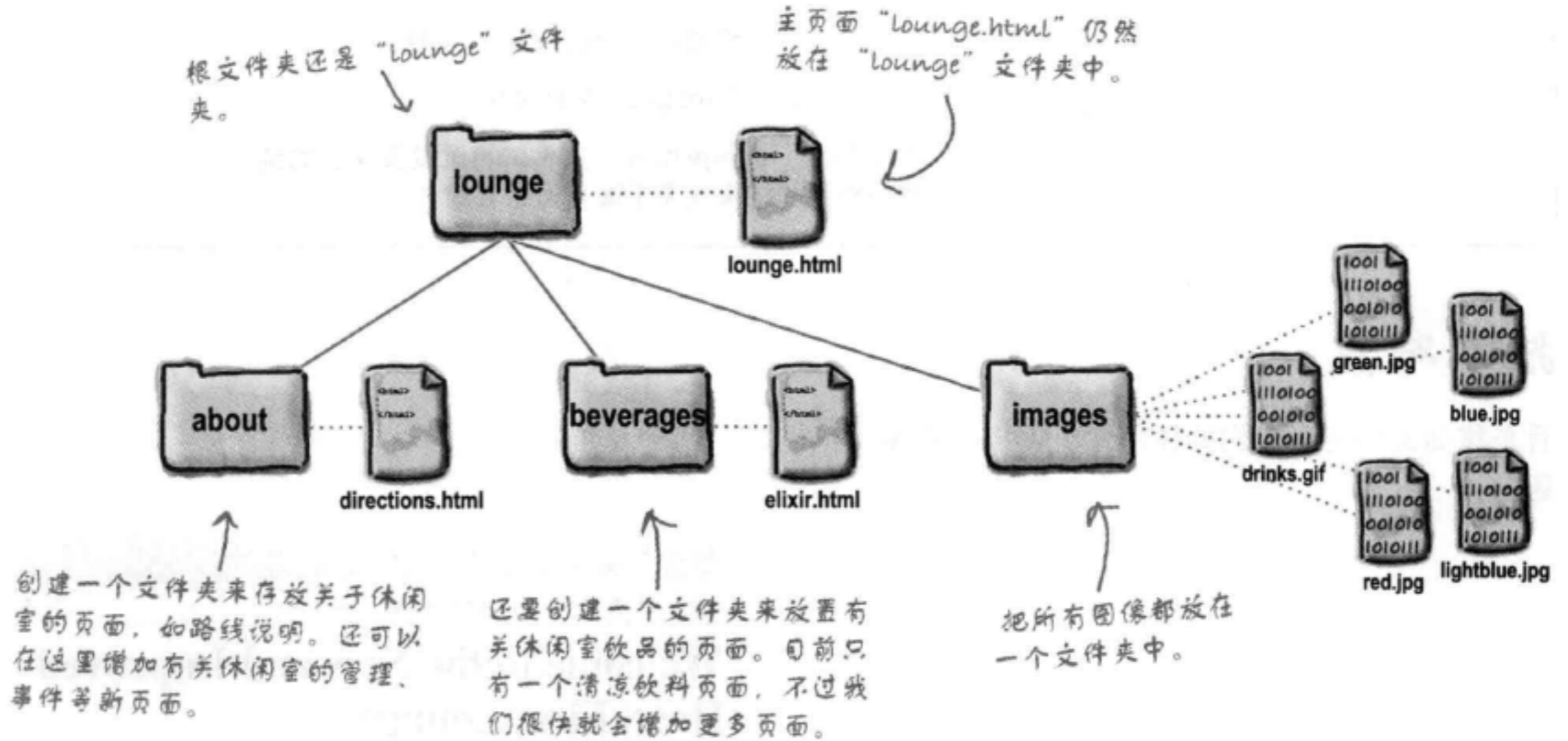
组织

开始创建更多HTML页面之前，先要组织好所有资源。到目前为止，我们一直都是将所有文件和图像放在一个文件夹中。你会发现，即使是规模很小的网站，如果把网页、图片和其他资源分别放在不同文件夹中维护，网站也会更容易管理。我们的现状是这样的：



组织休闲室……

下面让这个休闲室网站更有组织、更有条理。要记住组织网站的办法有很多，我们先从简单的做起，为页面创建一些文件夹，还要把所有图像归入一个位置。



there are no Dumb Questions

问：既然有一个文件夹来存放图像，为什么不创建另外一个“html”文件夹，把所有HTML文件都放在那个文件夹里呢？

答：也可以的。组织文件并没有“绝对正确”的方法。实际上，你会希望采用一种最适合你和用户的方式来组织网站。与大多数设计决策一样，你要选择足够灵活可以扩展的组织方案，同时还要力求简单。

问：那么，为什么不在各个其他文件夹中创建一个图像文件夹呢，比如在“about”和“beverages”下分别创建一个图像文件夹？

答：同样地，当然也可以这么做。我们希望某些图像能在多个页面上重用，所以把图像都放在根（顶级）目录下的一个文件夹里集中维护。如果你的网站需要在不同部分使用大量图像，则可以让每个分支都有自己的图像文

件夹。

问：“每个分支”？

答：可以把这样来理解描述文件夹的方式看成是倒垂的树，它最上面是根，每个向下到一个文件或文件夹的路径就是一个分支。





现在你需要创建前面所示的文件和文件夹结构。要完成下面的工作：

- 1 找到你的“lounge”文件夹，创建3个新的子文件夹，分别名为“about”、“beverages”和“images”。
- 2 把文件“directions.html”移动到“about”文件夹中。
- 3 把文件“elixir.html”移动到“beverages”文件夹中。
- 4 将所有图像都移动到“images”文件夹中。
- 5 最后，加载你的“lounge.html”文件，再试试页面上的链接。与我们得到的结果（见下面）做个比较。

技术难点


看来移动文件之后我们的休闲室页面出了很多问题。



图像没有正常显示。通常我们把这叫做“损坏的图像”。

另外，单击“elixirs”（或“detailed directions”）时情况更糟糕。我们会得到一个错误，指出无法找到这个页面。

有些浏览器会把这个错误显示为一个Web页面而不是一个对话框。



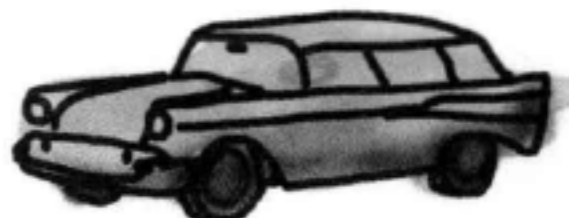
我觉得问题出在浏览器以为那些文件还在“lounge.html”所在的文件夹里。我们需要修改链接，让它们指向已经移动到新文件夹中的文件。

没错，我们要告诉浏览器这些页面的新位置。

到目前为止，我们使用的href值都指向相同文件夹中的页面。不过，网站通常会更为复杂，你还需要能够指向其他文件夹中的页面。

为此，需要追踪从页面到目标文件的路径。这可能意味着要下行一、两个文件夹，或者上行一、两个文件夹，不过不管怎样最后总能得到一个可以放在href中的相对路径。

规划你的路径……



如果你计划开着房车去度假，你会怎么做？你会找一个地图，从当前位置开始，画出一条到达目的地的路径。具体方向要由你的当前位置来定。如果你身处另一个城市，方向当然会不同，对不对？

OK，你当然可以访问 Google Maps，不过在这里还是照我们说的做吧！

要确定链接的相对路径，也是同样的道理：从包含这个链接的页面开始，然后沿着一条路径，经过一些文件夹直到找到你需要指向的那个文件。

还有其他类型的路径。后面几章会介绍。

下面来追踪几个相对路径（顺便修复休闲室的问题）。

向下链接到一个子文件夹

① 从“lounge.html”链接到“elixir.html”。

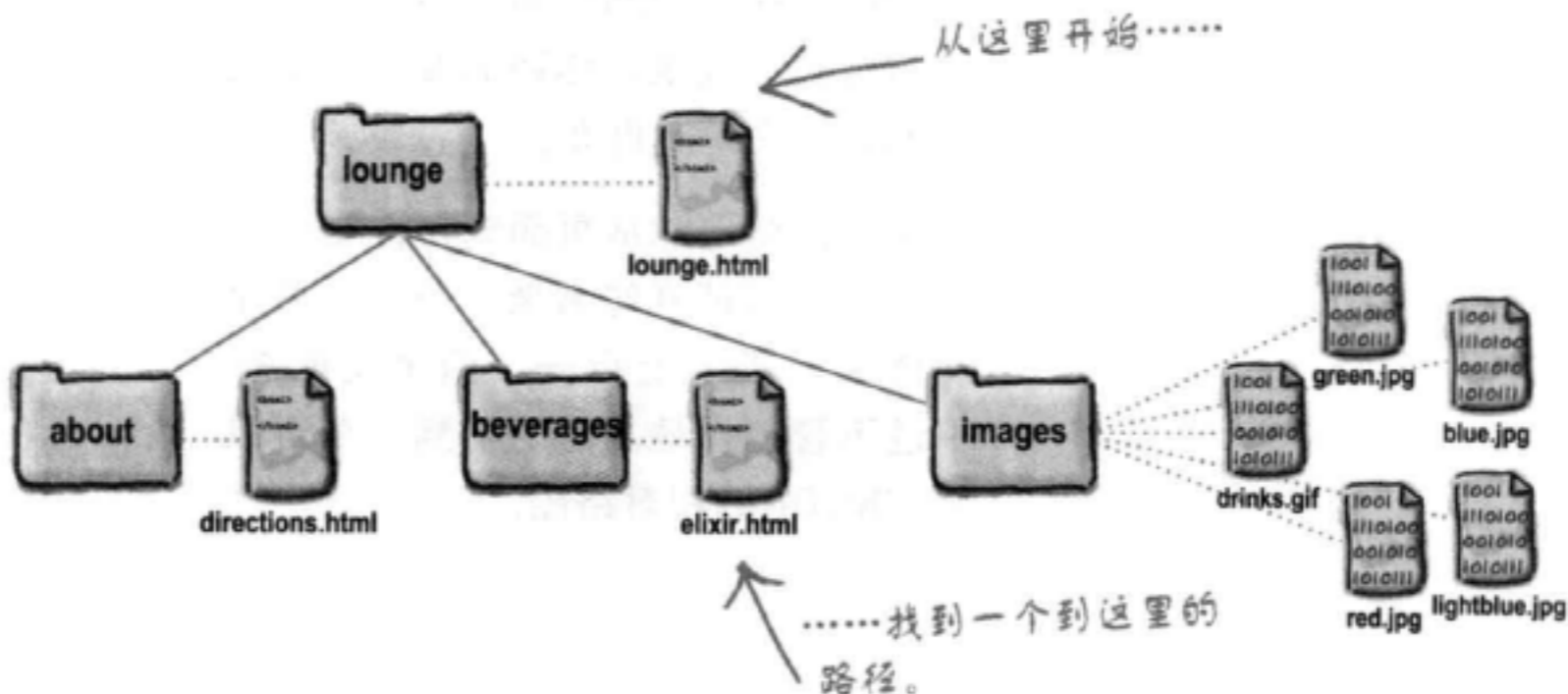
我们需要修复“lounge.html”页面中的“elixirs”链接。目前这个 `<a>` 元素是这样的：

```
<a href="elixir.html">elixirs</a>
```

现在我们只使用了文件名“elixir.html”，这就告诉浏览器要在“lounge.html”所在的相同文件夹里查找文件。

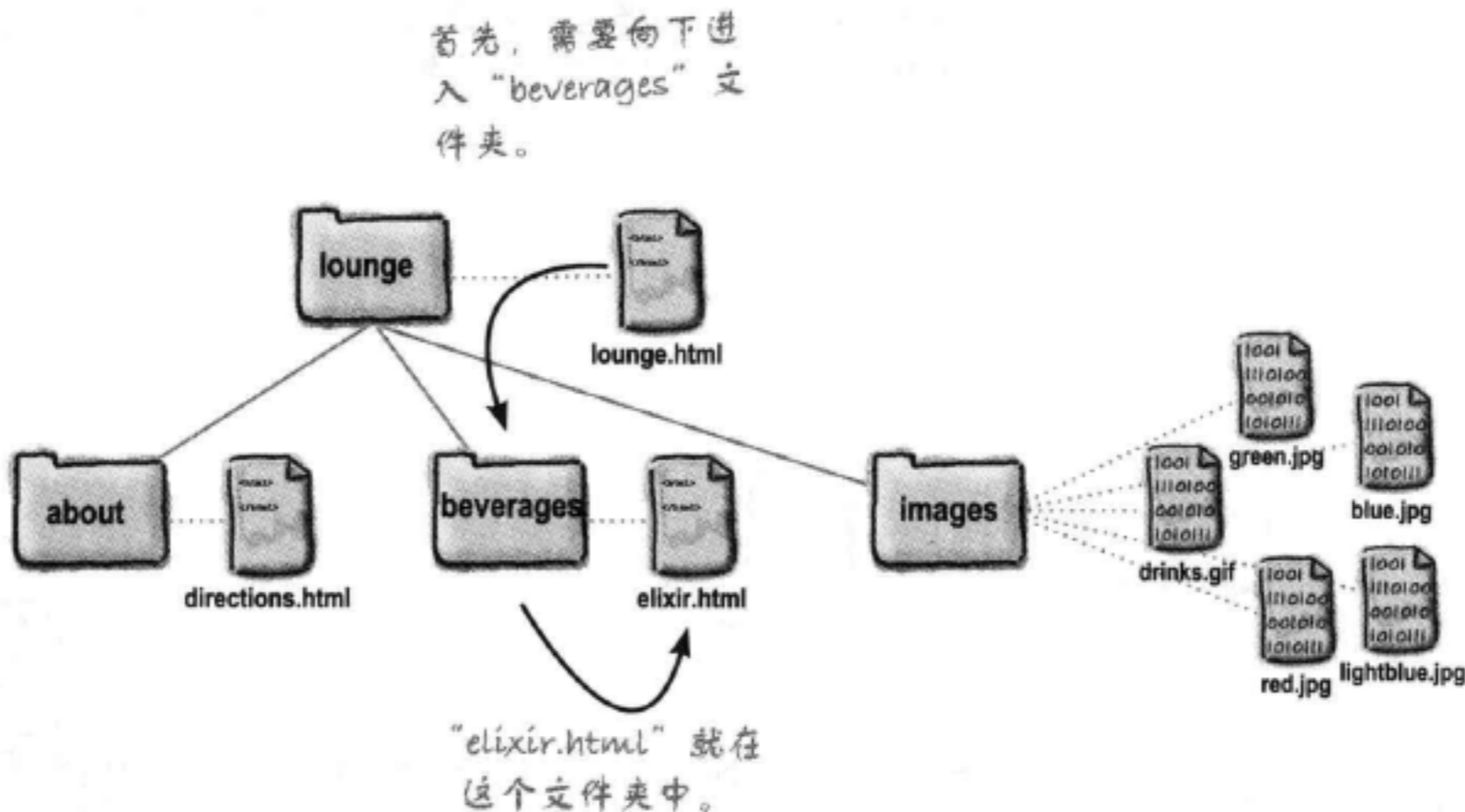
② 明确源文件和目标文件。

重新组织休闲室时，我们把“lounge.html”仍然留在“lounge”文件夹中，但把“elixir.html”移动到“beverages”文件夹中，这是“lounge”的一个子文件夹。



③ 追踪从源文件到目标文件的一个路径。

下面追踪这个路径。要从“lounge.html”文件到达“elixir.html”，需要先进入“beverages”文件夹，然后在这个文件夹中可以找到“elixir.html”。



④ 创建一个href表示我们追踪到的路径。

既然知道了路径，就需要把它写成浏览器能理解的一种格式。可以这样写路径：

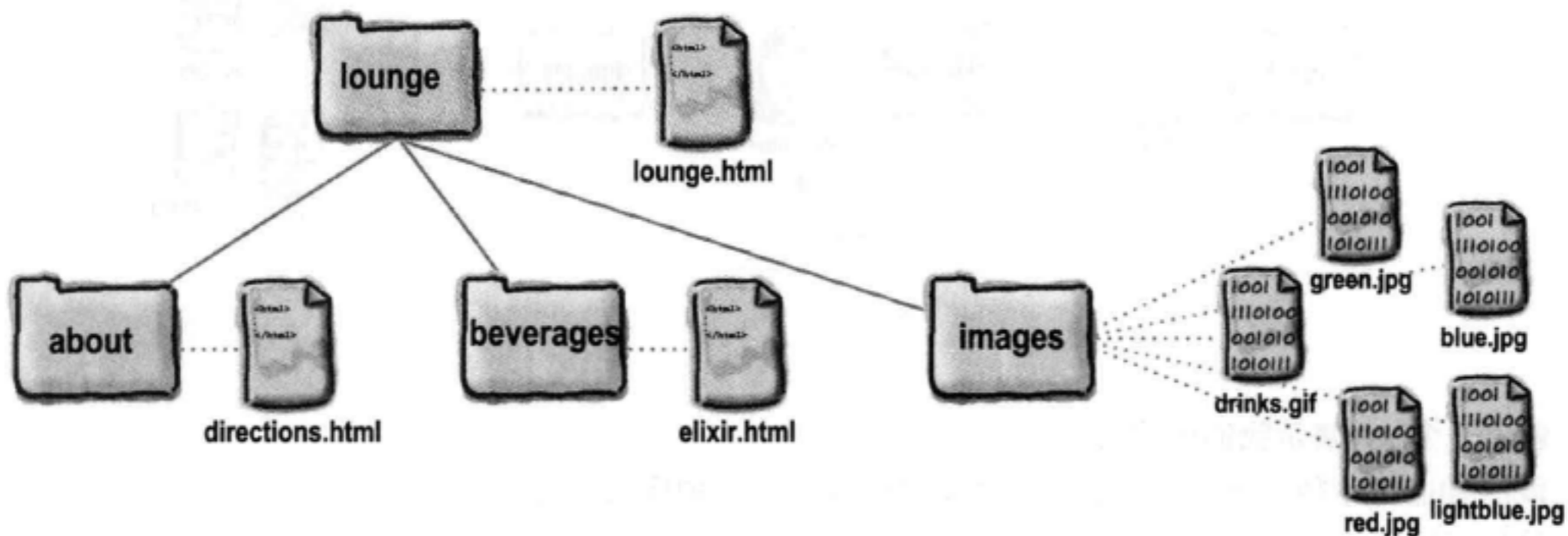


把这个相对路径放在href值中。现在单击这个链接时，浏览器就会在“beverages”文件夹中查找“elixir.html”文件。

Sharpen your pencil



该轮到你了：追踪从“lounge.html”到“directions.html”的相对路径。找到后，完成下面的<a>元素。对照这一章最后给出的答案检查你做的对不对，然后修改“lounge.html”中的这两个<a>元素。



`detailed directions`

把你的答案写在这里。

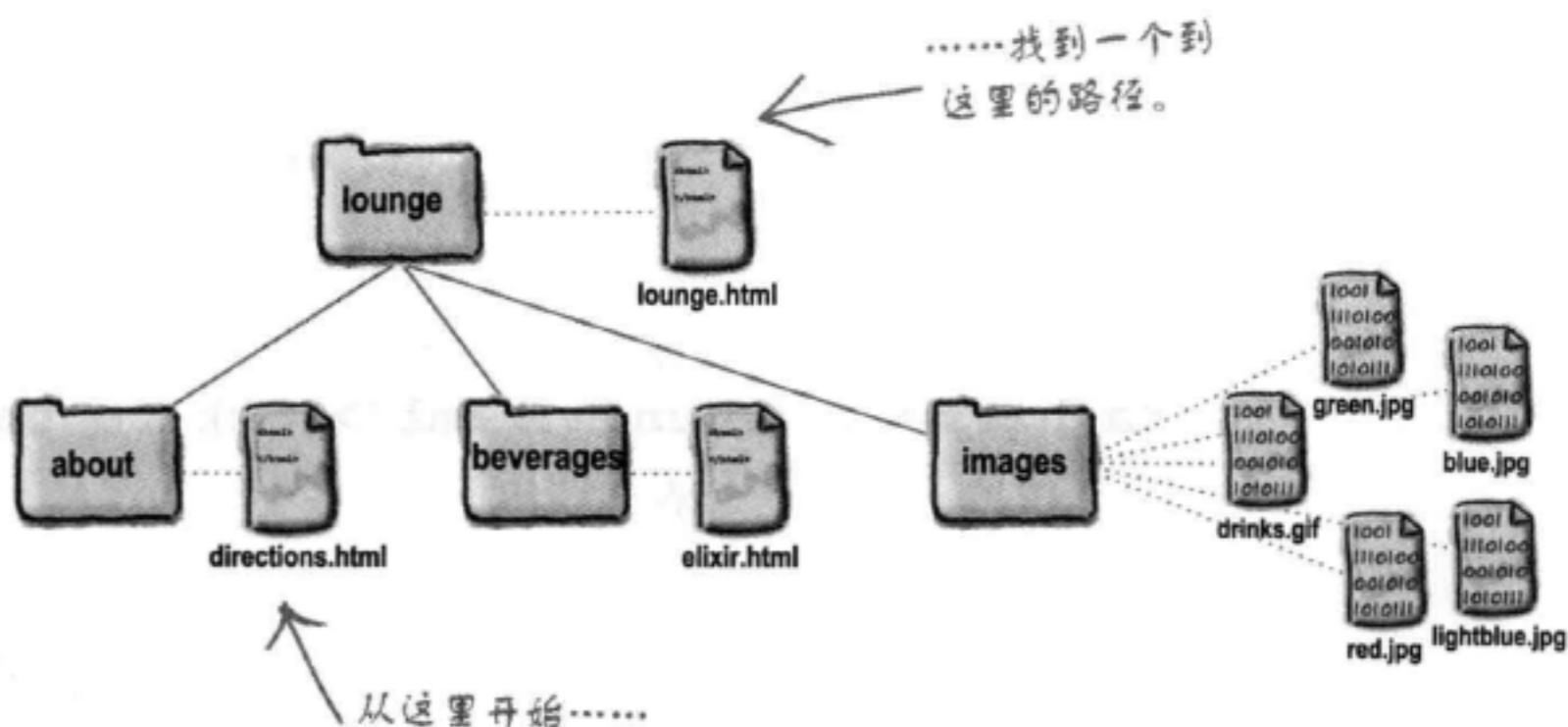
换个方向，向上链接到一个“父”文件夹

- ① 从“directions.html”链接到“lounge.html”。
现在需要修复那些“Back to the Lounge”链接。“directions.html”文件中的<a>元素是这样的：

```
<a href="lounge.html">Back to the Lounge</a>
```

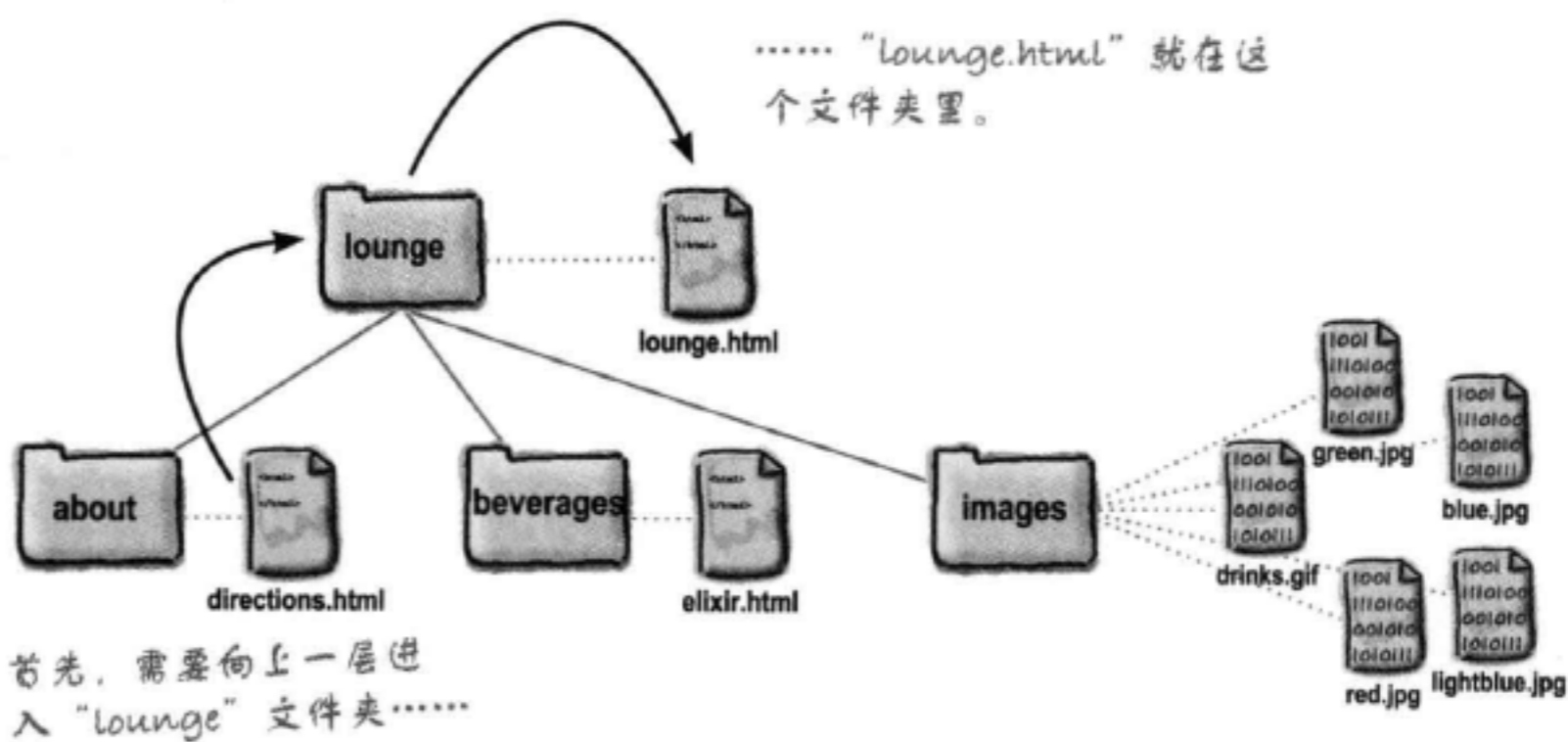
现在我们只使用了文件名“lounge.html”，这会告诉浏览器要在“directions.html”所在的相同文件夹里查找这个文件。肯定找不到。

- ② 明确源文件和目标文件。
下面来看源文件和目标文件。源文件现在是“directions.html”文件，它在“about”文件夹下。目标文件是“lounge.html”文件，它在“directions.html”所在的“about”文件夹之上。



- ③ 追踪从源文件到目标文件的一个路径。

下面来追踪这个路径。要从“directions.html”文件到“lounge.html”，需要向上进入“lounge”文件夹，可以在这个文件夹中找到“lounge.html”。



④ 创建一个href表示我们追踪到的路径。

差不多了。现在你已经知道了路径，需要把它写为浏览器能理解的一种格式。下面就来建立这个路径：

首先，需要上行一个文件夹。怎么做呢？可以用一个“..”来实现。没错，这里有两个点号。先把它写上，稍后我们还会解释。

用“/”分隔路径的各个部分。

最后，写上文件名。

`.. / lounge.html`

“..”读作“dot dot”。

把它们综合在一起……

`Back to the Lounge`

现在单击这个链接时，浏览器会在上一层文件夹中查找“lounge.html”文件。

dot dot
上, 下,
家居服, 内衣?



there are no Dumb Questions

问：什么是父文件夹？如果一个“fruit”文件夹里有一个“apples”文件夹，“fruit”就是“apples”的父文件夹，是吗？

答：完全正确。文件夹（你可能还听过有人把它们叫做目录）通常按家族关系来描述。例如，就用你的例子来说，“fruit”是“apples”的父文件夹，“apples”是“fruit”的子文件夹。如果还有另外一个文件夹“pears”作为“fruit”的一个子文件夹，那么它就是“apples”的兄弟。可以把这想象成是一个家族树。

问：OK，我懂了，不过“..”是什么呢？

答：需要告诉浏览器你要链接的文件在父文件夹中时，可以用“..”表示“向上移至父文件夹”。换句话说，对浏览器来讲，这就是“父文件夹”。

在我们的例子中，我们希望从“directions.html”（在“about”文件夹中）链接到“lounge.html”（在“lounge”文件夹中，这也是“about”的父文件夹）。所以必须告诉浏览器查看上一级文件夹，而告诉浏览器上行的方法就是使用“..”。

问：如果需要向上两级文件夹而不只是一级该怎么办呢？

答：每次要上行到父文件夹时都可以使用“..”。每一次使用“..”，就会向上到上一层父文件夹。所以，如果你想上行两级文件夹，可以输入“../..”。仍然要用“/”分隔每一部分，所以不要忘记这一点（浏览器可不知道“...”是什么意思）。

问：我上行了两个文件夹后，怎么告诉浏览器在哪里查找文件呢？

答：要把“../..”和文件名结合起来。所以，如果要链接到一个名为“fruit.html”的文件，它在向上两级的文件夹中，就可以写为“../../fruit.html”。你可能觉得我们要把“../..”称为“祖父”文件夹，不过我们一般不这么讲，而是会说“父文件夹的父文件夹”，或者简称为“../..”。

问：我能向上走多远，有限制吗？

答：你能一直向上走到网站的根目录。在我们的例子中，根目录就是“lounge”文件夹。所以，最远只能向上走到“lounge”。

问：另一个方向呢，我能向下经过多少个文件夹，有限制吗？

答：嗯，你创建了多少层文件夹，就能下行多少层。如果你创建了10层，就可以写一个带你下行10个文件夹的路径。不过我们可不推荐这么做，如果真的有那么多层文件夹，可能说明你的网站组织太过复杂！

另外，有些浏览器对路径中的字符个数会做出限制。规范中建议当心超过255个字符，但现代浏览器能支持更大的长度。不过，如果你有一个大型网站，则还是要注意这一点。

问：我的操作系统使用“\”作分隔符，我是不是应该用“\”而不是“/”？

答：不是这样的。对于网页，只能使用“/”（斜线）。不要用“\”（反斜线）。不同的操作系统使用不同的文件分隔符（例如，Windows使用“\”而不是“/”），但是在Web上，我们选择了一个通用分隔符，所有操作系统都要使用这个分隔符。所以，不管你使用Mac、Windows、Linux还是其他操作系统，在HTML的路径中都要使用“/”。



该轮到你了：追踪“Back to the Lounge”链接从“elixir.html”到“lounge.html”的相对路径。这与“directions.html”文件中的那个返回主页面的链接有什么不同？

°回辟寺等LH2 ‘耕一业身器 :孝景

修复那些损坏的图像……

休闲室基本上已经走入正轨了，接下来只需要修复那些不能正常显示的图像。

我们还没有深入分析元素（后面几章会详细介绍），不过现在你只需要知道元素的src属性有一个相对路径值，类似于href属性。

以下是“lounge.html”文件中的image元素：

```

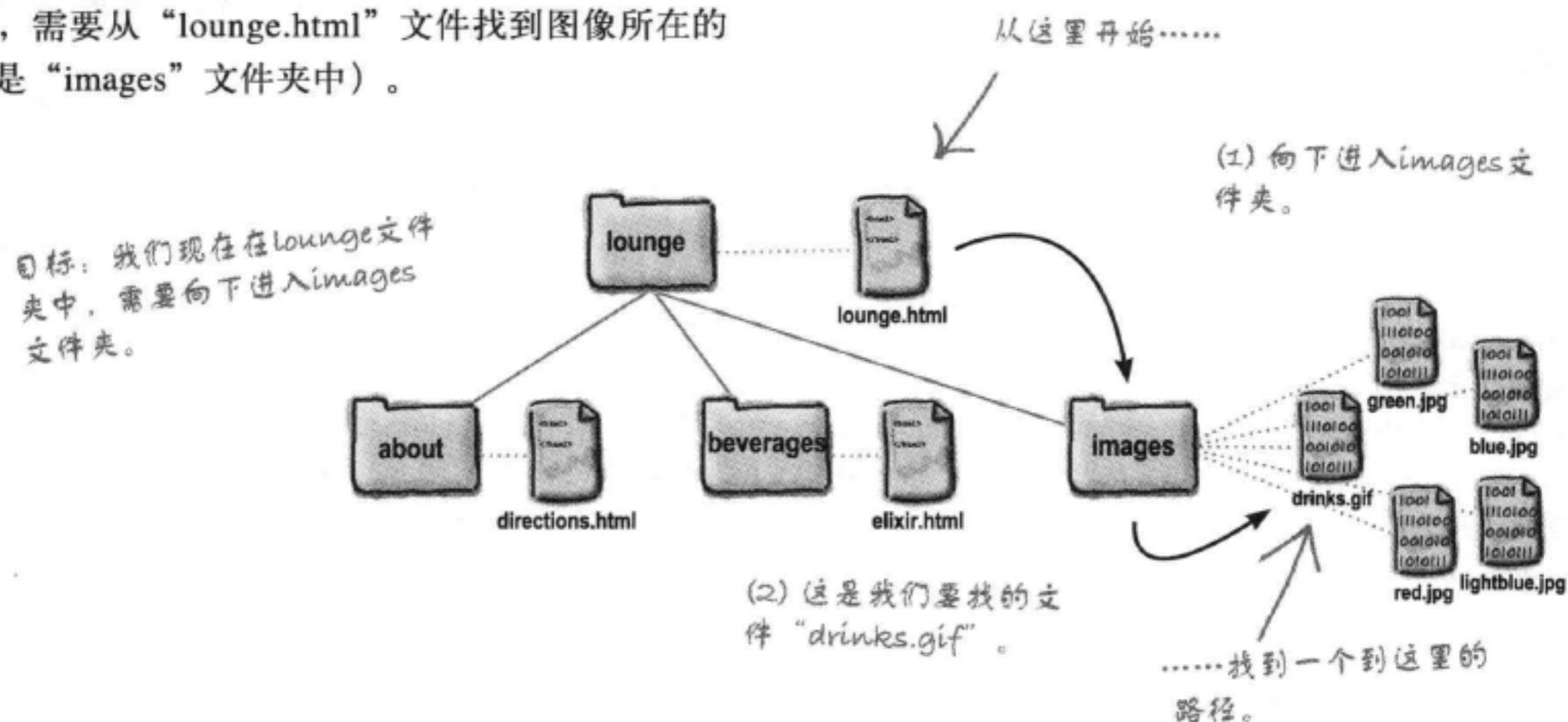
```

这是相对路径，告诉浏览器图像的位置。与指定<a>元素中的href属性类似，我们也要对它做同样的处理。



查找从“lounge.html”到“drinks.gif”的路径

要查找路径，需要从“lounge.html”文件找到图像所在的位置（也就是“images”文件夹中）。



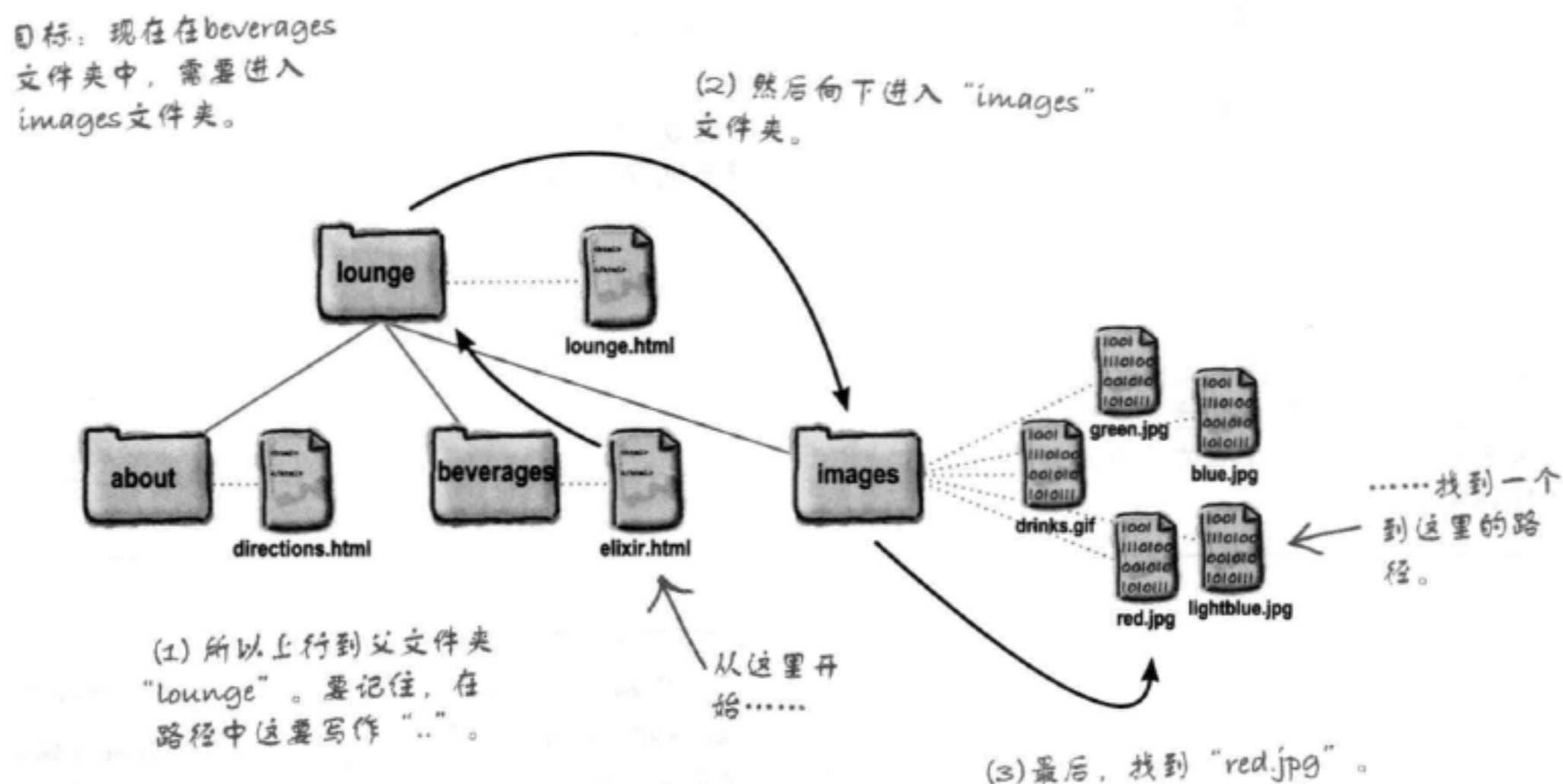
所以，把(1)和(2)综合在一起，可以得到路径“images/drinks.gif”，或

```

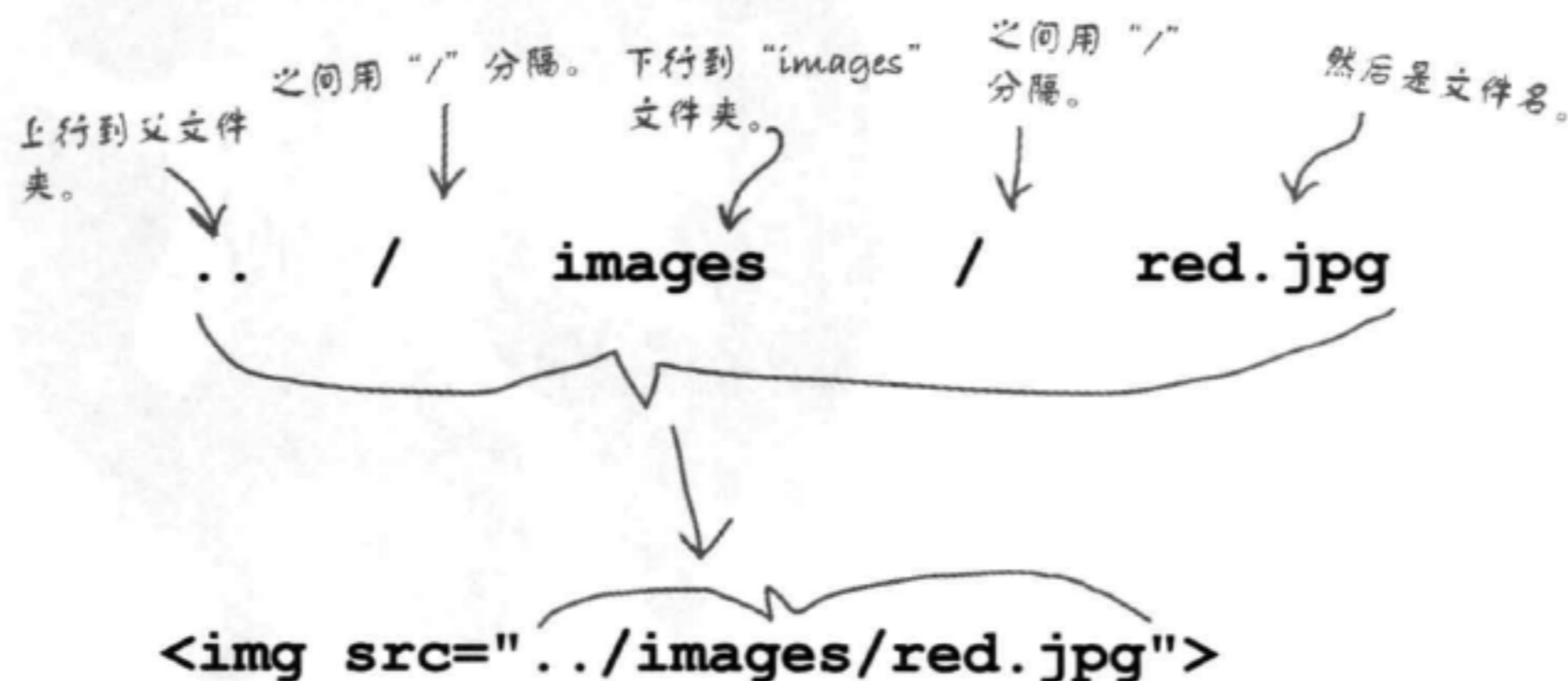
```

查找从“elixir.html”到“red.jpg”的路径

清凉饮料页面包含多种饮料的图像：“red.jpg”、“green.jpg”、“blue.jpg”等。下面来确定到达“red.jpg”的路径，其余图像的路径都是类似的，因为它们都在同一个文件夹中：



所以，把(1)、(2)和(3)综合在一起，可以得到：



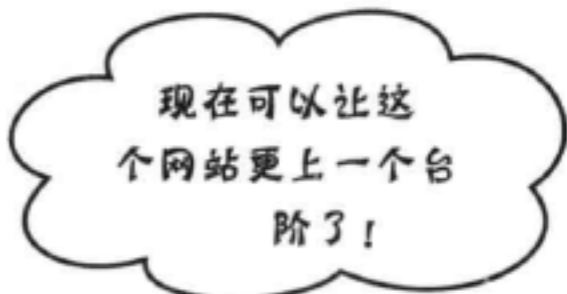
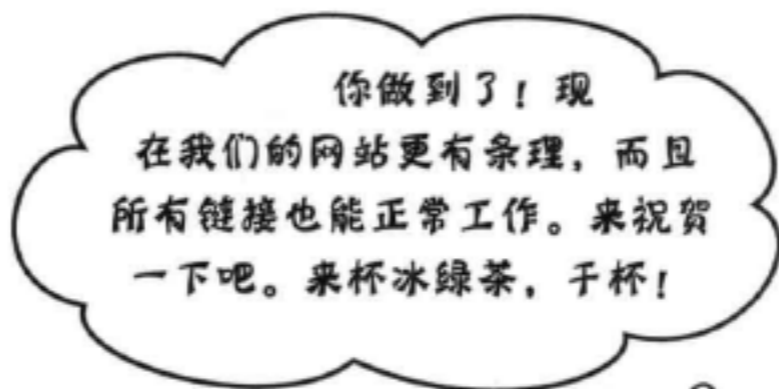


现在已经修复了重组休闲室时破坏的所有链接，不过还需要修复“lounge.html”和“elixir.html”文件中的图像。要完成下面的工作：

- 1 在“lounge.html”中，更新图像的src属性值为“images/drinks.gif”。
- 2 在“elixir.html”中，更新图像的src属性，在每个图像名前面加上“../images/”。
- 3 保存这两个文件，并在浏览器中加载“lounge.html”。现在应该能在页面间顺畅地导航，图像也能正常显示。



又及，如果还有问题，文件夹“chapter2/completelounge”中包含了一个可用的休闲室版本。对照这个版本仔细检查你的文件哪里有问题。





BULLET POINTS

- 想从一个页面链接到另一个页面时，要使用[<a>](#)元素。
- [<a>](#)元素的href属性指定了链接的目标文件。
- [<a>](#)元素的内容是链接的标签。这个标签就是你在网页上看到的链接文本。默认地，这个标签会有下划线，指示这是可以单击的。
- 文字或图像都可以用作链接的标签。
- 单击一个链接时，浏览器会加载href属性中指定的Web页面。
- 可以链接到相同文件夹中的文件，也可以链接到其他文件夹中的文件。
- 相对路径是相对于链接的源Web页面指向网站中其他文件的一个链接。就像在地图上一样，终点总是相对于起点。
- 使用“..”可以链接到源文件上一层文件夹中的一个文件。
- “..”表示“父文件夹”。
- 记住要用“/”（斜线）字符分隔路径中的各个部分。
- 指向一个图像的路径不正确时，会在Web页面上看到一个损坏的图像。
- 为网站选择的文件名和文件夹名中不要使用空格。
- 最好在构建网站初期组织网站文件，这样就不用在网站升级时修改一大堆的路径了。
- 组织网站有很多方法，具体如何组织由你决定。



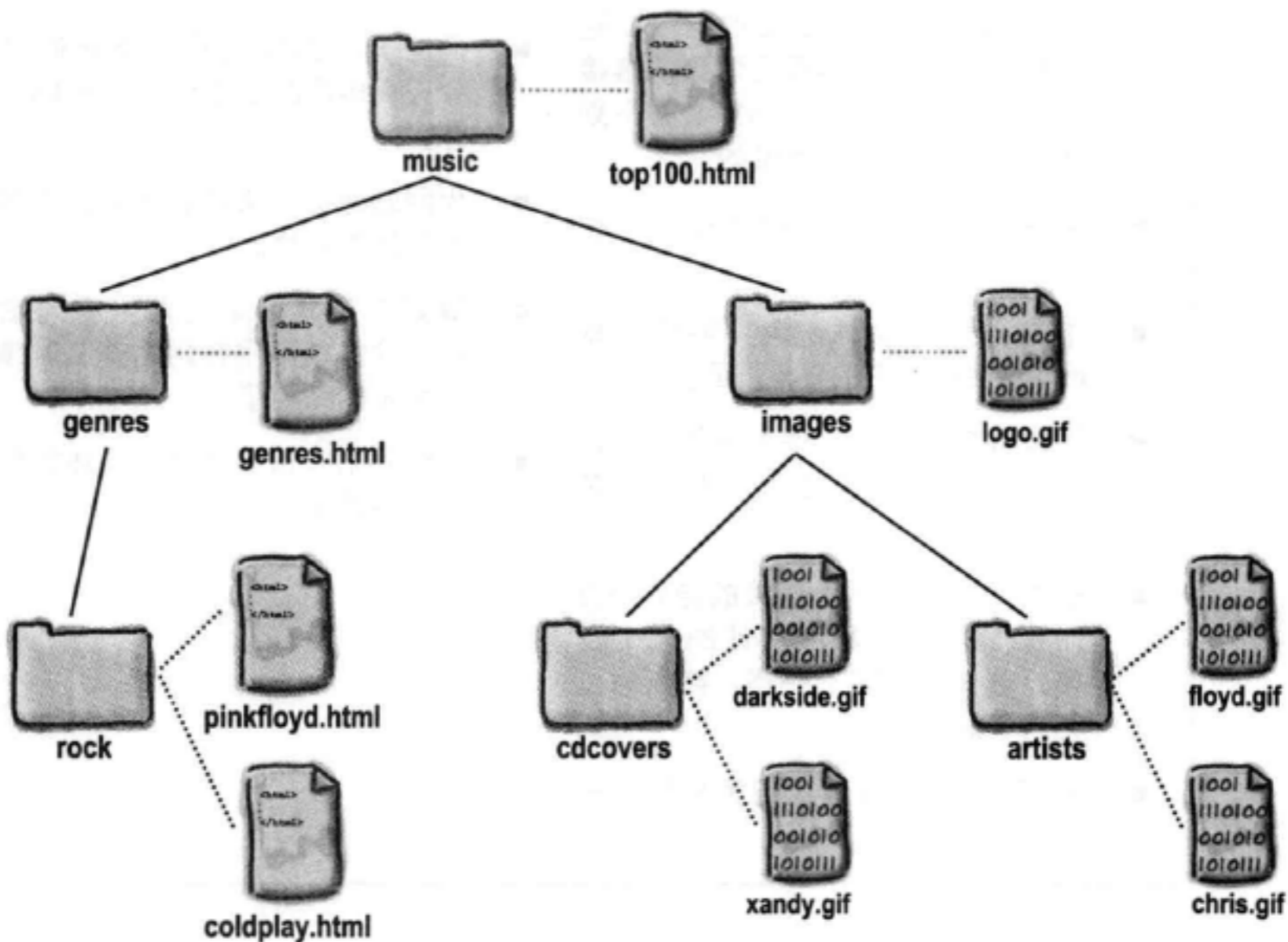
相对性的重大挑战

现在给你一个机会来测试你的相对性技能。我们为100强唱片建立了一个网站，所有文件都放在一个名为“music”的文件夹里。这个文件夹中有HTML文件、其他文件夹，还有图像。你的任务是找出我们需要的相对路径，以便从我们的Web页面链接到其他Web页面和文件。

这一页给出了这个网站的结构，下一页会测试你的能力，你会看到需要完成的任务。你要为每一对源文件和目标文件建立正确的相对路径。如果能成功地找出所有相对路径，则你将在相对路径赛场上夺冠。

祝你好运！

可以直接在这个网站结构图上画出路径。



比赛就要开始了。

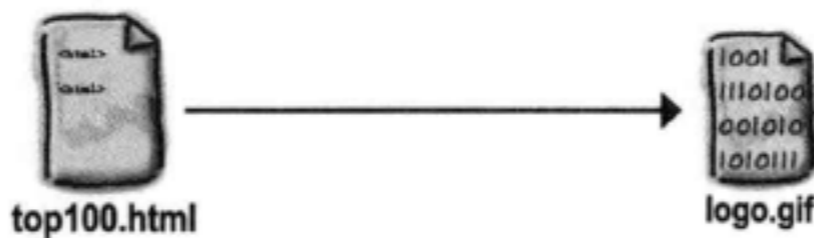
各就各位……预备……写！

例子

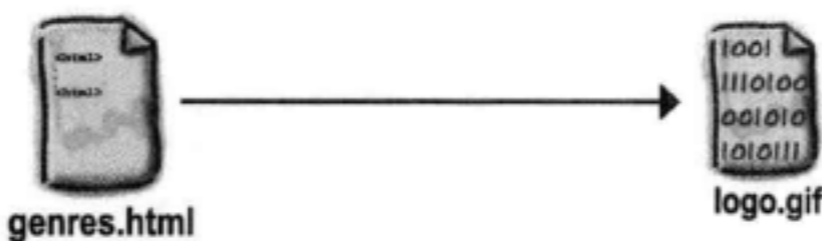


“top100.html”在“music”文件夹中，所以要找到“genres.html”，我们需要下行到“genres”文件夹中。

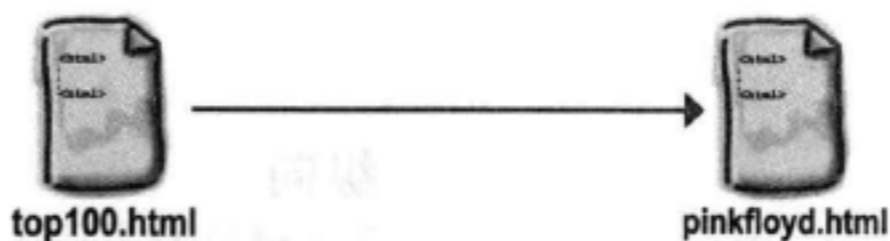
第一轮



第二轮



第三轮



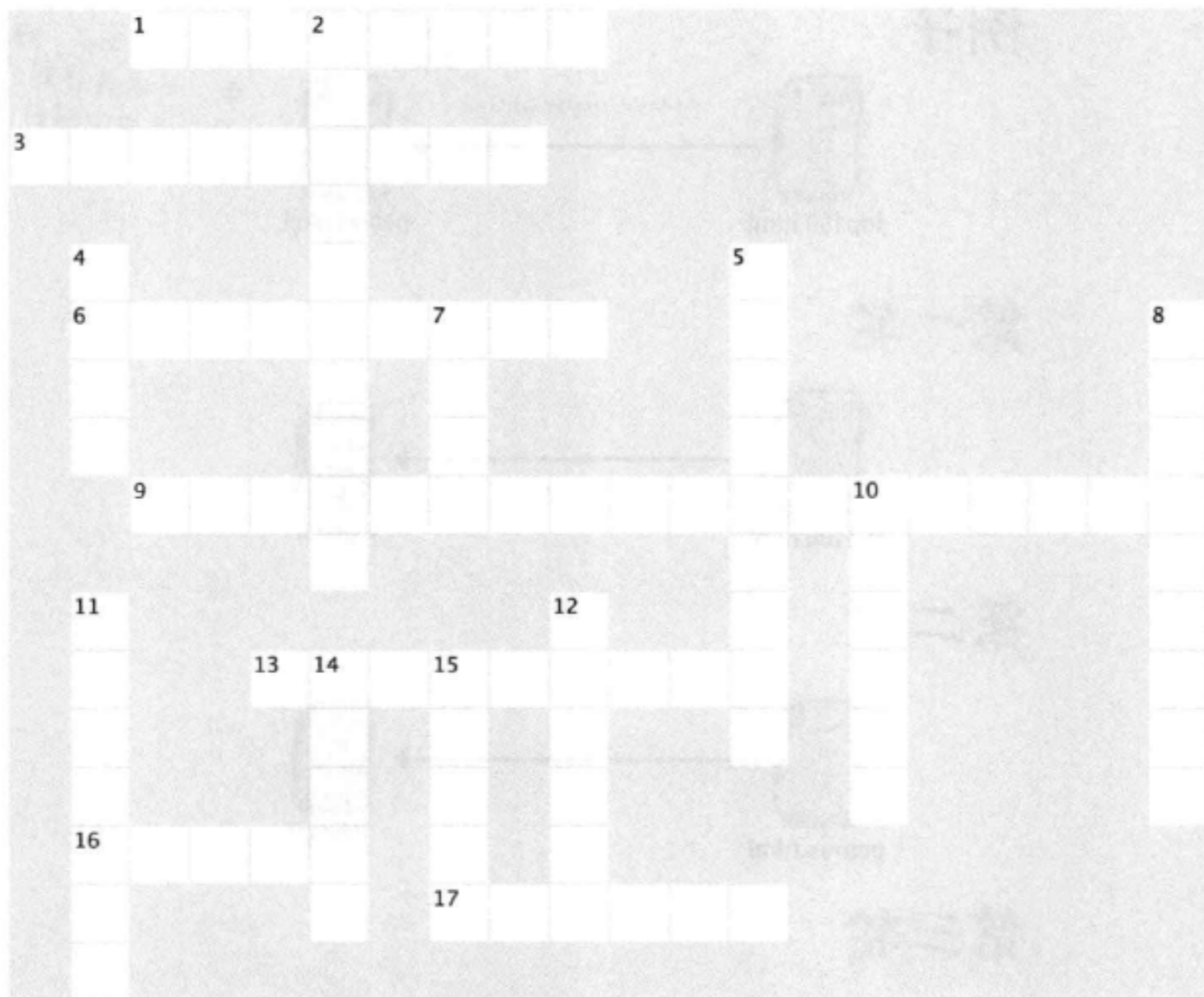
加试赛





HTML填字游戏

填字游戏对你学习HTML有什么帮助？嗯，所有这些词都与HTML有关，而且都在这一章出现过。另外，线索里有一些脑筋急转弯，可以帮你从不同角度将HTML牢牢记住！



横向

1. “../myfiles/index.html” 是这种链接。
3. 文件夹的别名。
6. blue drink的口味。
9. href表示的意思。
13. <a>和之间的所有内容。
16. 可以放在<a>元素中，就像文本一样。
17. 读作“..”。

纵向

2. href和src是两个_____。
4. Web上工作最勤奋的属性。
5. 与href押韵。
7. 网站的顶层文件夹。
8. HTML中的“HT”。
10. 健康饮料。
11. 同级的文件夹。
12. 使用..可以到达_____目录。
14. <a>标记之间的文本作为一个_____。
15. 子文件夹也称为_____。



Exercise Solution

在清凉饮料页面的最下面增加一个标签为“Back to the Lounge”的链接，指回到“lounge.html”。下面是我们的答案。

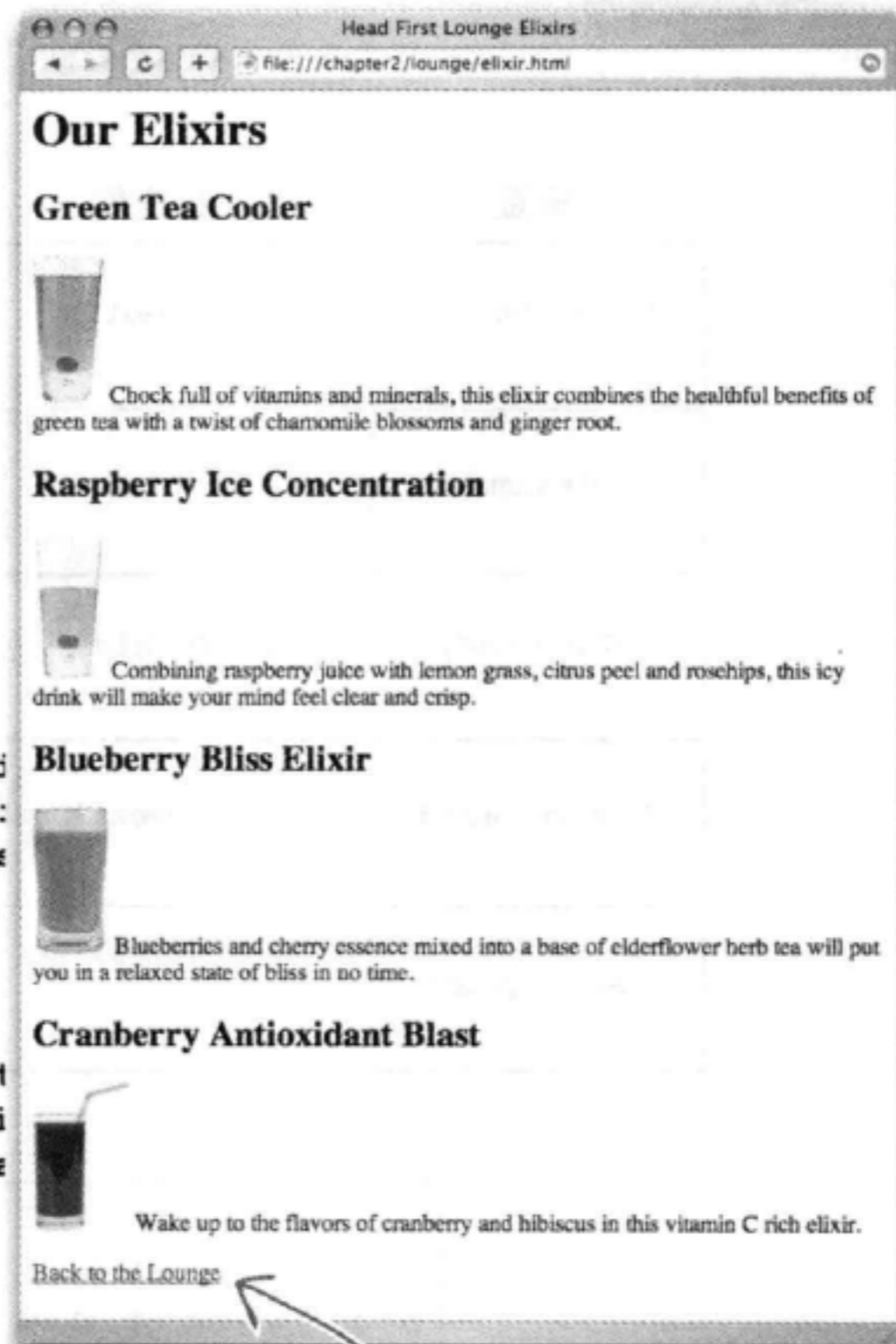
```

<html>
  <head>
    <title>Head First Lounge Elixirs</title>
  </head>
  <body>
    <h1>Our Elixirs</h1>

    <h2>Green Tea Cooler</h2>
    <p>
      
      Chock full of vitamins and minerals, this elixir combines the healthful benefits of green tea with a twist of chamomile blossoms and ginger root.
    </p>
    <h2>Raspberry Ice Concentration</h2>
    <p>
      
      Combining raspberry juice with lemon grass, citrus peel and rosehips, this drink will make your mind feel clear and crisp.
    </p>
    <h2>Blueberry Bliss Elixir</h2>
    <p>
      
      Blueberries and cherry essence mixed into a base of elderflower herb tea will put you in a relaxed state of bliss in no time.
    </p>
    <h2>Cranberry Antioxidant Blast</h2>
    <p>
      
      Wake up to the flavors of cranberry and hibiscus in this vitamin C rich elixir.
    </p>
    <p>
      <a href="lounge.html">Back to the Lounge</a>
    </p>
  </body>
</html>

```

为力求简洁，这里把链接单独放在一个段落中。下一章还会讨论更多有关的内容。



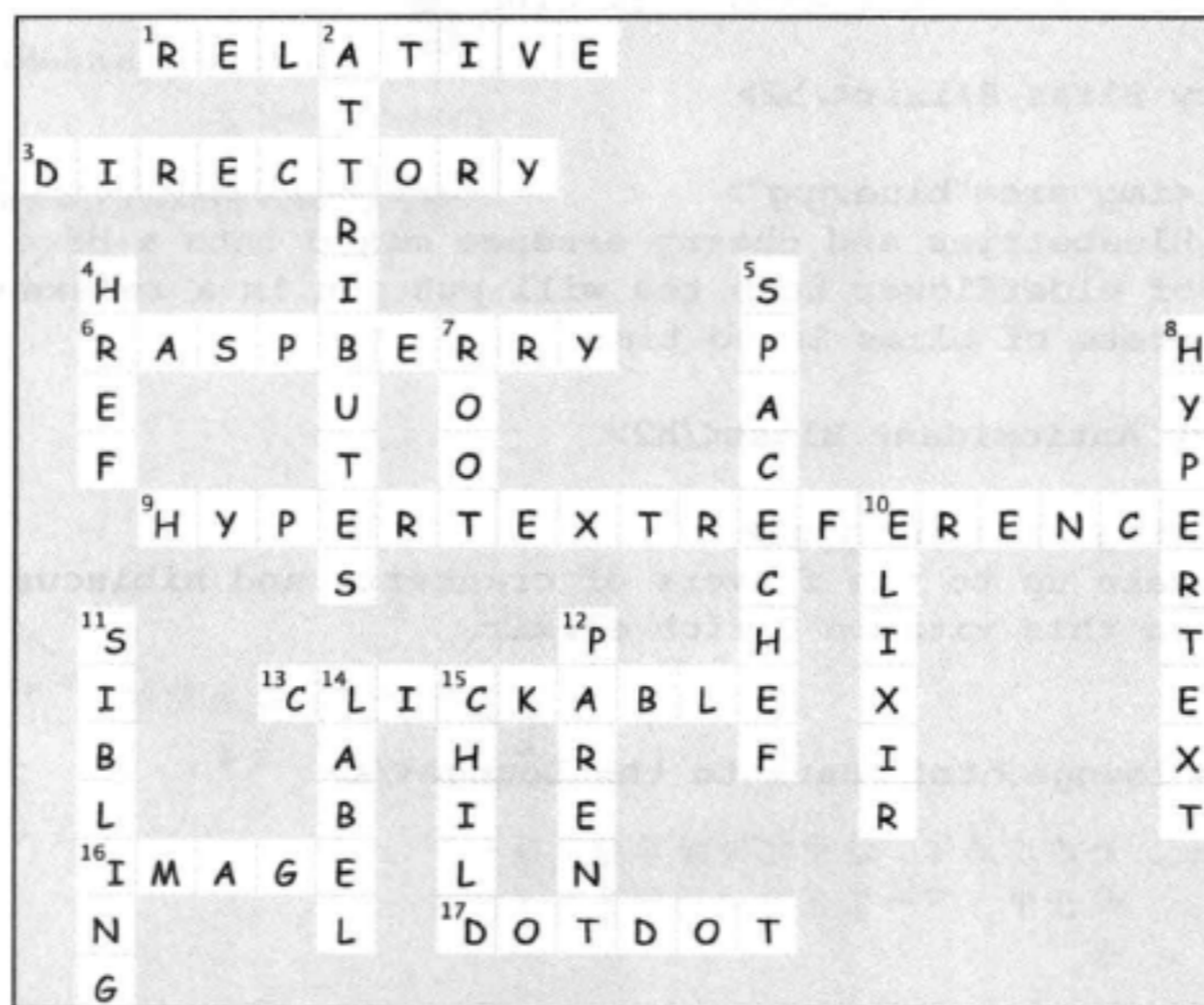
这是新加的<a>元素，指回到休闲室页面。



练习答案



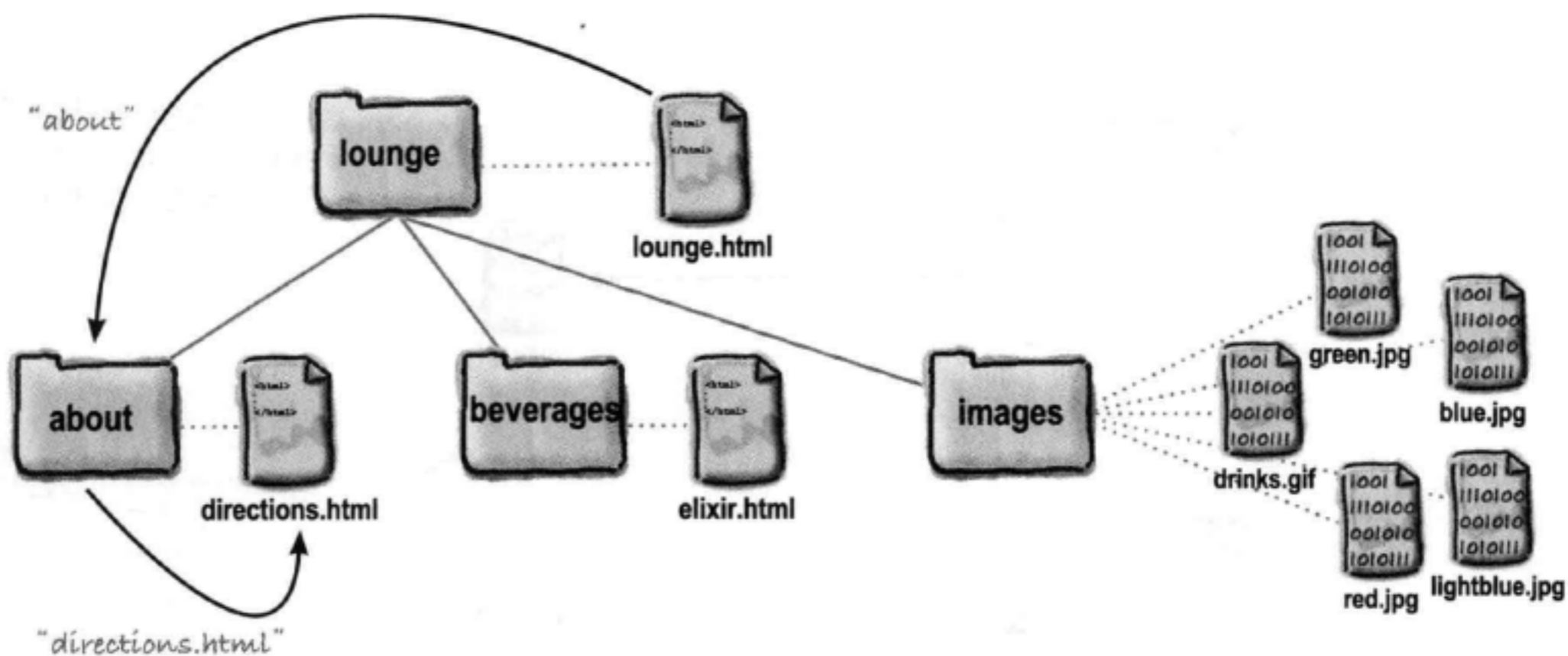
标签	目标文件	元素
Hot or Not?	hot.html	<code>Hot or Not?</code>
Resume	cv.html	<code>Resume</code>
Eye Candy	candy.html	<code>Eye Candy</code>
See my mini	mini-cooper.html	<code>See my mini</code>
let's play	millionaire.html	<code> let' s play </code>



Sharpen your pencil Solution

追踪从“lounge.html”到“directions.html”的相对路径。找到后，完成下面的<a>元素。

下面是我们的答案。“lounge.html”中的两个<a>元素都改了吗？



`detailed directions`

答案写在这里



相对性的重大挑战答案

第一轮



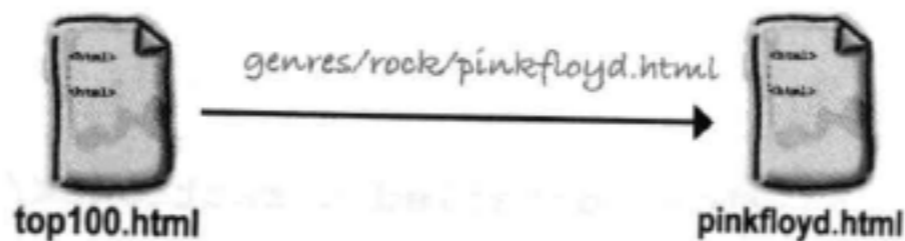
top100.html在music文件夹中，所以要找到logo.gif，需要向下进入images文件夹。

第二轮



genres.html在genres目录中，所以要找到logo.gif，首先必须上行到music文件夹，然后向下进入images文件夹。

第三轮



从top100.html需要下行到genres文件夹，然后进一步下行到rock目录，在这里可以找到pinkfloyd.html。

加试赛



这一题比较难。从coldplay.html（它在rock文件夹中），必须上行两层文件夹进找到music目录，然后下行进入images，最后再下行到artists目录，在这里才能找到图像chris.gif。哟，真不容易！

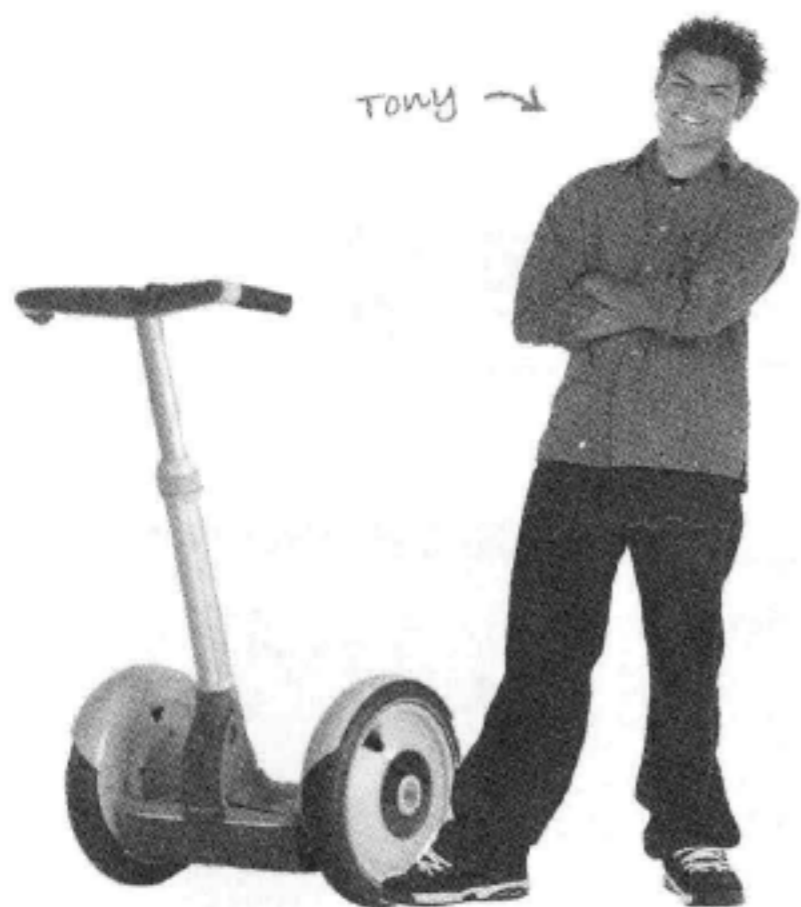
3 构建模块

Web页面建设

Betty, 我们最好找些安全帽。这里有一个建筑工地, 这些网页很快就会建起来!



我听说这本书会教我具体创建Web页面? 你已经学到不少了: 标记、元素、链接、路径……不过, 如果不学以致用, 创建一些一流的Web页面, 这些知识就毫无意义。这一章我们要把重点放在建设上: 你要从概念到蓝图来设计Web页面, 浇注地基, 完成建设, 甚至最后还可以加几笔润色。带上安全帽, 系好安全带, 我们会增加一些新工具, 让你掌握一些更深入的内部知识, 有了这些知识, 你也能成为这个行业的佼佼者。

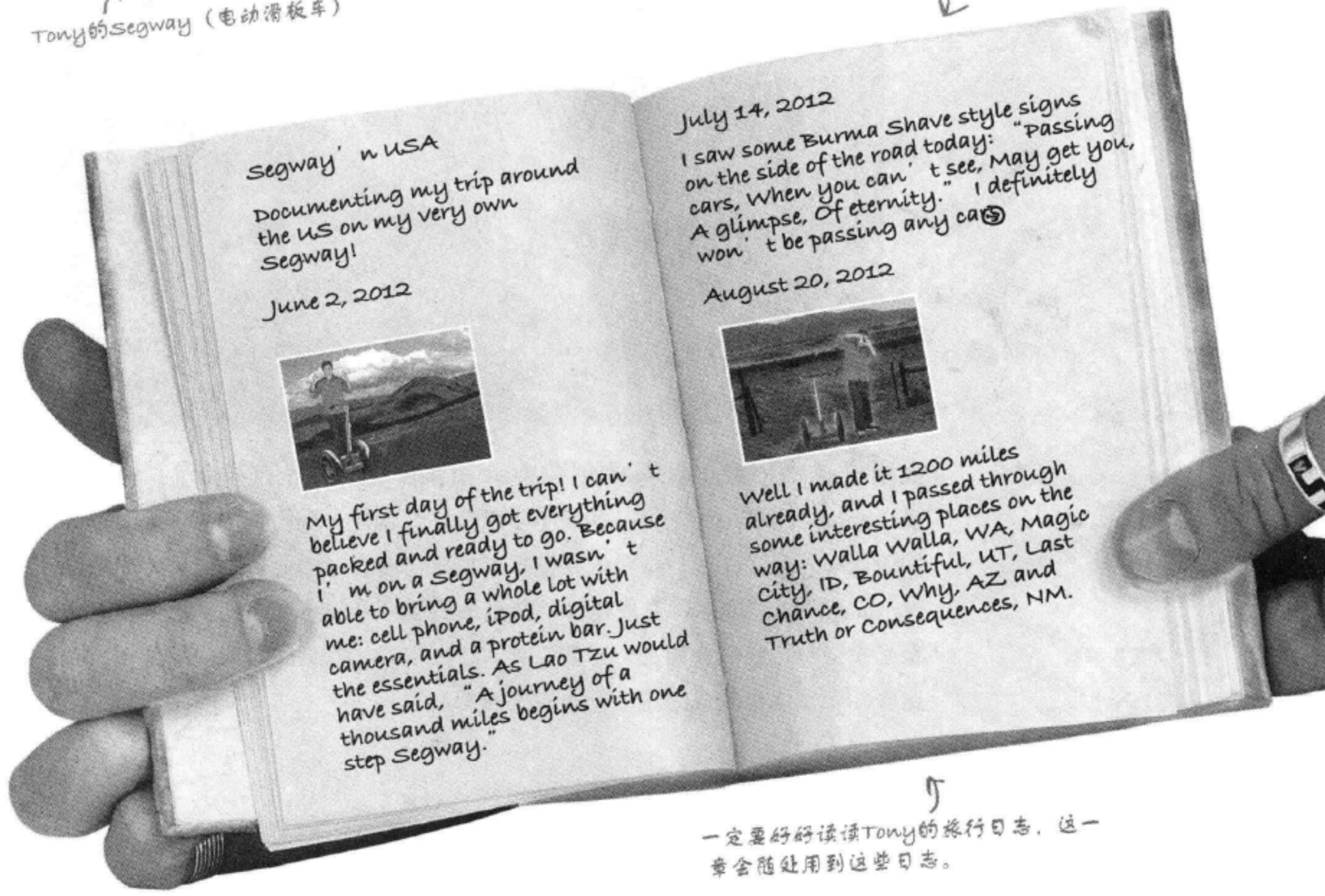


Tony →

↑
Tony的segway (电动滑板车)

要痛痛快快地玩我的新滑板车 (Segway)，再没有比骑着它上马路驰骋更过瘾的了！我要骑着它周游整个美国，还会把路上的所见所闻记在我的旅行日志里。我想把它们放到网页上，这样我的朋友和家人就能看到了。

↓
Tony的旅行日志



Segway' n USA

Documenting my trip around the US on my very own Segway!

