

重视大脑的学习指南

Head First jQuery (中文版)



快速实现复杂
的HTML表单



无需烦人的浏览器
插件就能构建
Web应用



只需区区几行代码
就能为Web页面
增加交互性



编写你自己
定制的jQuery
函数



在Web页面中使用
动画和Ajax



O'REILLY® 中国电力出版社

Ryan Benedetti & Ronan Cranley 著
林琪 等译

更多资源请访问稀酷客(www.ckook.com)

Head First jQuery

(中文版)

有没有一本能帮我学习如何使用
jQuery，而且起码要比看牙医更有
趣？可能只是异想天开吧……



Ryan Benedetti
Ronan Cranley 著
林琪 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

更多资源请访问稀酷客(www.ckook.com)

图书在版编目（CIP）数据

Head First jQuery（中文版）/（美）贝内代蒂（Benedetti, R.），（美）克兰利（Cranley, R.）著；林璇等译。—北京：中国电力出版社，2012.10

书名原文：Head First jQuery

ISBN 978-7-5123-3598-1

I. ①H… II. ①贝… ②克… ③林… III. ①JAVA语言－程序设计 IV. ①TP312

中国版本图书馆CIP数据核字（2012）第237424号

北京市版权局著作权合同登记

图字：01-2012-6888号

©2011 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2012.
Authorized translation of the English edition, 2011 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2011。

简体中文版由中国电力出版社出版，2012。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 jQuery（中文版）

书 号/ ISBN 978-7-5123-3598-1

责任编辑/ 刘炽

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/ 880毫米×1230毫米 20开本 27印张 727千字

版 次/ 2013年3月第1版 2013年3月北京第1次印刷

印 数/ 0001—3000册

定 价/ 78.00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

更多资源请访问稀酷客(www.ckook.com)

对《Head First jQuery》的高度赞誉

“利用jQuery，现在JavaScript开发容易多了，即使做一些非凡的事情也易如反掌，真是很神奇。这本书展示了如何迅速解决实际问题。另外还有一个额外奖励，你将从中学到有关JavaScript的主要内容，了解如何建立一个Web开发环境，并掌握一些PHP/MySQL知识。这是一本很实用的书。”

—— Jim Doran, Johns Hopkins大学软件工程师

“很多深奥的编程书通篇都是技术行话，与它们不同，《Head First jQuery》采用一种有趣而且容易理解的方式，指导初学者一步一步地创建他们的jQuery页面。”

—— Lindsey Skouras, 代理人，自学成才的程序员

“这堆技术（jQuery、DOM、Ajax、HTML5、CSS）真像一锅粥，实在让人心生畏惧，不过Ryan Benedetti和Ronan Cranley用巧妙的手法把它们分解成一些容易理解的概念，这确实让学习过程变得有趣得多。”

—— Bill Mietelski, 软件工程师

“JavaScript作为一种优秀的编程语言重出江湖，这在很大程度上归功于一组卓越的插件库的‘加盟’，jQuery就是其中很重要的一分子。《Head First jQuery》通过有重点的阐述，使现代Web开发人员有了一个既动脑又动手的机会来了解这种关键的JavaScript技术。”

—— Paul Barry, 作者，Carlow技术学院计算领域讲师

对其他Head First书的赞誉

“《Head First Object-Oriented Analysis and Design》采用一种全新的视角研究OOA&D。这本书独树一帜之处在于它着眼于学习。作者们不仅让实践者能很好地理解OOA&D的内容，更重要的是能让他们真正使用这些内容。”

—— Ivar Jacobson, Ivar Jacobson Consulting

“我刚刚读完《HF OOA&D》，实在太喜欢这本书了！这本书最让我称道的是它重点讨论了我们为什么要完成OOA&D来编写优秀软件！”

—— Kyle Brown, 杰出工程师, IBM

“在有趣的图片和夸张的字体后面，是对OO分析与设计认真、睿智的介绍，所有内容都做了精心设计。读这本书时，我感觉就像是站在一个资深设计人员的肩上，他一直在为我解释每一步有哪些重要的问题，并且告诉我为什么。”

—— Edward Sciore, 副教授，波士顿学院计算机科学系
波士顿学院

“总之一句话，《Head First Software Development》对任何想要提升编程技术的人来说都是一个绝好的资源，它能让各个不同层次的读者都乐于参与其中。”

—— Andy Hudson, Linux Format

“如果你还是一个软件开发新手，《Head First Software Development》能让你有一个正确的起点。如果你是一个有经验的开发人员（在这个领域已经打拼多年），也不要那么急于把它抛在一边……”

—— Thomas Duff, Duffbert's Random Musings

“《Head First Java》适合所有人。不论是视觉学习者还是动觉学习者，每个人都能从这本书中学到东西。视觉效果可以让知识更容易记住，这本书采用了一种非常易读的方式编写，这与大多数Java手册截然不同……《Head First Java》是一本极有价值的书。我发现《Head First books》已经用于课堂，这包括高中以及成人考试课程。我肯定还会参考这本书，当然也会告诉别人参考这本书。”

—— Warren Kelly, Blogcritics.org

“不再是一种照本宣科式的学习，《Head First iPhone and iPad Development》提供了一种幽默风趣、生动形象甚至充满乐趣的方式来学习iOS开发。这里涵盖了很多关键技术，包括核心数据甚至诸如接口设计等重要方面。这些内容都经过精心挑选，而且都是最顶尖的新技术。你在哪里能看到UIWebView和UITextField居然能坐在一起亲切交谈！”

—— Sean Murphy, iOS设计师和开发人员

对其他Head First书的赞誉

“《Head First Java (第二版)》还有一个优点，它能满足更多人的口味。现在它涵盖了一些更高级的主题，如Swing和RMI，你可能早已经等不及想要深入了解这些API。用Java.net编写长达十万行代码却无一点瑕疵的程序，这会让你名利双收。书中还提供了有关网络和线程的大量资料，甚至一些最佳实践，这是我的弱项。在这方面，我实在要多夸几句，作者们居然用一个20世纪50年代的电话接线员（嗯，你知道的，就是那种戴着耳机人工接线的女生）模拟TCP/IP端口……你真得到书店翻翻《Head First Java (第二版)》。即使你已经了解Java，也可以从中得到一、两个收获。如果没有，不妨随手翻翻，这也很有意思。”

—— Robert Eckstein, Java.sun.com

“当然，并不是内容让《Head First Java (第二版)》独树一帜，它的突出之处在于这本书的风格和方法。利用卡通画、谜题、冰箱磁贴（没错，就是冰箱磁贴），使得这本书绝不是你以前见过的计算机科学教材或技术手册。另外，取代常见的读者练习，这本书要求你假扮成编译器来编译代码，或者可能要通过填空来组织一些代码，或者……你懂的…… 这本书的第一版是我向初学Java和对象的人推荐的书目之一。这个新版本也没有让我失望，它很好地延续了上一版的优点。如果你像很多人一样，看着一本传统的计算机书就昏昏欲睡，相信这本书肯定能让你一直保持清醒，乐于继续学习。”