June 2, 2012

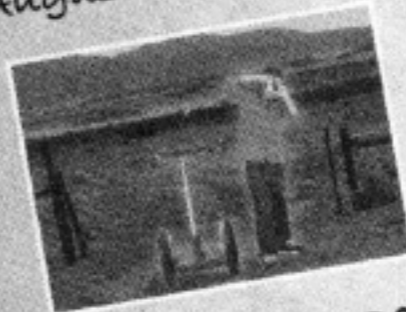


My first day of the trip! I can't believe I finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cell phone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one step Segway."

July 14, 2012

I saw some Burma Shave style signs on the side of the road today: "Passing cars, when you can't see, may get you. A glimpse, of eternity." I definitely won't be passing any car!

August 20, 2012



Well I made it 1200 miles already, and I passed through some interesting places on the way: Walla Walla, WA, Magic City, ID, Bountiful, UT, Last Chance, CO, Why, AZ, and Truth or Consequences, NM.

↑
一定要好好读读Tony的旅行日志，这一章会随处用到这些日志。

从日志到网站，以12迈的速度出发

← segway的最高时速。
建议

Tony在全力准备骑着他的Segway周游美国，已经无暇顾及其他了。你为什么不帮他一把，为他创建一个网页，好吗？

你要完成下面的工作：

- 1 首先，画一个粗略的日志草图，这将作为页面设计的基础。
- 2 接下来，创建HTML的基本构建模块（<h1>、<h2>、<h3>、<p>等），把你的草图翻译成HTML页面的略图（或蓝图）。
- 3 一旦有了蓝图，接下来就可以把它翻译成真正的HTML了。
- 4 最后，在完成的基本页面上还可以加一些改进，顺便认识一些新HTML元素。

Sharpen your pencil



先停一下！翻开下一页之前先完成这个练习。

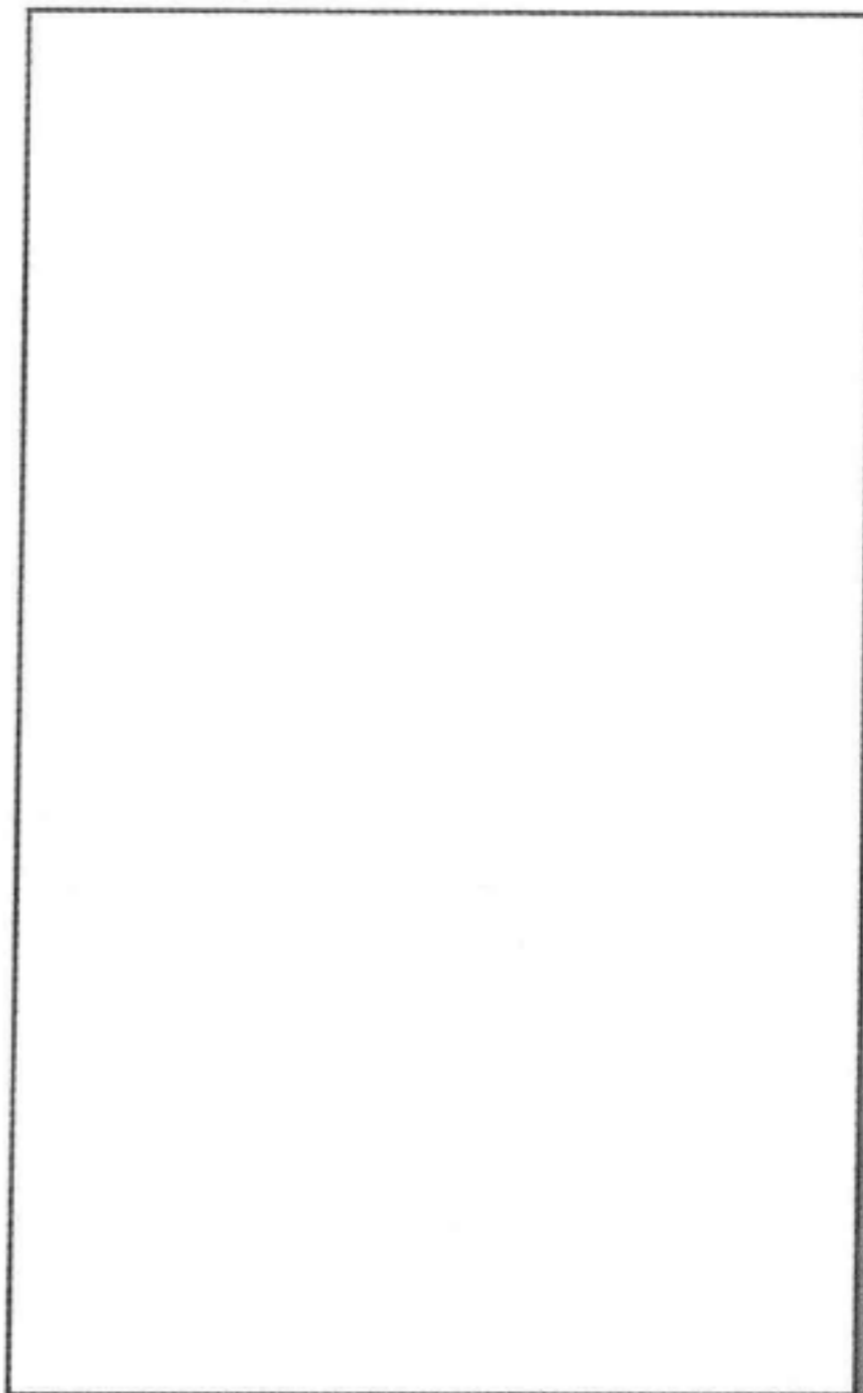
仔细研究Tony的日志，想想看如何把这些信息展示在Web页面上。

在右边画出这个页面。不需要很漂亮、很精致，只需要一个大致的草图就可以了。假设他的所有日志都放在一个页面上。

从下面几个方面来考虑：

- 从大的结构元素来考虑页面的构成：标题、段落、图像，等等。
- 有没有办法把他的日志改为更适合在Web上发布。

在这里画出你的草图。



粗略的设计草图

Tony的日志看起来很像一个网页，要创建设计草图，我们要做的就是把他的所有日志都放在一个页面上，并安排好总体组织布局。看起来如果Tony某一天写日志，他会先写当天的日期作为标题，然后可能有一个图片，最后是对当天见闻的描述。下面来看这个草图……

他还为旅行日志写了一个介绍。我们将把它作为一个小段落放在最上面。

每天Tony创建一个日志，包括日期，往往还有一个图片，最后是当天见闻的描述。所以要有个标题、一个图像和另外一个文本段落。

有时他没有加图片。在这个日志中，只有一个标题（日期）和当天见闻的描述。

第3个日志与第1个看上去很类似，也有一个标题、一个图像和一个段落。

与Tony手写的日志不同，我们的页面长度是无限的，所以可以把很多个日志放在同一个web页面上。他的朋友和家人只需要使用滚动条来浏览他的日志……

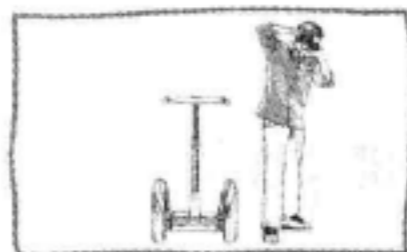
不过，注意我们把日志的顺序倒换了一下，这里按从最新到最早的顺序排列。这样一来，最新的日志会出现在页面最上面，用户不用滚动就能直接看到。

Tony的旅行日志有一个大标题“Segway” n USA”，我们要把它作为标题放在页面顶部。

Segway' n USA

Documenting my trip around the US on my very own Segway!

August 20, 2012

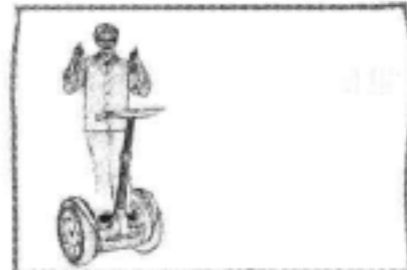


Well I made it 1200 miles already, and I passed through some interesting places on the way: Walla Walla, WA, Magic City, ID, Bountiful, UT, Last Chance, CO, Why, AZ and Truth or Consequences, NM.

July 14, 2012

I saw some Burma Shave style signs on the side of the road today: "Passing cars, When you can't see, May get you, A glimpse, Of eternity." I definitely won't be passing any cars.

June 2, 2012



My first day of the trip! I can't believe finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cell phone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one Segway."

从草图到略图

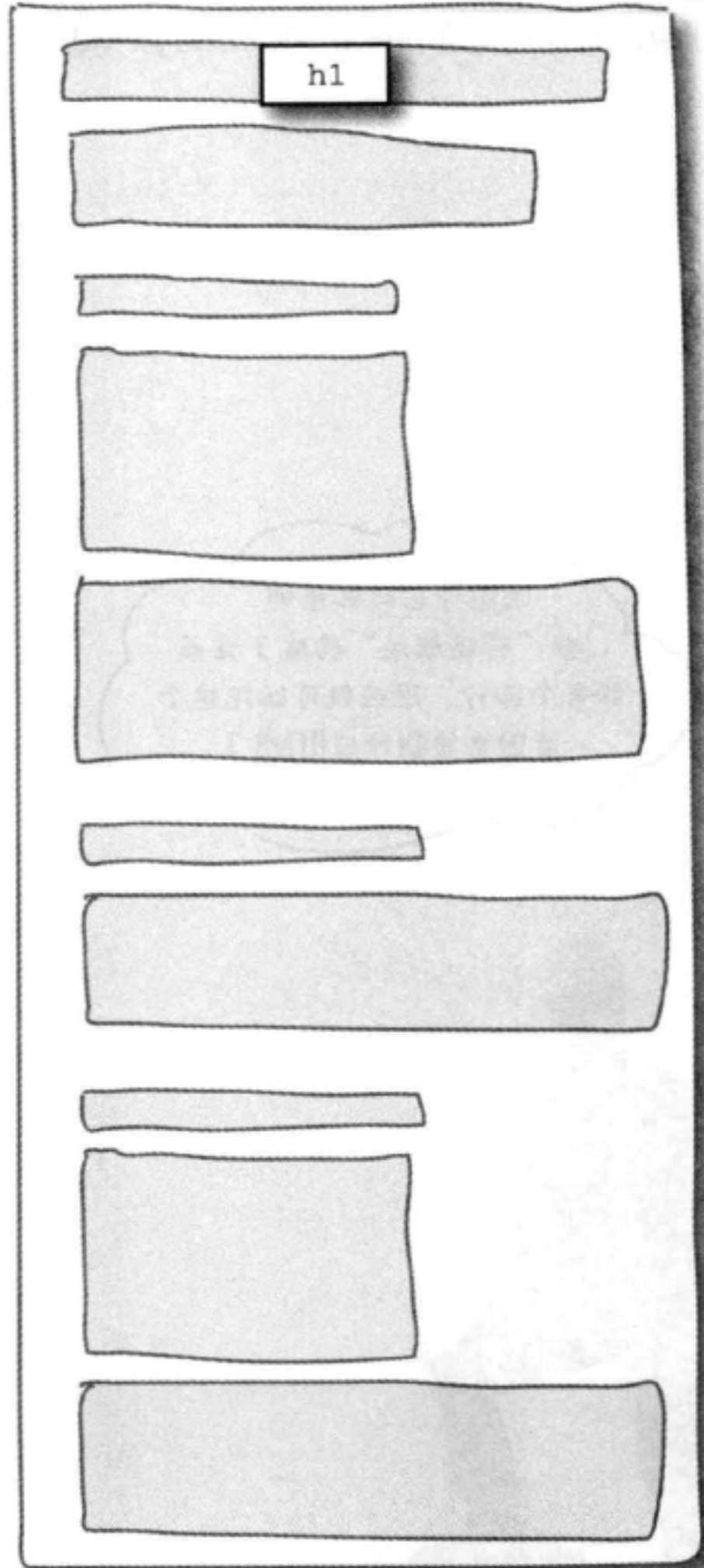
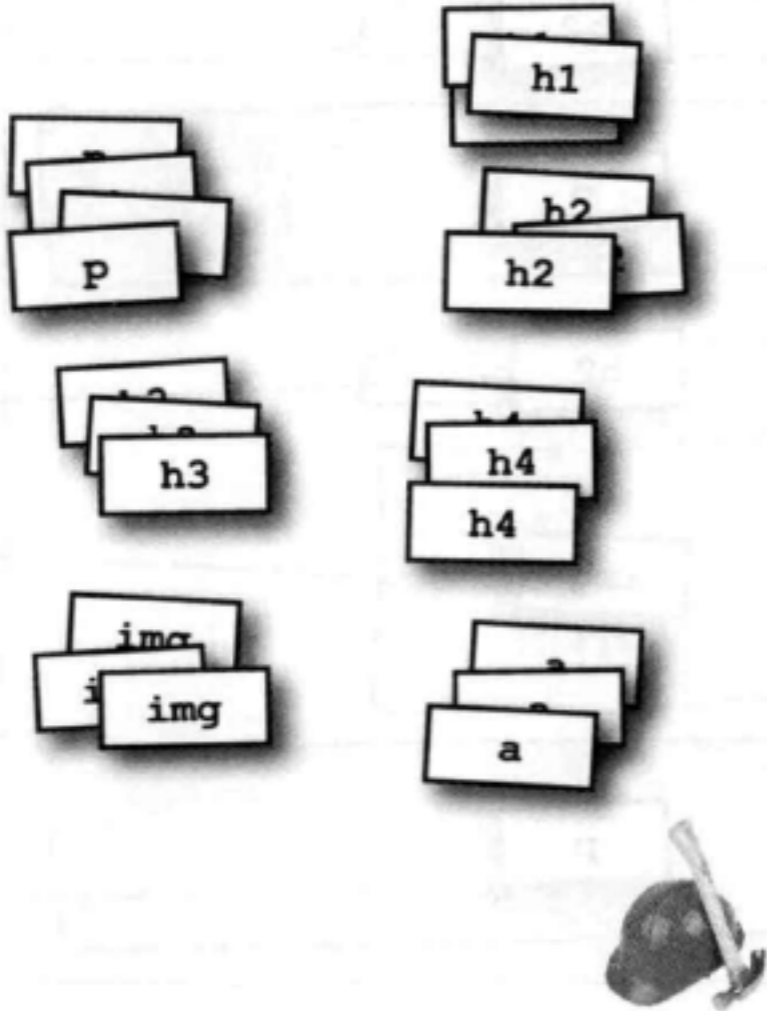
既然有了页面的草图，下面就来分别研究各个部分，画出更精细的HTML页面略图（或蓝图）……

取草图的各个部分，在蓝图中创建相应的内容区。

现在要做的就是确定哪些HTML元素对应各个内容区，然后才能开始写HTML。

练习: Web构建

你已经确定了页面的主要建设区域，现在只需要落实建筑材料。使用下面的元素来标注各个区域。这些元素不一定都会用到，所以如果剩下几个建筑材料也不用担心。另外，别忘了带上你的安全帽。

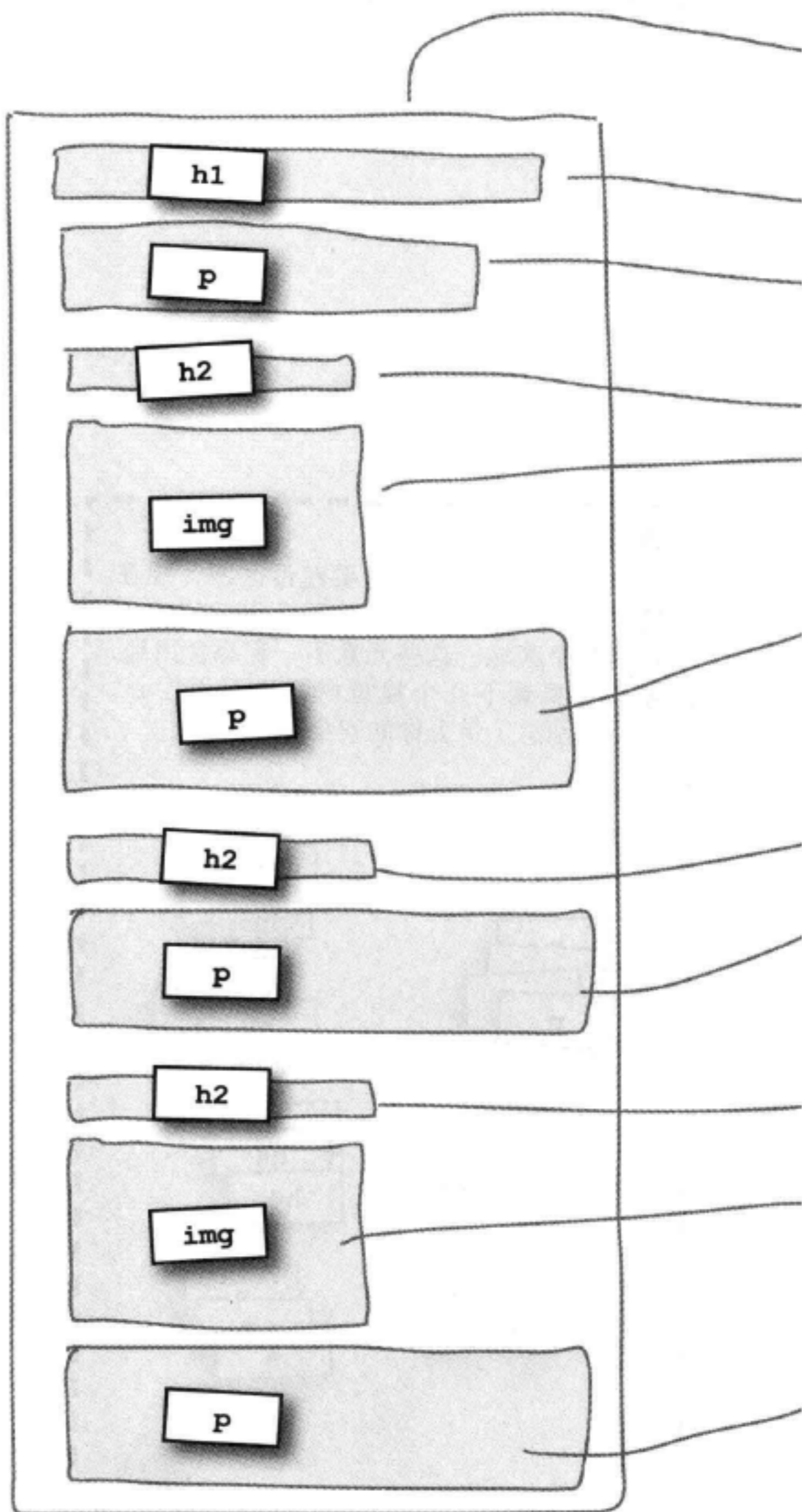


从略图到网页

胜利在望了。你已经为Tony的网页创建了一个略图。现在只需要创建相应的HTML来表示这个页面，并填入Tony的文字。

开始之前，要记住每个Web页面都需要从<html>元素开始，而且要包含<head>和<body>元素。

既然你已经知道哪些“构建模块”构成了页面的各个部分，现在就可以把这个蓝图直接翻译成HTML了。



不要忘了, `<html>`, `<head>`, `<title>` 和 `<body>` 元素是必不可少的。

我们使用旅行日志的大标题作为这个Web页面的标题。

```

<html>
  <head>
    <title>My Trip Around the USA on a Segway</title>
  </head>
  <body>

    <h1>Segway'n USA</h1>
    <p>
      Documenting my trip around the US on my very own Segway!
    </p>

    <h2>August 20, 2012</h2>
    
    <p>
      Well I made it 1200 miles already, and I passed
      through some interesting places on the way: Walla Walla,
      WA, Magic City, ID, Bountiful, UT, Last Chance, CO,
      Why, AZ and Truth or Consequences, NM.
    </p>

    <h2>July 14, 2012</h2>
    <p>
      I saw some Burma Shave style signs on the side of the
      road today: "Passing cars, When you can't see, May get
      you, A glimpse, Of eternity." I definitely won't be passing
      any cars.
    </p>

    <h2>June 2, 2012</h2>
    
    <p>
      My first day of the trip! I can't believe I finally got
      everything packed and ready to go. Because I'm on a Segway,
      I wasn't able to bring a whole lot with me: cell phone, iPod,
      digital camera, and a protein bar. Just the essentials. As
      Lao Tzu would have said, "A journey of a thousand miles begins
      with one Segway."
    </p>

  </body>
</html>

```

这里是Tony旅行日志的标题和描述。

标题
图像
描述

这里是Tony的最新日志。

这是第2个日志, 这里没有图像。

最下面是Tony的第一个日志, 包含图像 "segway1.jpg"。

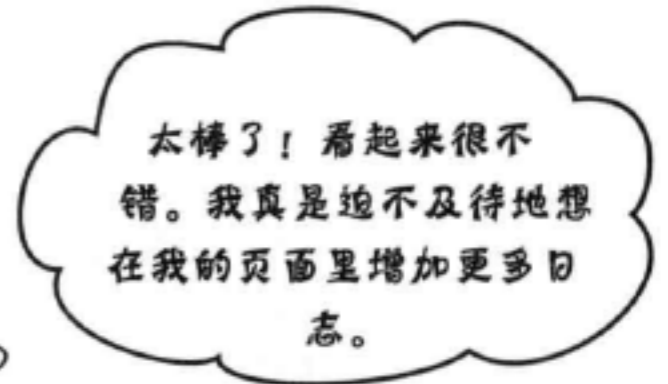
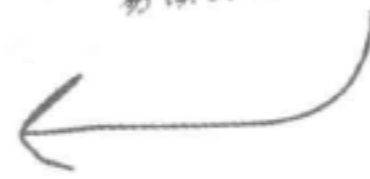
最后, 但绝不是最不重要, 不要忘记结束 `<body>` 和 `<html>` 元素。

输入上面的HTML。把你的文件保存到 "chapter3/journal" 文件夹, 文件名为 "journal.html"。你会发现 "segway1.jpg" 和 "segway2.jpg" 已经在 "images" 文件夹中。完成后, 对这个页面做个测试。

测试Tony的Web页面



看看这个页面怎么样。Tony的所有日志都已经放在一个结构清晰、简单易读的网页中。

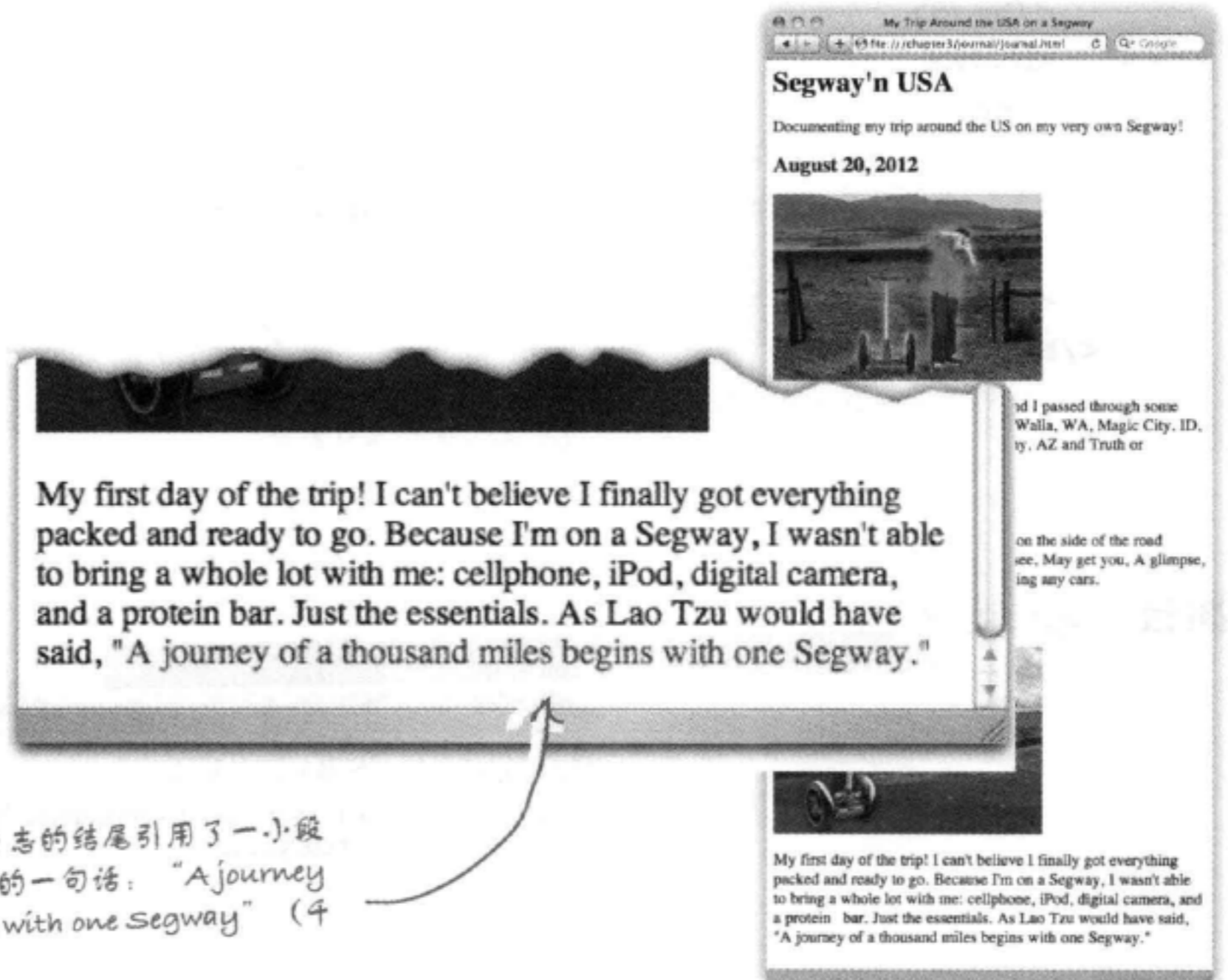


Tony在路上打来电话……

增加一些新元素

你已经掌握了HTML的基本元素，而且只经过简单的几步，你已经利用这些基本HTML元素（<p>、<h1>、<h2>和）把手写的旅行日志变成了网上版本。

现在我们要让你的思维再拓宽一点，增加一些更常见的元素。下面再来看Tony的旅行日志，看看哪里还可以加以改进……



注意这里，Tony在第一篇日志的结尾引用了一小段文字。他在这里改写了老子的一句话：“A journey of a thousand miles begins with one segway”（千里之行，始于segway）。

HTML有一个元素<q>可以支持这种引用。翻到下一页，我们来分析这个元素……

认识 <q> 元素

想在HTML里加段简短的引用？<q>元素正是你需要的。下面给出一小段测试HTML，来说明它是如何工作的：

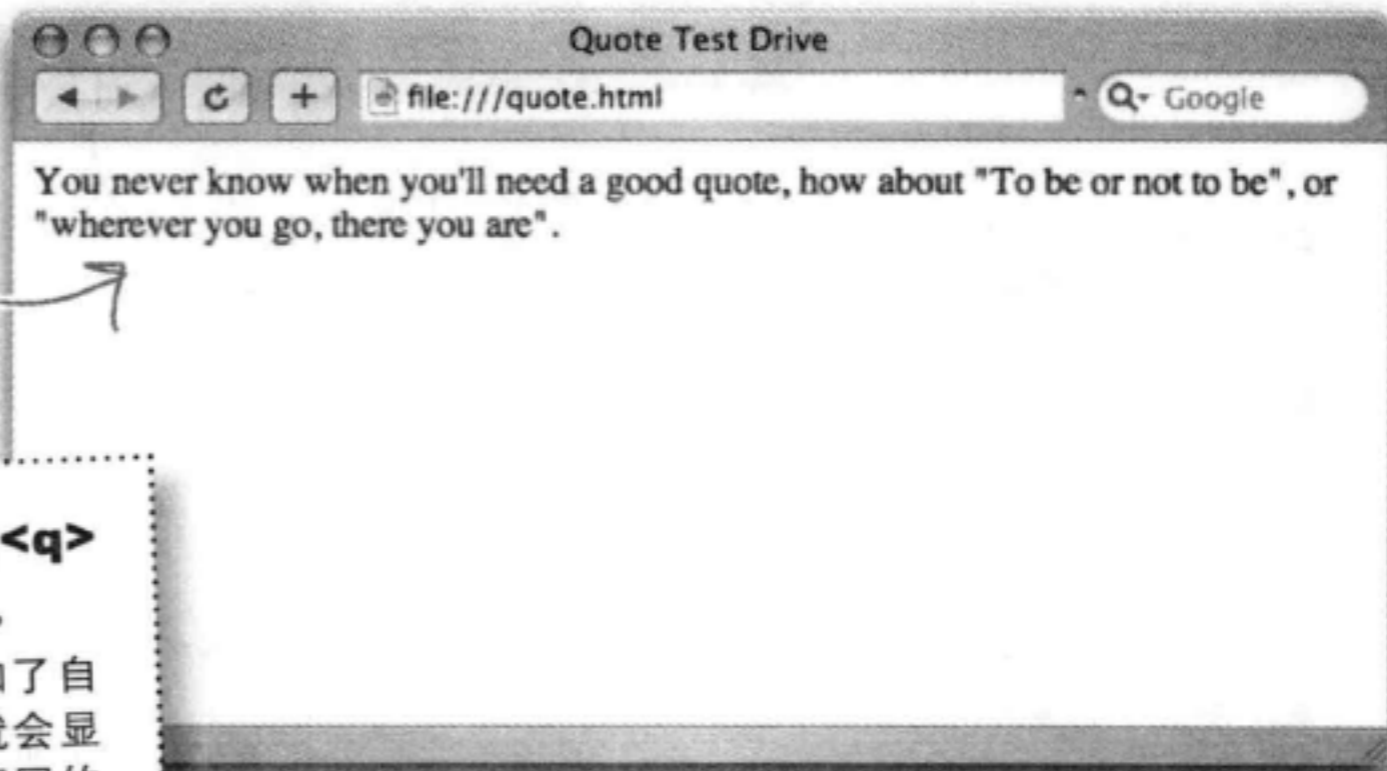
```
<html>
  <head>
    <title>Quote Test Drive</title>
  </head>
  <body>
    <p>
      You never know when you'll need a good quote, how
      about <q>To be or not to be</q>, or <q>Wherever you go, there you are</q>.
    </p>
  </body>
</html>
```

这个HTML中有两个引用……

每个引用都要用一个<q>开始标记和一个</q>结束标记包围。注意，这里没有在引用周围加双引号。

测试

这些引用在浏览器中会这样显示。注意浏览器会负责加上双引号。



Watch it!

并不是所有浏览器都会在<q>元素的内容两边加双引号。

这很糟糕，因为如果你增加了自己的双引号，有些浏览器就会显示两组双引号。建议你在不同的

浏览器中对<q>做个测试，看看会得到什么结果。



等一下……你去掉了双引号，换成了一个<q>元素，只是为了显示双引号？我没看错吧？这样不是更复杂了吗？

不。这可以让页面更结构化，更有意义。

人们之所以在文本中使用双引号，原因可能有很多，不过如果使用<q>，这就意味着这是某种特定的内容，它表示一段引用文字（在Tony的例子中，则是一段“改写的”引用）。

注意！使用双引号并不代表它确实是引用文字。

换句话说，通过标记引用，我们可以为它增加更多含义。在增加<q>元素之前，浏览器只知道这是一个文本段落，其中包含一些双引号字符。而现在，由于我们使用了<q>元素，浏览器知道了这些文字是真正的引用。

那又如何呢？嗯，既然浏览器知道这是一个引用，它就能用最合适的方式来显示。有些浏览器会在文本两边显示双引号，有些却不会，另外如果浏览器使用非英语语言，可能还会使用其他方法。不要忘记移动设备，比如手机，或者为有视力缺陷人士提供的语音HTML浏览器和大屏幕阅读器。在其他情况下这也很有用，如搜索引擎可能要搜寻包含引用的网页。页面中增加结构和含义绝对是有意义的。

最重要的一个原因是（本书后面谈到表现和CSS时就会看到），你可以对引用设置样式，使它的外观如你所愿。假设你希望引用文字用灰色斜体显示，怎么做呢？如果使用<q>元素在网页中指定了引用内容，就能轻松做到。



Exercise

下面是Tony的旅行日志。修改他引用的老子的话，现在改用<q>元素。在纸上完成后，再修改“journal.html”文件，然后测试这个页面。答案在本章最后给出。

```
<html>
  <head>
    <title>Segway'n USA</title>
  </head>
  <body>

    <h1>Segway'n USA</h1>
    <p>
      Documenting my trip around the US on my very own Segway!
    </p>

    <h2>August 20, 2012</h2>
    
    <p>
      Well I made it 1200 miles already, and I passed
      through some interesting places on the way: Walla Walla,
      WA, Magic City, ID, Bountiful, UT, Last Chance, CO,
      Why, AZ and Truth or Consequences, NM.
    </p>

    <h2>July 14, 2012</h2>
    <p>
      I saw some Burma Shave style signs on the side of the
      road today: "Passing cars, When you can't see, May get
      you, A glimpse, Of eternity." I definitely won't be passing
      any cars.
    </p>

    <h2>June 2, 2012</h2>
    
    <p>
      My first day of the trip! I can't believe I finally got
      everything packed and ready to go. Because I'm on a Segway,
      I wasn't able to bring a whole lot with me: cell phone, iPod,
      digital camera, and a protein bar. Just the essentials. As
      Lao Tzu would have said, "A journey of a thousand miles begins
      with one Segway."
    </p>
  </body>
</html>
```

五分钟 谜案



出生后就被分开的元素

很多年前，一对孪生兄弟在Web镇降生，由于互联网路由器故障导致的一起意外事件，这对双胞胎出生后不久就分开了。他们渐渐长大，都不知道对方的存在，不过后来通过一系列匪夷所思的巧合，他们终于相遇，认出对方，不过他们决定保守这个秘密。

认出之后，他们很快发现彼此有太多共同之处。他们都娶了一个名叫Citation的妻子。他们都酷爱引用。哥哥<q>元素喜欢短引用，而弟弟<blockquote>喜欢比较长的引用，通常能记住书上或诗歌中的大段文字。

作为孪生兄弟，他们长相极其相似，所以他们决定制定一个邪恶的计划，相互交换身份。他们先拿自己的妻子做实验（细节不便详述），很成功，他们的妻子都没有发觉（或者至少假装没有发觉）。

接下来，他们想在工作场所测试这个交换计划，恰好他们都做同样的工作：在HTML文档中标记引用。所以，选好日子后，兄弟俩来到对方的工作场所，信心满满地实施他们的邪恶计划（毕竟，既然他们的妻子都没能区分，更何况老板呢），不过，这就出问题了。工作仅仅10分钟后，兄弟俩都被发现是冒名顶替的，并立即向标准权威机构发出了警报。

这对双胞胎怎么被发现的呢？
继续读下去，你会找到更多线索……

很长的引用

既然知道了怎么使用短引用，下面来看长引用。Tony给了我们一个柏玛刮胡膏广告(long quote)的长引用。

在他的旅行日志里，Tony直接把柏玛刮胡膏引用放在段落里，不过如果把这个引用拿出来放在单独的“块”里不是更好吗？就像这样：

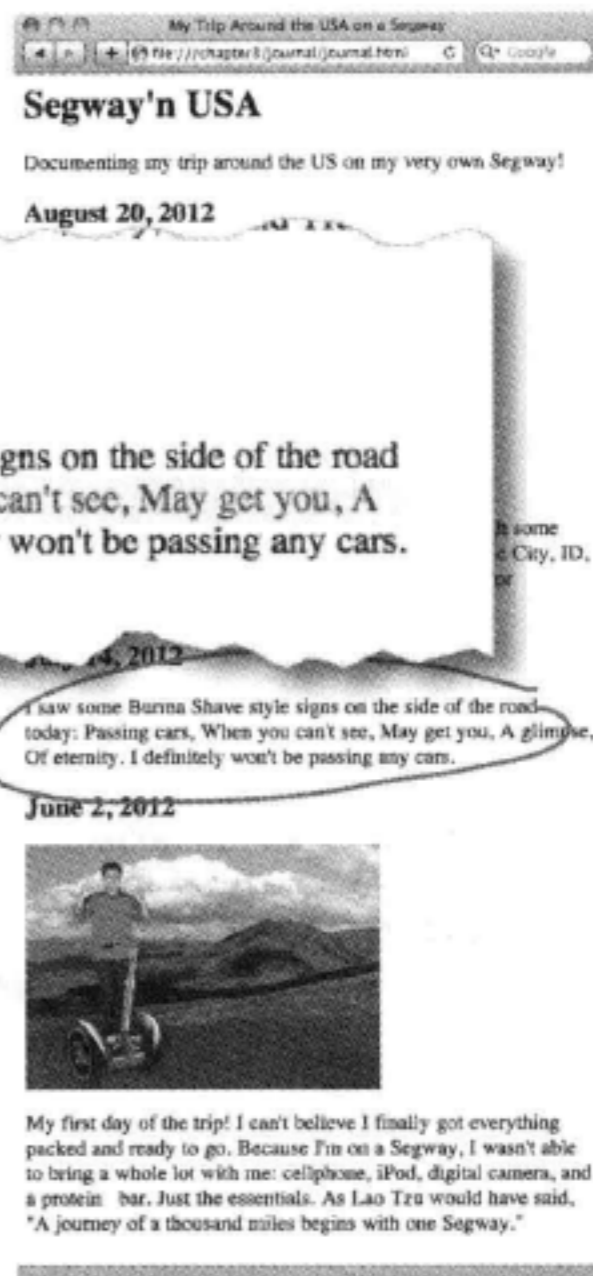
I saw some Burma Shave style signs on the side of the road today:

*Passing cars,
When you can't see,
May get you,
A glimpse,
Of eternity.*

I definitely won't be passing any cars.

如果你不知道“柏玛刮胡膏”广告，后面几页就会告诉你……

这里就用到了<blockquote>元素。与<q>元素不同（<q>用于短引用，作为现有段落的一部分），<blockquote>元素用于较长的引用，需要单独显示。



工欲善其事，必先利其器，这很重要，<blockquote>元素是完成这个任务的最佳工具。



增加 <blockquote>

下面在 Tony 的网上日志中加入一个 <blockquote>。

- 1 打开你的“journal.html”文件，找到7月14日的日志。修改这个段落，如下所示：

```

<h2>July 14, 2012</h2>
<p>
  I saw some Burma Shave style signs on the
  side of the road today:
</p>
<blockquote>
  Passing cars,
  When you can't see,
  May get you,
  A glimpse,
  Of eternity.
</blockquote>
<p>
  I definitely won't be passing any cars.
</p>

```

要插入 <blockquote> 元素，需要先结束这个段落。

接下来把柏瑞刮胡膏广告词放在 <blockquote> 元素中。

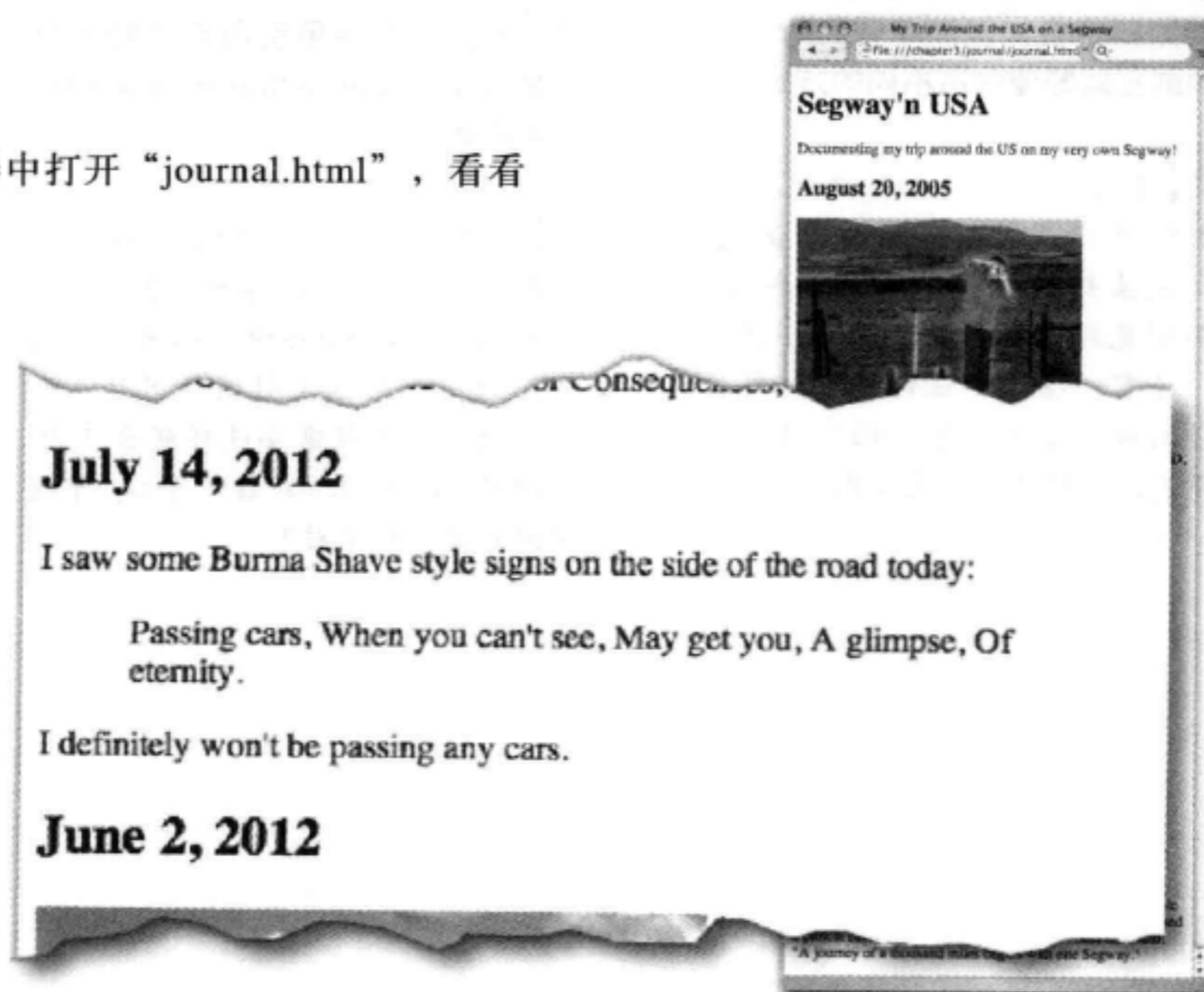
每行文字都另起一行显示，这样更像柏瑞刮胡膏广告。

最后，需要增加一个 <p> 标记，在 <blockquote> 之后开始下一个段落。

- 2 再做一次测试。在你的浏览器中打开“journal.html”，看看你的成果：

<blockquote> 创建了一个文本块（与 <p> 类似），另外还把文字稍稍缩进，使它更像一个引用，这正是我们想要的……

不过我们的引用与原先预想的还有差距，因为所有行都在一起。我们希望它们分行显示。嗯，稍后再来解决这个问题……



there are no Dumb Questions

问：看我这样做对不对：我想在段落里加一些引用就使用<q>，如果希望引用在网页上自成一段就用<blockquote>，这样对吗？

答：说的没错。一般来讲，如果想引用一段或者多段文字，就要使用<blockquote>，不过如果只想把一个引用放在现有的文字里，作为其中的一部分，就可以使用<q>。

问：一个块引用中有多段？怎么做到的呢？

答：很容易。只需要把段落元素放在<blockquote>中，一个段落元素对应一个段落。你可以自己试试看。

问：怎么知道我的引用或块引用在其他浏览器中会如何显示呢？听上去不同的浏览器好像会用不同的方式处理。

答：是的。互联网世界里就是这样。如果没有在不同的浏览器上试试看，你就无法知道你的引用会如何显示。有些浏览器使用双引号，有些使用斜体，还有一些不加任何效果。要想确定它们如何显示，唯一的办法就是指定样式，稍后会讨论怎么做到这一点。

问：我知道了，<blockquote>元素的文本会自成一段，还会缩进，那么为什么<blockquote>不像<q>元素那样嵌在段落中呢？

答：因为<blockquote>实际上就像一个新段落。可以这样来考虑，就好像你在一个字处理器中输入文字。输入一段之后，你按两次Enter键，开始一个新段落。要输入一个块引用，也要做同样的工作，并缩进引用。先记住这个内容，这是一个重点，稍后还会仔细讨论。

另外，要记住只是有些浏览器显示<blockquote>时会缩进。一些浏览器对于<blockquote>可能不使用缩进。所以，不要指望<blockquote>在所有浏览器中都有相同的显示。

问：我能结合使用引用元素吗？例如，能不能在<blockquote>元素中使用<q>元素？

答：当然可以。你可以把一个<q>元素放在<p>元素中，同样的，也可以将<q>嵌在<blockquote>中。如果你引用某个人的话，而他又引用了另外某个人的言论，可能就需要这样嵌套使用。不过把<blockquote>放在一个<q>中没有任何意义，对不对？

问：你说我们可以用CSS对这些元素指定样式，这么说来，如果我希望让<q>元素中的文本用灰色斜体显示，可以用CSS来做到。不过，难道不能用元素把引用置为斜体吗？

答：嗯，也可以，不过这么做不太好，因为你这样使用元素只是为了实现显示效果，而不是因为你写了要强调的文字。如果你引用某人的话，其中确实强调了某个词，或者如果你想强调引用中的某个重点，那么完全可以在引用中使用元素。不过，不要只是为了显示斜体而使用元素。利用CSS，有很多更好、更容易的方法来实现你希望的元素显示效果。

谜底：出生后就被分开的元素

这对孪生引用兄弟为什么会这么快被人发现是冒牌的？

你可能已经猜到了，<q>和<blockquote>一开始工作想要标记文本时就被发现了。<q>通常只是些不起眼的小引用，现在却自成一块，格外突出，而<blockquote>的引用突然湮没在普普通通的文本段落中。接下来采访这个恶作剧的受害者时，一位编辑抱怨道，“就是因为这些疯子，我的整个引用页面变得乱七八糟。”两兄弟分别受到批评，回到各自原来的岗位。<blockquote> 和<q>向他们的妻子坦白后，她们随即驾车离开了小镇。故事就这样结束了（尽管结局不完美）。



五分钟谜案 之谜底

<q> 和 <blockquote> 谜案的真相

OK，揭开谜底的时候到了：<blockquote>和<q>实际上是两类不同的元素。<blockquote>元素是一个块（block）元素，而<q>元素是一个内联（inline）元素。有什么区别吗？块元素显示时就好像前后各有一个换行，而内联元素在页面文本流中总在“行内”出现。

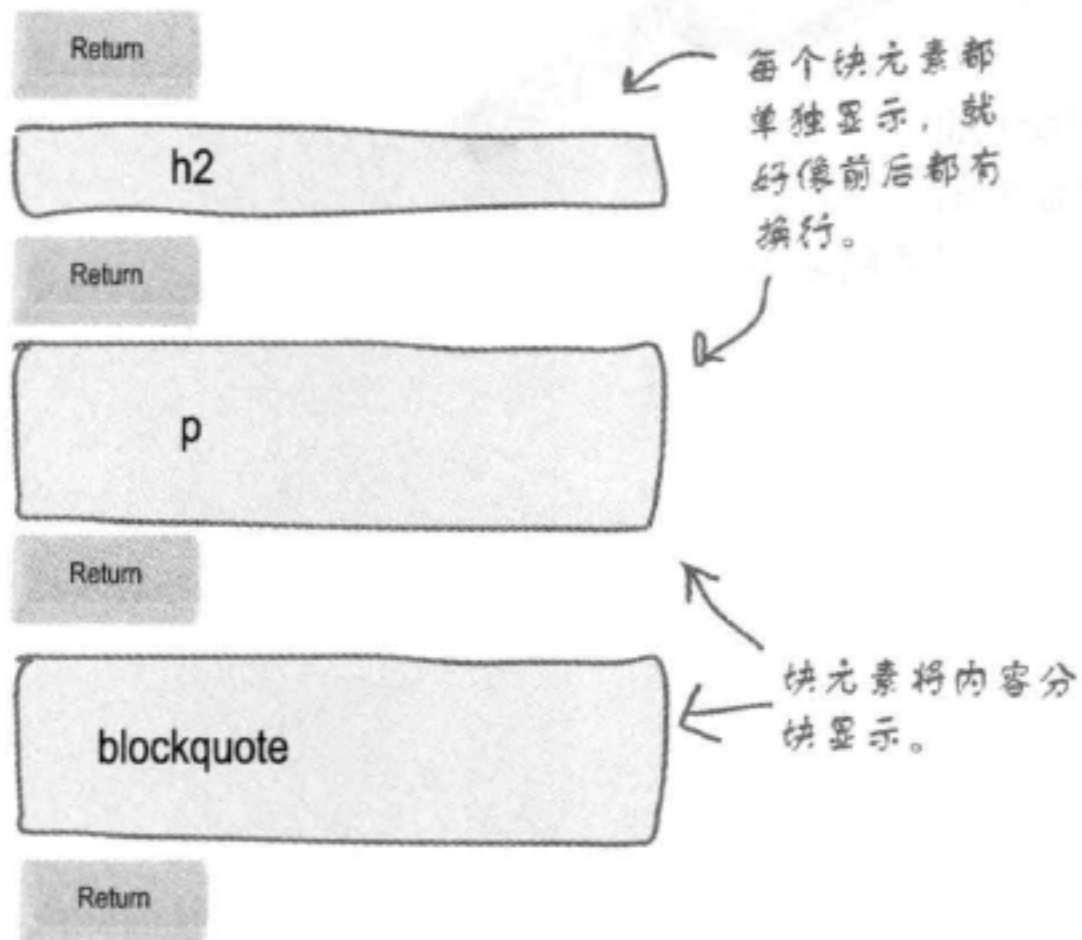


块元素：特立独行



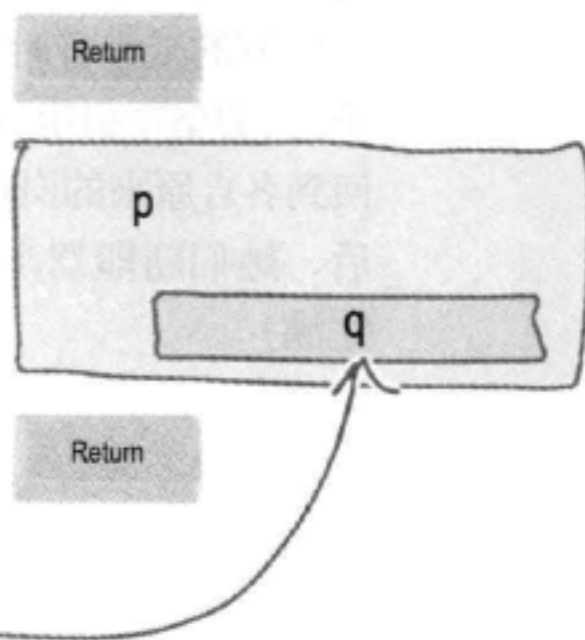
内联元素：随波逐流

<h1>、<h2>、...、<h6>、<p>和<blockquote>都是块元素。



<q>、<a>和是内联元素。

<q>则不同，与所有元素一样，<q>会显示在所在的段落中。



要记住：块元素特立独行；内联元素随波逐流。

there are no
Dumb Questions

问:我想我知道换行是什么，就像在打字机或计算机键盘上按Enter键，是吗？

答:差不多吧。“换行”字面意思就是“换一行”，你说的没错，就像按Enter键一样。你已经知道，浏览器显示一个页面时，HTML文件中的换行符并不会出现，对不对？不过，现在你已经看到了，只要使用块元素，浏览器就会使用换行符来分隔每一“块”。



嗯，听起来不错，不过说这些换行、块和内联元素有什么用呢？该回来谈谈Web页面了吧？



你要了解HTML是如何工作的，不要低估掌握这些知识的重要性。很快你就快会看到，在页面上如何结合元素对于元素将显示为块元素还是内联元素有很大影响。接下来就会讨论这个内容。

另外，还可以这样考虑块元素和内联元素：块元素通常用作Web页面中的主要构建模块，而内联元素往往用来标记小段内容。设计一个页面时，一般先从较大的块开始（块元素），然后在完善页面时再加入内联元素。

在用CSS控制HTML的表现时，这些知识就能派上用场了。如果你清楚内联元素和块元素的区别，就可以轻松设计好布局，在别人为布局设计忙得焦头烂额时，你却能悠闲地喝着马提尼。



我一直在考虑柏玛刮胡膏广告词。它们没有换行，但我一点都不奇怪，因为一开始我们就说过，浏览器不会显示空白符和换行……

……不过我想能修复这个问题只有一个办法，把每一行放在一个块元素里（比如一个段落）。否则，浏览器怎么能显示换行呢？



如果有这样一个元素，它的唯一任务就是在你需要时提供一个换行，你觉得怎么样？

这样不是很好吗？你能让浏览器注意到并插入一些回车符。

事实上确实有这样一个元素，这就是
元素，它的作用正是如此。可以这样来使用：

```
<h2>July 14, 2012</h2>
<p>
  I saw some Burma Shave style signs on the
  side of the road today:
</p>
<blockquote>
  Passing cars, <br>
  When you can't see, <br>
  May get you, <br>
  A glimpse, <br>
  Of eternity. <br>
</blockquote>
<p>
  I definitely won't be passing any cars.
</p>
```

这是从Tony页面中节取的7月14日日志的片段。

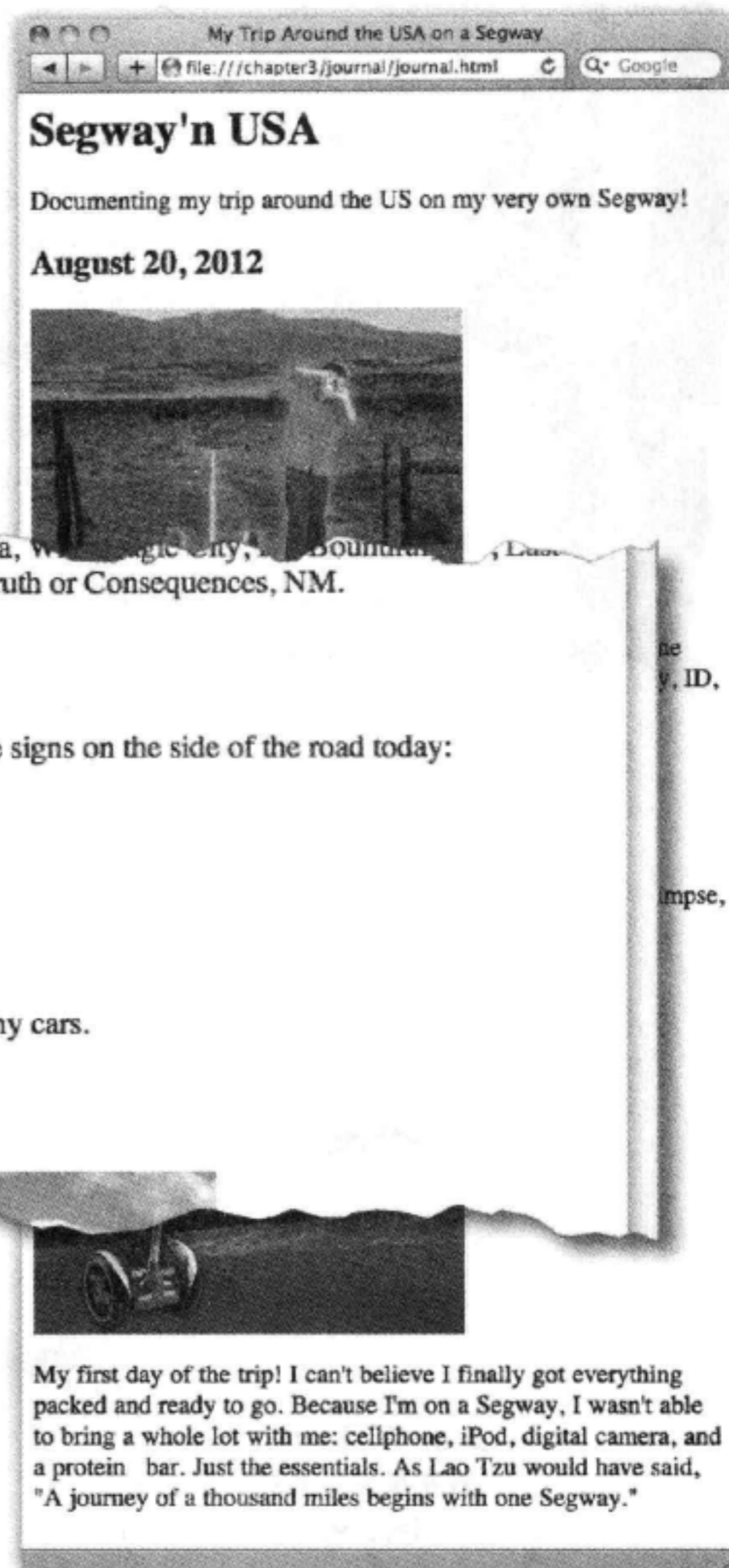
在希望换行的地方增加一个
元素，这里就会插入一个“换行符”。



为Tony的日志增加
元素。修改之后，保存文件，再对页面做个测试。

我们修改的结果如下。现在看起来像是真正的柏瑞刮胡膏广告了。

现在每一行后面都有一个换行。





在第1章中我们说过，元素是开始标记+内容+结束标记。`
`算是一个元素吗？它没有任何内容，甚至还没有结束标记。

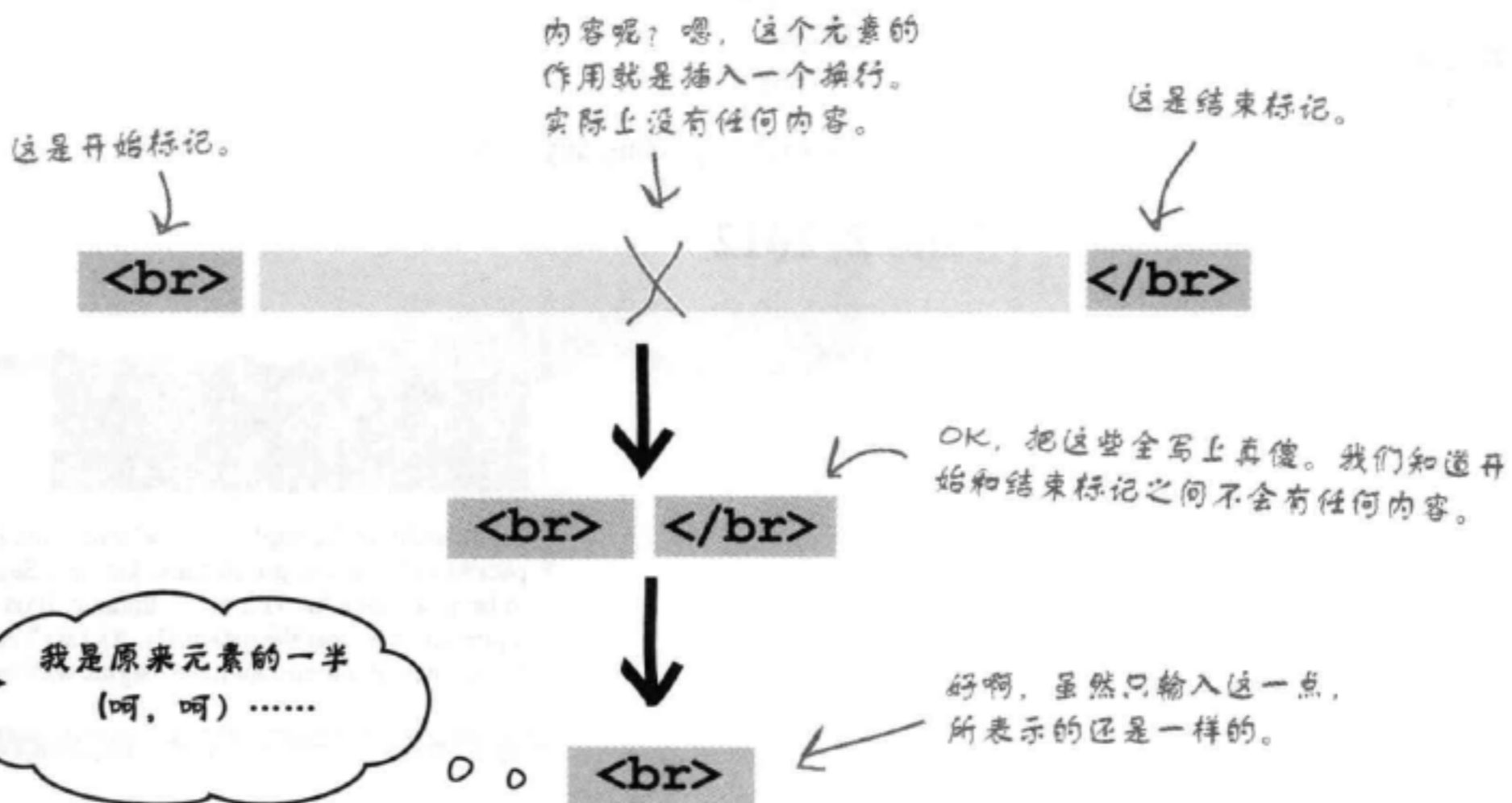
的确，它没有内容。

`
`元素是一个没有任何内容的元素。为什么？因为它只是一个换行，没有其他内容。所以，如果一个元素设计为没有任何实际内容，我们会使用简写来表示这个元素，最后就会像`
`一样。毕竟，如果没有这个简写，每次需要一个换行时都需要写为`
</br>`，这有意义吗？

`
`并不是唯一没有实际内容的元素，还有很多类似这样的元素，我们把这些元素叫做void元素。实际上，之前我们已经见过另一个void元素，那就是``元素。后面几章还会详细讨论``元素。

要记住，之所以用简写不是因为懒惰，更多的是为了提高效率。这样表示void元素会更高效（输入的字符更少，相应的页面中字符数也更少）。实际上，读过HTML之后，你会发现这样看起来也更轻松。

原先它们称为“空元素”，显然不太贴切，所以后来又改名为void元素。就个人而言，我们还是喜欢空元素这个名字。



问:那么, `
`的唯一作用就是插入一个换行, 是吗?

答:没错, 通常只有当你开始一个新的块元素(如 `<p>`、`<h1>`等)时, 浏览器才会插入换行。如果你希望在文本中插入一个换行, 就要使用`
`元素。

问:为什么把`
`叫做“void”元素?

答:因为它没有任何内容, 按理说元素 = 开始标记+ 内容 + 结束标记。因为没有内容也没有结束标记, 所以它是空元素(void)。就像一个“空位”, 这里什么也没有, 是空的。

问:我还是不明白, 请解释一下为什么`
`元素是“void”?

答:考虑另外一个元素, 比如`<h1>`(或者`<p>`或`<a>`)。这个元素的作用是标记内容, 如:

```
<h1>Don't wait, order now</h1>
```

而`
`元素不同, 它的作用就是在HTML中插入一个换行。没有需要标记的内容。我们不需要额外的尖括号和标记, 所以可以把它简写为一种更方便的形式。如果你觉得“void”这个名字很怪, 没错: 这个词来自计算机科学, 原意是“无值”。

问:还有没有其他void元素? 我觉得``肯定也是一个void元素, 对吗?

答:没错, 还有另外一些void元素。你已经见过我们使用了``元素, 很快就会详细介绍这个元素。

问:我能把某个元素变成void元素吗? 例如, 如果有一个链接, 那么我不想指定任何内容, 可不可以

直接写为 `` ?

答:不行。世界上有两类元素: 正常元素和void元素, 正常元素比如`<p>`、`<h1>`和`<a>`, void元素如`
`和``。不能在这两个阵营之间来回切换。例如, 如果你只输入``, 这只是个开始标记, 没有内容或结束标记(这是不对的)。如果写为``, 这就是一个空元素, 这样是可以的, 不过放在页面上没有什么用处!

问:我见过有些页面上没有使用`
`而用了`
`。这是什么意思?

答:意思完全是一样的。处理XHTML时, `
`的语法更为严格。只要看到`
`, 都可以把它想象成是`
`, 不过除非要规划或编写与XHTML兼容的页面(关于XHTML的更多信息参见附录), 否则HTML中都应该使用`
`。

**设计为没有任何内容的元素称为void元素。需要使用void元素时, 如`
`或``, 只需使用一个开始标记。这是一种方便的简写形式, 可以减少HTML中的标记数量。**

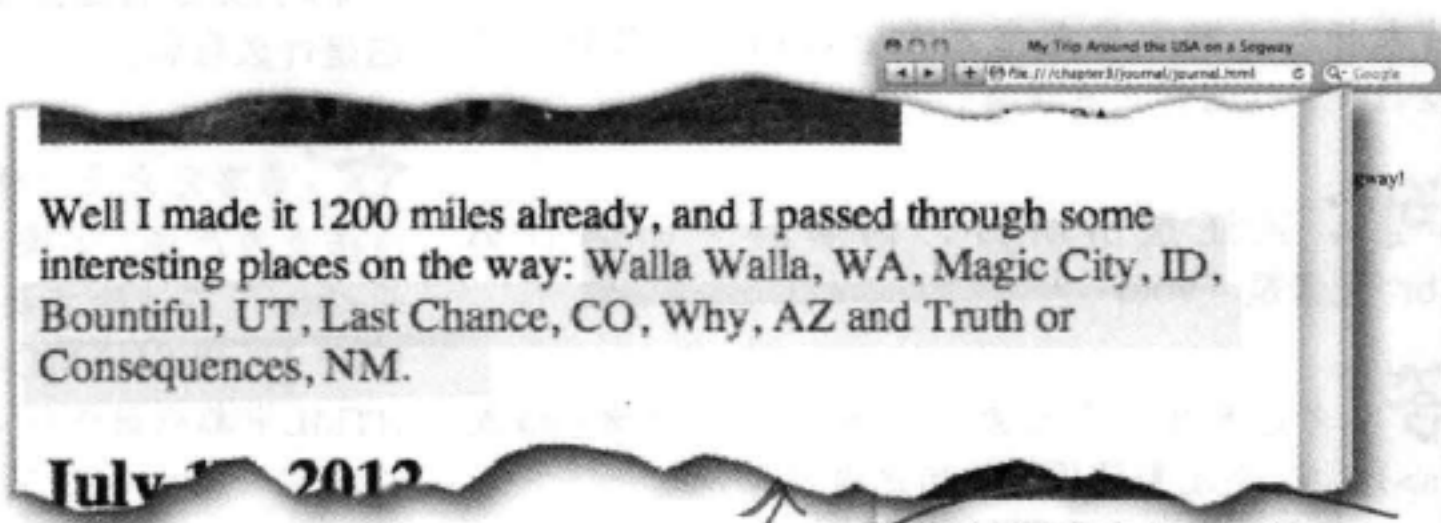
再回到Tony的网站……

这一章已经做了不少工作了：已经设计并创建了Tony的网站，你还认识了一些新元素，另外了解了关于元素的一些内部信息，大多数创建Web页面的人对此可能都不清楚（比如块元素和内联元素，这些知识在后面的章节中很有用处）。

不过你的工作还没有结束。我们要让Tony的网站更上一层楼，看看还有哪些地方可以增加一些标记。

比如说？列表？尝试一下：

这里有一个列表。Tony在8月的旅行日志里写了他曾到过的城市的一个列表。



Well I made it 1200 miles already, and I passed through some interesting places on the way: Walla Walla, WA, Magic City, ID, Bountiful, UT, Last Chance, CO, Why, AZ and Truth or Consequences, NM.

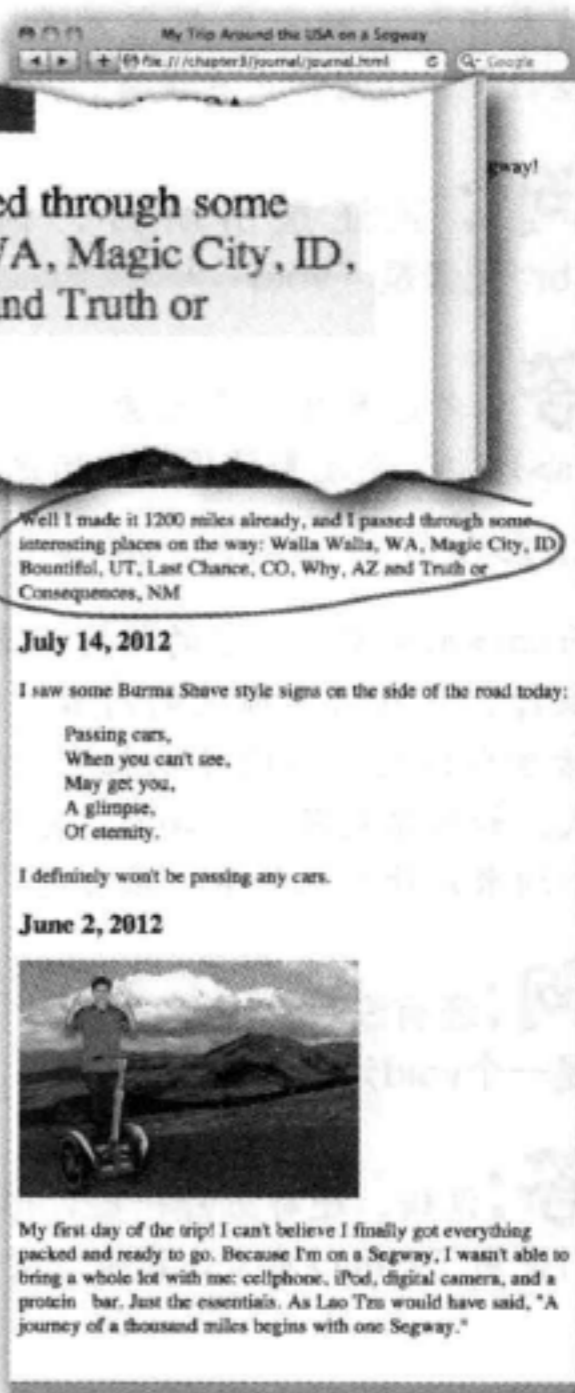
July 14, 2012

如果能标记这个文本，让浏览器知道这个文本是一个列表就好了！这样浏览器就能用一种更有用的方式显示列表项。比如：

Well I've made it 1200 miles already, and I passed through some interesting places on the way:

1. Walla Walla, WA
2. Magic City, ID
3. Bountiful, UT
4. Last Chance, CO
5. Why, AZ
6. Truth or Consequences, NM

注意这不仅是一个列表，还是一个有序列表。Tony是按特定的顺序游历这些城市的。



当然，也可以使用<p>元素创建列表……

使用<p>元素创建列表并不难。最后可能像这样：

```
<p>
1. Red Segway
</p>
<p>
2. Blue Segway
</p>
```

← Segway最优选的两种颜色。

不过，出于多方面原因，我们不这么做

你现在应该有点常识了。我们总希望选择与内容结构含义最接近的HTML元素。如果这是一个列表，就使用一个列表元素。这样做可以让浏览器和你（稍后会看到）有更大的能力和灵活性，可以用一种有用的方式显示内容。

要记住，工欲善其事，必先利其器，这很重要，<p>元素可不是完成这个任务的理想工具。



BRAIN POWER

为什么不用<p>建列表？（多选题）

- A. HTML专门提供了一个列表元素。如果使用这个元素，浏览器会知道这个文本是一个列表，就能用最佳方式来显示。
- B. 段落元素表示的是文本段落，而不是列表。
- C. 可能看上去不是一个列表，而是一堆有编号的段落。
- D. 如果想改变列表顺序，或者想插入一个新列表项，就必须对所有列表项重新编号。这太麻烦了。

答案：A、B、C、D

两步轻松构建HTML列表

创建一个HTML列表需要两个元素，结合使用这两个元素就构成了列表。第一个元素用来标记每个列表项。第二个元素确定你创建的是哪种类型的列表：有序列表还是无序列表。

下面分两步在HTML中创建Tony的城市列表。

第一步：

将每个列表项放在一个 `` 元素中。

要创建一个列表，需要把每个列表项放在单独的 `` 元素中，这说明需要把内容用一个开始 `` 标记和一个结束 `` 标记包围起来。与所有其他HTML元素一样，标记之间的内容由你来定，可以很短，也可以很长，分为多行。

这里我们给出了Tony旅行日志的一个HTML片段。

在“journal.html”文件中找到这个HTML片段，跟着我们做相应的修改。

```
<h2>August 20, 2012</h2>
  
```

```
<p>
```

```
Well I've made it 1200 miles already, and I passed
through some interesting places on the way:
```

```
</p>
```

首先，将列表项移出段落。
列表会独立显示。

```
<li>Walla Walla, WA</li>
<li>Magic City, ID</li>
<li>Bountiful, UT</li>
<li>Last Chance, CO</li>
<li>Why, AZ</li>
<li>Truth or Consequences, NM</li>
```

……然后将每个列表项用一对 ``, `` 标记包围。

每个 `` 元素
会成为列表中
的一个列表项。

```
<h2>July 14, 2012</h2>
```

```
<p>
```

```
I saw some Burma Shave style signs on the side of
the road today:
```

```
</p>
```

第二步:

用或元素包围列表项。

如果使用一个元素包围列表项，则这些列表项将作为一个有序列表显示；如果使用，则列表将显示为一个无序列表。下面将列表项包围在一个元素中。

同样地，这里再给出Tony旅行日志的一个HTML片段。

```
<h2>August 20, 2012</h2>
```

```

```

```
<p>
```

```
Well I've made it 1200 miles already, and I passed through some interesting places on the way:
```

```
</p>
```

```
<ol>
```

```
<li>Walla Walla, WA</li>
```

```
<li>Magic City, ID</li>
```

```
<li>Bountiful, UT</li>
```

```
<li>Last Chance, CO</li>
```

```
<li>Why, AZ</li>
```

```
<li>Truth or Consequences, NM</li>
```

```
</ol>
```

```
<h2>July 14, 2012</h2>
```

```
<p>
```

```
I saw some Burma Shave style signs on the side of the road today:
```

```
</p>
```

我们希望这是一个有序列表，因为Tony是按一个特定顺序游历这些城市的。所以这里使用一个开始标记。

所有列表项放在元素中间，成为它的内容。

这里结束这个元素。



是一个块元素还是内联元素？呢？

记下来

HTML
is for
structure

Use
ul or ol
for lists

Wash
the
cat

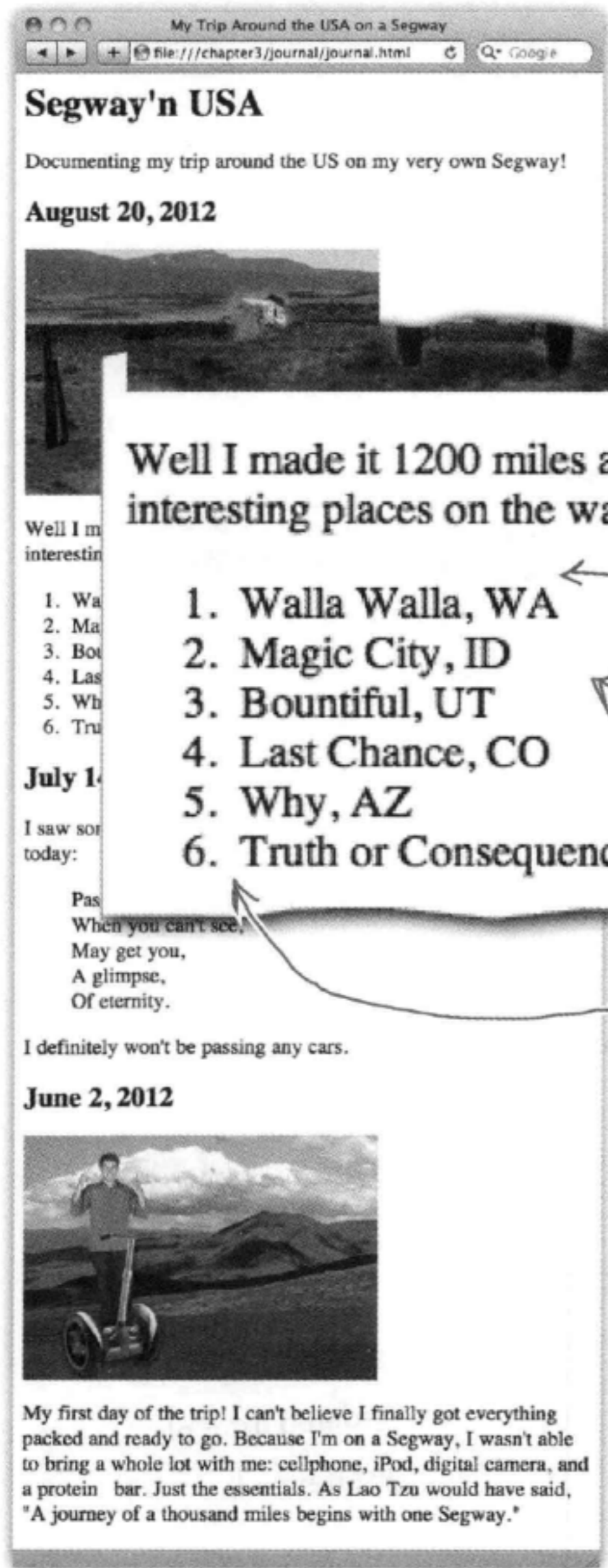
unordered list = ul

ordered list = ol

list item = li

城市列表测试

确保增加了列表的所有HTML，重新加载你的“journal.html”文件，应该能看到这样的结果：



这里是新的改进的城市列表。

列表开始之前有一个换行，所以肯定是一个块元素。

不过，每个列表项之后也有一个换行，所以也肯定是一个块元素！

注意浏览器会负责自动对每个列表项编号（所以不用你操心）。

Sharpen your pencil

事实上，Tony到了New Mexico之后又去了Arizona，你能调整这个列表给出正确的序号吗？



Exercise

这是Tony旅行日志中的另一个列表：cell phone, iPod, digital camera和一个protein bar。在他6月2日的日志中可以找到。这是一个无序列表。

下面给出了这个日志的HTML。请增加必要的HTML，把这些内容变成一个HTML无序列表（要记住，无序列表要用）。我们已经为你调整了一些文本的格式。

完成后，对照本章最后给出的答案检查你的HTML。再在“journal.html”文件中完成修改，并做个测试。

```
<h2>June 2, 2012</h2>
```

```

```

```
<p>
```

```
My first day of the trip! I can't believe I finally got
everything packed and ready to go. Because I'm on a Segway,
I wasn't able to bring a whole lot with me:
```

```
    cell phone
    iPod
    digital camera
    and a protein bar
```

```
Just the essentials. As
Lao Tzu would have said, <q>A journey of a
thousand miles begins with one Segway.</q>
```

```
</p>
```

问: 和总要一起使用吗?

答: 对, 和 (或者和) 总是要一起使用。这些元素离开彼此就没有任何意义了。要记住, 列表实际上就是一组列表项: 元素用来标识每个元素, 元素把它们归为一组。

问: 能把文本或其他元素放在或元素里吗?

答: 不行, 和元素设计为只能包含元素。

问: 无序列表呢? 能不能用不同的项目符号?

答: 可以。不过最好别这么干。后面介绍CSS和表现时还会再来讨论这个问题。

问: 如果我想在列表中嵌一个列表呢? 能行吗?

答: 可以, 当然可以。可以把一个或作为某个的内容, 这样就在列表中有了另一个列表 (我们把这叫做嵌套列表)。

```

<ol>
  <li>Charge Segway</li>
  <li>Pack for trip
    <ul>
      <li>cell phone</li>
      <li>iPod</li>
      <li>digital camera</li>
      <li>a protein bar</li>
    </ul>
  </li>
  <li>Call mom</li>
</ol>

```

嵌套列表

这是一个, 它包围了嵌套列表。

问: 我觉得我基本上已经了解浏览器如何显示块元素和内联元素, 不过还是不清楚哪些元素放在其他元素里, 就像你说的, 哪些元素可以“嵌套”在另外哪些元素里。

答: 这是掌握HTML最大的难点之一。你需要通过好几章来学习这个内容, 我们会向你介绍很多方法, 确保你能把这个关系真正搞懂。不过我们先回到正题, 稍后再讨论嵌套。实际上, 既然你感兴趣, 接下来我们就会介绍这个内容。

问: 这么说, HTML有有序列表和无序列表。还有其他类型的列表吗?

答: 实际上, 确实还有另外一类列表: 定义列表。定义列表如下所示:

```

<dl>
  <dt>Burma Shave Signs</dt>
  <dd>Road signs common in the U.S. in the 1920s and 1930s advertising shaving products.</dd>
  <dt>Route 66</dt>
  <dd>Most famous road in the U.S. highway system.</dd>
</dl>

```

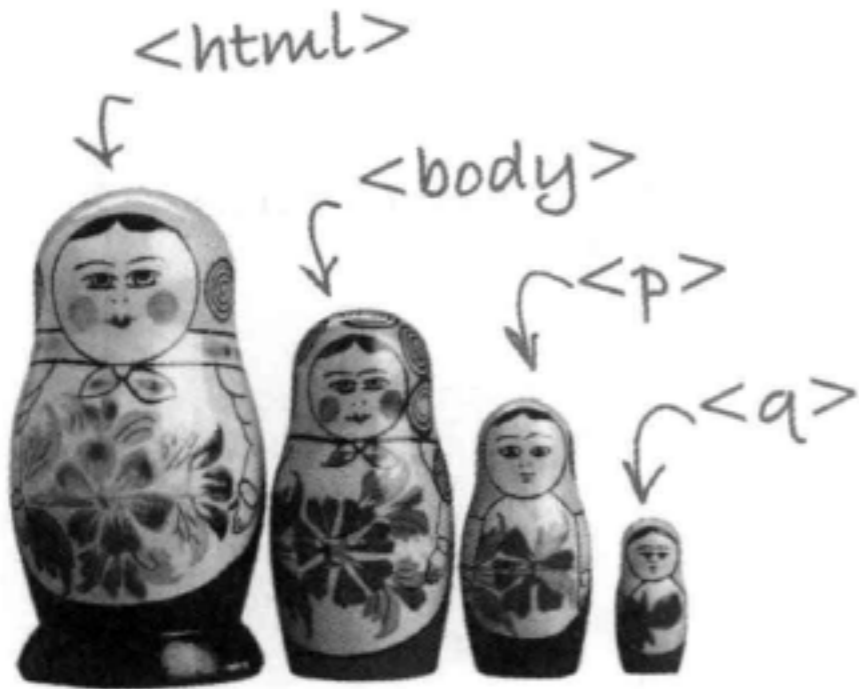
列表中每一项都有一个定义术语<dt>和一个定义描述<dd>。

可以输入这个列表, 试一试。

问: 柏玛刮胡膏是什么?

答: 柏玛刮胡膏 (Burma Shave) 是20世纪初一家生产刮胡膏的公司。他们从1925年开始在路边树立广告牌来推广他们的产品, 事实证明这些广告相当深入人心 (当然也可能让司机分心)。

这些广告4、5或6个一组, 每行一句广告语。曾经一段时间, 美国道路两旁有7000个这样的广告牌。现在大多数都已经没有了, 不过偶尔还能看到剩余的少许几个。



把一个元素放在另一个元素中称为“嵌套”。

把一个元素放在另一个元素中时，我们称之为“嵌套”。我们可能这么说，“<p>元素嵌套在<body>元素中。”目前，你已经见过很多元素可以嵌套在其他元素中。之前我们将<body>元素嵌在<html>元素中，把<p>元素放在<body>元素中，把<q>元素嵌在一个<p>元素中，这样的嵌套还有很多。另外，还把<head>元素放在<html>元素中，<title>元素放在<head>中。HTML页面就是以这种方式构造的。

你对HTML了解得越多，很好地掌握嵌套就越显重要。不过不用担心，很快你就能很自然地以这种方式考虑元素了。



要理解嵌套关系，画一个图

画出网页中元素的嵌套关系就像画一棵家族树。最上面是曾祖父，下面是所有孩子和孙子。这里给出一个例子……

简单web页面。

```
<html>
  <head>
    <title>Musings</title>
  </head>
  <body>
    <p>
      To quote Buckaroo,
      <q>The only reason
      for time is so
      that everything
      doesn't happen
      at once.</q>
    </p>
  </body>
</html>
```

现在把它画成图，每个元素画成一个方框，用线连接元素和嵌套在其中的另一个元素。

<html>总是根元素。

<html>有两个嵌套元素：<head>和<body>。可以把它们都称为<html>的“子元素”。

<body>嵌套在<html>元素中，所以我们说<body>是<html>的子元素。

<title>嵌套在<head>元素中。

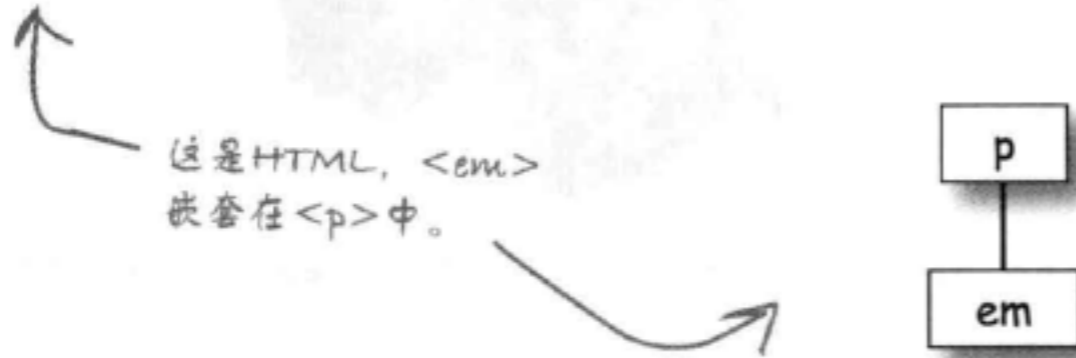
<q>的父元素是<p>，
<p>的父元素是<body>，
<body>的父元素是<html>。

使用嵌套确保标记匹配

如果你了解元素如何嵌套，则第一个好处是你能避免标记不匹配。
(后面还会有更多好处，请耐心等待)。

“标记不匹配”是什么意思？怎么会发生这种情况？下面来看一个例子：

```
<p>I'm so going to tweet <em>this</em></p>
```

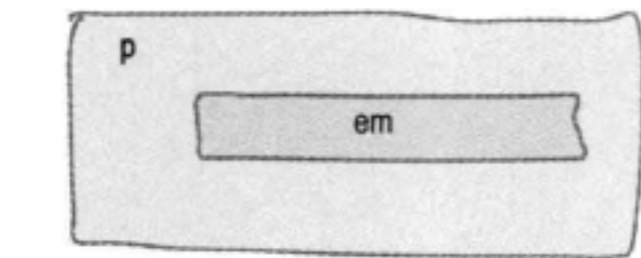


到目前为止，一切都还好，不过很容易出现手误，写出这样的HTML：

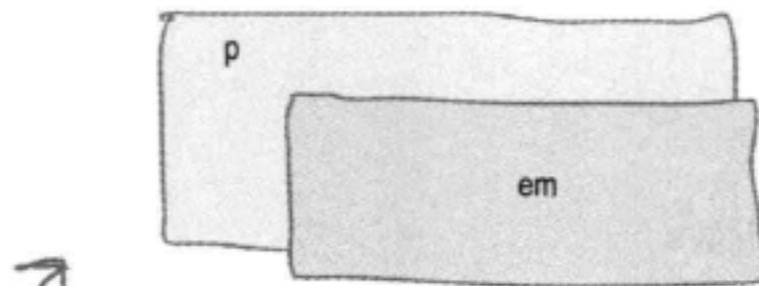
```
<p>I'm so going to tweet <em>this</p></em>
```

错误：`<p>`在``标记之前结束！``元素应当在`<p>`元素里面。

根据你对嵌套的了解，现在你应该知道``元素需要完全嵌套或包含在`<p>`元素中。



正确：这里``元素嵌套在`<p>`里面。



错误：这里``元素有一部分漏到`<p>`元素之外，这说明它没有正确地嵌套在`<p>`元素中。

那又怎样？

如果你喜欢玩俄罗斯转盘，那么把嵌套弄成这样一团糟也无所谓。不过如果你写HTML时没有正确地嵌套元素，也许你的页面在一些浏览器上能正常工作，但在另外一些浏览器上可能就不正常了。如果能很好地掌握嵌套，就能避免标记不匹配，确保你的HTML在所有浏览器上都能正常工作。在后面几章详细讨论“工业强度的HTML”时这一点会更为重要。

扮演浏览器

下面的HTML文件中有一些不匹配的标记。你的任务是扮演浏览器，找出所有错误。完成这个练习后，对照本章最后的答案看看你有没有找出所有错误。



```
<html>
<head>
  <title>Top 100</title>
<body>
<h1>Top 100
<h2>Dark Side of the Moon</h2>
<h3>Pink Floyd</h3>
<p>
  There's no dark side of the moon; matter of fact <q>it's all dark.
</p></q>
<ul>
  <li>Speak to Me / Breathe</li>
  <li>On The Run</li>
  <li>Time</li>
  <li>The Great Gig in The Sky</li>
  <li>Money</li>
  <li>Us And Them</em>
  <li>Any Colour You Like</li>
  <li>Brain Damage</li>
  <li>Eclipse</li>
</ul>
</p>
<h2>XandY</h3>
<h3>Coldplay</h2>
<ol>
  <li>Square One
  <li>What If?
  <li>White Shadows
  <li>Fix You
  <li>Talk
  <li>XandY
  <li>Speed of Sound
  <li>A Message
  <li>Low
  <li>Hardest Part
  <li>Swallowed In The Sea
  <li>Twisted Logic
</ol>
</ul>
</body>
</head>
```

我是谁?



一堆HTML元素身着盛装在玩一个派对游戏“我是谁？”它们给了一个线索，你要根据它们说的好好猜猜它们到底是谁。当然假设它们说的都是真话。确认它们的身份，填在右边的空格上。另外，对于每个参加者，还要写出这个元素是内联元素还是块元素。

今晚参加派对的有：

目前为止你见过的所有迷人的HTML元素都可能到场！

名字

内联元素还是块元素？

我是1级标题。

.....

.....

我时刻准备链接到另一个页面。

.....

.....

用我强调文本。

.....

.....

我是个列表，不过我不关心顺序。

.....

.....

我是真正的换行符。

.....

.....

我是放在列表中的列表项。

.....

.....

我要保证我的列表项有序。

.....

.....

我的任务就是提供图像。

.....

.....

用我可以在段落中加引用。

.....

.....

用我来引用独立的文字。

.....

.....



我刚创建了一个Web页面来解释从这本书中学到的知识，我希望在页面中提到<html>元素。这会不会把嵌套弄乱？需要用双引号或者其他什么字符把它括起来吗？

你说的没错，这确实会有问题。

因为浏览器要用<和>来开始和结束标记，如果在HTML内容中使用这两个字符，就会有问题。不过HTML提供了一种简便的方法，可以使用一种称为字符实体（character entity）的简单缩写来指定这样一些特殊字符。它是这样工作的：对于被认为“特殊”的字符，或者你可能希望在Web页面中使用某个字符但在你的编辑器里无法输入（比如版权符号），就可以查找相应的缩写在HTML中直接输入。例如，>字符的缩写是 >，<字符的缩写是<。

所以，假如你希望在页面中输入“The <html> element rocks.”。通过使用字符实体，可以这样输入：

The **<html>** element rocks.

还有一个重要的特殊字符需要知道，这就是&字符（与字符）。如果希望HTML内容中出现一个&，则可以使用字符实体&，而不要直接使用字符&。

那么版权符号呢（版权符号是©right;）？还有所有其他符号和外文字符呢？可以在下面这个URL找到一些常用的字符实体：

http://www.w3schools.com/tags/ref_entities.asp

或者，如果需要更详尽的字符实体清单，则可以访问这个URL：

<http://www.unicode.org/charts/>

there are no Dumb Questions

问：哇，我以前都不知道浏览器居然还可以显示这么多不同的字符。www.unicode.org网站上有太多的不同字符和语言。

答：要当心。只有当你的计算机或设备安装了正确的字体时，你的浏览器才有可能显示所有这些字符。所以，你可能以为www.w3schools.com页面上的所有基本实体在任何浏览器上都可用，但实际上并不能保证所有这些实体都能显示。不过，假设你了解你的用户，应该很清楚他们的机器上通常使用哪些外文字符。

问：你说&是特殊字符，我得用字符实体&来取代它，不过要输入实体我又必须使用&。这么说，如果我要输入>实体，就必须输入&gt;，是吗？

答：不，不是这样的！&之所以特殊正是因为它是所有实体的第一个字符。所以在实体名中使用&是完全可以的，但不能单独使用&。

只要记住，输入实体时都会用到&，而如果内容中确实需要一个&，就要使用&。

问：我在www.w3schools.com上查看了字符实体，注意到每个实体还有一个编号。这个编号做什么用？

答：你可以在HTML中使用这个编号，比如d，也可以使用实体名，它们的作用都是一样的。不过，并不是所有实体都有名字，所以如果没有实体名，那么使用这个编号就是唯一选择了。



破解位置挑战

为了实现统治世界的欲望，Evel博士建立了一个私人网页，供他的党羽使用。你接收到截获的一个HTML片段，其中可能包含他的行踪线索。根据你掌握的HTML专业知识，现在要求你破解这个代码，找出他的位置。下面是他的主页的部分文字：

```
There's going to be an evil henchman meetup next
month at my underground lair in
&#208;&epsilon;&tau;&#114;&ouml;&igrave;&tau;. Come
join us.
```

提示：访问http://www.w3schools.com/tags/ref_entities.asp 和/或输入这段HTML，看看浏览器会显示什么。

元素大杂烩

使用这个元素来标记你想用不同方式展示的本，比如你想强调一个要点。

**** 使用这个元素来标记你希望特别强调的文本。

<pre> 希望浏览器按你输入的方式原样显示文本时，使用这个元素来指定文本的格式。

只需要建立链接，就需要<a>元素。

<a>

使用这个元素提供短引用……你懂的，比如“to be or not to be,”或“No matter where you go, there you are.”。

<q>

给我一个段落，谢谢。
<p>

code元素用来显示计算机程序代码。

<code>

<time>

这个元素告诉浏览器这个内容是一个日期或时间，或者同时包含日期和时间。

需要显示一个列表？比如说，菜谱中的配料列表，或者一个任务列表？可以使用元素。

如果需要一个有序列表，就要使用元素。

对应列表中的列表项，如巧克力、热巧克力、巧克力酱……

这是一个void元素，用来提供换行。

**
**

这个元素用来在页面中包含一个图像，比如照片。

这个元素用于长引用，就是你希望作为一大段文字来强调的内容，比如从书中节选的一部分。

<blockquote>

这里有一大堆你已经知道的元素，还有一些你没有见过的元素。

要记住，HTML的一半乐趣在于动手实践！所以你要自己创建一些HTML文件来试试。

页面太棒了。它完美地再现了我的旅程，提供了一个绝妙的网络版旅行日志！而且HTML组织得也很好，这样我就能自己增加新素材了。不过现在它还只是在你的计算机上，什么时候能把它真正上传到Web呢？



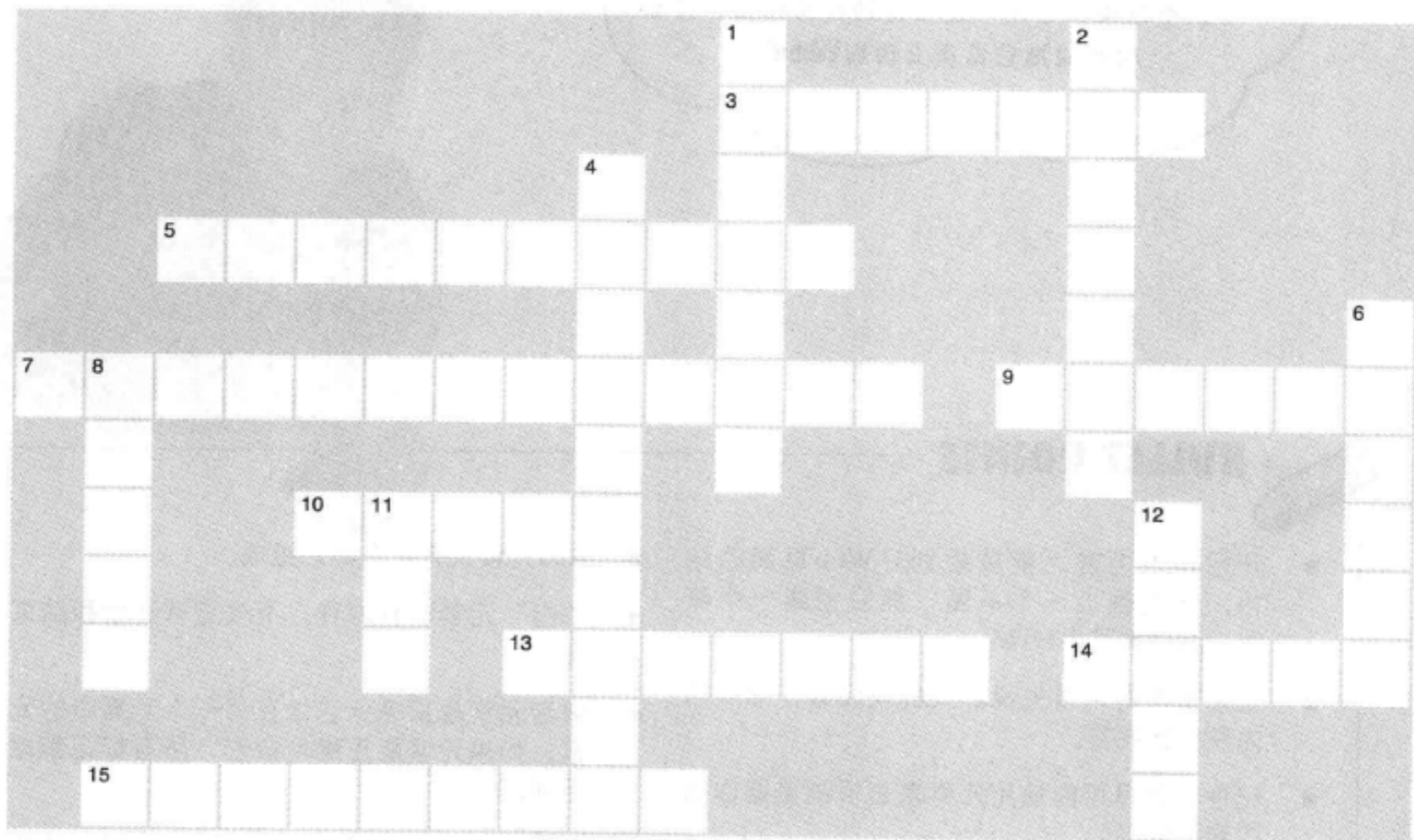
BULLET POINTS

- 开始输入内容之前要规划好Web页面的结构。首先画出一个草图，然后创建一个略图，最后再写出HTML。
- 规划页面时，首先设计大的块元素，然后用内联元素完善。
- 记住，要尽可能使用元素来告诉浏览器你的内容的含义。
- 一定要使用与内容含义最接近的元素。例如，如果需要一个列表，就不要使用段落元素。
- `<p>`、`<blockquote>`、``、``和``都是块元素。它们单独显示，在内容前后分别有一个换行（默认地）。
- `<q>`和``是内联元素。这些元素中的内容与其包含元素的其余内容放在一起。
- 需要插入你自己的换行时，可以使用`
`元素。
- `
`是一个“void”元素。
- void元素没有内容。
- void元素只有一个标记组成。
- “空”元素没有内容。不过它有开始和结束标记。
- 嵌套元素是指完全包含在另一个元素中的元素。如果元素能正确地嵌套，所有标记都能正确匹配。
- 要结合两个元素建立一个HTML列表：使用``和``建立有序列表，使用``和``可以建立一个无序列表。
- 浏览器显示一个有序列表时，它会为列表创建序号，所以无需你费心。
- 可以在列表中建立嵌套列表，将``或``元素放在``元素中。
- 要对HTML内容中的特殊字符使用字符实体。



HTML填字游戏

该让你的右脑歇一歇了，让左脑开动起来：所有单词都与HTML有关，而且都在本章出现过。



横向

1. Tony的交通工具。
8. 著名的路边广告牌。
10. `<q>`是这种类型的元素。
11. 另一个void元素。
13. 没有内容的元素。
14. 页面的主要构建模块。
15. 使用``建立这种列表。

纵向

2. 驾车一同离开小镇的人。
3. Segway的最大时速。
4. Tony不会做这些。
5. 把一个元素放在另一个元素中叫做什么？
6. 需要两个元素。
7. 提供引用的块元素。
9. 使用``建立这种列表。
12. void元素没有这些。



Exercise Solution

下面是Tony改写的老子的话，这里使用<q>元素实现。你对你的答案做了测试吗？

这是改动过的部分……

<p>

My first day of the trip! I can't believe I finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cell phone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, <q>A journey of a thousand miles begins with one Segway.</q>

</p>

我们在引用开始之前增加了<q>开始标记，并在最后增加了</q>结束标记。

注意我们还去掉了双引号。

这里是我们的测试……



OK，尽管看起来没有不同，不过你现在感觉是不是好多了？

My first day of the trip! I can't believe I finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me: cellphone, iPod, digital camera, and a protein bar. Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one Segway."



Exercise Solution

这是Tony旅行日志中的另一个列表：cell phone、iPod、digital camera和一个protein bar。在他6月2日的日志中可以找到。这是一个无序列表。

再在“journal.html”文件中完成这些修改，和你的答案一样吗？

```
<h2>June 2, 2012</h2>
```

```

```

```
<p>
```

My first day of the trip! I can't believe I finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me:

```
</p>
```

```
<ul>
```

```
<li>cell phone</li>
```

```
<li>iPod</li>
```

```
<li>digital camera</li>
```

```
<li>and a protein bar</li>
```

```
</ul>
```

```
<p>
```

Just the essentials. As

Lao Tzu would have said, **<q>A journey of a thousand miles begins with one Segway.</q>**

```
</p>
```

首先，结束上一个段落。

开始这个无序列表。

把各个列表项分别放在一个元素中。

结束这个无序列表。

需要开始一个新段落。

扮演浏览器

答案



```

<html>
<head>
  <title>Top 100</title>
<body>
<h1>Top 100
<h2>Dark Side of the Moon</h2>
<h3>Pink Floyd</h3>
<p>
  There's no dark side of the moon; matter of fact <q>it's all dark.
</p></q>
<ul>
  <li>Speak to Me / Breathe</li>
  <li>On The Run</li>
  <li>Time</li>
  <li>The Great Gig in The Sky</li>
  <li>Money</li>
  <li>Us And Them</em>
  <li>Any Colour You Like</li>
  <li>Brain Damage</li>
  <li>Eclipse</li>
</ul>
</p>
<h2>XandY</h3>
<h3>Coldplay</h2>
<ol>
  <li>Square One
  <li>What If?
  <li>White Shadows
  <li>Fix You
  <li>Talk
  <li>XandY
  <li>Speed of Sound
  <li>A Message
  <li>Low
  <li>Hardest Part
  <li>Swallowed In The Sea
  <li>Twisted Logic
</ul>
</body>
</head>

```

缺少</head>结束标记。

缺少</h1>结束标记。

<p>和<q>没有正确嵌套：</p>标记应该在</q>标记后面。

有一个结束，这里原本应该是一个结束标记。

这里有一个结束</p>，但没有与之匹配的开头<p>标记。

这些标题的</h2>和</h3>结束标记弄混了。

开始了一个列表，但它与一个结束标记匹配。

所有结束标记都丢了。

这与以上列表的开始标记不匹配。

这是前面缺少的</head>标记，不过还缺少一个结束</html>标记。



一堆HTML元素身着盛装在玩一个派对游戏“我是谁？”它们给了一个线索，你要根据它们说的好好猜猜它们到底是谁。

今晚参加派对的有：

目前为止你见过的很多迷人的HTML元素都可能到场！



我是1级标题。

我时刻准备链接到另一个页面。

用我强调文本。

我是个列表，不过我不关心顺序。

我是真正的换行符。

我是放在列表中的列表项。

我要保证我的列表项有序。

我的任务就是提供图像。

用我可以在段落中加引用。

用我来引用独立的文字。

名字

h1
.....

a
.....

em
.....

ul
.....

br
.....

li
.....

ol
.....

img
.....

q
.....

blockquote
.....

内联元素还是块元素？

block
.....

嗯.....

inline
.....

block
.....

嗯.....

block
.....

block
.....

inline
.....

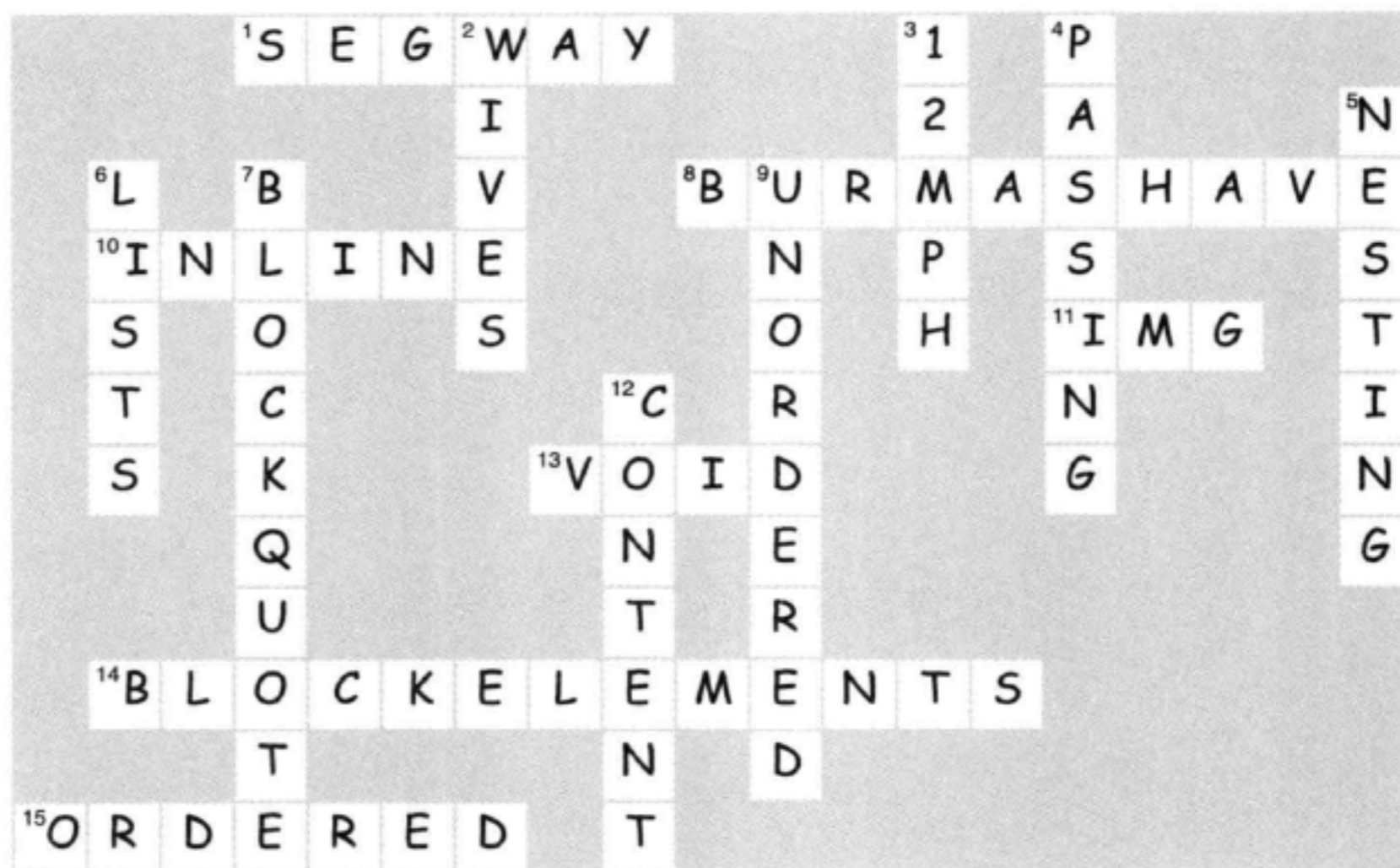
inline
.....

block
.....

嗯，看起来像是一个内联元素，但是 `<a>` 还可以包围块元素，而不只是文本。所以，根据具体的上下文，`<a>` 既可以是内联元素，也可以是块元素。

糊涂了？
`
` 是块元素和内联元素之间的一个模糊地段。它确实会创建一个换行，不过不会像有两个 `<p>` 元素那样把文本分成单独的两块。

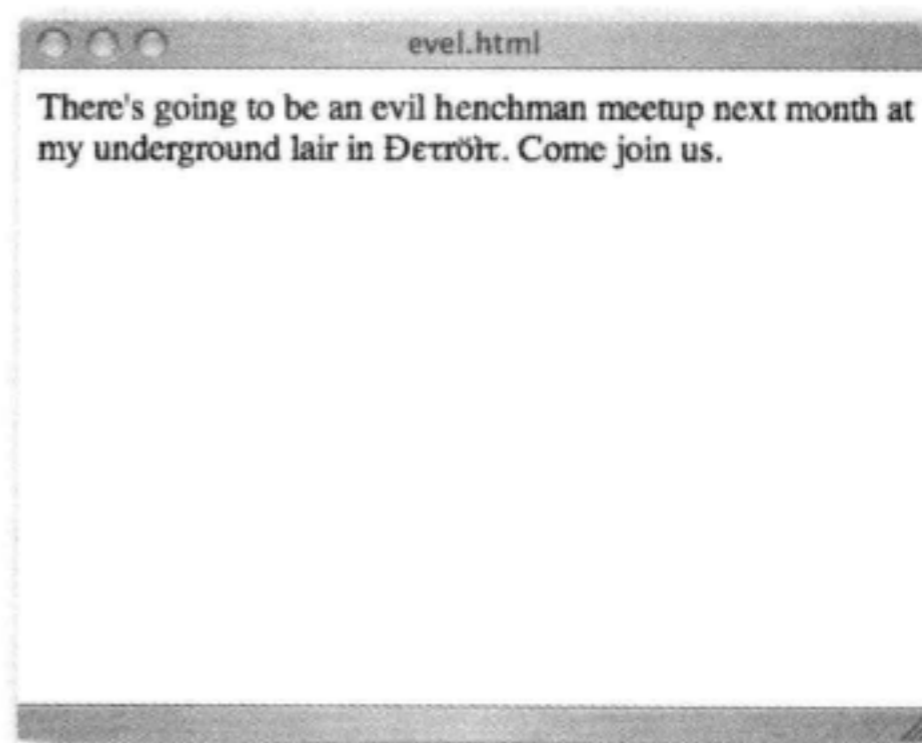
我们还没有详细讨论这个元素，不过，没错，`` 确实是内联元素。先记住这一点，我们在第5章还会专门介绍。



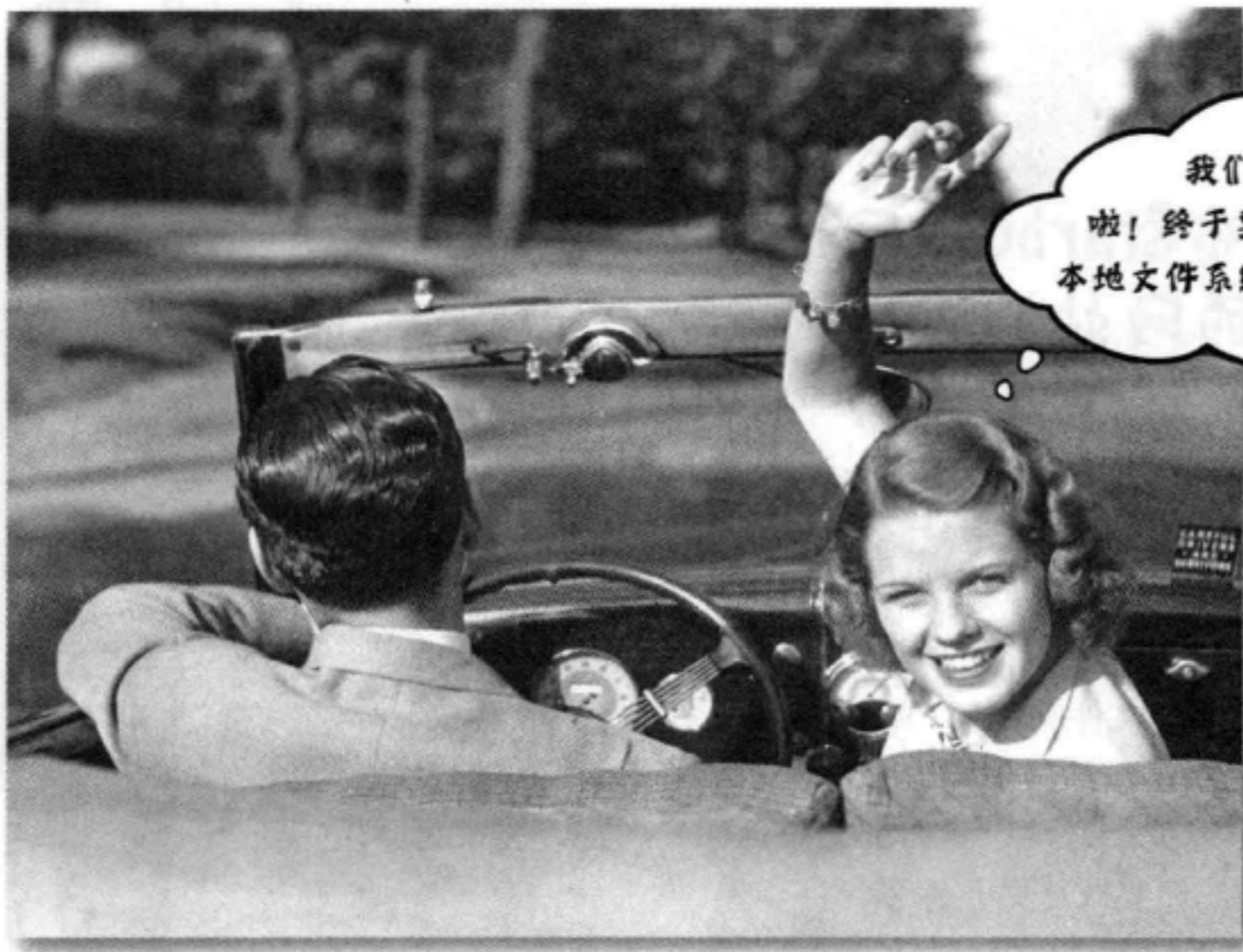
破解位置挑战

你可能已经查出每个实体，或者已经输入这个HTML。不管怎样，答案看来都是Detroit！

There's going to be an evil henchman meetup next month at my underground lair in `Ðετröìτ`. Come join us.



Web镇之旅



Web页面是互联网给我们的最好的一道菜。到目前为止，你只是在自己的计算机上创建HTML页面，所链接的页面也都在自己的计算机上。接下来我们会有所改变。这一章会鼓励你把这些Web页面发布到互联网上，让你的朋友、粉丝和客户都能真正看到。通过破解 `h, t, t, p, :, /, /, w, w, w` 代码，我们还将揭开链接到其他页面的秘密。好了，收拾好行李，向下一站Web镇进发。

提醒一句：一旦到了Web镇，你可能就永远不会回来了。别忘了给我们寄明信片。

还记得我吗？第1章见过的。你还要把Starbuzz网站上线，让我们的顾客都能真正看到它。

在Web上发布Starbuzz网站 (或你自己的网站)

你自己都没有想到，居然这么快就要把Starbuzz网站（或者也许更棒，要把你自己的网站）发布到Web上。你要做的就是找到一个“Web托管公司”（以后我们就把它叫做“托管公司”），在他们的服务器上托管你的页面，然后把页面从你的计算机复制到他们的某个服务器上。

当然，如果了解你的本地文件夹如何“映射”到服务器上的文件夹，另外知道一旦把页面放在服务器上，该如何指示浏览器访问你的页面，那就好办了。不过关于这些，我们都会告诉你。现在，先来讨论如何发布到Web上。你要做这样几件事：

- 1 找一家托管公司。
- 2 为你的网站选一个名字（比如www.starbuzzcoffee.com）。
- 3 想办法把文件从你的计算机上传到托管公司的服务器上（这样的方法有很多）。
- 4 把你的新网站告诉你的朋友、家人和粉丝，开始享受你的成果。



我们会带你一步一步完成网站的发布，即使你不打算现在就让网站上线，也可以跟着我们一起学一学，因为你会学到以后需要知道的一些重要知识。来吧，做好准备，从HTML开始我们的Web之旅吧……



找一家托管公司

要在Web上发布网页，需要一个在Web上全天候工作的服务器。你最好的选择是找一家托管公司，让他们负责考虑保证服务器一直运行的诸多细节问题。不过，不用担心，找一家托管公司很容易，也不贵。

哪家公司？嗯，我们希望你在Head First Hip Web Hosting公司完成托管，开玩笑的，这家公司实际上并不存在。所以，你得自己做点功课。尽管找一家公司托管你的页面并不难，不过这有点类似于选择有线电视公司：有太多的选择和方案。你需要货比三家，找到最合适的价位和服务。

有个好消息，即使你几乎身无分文也可以开始，可以等以后需要额外特性时再进行升级。尽管我们不会对具体哪一家提供商最好给出建议，不过可以告诉你寻找提供商时要注意的一些事项，另外我们还列出了一些比较受欢迎的提供商，见<http://wickedlysmart.com/hosting-providers/>

市场部提示：当然了，如果哪家托管公司给我们一大笔钱，我们也可以在这里给它做广告！



轻松一刻

学习这本书并不要求把页面发布到Web上。

如果你的页面能真正发布到Web上，肯定很有意思，不过，只用你自己的计算机也完全能顺利学完这本书。

不管是否真正发布网站，都应当好好学习后面几页，了解网站发布的全过程。



一分钟托管指南

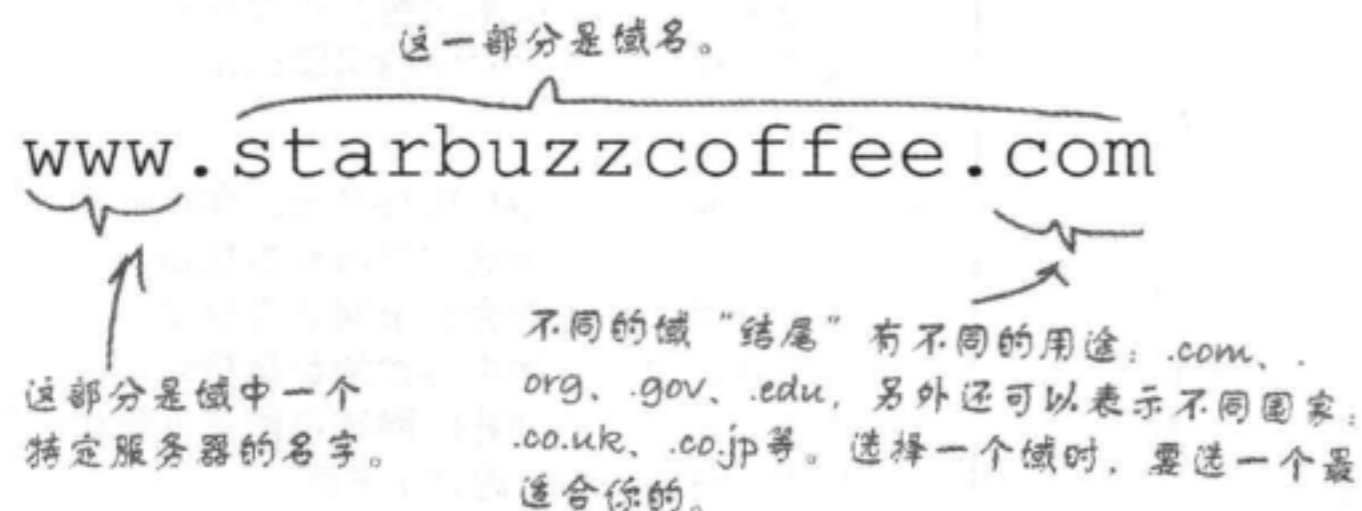
我们肯定没办法把选择托管公司需要了解的方方面面全部告诉你（毕竟，这是一本关于HTML和CSS的书），不过我们可以给你指出一个正确的方向。选择托管公司时要考虑以下几点。

- 技术支持：这家托管公司有没有一个好的系统来处理你的技术问题？比较好的公司会通过电话或邮件对你的问题迅速做出回应。
- 数据传输：这是指托管公司允许你在一定时间内向访问者发送的页面和数据量。大多数托管公司可能会为小网站提供最基本的方案，这些方案中提供的数据传输量并不大。如果你要创建的网站可能有大量访问者，就要仔细考虑这个方面。
- 备份：这家托管公司是否会定期对你的页面和数据做备份，从而在服务器出现硬件故障时能够恢复？
- 域名：托管公司定价中是否包括一个域名？有关的更多内容会在下一页介绍。
- 可靠性：大多数托管公司都声称保证网站99%以上的时间内都能正常运行。
- 赠品：是否附送了其他赠品，比如Email地址、论坛或脚本语言支持（这对你将来会很重要）？

域 你好,我的名是……

也许你从未听说过“域名”这种说法,但实际上你肯定早已经见过并用过无数次域名了,要知道……google.com、facebook.com、amazon.com、disney.com还有一些你不希望我们提到的网站都是域名。

那么域名到底是什么?就是一个唯一的名字,可以用来定位你的网站。下面给出一个例子:



我们要重视域名,这有两方面原因。如果你希望你的网站有独一无二的名字,就需要有自己的域名。域名还会用来将你的页面链接到其他网站(后面几页会介绍这个内容)。

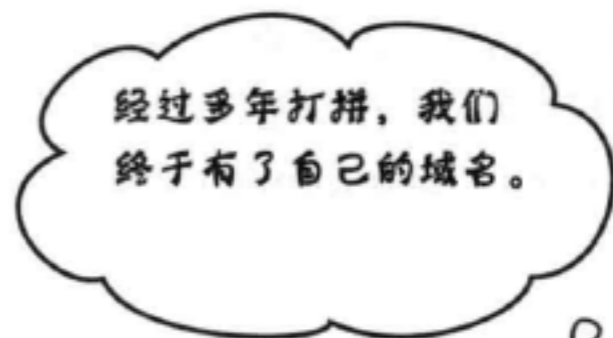
还有一件事你要知道。域名由一个集中的权威机构(ICANN)控制,以确保一次只能有一个人使用某个域名。另外(你可能已经预料到了),要保留你的域名,每年还需要交纳少许注册费。

如何得到一个域名?

简单的答案是,这个问题可以交给你的托管公司来考虑。他们通常会提供域名注册作为其服务中的一项。不过,还有上百家公司很乐于提供帮助,可以在这里找到这些公司的一个清单:

<http://www.internic.net/regist.html>

与寻找托管公司一样,恐怕也得你亲自去寻找和注册你自己的域名。你会发现,交给托管公司来办是最容易的。



there are no Dumb Questions

问:为什么把它叫做“域名”而不是“网站名”?

答:因为它们完全是两码事。如果你看到www.starbuzzcoffee.com, 这就是一个网站名, 不过如果只有“starbuzzcoffee.com”部分, 就是域名。还可以创建使用相同域名的其他网站, 比如corporate.starbuzzcoffee.com或employees.starbuzzcoffee.com。所以域名可以用于多个网站。

问:如果我想获得Starbuzz的域名, 是不是会得到www.starbuzzcoffee.com? 看起来所有人都在使用www开头的网站。

答:重申一次, 不要把域名和网站名弄混了: starbuzzcoffee.com是一个

域名, 而www.starbuzzcoffee.com是一个网站名。购买域名就像买下一块地, 比如说100mainstreet.com。在这块地上, 你可以随你所愿建很多网站, 例如: home.100mainstreet.com、toolshed.100mainstreet.com和outhouse.100mainstreet.com。所以www.starbuzzcoffee.com只是starbuzzcoffee.com域上的一个网站。

问:那么, 域名到底有什么好处呢? 我真的需要一个域名吗? 我的托管公司说我可以直接使用他们的域名, 比如www.dirtcheaphosting.com, 这样可以吗?

答:如果能满足你的需要的话, 则使用他们的域名倒也无妨。不过(特别强调一下)这样做有一个缺点: 如果你想选择另一家托管公司, 或者

如果这家托管公司破产了, 以前知道你的网站的人就没办法找到你的网站了。另一方面, 如果你有自己的域名, 就可以带着它选择新的托管公司(而且你的用户甚至根本不知道你已经换了托管公司)。

问:如果域名是独一无二的, 这说明有人可能已经在用我想用的域名, 怎么查出来呢?

答:这个问题问得好! 大多数提供域名注册服务的公司都允许你搜索一个域名是否已经被占用(有点像搜索未用的车牌号)。可以在<http://www.internic.net/regist.html>找到这些公司的一个清单。

这里有一个练习, 你要自己来完成。我们个人很乐于帮忙, 不过作为这本书的作者, 有些问题我们实在无法给出答案(另外也无法在你外出度假时帮你喂猫)。



自己试一试

现在要找到一家托管公司, 为网站获得一个域名。要记住, 你可以访问Wickedly Smart来得到一些建议和资源。另外还要记住, 即使没有完成这个练习, 你也能顺利读完这本书(不过确实应该做一做)。

我的Web托管公司:

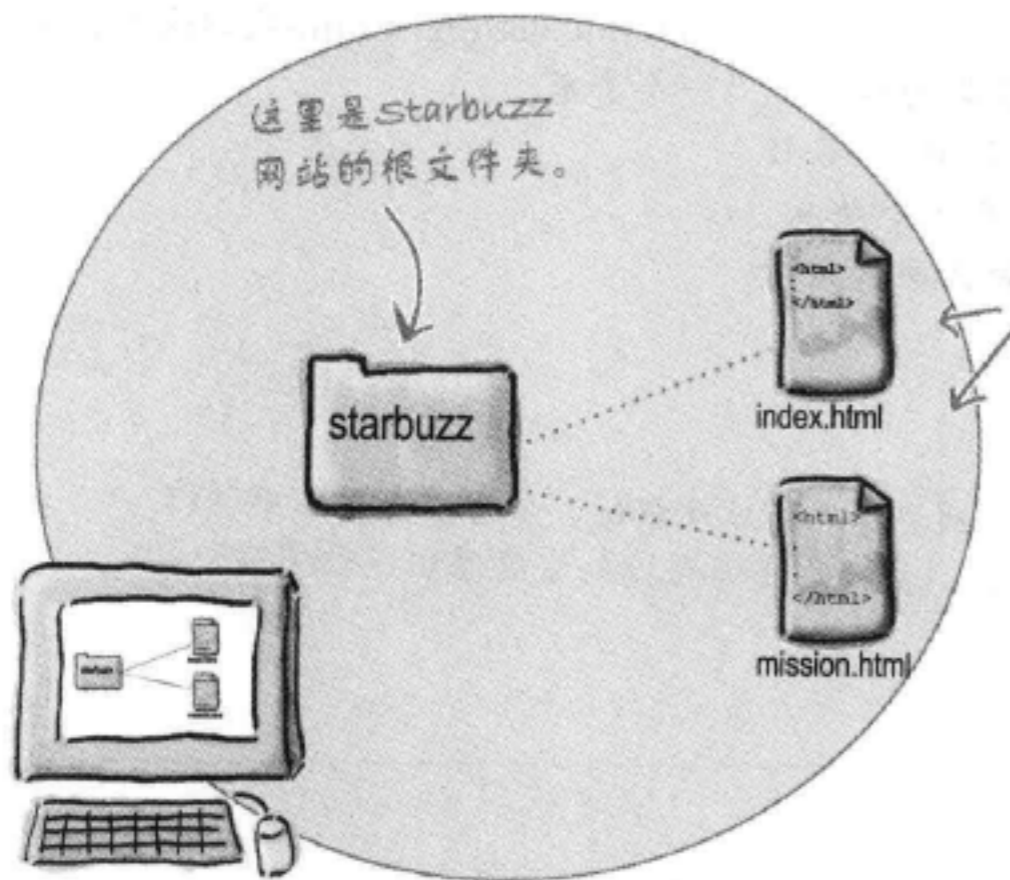
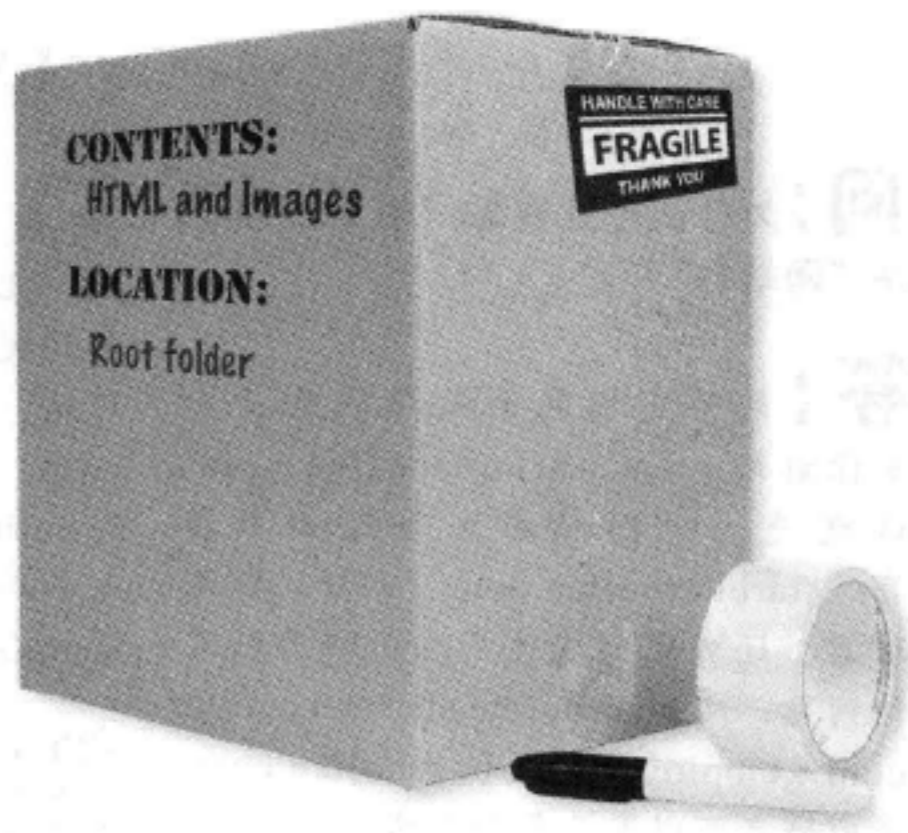
我的域名:



搬家

祝贺你！你已经找到了托管公司，得到了一个域名，另外已经找好了一个服务器来维护你的Web页面（即使还没有，也请先跟着我们继续学习，因为现在要讲很重要的内容）。

现在该做什么呢？嗯，当然是搬家。取下“出售”的牌子，整理好所有文件，我们要把它们搬到新服务器上去。与平常的搬家一样，我们的目标是把东西从一个地方（比如说老房子的厨房）搬到另一个地方（如新家的厨房）。在Web上，我们只考虑把文件从你自己的根文件夹转移到Web服务器上的根文件夹。下面回到Starbuzz，一步一步来搬。现在的情况是这样的：

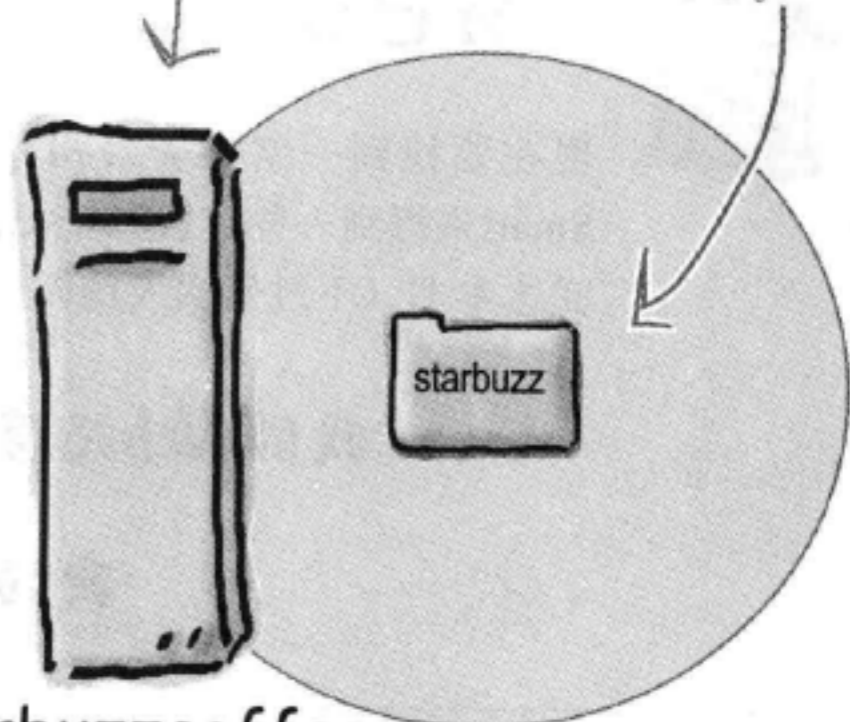


还记得Starbuzz网站的页面吗？总共有两个页面：主页面（index.html）和包含宗旨的页面（mission.html）。

你的计算机，Starbuzz网站的页面现在都在这里。

这是新的网站名。我们将使用starbuzzcoffee.com域（因为我们已经捷足先登了，所以你必须使用你自己的域名）。

这是新的Web服务器。托管公司已经为你创建了一个根文件夹，你的所有页面都要放在这里。



www.starbuzzcoffee.com

there are no Dumb Questions

问:等一下,再问一次“根文件夹”是什么?

答:到现在为止,根文件夹一直都是页面的顶级文件夹。在Web服务器上,根文件夹更为重要,因为根文件夹中的内容可以从网上访问到。

问:我的托管公司好像把我的根文件夹命名为“mydomain_com”。这样有问题吗?

答:没问题。托管公司可能会对你的根文件夹取很多不同的名字。重点在于你要知道你的根文件夹在服务器的什么位置,而且可以把你的文件复制到这里(稍后就介绍这个内容)。

问:我再确认一下。我们把这个网站的所有页面都放在一个文件夹里,这个文件夹称做根文件夹。现在我们要把它们统统复制到服务器的根文件夹,是这样吗?

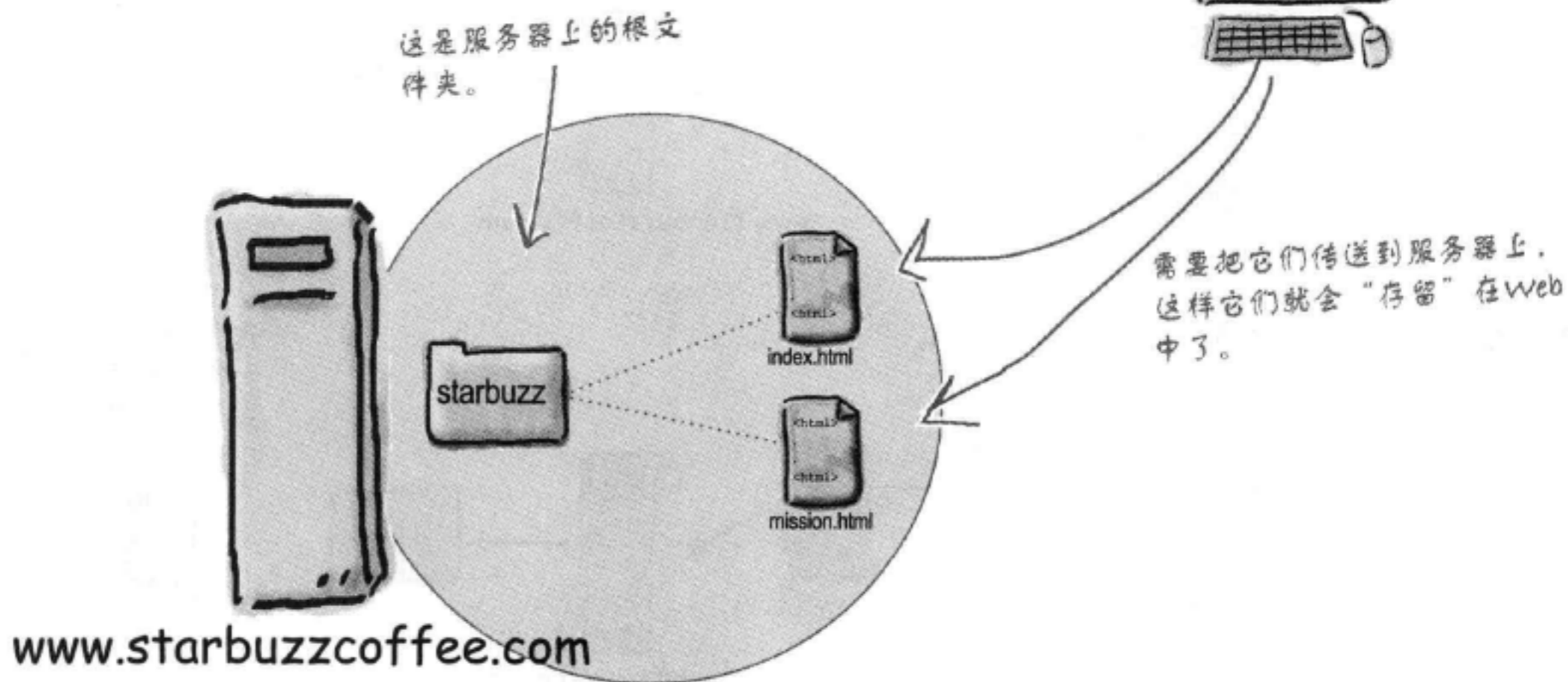
答:完全正确。你要把你自己的计算机上的所有页面统统都放到托管公司服务器上,具体来说要把它们放在对应你的网站的根文件夹下。

问:子文件夹呢?比如“images”文件夹?这些也要复制吗?

答:没错,实际上你要把你自己的根文件夹下的所有页面、文件和文件夹都复制到服务器上。所以,如果你的计算机上有一个“images”文件夹,那么服务器上也应该有这样一个文件夹。

把你的文件复制到根文件夹

现在离把Starbuzz网站发布到Web上只有一步之遥了:你已经找到托管公司服务器上的根文件夹,现在要做的就是把你的页面复制到这个文件夹。不过,怎么把文件传送到Web服务器呢?有很多办法,不过大多数托管公司都支持一种叫做FTP的文件传输方法,FTP表示文件传输协议(File Transfer Protocol)。你会发现,有很多应用允许你通过FTP传输文件,下一页会介绍这是如何工作的。



用两页尽可能讲清楚FTP

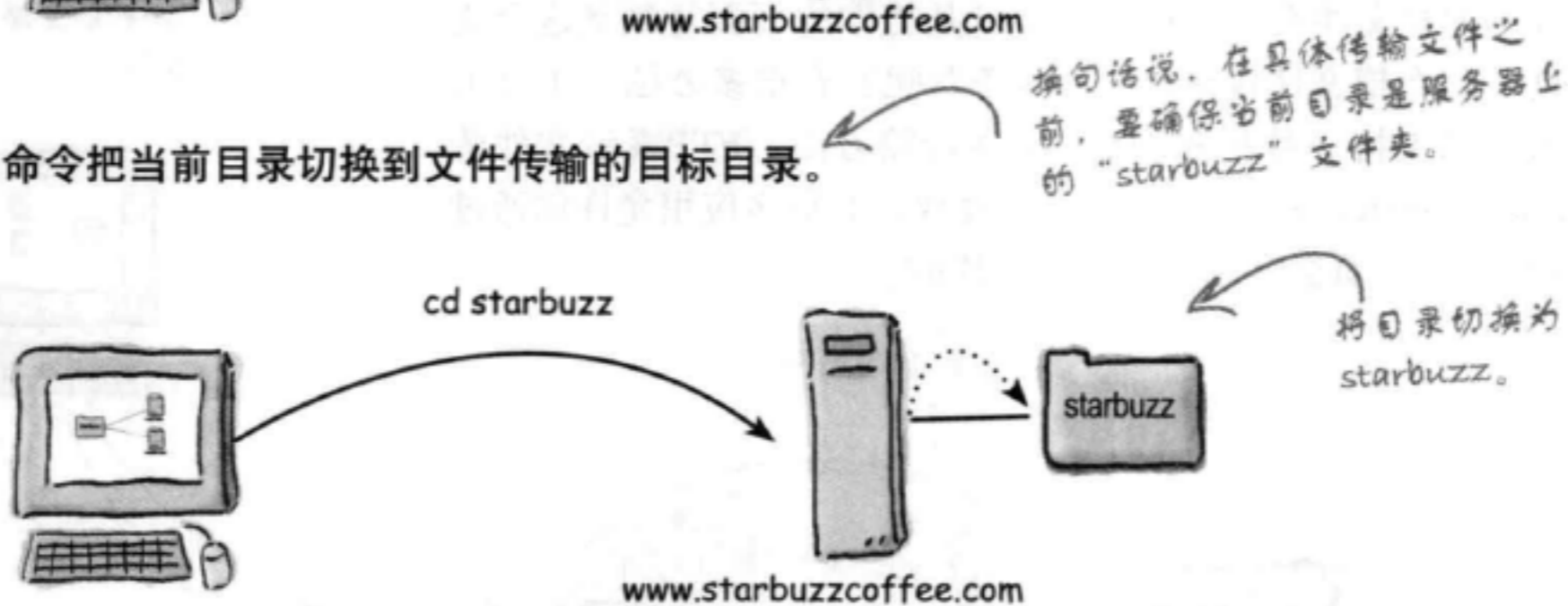
严格地说，这是一本关于HTML和CSS的书，不过我们不想让你独自面对挑战。所以，下面给出一个简短的指南，告诉你如何使用FTP将文件上传到Web。要记住，你的托管公司可能会对如何将你的文件上传到他们的服务器给出一些建议（你已经付过钱了，可以让他们来帮你）。这几页之后，我们还会回到正题上，继续讨论HTML和CSS，直到全书结束（我们保证）。

假设你已经找到一个FTP应用。有些FTP应用是命令行应用，有些有完备的图形界面，还有一些甚至内置于Dreamweaver和Expression Web之类的应用中。它们都使用相同的命令，不过有些应用要求你自己输入命令，而另外一些允许你使用图形界面。下面概要地介绍一下FTP如何工作：

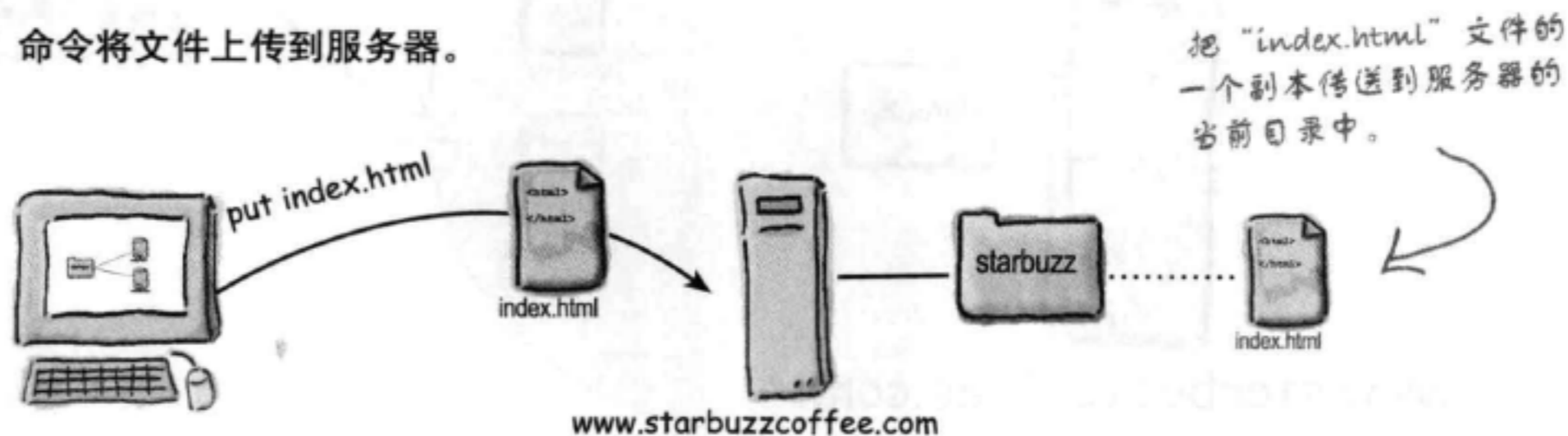
① 首先，使用FTP连接到你的服务器。



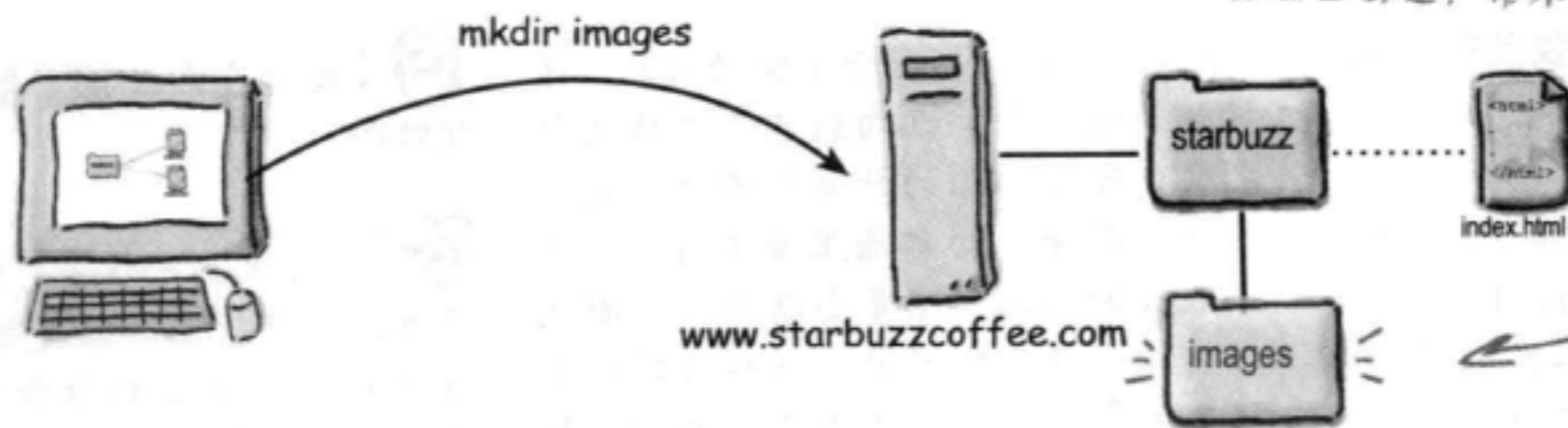
② 使用“cd”命令把当前目录切换到文件传输的目标目录。



③ 使用“put”命令将文件上传到服务器。



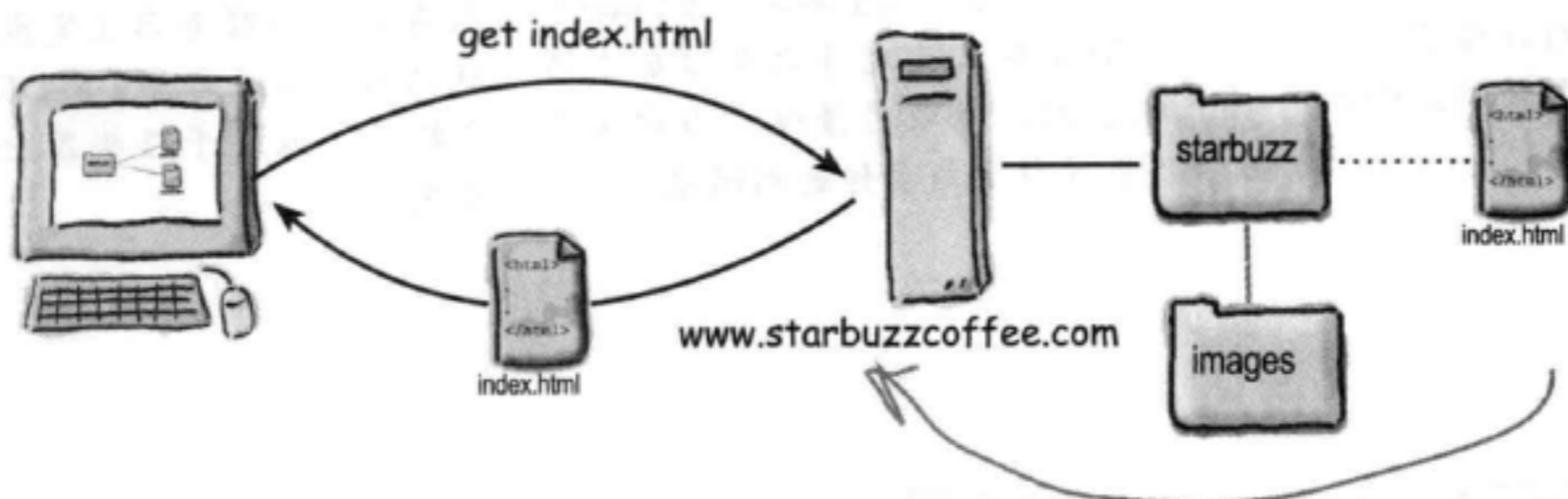
④ 还可以用“mkdir”命令在服务器上创建一个新目录。



这与新建一个文件夹类似，只不过这是在服务器上创建，而不是在你自己的计算机上。

在服务器上的starbuzz中创建一个名为“images”的新目录。

⑤ 还可以用“get”命令获取文件。



把一个文件的副本从服务器传回你的计算机。

下面把这些步骤综合起来。这里给出一个从命令行应用使用FTP的例子：

大多数FTP应用都提供了更为友好的图形界面，所以如果你在使用这样一个FTP应用，可以略过这个例子不看。

```
File Edit Window Help Jam
%ftp www.starbuzzcoffee.com
Connected to www.starbuzzcoffee.com
Name: headfirst
Password:*****
230 User headfirst logged in.
ftp> dir
drwx----- 4096 Sep 5 15:07 starbuzz
ftp> cd starbuzz
CWD command successful
ftp> put index.html
Transfer complete.
ftp> dir
-rw----- 1022 Sep 5 15:07 index.html
ftp> mkdir images
Directory successfully created
ftp> cd images
CWD command successful
ftp> bye
```

连接并登录。
查看当前目录有什么。
一个名为starbuzz的目录。
切换到starbuzz目录。
传入index.html。
查看这个目录，现在包含有index.html。
为图像创建一个目录，然后用bye命令退出。

FTP命令

不论你是在命令行上输入FTP命令，还是使用一个有图形界面的FTP应用，执行的命令或操作基本上都是一样的。

- dir: 得到当前目录的文件列表。
- cd: 切换到另一个目录。“..”也表示上一层目录。
- pwd: 显示当前目录。
- put <filename>: 将指定的文件传送到服务器。
- get <filename>: 从服务器获取指定的文件，传回你的计算机。

there are no Dumb Questions

问:我的托管公司告诉我使用SFTP,而不是FTP。有什么区别吗?

答:SFTP就是安全文件传输协议(Secure File Transfer Protocol),这是FTP的一个更安全的版本,不过工作基本上是一样的。购买之前要确保你的FTP应用支持SFTP。

问:这么说我要在我的计算机上编辑文件,然后每次想要更新网站时都要上传这些文件,是吗?

答:是这样的,对于小型的网站,通常就是这么做。在向服务器上传文件之前,可以用你的计算机测试你做的修改,确保一切都能正常工作。对于比较大的网站,通常会创建一个测试网站和一个真实网站,这样在转向真实网站之前,可以先在测试网站上预览修改的结果。

如果你使用类似Dreamweaver或Coda等工具,则这些工具允许你在你自己的计算机上测试所做的修改,当你保存文件时,文件会自动传送到网站。

问:我能在Web服务器上直接编辑文件吗?