—— TechBookReport.com

“《Head First Web Design》是掌握所有这些复杂主题并了解Web设计领域最新进展的钥匙。如果你用过类似Dreamweaver的棘手工具并深受折磨，这本书将是掌握优秀Web设计的绝好途径。”

—— Robert Pritchett, MacCompanion

“还能从书中学习真正的Web设计，这可能吗？《Head First Web Design》就是设计用户友好的网站的法宝，让你对网站开发全过程的设计有所了解，从客户需求，到手绘情节图板，再到正常运行的在线网站。这本书与其他教你‘如何构建网站’的书有一个最大的不同，它使用了关于认知科学和学习的最新研究成果，为读者提供了一个包含丰富图片的视觉学习过程。设计这种学习方式就是为了让大脑更好地工作和学习。效果很明显，这将成为Web设计基础的强有力补充，任何通用计算机库都能从中找到成功之路。”

—— Diane C. Donovan, California Bookwatch: The Computer Shelf

“我要向所有涉及艺术方面的程序员同事推荐《Head First Web Design》。”

—— Claron Twitchell, Utah Java用户组

O'Reilly的其他相关图书

jQuery Cookbook
jQuery Pocket Reference
jQuery Mobile
JavaScript and jQuery: The Missing Manual

O'Reilly的其他 Head First 系列图书

Head First C#
Head First Java
Head First Object-Oriented Analysis and Design (OOA&D)
Head First HTML with CSS and XHTML
Head First Design Patterns
Head First Servlets and JSP
Head First EJB
Head First SQL
Head First Software Development
Head First JavaScript
Head First Physics
Head First Statistics
Head First Ajax
Head First Rails
Head First Algebra
Head First PHP & MySQL
Head First PMP
Head First Web Design
Head First Networking
Head First iPhone and iPad Development

本书谨献给諸位JavaScript大师：John Resig (jQuery库的创建人和首席开发人员)、Douglas Crockford、David Flanagan，以及Brandon Eich。

献给我的三个宝贝：Josie、Vin和Shonna。

——Ryan

献给Caitlin和Bono：感谢你们所做的一切！

——Ronan

Ryan



Ryan Benedetti 拥有蒙大纳大学创意写作美术硕士学位，在波特兰大学担任 Web 开发人员和多媒体专家。他对 jQuery、Flash、ActionScript、Adobe 的 Creative Suite、Liferay Portal、Apache 的 Jakarta Velocity 模板语言以及 Drupal 都很有经验。

Ryan 在 Salish Kootenai 学院信息技术和计算机工程系担任了 7 年的系主任。在此之前，他做过编辑，还为蒙大纳大学 Forestry 学院的一项关于河流、小溪和湿地的研究做过信息系统专家。

Ryan 的诗作曾发表在 Cut Bank 和 Andrei Codrescu 的 Exquisite Corpse 上。业余时间里，他喜欢画画、做卡通、玩布鲁斯，还很喜欢打坐。他最喜欢与他的女儿、儿子还有他的宝贝 Shonna 在俄勒冈州的波特兰共度美好时光。他还喜欢带他的动物朋友四处闲逛：Rocky、Munch、Fester 和 Taz。

Ronan



Ronan Cranley 一直在波特兰大学工作，从 Web 开发人员到高级 Web 开发人员/从系统管理员到 Web 和管理系统助理主管，2006 年由爱尔兰的都柏林搬到俄勒冈州的波特兰。

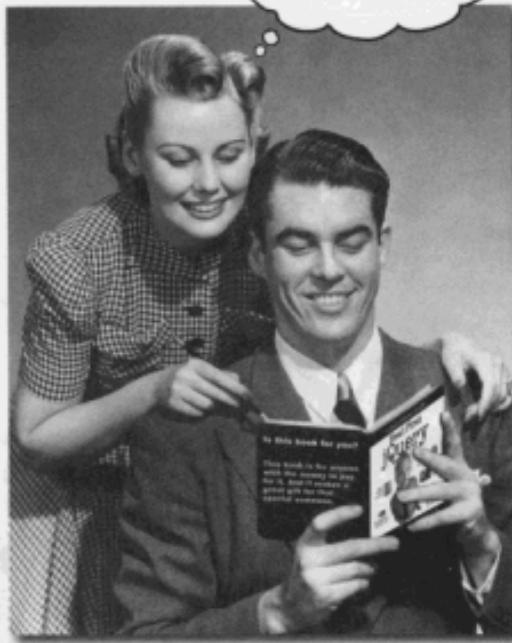
他于 2003 年作为荣誉学生获得了都柏林科技学院的计算机科学学士学位。上大学期间，以及之前在都柏林 ESB 国际公司任职和目前在波特兰大学工作期间，Ronan 曾经参与过用 PHP、VB.NET、C 和 Java 开发的很多不同项目。这包括（但绝不仅限于）一个客户端 GIS 系统、一个自用的内容管理系统、一个日历/调度系统，还有一个 jQuery/Google Maps 混合系统。

除了设计和构建前端 Web 应用，他还担任波特兰大学的 SQL Server DBA。业余时间里，Ronan 会数小时奔跑在足球场上。在高尔夫球场上潇洒挥杆，他喜欢和妻子 Caitlin 还有他们的英国牧羊犬 Bono 遛弯，还喜欢研究西北太平洋。

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jQuery书里！



有个问题总是吵得我们兄弟都磨出茧了：“你们到底为什么要把这样一些东西放在一本jQuery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 以前有Web设计或开发经验吗？
- ② 你是不是想学习、理解、记住并应用重要的jQuery和JavaScript概念，让你的Web页面更有交互性、更让人兴奋？
- 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？

如果仅有编写脚本的经验，这肯定也有帮助。JavaScript经验很有帮助，不过并非绝对必要。

这正是你要的书。

谁可能不适合看这本书？

如果满足下面任何一种情况：

- ① 你是不是对于Web开发一无所知？
- ② 你是不是在开发Web应用，正在找一本关于jQuery的参考书？
- 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，如果加入了大脚怪，这样一本书肯定不是一本正经八百的技术书？

可以参考《Head First HTML with CSS & XHTML》，这本书很好地介绍了Web开发，然后再来加入我们的jQuery之旅。

那么，这本书将不适合你。



【来自市场的呼声：只要买得起、不介意用信用卡还是现金，都可以拥有这本书。现金也不错——小猫】

我们知道你在想什么。

“这算一本正经八百的jQuery开发书吗？”

“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，并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。

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。

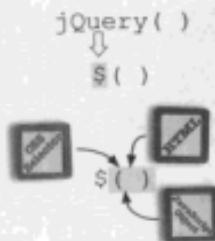
没有一种简单的办法来告诉大脑：“嘿，大脑，真是谢谢你了，不过不管这本书多没意思，也不管现在我对它多么无动于衷，但我确实希望你能把这些东西记下来。”



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学习法则：



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上面，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的问题。