答:这通常不是个好主意,因为在你有时间预览页面和修复错误之前,访问网站的人会看到你做的所有修改,错误也会一览无余。

不过,一些托管公司确实允许你登录服务器,在服务器上完成修改。想要这么做,你还需要了解DOS或Linux命令提示(取决于服务器上运行的操作系统)。



流行的FTP应用

下面是Mac和Windows上最流行的一些FTP应用:

对于Mac OS X:

- Fetch (<http://fetchsoftworks.com/>) 是Mac上最流行的FTP应用之一。收费。
- Transmit (<http://www.panic.com/transmit/>)。收费。
- Cyberduck (<http://cyberduck.ch/>)。免费。
- 对于Windows:
 - Smart FTP (<http://www.smartftp.com/download/>)。收费。
 - WS_FTP (<http://www.wsftple.com/>)。基本版免费,专业版收费。
 - Cyberduck (<http://cyberduck.ch/>)。免费。

大多数FTP应用都有一个测试版,可以先下载试用,再考虑是否购买。



自己试一试

这是给你留的另一个家庭作业（做完一项就打一个勾）：

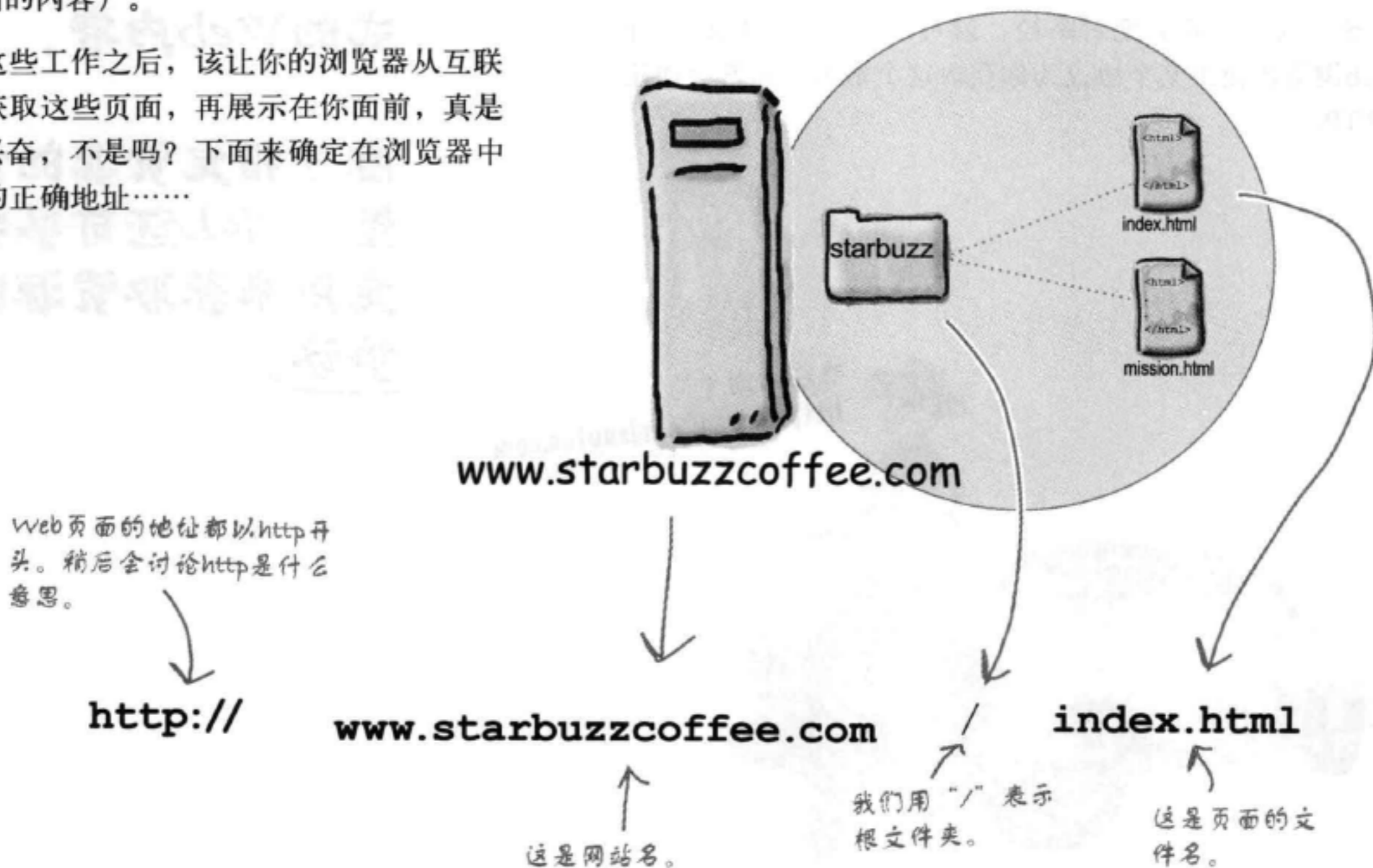
- 你要知道你的根文件夹在托管公司服务器上的位置。
- 找到从你的计算机向服务器传送文件的最佳方法（和最佳工具）。
- 现在将Starbuzz的“index.html”和“mission.html”文件传送到服务器的根文件夹中。



回到正题……

Web之旅结束了，现在我们再回到正道上。目前你已经有了两个Starbuzz页面：“index.html”和“mission.html”，它们都在你的根文件夹中（如果不是，则请先跟着我们继续学习下面的内容）。

完成这些工作之后，该让你的浏览器从互联网上获取这些页面，再展示在你面前，真是令人兴奋，不是吗？下面来确定在浏览器中输入的正确地址……



URL

我们的主干道, USA

听到“h”“t”“t”“p”“冒号”“斜杠”“斜杠”是不是很熟悉,已经听到过无数次了吧?那么这到底是什么意思?首先要知道,你在浏览器中输入的Web地址称为URL,或统一资源定位符(Uniform Resource Locators)。

如果让我们取名,则我们可能会叫它“Web地址”,不过没人问过我们,所以我们也只好沿用这个名字,把它叫做统一资源定位符。下面来剖析一个URL:

`http://www.starbuzzcoffee.com/index.html`

URL的第一部分指出用来获取资源的协议。

第二部分是网站名。现在你对这一部分应该很了解了。

第3部分是从根文件夹到资源的绝对路径。

要定位Web上的某个资源,只要你知道维护这个资源的服务器,以及资源的绝对路径,就可以创建一个URL,让Web浏览器使用某个协议为你获取这个资源,通常会使用HTTP。

统一资源定位符(Uniform Resource Locator, URL)是一个全局地址,可以用来定位Web上的任意资源,包括HTML页面、音频、视频和很多其他形式的Web内容。

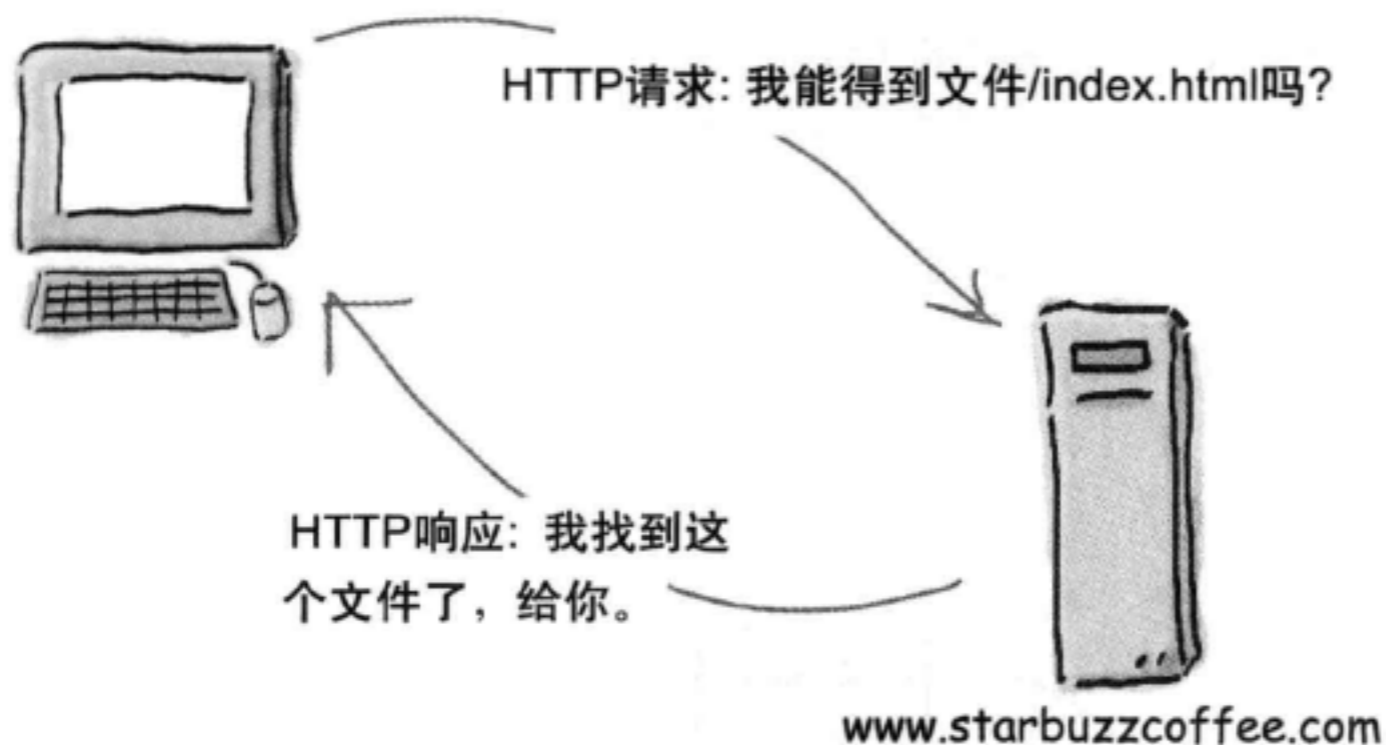
除了指定资源的位置,URL还可以指定用来获取资源的协议。



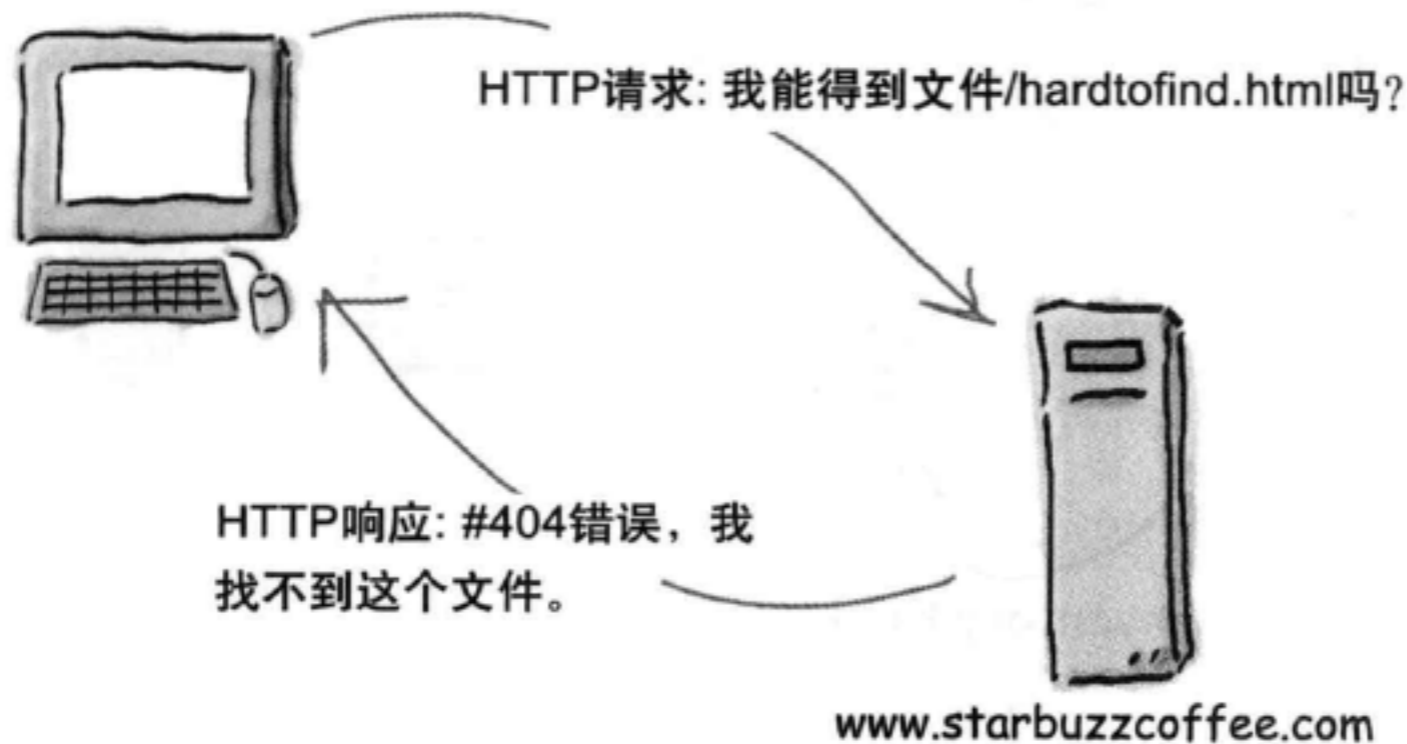
什么是HTTP?

HTTP也称为超文本传输协议 (HyperText Transfer Protocol)。换句话说, 这是在Web上传输超文本文档的公认的一种方法 (协议)。尽管“超文本文档”通常只是指HTML页面, 但这个协议实际上还可以用来传输图像或Web页面可能需要的任何其他文件。

HTTP是一个简单的请求和响应协议。它是这样工作的:



所以说, 每次在浏览器的地址栏中输入一个URL时, 浏览器就会使用HTTP向服务器请求相应的资源。如果服务器找到这个资源, 就会把它返回给浏览器, 由浏览器显示。如果服务器找不到呢?



如果无法找到资源, 则服务器会向浏览器报告我们熟悉的“404错误”。

不管怎样, 千万不要把URL读作“Earl”, 因为这是我的名字, 它应该读作U-R-L。

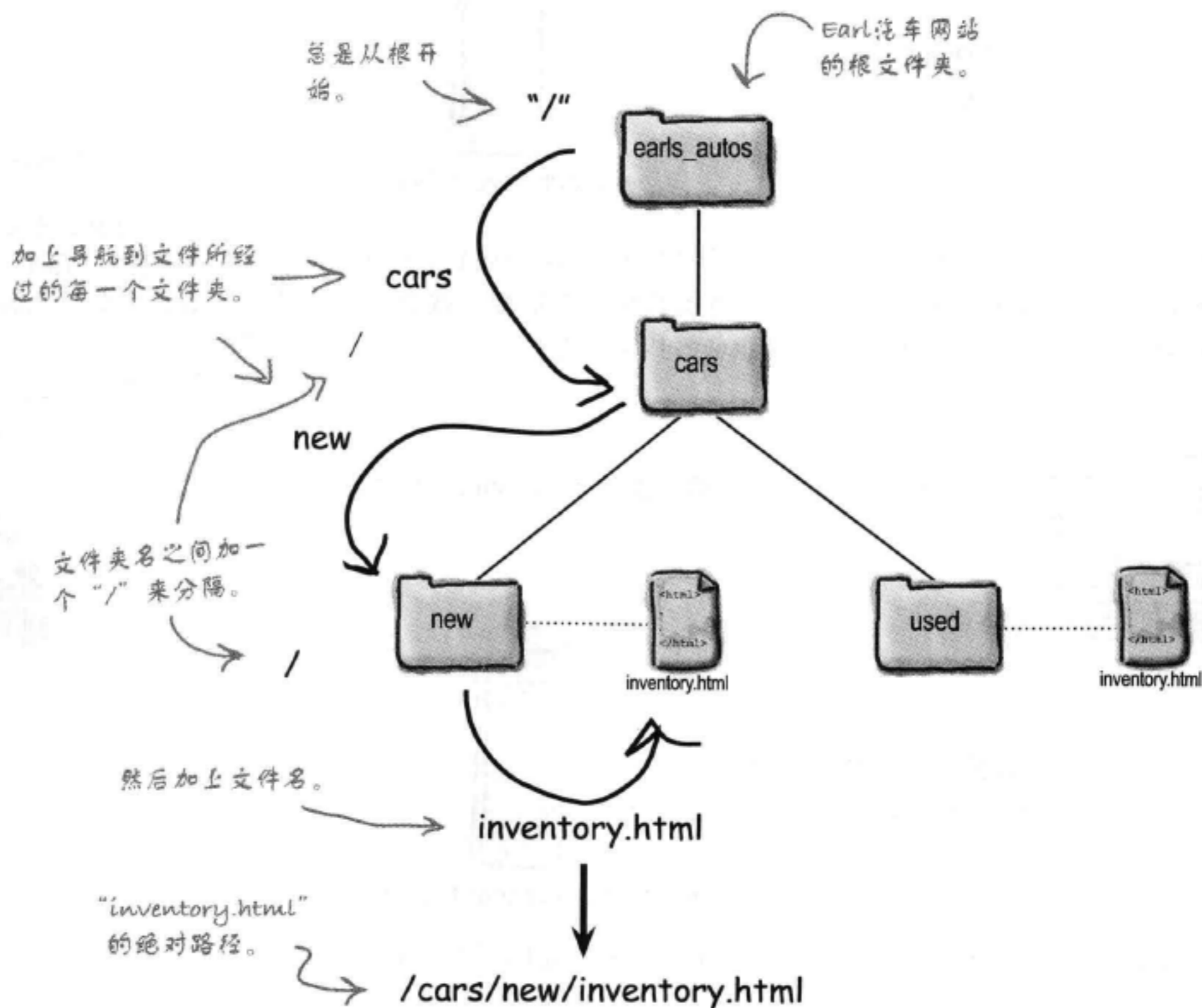


什么是绝对路径？

上一次我们讨论路径时，是要编写HTML用元素建立链接。现在我们要讨论的路径是URL中的绝对路径部分，就是在协议（HTTP）和网站名（www.starbuzzcoffee.com）之后的最后一部分。

绝对路径告诉服务器如何从你的根文件夹到达某个特定的页面或文件。例如，以Earl的汽车网站为例。假设你想查看Earl的库存，看看你订的Mini Cooper是否到货。为此，你需要确定到文件“inventory.html”的绝对路径，这个文件在“new”文件夹下。你要做的就是从根文件夹开始追踪文件夹，直到找到“inventory.html”文件所在的“new”文件夹。这个路径由沿路经过的所有文件夹组成。

所以，看起来这包括根（我们用“/”表示根）、“cars”和“new”，最后是文件本身“inventory.html”。下面把它们综合在一起：



there are no Dumb Questions

问:绝对路径有什么意义?

答:服务器需要绝对路径来找到你请求的文件。如果服务器没有得到绝对路径,它就不知道去哪里查找。

问:我觉得每个部分我都明白了(协议、服务器、网站和绝对路径),不过把它们联系在一起还是有些困难。

答:如果把所有这些部分结合在一起,你就有了一个URL,利用URL,你可以请求浏览器从Web获取一个页面(或者其他类型的资源)。怎么做到的?协议部分告诉浏览器应当使用

什么方法来获取资源(大多数情况下,协议都是HTTP)。网站部分(由服务器名和域名组成)告诉浏览器要从互联网上的哪个计算机获取资源。最后绝对路径会告诉服务器我们要找哪个页面。

问:我们学过可以在<a>元素的href属性中放入相对路径。既然它们不是绝对路径,服务器是怎么找到的呢?

答:嗨,这个问题问得好。单击一个相对链接时,在后台浏览器会根据这个相对路径和所单击页面的路径创建一个绝对路径。所以,所有Web服务器

看到的都是绝对路径,这要归功于浏览器。

问:如果在HTML中加入绝对路径,那么会不会对浏览器有帮助?

答:哈,这个问题也很妙,不过先把这个问题放一放,稍后再做解释。

Sharpen your pencil



让你久等了。现在该拿你的新URL试一试了。开始之前,先完成下面的填空,然后输入这个URL(如果还没有输入的话)。如果遇到问题,则现在可以请托管公司帮忙一起来解决问题。如果还没有找托管公司,则你可以先填上www.starbuzzcoffee.com,再在浏览器中输入这个URL。

.....://.....

协议

网站名

绝对路径

我希望访问者只需输入“http://www.starbuzzcoffee.com”，而不用再输入“index.html”。有没有办法做到？



要记住，我们讨论Web服务器或FTP时，通常会使用“目录”而不是“文件夹”。不过它们都是一样的意思。

可以的。有一种情况我们还没有谈到，就是如果浏览器向Web服务器请求一个目录而不是文件时会发生什么。例如，浏览器可能会请求：

http://www.starbuzzcoffee.com/images/

根目录中的images目录。

或

http://www.starbuzzcoffee.com/

根目录本身。

Web服务器接收到一个类似这样的请求时，它会尝试查找这个目录中的一个默认文件。通常默认文件名为“index.html”或“default.htm”，如果服务器找到这样一个文件，就会把它返回给浏览器显示。

所以，要从根目录（或任何其他目录）默认地返回一个文件，只需要把这个文件命名为“index.html”或“default.htm”。

不过你要了解你的托管公司希望你如何命名默认文件，因为这取决于他们使用哪种类型的服务器。

不过我请求的是“http://www.starbuzzcoffee.com”，看起来稍有不同。末尾没有“/”。

确实如此。如果服务器接收到这样一个末尾没有“/”的请求，而且这个目录确实存在，服务器就会帮你加上末尾的斜线。所以如果服务器接收到以下请求：

http://www.starbuzzcoffee.com

它会把这个请求改为

http://www.starbuzzcoffee.com/

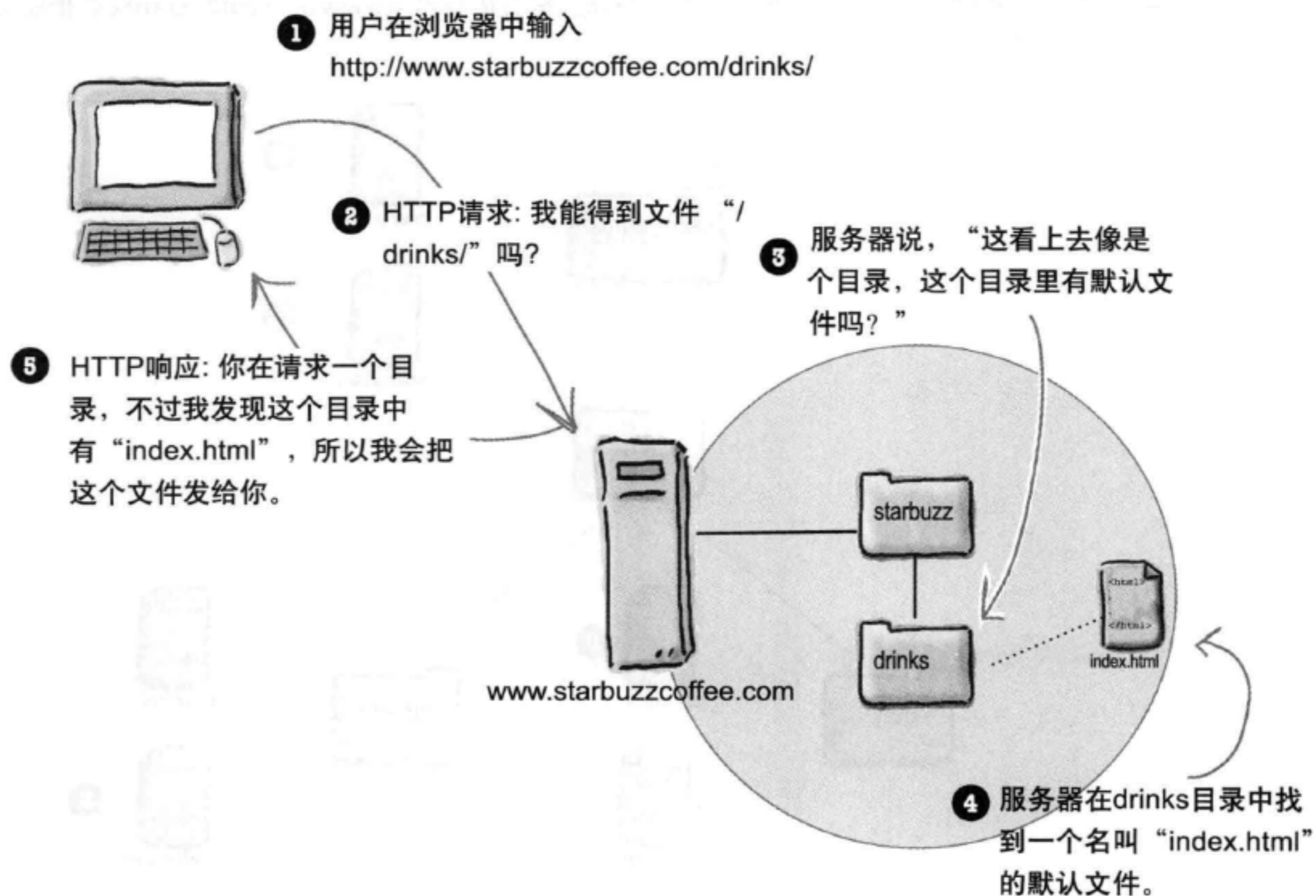
这会让服务器查找一个默认文件，最后会返回这个文件，就好像你之前输入了以下地址：

http://www.starbuzzcoffee.com/index.html



默认页面如何工作

在幕后

there are no
Dumb Questions

问: 这么说, 输入URL `http://www.mysite.com` 访问我的网站的人都会看到我的“index.html”页面, 是吗?

答: 对。或者也可能是“default.htm”, 这取决于你的托管公司使用哪种Web服务器(注意“default.htm”通常结尾没有“l”。这是Microsoft Web服务器的一个怪癖)。

还有其他一些可能的默认文件名, 如“index.php”, 你开始写脚本来生成页面时就会用到。这超出了本书的范畴, 不过你将来很可能还会遇到。

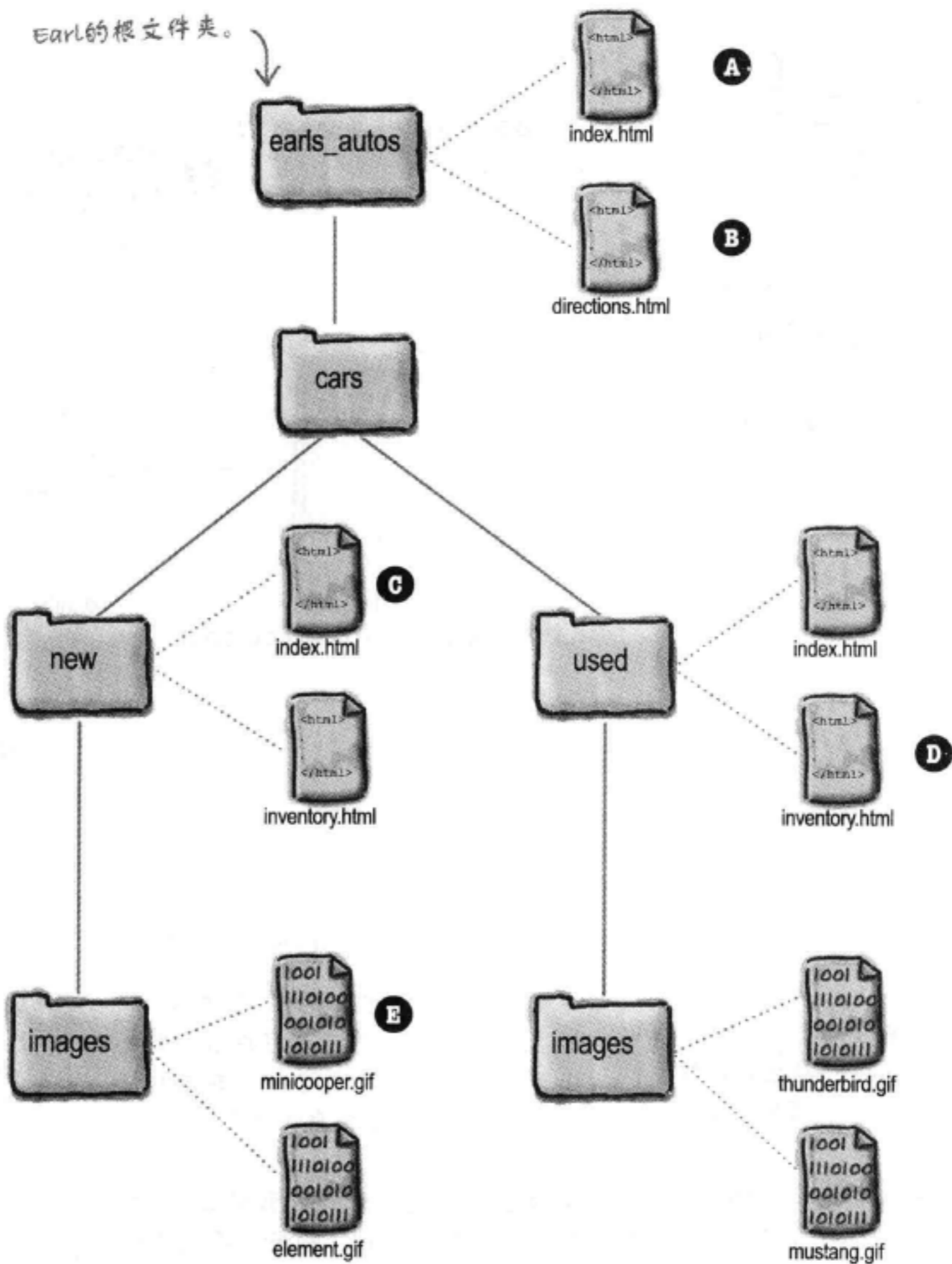
问: 所以我告诉别人我的URL时, 加上“index.html”部分是不是更好些?

答: 不。最好不要加。如果将来你改变了Web服务器, 而它使用另一个默认文件名, 比如“default.htm”, 则会怎么样呢? 或者如果你开始写脚本, 则使用“index.php”作为默认文件名, 又该如何? 这样一来, 你原来给的URL就不再有效了。

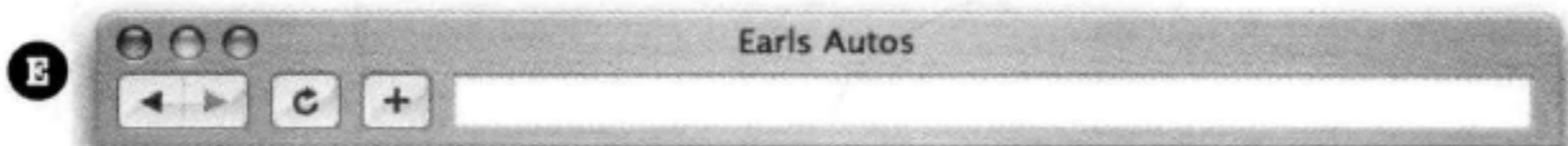
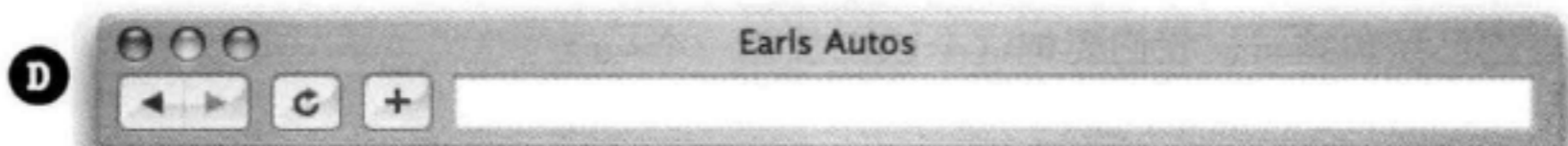
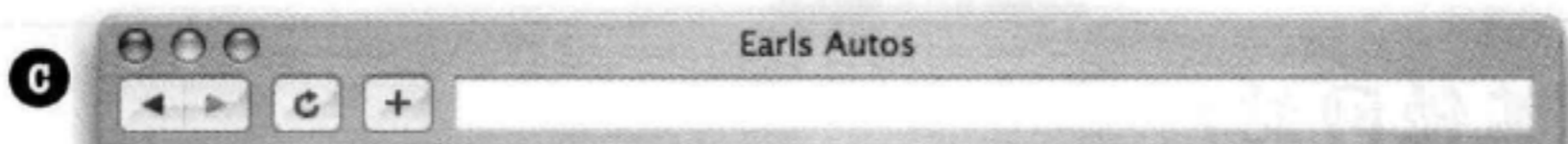
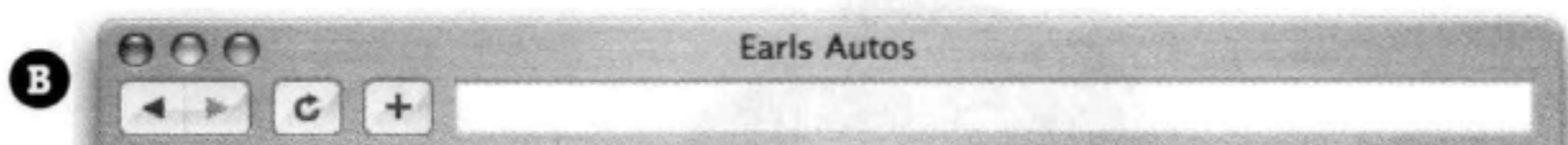
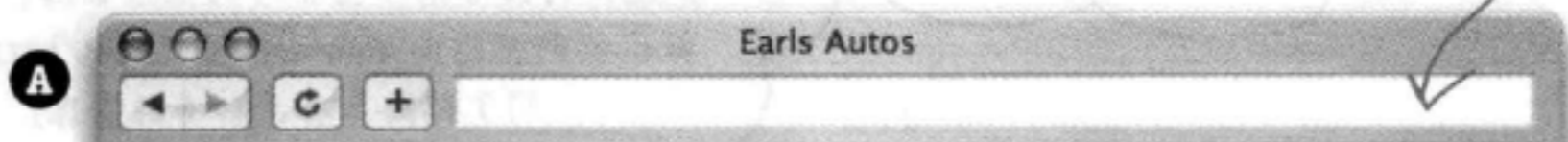


Earl需要你帮他确定URL

Earl只知道自己的名字 (Earl)，但不知道U-R-L。他需要你的帮助来确定下面各个文件的URL，分别标为A、B、C、D和E。在右边写出从www.earlsautos.com获取相应文件所需的URL。



在这里写出URL。





如何链接到其他网站？

URL并不只是在浏览器中输入，还可以在HTML中使用。当然，我们已经得到提示，Starbuzz CEO又给你布置了一个新任务：要建立一个从Starbuzz主页面到咖啡信息网站（<http://wickedlysmart.com/buzz>）的链接。你可能已经猜到了，我们要把这个URL放在一个元素中。可以这样做：

```
<a href="http://wickedlysmart.com/buzz">Caffeine Buzz</a>
```

↑
最普通、最平常的元素。

↑
我们在href中放入一个URL。单击“Caffeine Buzz”，将从wickedlysmart.com/buzz获取一个页面。

就这么简单。要链接到Web上的资源，只需要它的统一资源定位符（URL），把这个URL放在元素中作为href属性值。下面把它增加到Starbuzz“index.html”页面中。

链接到Caffeine Buzz

打开你的Starbuzz “index.html” 文件（在 “chapter4/starbuzz” 文件夹中），在最后增加两个新链接：一个指向 “mission.html” 宗旨页面的相对链接，以及一个指向Caffeine Buzz的链接。完成下面的修改，然后在你的浏览器中加载 “index.html” 文件。单击链接，感受一下Caffeine Buzz。

```

<html>
  <head>
    <title>Starbuzz Coffee</title>
    <style type="text/css">
      body {
        background-color: #d2b48c;
        margin-left: 20%;
        margin-right: 20%;
        border: 2px dotted black;
        padding: 10px 10px 10px 10px;
        font-family: sans-serif;
      }
    </style>
  </head>

  <body>
    <h1>Starbuzz Coffee Beverages</h1>
    <h2>House Blend, $1.49</h2>
    <p>A smooth, mild blend of coffees from Mexico,
      Bolivia and Guatemala.</p>

    <h2>Mocha Cafe Latte, $2.35</h2>
    <p>Espresso, steamed milk and chocolate syrup.</p>

    <h2>Cappuccino, $1.89</h2>
    <p>A mixture of espresso, steamed milk and foam.</p>

    <h2>Chai Tea, $1.85</h2>
    <p>A spicy drink made with black tea, spices,
      milk and honey.
    </p>
    <p>
      <a href="mission.html">Read about our Mission</a>.
      <br>
      Read the <a href="http://wickedlysmart.com/buzz">Caffeine Buzz</a>.
    </p>
  </body>
</html>

```

这是指向 “mission.html” 文件的链接。这里使用了一个相对路径来链接 “mission.html”。

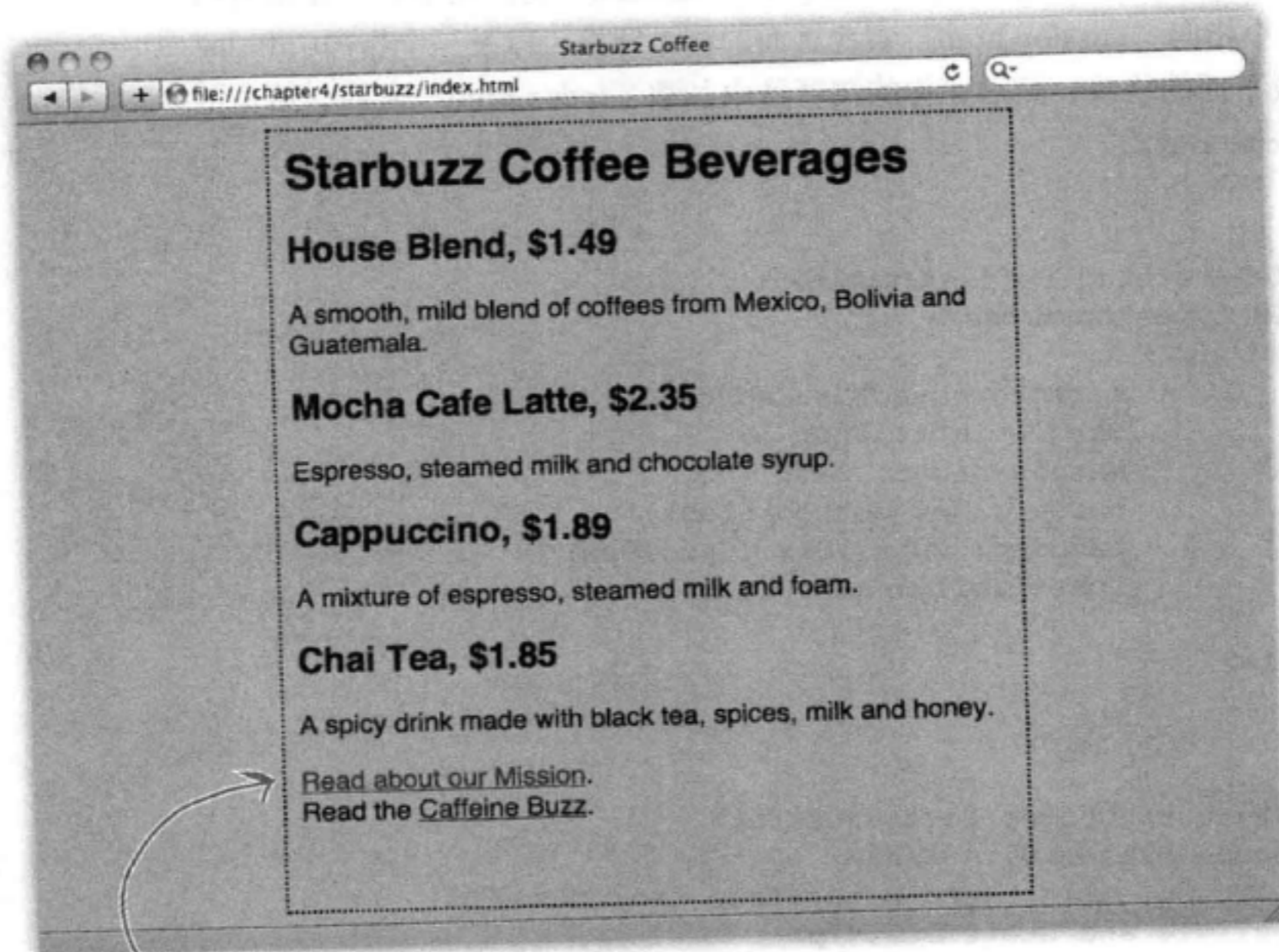
我们增加了一个
，让这两个链接分两行显示。

在这里增加指向 wickedlysmart.com/buzz 页面的链接。

这里增加了一些结构，把链接和文本组织到一个段落中。

现在来试一试……

这是增加了新链接的页面，
与我们计划的完全一致。



这里是新链接。注意，我们只链接了单
词“Caffeine Buzz”，所以它看上去与
另一个链接有所不同。

单击这个链接时，浏览器会向
wickedlysmart.com/buzz发出
一个HTTP请求，然后显示结果。



there are no Dumb Questions

在Caffeine Buzz上我们使用相对链接来链接我们网站上的其他页面，另外使用URL来链接其他网站，如：
www.caffeineanonymous.com



问：看来现在链接页面有两种方法：相对路径和URL。

答：相对路径只用来链接同一网站内的页面，而URL通常用来链接其他网站。

问：如果我对网站内部页面和外部页面都统一使用URL，那么不是更容易吗？这样也行，是不是？

答：当然，这样确实是可以的，不过出于几方面的原因你可能不想这么做。一个问题是，如果一个Web页面中有很多URL，它们会很难管理：URL很长，不容易编辑，而且也会影响HTML的可读性（对页面编写者而言）。

另外，如果一个网站都使用URL链接到本地页面，移动这个网站或者改变它的名字时，就必须修改所有这些URL来反映新的网站位置。如果使用相对路径，只要页面仍在原来的文件夹中，就不必对[a](#)元素的href属性做任何修改，因为链接都是相对的。

所以，应当使用相对链接来链接同一网站中的页面，而使用URL来链接其他网站上的页面。

问：还见过其他协议吧？开始使用Web服务器之前我总看到“file:///”。

答：没错。问得好。浏览器从你的计算机本地读取文件时会使用file协议。例如，文件URL“file:///chapter4/starbuzz/index.html”会告

诉浏览器文件“index.html”位于“/chapter4/starbuzz/”路径。取决于你的操作系统和浏览器，这个路径可能有所不同。

如果你想输入一个文件URL，则要注意重要的一点是与HTTP不同，文件URL有3个斜线而不是2个。可以这样来记：如果删去一个HTTP URL中的网站名，那么也会有3个斜线。

问：还有其他协议吗？

答：对，很多浏览器可能支持使用FTP获取页面，还有一个mail协议，可以通过Email发送数据。大多数情况下都会使用HTTP。

问：我见过这样的URL，`http://www.mydomain.com:8000/index.html`。为什么这里还有一个“:8000”？

答：“:8000”是一个可以放在HTTP URL中的可选的“端口”。可以这样来考虑端口：网站名就像一个地址，端口则像是这个地址的邮箱号（例如，在一个复合公寓里）。通常Web上的所有东西都会传送到一个默认端口（80），不过有时Web服务器会配置为在另外一个不同的端口接收请求（如8000）。这种情况经常在测试服务器上出现。正常的Web服务器几乎都在端口80接收请求。如果你没有指定端口，则默认为80。

五分钟 谜案



相对与绝对

PlanetRobots公司现在面对这样一项任务，要为它的两个分公司分别开发一个网站（PlanetRobot Home和PlanetRobot Garden），最后决定与外包公司签约来完成网站建设。RadWebDesign是一家看上去很有经验的公司，承接了Home公司网站，他们只用URL写网站的内部链接（毕竟，URL更复杂，所以应该更好一些）。另一家公司CorrectWebDesign经验较少，不过经过了很好的培训，由它接手PlanetRobot Garden网站的建设，他们使用相对路径来编写网站内部所有页面间的链接。

项目接近尾声时，PlanetRobots发出一个紧急消息：“我们被控商标侵权，所以要把域名为RobotsRUs。我们的新Web服务器将是www.robotsrus.com。”CorrectWebDesign只花了五分钟做了一些小改动，就一切OK，开始准备RobotsRUs公司总部的新网站发布会。RadWebDesign则不同，他们一直工作到凌晨4点才修复他们的页面，不过还好，总算在发布会之前完成了工作。但在发布演示中，最可怕的事情出现了：RadWebDesign项目小组组长在演示网站，他单击了一个链接，结果得到一个“404—Page Not Found”（文件未找到）错误。RobotsRUs CEO很不高兴，讽刺RadWebDesign可能应该改名叫BadWebDesign（糟糕的Web设计），然后问CorrectWebDesign能不能协助修复Home网站。

怎么回事？只是改了Web服务器名，为什么RadWebDesign会搞得一团糟？

完善的Web页面

你是不是说过“Web职业生涯”？你已经很好地完成了Starbuzz CEO要求的任务，现在已经建立了一个知名度很高的网站（你的名气也不小）。

不过你可能不想就此止步。你希望你的网站有那种专业的“完善性”，这正是普通网站与最佳网站之间的差距。在这本书中你还会看到可以用很多方法对网站“抛光”，让它更加出色，不过这里先介绍一种改进链接的方法。

为链接增加标题以便访问

对于我们要单击的链接，如果能得到更多信息就好了。这对那些有视力障碍使用屏幕阅读器的人来说尤其重要，因为他们通常不希望把整个URL读出来：（“h” “t” “t” “p” “:” “slash” “slash” “w” “w” “w” “dot”），而链接的标签通常只能提供有限的描述，如“Caffeine Buzz”。

<a>元素有一个title属性，就是用来提供链接信息的。有些人可能被这个属性名搞糊涂了，因为<head>中还有一个名为<title>的元素。之所以名字相同，这是因为它们本来就是相关的，通常建议title属性值与所链接的Web页面的<title>元素值相同。不过这不是一个严格的要求，在title属性中提供你自己的更具体的描述可能更有意义。

可以像这样为<a>增加title属性：

```
Read the <a href="http://wickedlysmart.com/buzz"
      title="Read all about caffeine on the Buzz">Caffeine Buzz</a>
```



title元素有一个值，这是所要链接的页面的文本描述。

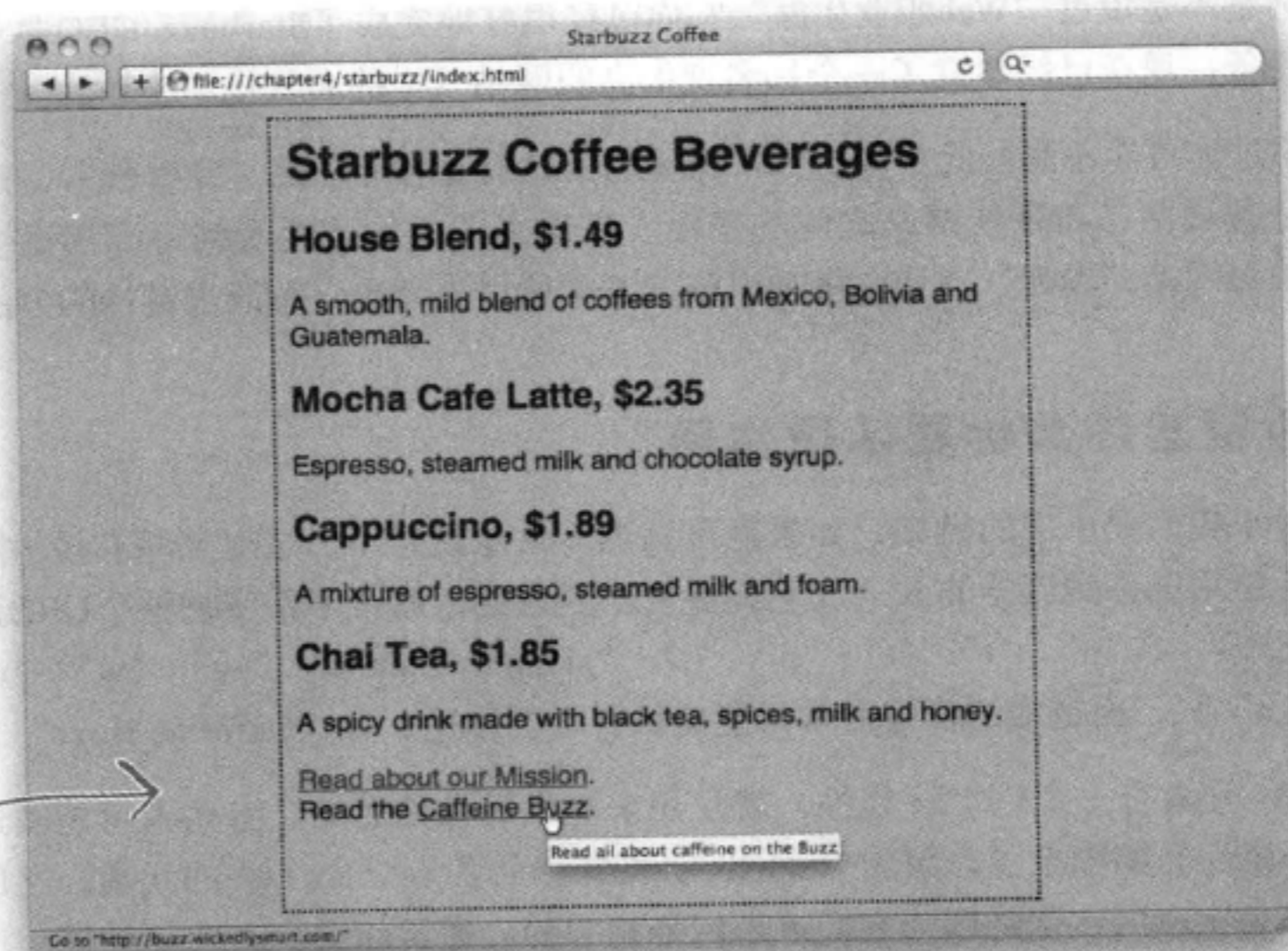


既然有了一个title属性，下面就来看访问者如何使用这个属性。不同的浏览器对title有不同的用法，不过很多浏览器都会显示一个工具提示。在你的“index.html”文件中完成上面的修改，然后重新加载页面，看看你的浏览器会如何处理。

试一试标题……

对于大多数浏览器来说，鼠标移动到一个链接上时，标题会显示为一个工具提示。要记住，对于有视力障碍的人来说，浏览器可能会大声读出链接标题。

大多数浏览器中，标题会显示为一个工具提示。把你的鼠标移到链接上，停留一秒，就可以看到这个工具提示。

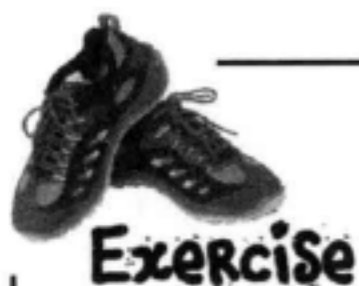


完善链接的Head First指南



进一步完善链接时要记住以下技巧：

- ◆ 保证链接标签很简洁。不要把整个句子或大段文字放在链接里。一般来讲，要保证只有几个单词。可以在title属性中提供额外的信息。
- ◆ 保证链接标签是有意义的。不要使用类似“单击这里”或“这一页”之类的链接标签。用户往往会首先粗略扫一眼页面，看看有没有链接，然后才会通读页面。所以，通过提供有意义的链接，可以改善页面的可用性。可以这样测试一下你的页面，只读页面上的链接，能理解吗？或者是不是还需要读它们周围的文字？
- ◆ 不要把链接放在一起，用户将很难区分放在一起的链接。



打开你的Starbuzz “index.html” 文件，为 “mission.html” 链接增加一个标题，设置为 “Read more about Starbuzz Coffee’s important mission”。注意这个宗旨页面链接的标签还不够简洁。可以把这个链接的标签简写为 “our Mission”。对照本章最后给出的答案，测试你做的修改。

链接处理得不错。我真希望人们能直接链接到Buzz网站的coffee部分。能做到吗？

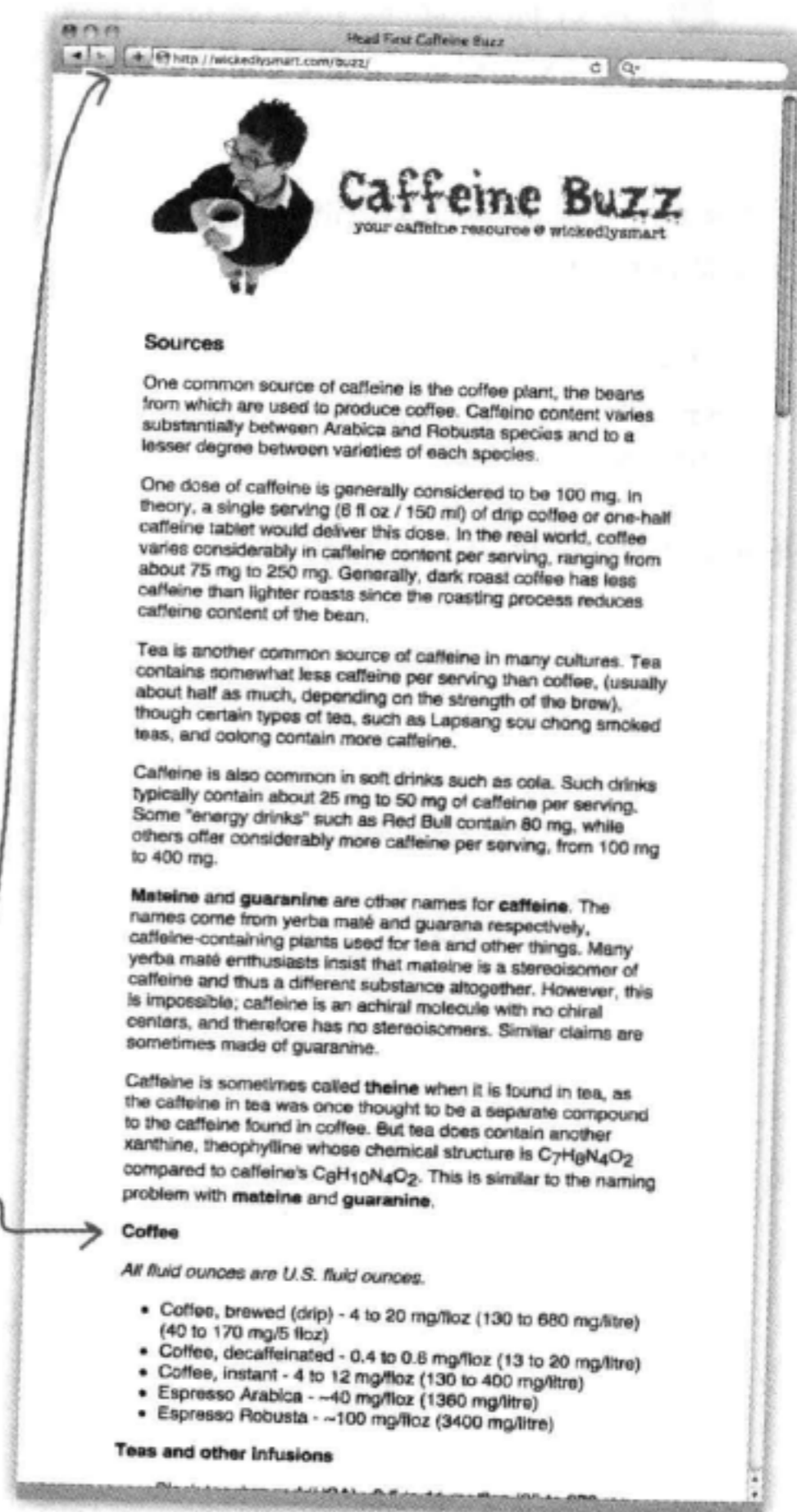


链接到一个页面

到目前为止，只要链接到另一个页面，浏览器就会加载整个页面，并从头开始显示。

不过CEO希望你能直接链接到页面中的某个特定位置：Coffee部分。

听上去不太可能？别担心，这可是Head First，我们有办法。怎么做呢？嗯，我们还没有告诉你关于[<a>](#)元素的全部呢。实际上[<a>](#)元素可以带一个id属性，允许你直接访问页面中的一个特定点。



使用id属性为<a>创建目标

我们还没有介绍过id属性，这是一个重要的属性，它有一些特殊的特性，本书后面还会更详细地讨论id的其他特殊特性。不过现在可以认为它是一种唯一标识元素的方法。带id的元素有一个特殊特性：你可以直接链接这些元素。下面来看如何使用id属性在页面中为<a>创建目标。

- 1 找到页面中你希望创建锚点的位置。这可以是页面上的任何文本，不过通常是标题。
- 2 为目标选择一个标识符名，如“coffee”或“summary”或“bio”，并在元素的开始标记中插入一个id属性。

下面来试一试。假设你希望链接Starbuzz页面上的Chai Tea（香辣奶茶），现在是这样的：

```
<h2>Chai Tea, $1.85</h2>
<p>A spicy drink made with black tea, spices, milk and honey.</p>
```

这是“index.html”中关于Chai Tea标题和描述的片段。

按照前面所说的两步，可以得到：

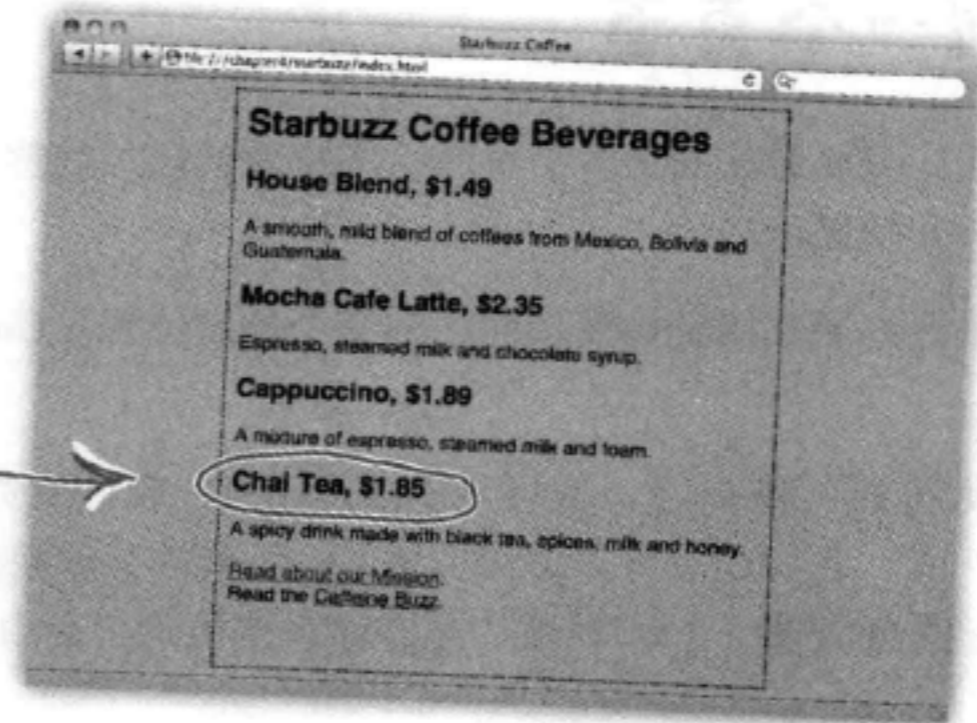
为标题的开始标记增加id。

为这个目标指定标识符“chai”。

id必须是唯一的，这很重要。也就是说，“chai”id必须是这个页面中唯一的“chai”id！

```
<h2 id="chai">Chai Tea, $1.85</h2>
<p>A spicy drink made with black tea, spices, milk and honey.</p>
```

通过指定id，你建立了一个目标，指向“index.html”页面中的Chai Tea标题。



如何用id链接到元素

你已经知道如何使用相对链接或URL链接到页面。不论是相对链接还是URL，要链接到页面中的一个特定目标，只需在链接最后加一个#，再加上目标标识符。所以，如果你想从任何Starbuzz Coffee Web页面链接到“chai”目标，可以这样写<a>元素链接：

```
<a href="index.html#chai">See Chai Tea</a>
```

遗憾的是，链接到Chai Tea并不会给人留下太深印象，因为整个页面很小，所以可以在浏览器中完整地显示。下面我们要链接到http://wickedlysmart.com/buzz的Coffee部分。你需要做到：

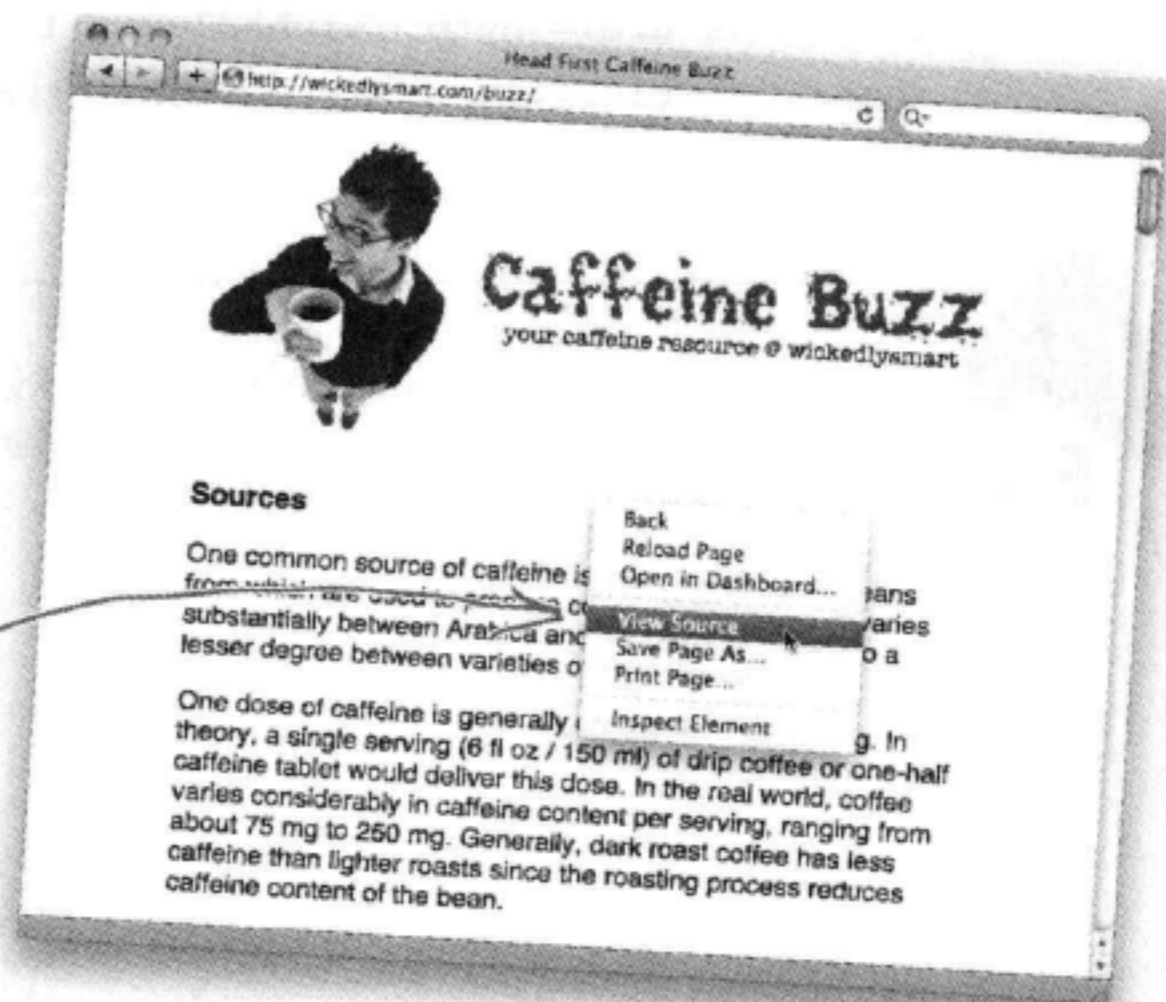
特定目标的主要好处是链接到长文件中的某个位置，使得访问者不必向下滚动文件来查找所需的部分。

- 1 明确Coffee标题的id。
- 2 修改 Starbuzz Coffee “index.html” 文件中现有的<a>元素，指向目标标题。
- 3 重新加载“index.html”页面，然后测试链接。

查找目标标题

要找到目标标题，需要查看wickedlysmart.com/buzz页面和这个页面的HTML。怎么看呢？几乎所有浏览器都有一个“查看源代码”（View Source）选项。所以，可以访问这个页面，完全加载后，选择“查看源代码”选项，你就会看到这个页面的标记。

大多数浏览器中，可以用鼠标右键单击选择“查看源代码”（View Source）。



既然得到了HTML……

向下滚动，直到看到Coffee部分；如下所示：

...

This is similar to the naming problem with `mateine` and `guaranine`.

`</p>`

`<h3 id="Coffee">Coffee</h3>`

`<p>`

`<i>All fluid ounces are U.S. fluid ounces.</i>`

`</p>`

这是Caffeine Buzz页面的一个片段。

这里是Coffee部分。可以看到它的标题和下一段的开头。

哈，这里是目标标题，名为“Coffee”。

调整“index.html”中的链接

现在要做的就是调整指向Caffeine Buzz的链接，在后面增加目标锚名，如下所示：

这是Starbuzz “index.html”文件的一个片段。

wickedlysmart.com/buzz的默认文件是index.html。所以我们把它增加到URL，这样才能结合使用目标id。

在href中增加#以及目标id。

Read the `Caffeine Buzz`



Exercise

对Starbuzz “index.html”文件完成以上修改。重新加载页面，单击“Caffeine Buzz”链接。会把你直接带到Caffeine Buzz首页的Coffee部分。



there are no Dumb Questions

问：如果一个元素里有两个属性，属性的先后顺序重要吗？例如，title属性是不是总在href后面？

答：所有元素中，属性的顺序都不重要（如果重要，那我们可就太头疼了）。所以，你可以用你喜欢的任何顺序。

问：如果元素不是，那么怎么为这样一个元素创建一个工具提示呢？

答：可以为任何元素增加title属性，所以如果希望某个元素上有一个工具提示，比如一个标题，则可以在<h1>开始标记中增加一个title属性，就像对<a>一样。有些元素不只是使用title属性作为工具提示，可能还有其他用途，不过工具提示是最常见的用法。

问：可以为任意元素增加id属性吗？

答：可以。例如，如果为元素增加一个id，那么还可以链接到段落中间。通常不太需要这么做，不过如果你愿意也无不可。

问：能不能链接到一个链接，也就是在目标中为一个<a>元素增加id属性？

答：可以！

问：我注意到，在id名中，你使用的是“chai”，所有字母都是小写，而Caffeine Buzz用的是“Coffee”，有一个大写的“C”。这有影响吗？

答：在id属性中可以使用大写和小写字符的任意组合。只是要确保保持一

致，在href和目标id中要用相同的大小写字母（正是因为这个原因，每次都全用小写字母会更容易一些）。如果没有做到一致，那就别指望你的链接在所有浏览器上都能正常工作。关于你选择的id名，最重要的一点是它在你的页面中必须唯一。

问：我可以在文档中放置一个指向相同文档中某个目标的链接吗？

答：当然可以。实际上，通常会在页面最上面（比如，在页面的顶部）定义一个目标“top”，并在页面最下面定义一个目标“Back to top”。在长文档中，往往会有整个页面的一个目录，这也很常见。例如，要链接到同一个页面上的“top”目标，可以写为Back to top。

问：为什么需要为Buzz URL增加“/index.html”来创建指向目标标题的链接呢？为什么不能直接写为：<http://wickedlysmart.com/buzz#Coffee>？

答：不行，这不一定能正常工作，因为浏览器会帮你在URL末尾加斜线，这有可能取代id引用。不过，你可以这样写：<http://wickedlysmart.com/buzz/#Coffee>，类似于使用“index.html”创建的链接，这会得到同样的结果。如果你不知道默认文件是否是“index.html”，这就会很方便。

问：如果一个Web页面没有提供一个目标，但我还需要链接到页面的某个特定部分，该怎么做呢？

答：这是办不到的。如果没有目标（换句话说，没有带id的元素），就无法指示浏览器前往Web页面中的一个特定位置。你可能需要联系页面作者，让他们增加这样一个目标（甚至可以更棒，你还可以告诉他们怎么加目标）。

问：能不能有一个类似“Jedi Mindtrick”的目标id，或者是不是id只能有一个单词？

答：要在大多数浏览器上得到一致的结果，id一定要用一个字母开头（A~Z或a~z），后面可以是任意字母、数字、横线、下划线、冒号或点号。所以尽管不能加空格，不能有类似“Jedi Mindtrick”的名字，但这个限制不算太苛刻，因为你还可以使用“Jedi-Mindtrick”、“Jedi_Mindtrick”、“JediMindtrick”之类的名字。

问：怎么告诉别人可以链接到哪些目标呢？

答：这没有既定的方法，实际上，“查看源代码”仍然是发现链接目标最古老也是最好的方法。

问：总得使用文字作为<a>元素的内容吗？

答：不是的。<a>元素能够由文字和图像（内联内容）创建链接，最近（在HTML5中）则进一步更新，现在还可以从块元素（如<p>和<blockquote>）创建链接！所以可以用<a>由各种不同内容创建链接。

五分钟
谜案



之谜底

相对与绝对

RadWebDesign的演示为什么会搞砸了？嗯，因为他们使用URL来指定href而不是相对链接，必须把每一个链接从http://www.planetrobots.com改为http://www.robotsrus.com。

能不容易出错吗？凌晨3点，有人呵欠连天，不小心输入了http://www.robotsru.com（命运真是捉弄人，CEO在演示中单击的恰恰就是这个链接）。

CorrectWebDesign则不同，他们使用相对链接来建立所有内部链接。例如，从公司宗旨声明到产品页面的链接为[../products.html](#)，不论网站叫做PlanetRobots还是RobotsRUs，这个链接都能正常工作。所以，CorrectWebDesign所要做的只是在一些页面上更新公司名字。

所以RadWebDesign灰头土脸地离开了演示会场，与之相反，CorrectWebDesign离开会场时则是眉开眼笑，又接了好几单业务。不过，故事还没有结束。在发布会之后RadWebDesign拜访了一些小咖啡馆和书店，决定不再闭门造车，还选了一本关于HTML和CSS的书攻读。后来怎么样了？请关注后面几章的“蛮力与样式迷案”。

唉……忘了在名字末尾加一个“s”。

你做到了，链接到了Buzz网站，真不赖……
我知道我总在要求修改，不过说真的，这是最后一次了。能不能在我单击时让Buzz网站在另一个窗口中显示？我不希望我的Starbuzz页面消失。

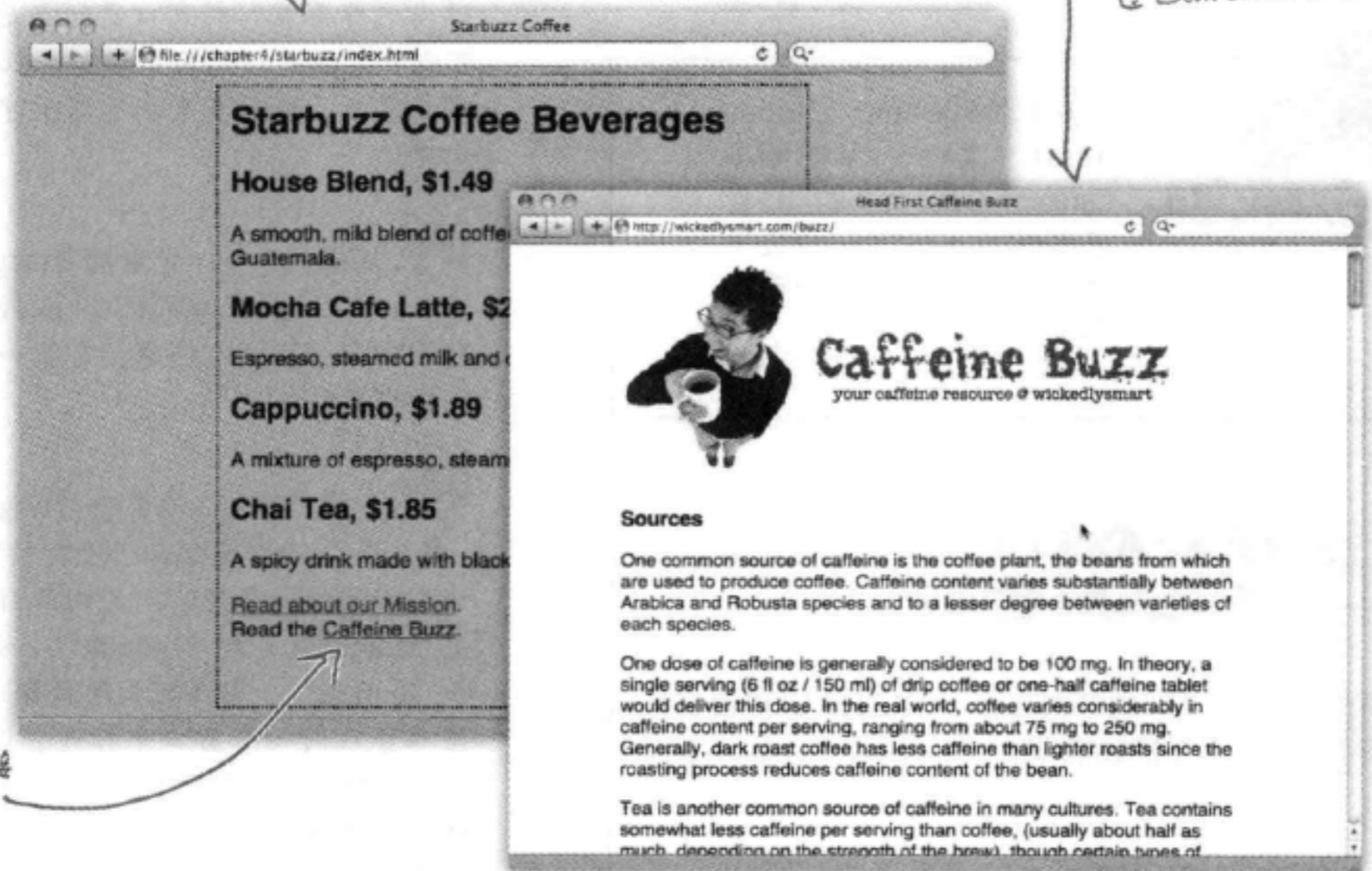


链接到一个新窗口

Starbuzz CEO又有一个新的要求（网站总是不断有新需求）。他希望这样：你单击Starbuzz Coffee页面中的“Caffeine Buzz”链接时，Starbuzz Coffee页面并不消失。而是出现一个全新的窗口，在这个窗口中打开Caffeine Buzz页面，如下所示：

这是Starbuzz Coffee主页面。

弹出Caffeine Buzz窗口时，它会盖在Starbuzz页面上面，不过Starbuzz页面还在。



CEO希望单击Caffeine Buzz链接时打开一个全新的窗口。

使用target打开新窗口

要在新窗口中打开一个页面，需要告诉浏览器你要打开的窗口名。如果没有告诉浏览器要使用哪个特定的窗口，浏览器就会在同一个窗口中打开这个页面。可以为元素增加一个target属性，告诉浏览器使用一个不同的窗口。target属性值会告诉浏览器页面的“目标窗口”。如果使用“_blank”作为目标，浏览器就总是打开一个新窗口显示页面。下面来仔细看一下：

```
<a target="_blank" href="http://wickedlysmart.com/buzz" title="Read all about caffeine on the Buzz">Caffeine Buzz</a>
```

target属性告诉浏览器在哪里打开href属性中链接指示的web页面。如果没有target，浏览器就会在同一个窗口中打开这个链接。如果目标为“_blank”，浏览器就会在一个新窗口中打开链接。



Exercise

打开你的Starbuzz “index.html”文件。为链接到Caffeine Buzz页面的标记增加target属性。现在来试一试，打开一个新窗口了吗？



你能想到使用target属性在一个新窗口打开页面有什么优点和缺点吗？

there are no Dumb Questions

问：我得到一个新的标签页，而不是新窗口。是不是哪里做错了？

答：不，你没做错。大多数浏览器现在都有一个默认设置，会在一个标签页中打开新窗口，而不是一个全新的浏览器窗口，因为用户好像更喜欢这样。不过新的标签页和新窗口实际上是一样的，只是标签页可以共享原窗口的窗口边框。如果想强制打开一个全新的窗口，则大多数浏览器上可以通过首选项设置来实现。

问：如果多个元素都有target呢？如果已经打开了一个“_blank”新窗口，那么会在已经打开的这个窗口中打开页面？还是会在另一个新的“_blank”窗口中打开？

答：如果所有元素中的target都指定为“_blank”，那么每个链接都会在一个新的空窗口中打开。不过，这个问题问得很好，因为由此引出一个重点：不一定要把target指定为“_blank”。如果指定另一个名字，如“coffee”，那么有相同目标名“coffee”的所有链接都会在同一窗口中打开。这是因为，为target指定一个特定的名字时，如“coffee”，实际上就是在对显示链接页面的新窗口命名。“_blank”则是一种特殊情况，告诉浏览器总是使用一个新窗口。



target属性闪亮登场

本周访谈：
使用target很不好吗？

Head First: 你好，Target！很高兴你能接受我们的采访。

Target: 我也很高兴。你们还对我这么感兴趣，真让我感动。

Head First: 怎么这么说呢？

Target: 嗯，说实话，我没有以前那么流行了。

Head First: 为什么你会这么想？

Target: 我觉得，这是因为用户现在都想自己来控制什么时候打开窗口。它们不希望总是意外地弹出新窗口。

Head First: 噢，这确实很烦人，我们听到过很多人曾抱怨，因为屏幕上有太多窗口，他们简直没有办法找到原来的页面。

Target: 不过，去掉这些窗口并不难……只需要单击那个小关闭按钮。这有什么难的！

Head First: 没错，不过用户不知道已经打开一个新窗口，他们可能就会糊涂。有时新窗口会完全覆盖原来的窗口，这就很难说清楚到底出了什么问题。

Target: 嗯，浏览器在这方面越来越棒了。

Head First: 哦，是吗？

Target: 浏览器通常会在一个新的标签页中打开新页面，不过是在同一个浏览器窗口中，而不是在一个全新的窗口中打开。

Head First: 啊，这确实会有帮助，因为看到打开一个新标签页就不那么容易让人糊涂了，用户

希望什么时候访问都可以。与打开一个新窗口不同，这样不会让人那么困惑。

Head First: 不过，这对使用屏幕阅读器的人有帮助吗？

Target: 你是指有视力障碍的人使用的浏览器吧？

Head First: 没错。打开一个新窗口时，有些屏幕阅读器会发出一个声音，不过另外一些会完全忽略新窗口，也可能直接跳到新窗口。不管怎样，对于无法亲眼看到的人来说，都会很迷惑。我不知道他们如何处理标签页。

Target: [唉]是啊，我们还没办法提供好的工具来满足每一个人的需要，特别是有视力障碍的人。也就是说，我们需要能够把用户带到我们网站以外的其他页面，很多网站的做法都是打开另一个窗口（或者如果浏览器支持，可以打开另一个标签页）。

Head First: 对，我们需要你，不过我们还需要做得更好，不让用户困惑。

Target: 我希望Web标准和浏览器小组能让情况好起来。

Head First: 我想现在我们只需要记住在适当的情况下用你，不过还要考虑到那些可能有视力障碍的人，所以不能过度地用你。

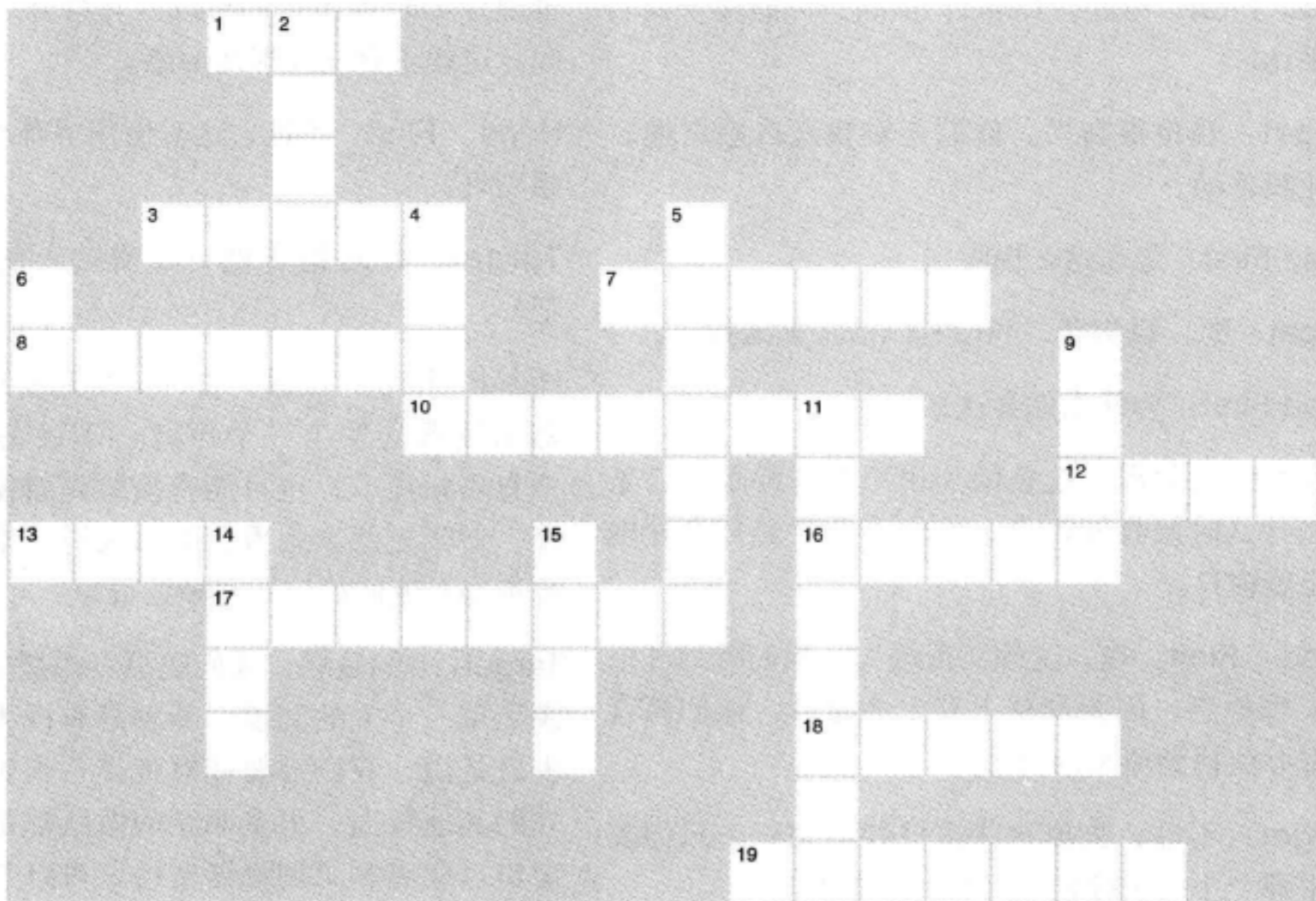
Target: 说得对。你让我松了一口气，非常感谢你帮我把这句话说出来！

Head First: 随时愿意效劳，Target！



HTML填字游戏

来锻炼锻炼你的左脑。



横向

1. 一个资源的Web地址。
3. 一个Mac FTP应用。
7. Web上的唯一名。
8. 请求一个目录时得到的文件。
10. 你从Web镇寄回的东西。
12. 网站的顶级目录。
13. 这一章之前一直使用的协议。
16. 人们可以扫描这些而不是通读文本。
17. 从根开始的路径。
18. 控制域名。
19. 请求一个目录时得到的文件。

纵向

2. 网站的顶级目录。
4. 请求/响应协议。
5. 保证链接标签_____。
6. 使用这个属性使一个元素成为目标。
9. Earl出售这些。
11. 链接到相同服务器上的页面时要使用这种链接。
14. 拼读URL的错误方式。
15. 提供大量信息的咖啡因网站。



BULLET POINTS

- 要把网站发布到Web上，通常最好的办法就是找一家托管公司来托管你的Web页面。
- 域名是一个唯一的名字，如amazon.com或starbuzzcoffee.com，用来唯一标识网站。
- 托管公司可能会为你的域创建一个或多个Web服务器。服务器通常命名为“www”。
- 文件传输协议（File Transfer Protocol, FTP）是向服务器传输web页面和内容的常用方法。
- FTP应用（如Mac上的Fetch for Mac或Windows上的WS_FTP）提供了一个图形用户界面，使FTP的使用更为容易。
- URL是统一资源定位符或Web地址，可以用来标识Web上的任何资源。
- 典型的URL由一个协议、一个网站名和资源的一个绝对地址组成。
- HTTP是一个请求和响应协议，用来在Web服务器和浏览器之间传送Web页面。
- 浏览器使用file协议从你的计算机读取页面。
- 绝对路径是从根文件夹到一个文件的路径。
- “index.html”和“default.htm”都是默认页面。如果指定一个目录而没有指定文件名，则Web服务器会查找一个默认页面返回到浏览器。
- <a>元素的href属性中可以使用相对路径或URL来链接其他Web页面。对于你的网站中的其他页面，最好使用相对路径，对外部链接才使用URL。
- 可以用id属性在页面中创建一个目标。使用#后面加一个目标id，可以链接到页面中的那个位置。
- 为了便于访问，可以在<a>元素中使用title属性提供链接的一个描述。
- 使用target属性在另一个浏览器窗口中打开链接。不要忘了，对于使用各种不同设备和浏览器的用户，target属性可能会有问题。

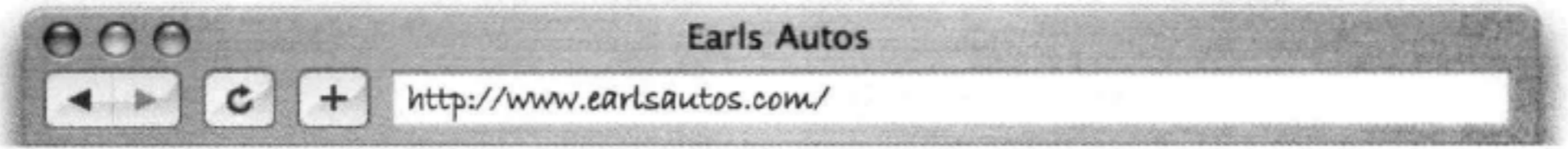
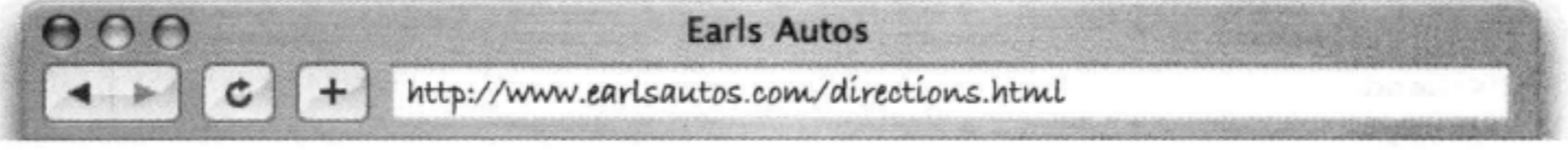
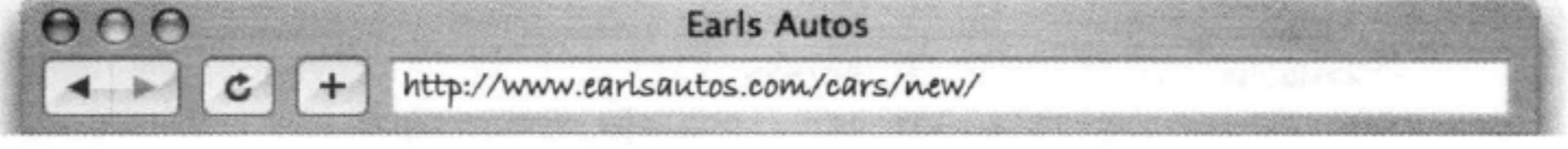
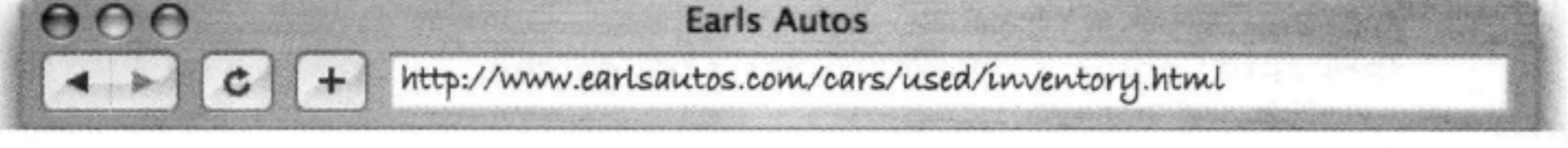
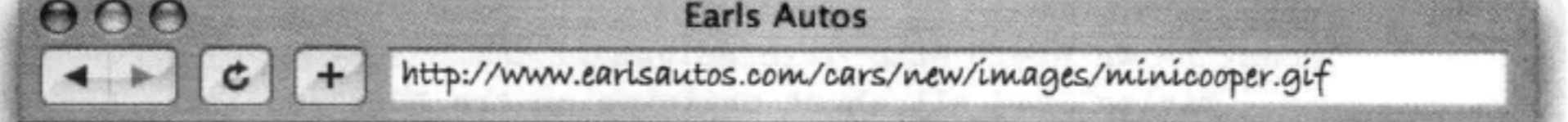
等一下！先别走，还需要在Web页面上加上我们的logo！
喂！噢，我想他们已经去第5章了……



Earl需要你帮他确定URL



答案

- A** 
- B** 
- C** 
- D** 
- E** 



Exercise Solution

为“mission.html”链接增加一个标题，设置为“Read more about Starbuzz Coffee's important mission”。注意这个宗旨页面链接的标签还不够简洁。可以把这个链接的标签简写为“our Mission”。下面是我们的答案，你对这些修改测试过吗？

```

<html>
  <head>
    <title>Starbuzz Coffee</title>
    <style type="text/css">
      body {
        background-color: #d2b48c;
        margin-left: 20%;
        margin-right: 20%;
        border: 1px dotted gray;
        padding: 10px 10px 10px 10px;
        font-family: sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Starbuzz Coffee Beverages</h1>
    <h2>House Blend, $1.49</h2>
    <p>A smooth, mild blend of coffees from Mexico,
      Bolivia and Guatemala.</p>

    <h2>Mocha Cafe Latte, $2.35</h2>
    <p>Espresso, steamed milk and chocolate syrup.</p>

    <h2>Cappuccino, $1.89</h2>
    <p>A mixture of espresso, steamed milk and foam.</p>

    <h2>Chai Tea, $1.85</h2>
    <p>A spicy drink made with black tea, spices,
      milk and honey.
    </p>
    <p>
      Read about <a href="mission.html"
      title="Read more about Starbuzz Coffee's important mission">our Mission</a>.
    <br>
    Read the <a href="http://wickedlysmart.com/buzz"
      title="Read all about caffeine on the Buzz">Caffeine Buzz</a>.
    </p>
  </body>
</html>

```

为宗旨页面
链接增加一
个title属性。

把“Read about”移出<a>元素。

5 为你的页面增加图像

认识媒体



笑一笑，说“茄子”。说真的，应该是笑一笑，说“gif”、“jpg”或“png”。你为Web“冲印照片”时，就会用到它们。这一章中，我们将学习如何为你的页面添加第一个媒体类型：图像。你是不是拍摄了一些数码照片想要发到网上？没问题。是不是需要在页面上加一个logo？小事一桩。不过在深入学习这些内容之前，是不是需要正式介绍一下元素？真是抱歉，我们不是有意疏忽，只是一直没有找到“合适的机会”。作为补偿，下面用一整章专门讨论。学完这一章，你就会全面地了解使用元素及其属性的所有细节了。你还会亲眼看到这个小元素是怎样让浏览器做大量额外的工作来获取和显示图像的。

浏览器如何处理图像

浏览器对元素的处理与其他元素稍有不同。以<h1>或<p>元素为例，浏览器在页面上看到这些标记时，只需要把它们显示出来就可以了，很简单。不过，浏览器看到一个元素时，会做不同的处理：浏览器在页面中显示图像之前，必须先获取这个图像。

要了解这是如何做到的，最好的办法就是通过一个例子来说明。下面再来简单地回顾一下Head First休闲室的清凉饮料页面，其中有4个元素：

```

<html>
  <head>
    <title>Head First Lounge Elixirs</title>
  </head>
  <body>
    <h1>Our Elixirs</h1>
    <h2>Green Tea Cooler</h2>
    <p>
      
      Chock full of vitamins and minerals, this elixir combines
      the healthful benefits of green tea with a twist of chamomile
      blossoms and ginger root.
    </p>
    <h2>Raspberry Ice Concentration</h2>
    <p>
      
      Combining raspberry juice with lemon grass, citrus peel and
      rosehips, this icy drink will make your mind feel clear and
      crisp.
    </p>
    <h2>Blueberry Bliss Elixir</h2>
    <p>
      
      Blueberries and cherry essence mixed into a base of elderflower
      herb tea will put you in a relaxed state of bliss in no time.
    </p>
    <h2>Cranberry Antioxidant Blast</h2>
    <p>
      
      Wake up to the flavors of cranberry and hibiscus in
      this vitamin C rich elixir.
    </p>
    <p>
      <a href="../lounge.html">Back to the Lounge</a>
    </p>
  </body>
</html>

```



这个HTML中有4个图像。

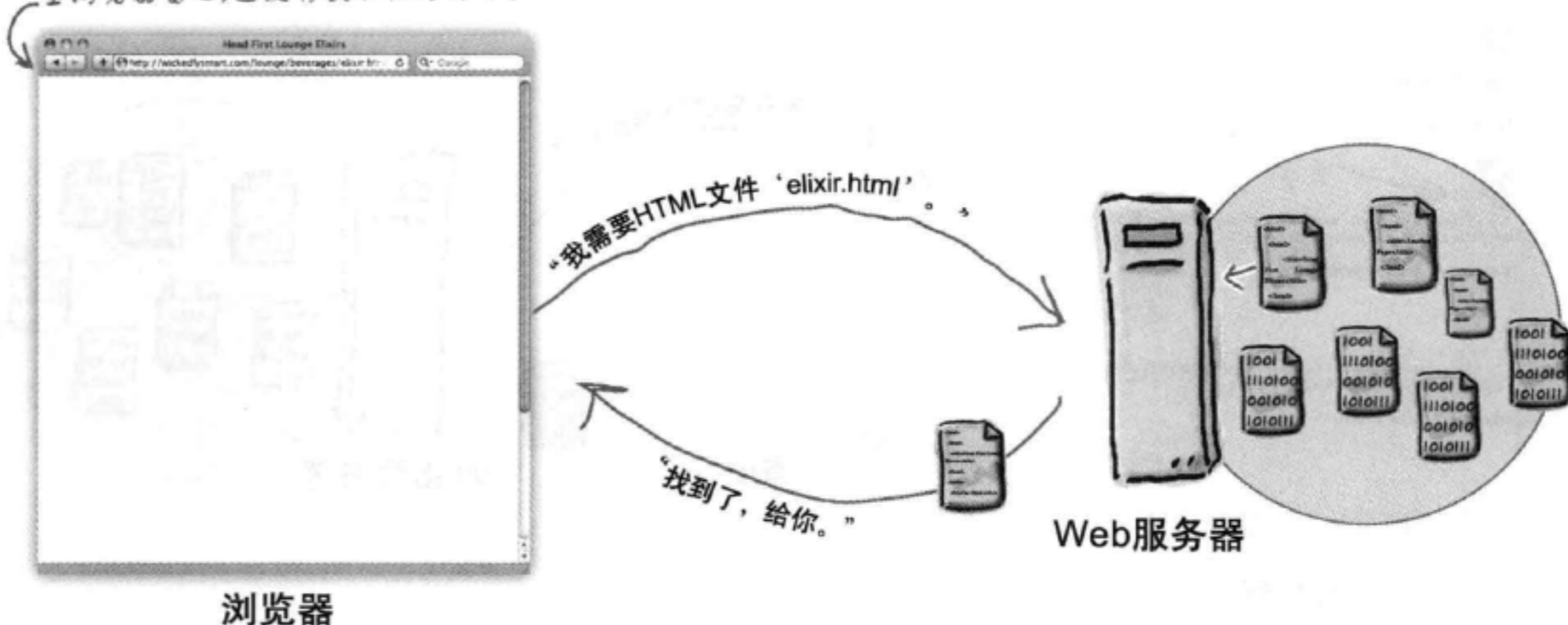
现在我们来幕后看一看，一步一步了解浏览器收到http://wickedlysmart.com/lounge/请求时如何获取和显示这个页面：

在幕后



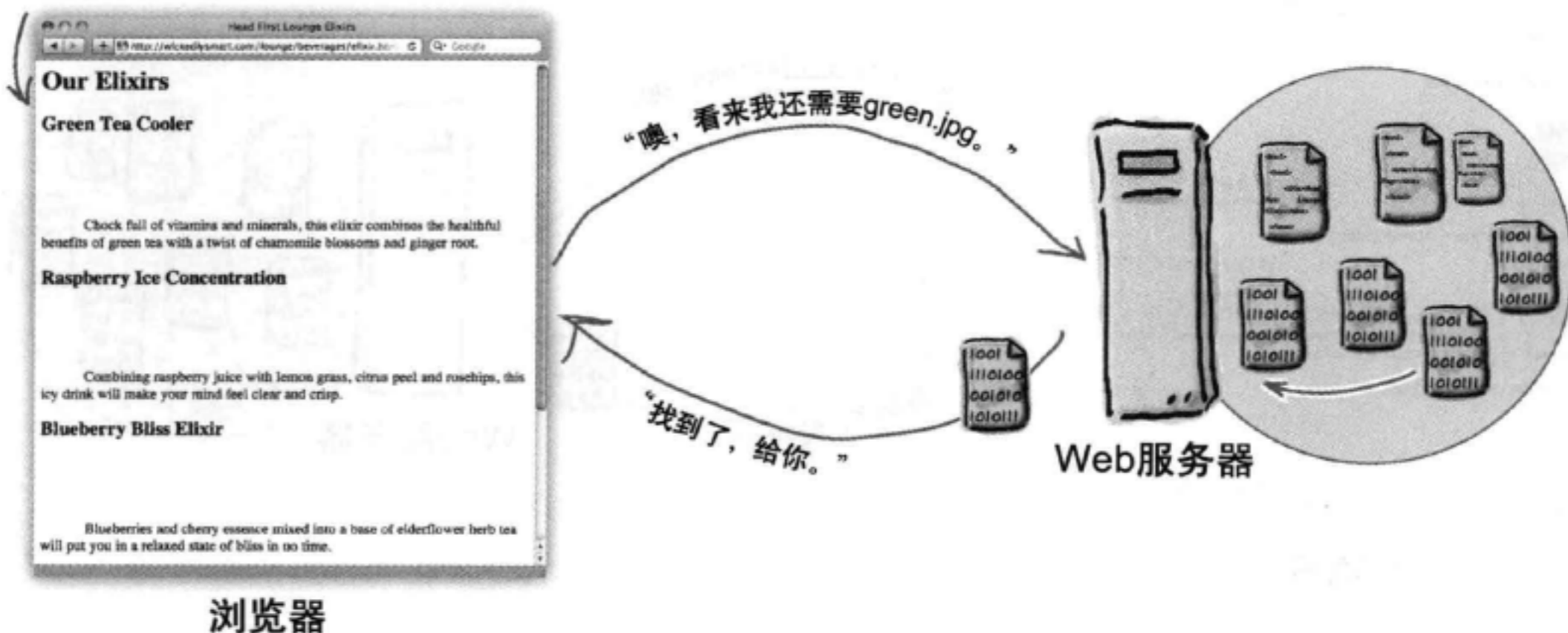
- ① 首先，浏览器从服务器获取文件“elixir.html”。

空浏览器窗口，还没有获取任何内容。

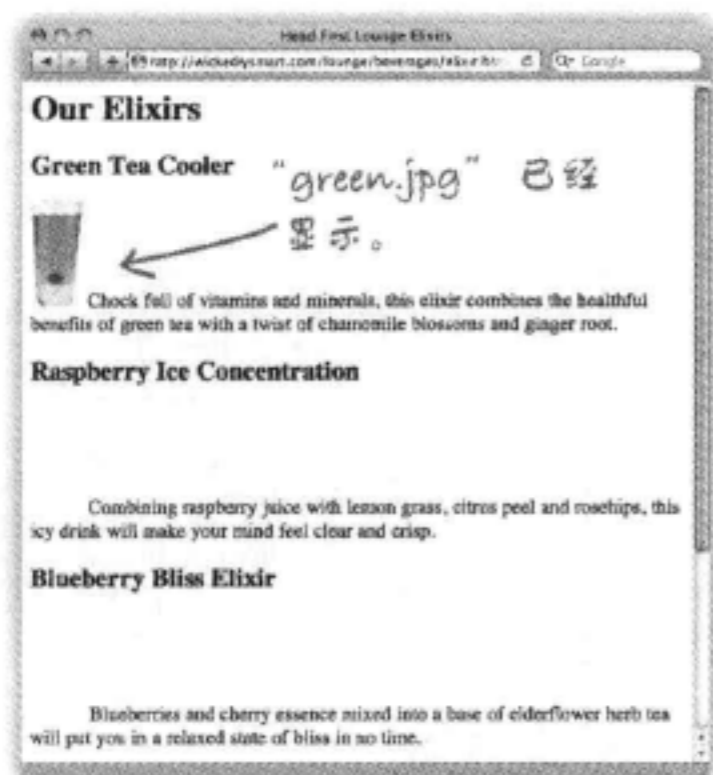


- ② 接下来，浏览器读取“elixir.html”文件，显示这个文件，发现其中有4个图像需要获取。所以它需要从Web服务器逐个得到这些图像，先从“green.jpg”开始。

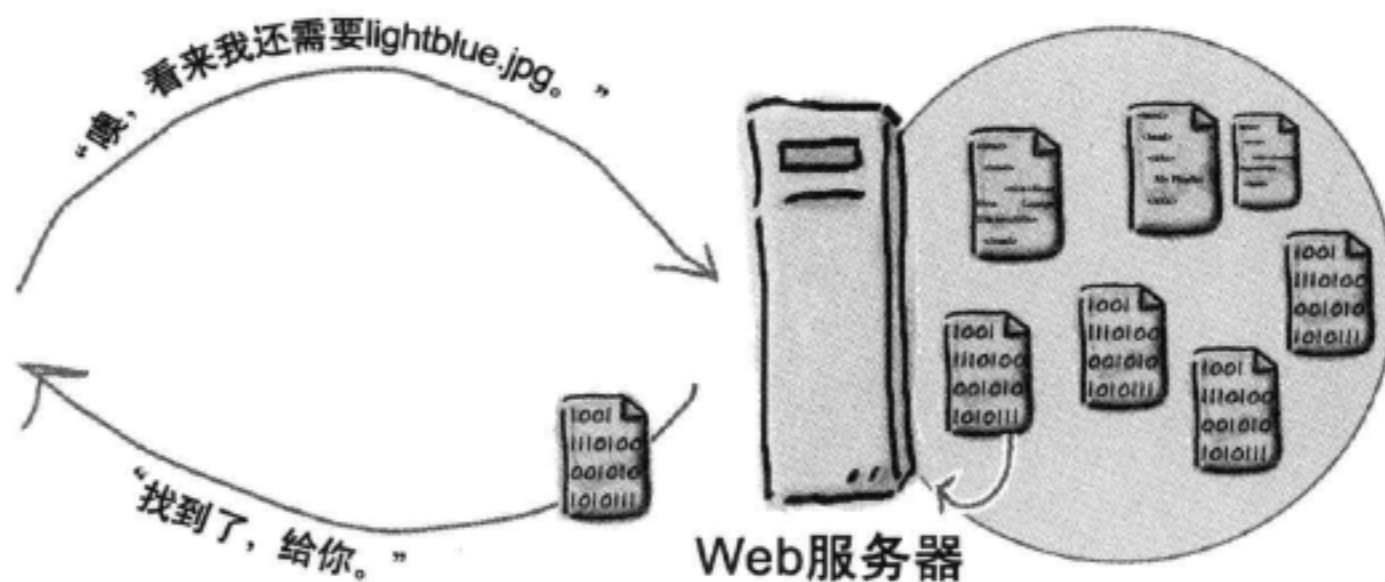
获取了HTML页面，但是浏览器还需要得到图像。



- ③ 获取到“green.jpg”之后，浏览器显示这个图像，然后转向下一个图像：“lightblue.jpg”。



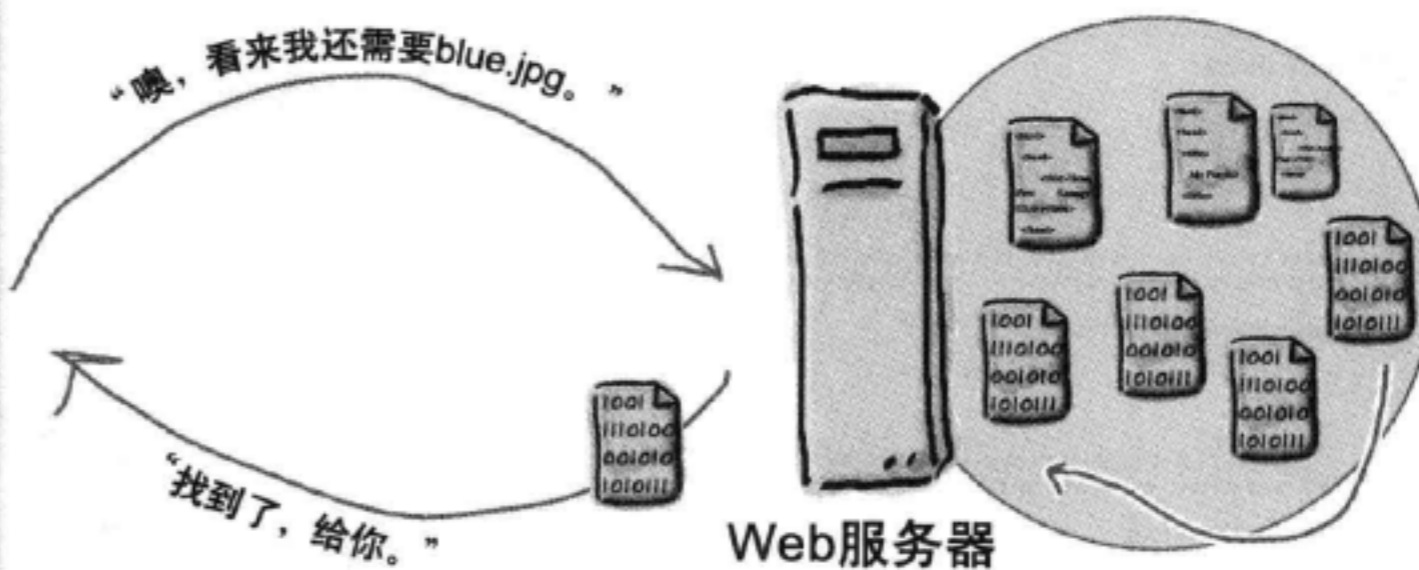
浏览器



- ④ 现在浏览器已经获取了“lightblue.jpg”，所以它会显示这个图像，然后转向下一个图像：“blue.jpg”。对页面上的每一个图像都会重复这个过程。



浏览器



图像是如何工作的

图像就是image，对吗？嗯，实际上，世界上图像格式实在太多了，这些图像格式都各有优缺点。幸运的是，Web上最常用的只是其中3种格式：JPEG、PNG和GIF。唯一比较困难的是要确定在什么情况下使用哪一种格式。

那么，JPEG、PNG和GIF有什么不同呢？



照片和复杂图像使用 JPEG

最适合连续色调图像，如照片。

可以表示包含多达1600万种不同颜色的图像。

这是一种“有损”格式，因为缩小文件大小时会丢掉图像的一些信息。

不支持透明度。

文件比较小，以便Web页面更高效地显示。

不支持动画。



单色图像、logo和几何图形使用 PNG或GIF

PNG最适合单色图像和线条构成的图像（如logo、剪贴画和图像中的小文本）。

PNG可以表示包含上百万种不同颜色的图像。PNG有3种：PNG-8、PNG-24和PNG-32，取决于你需要表示多少种颜色。

PNG会压缩文件来缩小文件大小，不过不会丢掉信息。所以这是一种“无损”格式。

允许将颜色设置为“透明”，使图像下面的东西可以显示出来。

与相应的JPEG文件相比，PNG文件更大一些，不过取决于使用的颜色数，可能比相应的GIF文件小，也可能更大。

类似于PNG，GIF最适合单色图像和线条构成的图像（如logo、剪贴画和图像中的小文本）。

GIF可以表示最多256种不同颜色的图像。

GIF也是一种“无损”格式。

GIF也支持透明度，不过只允许一种颜色设置为“透明”。

GIF文件往往比相应的JPEG文件大。

支持动画。



真正的图像格式请起立!

本周访谈: 混乱的图像格式

Head First: 嘿, 大家好。我想这可能是我们头一次同时介绍三位嘉宾!

JPEG: 你们好, GIF和PNG。

GIF: 我不知道为什么要和另外这两个家伙一起接受采访。所有人都知道GIF是Web最早的图像格式。

JPEG: 哈! 除非你能很好地表示复杂图像, 比如说照片, 那时人们可能才会重新对你另眼相看吧, 不过我实在不相信你用区区256种颜色能够做到。

Head First: PNG, 快来帮忙啊? 你太安静了, 到现在还没说一句话呢……

PNG: 嗯, 第一名一般就不用说什么。我可以像JPEG那样表示复杂图像, 还可以像GIF那样是无损的。所有优点都集于我一身。

Head First: 无损的?

PNG: 对, 用一种无损格式存储图像时, 你不会丢失图像中的任何信息或细节。

GIF: 我也是! 我也是无损的, 你知道的。

Head First: 嗯, 为什么会有人需要无损格式呢?

JPEG: 凡事总有取舍。有时你可能只想要一个很小的文件, 可以快速下载, 而且质量很不错。我们并不总需要最完美的质量。JPEG图像已经让人很满意了。

PNG: 当然, 当然, 不过你看过那些线条、logo、小文本和单色图像没有? 如果用JPEG格式, 则它们看起来可不怎么样。

Head First: 等一下, JPEG提出一个有意思的问题。我要问一下, GIF和PNG, 你们的文件大吗?

PNG: 我承认我的文件有时可能比较大, 不过我提供了3种格式: PNG-8、PNG-24和PNG-32, 你可以选择适当的文件大小。

GIF: 对我来说, 听上去很复杂, 这得让你的用户记更多东西。

PNG: 嗯, GIF, 如果所有图像都只使用256种颜色, 那该多好! 不过这样不行。

GIF: 嘿, 对于线图、图形之类的图像, 很容易用8位颜色表示, 而且我在这方面确实很棒。

JPEG: 哈, 你上一次见到用GIF格式存储的照片是什么时候? 人们已经发现的你短处了, GIF。

GIF: 我提到过我可以透明了吗? 你可以取走我的一部分, 我下面的东西就能透出来。

PNG: 在这方面, 你可比不上我, GIF。我可以把任意种颜色设置为透明, 而你仅限于一种颜色。

GIF: 一种还是多种颜色, 谁在意呢? 一种就够了。

PNG: 如果你希望图像里有无锯齿透明区, 则只设置一种颜色透明就不行!

GIF: 嗯?

PNG: 是的, 你要知道, 因为我允许将多种颜色设置为透明, 所以透明区的边缘是平滑的。

Head First: 听上去是一个很不错的特性。你能做到吗, JPEG?

JPEG: 不行, 不过我不担心这一点, 没有多少照片需要这么做。这是对logo来讲的。

PNG: 嗨, 我可发现Web上到处都在用我的透明性。

Head First: 嗯, 下次再做三人采访之前我可得三思了, 不过根据你们所说, GIF和PNG, 你们很擅长表示logo和文本图像; JPEG, 你擅长照片; 另外, PNG, 如果我们需要透明和很多种颜色时, 则你会很方便。好了, 再见!

PNG、JPEG、GIF: 等一下, 别急着走!

* * *
 * * *
 * * *
 哪种图像格式 * * *

祝贺你：你当选为今天的“超级图像格式选择能手”。对下面的每个图像，请选择在Web上显示的最佳格式。

JPEG 或 PNG 或 GIF











噢，我本来不想这么冒失，不过这一章不是要讲图像吗？现在已经到这一章的第8页了，你还没介绍我呢！JPEG、PNG、GIF没完没了……能介绍我了吗？

现在就来正式介绍：元素。

这个介绍真是拖了很久。可以看到，与处理其他HTML标记相比，处理图像要做更多工作。不管怎样，现在先不说这些了……来认识认识元素吧。

首先来仔细研究这个元素（不过你现在对如何工作可能已经了解得不少了）：



仅此而已吗？还不完全。你可能还希望了解几个属性，当然你还希望知道如何使用元素引用Web上的图像（这些图像可能不在你自己的网站上）。不过，说实在的，你已经掌握了使用元素的基本知识。

下面逐步分析使用元素的几个重点，然后学以致用。

: 不再只是相对链接

src属性不只是用于相对链接，还可以在src属性中放入URL。图像与HTML页面一同存储在Web服务器上，所以Web上的每个图像都有自己的URL，就像Web页面一样。

如果要指向另一个不同网站上的图像，通常要使用这个图像的URL（要记住，对于相同网站上的链接和图像，最好使用相对路径）。

要使用URL链接一个图像，如下所示：

```

```


要用URL包含一个图像，只需要把图像的整个URL都放在src属性中。

URL是图像的路径，所以最后的文件名总是一个图像文件名。与默认Web页面不同，这里没有默认图像之说。

Sharpen your pencil

这里有一个“练练笔”练习，它确实与铅笔有关（哦，还有图像）。这个练习提到这样一个问题：给你一根普通的新铅笔，如果用它画一条连续的线，把整个铅笔用完，这条线会有多长？

这和图像有什么关系？要找出答案，先写一些HTML。这个问题的答案就包含在URL <http://wickedlysmart.com/hfhtmlcss/trivia/pencil.png>图像中。你的任务是把图像增加到这个HTML，并找出答案：

```
<html>
  <head>
    <title>Sharpen your pencil trivia</title>
  </head>
  <body>
    <p>How long a line can you draw with the typical pencil?</p>
    <p>
      
    </p>
  </body>
</html>
```

把image元素放在这里。

there are no Dumb Questions

问:这么说, 元素很简单, 就是提供了一个途径, 可以通过它指定需要在页面上显示的图像的位置, 是吗?

答:是的, 总结起来就是这样。我们还会讨论这个元素中可以增加的一些属性。以后你还会看到, 可以采用很多方法利用CSS改变一个图像的视觉样式。

不过, 关于图像本身还有很多需要了解的知识。这些不同的图像格式有什么作用? 为什么要使用某一种格式而不是其他格式? 这些图像文件会有多大? 如何准备在Web页面中使用的图像?

问:我们已经了解到, void元素就是没有内容和结束标记的元素。我们还知道元素是void元素。不过它不是有内容(图像)吗?

答:嗯, 更确切地讲, void元素是指HTML页面中在开始标记和结束标记之间没有任何内容的元素。没错, 图像也是内容, 不过元素只是指向图像。图像并不是HTML页面本身的一部分。实际上, 浏览器显示页面时, 图像会取代元素。另外要记住, HTML页面是纯文本, 所以图像绝对无法直接作为页面的一部分。它是单独存在的。

问:再回到前面的例子, 加载一个有图像的页面……我加载一个Web页面时, 并没有看到图像一个接一个地加载, 为什么?

答:浏览器通常会同时获取图像。也就是说, 浏览器会同时请求多个图像。按照现有的计算机和网络速度, 这一切发生得太快, 所以你通常会看到页面与图像一同显示出来。

问:如果我看到Web页面上的一个图像, 那么如何确定它的URL, 从而能链接到它呢?

答:大多数浏览器都允许用鼠标右键单击图像, 这会弹出一个上下文菜单, 提供一些选项。在这些选项中, 你会看到“Copy Image Address”(复制图像地址)或“Copy Image Link”(复制图像链接), 这两个选项会把图像的URL放在你的剪贴板里。要找到URL, 另一种方法是用鼠标右键单击并选择“Open Image in New Window”(在新窗口打开图像), 这会在一个浏览器窗口中打开图像。你可以从浏览器的地址栏得到这个图像的URL。最后一种做法是使用浏览器的“查看源代码”(View Source)菜单选项, 检查HTML。不过要记住, 你找到的可能是图像的一个相对链接, 所以需要使用网站域名和图像路径“重新构造”它的URL。

问:为什么JPEG照片要比GIF或PNG照片好, 或者为什么GIF或PNG logo要优于JPEG logo?

答:“优于”通常是结合图像质量和文件大小而言的。JPEG照片一般比相同质量的PNG或GIF照片要小, 而PNG或GIF logo通常看起来比JPEG格式效果更好, 而且文件也可能更小。

问:在GIF和PNG之间如何选择呢? 看起来它们很相似。

答:PNG是最新的图像格式, 也很有意思, 因为它既能支持照片也能很好地支持logo。而且它还提供了比GIF更高级的透明性。现在所有主流浏览器都支持PNG, 而几年前情况可大不一样。

要在GIF和PNG中做出选择, 有几个方面需要考虑。首先, PNG的压缩要稍稍优于GIF, 所以对于颜色数相同的图像(也就是说, 最多256种颜色), PNG文件可能更小一些。不过, 如果需要更多颜色, 而GIF无法提供, 而且也不能选择JPEG(例如, 你需要透明性), 那么除了PNG就别无选择了。不过, 如果你需要动画, 那么还得使用GIF, 因为GIF是唯一支持动画并得到广泛支持的格式。

一定要提供候选格式

Web上有一点可以肯定，你永远也无法准确地知道人们会用什么浏览器和设备来查看你的页面。访问者很有可能使用移动设备，有视力障碍的人可能还要用屏幕阅读器，另外有些浏览器可能在网速极慢的环境中运行（可能只能获取一个网站的文本，而无法得到图像），还有可能使用手机、能上网的T恤……谁知道还会有什么！

不过尽管存在这些不确定性，你完全可以做好准备，沉着应对。尽管浏览器可能无法显示你的页面上的图像，但还有一种候选做法。你可以使用元素的alt属性为访问者提供一些指示，告诉他们图像里有什么信息。可以这样做：

```

```

alt属性需要指定描述这个图像的一些文本。

如果图像未能显示，就会使用这个文本来取代它。这有些像你通过电话为某个人读一个Web页面，读到图像时，你会读出alt文本来代替图像。



Exercise

在这个练习中，你会看到，如果无法显示图像，则你的浏览器将如何处理alt属性。基本原理是这样的：如果无法找到图像，则会取而代之显示alt属性。不过，并不是所有浏览器都会这样做，所以你的结果可能有所不同。你需要完成下面的工作。

- 1 得到上一个练习的HTML。
- 2 更新image元素，使它包含alt属性“The typical new pencil can draw a line 35 miles long.”
- 3 将图像名从“pencil.png”改为“broken.png”。这个图像并不真正存在，所以你会发现图像无法显示。
- 4 在浏览器中重新加载页面。
- 5 最后，下载几个其他的浏览器，再来试一试。是不是得到了不同的结果？

查看本章最后给出的结果……

例如，你可以试一试Firefox (<http://www.mozilla.org/>)或Opera (<http://www.opera.com/>)。

调整图像大小

你还需要知道元素的另一个属性。实际上，应该是一对属性：width和height。可以使用这些属性提前告诉浏览器你的页面中一个图像的大小。

可以这样使用width和height：

```

```

↑
width属性告诉浏览器在页面中显示图像的宽度。

↑
height属性告诉浏览器在页面中显示图像的高度。

宽度和高度都使用像素数指定。如果对像素还不熟悉，则这一章后面我们还会更详细地介绍。你可以为一个图像增加width属性和height属性。如果没有指定宽度和高度，则浏览器在页面中显示这个图像之前会自动确定图像的大小。

there are no Dumb Questions

问：既然浏览器可以确定图像大小，我又何必使用这些属性呢？

答：很多浏览器上，如果在HTML中提供了width和height，浏览器在显示图像之前就可以开始建立页面布局。如果没有指定，则浏览器在知道了图像大小之后，通常需要重新调整页面布局。要记住，浏览器是在下载了HTML文件并开始显示页面之后才下载图像。浏览器下载图像之前，除非你告诉它，否则它无法知道图像的大小。

还可以提供比图像实际尺寸更大或更小的宽度和高度值，这样浏览器就会缩放图像来满足你指定的大小。需要

按与原尺寸不同的大小显示一个现有图像时，很多人都会这么做。不过，以后你会看到，出于很多原因，最好不要使用width和height来达到这个目的。

问：我必须成对使用这些属性吗？能不能只指定width或height？

答：可以的，不过，既然你想告诉浏览器某一个尺寸，再提供第二个尺寸又何妨呢？工作量是相同的（而且只提供一个宽度或一个高度并没有多少好处，除非你想把图像缩放到某个特定的宽度或高度）。

问：我们说过很多次，希望用HTML来提供结构，而不要用于提供表现。这些属性看起来是表现属性，我理解的不对吗？

答：这取决于你要如何使用这些属性。如果你把图像宽度和高度设置为正确的尺寸，那么它们只是提供信息。不过，如果你使用width和height属性来调整浏览器中图像的大小，就是在用这些属性提供表现。在这种情况下，可能最好考虑使用CSS来得到同样的结果。

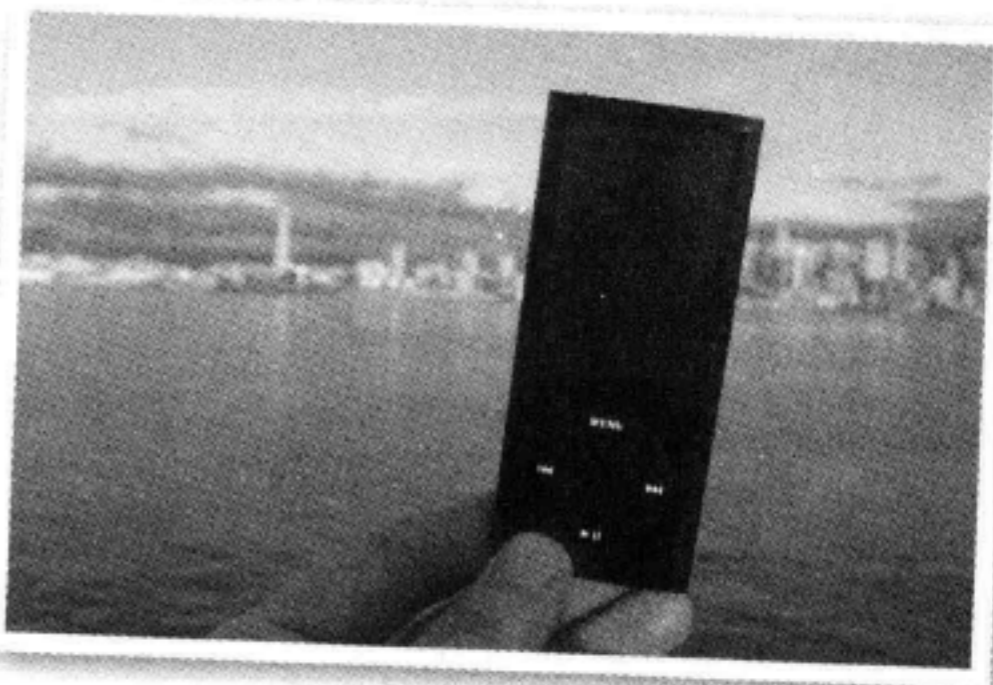
创建超级粉丝网站:myPod

有iPod的人都非常喜爱他们的iPod，不论到哪里都带着它。想象一下，现在要创建一个名为“myPod”的新网站，显示你的朋友带着他们的iPod在世界各地最喜欢的景点拍摄的照片。

首先需要些什么？只需要一些HTML知识、一些图像，还有对iPod的喜爱。

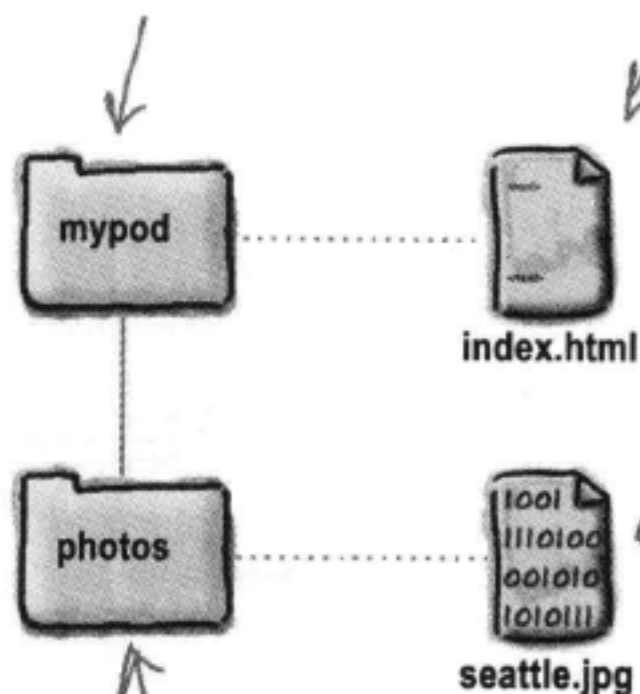
我们已经为这个网站写了一些HTML，不过还没有加入图像，所以要请你来。不过在你增加这些图像之前，先明确几点，在本书示例源码中查找“chapter5”文件夹。你会找到一个名为“mypod”的文件夹。打开“mypod”文件夹，你会看到下面的内容：

iPhone也行！



我的iPod在西雅图！你能看到西雅图的标志性建筑“太空针”。不过你看不到628家咖啡店。

在chapter5文件夹中能
找到这个文件夹。



我们已经为myPod网站写了一些HTML。可以在“index.html”文件中找到。

这是第一个iPod图像：一张在西雅图拍摄的照片。

我们要用photos文件夹存放这个网站的图像。

注意：在“mypod”中你还能找到另外几个文件夹，不过现在先不考虑它们。

检查myPod的“index.html”文件

打开文件“index.html”，你会看到myPod已经有内容了。
下面是现在的HTML：

```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>

    <h1>Welcome to myPod</h1>
    <p>
      Welcome to the place to show off your iPod, wherever you might be.
      Wanna join the fun? All you need is any iPod from the early classic
      iPod to the latest iPod Nano, the smallest iPod Shuffle to the largest
      iPod Video, and a digital camera. Just take a snapshot of your iPod in
      your favorite location and we'll be glad to post it on myPod. So, what
      are you waiting for?
    </p>

    <h2>Seattle, Washington</h2>
    <p>
      Me and my iPod in Seattle! You can see the
      Space Needle. You can't see the 628 coffee shops.
    </p>

  </body>
</html>
```

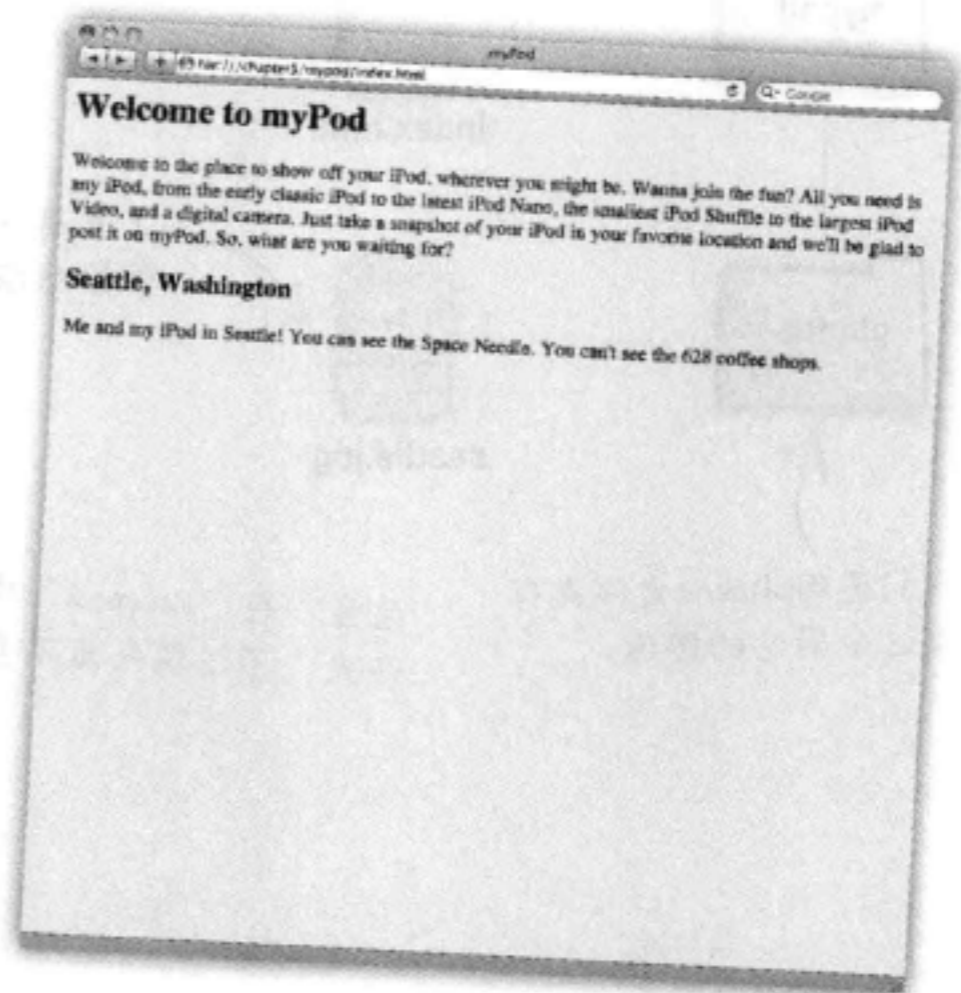


这里加入了一些成品CSS。现在直接输入就可以了。这些CSS的作用就是让页面有一个淡绿色背景。后面几章我们还会仔细讨论CSS，我保证！



这个HTML看起来很熟悉，因为这里使用的都是基本构建模块：<h1>、<h2>和<p>。

这是浏览器中显示的结果。还不错，不过我们还需要图像。



Sharpen your pencil

可以看到，myPod网站的大部分HTML都已经写好，准备运行了。现在要做的就是为你希望加入的每张照片增加一个

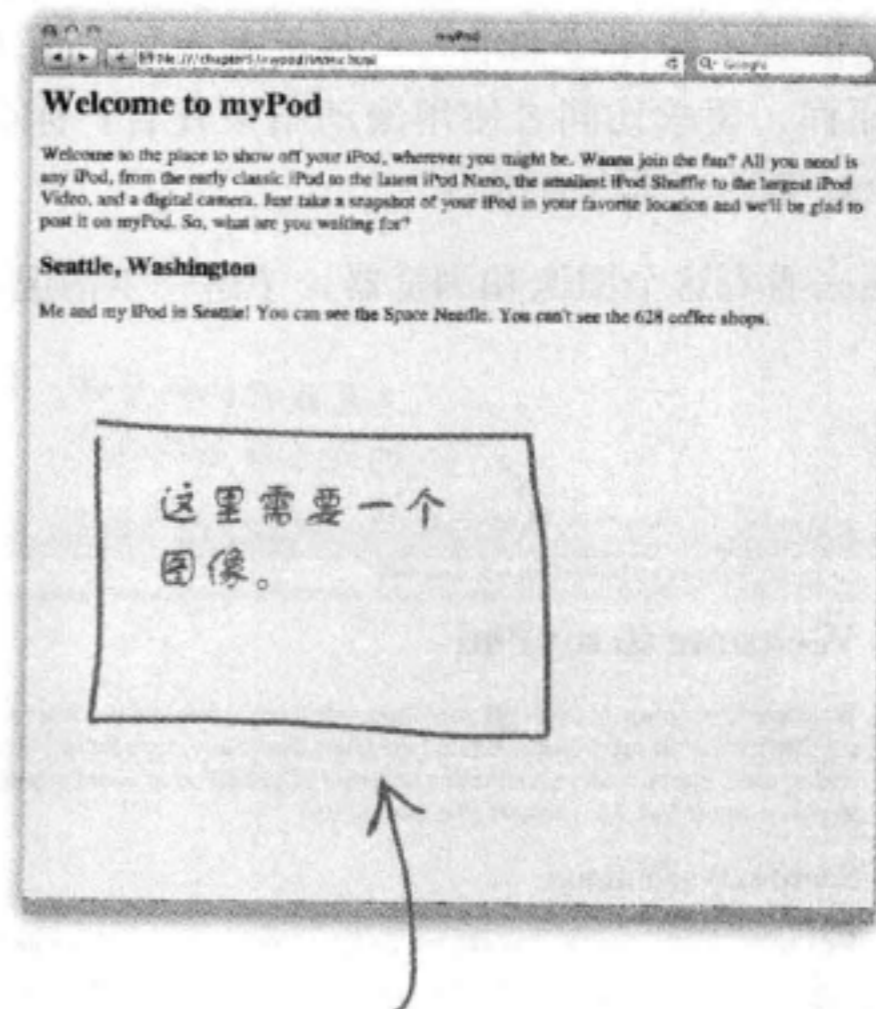
```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>

    <h1>Welcome to myPod</h1>
    <p>
      Welcome to the place to show off your iPod, wherever you might be.
      Wanna join the fun? All you need is any iPod from, the early classic
      iPod to the latest iPod Nano, the smallest iPod Shuffle to the largest
      iPod Video, and a digital camera. Just take a snapshot of your iPod in
      your favorite location and we'll be glad to post it on myPod. So, what
      are you waiting for?
    </p>

    <h2>Seattle, Washington</h2>
    <p>
      Me and my iPod in Seattle! You can see the
      Space Needle. You can't see the 628 coffee shops.
    </p>

    <p>
    </p>

    </body>
  </html>
```



需要在这里放入第一张照片。

你的元素要放在这里。

哇！图像太大了

嗯，图像放好了，不过这是一个大图像。而且，如今数码相机拍摄的大多数图像都很大（甚至更大）。是不是应该让图像保持原样，要求访问者使用滚动条来查看？你会看到，这可不是一个好主意，原因有很多。

下面来看看这个图像和浏览器，了解一下情况有多糟糕……



如果图像在浏览器窗口中刚好能放下，说明你的浏览器可能打开了“auto image resize”（自动调整图像大小）选项。有关的更多内容稍后介绍……

这是我们的浏览器，典型的浏览器窗口大小。



这是要增加到“index.html”的“seattle.jpg”图像。

这是这个图像的完整大小，比浏览器窗口要大……大多了。

可以使用滚动条查看图像的其余部分，不过如果图像能在浏览器窗口中刚好放下不是更好吗？

浏览器窗口宽度大约为800像素。

图像宽度大约1200像素。

there are no Dumb Questions

问:让用户用滚动条查看图像有什么不对?

答:一般来讲,如果页面中有很大的图像,则这个页面会很难使用。访问者没有办法一次看到完整的图像,不仅如此,使用滚动条也很麻烦。另外大图像还需要服务器和浏览器之间传输更多数据,这会耗费很多时间,可能导致你的页面显示速度很慢,特别是对于那些通过拨号上网或使用其他慢速连接的用户。

问:为什么不能直接使用width属性和height属性来调整页面上图像的大小?

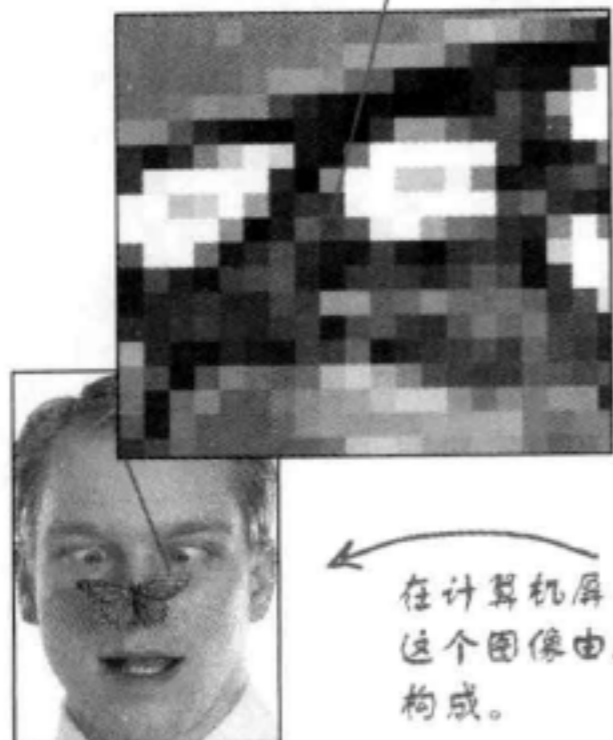
答:因为浏览器在缩放图像使之适应页面大小之前,仍然需要获取整个大图像。

问:你刚才说浏览器窗口宽度是800像素,这是什么意思?

答:你的计算机显示屏是由数百万个称为像素的点组成的。如果仔细看你的显示屏,就应该能看到它们:

这里有一些像素,
共同构成了蝴蝶右
翅膀的上半部分。

这是一个像素。



在计算机屏幕上显示时,
这个图像由数千个像素
构成。

尽管屏幕大小和分辨率可能有变化(有些人的显示器小,有些则很大),大多数人通常都会把浏览器宽度设置为800到1280像素之间。所以一般经验是将图像最大宽度设置为800像素(Web页面也一样,不过这个内容在后面一章再做介绍)。

问:像素数与屏幕上图像的大小有关系吗?

答:一般经验是每英寸96像素,不过如今的显示器分辨率都很高,所以还可以更高。我们习惯使用标准的72像素/英寸(ppi),不过为了应对现代显示器,又创造了一个CSS像素的概念。CSS像素是1英寸的1/96(96 ppi)。所以对于一个3"宽×3"高的图像,要使用96(像素)×3(英寸)=288×288像素。

问:嗯,那么我的图像应该设置为多大呢?

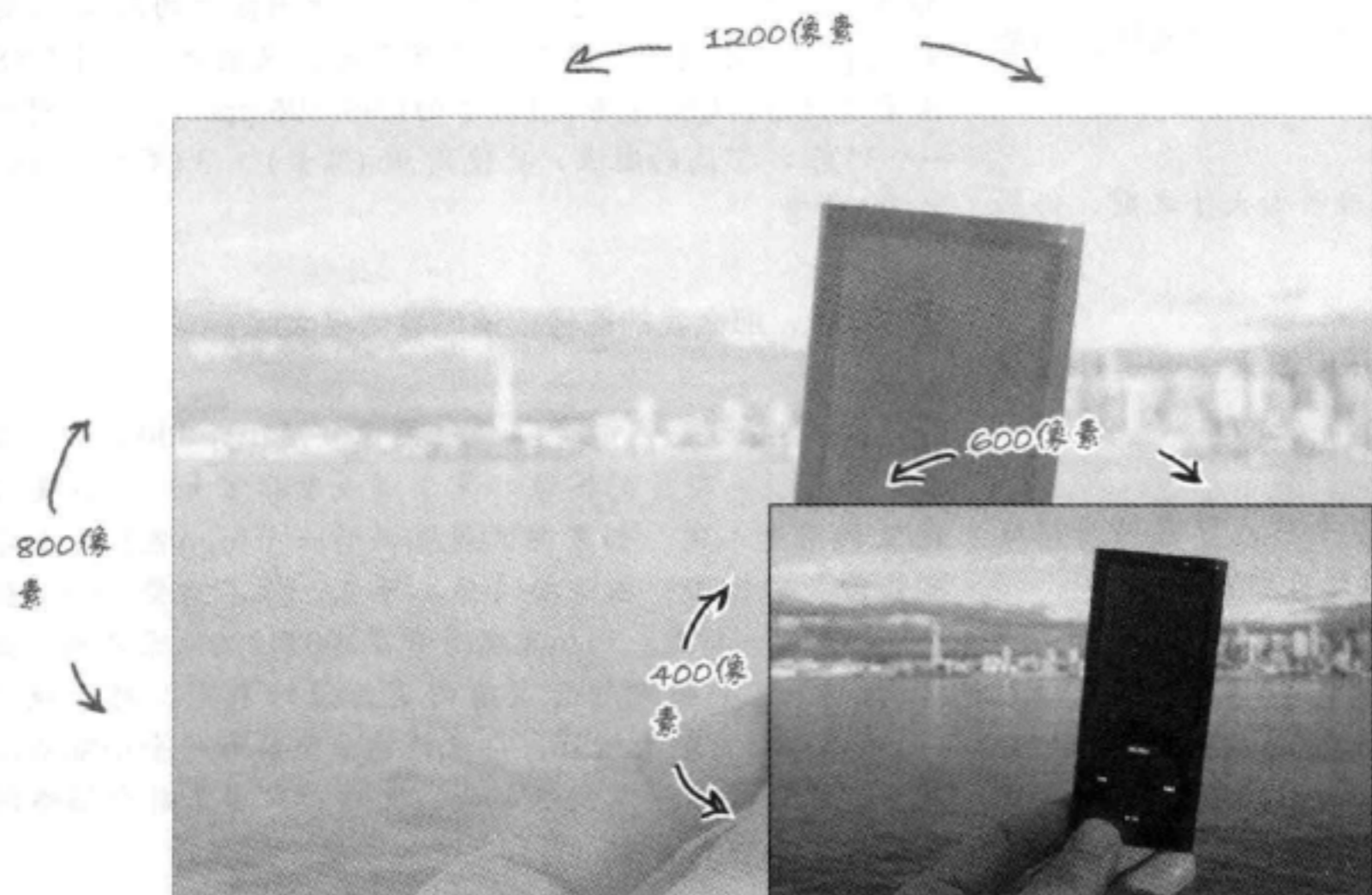
答:一般来讲,往往希望保证图像宽度小于800像素。当然,你可能希望你的图像非常小(或者非常大),这要由图像的用途而定。如果图像是页面的一个logo呢?那么你可能希望它很小,但是要清楚。毕竟,你不需要logo与整个Web页面一样宽。logo宽度通常在100到200像素之间。归根结底,这个问题的答案要由页面设计来定。对于照片(通常你希望它越大越好),也许你希望能有一个小缩略图页面,这个页面可以快速加载,允许用户单击各个缩略图来查看原来的大图像。稍后就要讨论这个问题。

问:我觉得我的浏览器会自动调整西雅图图像的大小,因为它在窗口里看起来刚好合适。为什么我的浏览器会这么做?

答:有些浏览器有一个特性,可以对不适合浏览器宽度的图像调整大小。不过很多浏览器不这么做,所以不要依赖这个特性。即使每个浏览器确实有这个特性,这样做也不好,这会在服务器和浏览器之间不必要地传输更多数据,导致你的页面加载很慢,可用性降低。另外要记住,越来越多的人在移动设备上查看Web页面,太大的图像会影响这些设备上的数据使用。

调整图像大小以适合浏览器


下面调整这个图像的大小，让它能更适合浏览器窗口。现在这个图像是1200像素宽、800像素高（稍后你就会了解这是如何确定的）。因为我们希望图像的宽度小于800像素，所以要确定一个适当的宽度，能适合我们的myPod Web页面。myPod网站的目的就是浏览在不同场合拍摄的iPod的照片，所以我们可能希望看到足够大的图像。如果把图像大小缩小一半为600像素宽、400像素高，则宽度上会基本占满浏览器，不过两边还会有一些空白。听起来还不错？下面就来调整图像的大小……



我们需要调整这个图像的大小，使它仍然很大，但是宽度要小于800像素。600像素宽似乎不错，这正好是当前大小的一半。

你要做的工作：

- ① 用一个照片编辑应用打开这个图像。
- ② 把图像大小缩小为一半（600像素×400像素）。
- ③ 将图像保存为“seattle_video_med.jpg”。



开始之前，我想问一下，我们要用哪个照片编辑应用来调整图像大小？我有Photoshop Elements，这个能用吗？

问得好，市场上有很多照片编辑应用（有一些还是免费的），它们都差不多。我们要用Adobe公司的Photoshop Elements来调整图像大小，因为这是最流行的照片编辑应用之一，在Windows和Macintosh上都可以使用。如果你有自己的照片编辑应用，则完全可以跟我们一起继续学习，在你自己的应用里完成各个编辑任务。

如果你没有照片编辑应用，则可以先看看你的操作系统里有没有包含一个这样的应用。如果你的操作系统是Mac，则可以使用iPhoto来编辑照片。如果你是Windows用户，则可以看到你的计算机上已经有Microsoft提供的Digital Image Suite。如果确实没有一个可用的编辑应用，则可以先跟着我们继续学习，每一步都可以使用示例文件夹中已经包含的HTML和图像。

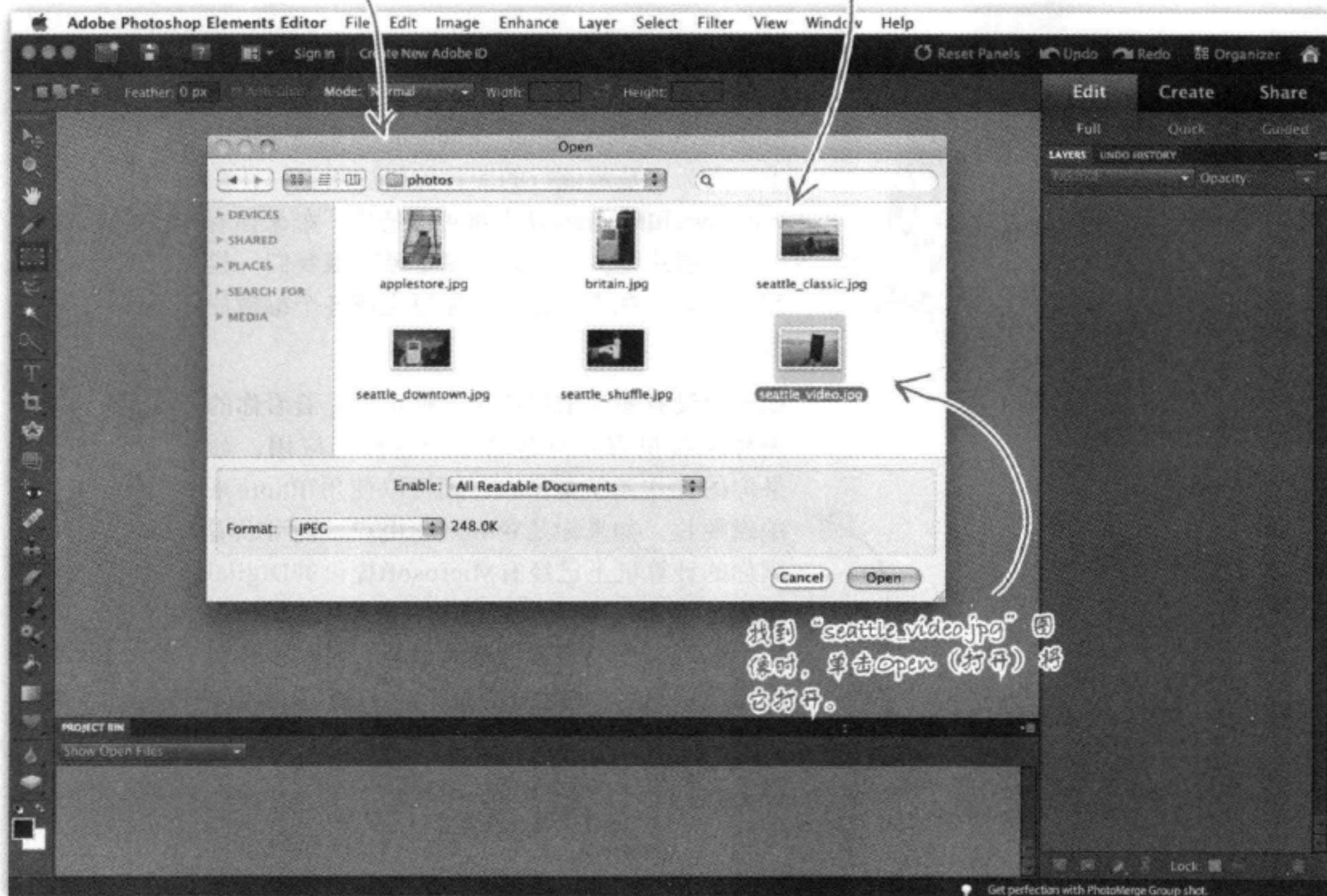
如果没有Adobe Photoshop Elements，但还想跟着我们学习这一章，则可以先下载这个软件的试用版，能免费使用30天。下载URL为：http://www.adobe.com/go/tryphotoshop_elements。

打开图像

首先，启动你的照片编辑应用，打开“seattle_video.jpg”图像。在Photoshop Elements中，可以选择File（文件）菜单下的“Open…”（打开……）菜单项，这会打开一个Open（打开）对话框。可以使用这个对话框导航到“chapter5/mypod/photos”文件夹中的图像“seattle_video.jpg”。

这就是Open（打开）对话框。可以使用这个对话框导航到“seattle_video.jpg”图像。

在文件夹中导航时，你会在这里看到这些文件夹中图像的预览图。



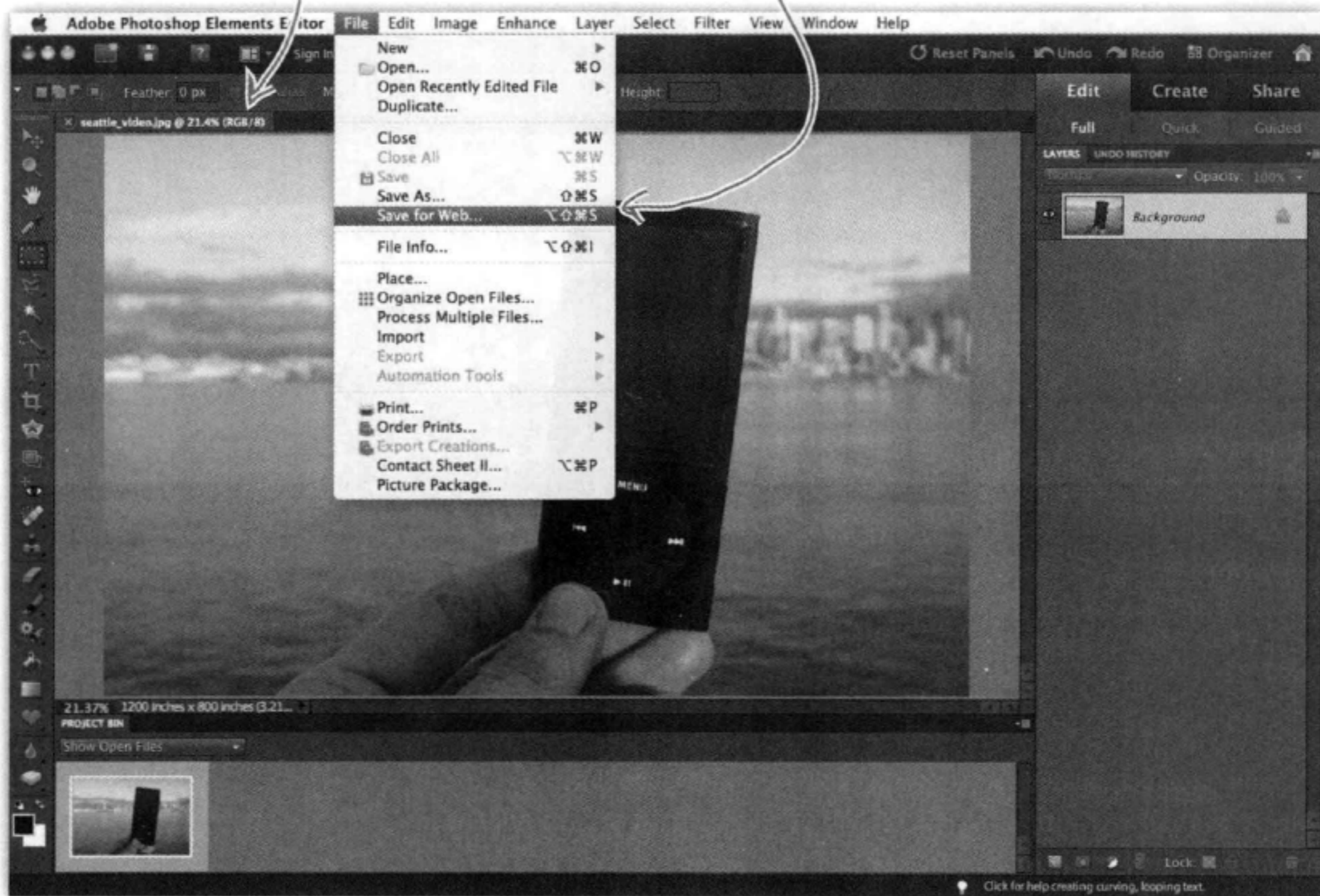
找到“seattle_video.jpg”图像时，单击Open（打开）将它打开。

调整图像大小

现在“seattle_video.jpg”已经打开，我们要用“Save for Web”（保存为Web格式）对话框调整图像大小并保存。要打开这个对话框，只需要从File（文件）菜单选择“Save for Web”（保存为Web格式）单项。

这里是Photoshop Elements中打开的“seattle_video.jpg”图像。

要调整图像大小，从File（文件）菜单选择“Save for Web”（保存为Web格式）。

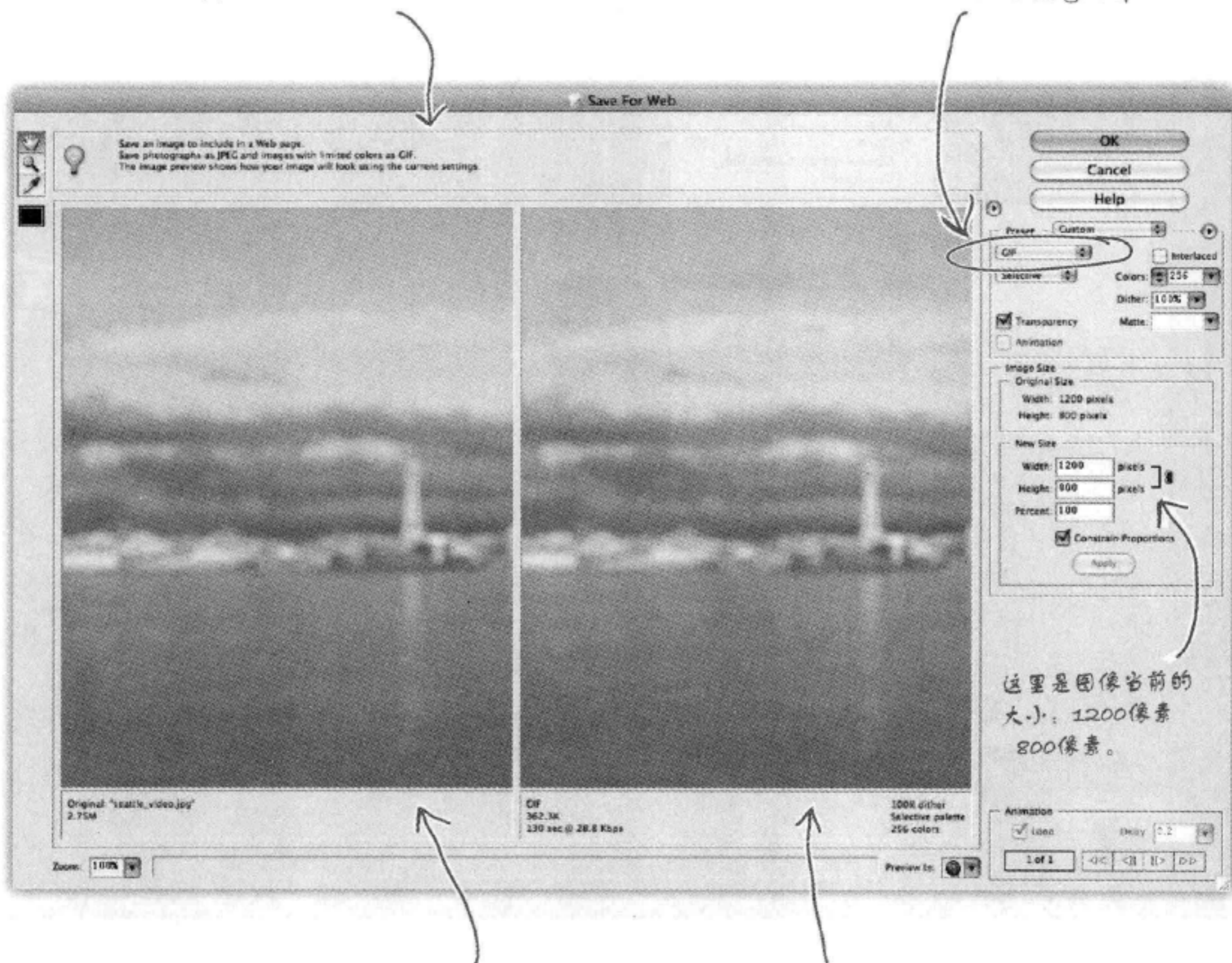


调整图像大小（续……）

选择“Save for Web”（保存为Web格式）菜单项之后，应该会看到下面的对话框，在使用之前先来熟悉一下。

这个对话框允许做各种各样有意思的事情。现在我们只关心如何用它调整图像大小，并为Web页面保存为JPEG格式。

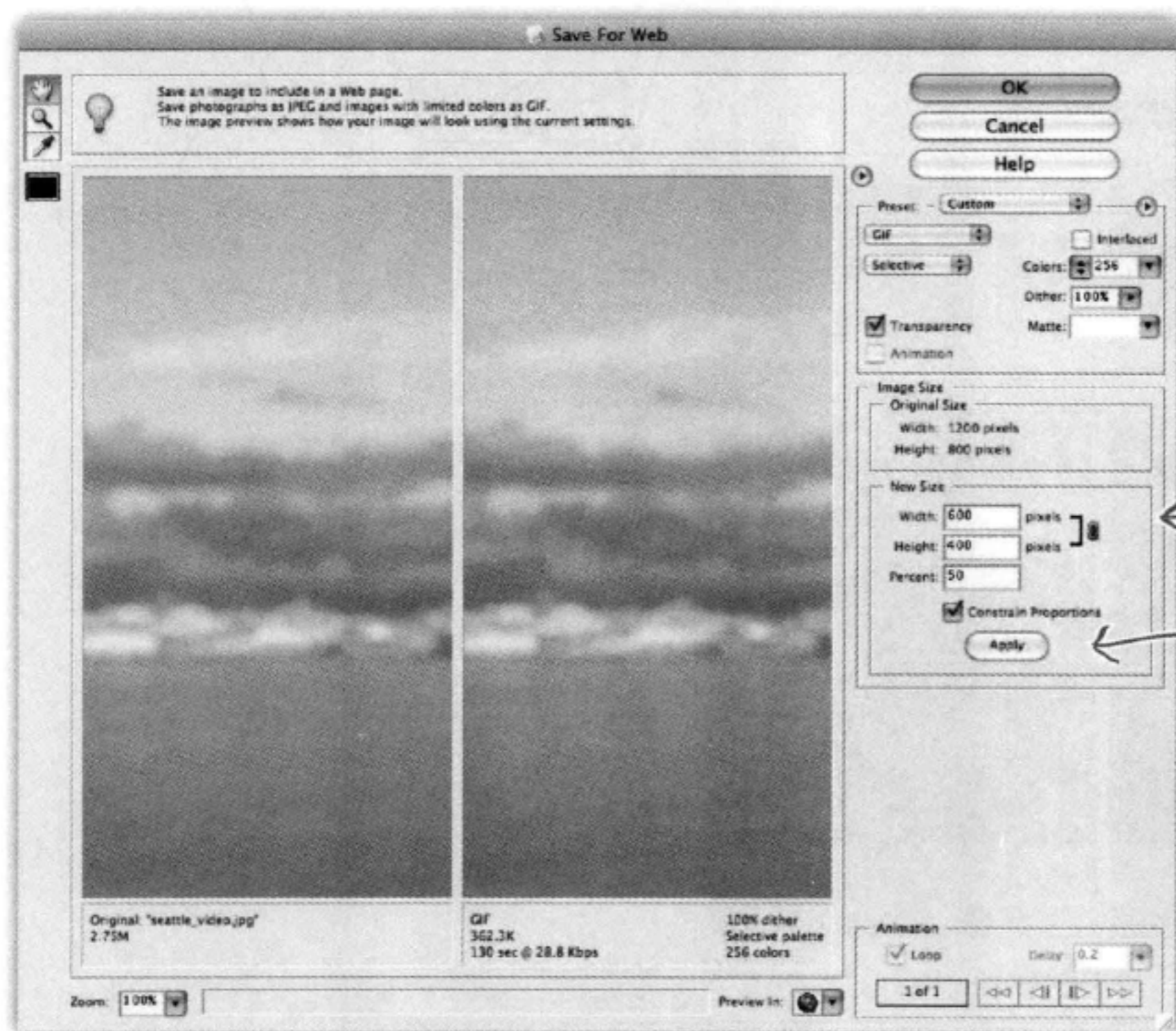
要在这里选择需要将文件保存为什么格式。目前设置为GIF，后面几页我们要将它改为JPEG……