另外，目前宠物猫最爱的是滚动的。能不能让它慢慢滚动，并在这个过程中逐渐淡化呢？

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，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，喋喋不休地对你诉说，在这两个人中，你会更注意哪一个呢？

让学习的人想得更深。换句话说，除非你很积极地让神经系统活动起来，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。



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，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最慢家伙所不知道的，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谈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任，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jQuery开发看做是一只饥饿的老虎呢？

这有两条路，一条比较慢，很乏味。另一条路不仅很快，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，即使再没有意思，你也能学会并记住。如果做了足够的重复，你的大脑就会说，“尽管看上去这对他来说好像不重要，不过，既然他这样一面再、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，而这就会上更多的神经元点火。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。

我想知道怎样才能
通过我的大脑，让它
记住这些东西……



我们是这么做的：

我们用了很多图，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，则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意。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，你的大脑就会投人更大的注意。而这会让你感觉到这些东西更有可能要被记住，其实这种感觉可能只是很点幽默，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，我们加入了大量实践活动，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。

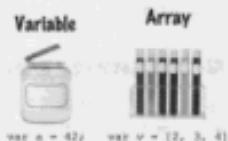
我们采用了多种学习模式，因为尽管你可能想循序渐进地学习，但是其他人可能希望先对整体有一个全面的认，另外可能还有人只是想看一个例子。不过，不管你想怎么学，要是同样的内容能以多种方式来表述，这对每一个人都会有好处。

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开动起来，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，它就更能学会并记住。想想吧，如果只是在体育馆里看着别人流汗，这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陋而让人无从下手。

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，则相对于某件东西而言，你的大脑会更为关注。





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。以下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。要尝试新鲜事物。
把这一节撕下来，贴到你的冰箱上。

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。书中提出问题的时候，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手，这将改善你的学习。

③ 阅读“没有像问题”。

顾名思义。这些问题不是可有可无的旁枝，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，而且可能会有以前光看不说时不曾有的新发现。

⑦ 听听你的大脑怎么说。

注意你的大脑是不是负荷太重了。如果发现自己开始浮光掠影地翻看，或者刚看的东西就忘记了，这说明你该休息一会儿了。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中去，利用你所学的想法完成项目决策。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……通过使用针对考试所学的工具和技术可以更好地解决这个问题。

重要说明

要把这看做一个学习过程，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。

希望你了解HTML和CSS。

如果你不了解HTML和CSS，读这本书之前请先找一本《Head First HTML, with CSS & XHTML》看看。我们会对CSS选择器做一点复习，不过不要指望在这里学到关于CSS所要了解的一切。

不要求你了解JavaScript。

我们知道……这是一个有争议的选择，不过我们确实认为即使之前不了解JavaScript也能学习jQuery。编写jQuery代码之前当然要知道一些JavaScript，不过我们会在介绍jQuery代码的同时教你认识所有那些重要的JavaScript概念。我们衷心相信jQuery口号：少写多做。

建议你学习这本书时最好使用多种浏览器。

建议你用至少3种不同的浏览器测试这本书中的页面。这会让你了解不同浏览器之间的差别，另外也能对创建适用于多种浏览器的页面有所认识。

这里并不讨论Head First浏览器开发工具……

……不过我们希望你知道如何使用这些工具。强烈建议使用Google Chrome，可以在<http://www.google.com/chrome>。要了解以下浏览器和相关开发工具的更多信息，可以访问下面的网站：

Google Chrome	http://code.google.com/chrome/devtools/docs/overview.html
Firefox's Firebug	http://getfirebug.com/wiki/index.php/FAQ
Safari	http://www.apple.com/safari/features.html#developer
Internet Explorer 8	http://msdn.microsoft.com/en-us/library/dd565628(v=vs.85).aspx
Internet Explorer 9	http://msdn.microsoft.com/en-us/ie/aa740478
Opera's Dragonfly	http://www.opera.com/dragonfly/

不要仅限于这本书

要学习一个新知识，最好的办法就是加入一个学习社区。我们认为jQuery社区是技术世界最好、最活跃的社区之一。可以在这里了解更多情况：<http://www.jquery.com>。

书里的实践活动绝不是可有可无的。

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，它们可以帮助你将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。

“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。

软件需求

要编写jQuery代码，你需要有一个文本编辑器、一个浏览器，还需要一个Web服务器（可以本地安装在你的个人台式机上），以及jQuery库。

对于Windows，我们推荐的文本编辑器是PSPad、TextPad或EditPlus（不过，如果别无选择，也可以使用Notepad）。对于Mac，我们推荐的文本编辑器是TextWrangler。如果你使用的是一个Linux系统，其中已经内置了很多文本编辑器，相信不需要我们多说了。

如果你打算完成Web开发，则需要一个Web服务器。在学习后面几章（第9章、第10章和第11章），你要先看看附录，了解如何安装PHP、MySQL和一个Web服务器（Apache或IIS），并按操作说明正确地安装。建议你现在就完成这个工作。说真的，现在就动手，按照操作说明安装开发环境，完成之后再返回这一页。

你还要有一个浏览器，另外需要使用浏览器开发工具。请阅读前一页。现在就该了解如何使用Google Chrome开发工具中的JavaScript控制台。这是留给你完成的家庭作业。

最后，当然还需要jQuery库，翻开下一页，我们会告诉你在哪里可以找到它。

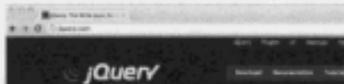
下载jQuery

现在来具体下载jQuery。访问jQuery网站，下载一个jQuery副本本来使用。

第1步：

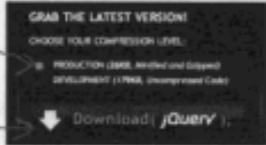
打开你喜欢的浏览器，指向这个地址：

<http://www.jquery.com>



第2步：

找到“Grab the Latest Version!”（获取最新版本！）。然后选中“Production”（产生版本）旁边的复选框。



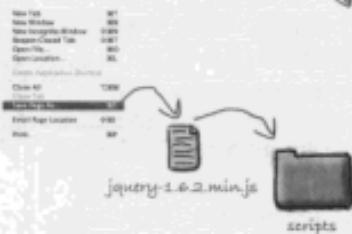
第3步：

单击“Download jQuery”（下载jQuery）按钮。

第4步：

你会看到下一页与下面这个页面类似。

把这个页面保存到硬盘上名为scripts的文件夹中。

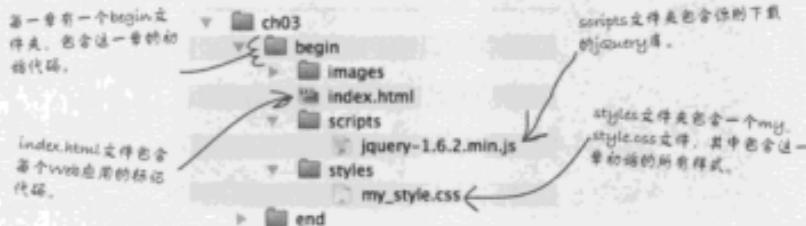


生产版本与开发版本有什么区别？

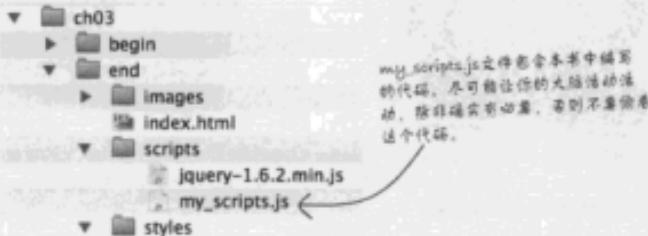
jQuery的生产版本是一个简化版本，主要是为了提高在Web服务器上的执行程度。开发版本则面向那些对探索和扩展jQuery库内部工作感兴趣的开发人员。如果你是那种喜欢查看发动机内部原理的人，则可以把这两个版本都下载下来。

文件夹结构

从Head First labs (<http://www.headfirstlabs.com/books/hfjquery>) 下载并解压缩本书的代码之后，你会看到它分为多个文件夹，按各章来组织。下面以 ch03 为例：



每一章的 end 文件夹包含了这一章最终的代码。建议你只是在需要参考时才使用这个 end 文件夹。



可以在你的任何项目中使用jQuery库。为了方便你，我们在本书的代码文件夹中包含了jQuery库。不过，你要知道从哪里可以得到它，以便jQuery库更新时可以得到最新版本，以及在将来的项目中使用。jQuery开发人员们会定期更新这个库。

技术审校团队

Lindsey Skouras



Paul Barry



Bill Mietelski



Jim Doran



Jim Doran在马里兰州巴尔的摩的Johns Hopkins大学担任软件工程师。他在巴尔的摩郡的社区学院讲授JavaScript，另外还在Web会议上做一些关于jQuery的演讲。除此之外，Jim喜欢在维护他的博客<http://jimdoran.net>，还参加了一个男女选手都参与的轮滑公开赛团队。

Bill Mietelski做过多本Head First书的技术审校。目前他是芝加哥地区一个领先的国际学术医疗中心的软件工程师，主要从事生理统计方面的研究工作。如果没有在收集或处理数据，你可能会看到他活跃在当地一个高尔夫球场上。

Lindsey Skouras是华盛顿地区的一位代理人。她利用空闲时间通过Head First系列自学了如何编写代码。她的兴趣很广泛，包括读书、做手工、参观博物馆，当然还喜欢和她的丈夫和狗狗在一起。

Paul Barry在爱尔兰的Carlow技术学院讲授计算机课程。Paul是Linux Journal杂志的特约编辑，同时也是一位发表过多部专著的技术作者。他是《Head First Python》的作者，并且合作撰写了《Head First Programming》。有空闲时间时，Paul会与SME和新兴公司讨论软件开发项目的有关问题。

致谢

致编辑：

感谢（同时祝贺）Courtney Nash，在她的推动下，我们终于完成了也许是我们的最棒的一本书。长期以来，她一直在忍受着我们的大量邮件、问题、长篇大论，有时甚至有些偏执。她坚持带领我们完成了这本书，一直相信我们有信心克服所有困难。



Courtney Nash ↗

致O'Reilly人员：

感谢Lou Barr高效、绝妙、魔法般的工作，让这本书得以顺利进行，而且如此漂亮。

感谢Laurie Petrycki的支持。Ryan很怀念在波士顿的HF培训经历。永远忘不了在那里Laurie营造的那种家一般的温馨氛围。

感谢Karen Shaner。感谢技术审校小组的所有人。

Ryan忘不了在书店发现Head First系列的那一天。非常感谢Kathy Sierra和Bert Bates创造了这种方式让所有人的神经元活跃起来。感谢Bert倾听我们的长篇大论，使我们能摆脱困境，而且一直能认清我们的目标。

感谢Tim O'Reilly的真知灼见，创造了有史以来最棒的出版公司！

致Ronan的朋友和家人：

要特别感谢我的妻子Caitlin，借助她非凡的设计能力和对Adobe产品的了解，这本书才得以问世。还要感谢她的耐心，没有你我肯定做不到！非常感谢支持我们的所有人，最好的邻居们，波特兰大学的棒球队员们、理解我的足球队和高尔夫球友们。还要感谢我在爱尔兰的家人给予的支持和鼓励。最重要的是，感谢Ryan Benedetti，一流的合作者、同事和朋友。感谢你一直与我同行，给我这个机会。这真是难忘的经历！



Lou Barr ↗

致Ryan的朋友和家人：

感谢我的女儿Josie、我的儿子Vinny和妻子Shonna，感谢在完成这本书期间你们一直以来各种方式的信任和支持。真心感谢你们。我爱你们，我的三个宝贝！

还要感谢我的妈妈和爸爸、我的兄弟Jeff，还有我的侄女Claire和Quinn。感谢我的棒球球友，还有波特兰大学的WAS球队。他们是Jenny Walsh、Jacob Caniparoli和星期二早晨技术小组（你们知道我说的是谁）。感谢Caitlin Pierce Cranley高超的设计水平。还要感谢我的拍档Irish Ninja（也就是Ronan Cranley）。感谢把他娴熟的jQuery、JavaScript和PHP编码技术、幽默感，以及无限的工作热情带入这本书。

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	xxiii
1	jQuery入门：Web页面动作	1
2	选择器与方法：获取，行动	35
3	jQuery事件与函数：在页面上有所作为	75
4	jQuery Web页面管理：修改DOM	123
5	jQuery效果与动画：迈出一小步，前进一大步	175
6	jQuery与JavaScript： 不是 jQuery，我是你父亲！	215
7	定制函数提供定制效果：你最近为我做了什么？	253
8	jQuery与Ajax：请把数据传给我	291
9	处理JSON数据：客户端，来见见服务器	325
10	jQuery UI：表单改头换面	371
11	jQuery和API：对象，无处不在	411
i	其他：十大主题（我们没有谈到的）	447
ii	建立开发环境：整装待发	461
	索引	483

详细目录

引子

你的大脑与jQuery。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jQuery你将无法生存！

谁适合看这本书？	xxiv
元认知：有关思考的思考	xxvii
可以用下面的方法让你的大脑就范	xxix
重要说明	xxx
技术审校团队	xxxiv
致谢	xxxv

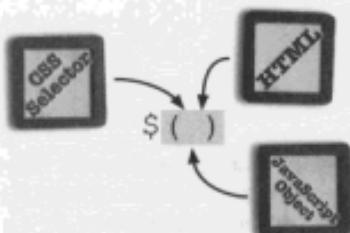
jQuery入门

Web页面动作

想让你的Web页面更出色。

现在HTML和CSS已经在你的掌控之中，你还希望再增加一项编写脚本的技能，不过又不想浪费大好时光一行一行地写脚本。现在需要一个脚本库，能让你动态地修改Web页面。既然是许愿，是不是用Ajax和PHP也能很好地做到？能不能用3行代码就做到大多数客户端语言用15行代码才能完成的事情？美好的愿望？可惜，这是做不到的！你得认识认识jQuery。

jQuery()
↓
\$()