这个拆分窗口在左边显示你原来的图像，在右边显示采用你选择的Web格式保存的图像。现在这里显示的是GIF格式，下一步我们要把它改为JPEG格式。

可以看到，这个对话框里内置有很多功能。下面来充分利用这个对话框。要调整图像的大小，需要把宽度改为600像素，高度改为400像素。然后需要用JPEG格式保存这个图像。下面先来调整图像的大小……

(1) 在这里调整图像大小，改为宽度600，高度400。如果选中了Constrain Proportions（保持纵横比），那么你只需要输入新的宽度600，Elements就会帮你把高度改为400。



(2) 一旦正确地设置了宽度和高度，就单击Apply（应用），让Elements知道这是你想要的大小。

这不会影响原来的图像，只会影响你要保存的那个文件。

必须单击Apply（应用）来缩小图像大小，否则图像将仍按原宽度和高度保存。

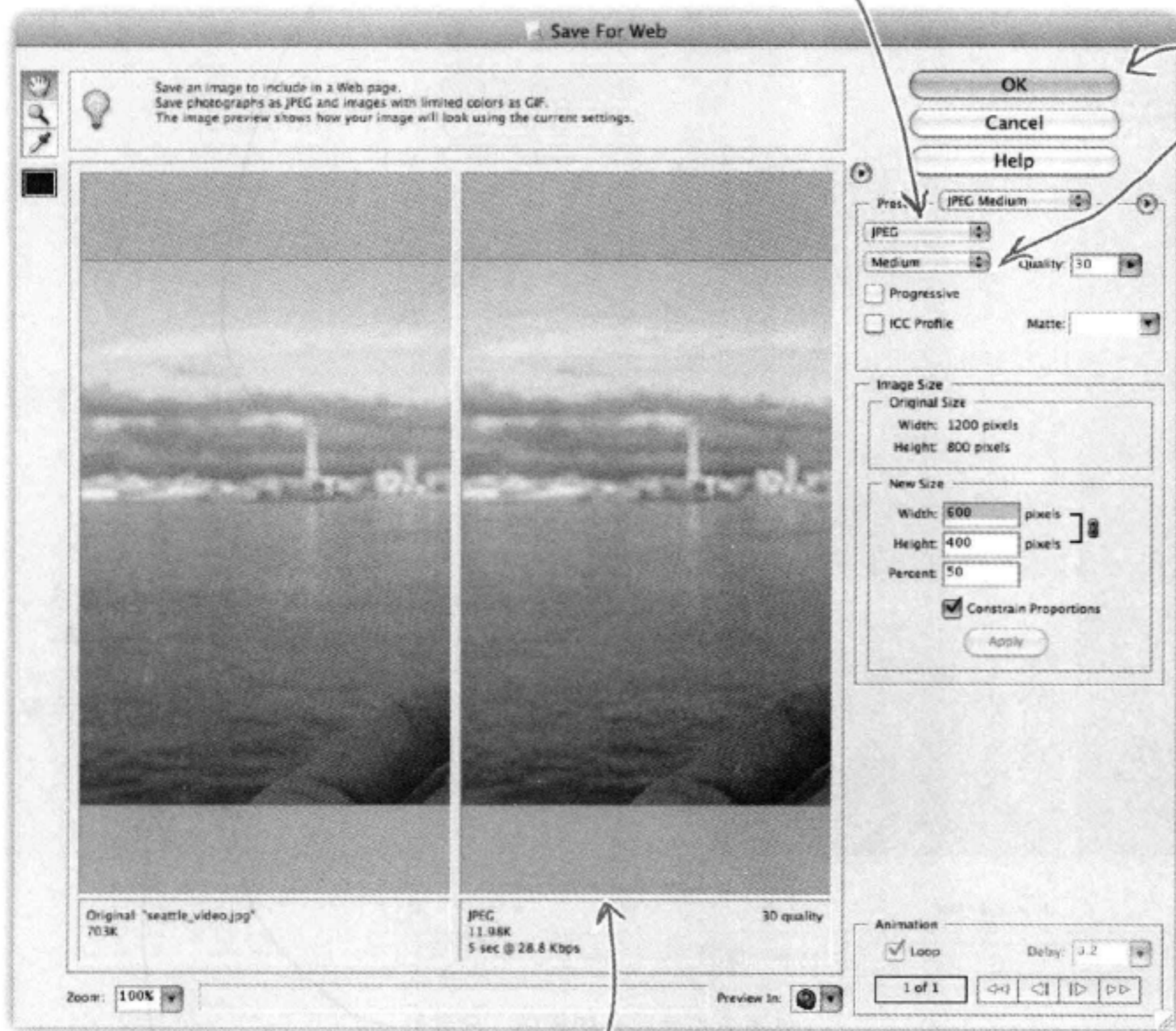
大小调整完毕，现在来保存

现在你只需要按正确的格式（JPEG）保存图像。为此，需要选择JPEG格式，将质量设置为Medium（中）。稍后我们再来讨论质量问题。

(1) 现在已经设置了图像大小，只需要选择图像格式。目前的设置是保存为GIF，现在就像我们一样，将它改为JPEG。

(2) 将质量设置为Medium（中）。

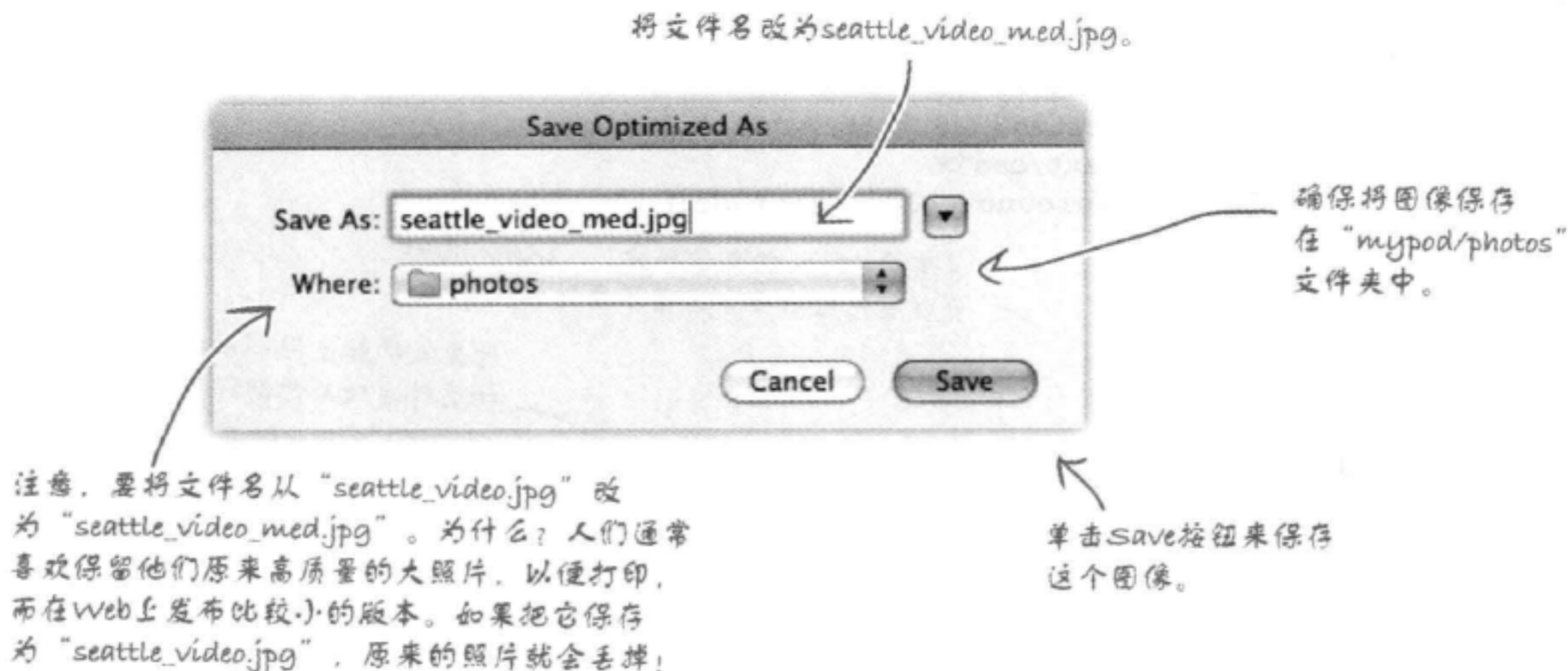
(3) 这就可以了，单击OK按钮，来看下一页。



注意，上一步单击Apply（应用）按钮时，会调整图像大小并重新显示。

保存图像

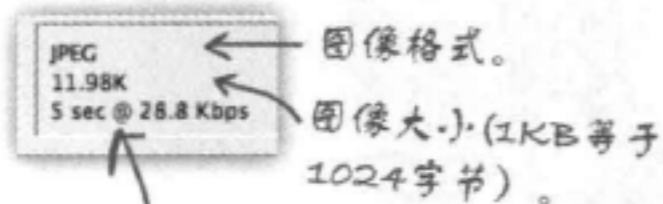
单击OK按钮之后，会打开一个Save（保存）对话框。将图像保存为“seattle_video_med.jpg”，这样就不会覆盖原来的照片了。



there are no Dumb Questions

问：能不能再讲讲“Save for Web”（保存为Web格式）中的质量设置？

答：JPEG格式允许指定你需要的图像质量等级。质量越低，文件大小就越小。如果查看“Save for Web”（保存为Web格式）对话框中的预览窗口，则可以看到改变质量设置时，质量和文件大小会随之改变。



Photoshop Elements甚至会告诉你通过一个拨号modem传输到浏览器需要多长时间。

要对质量设置和不同图像格式有所认识，最好的办法就是实践，用你自己的图像做些试验。你会很快发现，对于你的图像和所要开发的Web页面类型，需要怎样的质量等级。同时你还能逐渐了解何时使用JPEG格式而不是其他格式。

问：在JPEG选项对话框中，Quality（质量）标签旁边的数字30是什么意思？

答：30就是Photoshop Elements所认为的Medium（中）质量。JPEG实际上使用1%~100%范围内的数值，Low（低）、Medium（中）和High（高）等只是表示很多照片编辑应用常用的一

些值。

问：不能直接用元素的width属性和height属性来调整图像的大小吗？

答：可以使用width属性和height属性来调整一个图像的大小，不过这不是一个好主意。为什么？因为如果这样做，则仍然会下载完整的大图像。让浏览器来完成调整图像大小的工作，就像你打开了浏览器上的自动调整大小选项一样（如果浏览器支持这个特性）。width属性和height属性实际上是帮助浏览器确定要为此图像预留多大的空间。如果使用这两个属性，则它们应当与图像的实际宽度和高度一致。

修复myPod HTML

一旦保存了图像，就可以关闭Photoshop Elements。现在只需要修改myPod “index.html” 页面，在其中包含照片“seattle_video_med.jpg”的新版本。下面是“index.html”文件的一个片段，只给出了需要你修改的部分。

```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>
    .
    .
    .
    <h2>Seattle, Washington</h2>
    <p>
      Me and my iPod in Seattle! You can see the
      Space Needle. You can't see the 628 coffee shops.
    </p>

    <p>
      
    </p>

  </body>
</html>
```

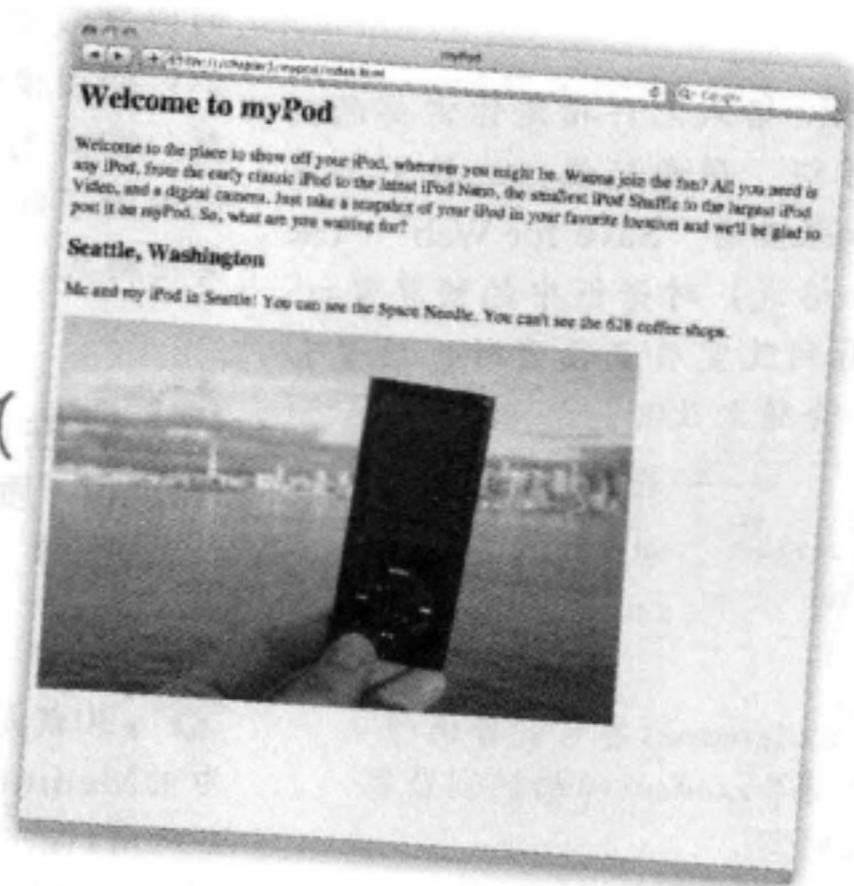
这里是HTML的其余部分，“index.html”文件里已经有了，这里不再列出。

你要做的就是将元素中的文件名改为你刚保存的图像名：“seattle_video_med.jpg”。

现在来试一试……

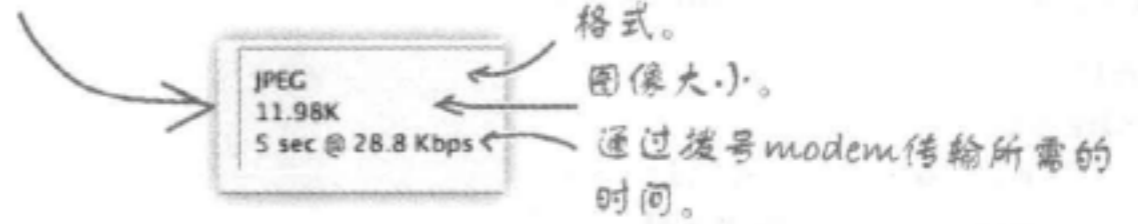
完成以上修改，保存“index.html”文件，并在浏览器中重新加载。看起来好多了。现在图像大小正合适，很方便访问者查看，再不会是一个庞大的照片让访问者不知所措了。

现在图像可以在浏览器窗口中很好地显示。而且文件大小也小多了，这样可以帮助页面更快地加载。



哪一种图像格式？

这一次的任务：在Photoshop Elements中打开文件“chapter5/testimage/eye.jpg”。打开“Save for Web”（保存为Web格式）对话框，选择JPEG的各种质量设置（Low、Medium、High等），再试试PNG-24和GIF，填写下面的空格。你会在图像下面的预览窗口中找到这些信息。完成后，确定对于这个图像哪种设置最合适。



也试试PNG-8!

格式	质量	大小	时间	优胜者
<u>PNG-24</u>	<u>N/A</u>	_____	_____	<input type="checkbox"/>
<u>JPEG</u>	<u>Maximum</u>	_____	_____	<input type="checkbox"/>
<u>JPEG</u>	<u>High</u>	_____	_____	<input type="checkbox"/>
<u>JPEG</u>	<u>Medium</u>	_____	_____	<input type="checkbox"/>
<u>JPEG</u>	<u>Low</u>	_____	_____	<input type="checkbox"/>
<u>GIF</u>	<u>N/A</u>	_____	_____	<input type="checkbox"/>

myPod的更多照片

又为myPod提供了一批新的照片：3张来自西雅图，还有一张是英国的一个朋友拍摄的。这些照片大小已经调整，宽度小于800像素。为它们增加元素（你会发现这些图像已经在photos文件夹中）：

```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>
    <h1>Welcome to myPod</h1>
    <p>
      Welcome to the place to show off your iPod, wherever you might be.
      Wanna join the fun? All you need is any iPod, from the early classic
      iPod to the latest iPod Nano, the smallest iPod Shuffle to the largest
      iPod Video, and a digital camera. Just take a snapshot of your iPod in
      your favorite location and we'll be glad to post it on myPod. So, what
      are you waiting for?
    </p>
    <h2>Seattle, Washington</h2>
    <p>
      Me and my iPod in Seattle! You can see the
      Space Needle. You can't see the 628 coffee shops.
    </p>
    <p>
      
      
      
      
    </p>
    <h2>Birmingham, England</h2>
    <p>
      Here are some iPod photos around Birmingham. We've obviously got some
      passionate folks over here who love their iPods. Check out the classic
      red British telephone box!
    </p>
    <p>
      
      
    </p>
  </body>
</html>
```

这里还可以增加你自己的一些照片。
不过要记住首先要调整照片的大小。

这里把所有西雅图
的照片放在一起。



伯明翰的照片也做
同样的处理……

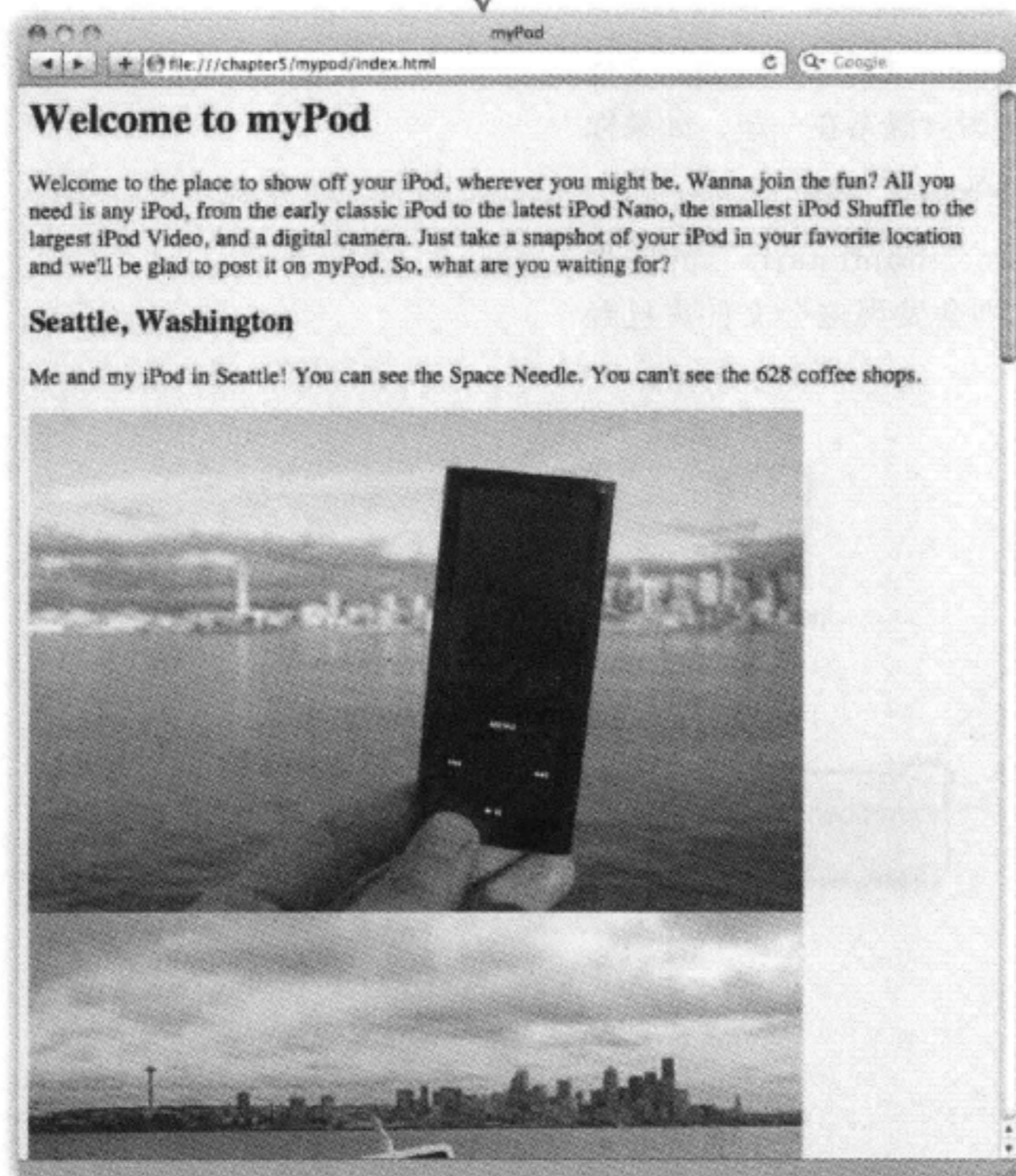


再试试myPod

现在不需要我们再多说，相信你肯定早就知道了：下面在浏览器中重新加载这个页面。哇，多几个图像就大不一样了，是不是？这个页面开始有点意思了。

不过，这并不表示你已经大功告成。现在已经在页面上放置了一组图像，不过即使已经调整了这些图像的大小，它们还是太大了。增加更多的图像时，不仅页面加载越来越慢，用户还必须大量滚动才能看到全部图像。如果用户能看到每个照片的一个小“缩略”图像，然后单击这个缩略图来查看更大的图像，这样不是更好吗？

近看页面是这样的。



现在加上所有图像，整个页面是这样的。



修改网站，使用缩略图

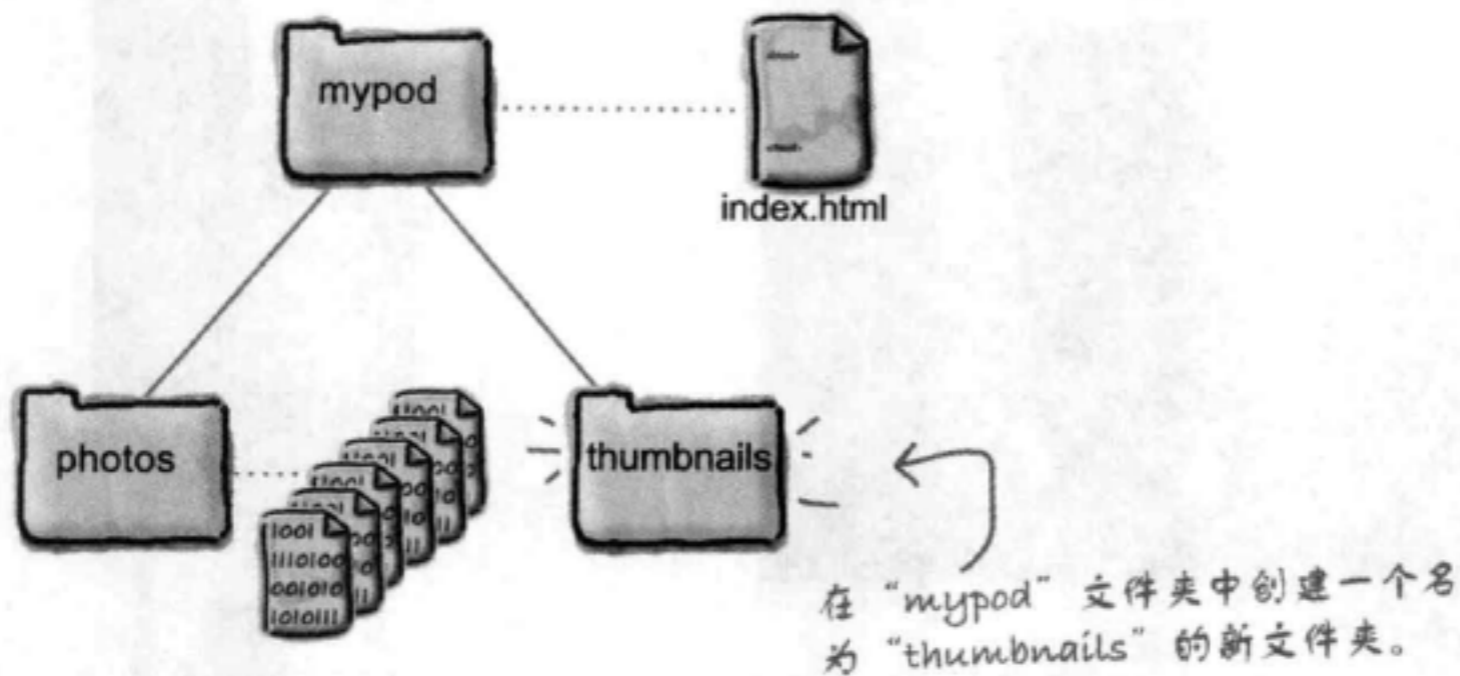
现在要让这个页面更可用，把每个照片替换为一个更小的图像（我们把它叫做缩略图），然后创建从这个缩略图到各个更大照片的链接。你要按以下步骤完成这个工作：

- 1 为缩略图创建一个新目录。
- 2 把各个照片的大小调整为150 像素 × 100像素，把它保存到“thumbnail”文件夹中。
- 3 将“index.html”中各个元素的src设置为照片的缩略版本。
- 4 增加从各个缩略图到一个新页面的链接，这个新页面中包含相应的大照片。

为缩略图创建一个新目录

为了更有条理、更有组织，就为这些缩略图创建一个单独的文件夹。否则，图像文件夹中大图像和缩略图会混杂在一起，如果你增加了大量照片，则情况会变得相当混乱。

在“mypod”文件夹下创建一个名为“thumbnails”的文件夹。如果你使用这本书的示例文件，则会发现这个文件夹已经在“mypod”文件夹下了。



创建缩略图

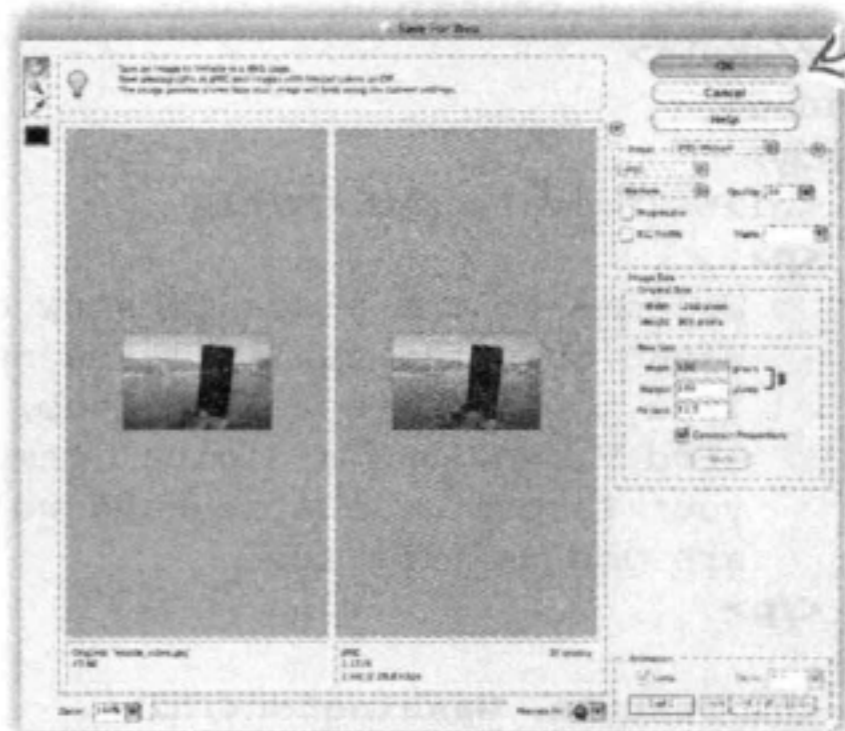
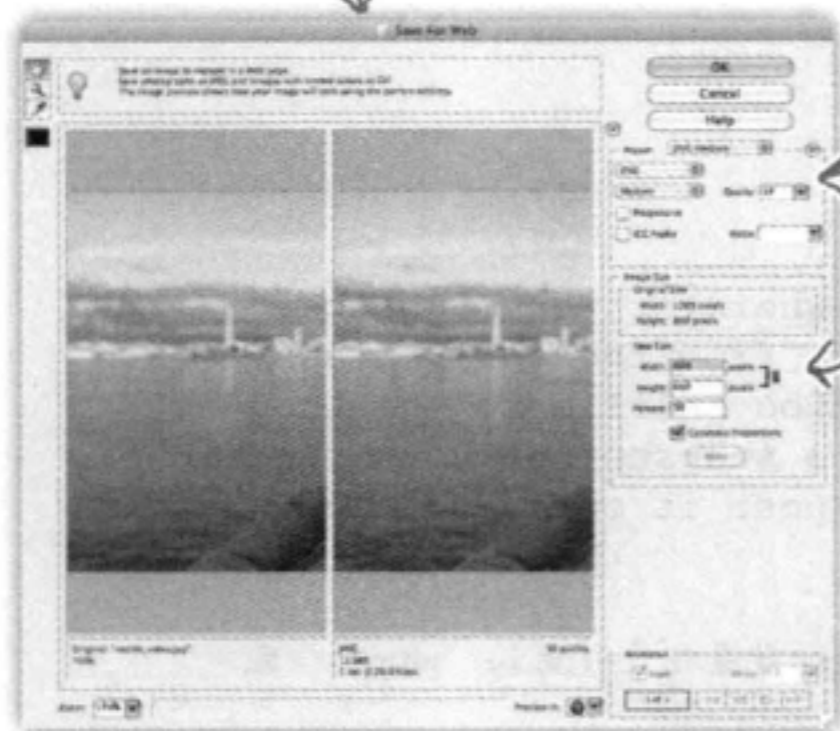
已经有了放置缩略图的地方，下面来创建这些缩略图。首先在你的照片编辑应用中打开“seattle_video_med.jpg”。使用之前创建600像素×400像素版本时同样的方法把它的大小调整为150像素×100像素：

在Photoshop Elements中，选择“Save for Web”（保存为Web格式）菜单项。

然后把宽度改为150，高度改为100，再单击Apply（应用）。

不要忘记把格式改为JPEG，设置为质量为Medium（中）。

最后，单击OK按钮。



调整图像大小后，选择OK按钮，并按同名保存，不过要保存到thumbnail文件夹中。要当心：如果把它保存到“photos”文件夹，就会替换原来的大图像。

现在，对“photos”文件夹中的每个照片重复这个处理。

如果使用示例文件，则你会发现这些缩略图已经在“thumbnails”文件夹中，所以你不用这样处理每一张照片（毕竟，你是要学习HTML，而不是批量处理照片）。

来自伯明翰的照片怎么处理，它们高度大于宽度。调整为150x100合适吗？



问得好。因为这些图像高大于宽，我们有两种选择：可以调整长宽比，把它们设置为100×150，或者也可以裁剪各个图像，由原图像裁剪出一个150像素×100像素的缩略图。我们打算采用前一种做法，调整为100×150的缩略图；如果你想研究一下如何在照片编辑应用中完成裁剪，则完全可以采用裁剪的方法，创建150像素×100像素的缩略图像。



修改HTML，使用缩略图

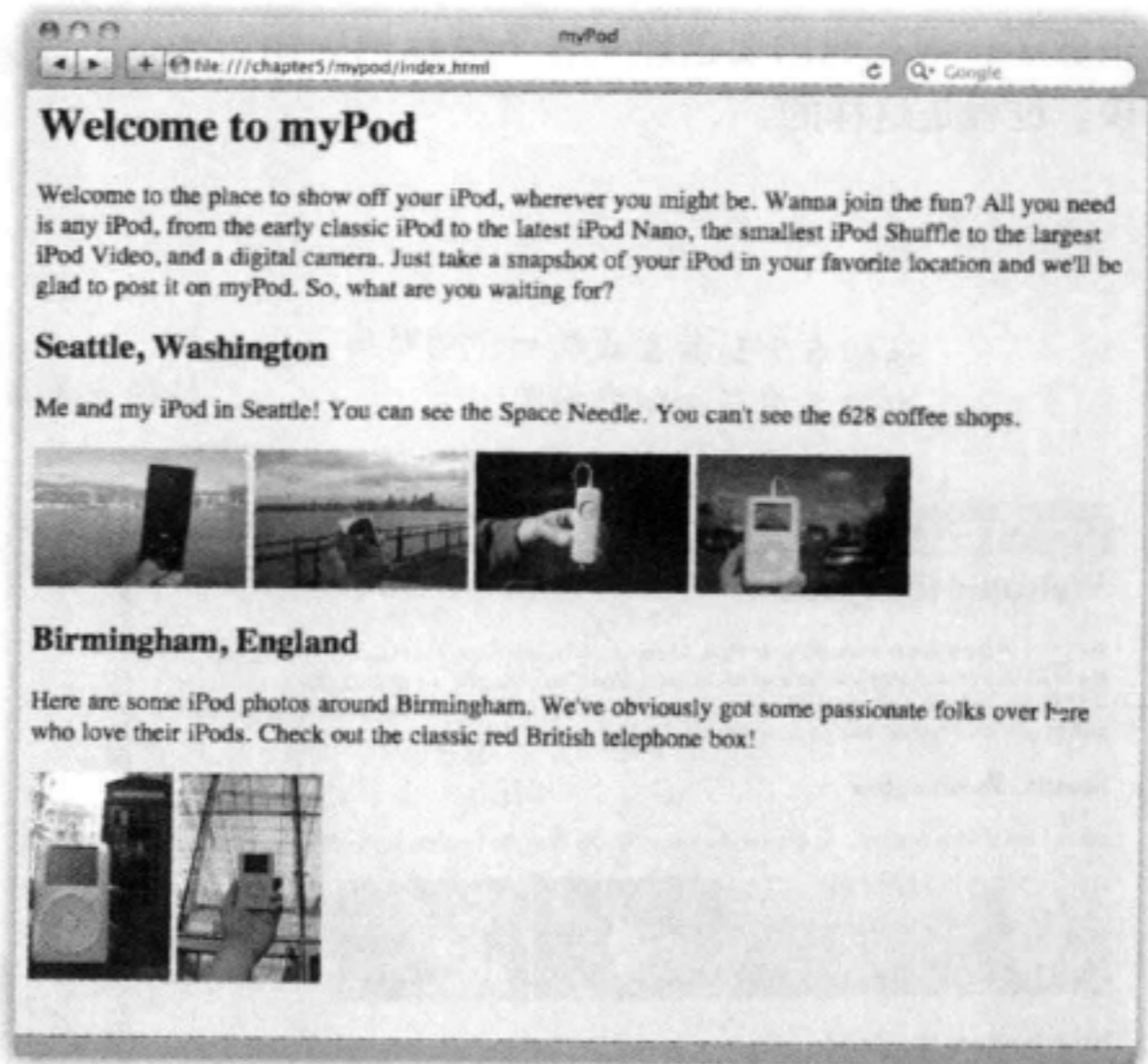
现在只需要修改HTML，使

```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>
    <h1>Welcome to myPod</h1>
    <p>
      Welcome to the place to show off your iPod, wherever you might be.
      Wanna join the fun? All you need is any iPod, from the early classic
      iPod to the latest iPod Nano, the smallest iPod Shuffle to the largest
      iPod Video, and a digital camera. Just take a snapshot of your iPod in
      your favorite location and we'll be glad to post it on myPod. So, what
      are you waiting for?
    </p>
    <h2>Seattle, Washington</h2>
    <p>
      Me and my iPod in Seattle! You can see the
      Space Needle. You can't see the 628 coffee shops.
    </p>
    <p>
      
      
      
      
    </p>
    <h2>Birmingham, England</h2>
    <p>
      Here are some iPod photos around Birmingham. We've obviously got some
      passionate folks over here who love their iPods. Check out the classic
      red British telephone box!
    </p>
    <p>
      
      
    </p>
  </body>
</html>
```

你要做的就是将文件夹从“photos”改为“thumbnails”。

再来测试myPod

哈……好多了。访问者可以一眼看到所有可用的图片。他们还能更容易地区分不同城市的照片。现在我们需要想办法从每个缩略图链接到相应的大图像。



等一下，你不觉得你在捣鬼吗？图像原来是上下摆放的，现在变成并排的了。



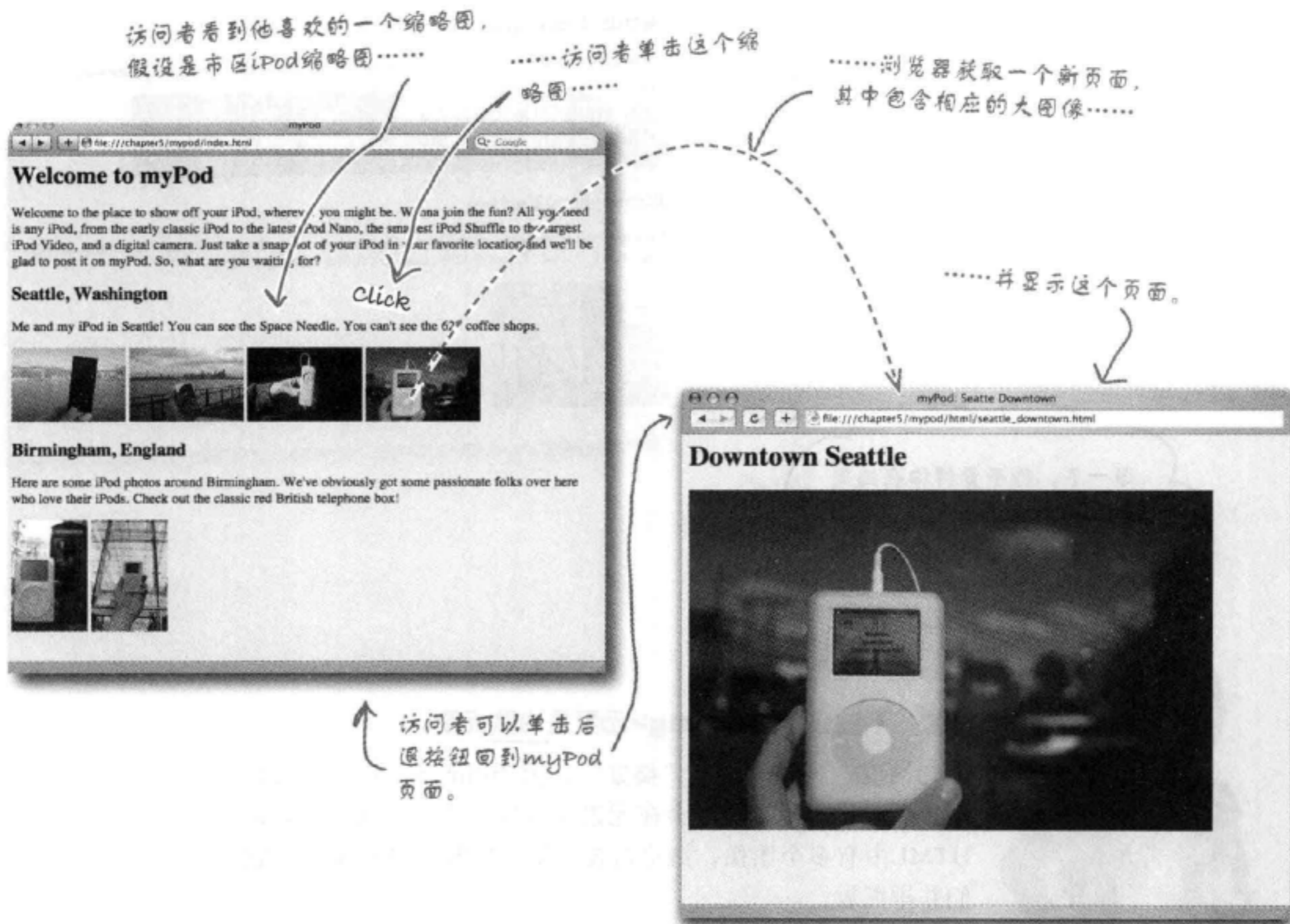
没错，不过应该记得元素是内联元素。

换句话说，我们并没有“捣鬼”。因为显示为内联元素，所以显示元素时不会在元素前后插入换行。因此，如果HTML中有多个图像，浏览器窗口足够宽时，浏览器会把它们并排摆放。

那些更大的照片之所以没有并排摆放，是因为浏览器没有足够的空间来并排显示它们。实际上，它只能从上到下显示各个大图像。浏览器总是在块元素前后在垂直方向上显示间距，如果再看看前面的截屏图，则你会看到那些图像尽管上下摆放，但是相互之间没有间距。这也说明了是一个内联元素。

把缩略图变成链接

差不多了。现在只需要创建从各个缩略图到相应的更大版本的一个链接。过程是这样的：



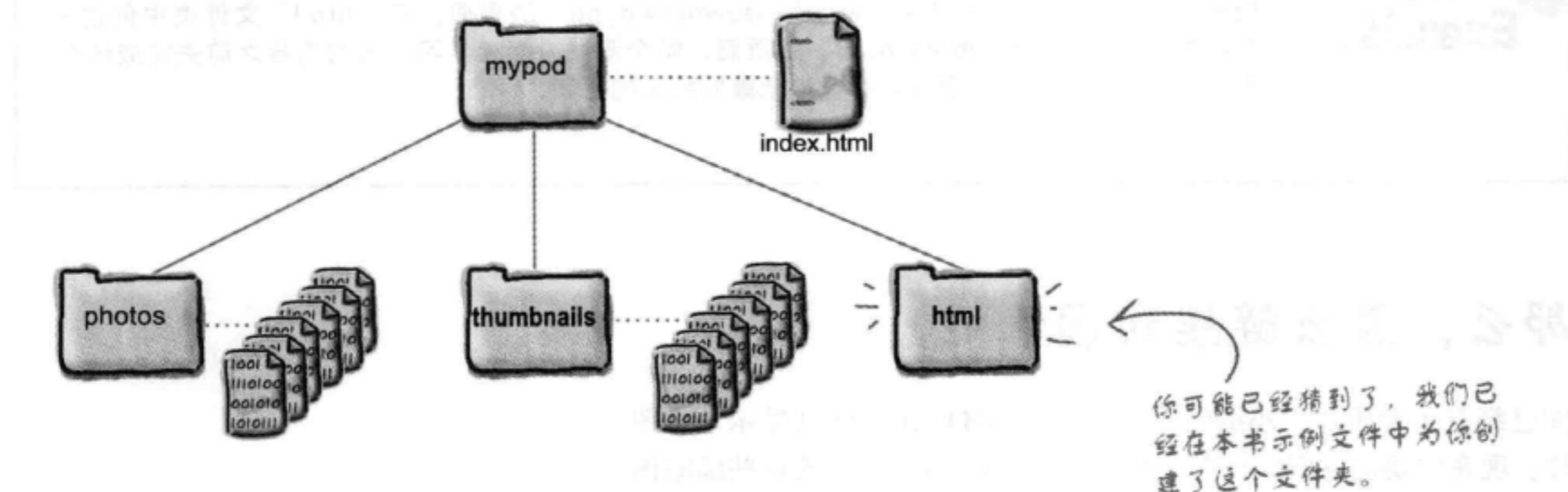
为此你需要两样东西：

- 1 对应每个照片要有一个页面，其中显示各个照片，还要有一个标题来描述照片内容。
- 2 “index.html” 中从每一个缩略图到相应照片页面的一个链接。

下面先来创建页面，然后再回来完成所有链接。

为照片创建单独的页面

首先，在“mypod”文件夹下创建一个名为“html”的新文件夹，存放所有这些单独的页面：



现在我们要为每个照片创建一个HTML文件。如果照片名为“seattle_video_med.jpg”，那么相应的HTML文件就命名为“seattle_video_med.html”，以保持一致。在每个HTML文件中，要有一个标题来描述这个照片，后面是照片本身。下面是第一张西雅图照片的HTML。所有其他页面也采用同样的格式：

```

<html>
  <head>
    <title>myPod: Seattle Ferry</title>
    <style type="text/css"> body { background-color: #eaf3da; } </style>
  </head>
  <body>
    <h1>Seattle Ferry</h1>
    <p>
      
    </p>
  </body>
</html>

```

页面标题，应当能够描述这个照片。

下面依然是我们之前提供的成品CSS，只是为了让页面保持一致的颜色。

这里为页面提供一个描述性的标题。

这里是元素，指向大照片“seattle_video_med.jpg”。下面为这个图像提供一个描述性的alt属性。

注意我们需要在相对路径中使用“..”，因为“html”文件夹是“photos”文件夹的“兄弟”，所以使用相对链接时必须进入上一层文件夹，然后再向下进入“photos”。

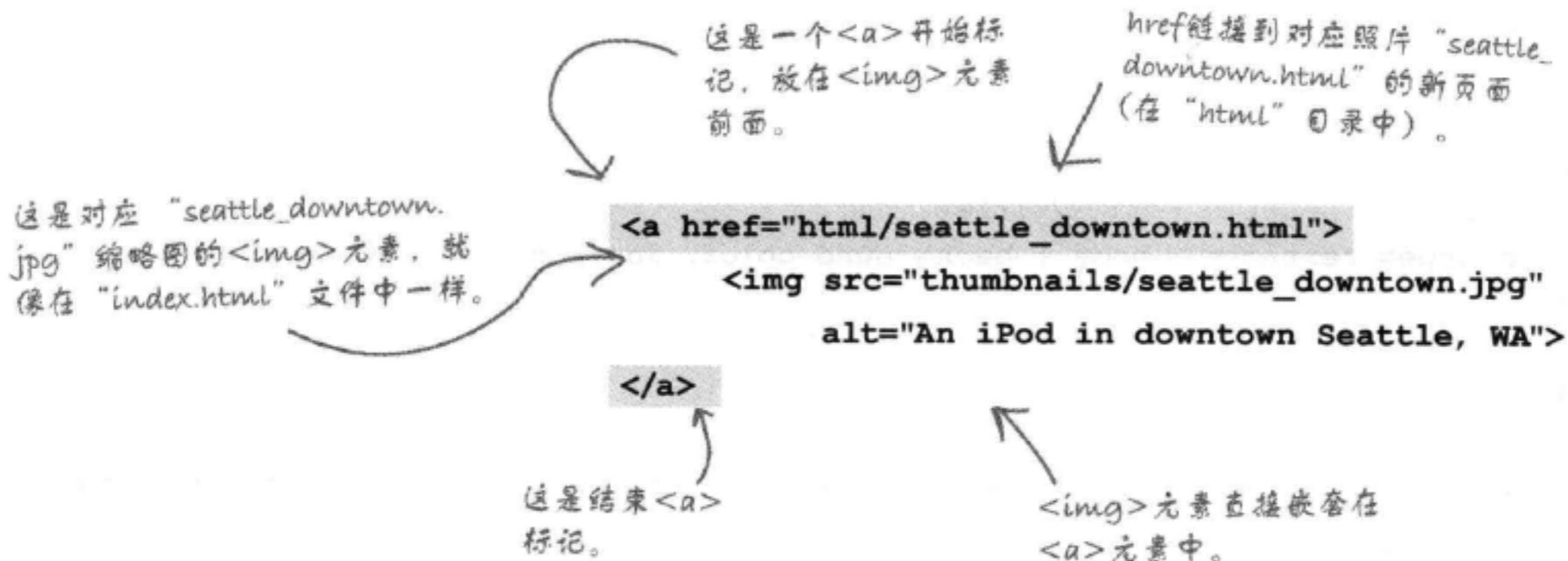


如果查看本章示例文件中的“html”文件夹，则你会发现其中已经提供了所有的照片页面，但有一个页面除外——对应“seattle_downtown.jpg”的页面。在“html”文件夹中创建一个名为“seattle_downtown.html”的页面，做个测试。继续学习下面的内容之前先完成这个工作。如果遇到问题，则可以参考本章最后给出的答案。

那么，怎么链接到图像呢？

你已经有了大照片，小缩略图，还有一组HTML页面来显示单个照片。现在需要把它们综合在一起，让“index.html”中的这些缩略图链接到“html”文件夹中的页面。不过怎么做到呢？

要链接到一个图像，需要把元素放在<a>元素中，就像这样：



一旦把元素放在一个<a>元素中，浏览器就会把这个图像当作一个可单击的链接。单击这个图像时，浏览器会获取href中的页面。

为“index.html”增加图像链接

这是最后一步了。你还需要在“index.html”文件中用<a>元素包围各个缩略图的元素。要记住，各个<a>元素的href应当链接到“html”文件夹中包含更大图像的页面。要确保你的链接、缩略图和页面能正确对应。

下面是完整的“index.html”文件。你只需要增加用灰色标注的HTML。

```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>

    <h1>Welcome to myPod</h1>
    <p>
      Welcome to the place to show off your iPod, wherever you might be.
      Wanna join the fun? All you need is any iPod, from the early classic
      iPod to the latest iPod Nano, the smallest iPod Shuffle to the largest
      iPod Video, and a digital camera. Just take a snapshot of your iPod in
      your favorite location and we'll be glad to post it on myPod. So, what
      are you waiting for?
    </p>

    <h2>Seattle, Washington</h2>
    <p>
      Me and my iPod in Seattle! You can see the
      Space Needle. You can't see the 628 coffee shops.
    </p>

    <p>
    <a href="html/seattle_video_med.html">
      
    </a>
    <a href="html/seattle_classic.html">
      
    </a>
    <a href="html/seattle_shuffle.html">
      
    </a>
  </body>
</html>
```

```
<a href="html/seattle_downtown.html">
  
</a>
</p>

<h2>Birmingham, England</h2>
<p>
  Here are some iPod photos around Birmingham. We've obviously got some
  passionate folks over here who love their iPods. Check out the classic
  red British telephone box!
</p>

<p>
<a href="html/britain.html">
  
</a>
<a href="html/applestore.html">
  
</a>
</p>
</body>
</html>
```

用一个<a>元素包围每一个缩略图图像。要小心，每个链接里的href要正确！

把这些<a>元素增加到你的“index.html”文件。保存文件，然后在浏览器中加载页面，感受一下你的myPod网站吧！



there are no Dumb Questions


问：用<a>元素包围文本时，会有一个下划线。为什么图像没有相应的标志呢？

答：确实，Internet Explorer会在图像周围加一个边框，显示这是一个链接（我们的浏览器是Safari，它不会这么做）。如果你的浏览器会在链接图像周围加一个边框或者在下面加一条线，但你并不喜欢这样，可以再等几章，后面会教你如何用CSS来改变这种效果。还要注意，鼠标经过图像时，光标会改变，指示你可以单击这个链接图像。大多数情况下，用户都可以通过上下文和鼠标光标了解到

一个图像是链接（即使图像周围没有边框）。

问：能不能直接链接到JPEG图像，而不是那些HTML页面？我想浏览器肯定很聪明，能直接显示图像。

答：你说的没错，确实可以直接链接到图像，就像这样：`...`。如果这样做，单击链接时，浏览器就会在一个空页面显示这个图像。不过，一般认为直接链接到一个图像并不是一个好的做法，因为通常你需要为所显示的图像提供一些上下文。



myPod Web页面看起来
很炫！我想你应该为页面
加一个logo,这样会增色不少。

好主意。实际上，我们已经有一个myPod logo。

再来看文件夹“chapter5/mypod”，你会找到一个名为“logo”的文件夹。在这个文件夹中，可以找到一个名为“mypod.psd”的文件。“psd”表示这个文件是以Photoshop格式保存的，这是图像编辑软件的一种常用格式。不过Photoshop格式文件主要用于处理数字图像，而不是用于Web页面，所以我们必须做些工作，由它得到一个为“为Web准备的”图像。

很多照片编辑应用都支持.psd文件，所以即使你没有Photoshop Elements，也可以继续阅读下面几页。如果你的应用无法打开“.psd”文件，则可以在“logo”文件夹中找到每一步的图像。

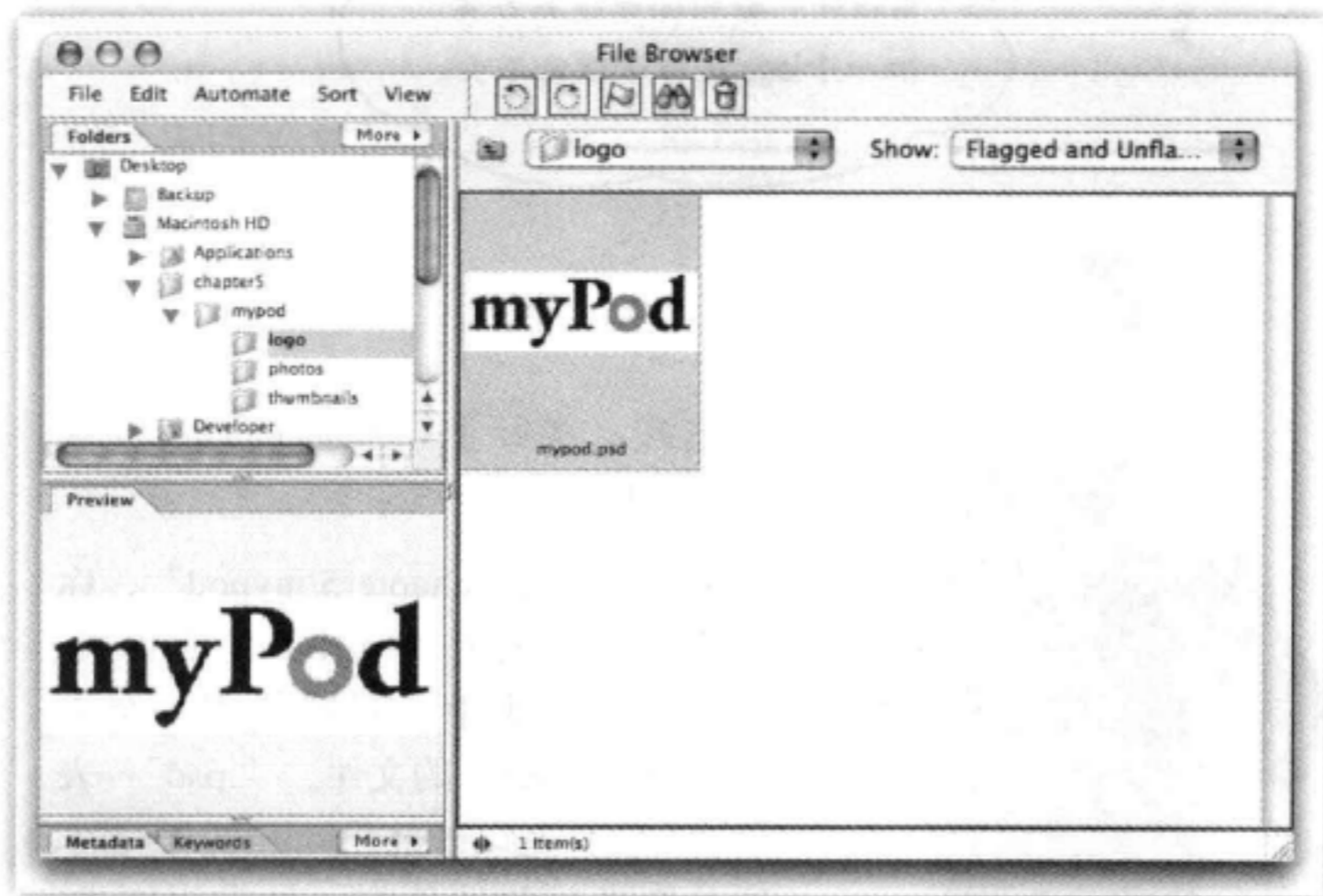
增加一个logo

打开myPod logo

下面来看myPod logo: 在Photoshop Elements中打开“chapter5/mypod/logo”文件夹中的“mypod.psd”:

在“chapter5/mypod”文件夹中可以找到“logo”文件夹。

如果你的照片编辑应用无法打开这个文件, 也请跟着我们继续学习, 这些原则也同样适用于其他格式。



进一步分析……

不错的logo。这里有一些文字, 还有两个圆, 一个灰色, 另一个白色 (这显然是受了iPod上经典的单击式触摸转盘控件的启发)。

不过, 背景里的棋盘图案是什么? 大多数照片编辑应用都会以这种方式告诉你这个区域是透明的。为logo选择图形格式时要记住这些……

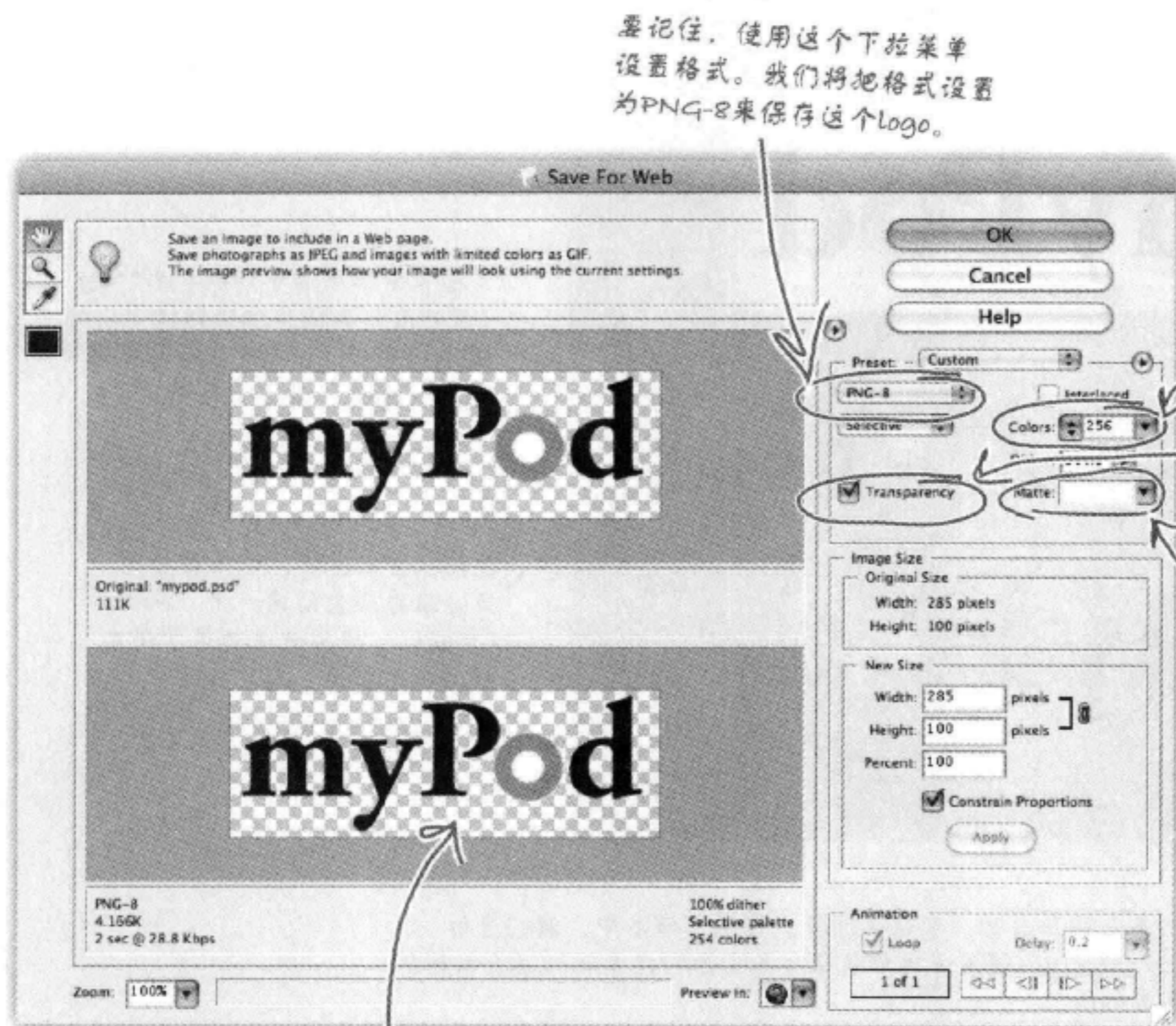


只要看到这种棋盘图案, 就说明图像中有一个透明区域。

要使用什么格式？

你已经知道，在决定如何保存这个图像时有几种选择：可以使用JPEG、PNG或GIF。这个logo只使用了3种颜色，另外有文本和一些几何形状。根据你对PNG和GIF这两种格式的了解，你可能已经倾向于PNG或GIF了。这两种格式都可以。图像质量相同时，PNG文件可能稍小一点，所以我们选择PNG。另外，因为我们只有3种颜色，所以完全可以使用只支持256种颜色的PNG-8，使用这种格式可以让文件大小进一步缩小。

因此，选择File（文件）菜单下的“Save for Web”（保存为Web格式）选项，再在格式下拉框中选择PNG-8。你会看到更多选项。下面来看一下……



要记住，使用这个下拉菜单设置格式。我们将把格式设置为PNG-8来保存这个Logo。

Photoshop Elements在这里告诉你保存这个PNG使用的颜色数。它已经设置为PNG-8的最大值，即256。保留这个值不变。

将格式设置为PNG时，会出现这个Transparency（透明度）复选框。默认地，这个复选框会选中。我们需要一个透明背景吗？

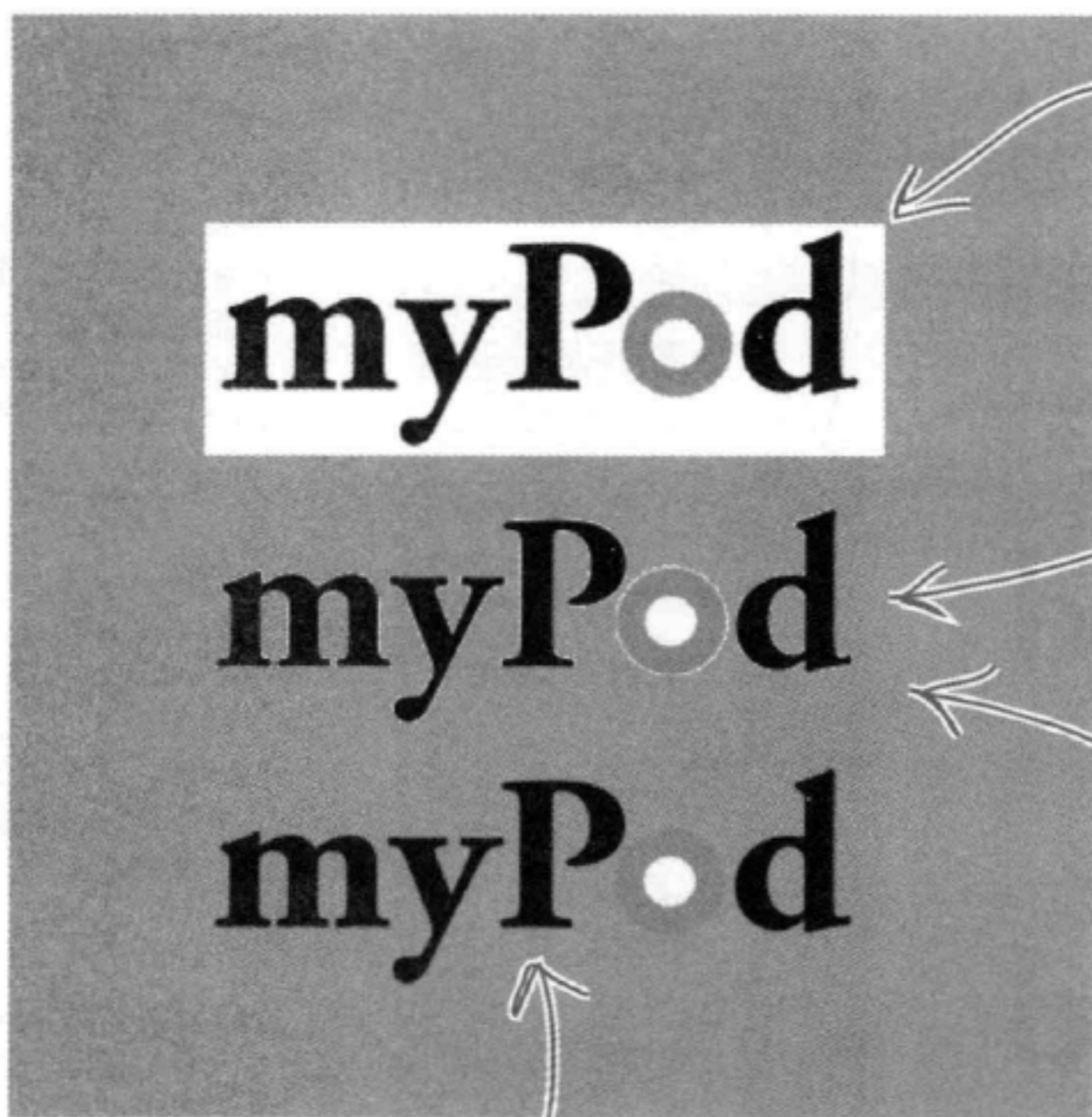
还要注意Matte（蒙版）选项。这也与透明度有关，稍后会看到。

试着取消选中Transparency（透明度）复选框，你会看到下面的预览变为一个白色背景。

透明，还是不透明？这是个问题……

myPod logo要放在一个淡绿色背景中，所以你可能认为透明比较好，是吗？嗯，下面来比较一下，看看使用“Save for Web”（保存为Web格式）对话框中的几种选项时logo看起来怎么样：

下面用3种不同方式保存这个logo，并在一个有绿色背景的Web页面中显示。



没有透明度，情况很糟糕。很显然，白色背景在绿色页面上很不合适（不过，在白色页面上可能还可以）。

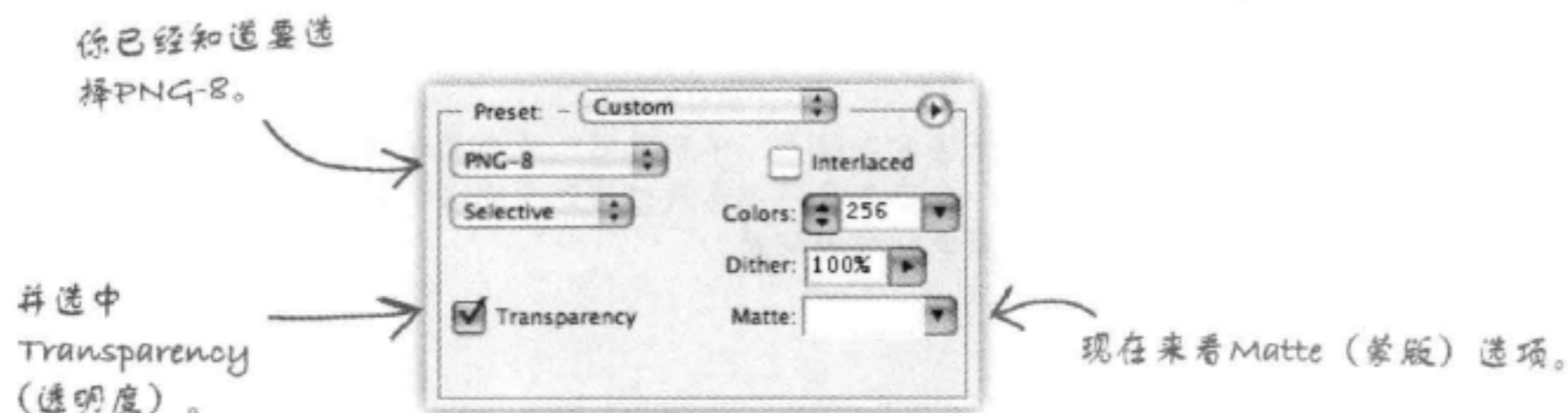
下面来看如果选中Transparency（透明度）并保存会得到什么。好一些……不过，logo中字母周围的那些白色“光晕”是怎么回事？

之所以会有光晕，这是因为照片编辑应用会创建一个“matte”（蒙版），根据背景颜色柔化文本边缘。不过，应用在这个logo上时，它假设是根据一个白色背景柔化文本边缘。

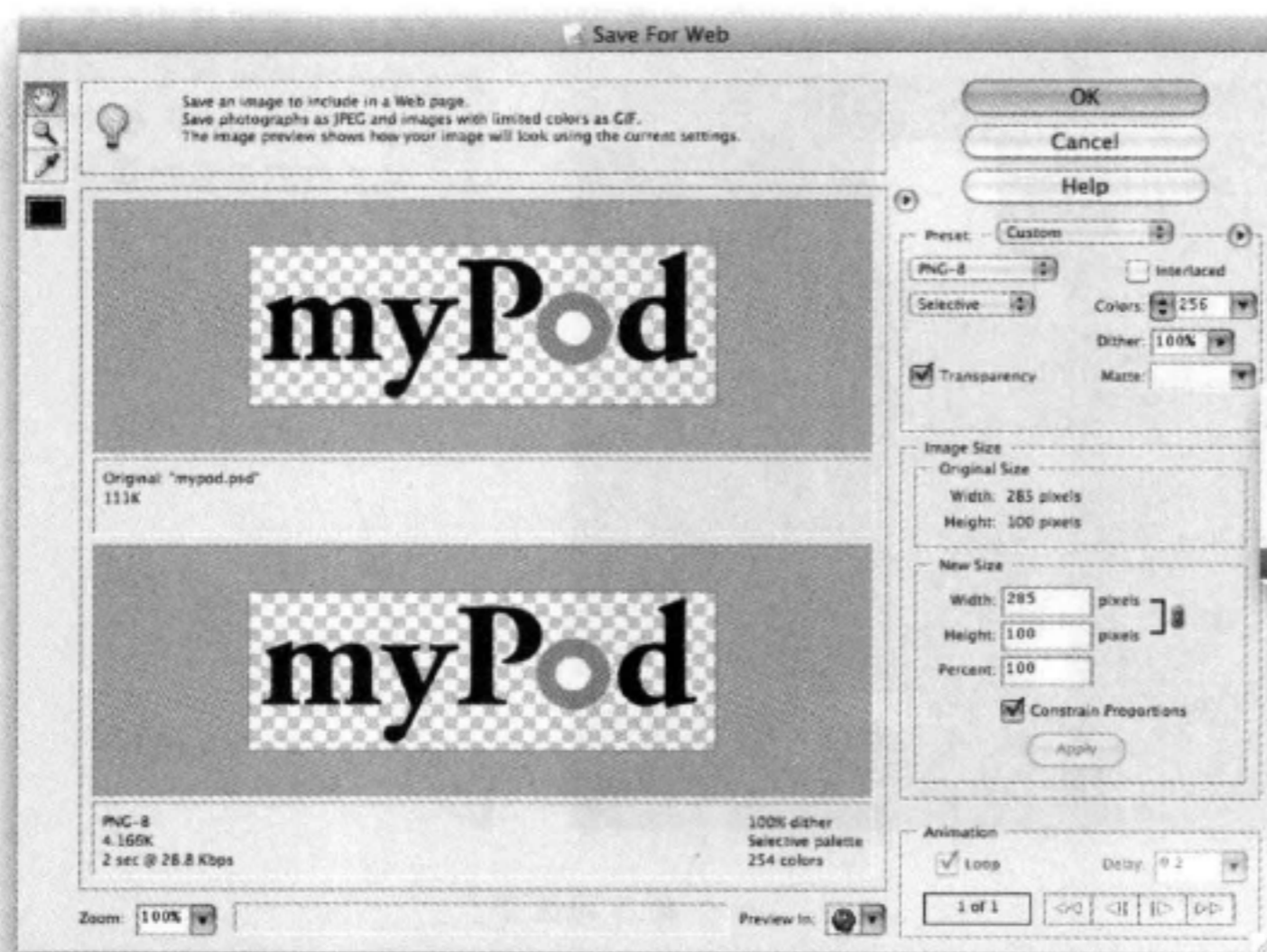
哈，这才是我们想要的，看起来很棒。在这个版本中，我们告诉Photoshop Elements使用绿色背景在文本周围创建蒙版。怎么做到的？下面就来告诉你。

保存透明PNG

你知道我们想要一个PNG版本的透明logo，另外还知道我们需要使用一个蒙版避免文本周围出现光晕。下面来分析“Save for Web”（保存为Web格式）对话框中的PNG面板。



Matte (蒙版) 选项允许你为文本周围的蒙版选择颜色，我们希望蒙版颜色就是Web页面的背景色。



背景色是什么？

等一下，Web页面背景色到底是什么？

还记得myPod “index.html”文件中的成品CSS吗？这个CSS就是用来将页面的背景色设置为淡绿色，从这里可以得到颜色：

```
<style type="text/css">
```

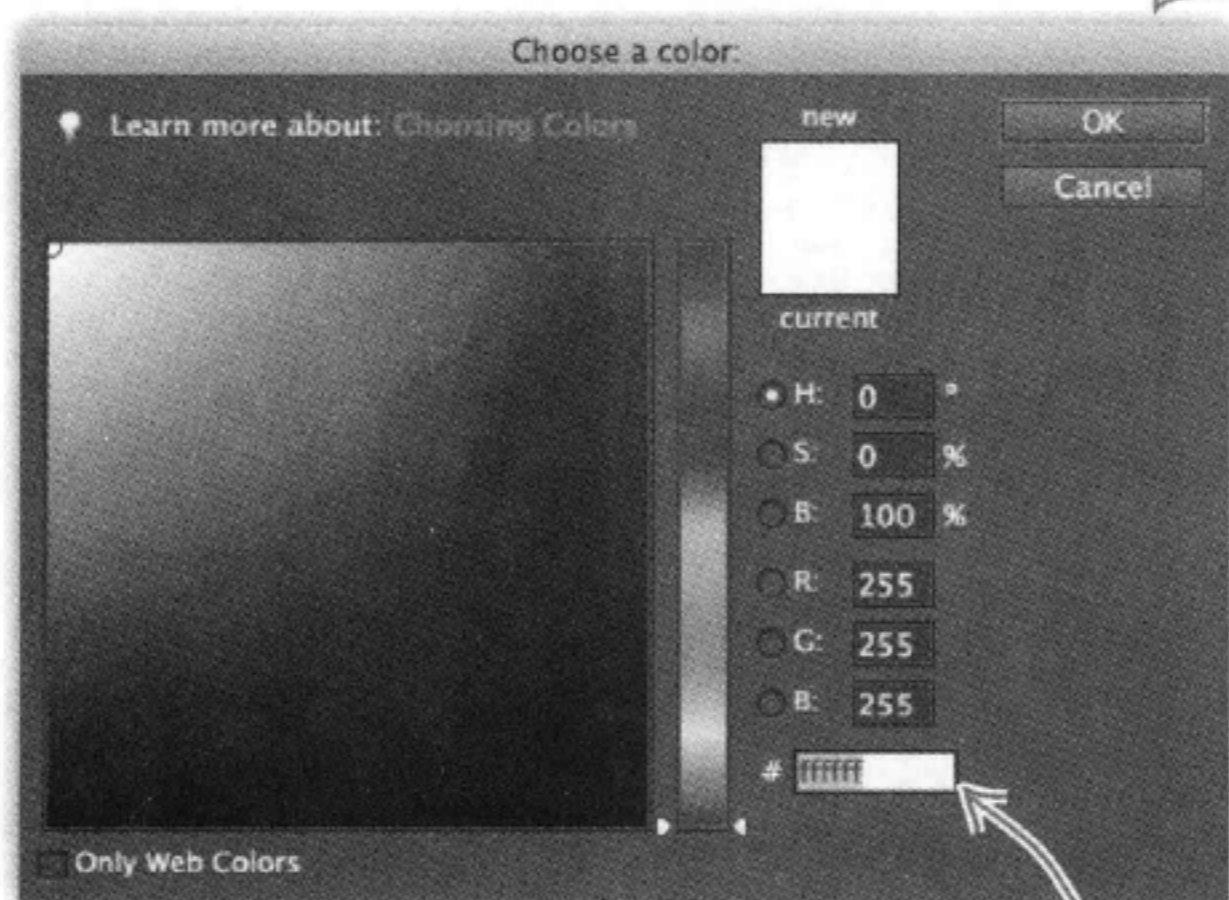
```
  body { background-color: #eaf3da; }  
</style>
```

这里就是背景色。

什么？你说这是淡绿色，不可能吧？现在先记住我们说的，后面几章还会再来讨论这个内容，并解释有关颜色的种种问题。

设置蒙版颜色

单击Matte（蒙版）下拉菜单并选择“Other…”（其他）菜单项时，Photoshop Elements会打开Color Picker（颜色选取工具）对话框。

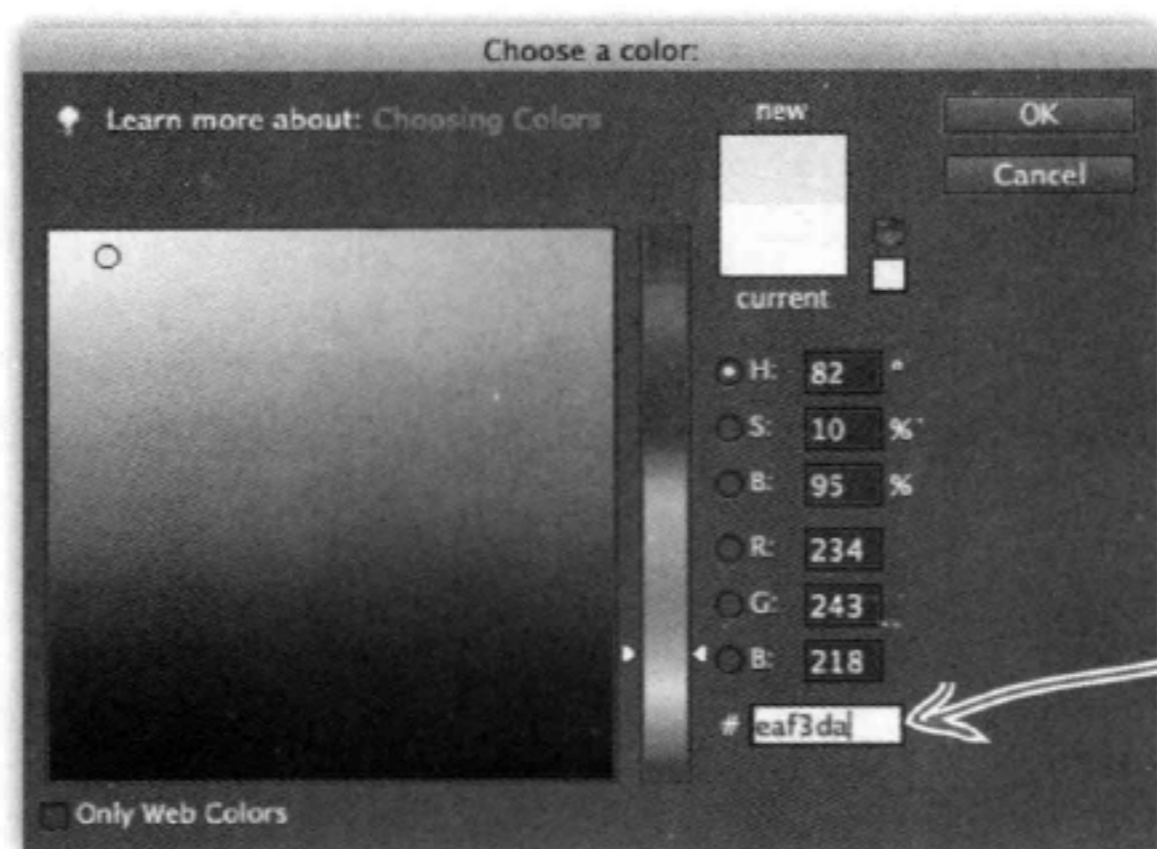


颜色选取工具提供了很多不同方法来选择蒙版颜色。我们希望把它设置为Web页面的背景色，而且我们已经知道页面背景色是eaf3da……

.....输入到这里。

设置蒙版颜色（续）

把颜色“eaf3da”输入到“颜色选取工具”对话框。你会看到颜色变成了myPod页面的背景色。

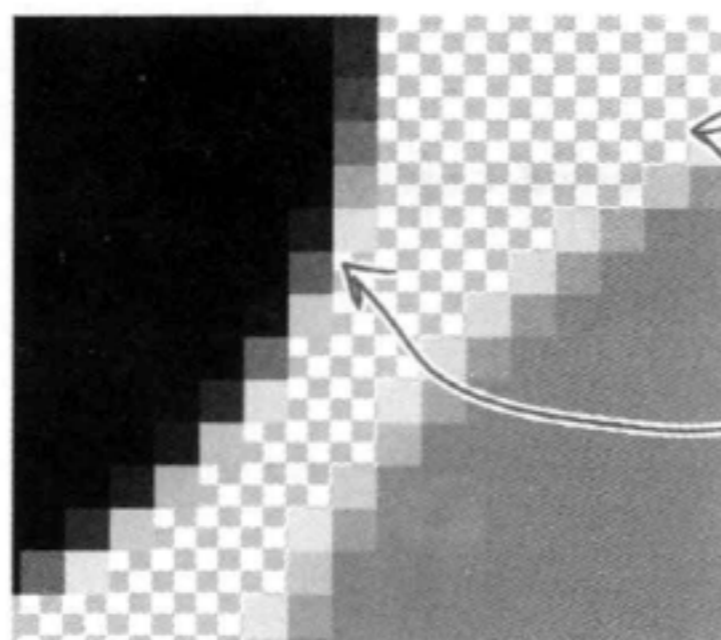


在这里输入这些字母。这个输入框专门设计用来输入Web格式的颜色。这些字母可以是大写，也可以是小写。大小写没有影响。

一旦将颜色输入到颜色选取工具中，就单击OK按钮，这个修改将应用到logo。

查看有蒙版的logo

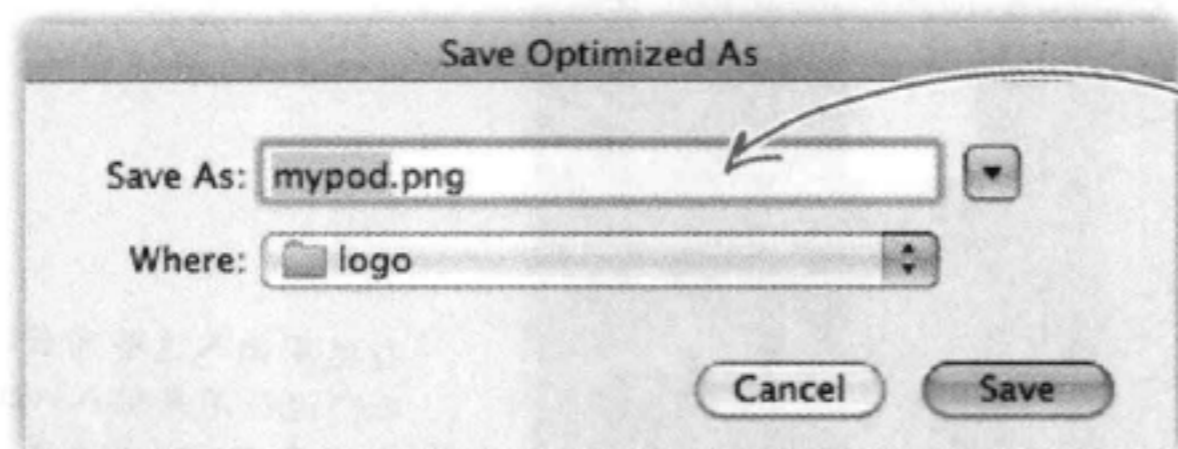
现在再在预览窗口中仔细查看这个logo。你会看到，Photoshop Elements已经在生硬的边线周围加上一个淡绿色的蒙版，当myPod logo在Web页面中显示时，这个蒙版会让logo中的文本外观更柔和、更漂亮。



现在，仔细查看这个logo，你会看到蒙版与myPod Web页面的淡绿色背景一致。

保存logo

OK, 你已经在“Save for Web”（保存为Web格式）对话框中完成了所需的全部调整, 下面单击OK按钮, 把这个图像保存为“mypod.png”。



Elements会自动将你的文件扩展名改为“.png”。把这个图像保存为“mypod.png”，放在“logo”文件夹中。

把logo增加到myPod Web页面

现在你要做的就是把这个logo增加到myPod Web页面。我们把它增加到最上面, 让它出现在网站描述和iPod图像之上。这样一来, 用户访问你的myPod页面上, 第一眼就能看到它。

```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>
    <p>
      
    </p>

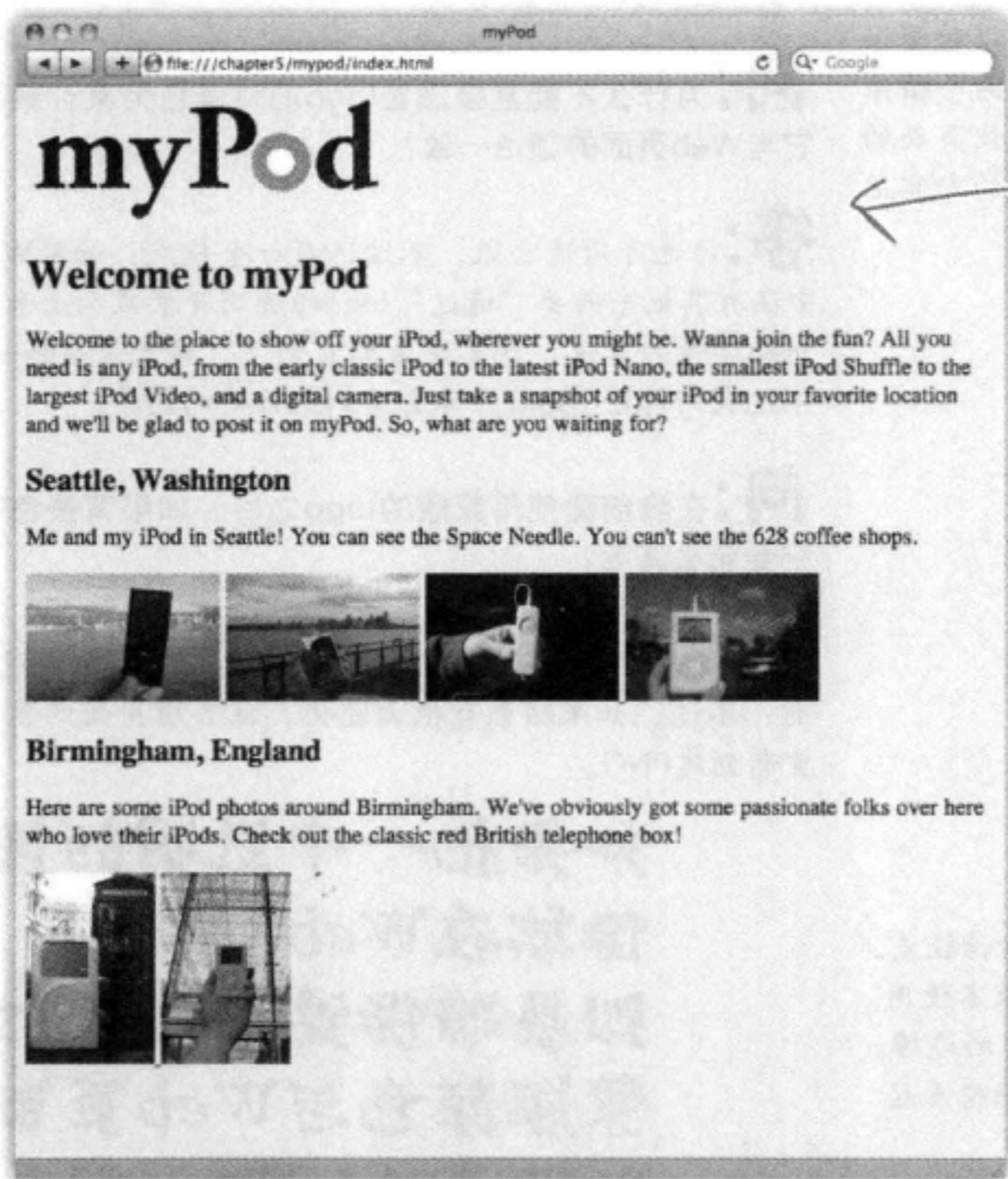
    <h1>Welcome to myPod</h1>
    .
    .
    .
  </body>
</html>
```

把这个logo图像增加到myPod Web页面的最上面。要记住, logo要使用正确的相对路径(在“logo”文件夹中), 另外要增加一个alt属性来描述图像。

← 这里是“index.html” HTML的其余部分……

最后再来试一试

来做个测试吧！在浏览器中重新加载这个页面，看看你的myPod透明PNG logo效果怎么样。



不错啊，之前的努力没有白费。现在你的myPod Web页面上有了一个漂亮的logo。

干得漂亮。这个logo看起来很棒。你构建了一个一流的myPod网站！



there are no Dumb Questions

问:我真的需要了解所有这些关于图像格式的知识才能写出好的Web页面吗?

答:不。即使没有图像,你也完全可以构建出一流的Web页面。不过,图像是Web中很重要的一部分,所以对图像如何工作有所了解肯定会有帮助。有时仅仅是一两个图像就会使页面大不相同,也许一个好页面和一个最佳页面的差距就在于这一、两个图像。关于图像需要了解的内容很多,不过学起来并不难。

问:为什么需要柔化文本的边缘?

答:请看下面两个myPod logo的版本:



可以看到,上面的版本边缘很生硬,有锯齿,可读性较差。这是计算机屏幕上显示文本的默认方式。第二个版本使用了一种称为反锯齿(anti-aliasing)的技术来柔化它的边缘。经过反锯齿处理的文字在计算机屏幕上更可读,眼睛看起来也更舒服些。

问:什么情况下会用到蒙版呢?

答:反锯齿处理可以相对于背景色柔化边缘。如果把上一个问题中logo的第二个版本放在一个彩色背景中,则你会看到它会有白色边缘。Photoshop Elements中的Matte(蒙版)允许你指定要文本背景的颜色,所以柔化文本时,会根据这个颜色来完成柔化。

问:这种技术只适用于文本吗?

答:不,它还适用于图形中可能导致“锯齿”的所有线条。查看myPod logo中的圆,它也使用了蒙版。

问:为什么不能直接设置logo的背景色为某种颜色,让它与Web页面的颜色一致?

答:你也可以这么做,不过这有一个缺点:如果Web页面中还有其他东西要“透过”logo的透明背景显示出来,倘若logo背景为单色,就无法看到它们了。这里没有给出例子,不过我们讨论CSS时,你会看到这方面的例子。

问:在我创建使用蒙版的logo之后,如果背景色有改变则会怎么样呢?

答:如果背景色只是稍有调整,那么可能注意不到任何差别;不过,如果颜色有很大变化,就必须用新的蒙版颜色重新创建PNG。

如果把一个透明的图像放在Web页面中,则要确保这个图像的蒙版颜色与Web页面的背景色一致。

透明图像可以使用PNG或GIF格式。



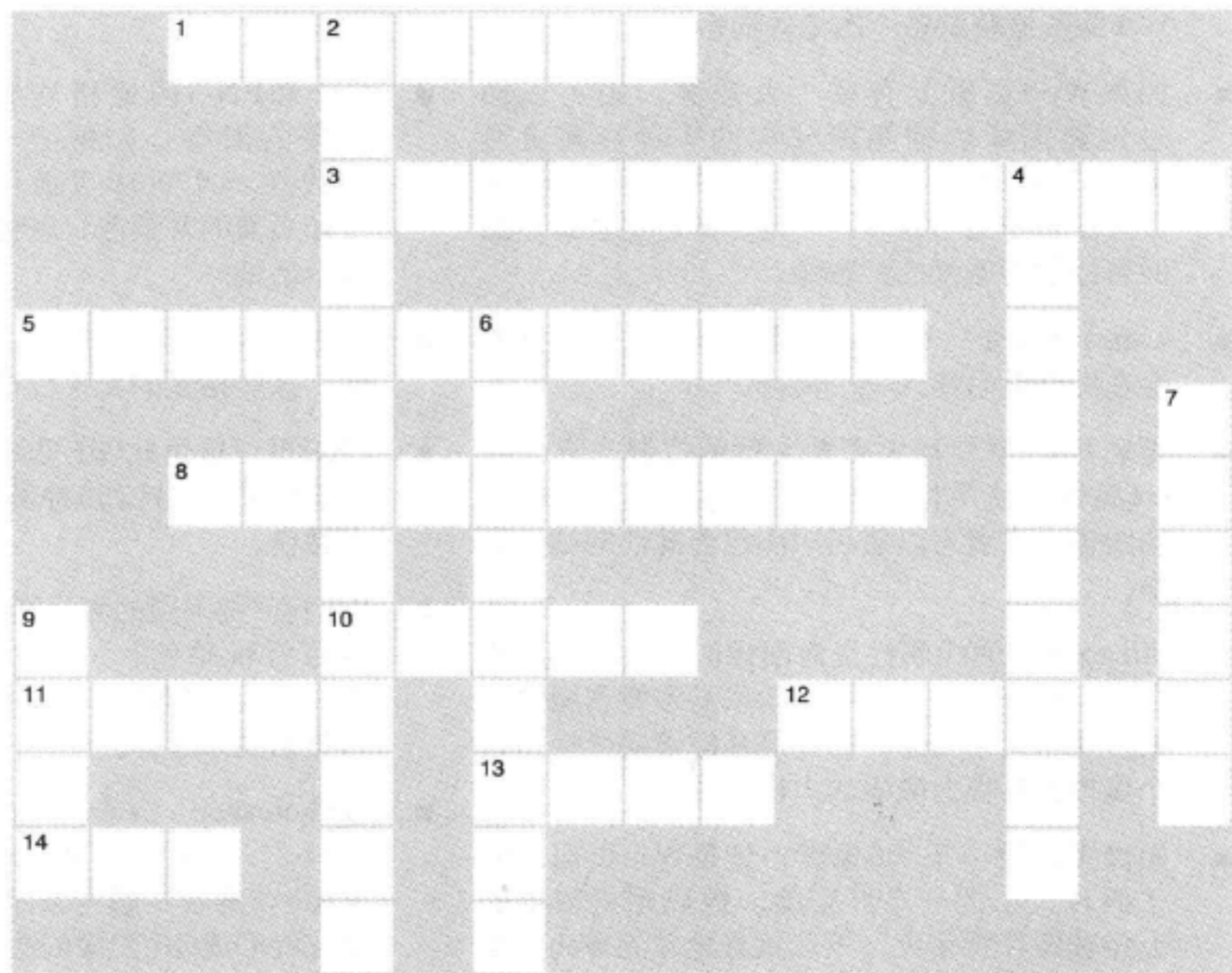
BULLET POINTS

- 使用元素在Web页面中放置图像。
- 浏览器对元素的处理与其他HTML元素稍有不同。读取HTML页面之后，浏览器会从Web服务器获取各个图像并显示。
- 如果Web页面上有多个大图像，则可以通过创建图像的缩略图使你的Web页面更可用，下载也更快，缩略图是一些小图像（大图像的缩小版本），用户单击这些缩略图时可以看到原来的大图像。
- 元素是一个内联元素，这说明浏览器不会在图像前后插入一个换行。
- 要利用src属性指定图像文件的位置。可以在src属性中使用相对路径包含你自己的网站中的图像，或者可以使用URL包含其他网站的图像。
- 元素的alt属性是对图像的一个有意义的描述。在一些浏览器中，如果无法找到图像，就会显示这个描述，另外屏幕阅读器会使用这个属性为有视力障碍的人描述图像。
- 图像宽度要小于800像素，这是Web页面中关于照片大小的一个好经验。数码相机拍摄的大多数照片都太大，不能很好地放在Web页面中，所以需要调整它们的大小。
- 有很多照片编辑应用，Photoshop Elements就是其中之一，可以用来调整图像的大小。还可以使用很多免费的联机工具调整图像大小。可以在网上搜索“free online image editor”（免费联机图像编辑器）。
- 对于浏览器来说太大的图像会使Web页面很难使用，而且下载和显示都很慢。
- JPEG、PNG和GIF是Web浏览器广泛支持的3种图像格式。
- JPEG格式最适合保存照片和其他复杂图像。
- GIF或PNG格式最适合保存logo和其他包含单色、线条或文本的简单图形。
- JPEG图像可以按不同质量压缩，所以可以很好地权衡图像质量和文件大小，来满足你的需要。
- GIF和PNG图像格式允许建立一个有透明背景的图像。如果把一个有透明背景的图像放在一个Web页面中，图像后面的东西（如页面的背景色）就会透过图像的透明部分显示出来。
- GIF和PNG是无损格式，这说明相比于JPEG文件，这些格式的文件往往更大。
- PNG可以提供比GIF更好的透明度控制，而且不像GIF只支持256种颜色，PNG可以支持更多颜色。
- PNG有3种不同的大小选择：PNG-24（支持数百万种颜色）、PNG-16（支持数千种颜色），以及PNG-8（支持256种颜色），可以根据需要来选择。
- 在Photoshop Elements中，使用“Save for Web”（保存为Web格式）对话框中的Matte（蒙版）颜色菜单来选择合适的颜色，柔化PNG或GIF图像的边缘。
- 图像可以用作指向其他Web页面的链接。要由图像创建一个链接，可以使用元素作为<a>元素的内容，将链接放在<a>元素的href属性中。



HTML填字游戏

该让你的右脑歇一歇了，让左脑开动起来：所有单词都与HTML有关，而且都在本章出现过。



横向

1. 利用JPEG，你可以控制这方面。
3. 大多数浏览器会以这种方式获取图像。
5. PNG和GIF有这个特性，而JPEG没有。
8. 用一支铅笔可以画多少英里。
10. Web服务器会为每一个 _____ 发出一个请求。
11. 屏幕上最小的元素。
12. 使用Photoshop Elements对图像做这个处理。
13. 最喜爱的MP3播放器。
14. 更适合包含单色、线条和小文本的图像。

纵向

2. alt属性可以改善 _____。
4. 页面上的小图像。
6. 柔化文本边缘的技术。
7. 图像越大，传输时间就_____。
9. 更适合有连续色调的照片。

* * ? * *

* 哪一种图像格式？ *

答案

祝贺：你当选为今天的“超级图像格式选择能手”。对下面的每个图像，请选择在Web上显示的最佳格式。

JPEG 或 PNG 或 GIF



有大量连续灰色阴影的照片。



只有两种颜色和一些文本，肯定是一个PNG或GIF。不透明？选择PNG的话，可以得到更小的文件。



有很多颜色的照片。肯定是一个JPEG或PNG。如果你想有透明背景，那就得使用PNG。



只是一个简单的黑白图标，这是一个PNG或GIF。如果需要它透明，可能希望对边缘做反锯齿处理，PNG在这方面可能更擅长。



这个图像介于它们之间。它包含大量连续颜色（适合采用JPEG），不过也算是一个几何图形（GIF），另外可能会以某种需要透明度的方式使用这个图像（PNG）。



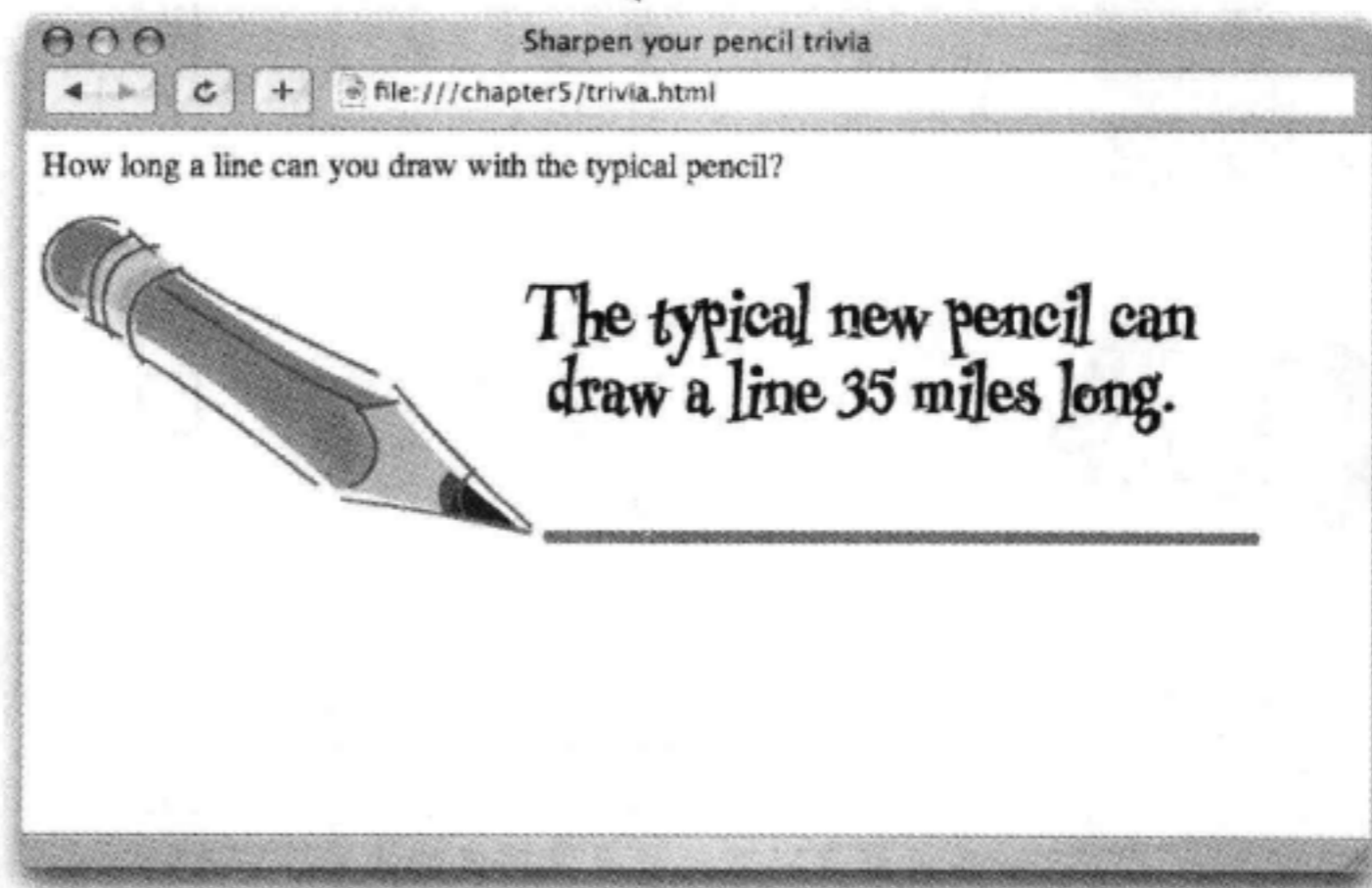
Sharpen your pencil Solution

这里有一个“练练笔”练习，它确实与铅笔有关（哦，还有图像）。这个练习提到这样一个问题：给你一根普通的新铅笔，如果用它画一条连续的线，则把整个铅笔用完，这条线会有多长？

这和图像有什么关系？要找出答案，先写一些HTML。这个问题的答案就包含在URL为<http://wickedlysmart.com/hfhtmlcss/trivia/pencil.png>图像中。你的任务是把图像增加到这个HTML，并找出答案。下面是我们的答案。

```
<html>
  <head>
    <title>Sharpen your pencil trivia</title>
  </head>
  <body>
    <p>How long a line can you draw with the typical pencil?</p>
    <p>
      
    </p>
  </body>
</html>
```

如果把图像放在这里，加载
页面时你就会看到答案。



来源：<http://www.papermate.com>

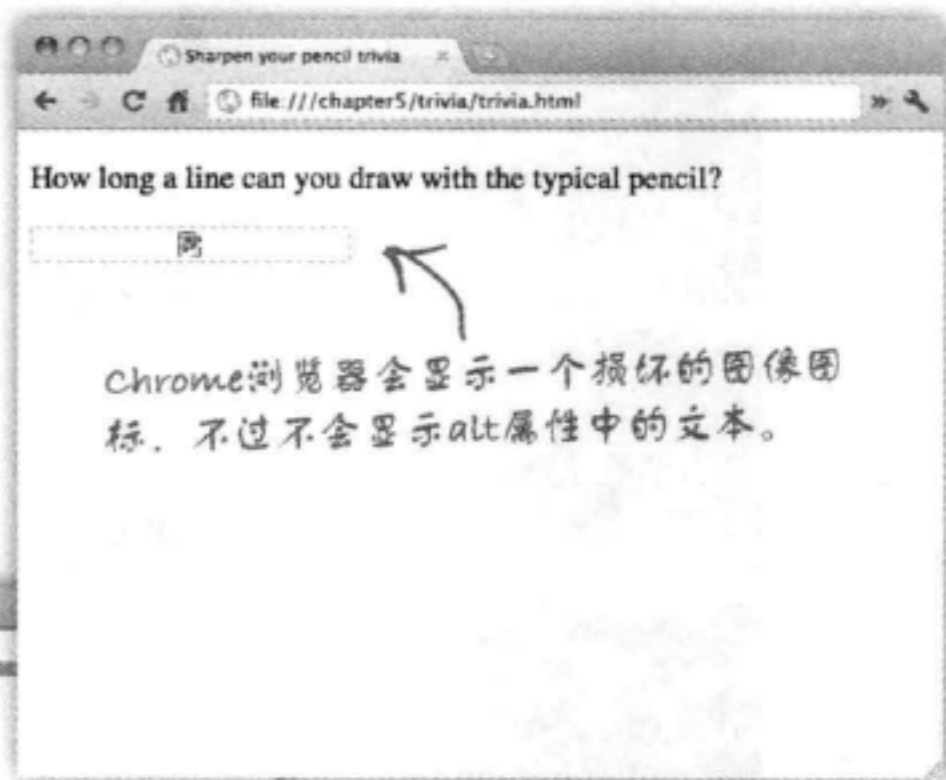


Exercise Solution

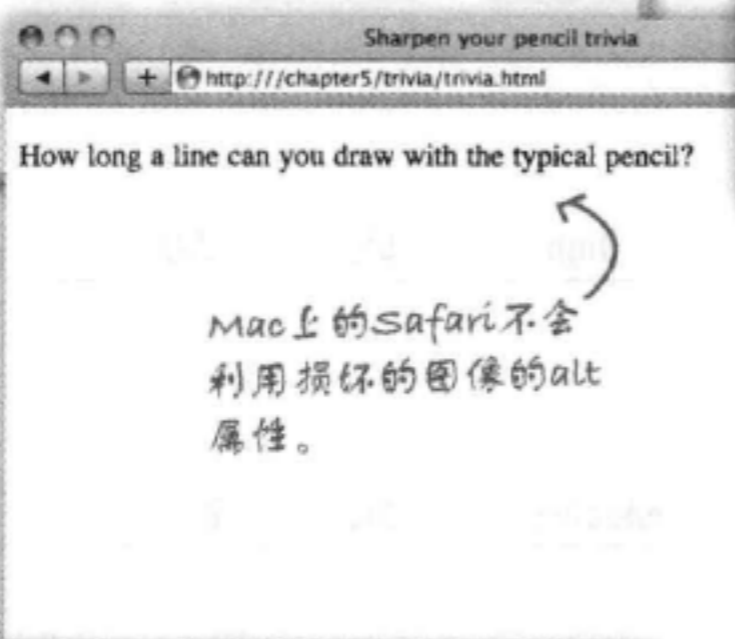
以下是一些不同浏览器中无法正常显示图像时得到的结果。大多数情况下，浏览器都能使用额外的alt属性信息来改善显示。我们为什么关心这个问题呢？毕竟，这是Web页面中存在的一个错误，我们要修复它，对不对？嗯，在真实世界里，情况往往并不完美。有些东西会损坏，互联网连接可能在页面加载过程中断开，或者有视力障碍的用户需要听到图像里有什么，因为他们看不见。



IE会显示一个损坏的图像图标，同时显示图像的alt属性文本。



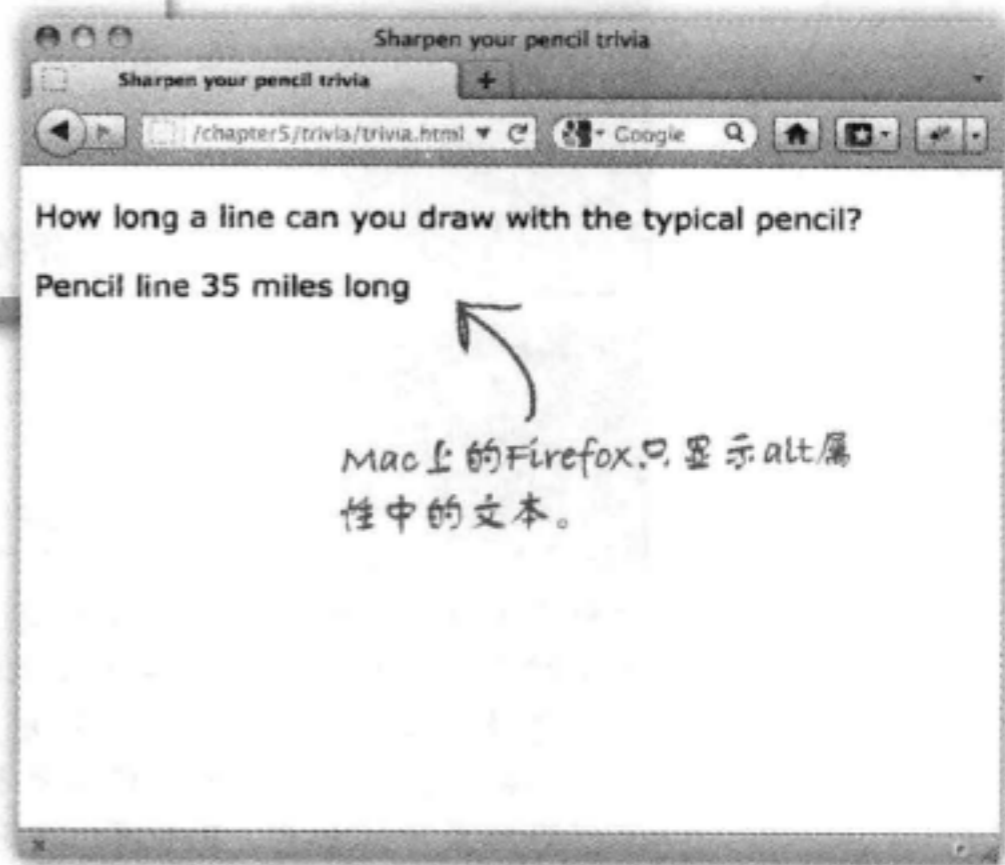
Chrome浏览器会显示一个损坏的图像图标，不过不会显示alt属性中的文本。



Mac上的Safari不会利用损坏的图像的alt属性。



Mac上的Opera会显示alt属性文本。



Mac上的Firefox只显示alt属性中的文本。

哪一种图像格式？

答案

这一次的任务：在Photoshop Elements中打开文件“chapter5/testimage/eye.jpg”。打开“Save for Web”对话框，选择JPEG的各种质量设置（Low、Medium、High等），再试试PNG-24和GIF，填写下面的空格。你会在图像下面的预览窗口中找到这些信息。完成后，确定对于这个图像哪种设置最合适。

注意，取决于你使用软件的哪个版本，得到时间可能有所不同。

注意到了吗？
从JPEG
Maximum
变到Low时，
图像质量在
下降。



格式	质量	大小	时间	优胜者
<u>PNG-24</u>	<u>N/A</u>	<u>32K</u>	<u>13秒</u>	<input type="checkbox"/>
<u>JPEG</u>	<u>Maximum</u>	<u>21K</u>	<u>8秒</u>	<input type="checkbox"/>
<u>JPEG</u>	<u>High</u>	<u>6K</u>	<u>3秒</u>	<input type="checkbox"/>
<u>JPEG</u>	<u>Medium</u>	<u>3K</u>	<u>2秒</u>	<input checked="" type="checkbox"/>
<u>JPEG</u>	<u>Low</u>	<u>2K</u>	<u>1秒</u>	<input type="checkbox"/>
<u>GIF</u>	<u>N/A</u>	<u>22K</u>	<u>9秒</u>	<input type="checkbox"/>

优胜者真的是Medium吗？不一定。这完全取决于你需要什么。如果你想要一个真正高质量的图像，那么你可能要选择Very High。如果希望有最快的网站，则可以使用Low。我们选择Medium的原因是，它是文件大小和图像质量的一个很好的折衷方案。你可能认为Low就足够了，或者可能认为有必要把质量上调到High。所以如何选择是很主观的。不过有一点可以肯定：PNG和GIF不太适合这个图像（这应该并不奇怪）。



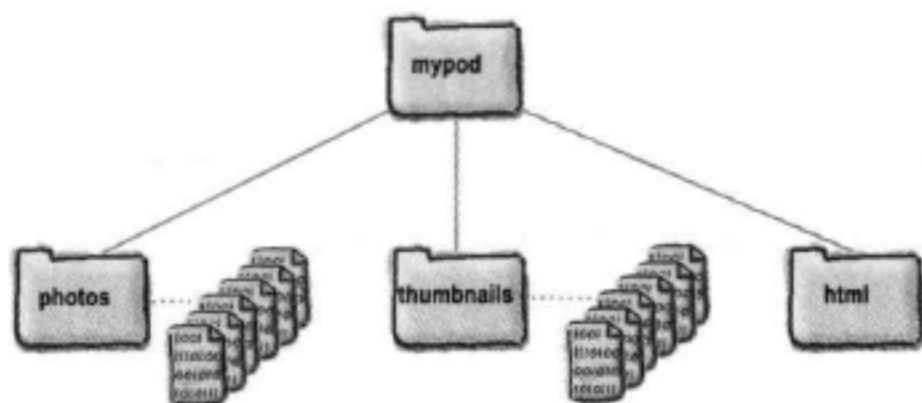
Exercise Solution

如果查看本章示例文件中的“html”文件夹，那么你会发现其中已经提供了所有的照片页面，但有一个页面除外——对应“seattle_downtown.jpg”的页面。在“html”文件夹中创建一个名为“seattle_downtown.html”的页面，做个测试。继续学习下面的内容之前先完成这个工作。下面是我们的答案。

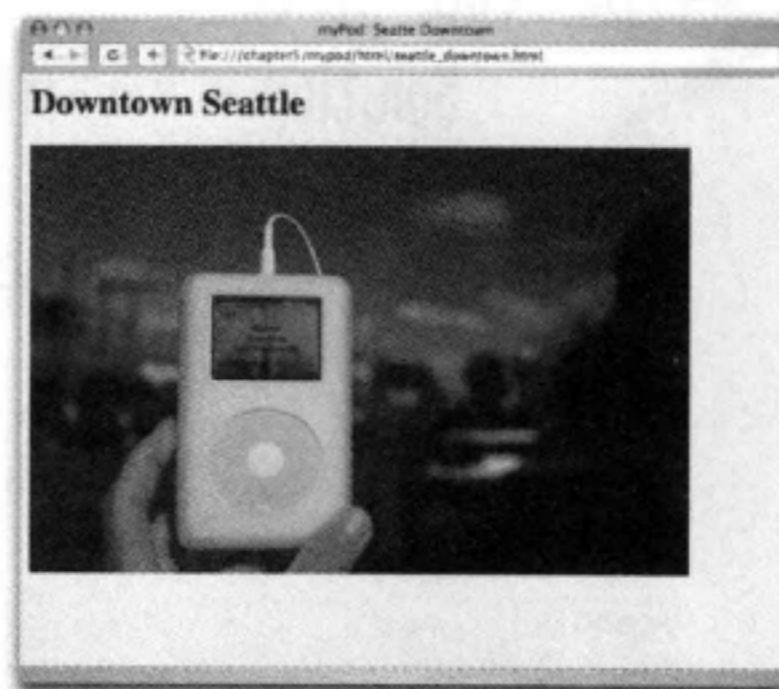
这里给出了HTML，这个文件应当命名为“seattle_downtown.html”。

```
<html>
  <head>
    <title>myPod: Seattle Downtown</title>
    <style type="text/css"> body { background-color: #eaf3da; } </style>
  </head>
  <body>
    <h1>Downtown Seattle</h1>
    <p>
      
    </p>
  </body>
</html>
```

这个文件应当放在“mypod”下的“html”文件夹里。

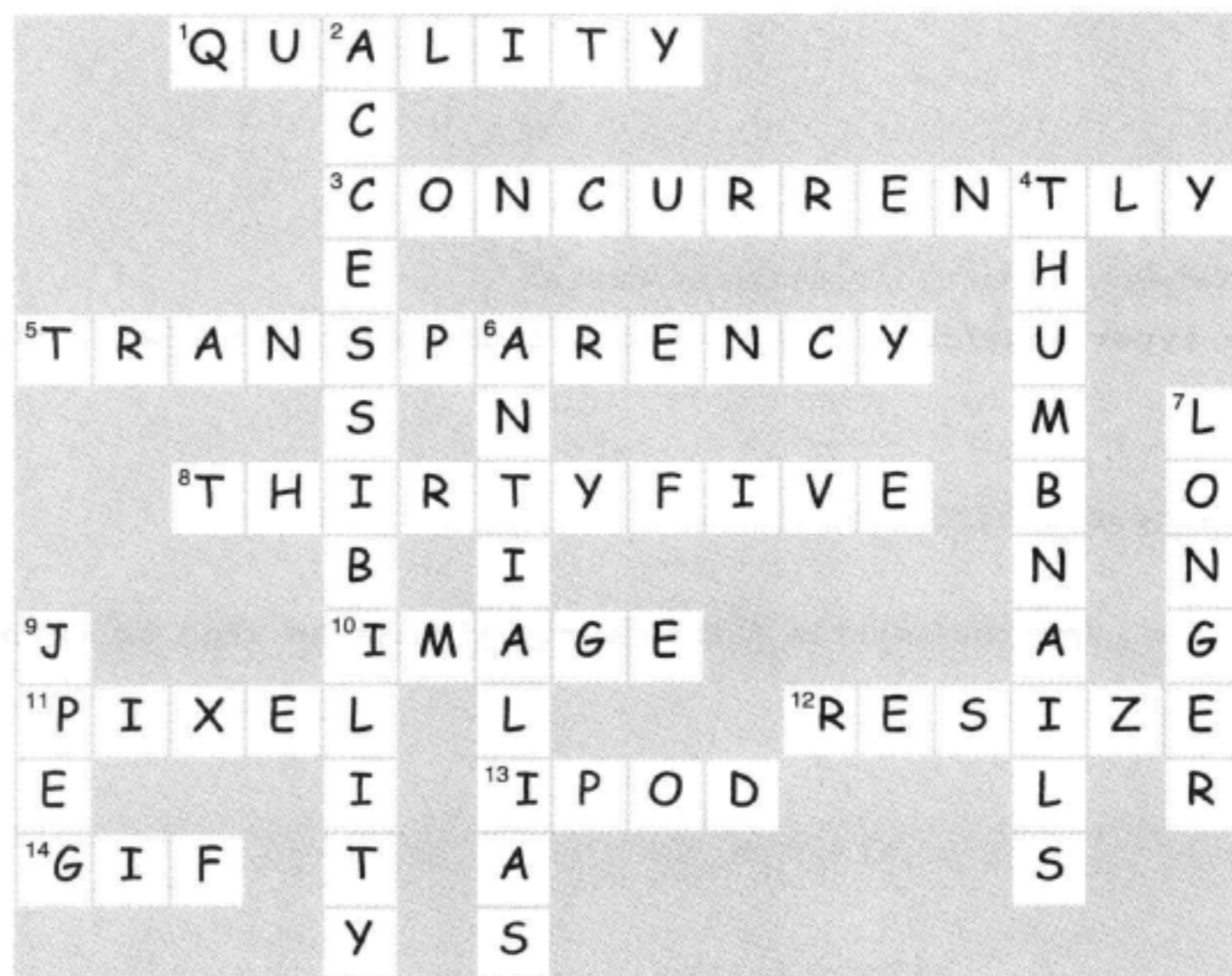


这是测试结果。





HTML填字游戏答案



Sharpen your pencil

Solution 可以如下将图像“seattle.jpg”增加到文件“index.html”中。

```
<h2>Seattle, Washington</h2>
```

```
<p>
```

```
Me and my iPod in Seattle! You can see rain clouds and the
Space Needle. You can't see the 628 coffee shops.
```

```
</p>
```

```
<p>
```

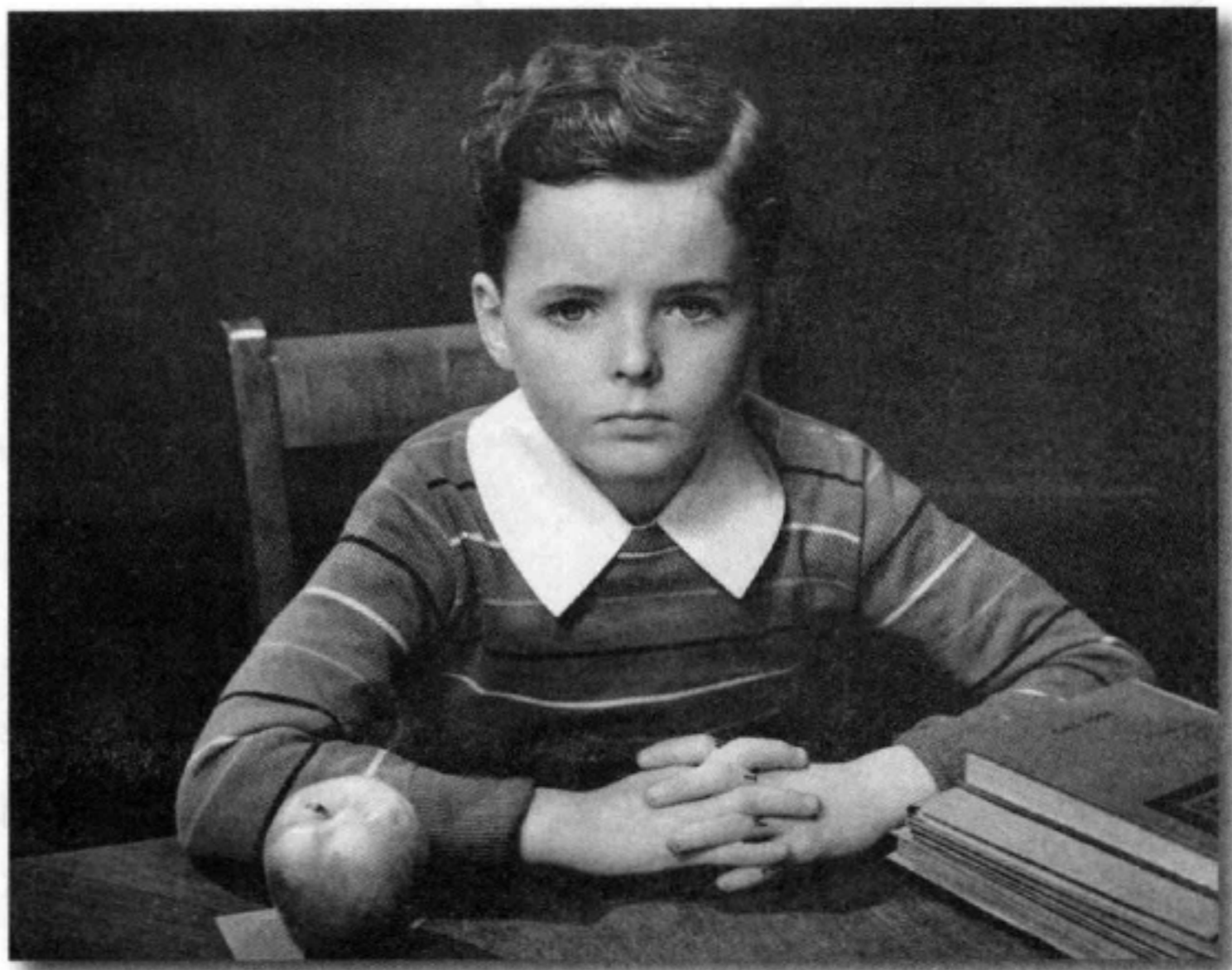
```

```

```
</p>
```

6 标准及其他

严肃的HTML



关于HTML还要知道些什么？很不错，现在你已经掌握了HTML。实际上，是不是该转向CSS了？来学习如何让这些乏味的标记看起来更漂亮。在此之前，我们需要确保你已经掌握了足够的HTML知识，有能力应对激烈的竞争。不要误解我们的意思，没错，你一直在写一流的HTML，不过还需要了解额外的一些知识，才能真正建立“工业标准”的HTML。还要利用这个机会确保你使用的是最新、最棒的HTML标准，也就是HTML5。保证了这一点，就可以确保HTML在所有浏览器上（至少是你关心的浏览器上）能有更一致的显示，不仅如此，还可以在最新的iPhone、iPod之类的设备上（选择你最喜欢的设备）很好地显示。你的页面会加载得更快，可以很好地结合CSS，能够随着标准发展稳步走向未来。准备好了吗？这一章会让你从一个Web工匠变成真正的Web高手。



Jim: 做好准备，应对重大时刻？

Frank: 是啊，你知道的，要保证它完全合法，而且能适应HTML5。

Jim: 我们的HTML很不错了……看这里，看它在浏览器里的表现。很漂亮，不是吗？

Joe: 对，我也这么想……他只是想给我们找点事干。

Frank: 说真的，虽然我不想承认，不过我觉得这一次老板说的没错。

Jim, Joe: 嗯？

Frank: 到目前为止，我们一直都忽略了这样一个事实，在这个领域有很多标准。甚至HTML也有很多版本，比如HTML 4.01，还有现在的HTML5。对我们来说，是不是该做的都做了？确实能保证符合HTML5吗？

Joe: 喂，这意味着还要做更多工作。我们已经受够了。实际上，页面看起来够好的了，我甚至在一些更新的设备上都已经做过测试。

Frank: 也许是吧，不过我的意思是说，我觉得这样反而能帮助我们减少将来的工作。

Joe: 哦? 真的吗? 怎么会这样?

Frank: 嗯, 如果能确保HTML与当前最新标准保持一致, 在标准发展过程中, 我们就不用做那么多的修改。另外, 我们还要确保所有其他方面没有问题; 你知道的, 这包括我们的语法等方面。有太多不同的浏览器, 而且这些浏览器还有不同的版本, 所以如果HTML中有错误, 我们的页面在不同浏览器上显示时肯定会不一样。用CSS为HTML增加表现效果时, 如果HTML有问题, 则这些差别甚至会更严重。

Joe: 这么说, 通过确保遵循“标准”, 为客户显示页面时就能少遇到麻烦, 不能正确显示页面的问题也会少得多, 是这样吗?

Frank: 没错。

Jim: 如果这样能让客户少在半夜打来求助电话, 听上去这个主意也还不坏。

Joe: 好的, 那么从哪里开始呢? 难道我们现在没有遵循标准吗? 我们的HTML有什么问题?

Frank: 可能没什么问题, 不过老板希望用最新的HTML5, 所以我们需要明确正在使用HTML的哪个版本, 如果不是HTML5, 我们就得改为HTML5。另外, 如果我们做到了这一点, 使用CSS时日子也会好过得多。



如果你编写了正确的HTML, 各种浏览器都能一致地显示你的页面, 不过如果HTML中有错误, 或者你在HTML中做了非标准的事情, 则在不同浏览器中页面通常会有不同的显示。你觉得为什么会出现这种情况?

HTML简史



HTML 1.0 ~ 2.0

那可真是很早以前了。关于HTML需要知道的东西并不多，甚至在汽车后备箱里都能全部放下。页面不好看，不过至少已经支持超文本。没有人关心表现，几乎Web上的每一个人都有他们自己的“主页”。那时甚至课桌上的铅笔、曲别针和即时贴数量都被认为是“Web内容”（你肯定认为我们是在开玩笑）。



HTML 3

那时正经历漫长、残酷的“浏览器战争”。Netscape 和Microsoft 都在试图争霸世界。毕竟，谁统治了浏览器，谁就能统治宇宙，是不是？

陷入这场争端的受害者是Web开发人员。在这场战争中，每个浏览器公司都在不断增加自己的专用扩展包，试图保持领先，军备竞赛就此展开。谁能坚持到最后？不仅如此，那个时候，通常你必须写两个单独的Web页面：一个用于Netscape浏览器，另一个用于Internet Explorer。真是很不好。



HTML 4

哈……浏览器大战终于结束了，万维网协会（World Wide Web Consortium）来解救我们了，我们亲切地称它为W3C。他们的计划是：创建一个唯一的HTML“标准”，让这个世界恢复平静。

这个计划的关键是什么？将HTML的结构和表现分解到两种语言，一种语言用于实现结构（HTML），另一种语言用于表现（CSS），另外要求获得最大利益的浏览器制造商采用这些标准。

不过他们的计划奏效了吗？

嗯，差不多吧……只是稍有些调整（见HTML 4.01）。

1989

1991

1995

1998

从这一章开始，我们的目标是写正确的HTML5。与往常一样，这个世界在不断变化，所以我们还会介绍今后的走向。



HTML 4.01

这段日子过得很惬意，HTML 4.01在1999年闪亮登场，成为接下来十年中HTML的“必备”版本。

4.01与4.0并没有太大变化，只是在一些方面做了些修补。不过，与HTML的早期阶段相比（那时可真是艰辛，就好像必须光脚在6英尺深的雪地里艰难跋涉，而且都是上坡路），HTML 4.01可以让我们晚上睡个好觉，因为我们知道，几乎所有浏览器（至少你关心的那些浏览器）都能很好地显示你的内容。



XHTML 1.0

正当我们感觉安逸的时候，一个新兴事物开始引起所有人的注意。这个新兴事物就是XML。实际上，它让HTML开始心烦意乱，它们两个终于在“拉郎配式”的婚姻中不情愿地结合在一起，XHTML 1.0就此诞生。

XHTML承诺，由于它的严格，再加上它提供的一些新方法，只要遵循这个标准，Web的所有争端将就此平息。

唯一的问题是，^{大多数}人们很讨厌XHTML。他们并不想要一种编写Web页面的新方法，只是希望改进HTML 4.01已有的特性。Web开发人员对HTML的灵活性更感兴趣，而不是XHTML的严格性。另外，这些开发人员越来越希望把时间用来创建更像是应用的Web页面，而不只是文档（稍后还会讨论Web应用）……



HTML5

当然，由于没有得到大家的支持，这场婚姻的结局并不好，很快被HTML的新版本所取代，这就是HTML5。由于它支持HTML 4.01标准的大部分特性，而且还提供了一些新特性，可以体现Web新的发展，所以HTML5得到了大家的欢迎，这正是开发人员一直想要的。另外，基于一些新特性，如支持类似博客的元素、新的视频和图形功能，以及一大堆用来构建Web应用的功能，HTML5注定成为大家公认的标准。

说实话，HTML和XML的分手还是让很多人感到诧异，曾在一段时间让人们很困惑HTML5到底是什么。不过现在已经都清楚了，所以继续读下去，看看HTML5对你来说意味着什么，你将如何加入这个欢乐的世界。

1999

2001

2009

2012

????

未来会怎样？我们会不会在会飞的汽车里工作，能不能只吃营养药片代替进餐？继续读下去，自己来找答案吧。



浏览器闪亮登场

本周访谈：
为什么你这么关心我用哪个HTML版本？

Head First: 很高兴能请到你，浏览器。你知道的，“HTML版本”已经成为一个热门的话题。这是怎么回事？毕竟，你是一个Web浏览器。我把HTML交给你，你尽你所能显示这些HTML就可以了。

浏览器: 当个浏览器可真不容易……有那多么的Web页面，而且很多还是用HTML的老版本写的，或者标记里可能还有错误。就像你说的，不管怎样，我的任务就是努力显示每一个页面。

Head First: 那么，有什么问题呢？看来你做得不错啊。

浏览器: 有些情况下是这样，不过，你有没有在不同的浏览器上查看你的页面？如果你使用比较老的浏览器，或者页面中有不正确的标记，就可能遭遇这样一种尴尬的情况：你的页面在一个浏览器上看起来很棒，但在另一个浏览器上可就不怎么样了。

Head First: 真的吗？为什么会这样？难道你做的不一样吗？

浏览器: 我们确实在做同样的事情，前提是我们显示的是正确、最新的页面。前面我说过，如果你把有问题的页面交给我，情况就很难讲了。告诉你原因吧：我们浏览器都听HTML规范的，它会告诉我们如何显示正确的HTML，但是如果是不正确的HTML，我们就不清楚该怎么做了。所以你会在不同的浏览器上看到截然不同的表现。

Head First: 噢。那么有什么解决办法吗？我们确实希望页面能很好地显示。

浏览器: 很容易。提前告诉我你在使用哪个版本的HTML。你会惊奇地发现，居然有那么多的页面没有这样做。另外要确保你的页面没有任何错误，你知道

的，就是标记不匹配之类的错误。

Head First: 我们怎么告诉你在用哪个版本呢？特别是现在我们都要转向HTML5了。

浏览器: 嗯，HTML5确实让问题简单一些了。

Head First: 是吗？新版本的HTML有什么帮助？我原来以为另一个版本只会让问题更困难呢。

浏览器: 嗯，语言的新版本确实会带来“成长的烦恼”，因为所有人都得跟上最新的标准。不过HTML5则有所简化，你能更容易地告诉我你在使用哪个版本的HTML。HTML5标准还明确指出了Web页面中可能出现的很多错误，所以所有浏览器都能更一致地处理这些错误。

Head First: 噢，那么这是不是意味着，我们写HTML时不用再担心犯错误了，是吗？

浏览器: 不是这样的！尽管我们能更好地处理错误，但是并不表示你就能偷懒。你还是需要保证你的页面与标准一致，而且没有错误。如果不能做到，你可能会在不同浏览器上得到不一致的结果，另外别忘了还有那些移动设备上的浏览器。

Head First: 回答我前面的问题吧，怎么告诉你我们在用什么版本？

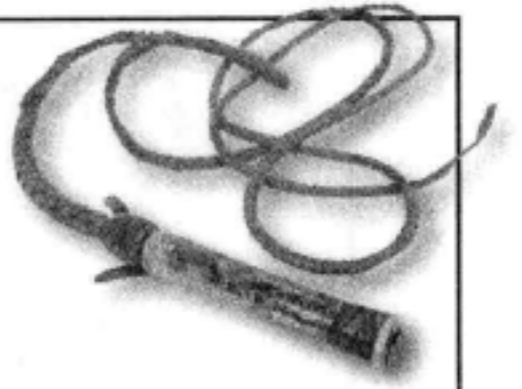
浏览器: 嗯，原来这可麻烦呢……

Head First: 嘿，拜托，观众们都可可是大忙人，我们没时间了，快一点儿好吗！

浏览器: 好的，好的，你可以用一个doctype告诉我你正在使用的HTML版本。只是这么一点点“标记”，要放在HTML文件的最上面。好了，既然时间到了，你自己试试看吧！



HTML 考古



我们挖到了一些古老的HTML 4.01和XHTML 1.1页面。这些页面使用了一个doctype，放在HTML页面的最上面，告诉浏览器所使用的HTML版本。我们已经把doctype挑出来了，你可以好好分析一下。请看下面的doctype……

这里为浏览器指定这个页面的文档类型。

这表示这个<html>是页面中的根(第一个)元素。

这表示HTML 4.01标准是公共可用的。

这部分表示我们在使用HTML 4.01版本，另外这个HTML标记用英语编写。

可以把这些写在一行上，或者如果你愿意，也可以像我们一样加一个回车。不过要确保只能在加引号的部分之间按Enter键。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

注意这不是一个HTML元素。在开始的“<”后面有一个“!”，这说明它与众不同。

这指向一个文件，标识这个特定的标准。

类似于HTML DOCTYPE，这是一个公共文档类型。

这仍然是一个HTML版本(结合XML的版本)。

这个对应XHTML 1.1版本。

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

关于XHTML的更多内容，请参考附录。

另外它有一个URL指向XHTML 1.1的定义。

Sharpen your pencil

我们认为与其告诉你HTML5的doctype定义，还不如让自己动手把它找出来。再来看下面的HTML 4.01 doctype定义：

记住，这是“html”的doctype。

这表示这个标准是公共可用的。

这部分说明我们在使用HTML 4.01版本，而且这个标记用英语编写。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

这指向标识这个标准的一个文件。

要记住，doctype定义要放在HTML文件的最上面，告诉浏览器你的文档类型是什么，在这里就是HTML 4.01。通过使用doctype，浏览器可以更准确地解释和显示页面。

所以，下面要发挥你的推理能力了，你认为HTML5的doctype定义是什么样？把它写在下面（下一页具体介绍时你可以翻回来对照检查一下，不过现在不要偷看答案）：

.....

.....

.....

.....

.....

↑
你的答案写在这里。

新的、改进的HTML5 doctype

OK，准备好了吗。下面给出HTML5 doctype：

```
<!doctype html>
```

← 只有一行，别漏了。

↑
真是很简单！

你的“练练笔”练习的答案是这样吗？你肯定会说，这个要简单得多。哇，现在每次需要一个doctype时甚至不用查找都能记住。

← 有些人记不住老的doctype，所以把它当作纹身纹在手掌上，对于他们，我们只能表示深深的同情。

请等一下，这并没有告诉浏览器版本呀？版本号呢？是不是有印刷错误？

问得好。不，这可不是印刷错误，下面来解释为什么：你很清楚原来的doctype相当复杂，版本号和丑陋的语法混杂在一起。不过，随着HTML5的到来，doctype已经得到简化，所以现在我们要做的就是告诉浏览器我们在使用“HTML”，不用再担心特定的版本号或语言，也不用指向某个标准。

真的能这样吗？我们可以只指定“HTML”，而不再需要其他的部分，真的吗？浏览器不需要其他信息吗？嗯，实际上，当浏览器看到：

```
<!doctype html>
```

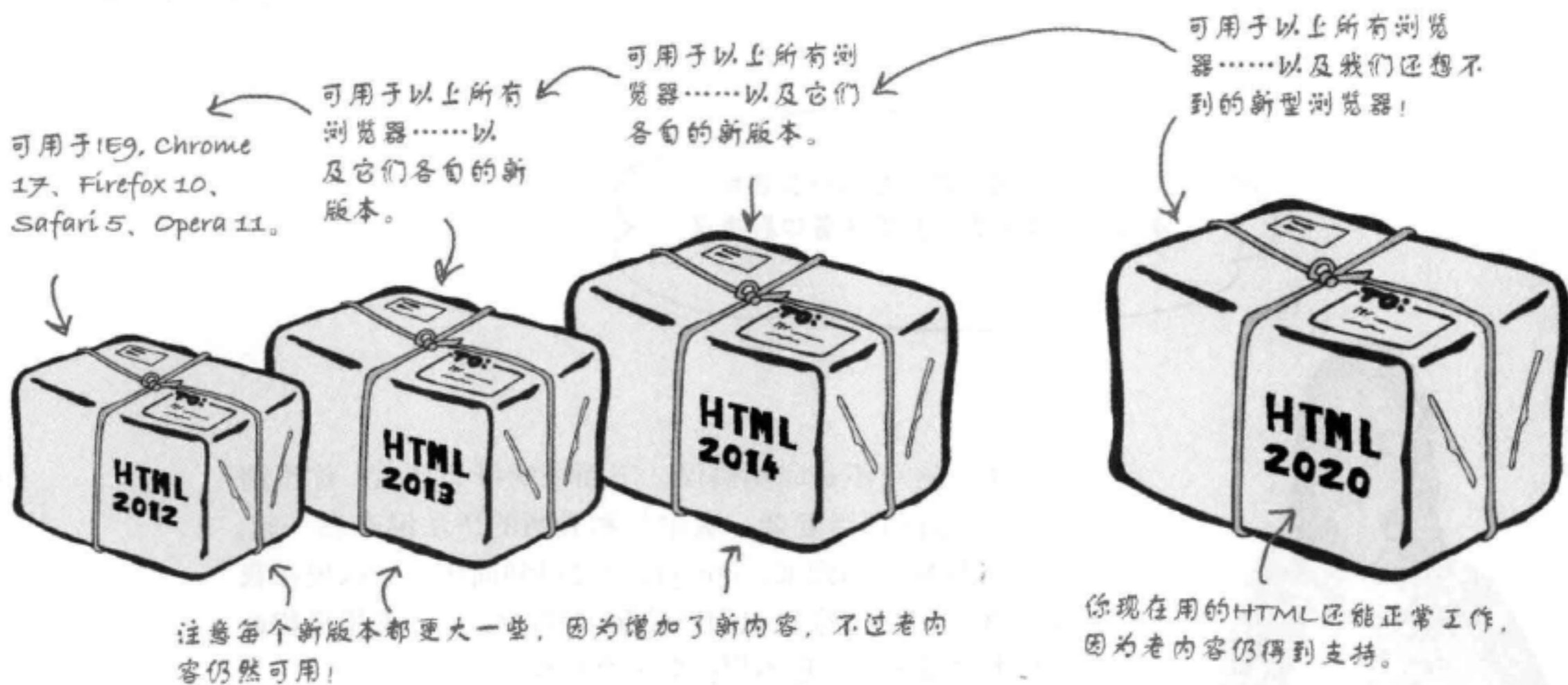
它就认为你在使用标准HTML。不再考虑版本号，也不考虑标准在什么位置；实际上，HTML标准将变成一个“活的标准”，这意味着它会根据需要继续发展和变化，不过不再有固定的版本号。现在你可能会想，“活的标准到底是什么意思？它是怎么工作的呢？”下一页就会告诉你……



HTML, 新的“活标准”

你没听错……HTML不会再有版本6、7、8……制订标准的人已经把这个规范变成了一个活的标准，它会随着技术的发展形成相应文档。所以，不再有版本号。你甚至可以不再把它叫做HTML5，因为从现在开始它只是“HTML”。

现在，你可能在考虑这在实际中如何工作。毕竟，如果规范在不断变化，这对于可怜的浏览器来说意味着什么？另外，对于你们Web开发人员又意味着什么？这里的关键是向后兼容性。向后兼容性（backwards compatibility）表示我们可以继续向HTML增加新的内容，浏览器（最终）会支持这个新内容，不过它们还会继续支持原来的内容。所以，你今天写的HTML页面将会继续正常工作，甚至以后增加了新的特性之后也仍然能很好地工作。



there are no Dumb Questions

问：如果规范明天改变了怎么办？我该怎么办？

答：如果你今天编写了稳定可靠的HTML，即使明天规范改变了，比如增加了一个新元素，你还能像以往一样使用你的HTML。是否用这个新元素要由你来决定。

如果规范对你之前的工作有改变，比如改变了一个元素或属性的工作方式，那么浏览器会继续支持你使用的老方式，同时支持新的方式。这正是“向后兼容性”的含义。现在，如果对现有特性的改变尽可能少，而且你（作为一个Web开发人员）始终跟进最新规范，保持与规范一致，随着规范的变化修改你的页面，这当然很好，不过关键是在规范不断变化的同时，你的HTML还会继续正常工作。

问：规范到底是什么？

答：规范就是一个文档，指定了HTML标准是什么。也就是说，HTML中有哪些元素和属性等。这个文档由万维网协会（World Wide Web Consortium，简称为W3C）维护，不过任何人都可以为它做出贡献，都可以就这个标准如何发展进言。

OK, 我想现在我们明白了。下面在休闲室文件里加上这个doctype, 把这些页面更新到HTML5。



增加文档类型定义

说的够多了, 下面在HTML里加上这个doctype。

这就是doctype行。把它作为第一行增加到“lounge.html”文件中。

可以写为DOCTYPE或doctype, 这两个都可以。

```

<!doctype html>
<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing
      <a href="elixir.html">elixirs</a>, conversation and
      maybe a game or two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided; BYOWS (Bring
      your own Web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown
      Webville. If you need help finding us, check out our
      <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>

```


测试doctype

对“chapter6/lounge”文件夹中的“lounge.html”文件完成上述修改，然后在你的浏览器中加载这个页面。



哇，没有任何差别。嗯，我们并不希望有差别，因为doctype所做的就是让浏览器明确知道你在使用HTML5。



Exercise

还要为“directions.html”和“elixir.html”文件增加这个doctype。来好好做个测试。就像“lounge.html”一样，你看不到任何差别（不过今天晚上你能睡得更好一点）。



HTML5闪亮登场

本周访谈：
HTML5有什么过人之处？

Head First: HTML5, 你是HTML“最新最棒”的版本, 所有的人都在为你欢呼, 不过我们的读者想知道你到底有什么过人之处。

HTML5: 首先, 我增加了一堆新元素, 还有一些新属性。

Head First: 看来我们还没有使用任何新元素, 是吧?

HTML5: 你现在使用的所有元素都是HTML5标准的一部分, 所以你使用的就是HTML5元素。不过, 你说的没错, 确实还没有用到任何新元素……

Head First: 为什么不用? 难道我们不该尽快使用所有这些最新的元素吗?

HTML5: 没必要。应该记得(第3章中提到过): 一定要物尽其用, 使用最合适的元素完成它最擅长的工作! 这些最新的元素有一些特定的工作。其中一些要用来为页面增加更多结构和含义。比如我的新<article>元素, 它就专门用于博客帖子和新闻报道之类的内容。

Head First: 我们也可以在第3章Tony的博客中使用这些新元素, 对吗?

HTML5: 那是当然了……我相信你以后肯定会增加这些新元素。

Head First: 我想读者们可能还很困惑, 因为他们在这本书里学的是HTML, 是不是应该学HTML5?

HTML5: 不! HTML5只是演进路上的下一步, 你学到的所有东西在HTML5中都是一样的。HTML5只是增加了一些新东西。实际上, 你不应该再说“HTML5”了。我就是HTML的最后一个版本, 所以请叫我HTML。现在如果还讲HTML5只会让人糊涂。

Head First: 请等一下, 关于HTML5做了那么多宣传, 你真的建议我们把你叫做HTML吗?

HTML5: 没错。你已经知道, 我是一个活的标准, 而版本号是死的。嗯, 确切地讲, 我是一个活的HTML标准, 而不是HTML5。

Head First: 我懂了。我们的读者确实应该继续学习HTML5, 噢, 抱歉, 应该是继续学习HTML, 他们目前为止学到的所有知识都很重要, 而且接下来要学习的所有新知识都是最新、最棒的HTML技术。

HTML5: 完全正确。

Head First: 不过, 我还得问一句。我听说你的一些新内容是用来构建Web应用的。这是什么意思?

HTML5: 最重要的是, 我不再只是用来建立Web页面, 现在我被设计为可以建立成熟的Web应用。

Head First: 这有什么区别?

HTML5: Web页面主要是静态页面。页面上有一些图像和一堆链接, 可能还有一些漂亮的效果, 比如菜单。不过, 总的来讲, 页面还只是用来阅读。另一方面, Web应用则用来交互, 用来完成具体工作。它们就像你的桌面计算机上的应用一样, 只使用Web应用就能在Web上完成工作。

Head First: 能不能给我一个例子?

HTML5: 社交媒体应用、地图应用、游戏……这样的例子不计其数。

Head First: 在HTML5之前这些做不到吗?

HTML5: 嗯, 有些可以做到, 不过构建这些应用所需的很多特性在HTML5中才第一次成为标准。在此之前, 即使某些特性存在, 也实属偶然。

Head First: 不过, 我不认为我们会在这本书里构建任何应用。

HTML5: 确实如此, 但你可以看看《Head First HTML5 Programming》。那本书就是在介绍如何用我来构建Web应用!

Head First: 我们肯定会! 感谢你接受我们的采访, HTML5。



OK, 这可真不赖, 现在我们告诉浏览器我们在使用标准HTML。

Jim: 太好了, 真是很容易。不过, 也有一点虎头蛇尾吧……我们把这个doctype放在文件最上面, 告诉浏览器我们的页面是HTML, 不过然后呢? 什么都没有改变。

Frank: 对, 你看不到任何改变, 但它确实告诉了浏览器我们在使用标准HTML。浏览器可以最大程度地使用这个信息。另外, 老板希望我们写完全合法的HTML, 为此我们也需要这个doctype。

Jim: OK, 仅此而已吗? 然后呢? 我们现在是在写工业标准HTML吗?

Frank: 据我所知是这样, 不过现在开始有点意思了。现在只有一样东西可能导致失败, 这就是我们可能在页面中引入的错误。假设我们忘记加一个结束标记会怎么样? 标记名敲错了呢?

Jim: 噢, 没错, 嗯, 如果确实存在这些错误, 则难道我们不知道吗?

Frank: 不一定, 浏览器看到错误时, 它很善于绕开这些错误。

Jim: 如果我把人员都召集起来, 则我们一起对整个页面完成审查, 这样好不好?

Frank: 可能不需要……现在有很多工具可以帮助你验证页面。

Jim: 验证?

Frank: 对, 就是通查页面, 确保所有标记都是合法的。确保与标准保持一致。这有些像HTML的一个拼写检查工具。

Jim: 听上去是个不错的主意。从哪里可以得到这些工具呢?