为Web页面赋予生机	2
HTML和CSS很不错，不过……	3
……需要脚本的力量	4
jQuery（和JavaScript）！	5
调查浏览器	7
隐藏的Web页面结构	8
jQuery让DOM不那么吓人	9
到底怎么做的？	11
jQuery选择元素的方式与CSS完全相同	13
样式，来见见脚本	14
jQuery选择器悉听吩咐	15
jQuery翻译	16
你的第一个jQuery任务	20
建立HTML和CSS文件	24
潜入……	26
实现淡入	27
你挽救了宠物朋友活动	30
jQuery工具箱	33

我听到一个神秘的声音。

把这个元素上移。



2

选择器与方法

获取，行动

jQuery可以帮助你获取Web页面元素，用它们完成各种工作。

这一章中，我们将深入研究jQuery选择器和方法。利用jQuery选择器可以获取页面上的元素，利用方法可以对这些元素做一些处理。就像一本大部头的魔法书，jQuery库允许我们动态地做出各种各样的变化。我们可以让图像消失，然后突然又重新出现。我们可以选择一段文本，动画地展示文本字体大小的改变。嗯，演出就要开始了，来获取一些Web页面元素，行动吧！



“Jump for Joy” 需要你的帮助	36
项目需求是什么？	37
深入了解div	39
单击事件特写	42
为页面增加click方法	45
更为特定	47
元素分类	48
用ID标识元素	49
连接Web页面	52
再来看我们的列表	55
创建一些存储空间	56
通过链接混合在一起	57
再来看代码……	58
用append插入消息	59
一切都很好，不过……	61
用\$(this)返回当前元素	63
让\$(this)发挥作用	64
用remove妥善删除	66
深入了解后代选择器	67
回到“Jump for Joy”	73
JQuery工具箱	74

JQuery事件与函数

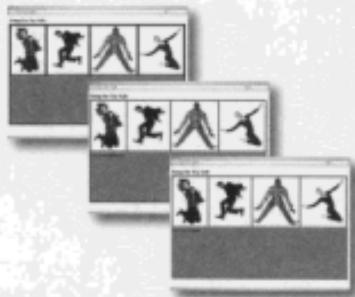
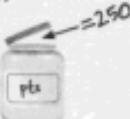
3

在页面上有所作为

利用jQuery，可以轻松地为Web页面添加动作和交互性。这一章中，我们将了解人们与页面交互时如何让页面做出反应。通过运行代码来响应用户的动作，这会让你的网站更上一层楼。我们还会讨论如何建立可重用的函数，这样就可以编写一次代码，然后多次使用。



```
var pts = 250;
```



你的jQuery技能炙手可热	76
管钱的人总是见解独到……	77
让页面响应事件	79
事件监听者的幕后	80
绑定事件	81
触发事件	82
删除事件	86
遍历处理动作元素	90
项目结构	96
利用函数实现功能	100
函数剖析	101
匿名函数	102
命名函数作为事件处理程序	103
向函数传递变量	106
函数还可以返回值	107
使用条件逻辑做判断	109
Jump for Joy还需要更多帮助	113
方法可以改变CSS	115
增加一个hover事件	117
就快完成了……	119
jQuery工具箱	122

4

jQuery Web页面管理

修改DOM

页面加载完成并不意味着它会保持相同的结构。第1章中我们已经了解页面加载时如何创建DOM来建立页面的结构。这一章中，我们将介绍如何使用jQuery在DOM结构中上下移动，以及如何处理元素层次结构和父/子关系来动态改变页面结构。



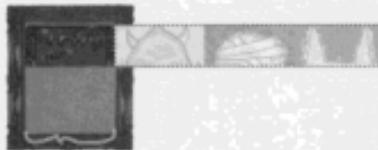
Web馆餐厅需要一个交互式菜单	124
爱素食	125
对元素分类	130
增加按钮	133
接下来呢？	135
在DOM树上荡来荡去	140
用遍历方法在DOM树上爬行	141
方法串链，爬得更远	142
变量还可以存储元素	149
美元符又出现了……	150
数组扩展存储选项	151
在数组中存储元素	152
用replaceWith替换元素	154
replaceWith对我们有什么帮助？	155
使用replaceWith之前先想想	157
replaceWith并不适用每一种情况	158
在DOM中插入HTML内容	159
使用过滤器方法缩小选择范围（第1部分）	161
使用过滤器方法缩小选择范围（第2部分）	162
放回汉堡	165
牛肉（嗯……肉）在哪里？	166
肉类数组	167
each方法循环处理数组	168
这样……对吗？	171
jQuery工具箱	174

5 JQuery效果与动画

迈出一小步，前进一大步

页面动作要漂亮。

不过如果页面本身不漂亮，人们可能根本不会使用你的网站。这就要引入jQuery效果和动画。这一章中，你将了解如何让页面上的元素随时间变换、显示或隐藏相关的特定元素部分，以及缩放页面上的元素，所有这些都会在用户面前闪亮进行。你还会看到如何调度这些动画，使它们按不同的间隔发生，为页面赋予十足的动感。



DoodleStuff需要一个Web应用	176
怪兽变脸	177
怪兽变脸需要布局和定位	178
再来一点结构和样式	181
实现界面点击	182
制造闪电效果	187
jQuery如何动画显示元素？	188
淡入淡出效果动画要改变CSS opacity属性	189
滑动的关键是height属性	190
实现淡入淡出效果	192
用方法串链结合效果	193
用定时函数反复闪烁	194
在脚本中增加闪电函数	197
用animate完成DIY效果	199
什么能动画，什么不能？	200
随时间动画改变样式	202
到底从哪里到哪里？	205
绝对 vs. 相对元素移动	206
利用操作符组合实现元素的相对移动	207
在脚本中增加animate函数	209
看呐，妈妈，不是Flash！	212
jQuery工具箱	214



6

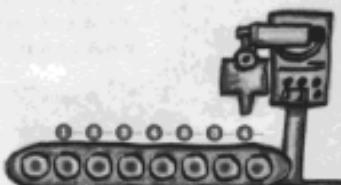
jQuery与JavaScript

卢克 jQuery，我是你父亲！

jQuery还不能完全独立。尽管这是一个JavaScript库，不过遗憾的是，它还不能完成它的父语言能够做到的所有事情。这一章中，我们将了解创建真正高水平网站需要的一些JavaScript特性，并介绍jQuery如何使用这些特性来创建定制列表和对象，以及如何循环处理这些列表和对象来简化你的工作。



充实Head First Lounge	216
对象提供更聪明的存储	218
建立你自己的对象	219
用对象构造函数创建可重用的对象	220
与对象交互	221
建立页面	222
数组归来	225
访问数组	226
在数组中增加和更新元素	227
反复（再反复……）地完成动作	229
干草堆里找根针	232
又到了……判断时间！	239
比较和逻辑操作符	240
用jQuery清空……	246
增加兴奋点	250
jQuery/JavaScript工具箱	252

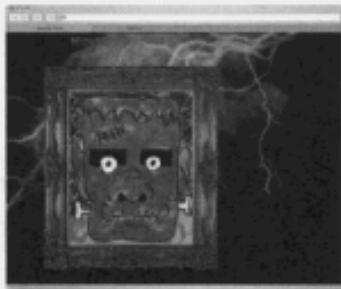


7

定制函数提供定制效果

你最近为我做了什么？

结合jQuery的定制效果和JavaScript函数，可以让你的代码和Web应用更高效、更有用、更强大。这一章中，将通过处理浏览器事件、使用定时函数，以及优化定制JavaScript函数的组织和可重用性，进一步改进jQuery效果。



暴风雨来了	254
我们创建了一个怪兽……函数	255
用Window对象控制定时效果	256
用onblur和onfocus响应浏览器事件	259
定时器方法告诉函数何时运行	263
编写stopLightning和goLightning函数	266
怪兽变脸应用的特性需求	274
下面（再）来点随机性	275
你已经知道当前位置……	276
……以及getRandom函数	276
相对于当前位置移动	280
怪兽变脸2大获成功！	289
jQuery工具箱	290



牙齿是我最明显的特征，
不过我还可以用脸上的
其他部分表现我的咬合！



8

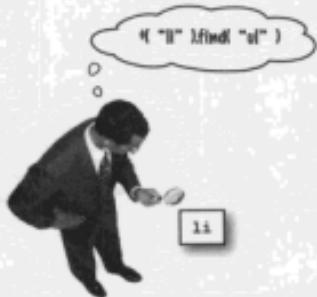
jQuery与Ajax

请把数据传给我

用jQuery做些酷炫的CSS和DOM处理很有意思，不过接下来可能还需要从一个服务器读取和显示信息（或数据）。甚至可能还要用从服务器读取的信息更新页面的某些部分，而无需重新加载页面。来认识认识Ajax，结合jQuery和JavaScript，它能很好地完成这些工作。这一章中，我们将学习jQuery如何向服务器做出Ajax调用，还会了解利用返回的信息能够做什么。



Bit to Byte长跑大赛与日俱进	292
看看去年的页面	293
力求动态	296
老Web，来认识一下新Web	297
了解Ajax	298
什么是Ajax?	298
X是什么	299
用Ajax方法获取数据	304
解析XML数据	306
调度页面上的事件	310
自引用函数	311
从服务器得到更多信息	314
什么时间?	315
关闭页面上的调度事件	320
jQuery/Ajax工具箱	324



处理JSON数据

9

客户端，来见见服务器

从XML文件读数据尽管很有用，不过并不总符合要求。利用一种更高效的数据交换格式（JavaScript对象记法，JavaScript Object Notation，简称为JSON），可以更容易地从服务器端获取数据。而且，与XML相比，JSON更容易生成和读取。通过使用jQuery、PHP和SQL，你将了解如何创建一个数据库存储信息，以便以后利用JSON获取以及使用jQuery在屏幕上显示。真正的Web应用绝对能力超凡！



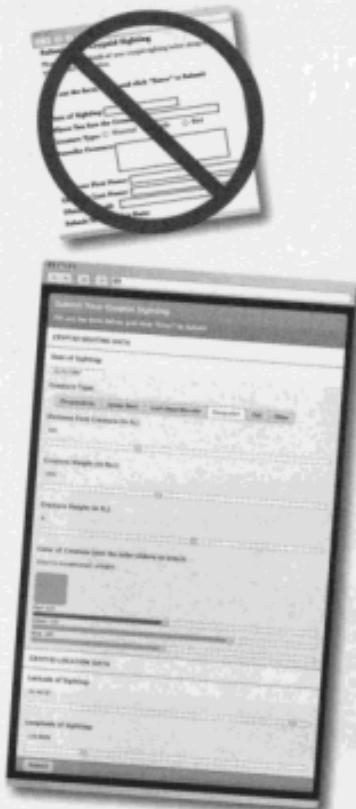
Web慎MegaCorp公司市场部不了解XML	326
XML错误会破坏页面	327
从Web页面收集数据	328
如何处理数据	331
发送数据前先格式化	332
向服务器发送数据	333
将数据存储在MySQL数据库中	335
创建数据库存储运动员信息	336
insert语句剖析	338
使用PHP访问数据	341
在服务器上处理POST数据	342
用PHP连接数据库	343
使用select从数据库读取数据	345
用PHP获取数据	347
JSON来拯救！	350
jQuery + JSON = 超级棒	351
一些PHP规则……	352
更多PHP规则……	353
使用PHP格式化输出	354
访问JSON对象中的数据	361
PHP的数据清理和验证	364
jQuery/Ajax/PHP/MySQL工具箱	369

10

JQuery UI

表单改头换面

Web的生死全由用户和数据决定。从用户那里收集数据是很重要的事情，对Web开发人员来说可能是一个很耗费时间的难题。你已经看到jQuery可以帮助Ajax、PHP和MySQL Web应用更有效地工作。现在来看jQuery对于建立用户界面来创建表单收集用户数据又有什么帮助。在这个过程中，你会得到一剂jQuery UI良药，这就是jQuery的官方用户界面库。



Cryptozoologists.org需要改头换面	372
美化HTML表单	373
用jQuery UI减轻编码负担（还能节约时间）	376
jQuery UI包的内部	380
观测表单中建立一个日期选择工具	381
幕后的jQuery UI	382
部件有可定制的选项	383
建立按钮样式	386
用滑动条控制数值输入	390
计算机使用红、绿和蓝混合颜色	399
建立refreshSwatch函数	402
还有最后一个小问题……	406
jQuery工具箱	410



jQuery和API



对象，无处不在

尽管你是颇有天赋的开发人员，但也不能一直单枪匹马……

我们已经看到，可以包含jQuery插件（比如jQuery UI或标签页导航），这样不需要花太多功夫就能大大提升我们的jQuery应用。为了让我们的应用更上一个台阶，互联网上已经有一些相当酷的工具可以使用。还可以使用一些软件巨头提供的信息，比如Google、Twitter或Yahoo!。我们还需要……更多。这些公司为他们的服务提供了API（应用编程接口），可以在你的网站中包含这些API。这一章中，我们将介绍一些API基础知识，还会使用一个相当常用的API：Google Maps API。



汽车老大脚怪在哪里？

Google Maps API

API使用对象

在页面中包含Google地图

用SQL和PHP获取JSON数据

地图上的点是标记

多生物列表

监听地图事件

大功告成！

jQuery API工具箱

412

414

415

417

420

424

428

438

442

445



其他

十大主题（我们没有谈到的）

毕竟，还有很多内容我们没有谈到。还有很多其他jQuery和JavaScript小点心我们没办法都塞在这本书里。如果完全不提及有些不太公平，所以这里会简单做个介绍，让你能做更好的准备。应对旅程中可能遇到的其他jQuery问题。