Frank: W3C制订标准的那些人提供了一个验证工具, 这是免费的。

Jim: 太好了, 下面就来完成验证吧。

认识W3C验证工具

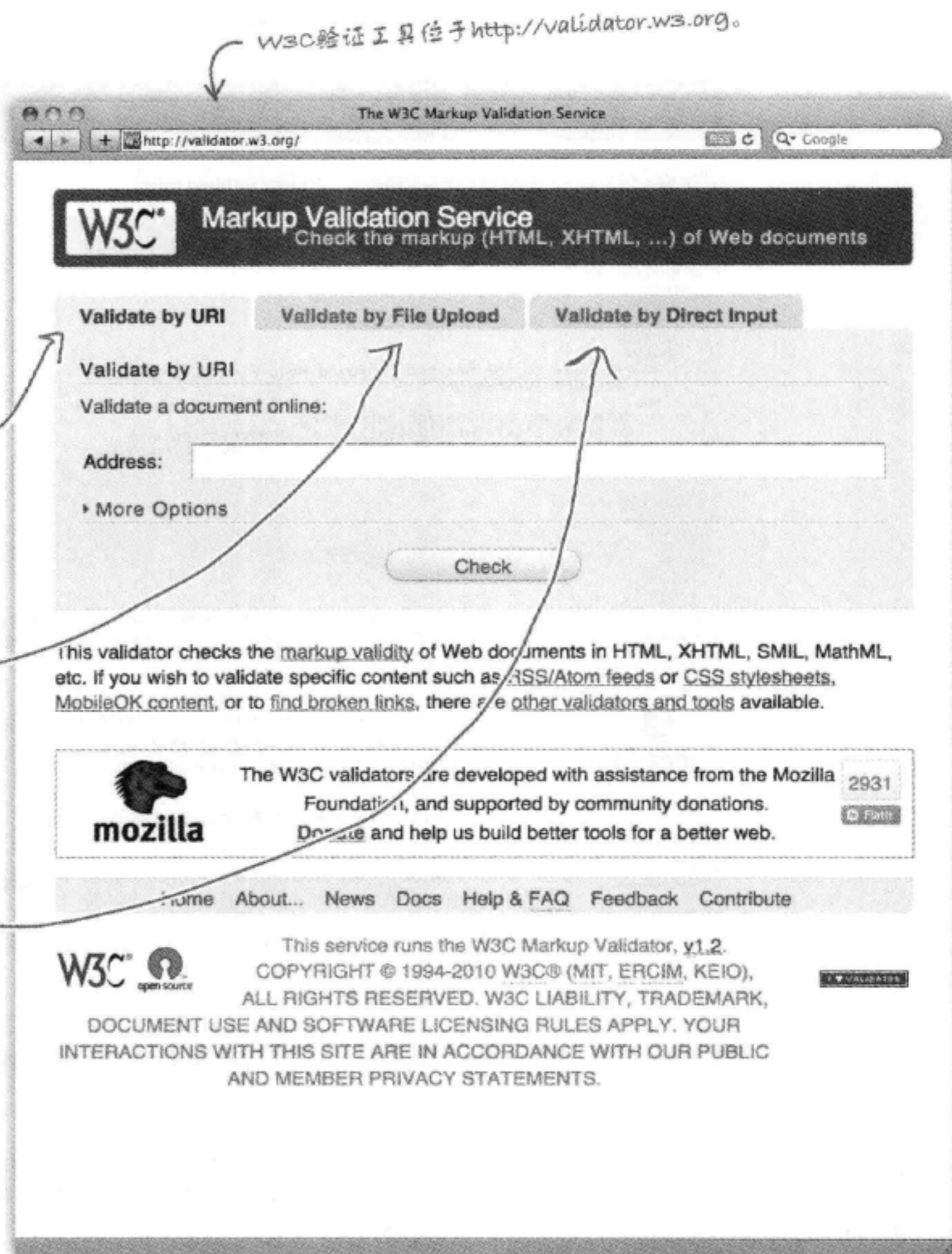
下面来看W3C验证工具，用它来验证我们的休闲室文件。要得到这个验证工具，只需将浏览器指向http://validator.w3.org。

检查HTML有3种方法：

(1) 如果你的页面在Web上，则可以在这里输入URL，单击Check（检查）按钮，这个验证服务会获取你的HTML，并完成检查。

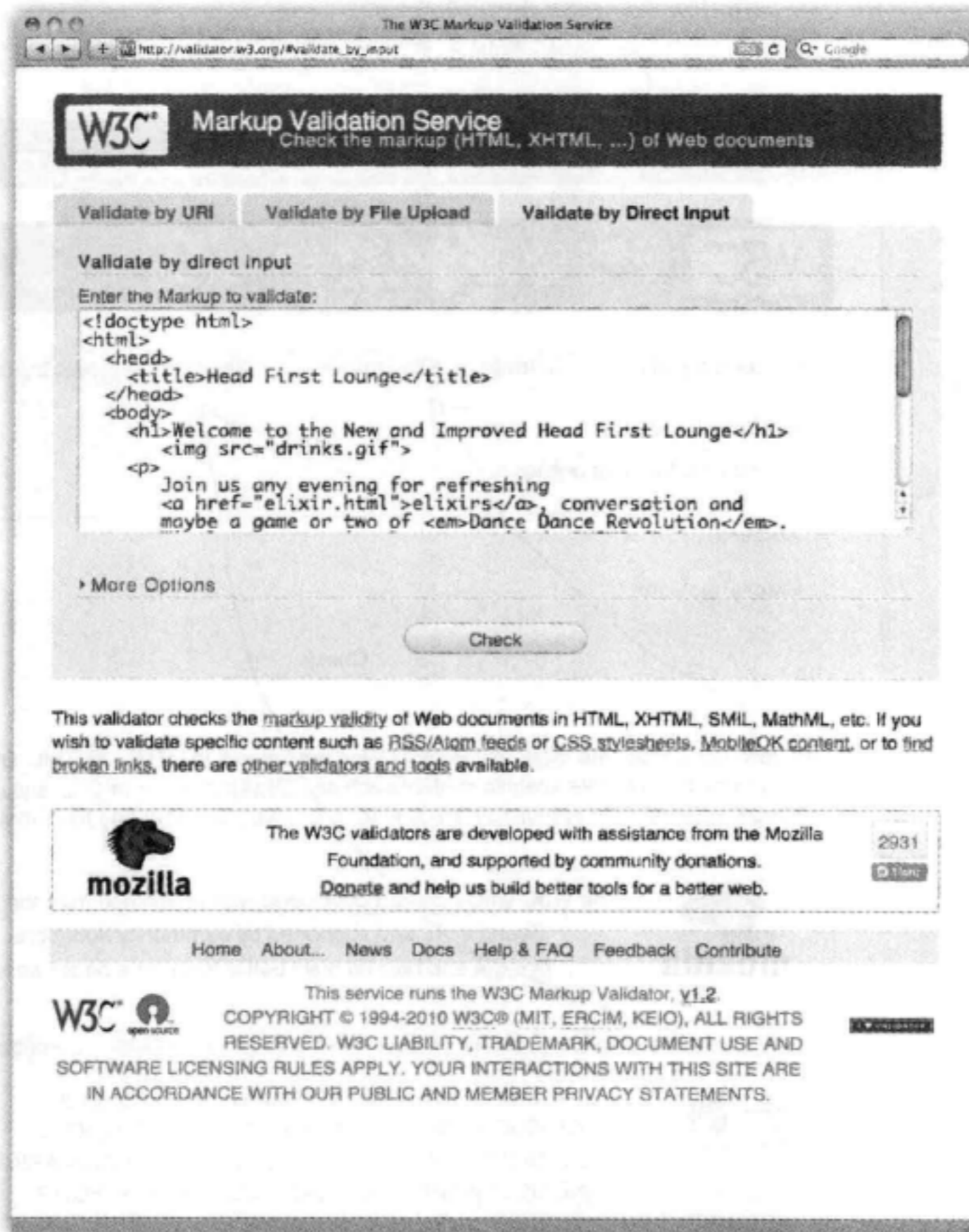
(2) 可以选择第2个标签页，从你的计算机上传一个文件。选择这个文件后，单击Check（检查），浏览器会上传这个页面，由验证服务检查。

(3) 或者，也可以选择第3个标签页，将你的HTML复制粘贴到这个标签页上的表单中。然后单击Check（检查），验证服务就会检查你的HTML。



验证Head First休闲室

我们要使用第3个标签页“Validate by Direct Input”（直接输入验证）来验证“lounge.html”文件。这说明，需要从“lounge.html”将HTML复制粘贴到这个标签页上的表单中，跟着我们一起来试一试……



这里我们使用了第3种方法。单击“validate by Direct Input”（直接输入验证）标签页，粘贴“lounge.html”的代码，现在这个HTML代码最上面有HTML5的doctype。我们已经等待这个时刻很久了……这个Web页面合法吗？绝对没有问题吗？单击Check（并翻开下一页）按钮，看看结果……

如果方法1或2更方便，也完全可以使这两种方法。

唉呀，我们遇到一个问题……

页面上出现红色肯定不是好事。看来这个页面没有通过验证。我们最好来看一下……

验证失败了。
看起来这里有一个错误。

The screenshot shows the W3C Markup Validation Service interface. At the top, it says "W3C Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below that, there are tabs for "Jump To: Notes and Potential Issues" and "Validation Output". The main content area displays a message: "Error found while checking this document as HTML5!". Underneath, it shows the "Result: 1 Error, 3 warning(s)" and a "Source" section with the following HTML code:

```

<!doctype html>
<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing
      <a href="elixir.html">elixirs</a>, conversation and
      maybe a game or two of <em>Dance Dance Revolution</em>.
    </p>
  </body>
</html>

```

Below the source code, there are fields for "Encoding: utf-8", "Doctype: HTML5", and "Root Element: html". The "Validation Output" section shows "1 Error" and the specific error message: "Line 8, Column 29: An img element must have an alt attribute, except under certain conditions. For details, consult guidance on providing text alternatives for images." The error is highlighted in the source code as "".

这肯定是错误。



Watch it!

W3C一直在修改这个验证工具。

因为W3C在频繁修改验证工具，你看到的错误消息可能不完全一样。不用担心，即使你没有看到上面的错误，也要跟着我们继续学习，因为后面几页的所有内容都很重要。

还不算太糟糕。看来我们必须在元素中使用alt属性。

修正错误

OK, 看来修正这个错误很简单。只需要在HTML5中为元素增加一个alt属性。下面打开“lounge.html”, 完成修改, 保存, 然后再来尝试验证。

```
<!doctype html>
<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing
      <a href="elixir.html">elixirs</a>, conversation and
      maybe a game or two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided; BYOWS (Bring
      your own Web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown
      Webville. If you need help finding us, check out our
      <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>
```

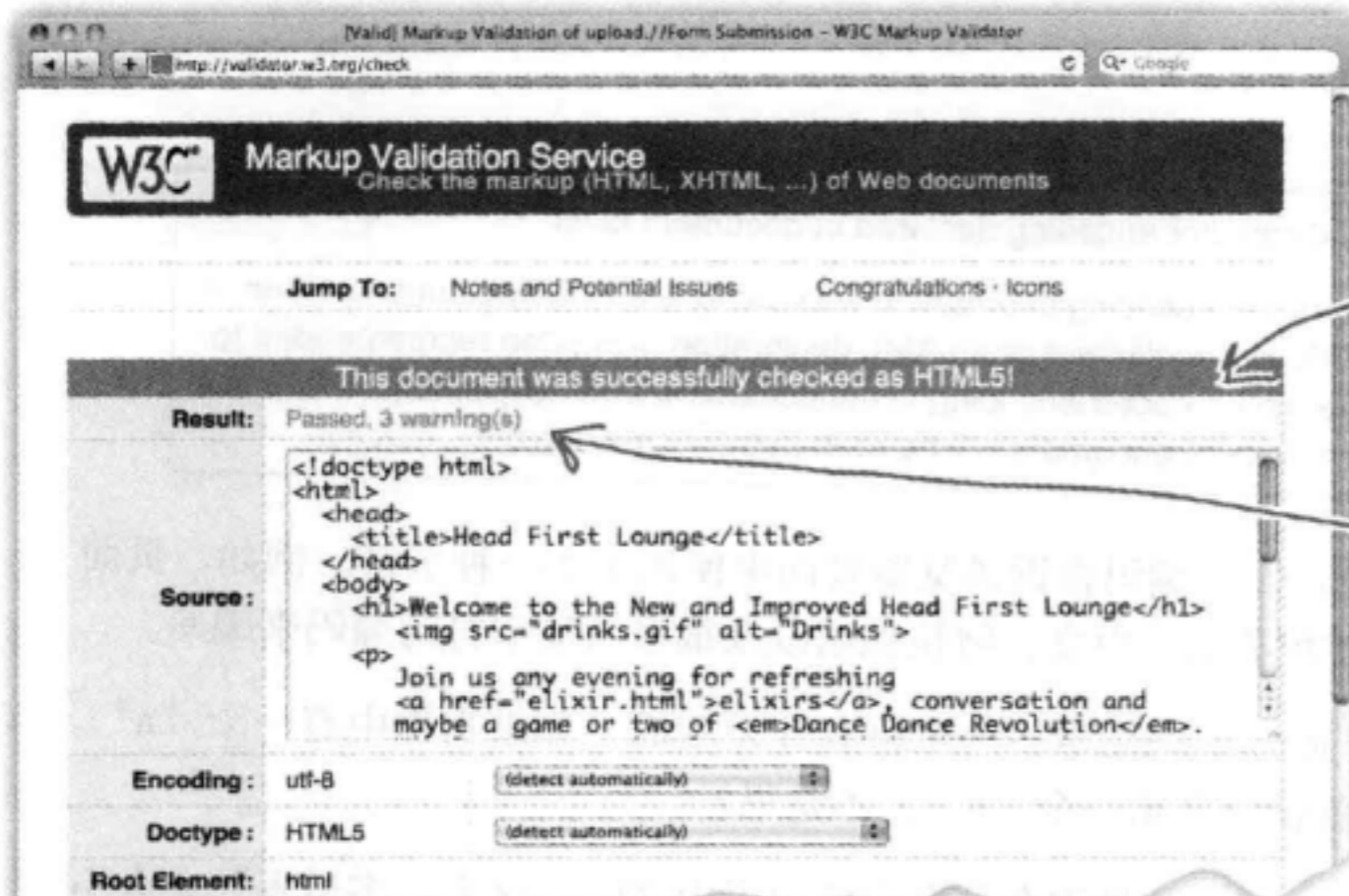
你已经知道alt属性是什么,把它增加到元素中。



在你看来, 为什么HTML5中alt属性是必要的?

就快完成了……

成功了！现在页面上出现一个绿条，这肯定是好事。不过，还有3个警告。看来我们还需要注意几个问题。下面来看一下：



通过了！不过……

……还有一些警告：我们要向下滚动，看一下这些警告……

Notes and Potential Issues

The following notes and warnings highlight missing or conflicting information which caused the validator to perform some guesswork prior to validation, or other things affecting the output below. If the guess or fallback is incorrect, it could make validation results entirely incoherent. It is *highly recommended* to check these potential issues, and, if necessary, fix them and re-validate the document.

Using experimental feature: HTML5 Conformance Checker.

The validator checked your document with an experimental feature: *HTML5 Conformance Checker*. This feature has been made available for your convenience, but be aware that it may be unreliable, or not perfectly up to date with the latest development of some cutting-edge technologies. If you find any issues with this feature, please report them. Thank you.

No Character encoding declared at document level

No character encoding information was found within the document, either in an HTML meta element or an XML declaration. It is often recommended to declare the character encoding in the document itself, especially if there is a chance that the document will be read from or saved to disk, CD, etc.

See this tutorial on character encoding for techniques and explanations.

Using Direct Input mode: UTF-8 character encoding assumed

Unlike the "by URI" and "by File Upload" modes, the "Direct Input" mode of the validator provides validated content in the form of characters pasted or typed in the validator's form field. This will automatically make the data UTF-8, and therefore the validator does not need to determine the character encoding of your document, and will ignore any charset information specified.

If you notice a discrepancy in detected character encoding between the "Direct Input" mode and other validator modes, this is likely to be the reason. It is neither a bug in the validator, nor in your document.

这里不用担心，这是一个标准警告，只要验证工具在W3C看来是实验性的（可能很长一段时间内都是如此），你就会经常看到这个警告。

嗯，这看起来是省略字符编码信息导致的问题。稍后我们就会讨论这个内容……

这表示，如果我们没有提供一个字符编码，他们会假设一个字符编码。

由此看来，从如何编写HTML来讲，我们已经得到了一个合法的文件，不过看起来还需要对“字符编码”做些处理。下面来看字符编码是什么意思……

看，我们得到了这个警告消息，验证工具无法找到一个字符编码。



◆ No Character encoding declared at document level

No character encoding information was found within the document, either in an HTML meta element or an XML declaration. It is often recommended to declare the character encoding in the document itself, especially if there is a chance that the document will be read from or saved to disk, CD, etc.

Frank: 字符编码告诉浏览器页面中使用了哪一种字符。例如，页面可以使用英语、中文、阿拉伯语以及很多其他字符的编码来编写。

Jim: 确定如何显示一个字符有什么难的？如果文件中有一个“a”，浏览器就会显示一个“a”。不是吗？

Frank: 嗯，如果你在页面中使用了中文呢？这个“字母表”完全不同，绝不仅仅是A~Z这26个字母。

Jim: 噢，说的对……不过难道浏览器不能区分吗？那些语言看起来与英语大不相同。

Frank: 不能，浏览器只能读取数据。它可能会猜测要使用哪种字符编码，不过，如果它猜错了呢？那么这不仅可能导致页面显示错误，还会带来潜在的漏洞，让黑客有机可趁。有了字符编码，就不需要浏览器去瞎猜了。

Jim: 我们建立这个网站已经很久了，为什么现在才谈到这个问题？

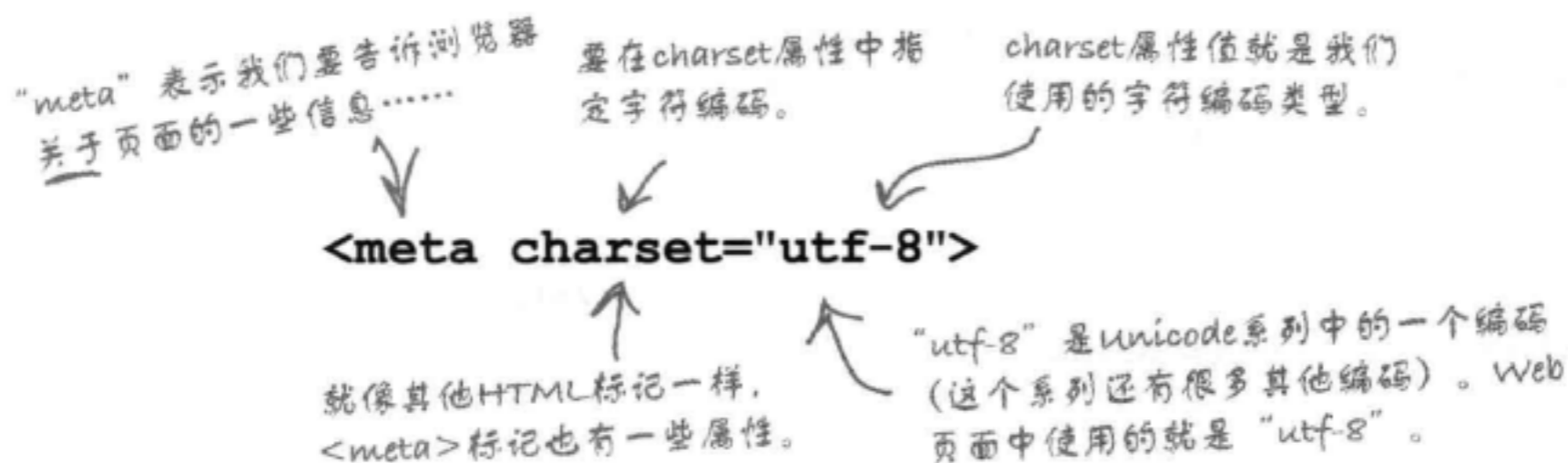
Frank: 因为验证工具在说“嘿，如果我要验证你的页面，那么你最好提前告诉我你要使用哪些字符！”想想看，我们总得满足浏览器的这个要求吧。别有压力，只需要为HTML再增加一行，这就是<meta>标记。我应该早点想到的。

Jim: 还有没有其他意外？我以为在文件中加入文档类型定义之后，我们的Web页面就合法了呢……

Frank: 但愿再没有其他意外了！下面加入<meta>标记，看看结果怎么样。

增加一个<meta>指定字符编码

字符编码为我们提供了一种方法，可以在计算机上表示某种语言中的所有字母、数字和其他符号。你可能知道这样一些编码，如ASCII，甚至莫尔斯码，还有很多其他的编码。幸运的是，如今标准已经统一为Unicode字符编码。采用Unicode，一种编码就可以表示所有语言。不过，由于还存在其他编码，所以我们还需要告诉浏览器我们在使用Unicode（或者你选择的另外一种编码）。要为Web页面指定Unicode，需要在HTML中加一个<meta>标记，如下所示：



there are no Dumb Questions

问：doctype、<meta>标记……噢，写Web页面真的需要这些吗？

答：指定doctype以及用<meta>标记指定字符编码有点像纳税：这是你的义务，必须履行。可以这样来看：在编写Web页面的人当中，你对它们的理解已经超过了98%的人，这很好。不过以后每个人都会在HTML中加上了doctype和<meta>标记，继续前进。所以要确保你的HTML也要加上它们，再来做更有意思的事情。

问：utf-8？

答：你要跟上我们的思路。这就像WD-40，你不用操心为什么它会叫这个名字，直接使用就可以了。正如我们所说，utf-8（有时也写作UTF-8）是Unicode编码系列中的一个编码。utf-8中的u表示Unicode。Unicode是很多常用软件应用和操作系统都支持的一个字符集，这也是Web选择的

编码，因为它支持所有语言和多语种文档（即多种语言的文档）。它还与ASCII兼容（ASCII是英语文档常用的一种编码）。如果你有兴趣了解关于Unicode或一般字符编码的更多内容，则可以查看<http://www.w3.org/International/O-charset.html>，那里提供了关于字符编码的更多信息。

问：我还看见过这样的<meta>标记：`<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`。是不是有时我得使用这个标记？

答：不用，这是HTML 4.01和更早版本中<meta>标记的格式。在HTML5中，可以直接写为`<meta charset="utf-8">`。

问：是不是由于这个原因，你才在第1章中使用utf-8编码来保存我们的文件？

答：没错。为服务器提供的文件的编码要与<meta>标记中指定的编码一致。

让验证工具（和很多浏览器）接受<meta>标记……

<meta>标记放在<head>元素中（应该记得，<head>包含有关页面的信息）。把这个<meta>标记行增加到你的HTML中。下面首先在“lounge.html”文件中增加这个标记：

```
<!doctype html>      这里是<meta>标记。我们把它增加到
<html>                <head>元素中，放在<title>元素上面。
  <head>
    <meta charset="utf-8"> ← ← ← 这一行要增加到<head>元
    <title>Head First Lounge</title> 素中所有其他元素的上面。
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing
      <a href="elixir.html">elixirs</a>, conversation and
      maybe a game or two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided; BYOWS (Bring
      your own Web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown
      Webville. If you need help finding us, check out our
      <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>
```

是不是想再试一试？这个HTML合法吗？首先，对“lounge.html”文件完成以上修改，保存，然后在浏览器中重新加载。同样的，你不会注意到任何变化，不过浏览器会注意到。现在来看它是否合法……

事不过三？

这一次，我们选择第2个标签页（通过文件上传验证）。你也可以选择对你最适用的方法。如果你想尝试上传方法，则可以把你的“lounge.html” HTML文件上传到W3C验证工具Web页面（<http://validator.w3.org>）。一旦上传，单击Check（检查）按钮……

“成功通过检查，这是HTML5”！
爱死这个绿条了！

还有一个警告……不过我们不需要担心它（见下面的说明）。

成功了！我们可以告诉老板，我们写的绝对是工业标准的HTML，甚至还可以说我们已经为HTML5做好了准备。



W3C Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Jump To: Notes and Potential Issues Congratulations - Icons

This document was successfully checked as HTML5!

Result: Passed, 1 warning(s)

File: Choose File no file selected
Use the file selection box above if you wish to re-validate the uploaded file lounge.html

Encoding: utf-8 (detect automatically)

Doctype: HTML5 (detect automatically)

Root Element: html

I ♥ VALIDATOR The W3C validators rely on community support for hosting and development. Donate and help us build better tools for a better web. 2936

Options

Show Source Show Outline List Messages Sequentially Group Error Messages by Type

Validate error pages Verbose Output Clean up Markup with HTML-Tidy

Help on the options is available. Revalidate

Notes and Potential Issues

The following notes and warnings highlight missing or conflicting information which caused the validator to perform some guesswork prior to validation, or other things affecting the output below. If the guess or fallback is incorrect, it could make validation results entirely incoherent. It is *highly recommended* to check these potential issues, and, if necessary, fix them and re-validate the document.

Using experimental feature: HTML5 Conformance Checker.

The validator checked your document with an experimental feature: HTML5 Conformance Checker. This feature has been made available for your convenience, but be aware that it may be unreliable, or not perfectly up to date with the latest development of some cutting-edge technologies. If you find any issues with this feature, please report them. Thank you.

Congratulations

The uploaded document "lounge.html" was successfully checked as HTML5. This means that the resource in question identified itself as "HTML5" and that we successfully performed a formal validation of it. The parser implementations we used for this check are based on validator.nu (HTML5).

这实际上是同一个警告，指出我们在使用一个“实验性的服务”。不用担心。

there are no Dumb Questions

问:验证工具说它对于HTML5还只是实验性的。这是什么意思?

答:验证工具给出消息“Using experimental feature: HTML5 Conformance Checker”（使用实验性特性：HTML5符合性检查工具），这是指，验证工具正在根据HTML5标准检查你的HTML，不过由于HTML5标准不是最终版本（还在增加新的特性），验证工具很可能会改变，所以验证页面得到的结果也不是固定的。这说明，作为一个尽职尽责的开发人员，你要尽可能与最新的HTML标准保持一致，经常检查你的页面。

问:这一章我们到底能得到什么？我的页面看起来还是一样的。

答:这一章我们稍稍对页面做了些调整，使它符合HTML规范。这有什么好处？与规范越接近，你的页面在实际中的表现就越好。如果要得到一个专业的Web页面，你要使用工业标准写这个页面，这正是我们在这一章中所做的，增加doctype、设置字符编码以及解决HTML中的小问题（alt属性），都是在做这方面的工作。

问:到底为什么需要那个alt属性？

答:有两个主要原因。首先，如果你的图像出于某种原因无法显示（比如说，你的图像服务器宕机了，或者你的连接实在太慢了），这个alt属性（在大多数浏览器上）会帮你显示所指定的alt文本，来取代图像。其次，对于有视力障碍的用户，他们会使用一个屏幕阅读器来阅读页面，屏幕阅读器可以为用户读出alt文本，这样能帮助他们更好地理解页面。

问:如果我告诉浏览器我在使用HTML5，但实际上并没有，会怎么样呢？

答:浏览器会发现你实际上没有写HTML5，并使用它提供的错误处理功能来尝试采取正确的做法。在这方面，你又会遇到老问题，不同的浏览器会采用不同的方式处理页面。要得到可预测的结果，唯一的办法就是告诉浏览器你

在使用HTML5，而且没有撒谎，确实是在使用HTML5。

问:我们对HTML5做了一些讨论，不过我想再明确一下：我们写的HTML和HTML5之间有没有差别？

答:我们在使用标准HTML，也就是HTML5。现在HTML5引入了一些新的标记（很快我们就会看到），另外还支持编写Web应用（这本书不作介绍）。不过HTML5还是HTML，你写的所有代码都“符合”HTML5。所以，很抱歉这些术语可能让你有些混乱，不过请相信，所有这些都是HTML，包括HTML5规范提供的所有新特性。

好消息是，你之前学到的所有知识都能很好地用于HTML5，实际上，你会看到，要从一个“非正式的HTML”页面变成一个专业的HTML页面，几乎不需要你做什么。也就是说，你可以告诉你的老板你已经在用HTML5，也许这能让你更快地升职加薪。

问:HTML5与HTML 4.01相比有哪些大的改进？

答:HTML5的好处有三方面。首先，HTML5中提供了一些新元素和属性，这些新元素和新属性很酷（如<video>元素），另外一些可以帮助你编写更好的页面（本书后面还会介绍这些元素）。

其次，它有很多新特性，允许Web开发人员用HTML5创建Web应用。Web应用就是表现得像应用一样的Web页面（就像你通常在笔记本电脑或移动设备上使用的那些应用），而不只是静态Web页面。如果你对创建Web应用感兴趣，读完这本书之后（不好意思，做个广告），你可以看看O'Reilly出版的另一本好书：《Head First HTML5 Programming》。

最后，HTML5规范比HTML以前的版本健壮得多。还记得吧，我们说过，这个规范现在会记录Web开发人员常犯的错误，还会帮助浏览器了解如何处理这些错误，是不是？这说明有错误的Web页面不会像以往那样导致恐慌，这对用户来说确实是一件好事。

总的来讲，与HTML 4.01相比，HTML5确实有了很大的改进，绝对值得学习。我们会在接下来几章加快速度，继续深入。



Exercise

该轮到你了。为“directions.html”和“elixir.html”增加 <meta>标记。尝试对这两个页面完成验证：它们通过验证了吗？如果没有，请进行修复，使它们能顺利通过验证。

在这里的空白处写出
你的验证体验！



Exercise

现在使用验证工具做个小练习。在你刚才成功验证为HTML5的代码（见241页）中，去掉doctype。没关系，先把它去掉，看看验证时会发生什么。再把这个版本的文件提交到验证工具，看看会发生什么。在下面记录你得到的错误。

```

<del>!doctype html</del> ← 删除doctype!
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing
      <a href="elixir.html">elixirs</a>, conversation and
      maybe a game or two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided; BYOWS (Bring
      your own Web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown
      Webville. If you need help finding us, check out our
      <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>

```

记录写在这里，你得到多少
个错误？

如果没有包含doctype，则从这里可以得出关于
HTML类型的哪些信息？

叫所有HTML专业人员来拿手册……

欢迎HTML开发精英的到来，他们知道如何创建专业的页面。要记的东西太多了，所以Web镇提供了一个简明指南，来帮助创建工业标准页面。作为刚刚到访Web镇的你来说，这个指南很有意义。这不是一个详尽的参考资料，只强调了构建页面时比较重要的一些最佳实践。在后面几章游览Web镇时，你还会对这个指南里的知识有所补充，不过，现在先拿一本吧，都是免费的。



Web镇HTML指南

在这个简明指南中，我们把编写合法HTML页面的做法汇集为一组常识性的原则。请依次核对：



一定要以<doctype>开头。

每个页面都要从一个doctype开始。这样才能让浏览器和验证工具顺利开展工作。

任何时候都要使用<!doctype html>，除非你确实是在写HTML 4.01或XHTML。



<html>元素：不能没有它。

紧接着doctype，<html>元素必须是Web页面的最顶层元素或根元素。所以，在doctype后面，由<html>标记开始你的页面，</html>标记结束页面，页面中的所有其他内容都嵌套在这个元素中。



记住，要使用<head>和<body>编写更好的HTML。

只有<head>和<body>元素能直接放在<html>元素中。这说明，所有其他元素都必须放在<head>或<body>元素中。没有例外！



在<head>中指定正确的字符编码。

在<head>中包含一个<meta charset= " utf-8 " >标记。浏览器会感谢你的，用户在你的博客上阅读世界各地用户发表的评论时，也会因为你指定了正确的字符编码而感谢你。

Web镇HTML指南 (续)

在这个简明指南中，我们把编写合法HTML页面的做法汇集为一组常识性的原则。请依次核对：



没有<title>的<head>算什么？

一定要在<head>元素中包含一个<title>元素。这是雷打不动的原则。如果没有做到，则会得到不符合标准的HTML。只能在<head>元素中放置<title>、<meta>和<style>元素。



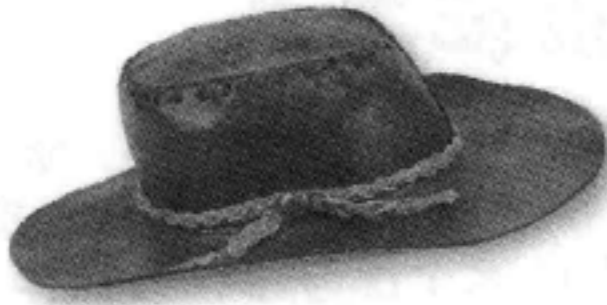
嵌套某些元素时要当心。

在这里提供的原则中，嵌套规则相当灵活。不过有一些情况是没有意义的。不要把<a>元素嵌在另一个<a>元素中，因为这样会让访问者很迷惑。另外，不允许在等void元素中嵌套其他内联元素。



检查属性！

有些元素属性是必要的，有些则是可选的。例如，如果元素没有src属性，就没有什么意义，现在你还知道了，alt属性也是必要的。在学习过程中，要逐渐熟悉各个元素的必要和可选属性。



HTML 考古



在这本书中，你一直在使用当前HTML标准中的元素和属性。所以，你没有多少机会看到那些过气被淘汰的元素和属性。其中大多数元素都是在HTML 4.01中被淘汰的，不过在老的Web页面中有时还会出现，因此对这些遗留的元素多些了解没有坏处。我们挖出了一个HTML 3.2页面，其中包含现在标准中已经没有一些元素和属性，另外这个页面还存在现代HTML不推荐的一些错误做法。

```
<html>
```

```
<head>
```

```
<title>Webville Forecast</title>
```

```
</head>
```

这里是一些控制表现的属性。bgcolor设置页面的背景色，text设置体文本的颜色。

```
<body bgcolor="tan" text="black">
```

```
<p>
```

```
The weather report says lots of rain and wind in store for
<font face="arial">Webville</font> today, so be sure to
stay inside if you can.
```

```
</p>
```

利用元素和它的face属性来改变字体。

```
<ul>
```

```
<li>Tuesday: Rain and 60 degrees.
```

```
<li>Wednesday: Rain and 62 degrees.
```

```
</ul>
```

可以没有结束标记，如和</p>。有时你仍然能这么做，不过不推荐这种做法！

```
<p align=right>
```

```
Bring your umbrella!
```

属性值缺少引号。现在建议都要加上引号，对于有多个值的属性，引号更是必要的。

```
<center><font size="small">This page brought to you buy Lou's
Diner, a Webville institution for over 50 years.
```

```
</font></center>
```

由元素使用size属性来控制文本大小。

这里有两种对齐文本的方法。右对齐一个段落，并居中一段文字。

```
</body>
```

```
</html>
```



扮演验证工具

下面你会看到一个HTML文件。你的任务是扮演验证工具，找出所有错误。完成这个练习后，对照本章最后给出的答案看看是否找到了所有错误。

完成之后（或者如果你需要提示），可以使用验证工具来检查你的答案对不对。

```
<html>
<head>
  <meta charset="utf-9">
</head>
<body>
  
  <h1>Tips for Enjoying Your Visit in Webville
  <p>
    Here are a few tips to help you better enjoy your stay in Webville.
  </p>
  <ul>
    <li>Always dress in layers and keep an html around your
      head and body.</li>
    <li>Get plenty of rest while you're here, sleep helps all
      those rules sink in.</li>
    <li>Don't miss the work of our local artists right downtown
      in the CSS gallery.
  </ul>
  </p>
  <p>
    Having problems? You can always find answers at
    <a href="http://wickedlysmart.com"><em>WickedlySmart</em></a>.
    Still got problems? Relax, Webville's a friendly place, just ask someone
    for help. And, as a local used to say:
  </p>
  <em><p>
    Don't worry. As long as you hit that wire with the connecting hook
    at precisely 88mph the instant the lightning strikes the tower...
    everything will be fine.
  </em></p>
</body>
</html>
```

建立HTML并不太难，
不过确定它无误确实需要花点功夫，
现在我们该用CSS对它增加样式了吧。
那是完全不同的一种语言，对不对？



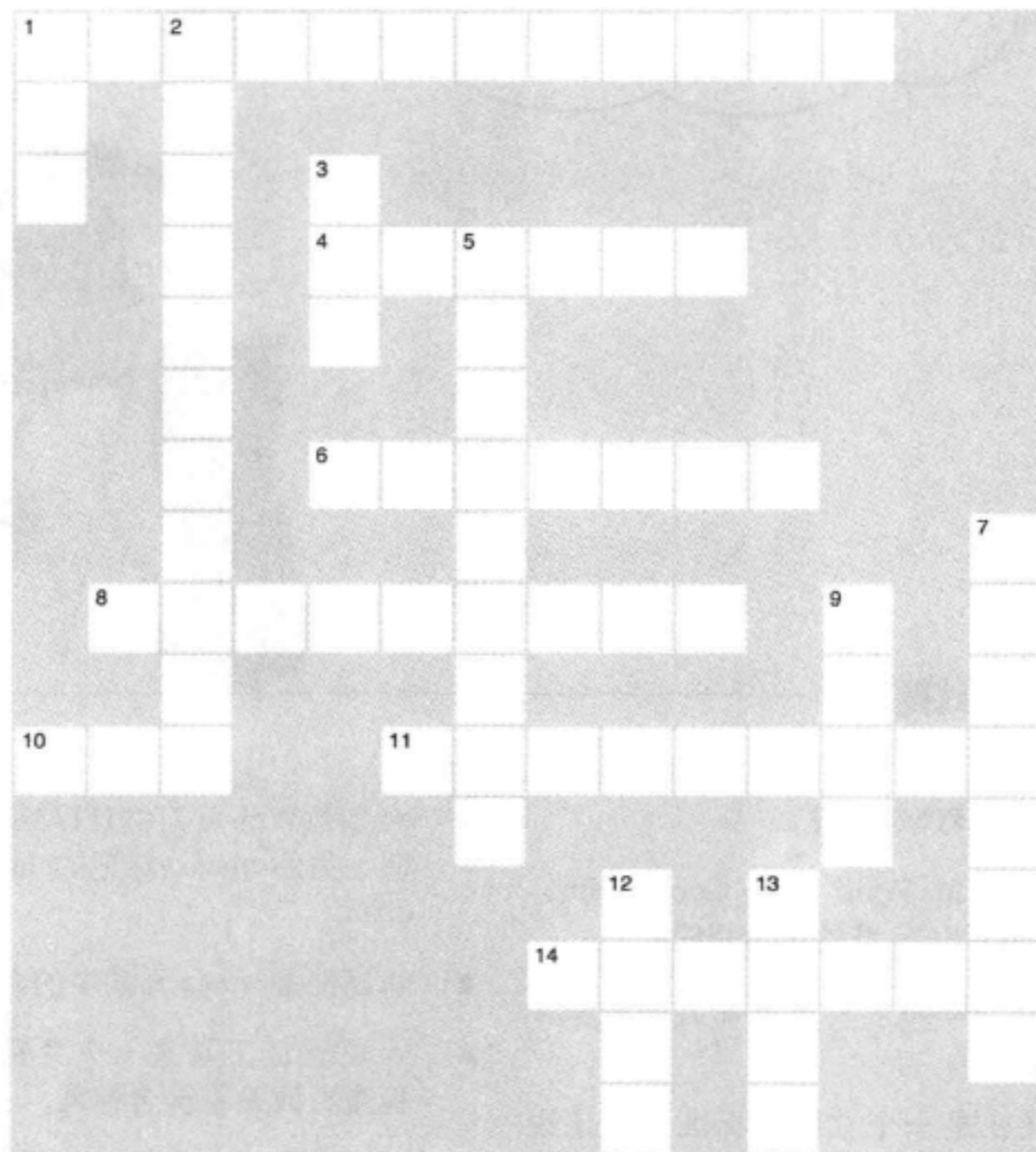
BULLET POINTS

- HTML5是当前的HTML标准。
- 万维网协会 (World Wide Web Consortium, W3C) 是定义HTML标准的标准组织。
- 文档类型定义 (doctype) 用来告诉浏览器你使用的HTML版本。
- HTML标准现在是一个“活的标准”，这说明这个标准会不断改变，加入新的特性和更新。
- <head>元素中的<meta>标记告诉浏览器关于一个Web页面的额外信息，如内容类型和字符编码。
- <meta>标记的charset属性告诉浏览器Web页面使用的字符编码。
- 大多数Web页面的HTML文件都使用utf-8编码，另外<meta>标记的charset属性值通常也是utf-8。
- alt属性是元素中的必要属性。
- W3C验证工具是一个免费的在线服务，可以检查页面是否符合标准。
- 可以使用这个验证工具确保你的HTML合法，而且元素和属性符合标准。
- 如果遵循标准，则你的页面会更快地显示，而且在不同浏览器中显示时差异会更小，CSS也能更好地工作。



HTML填字游戏

这一章内容真多。现在可以坐下来喝杯饮料了，下面做做这个填字游戏，强化一下学到的知识。所有答案都可以从这一章中找到。



横向

1. 浏览器战争中的受害者。
4. HTML标准是一个_____标准。
6. <head>元素中必要的属性。
8. 制订Web标准的人承诺将来的HTML会与老的HTML_____兼容。
10. 老板希望向休闲室页面增加_____之前先完成标准化。
11. 你的HTML满足标准时，就是_____。
14. 这个定义告诉浏览器和验证工具你在创建什么类型的文档。

纵向

1. 提供验证工具的标准组织。
2. Microsoft与Netscape之争。
3. 标准HTML中必要的属性。
5. 这个服务会检查你的HTML是否符合标准。
7. 老的_____与最新的相比复杂得多。
9. 在这里放置有关页面的信息。
12. 在这里放置Web页面内容。
13. Web页面最常用的编码。

扮演验证工具 答案



下面你会看到一个HTML文件。你的任务是扮演验证工具，找出所有错误。下面给出答案。

```

<html>
<head>
  <meta charset="utf-9">
</head>
<body>
  
  <h1>Tips for Enjoying Your Visit in Webville
  <p>
    Here are a few tips to help you better enjoy your stay in Webville.
  </p>
  <ul>
    <li>Always dress in layers and keep an html around your
      head and body.</li>
    <li>Get plenty of rest while you're here, sleep helps all
      those rules sink in.</li>
    <li>Don't miss the work of our local artists right downtown
      in the CSS gallery.
  </ul>
  </p>
  <p>
    Having problems? You can always find answers at
    <a href="http://wickedlysmart.com"><em>WickedlySmart</em></a>.
    Still got problems? Relax, Webville's a friendly place, just ask someone
    for help. And, as a local used to say:
  </p>
  <em><p>
    Don't worry. As long as you hit that wire with the connecting hook
    at precisely 88mph the instant the lightning strikes the tower...
    everything will be fine.
  </em></p>
</body>
</html>

```

缺少doctype。

应该是“utf-8”而不是“utf-9”
(这根本不存在)!

<title>应该放在<head>中。

没有alt属性。

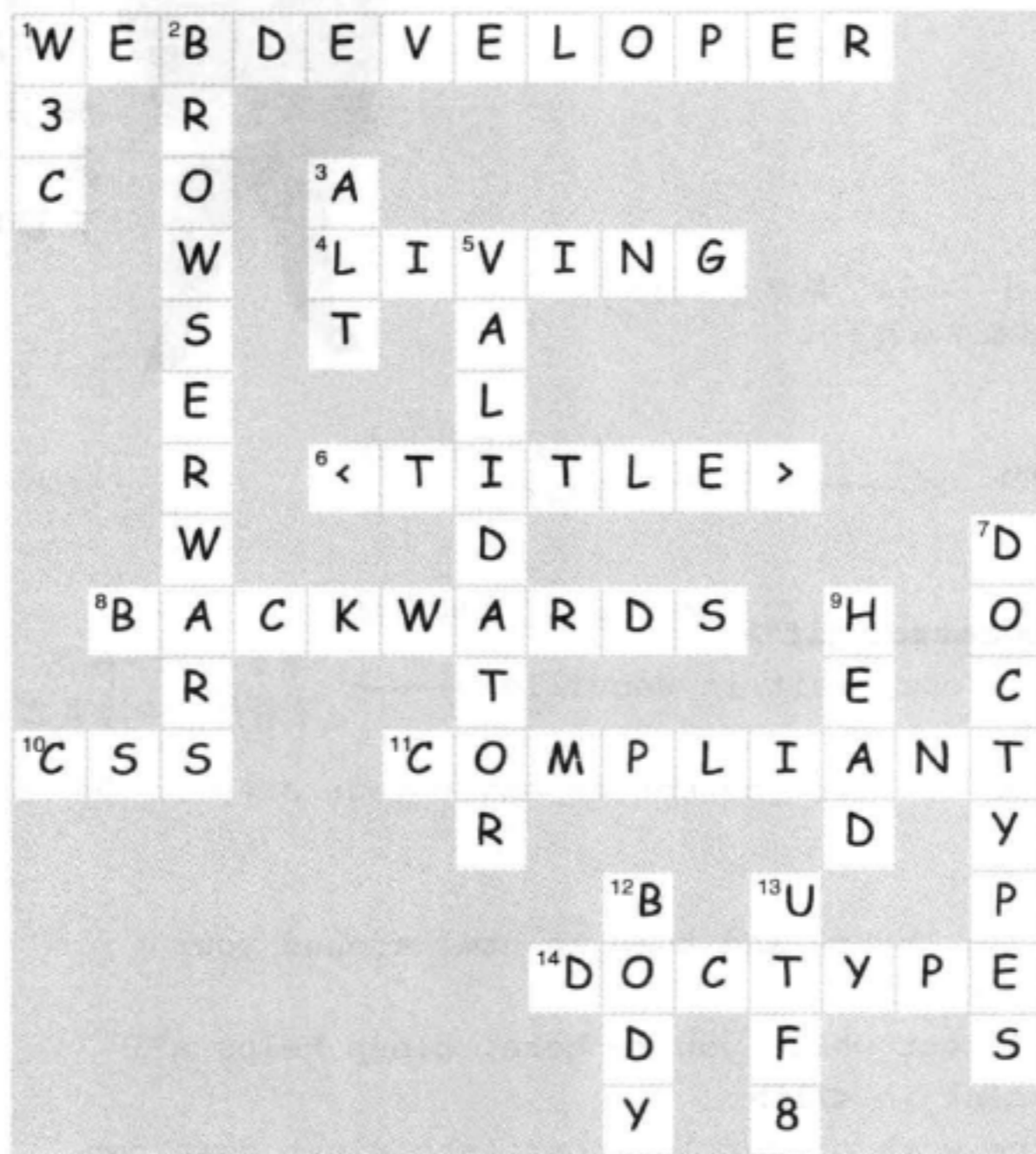
缺少</h1>标记。这会使下面的<p>元素出问题。

缺少标记。这样也能通过验证，不过不推荐这么做!

多余的</p>，没有与之匹配的<p>。

和<p>标记换了位置。

HTML填字游戏答案



该轮到你了。为“directions.html”和“elixir.html”增加<meta>标记。尝试对这两个页面完成验证：它们通过验证了吗？如果没有，则请进行修复，使它们能顺利通过验证。

答案：要想成功验证“elixir.html”，必须为各个元素增加alt属性。



Exercise

现在使用验证工具做个小练习。在你刚才成功验证为HTML5的代码（见241页）中，去掉doctype。没关系，先把它去掉，看看验证时会发生什么。再把这个版本的文件提交到验证工具，看看会发生什么。在下面记录你得到的错误。

```

<del>!doctype html</del> ← 删除doctype!
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the New and Improved Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing
      <a href="elixir.html">elixirs</a>, conversation and
      maybe a game or two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided; BYOWS (Bring
      your own Web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown
      Webville. If you need help finding us, check out our
      <a href="directions.html">detailed directions</a>.
      Come join us!
    </p>
  </body>
</html>

```

如果验证时没有doctype，则我们会得到3个错误，还有4个警告。验证工具会假设我们在用HTML 4.01 Transitional写代码（这是HTML 4.01的一个版本，专门设计为在“迁移”到XHTML时使用）。验证工具确实对没有doctype很不满意，所以抱怨了很多次。它还对<meta charset="utf-8">有怨言，因为在HTML5之前，charset不是<meta>标记的一个合法属性。通过这个练习你可以了解到，加上一个doctype可以让验证工具和浏览器更高兴。

记录写在这里，你得到多少个错误？ ↗

加一点样式

别误解我的意思，漂亮的发型，可爱的帽子，看上去确实很迷人。不过你不觉得吗？如果你多花点时间为你的HTML加一些样式，他会更喜欢的！



我听说这本书会介绍CSS。到目前为止，你一直在专心学习HTML来创建Web页面结构。不过可以看到，浏览器对样式有很多要求。当然，我们可以请时尚人士来帮忙，不过没有必要。利用CSS，你就能完全掌控页面的表现，通常甚至不用对HTML做任何改变。真的这么容易吗？嗯，你得学习一种新的语言。毕竟，Web镇是一个讲多种语言的小镇。读完这一章对CSS语言学习的介绍后，你就能胸有成竹地站在主干道两边与Web镇的人们交流了。

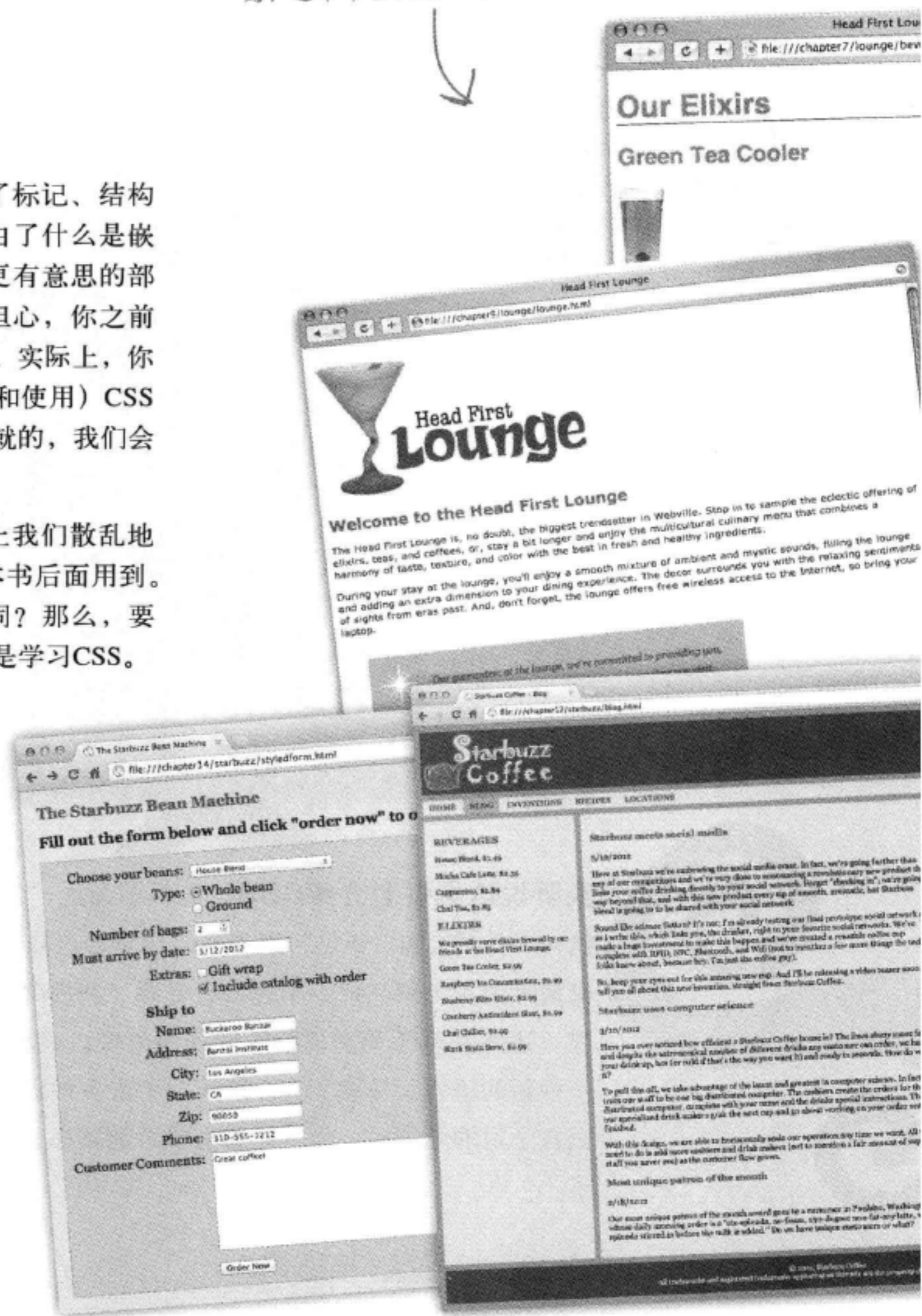
还记得绿野仙踪吗？嗯，从这一部分开始，这本书会从黑白变成彩色的。

你已经离开堪萨斯

你已经做了不错的热身运动，了解了标记、结构和验证，知道了正确的语法，还明白了什么是嵌套和符合标准，不过现在你会进入更有意思的部分，要为页面增加样式了。不过别担心，你之前做的所有这些HTML工作都不会白费。实际上，你会看到，充分掌握HTML对于学习（和使用）CSS至关重要。另外学习CSS不是一蹴而就的，我们会通过后面几章逐步学习CSS。

为了让你有些初步的认识，这两页上我们散乱地摆放了一些设计，这些设计都将在本书后面用到。是不是与你之前创建的页面大不相同？那么，要创建这些页面你需要做些什么？当然是学习CSS。

下面就开始吧……





to the Head First Lounge

Lounge is, no doubt, the biggest trendsetter in the industry to sample the eclectic offering of elixirs, teas, and more. Stay a bit longer and enjoy the multicultural atmosphere that combines a harmony of taste, texture, and color with fresh and healthy ingredients.

at the lounge, you'll enjoy a smooth mixture of live music, filling the lounge and adding an extra dimension to your dining experience. The decor surrounds the sentiments of sights from eras past. And, lounge offers free wireless access to the web on your laptop.

In guarantee of the lounge, we're committed to making you, our guest, each an exceptional experience every time you visit. Whether you're looking for a quick bite or a full meal or just want to relax, you'll find our knowledgeable service staff attentive to every detail. If you're not a member, have a Raspberry Bliss drink on us.

right, join us in the backroom as our house band plays a selection of trance and drum&bass music on a vinyl-themed dance floor. Or just hang out at the bar with our white vinyl booths. You'll be delivered from the main lounge right to the backroom. And, no matter where you find yourself,

Weekly Elixir Specials



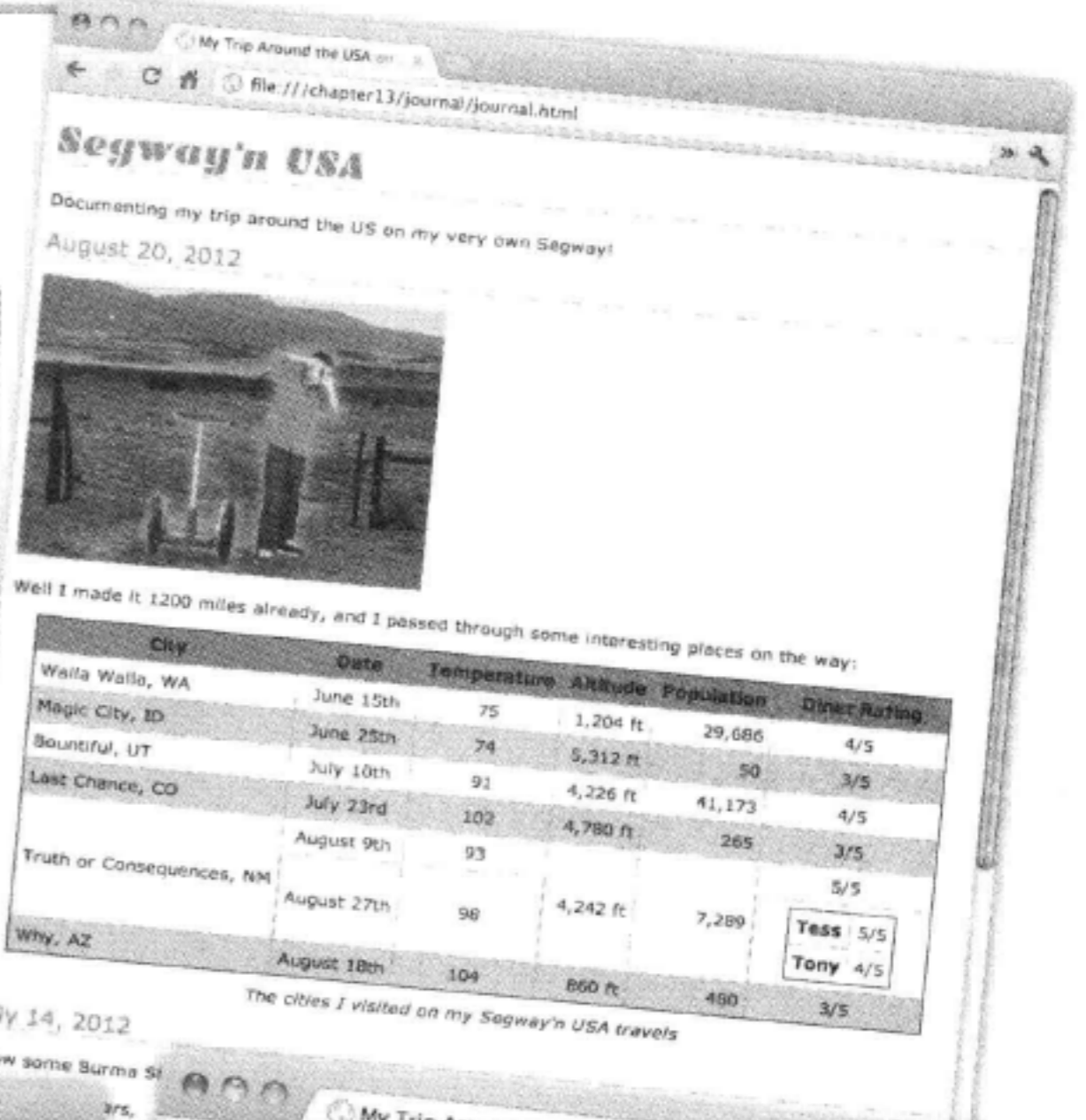
Lemon Breeze

The ultimate healthy drink, this elixir combines herbal botanicals, minerals, and vitamins with a twist of lemon into a smooth citrus wonder that will keep your immune system going all day and all night.



Chai Chiller

Not your traditional chai, this elixir mixes matcha with chai spices and adds an extra chocolate kick for a caffeinated taste sensation on ice.



Well I made it 1200 miles already, and I passed through some interesting places on the way:

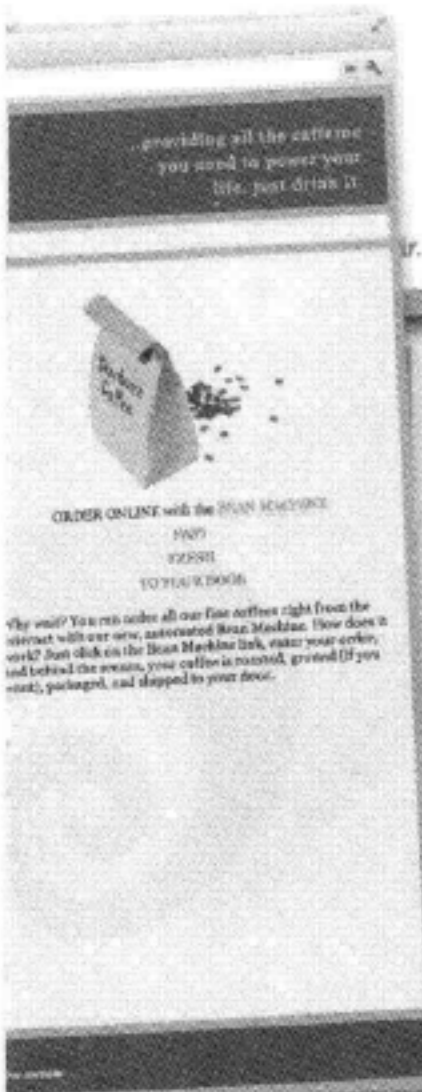
City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5
Bountiful, UT	July 10th	91	4,226 ft	41,173	4/5
Last Chance, CO	July 23rd	102	4,780 ft	265	3/5
Truth or Consequences, NM	August 9th	93	4,242 ft	7,289	5/5
Why, AZ	August 27th	98	4,242 ft	7,289	4/5
	August 18th	104	860 ft	490	3/5

The cities I visited on my Segway'n USA travels

July 14, 2012

I saw some Burma Si

to a base of elderflower herb tea will



l/starbuzz/index.html



...providing all the caffeine you need to power your life. Just drink it.

QUALITY COFFEE, QUALITY CAFFEINE

Starbuzz Coffee, we are dedicated to filling all your caffeine needs through our quality coffees and teas. Sure, we want you to great cup of coffee and a great coffee experience as well, but we're the only company that actively monitors and optimizes caffeine levels. So stop by and fill your cup, or order online with our Bean Machine online order form, and get that quality Starbuzz coffee that you know will meet your caffeine standards.

And, did we mention caffeine? We've just started funding the guys doing all the wonderful research at the Caffeine Room. If you want the latest on coffee and other caffeine products, stop by and pay them a visit.

OUR STORY

"A man, a plan, a coffee bean". Okay, that doesn't make a palindrome, but it resulted in a damn good cup of coffee. Starbuzz's CEO is that man, and you already know his plan: a Starbuzz on every corner.

In only a few years he's executed that plan and today you can enjoy Starbuzz just about anywhere. And, of course, the big news this year is that Starbuzz teamed up with Head First readers to create Starbuzz's Web presence, which is growing rapidly and helping to meet the caffeine needs of a whole new set of customers.

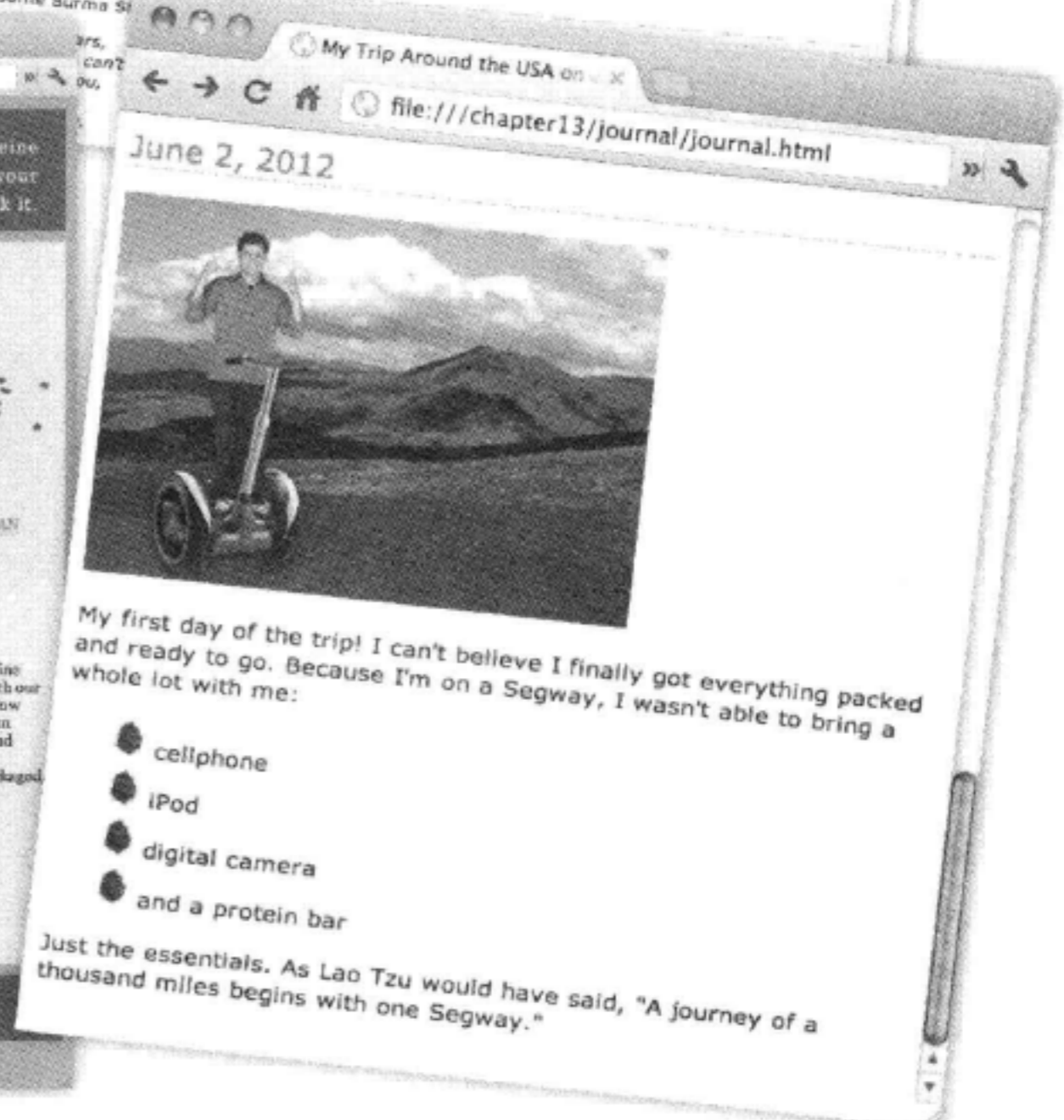
STARBUZZ COFFEE BEVERAGES

We've got a variety of caffeinated beverages to choose from at Starbuzz, including our House Blend, Mocha Cafe Latte, Cappuccino, and a favorite of our customers, Chai Tea.

We also offer a variety of coffee beans, whole or ground, for you to take home with you. Order your coffee today using our online Bean Machine, and take the Starbuzz Coffee experience home.

ORDER ONLINE with the BEAN MACHINE
FAST
FRESH
TO YOUR DOOR

Why wait? You can order all our fine coffees right from the Internet with our new, automated Bean Machine. How does it work? Just click on the Bean Machine link, enter your order, and behind the scenes, your coffee is roasted, ground (if you want), packaged, and shipped to your door.



My first day of the trip! I can't believe I finally got everything packed and ready to go. Because I'm on a Segway, I wasn't able to bring a whole lot with me:

- cellphone
- iPod
- digital camera
- and a protein bar

Just the essentials. As Lao Tzu would have said, "A journey of a thousand miles begins with one Segway."

偶然听到的Web镇“交换空间”节目

没有看过最近的真人秀节目？没关系，我来解释一下：有两个邻居，两所房子，还有1000美元。这两个邻居要交换房子，用这1000美元在48小时内完全重新设计一到两个房间。来听听他们说什么……

```
OK, 要在这个房间里做些设计……  
bedroom {  
  drapes: blue;  
  carpet: wool shag;  
}
```

```
……这个浴室确实需要重新设计一下！  
bathroom {  
  tile: 1in white;  
  drapes: pink;  
}
```



当然，在Web镇版的交换空间节目里，所有人谈论的都是CSS设计。如果你听不懂他们在说什么，下面给你一个小小的翻译提示：CSS中的每个语句包括一个场所（如卧室），以及这个场所的一个属性（如窗帘或地毯），还有要应用到这个属性的一个样式（如蓝色，或者1英寸的瓷砖）。

结合HTML和CSS

我们相信，在家装设计领域CSS会有光明的前景，不过下面先回到HTML。HTML没有房间，但它有元素，这些元素会作为将要指定样式的场所。想让你的<p>元素的墙漆成红色？没问题。只不过段落没有墙，所以必须换成段落的background-color属性。可以这样做：

你要做的第一件事是选择需要增加样式的元素，在这里就是<p>元素。注意，在CSS中，不用在名字两边加<>。

然后指定你想要设置样式的属性，在这里就是<p>元素的背景色。

p {

background-color: red;

}

把<p>元素的所有样式放在大括号{}之间。

在属性和相应的值之间有一个冒号。

另外要把background-color (背景色) 设置为red (红色)。

最后，加一个分号。

整个称为一个规则 (RULE)。

还可以这样写规则：

```
p { background-color: red; }
```

在这里，我们所做的只是去掉了换行。就像HTML一样，可以按你的想法安排CSS的格式。对于更长的规则，通常需要增加一些换行和缩进，让CSS（对你来说）更可读。

想要增加更多样式？

在每个CSS规则中可以根据需要增加多个属性和值。假设你想为段落再加一个边框。可以这样做：

```
p {
```

```
background-color: red;
```

```
border: 1px solid gray;
```

```
}
```

<p>元素有一个边框……

……粗细为1像素，灰色，实线。

你要做的就是再增加另一个属性和值。

there are no Dumb Questions

问:是不是每个<p>元素都有相同的样式?我能不能(比如说)让两个段落有不同的颜色?

答:目前为止我们使用的CSS规则会为所有段落定义样式,不过CSS表达能力很强。它可以采用很多不同方式为不同的元素指定样式,甚至可以为元素子集指定样式。这一章后面你就会看到如何让两个段落有不同的颜色。

问:我怎么知道可以对一个元素的哪些属性设置样式呢?

答:嗯,元素的很多属性都可以设置样式,太多了,你肯定记不住。通过后面几章的学习,你会非常熟悉较为常用的一些属性。可能还需要找一本好的CSS参考书。网上有很多参考资料,另外O'Reilly的《CSS Pocket Reference》是一本不错的小书,可供参考。

问:再帮我复习一下,为什么我非要用一种不同的语言来定义所有这些样式,而不是使用HTML呢?既然元素都是用HTML写的,用HTML设置样式不是更容易吗?

答:在后面几章你就会看到使用CSS的一些重要优点。不过先简单回答一下你的问题,CSS确实比HTML更适合指定样式信息。通过使用一点点CSS,你可以对HTML建立相当复杂的效果。你还会看到,CSS更有利于处理多个页面的样式。这一章后面会介绍这是如何做到的。



假设在段落中有一个元素。如果改变了段落的背景色,你是不是认为还要改变元素的背景色,才能与段落的背景色一致?

把CSS放入HTML

你现在已经知道一点CSS语法了。你知道如何选择一个元素，然后写一个规则，其中包含属性和属性值。不过，你还需要把这个CSS放到某个HTML中。首先，我们需要一些HTML来放入CSS。在后面几章中，还会再去拜访我们的老朋友——Starbuzz咖啡馆，还有Tony和他的Segway之旅日志网站，当然会让这些网站更有表现力。不过，你认为谁的愿望最迫切，希望最先增加样式？当然是Head First休闲室的那些人。所以，下面给出Head First休闲室主页的HTML。应该记得，上一章我们做过一些修复工作，让它成为正确的HTML（我们真是好人，对不对）。现在来增加一些样式标记，这是为页面增加样式的最容易的方法。

不过，不一定是最好的办法。这一章后面还会再来讨论这个问题，介绍另一种方法。

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Head First Lounge</title>
```

```
<style>
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to the Head First Lounge</h1>
```

```
<p>
```

```

```

```
</p>
```

```
<p>
```

```
Join us any evening for refreshing
```

```
<a href="beverages/elixir.html">elixirs</a>，
```

```
conversation and maybe a game or two of
```

```
<em>Dance Dance Revolution</em>.
```

```
Wireless access is always provided;
```

```
BYOWS (Bring your own Web sever).
```

```
</p>
```

```
<h2>Directions</h2>
```

```
<p>
```

```
You'll find us right in the center of downtown
```

```
Webville. If you need help finding us, check out our
```

```
<a href="about/directions.html">detailed directions</a>.
```

```
Come join us!
```

```
</p>
```

```
</body>
```

```
</html>
```

这是我们感兴趣的東西：`<style>`元素。

要为HTML直接增加CSS样式，需要在`<head>`元素中增加开始和结束`style`标记。

你的CSS规则要放在这里。



为休闲室增加样式

既然在HTML中增加了<style>元素，下面再为休闲室增加一些样式，使你对编写CSS规则有一点具体认识。这个设计可能不会让你赢得“设计大奖”，不过你可以从这里起步。

首先要做的是改变段落中文本的颜色（要与休闲室里那些红色沙发搭配）。为此，可以使用CSS color属性，如下所示：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
    <style>
```

这个规则将指定段落的字体颜色。

我们选择<p>元素来应用这个样式。

```
p {
  color: maroon;
}
```

改变字体颜色的属性名为“color”（你可能以为是“font-color”或“text-color”，可惜并不是）。

我们要把文本颜色设置为可爱的茶红色，这正好与休闲室沙发的颜色搭配。

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to the Head First Lounge</h1>
```

```
<p>
```

```

```

```
</p>
```

```
<p>
```

```
Join us any evening for refreshing
<a href="beverages/elixir.html">elixirs</a>,
conversation and maybe a game or two of
<em>Dance Dance Revolution</em>.
Wireless access is always provided;
BYOWS (Bring your own Web server).
```

```
</p>
```

```
<h2>Directions</h2>
```

```
<p>
```

```
You'll find us right in the center of downtown
Webville. If you need help finding us, check out our
<a href="about/directions.html">detailed directions</a>.
Come join us!
```

```
</p>
```

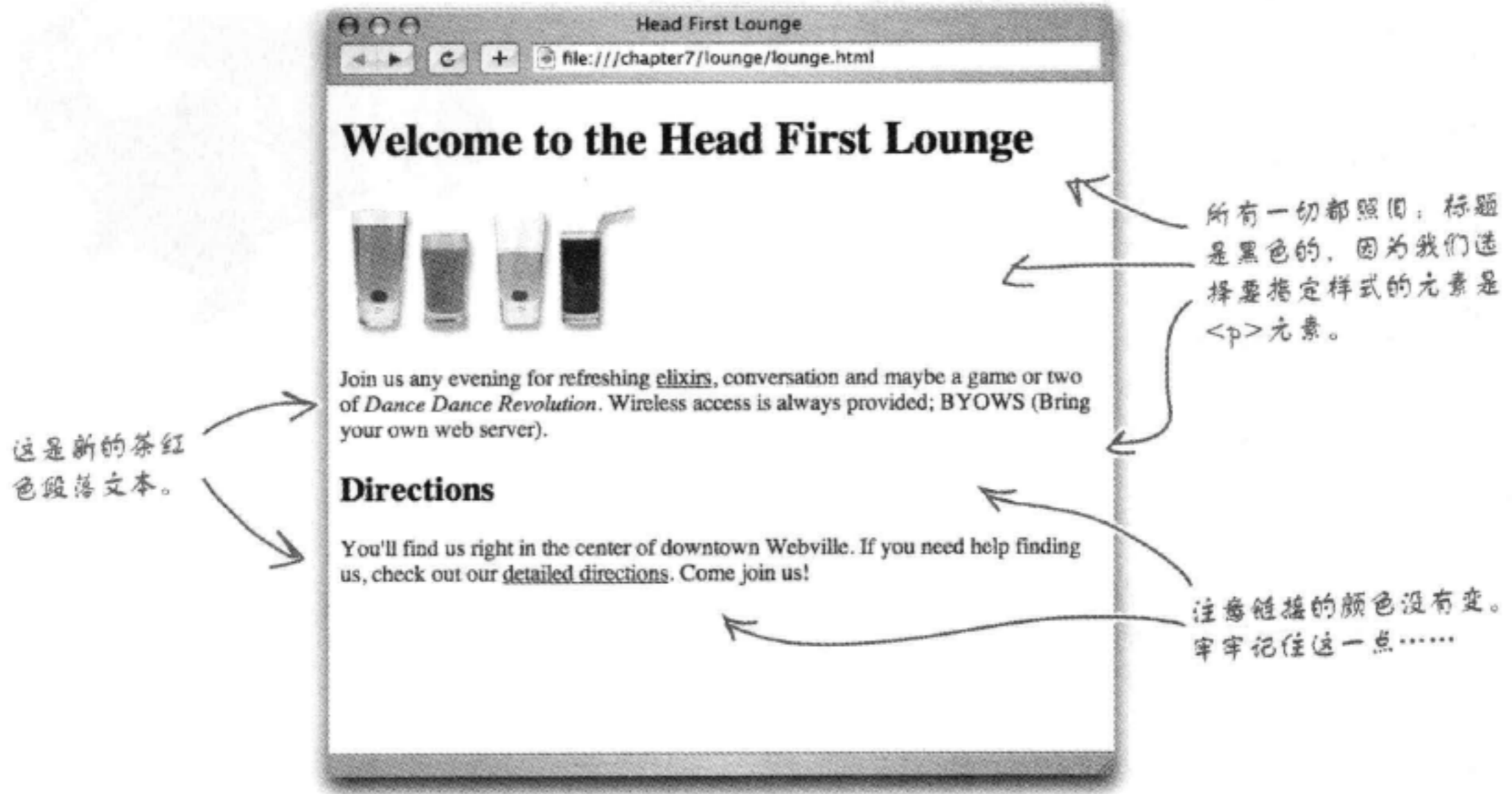
```
</body>
```

```
</html>
```

p选择器会选择HTML中的所有段落。

试一试：展现样式

对“chapter7/lounge”文件夹中的“lounge.html”文件完成前面几页的所有修改，然后保存，并在浏览器中重新加载这个页面。你会看到段落的文本颜色已经改为茶红色：



如果不设置color，而是要把<p>元素的background-color设置为maroon会怎么样呢？浏览器显示页面时会有变化吗？

对标题加样式

现在为这些标题加一些样式。把字体稍做改变怎么样？下面就来改变标题的字体和颜色：

```
h1 {  
  font-family: sans-serif;  
  color:      gray;  
}  
  
h2 {  
  font-family: sans-serif;  
  color:      gray;  
}  
  
p {  
  color: maroon;  
}
```

这个规则会选择<h1>元素，将font-family改为sans-serif，将字体颜色改为gray。稍后还会更详细地讨论字体。

这是另一个规则，对<h2>元素做同样的设置。



实际上，由于这些规则是一样的，可以把它们合并起来，就像这样：

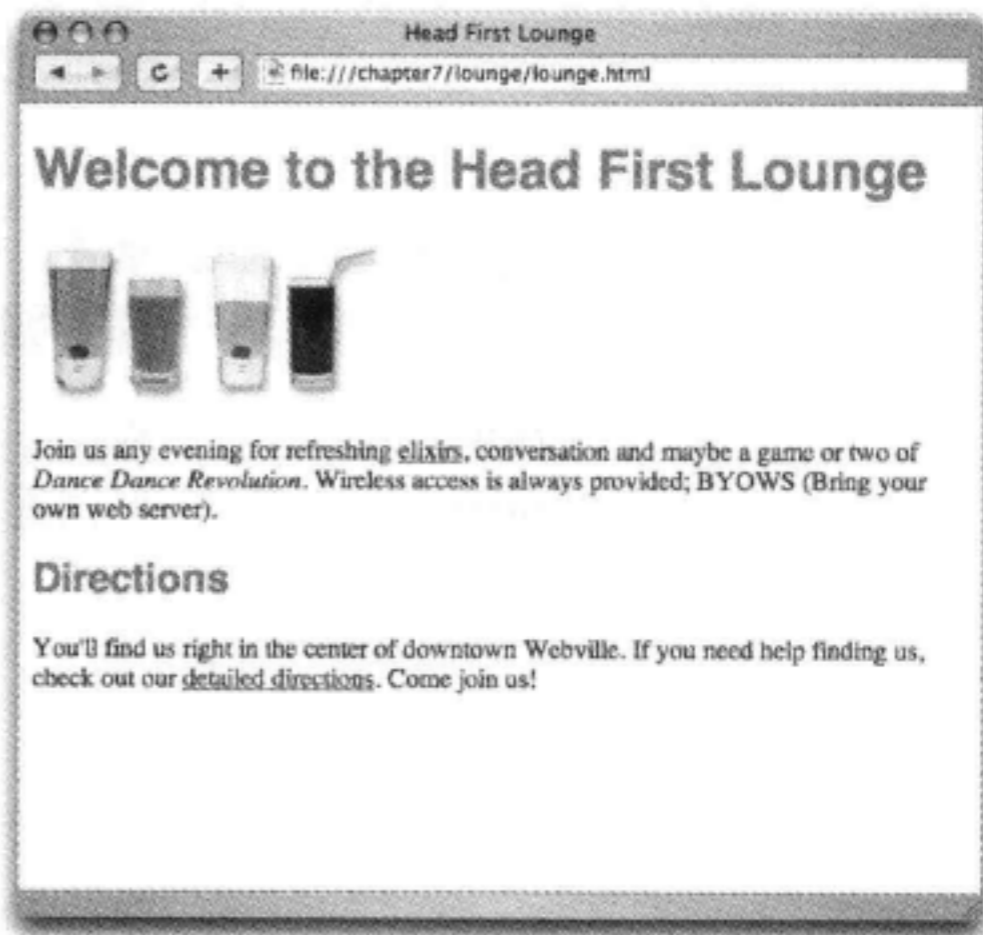
```
h1, h2 {  
  font-family: sans-serif;  
  color:      gray;  
}  
  
p {  
  color: maroon;  
}
```

要为多个元素编写一个规则，只需要在选择器之间加上逗号，如“h1, h2”。

试一试……

把这个新CSS增加到你的“lounge.html”文件中，然后重新加载页面。你会看到，只需要一个规则，可以同时选择<h1>和<h2>标题。

现在页面上的两个标题都指定了新样式，字体为sans-serif，颜色为灰色。



再在欢迎消息下面加一条线

下面再对欢迎标题稍稍做些处理。在下面加一条线怎么样？这样可以在视觉上让主标题更为显眼，效果更好。我们要使用下面这个属性来达到这个目的：

```
border-bottom: 1px solid black;
```

这个属性控制元素下边框的外观。

要设置下边框的样式：粗细为1像素，黑色，实线。

这里有个小麻烦，如果在CSS的“h1, h2”合并规则中增加这个属性和属性值，那么最后两个标题都会加上边框：

```
h1, h2 {
  font-family: sans-serif;
  color: gray;
  border-bottom: 1px solid black;
}

p {
  color: maroon;
}
```

这里增加一个属性，这会同时改变<h1>和<h2>元素的下边框。

如果这样做……

……两个标题都会有下边框。

所以，能不能只对<h1>元素设置下边框，而不影响<h2>元素？如何做到？必须把规则再分开吗？要想知道答案，请翻到下一页……