#1. jQuery库全貌大观	448
#2. jQuery CDNs	451
#3. jQuery命名空间：noConflict方法	452
#4. 调试jQuery代码	453
#5. 高级动画：队列	454
#6. 表单验证	455
#7. jQuery UI效果	456
#8. 创建你自己的jQuery插件	457
#9. 高级JavaScript：闭包	458
#10. 模板	459



建立开发环境

整装待发

你需要一个环境来实践刚刚掌握的PHP技能，但又不希望影响Web上的实际数据。最好能有一个安全的地方开发你的PHP应用，然后再（在Web上）对外发布，这是一个很好的想法。本附录包含了安装Web服务器、MySQL和PHP的操作说明，能提供一个安全的场所让你开展工作，着手实践。

创建一个PHP开发环境	462
看看你有什么	462
有Web服务器吗？	463
有PHP吗？哪个版本？	463
有MySQL吗？哪个版本？	464
开始安装Web服务器	465
Apache安装……完毕	466
PHP安装	466
PHP安装步骤	467
PHP安装步骤……完毕	468
安装MySQL	468
在Windows上安装MySQL的步骤	469
在Mac OS X上启用PHP	474
在Mac OS X上安装MySQL的步骤	474

Web页面动作 *



想让你的Web页面更出色。现在HTML和CSS已经在你的掌控之中，你还希望再增加一项编写脚本的技能，不过又不想浪费大好时光一行一行地写脚本。现在需要一个脚本库，能让你动态地修改Web页面。既然是许愿，是不是用Ajax和PHP也能很好地做到？能不能用3行代码就做到大多数客户端语言用15行代码才能完成的事情？美好的愿望？可惜，这是做不到的！你得认识认识jQuery。

更多资源请访问稀酷客(www.ckook.com)

为Web页面赋予生机

你已经知道如何使用简洁的常规HTML和CSS构建漂亮的Web页面。不过静态Web页面不再是全部，人们还希望Web页面能够做出响应。他们希望要有动作、动画、交互，还有各种各样的炫酷效果。



Sharpen your pencil



是不是想要控制你的Web页面，让访问者能体会到它更有用？从下面的列表选出你认为适用的选项：

- 动态地向Web页面增加元素，而不必每次都重新加载页面。
- 用户鼠标移动到菜单项上时改变菜单项。
- 一个表单域漏填时会提醒用户。
- 为文本和图片增加动作和渐变。
- 只在用户需要时才从服务器加载数据。

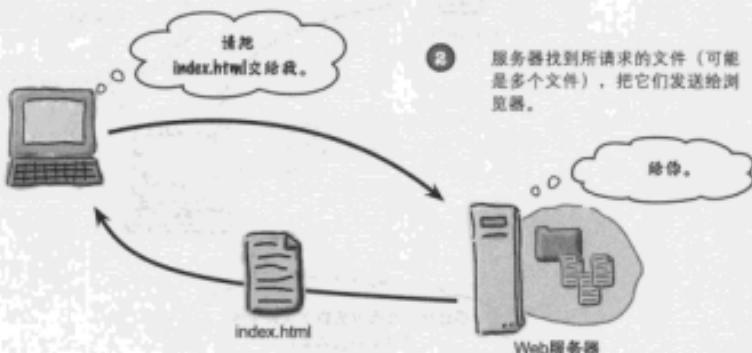
更多资源请访问稀酷客(www.ckook.com)

HTML和CSS很不错，不过……

常规HTML和CSS对于指定页面结构和样式确实很不错。一旦绘制了HTML页面，它就会很漂亮地出现在你眼前，只不过是静态的。

如果你想改变页面的外观，或者想要增加或者从中删除元素该怎么办呢？要么必须做些复杂的CSS处理，要么就不得不加载一个新页面。而且页面很快就会变得很丑陋。为什么？这是因为HTML和CSS所做的实际上只是在控制页面如何显示。

- ① 有人在浏览器的URL地址栏中输入一个Web地址时，浏览器会从服务器请求一个Web页面。



- ② 浏览器根据服务器发送的文件显示一个HTML页面。



更多资源请访问稀酷客(www.ckook.com)
你现在的位置：

……需要脚本的力量

要想动态修改Web页面而又不希望重新加载页面，这就需要与浏览器交互。怎么做到呢？可以利用一个名为<script>的HTML标记。



这个问题问得好。要记住，**HTML**是一种处理文档结构的标记语言。

而层叠样式表（Cascading Style Sheets, CSS）只是用来控制这些元素的外观和位置。HTML和CSS能控制Web页面如何构建和显示，但是它们都无法为Web页面增加行为。要想增加行为，我们需要一个脚本语言，jQuery正是我们需要的。

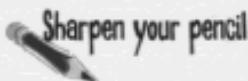
jQuery（和JavaScript）！

向浏览器下达指令所用的语言是JavaScript。所有浏览器都提供了一个内置的JavaScript解释器，它会得到你在<script>标记之间编写指令，并把这些指令翻译成Web页面上的不同动作。



要向解释器下达指令，归根结底还是要讲JavaScript。不过别担心！这里jQuery就要派上用场了。jQuery是一个专门用来动态改变Web页面文档的JavaScript库。下面就来看一些jQuery脚本。

jQuery是一个专门用来动态改变Web页面文档的JavaScript库。



下面的脚本会动态改变一个Web页面。仔细阅读每一行代码，根据你对HTML和CSS的了解，考虑这里的每一行代码可能做什么。然后把你的想法写下来。如果你不太确定某一行代码的作用，则完全可以大胆去猜。我们已经帮你写出了其中一行代码要做什么。

```
<script>
$(document).ready(function() {
    $("button").click(function() {
        $("h1").hide("slow");
        $("h2").show("fast");
        $("img").slideUp();
    });
});
</script>
```

Web页面文档准备就绪时，做下面的工作。

更多资源请访问稀酷客(www.ckook.com)



Sharpen your pencil Solution

下面的脚本会动态改变一个Web页面。仔细阅读每一行代码，根据你对HTML和CSS的了解，考虑这里的每一行代码可能做什么。然后把你的想法写下来。如果你不太确定某一行代码的作用，就完全可以大胆去猜。下面给出我们的答案。

```
<script>
$(document).ready(function() {
  $("button").click(function() {
    $("h1").hide("slow");
    $("h2").show("fast");
    $("img").slideUp();
  });
});
</script>
```

Web页面文档准备就绪时，做下面的工作。

单击任何按钮元素时，完成以下任务。

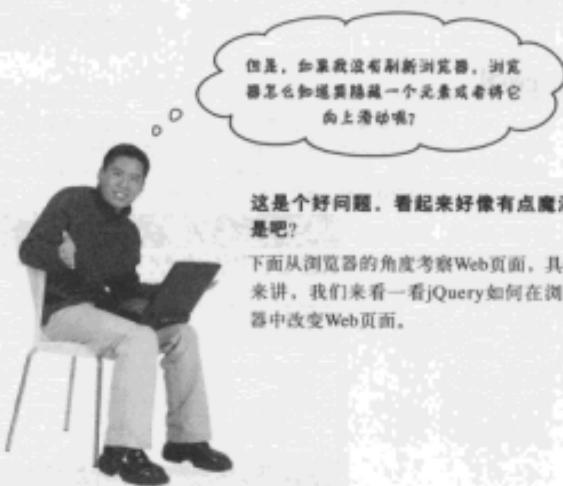
让所有h1元素慢慢从页面消失。

让所有h2元素快速出现在页面上。

让所有img元素向上滑动并消失。

结束按钮click函数。

结束文档ready函数。



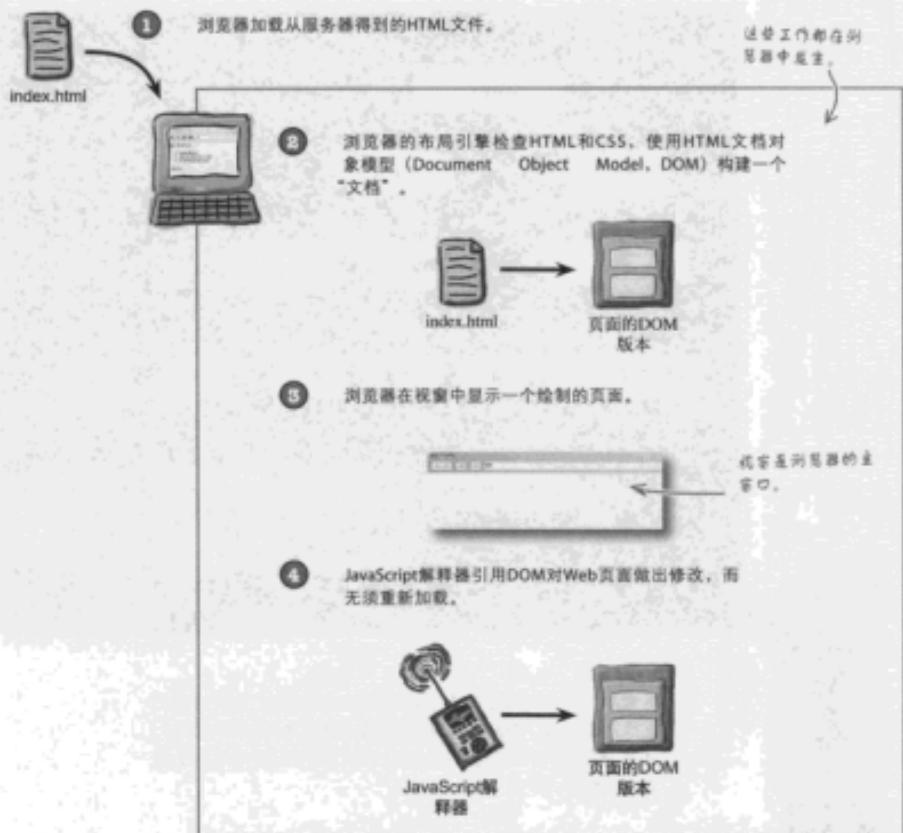
但是，如果我没有刷新浏览器，浏览器怎么知道要隐藏一个元素或者将它向上滑动呢？

这是个好问题。看起来好像有点魔法，是吧？

下面从浏览器的角度考察Web页面。具体来讲，我们来看一看jQuery如何在浏览器中改变Web页面。

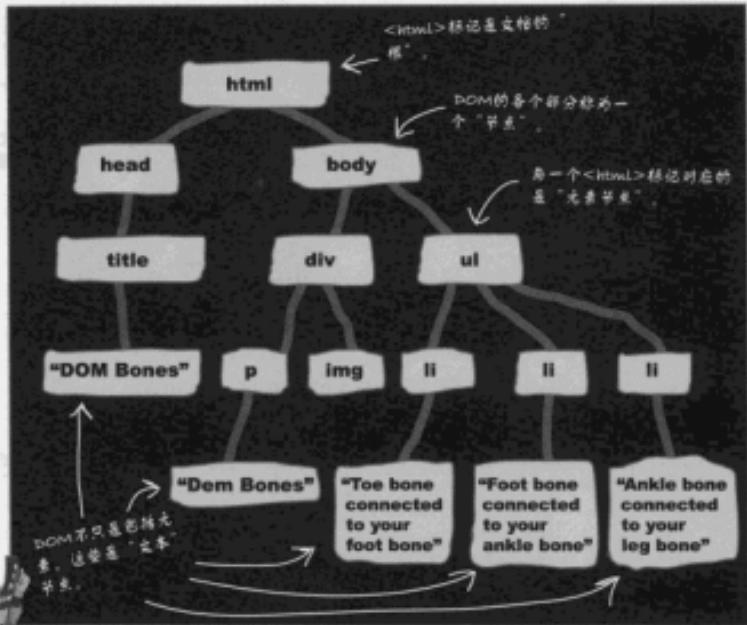
调查浏览器

现在拉开大幕，看看浏览器显示一个Web页面时后台到底发生了什么。浏览器使用HTML文档对象模型（Document Object Model, DOM）将一个只包含HTML标记和CSS代码的页面构建为一个可点击的页面，其中包含文本、图像、视频和我们喜欢浏览的所有其他内容。



隐藏的Web页面结构

多年以来，DOM一直在帮助HTML、CSS和JavaScript更有效地协同工作。它提供了一个标准化的框架，所有现代浏览器都使用这个框架，以便更有效地浏览Web。很多人把DOM认为是一棵树：它有一个根，还有分支，分支的末端是节点，或者你也可以把它看做一个X光片，可以看到页面是如何构造的。



X光片可以告诉医生病人身体中看不到的结构出现的病变。与X光片类似，DOM会向我们显示出页面底层隐藏的结构。不过与X光片不同的是，JavaScript和jQuery使用DOM来改变页面的结构。

更多资源请访问稀酷客(www.ckook.com)

JQuery让DOM不那么吓人

DOM看上去可能很复杂，很恐怖。好在jQuery让一切变得很简单。不要忘了：jQuery就是JavaScript，只不过是一个好用得多的版本。想要控制DOM时，有了jQuery就容易得多了。例如，假设我们想改变页面上唯一一个段落元素中的HTML。

原始JavaScript方法

```
我在局部文档（也就是
DOM中的p）交互。
↓
document.getElementsByTagName("p")
[0].innerHTML = "Change the page.";
```

我需要0个元素。
把这个元素中的
HTML设置为……
……这个字符串。

jQuery方法

```
给我一个段落元素。
↓
$("p").html("Change the page.");
```

把这个元素的HTML设
置为括号内的内容。
jQuery使用了一个“选择器引
擎”，这意味着你可以像CSS
中一样使用选择器获得元素。

或者假设我们想改变页面上5个段落元素中的HTML：

```
根据指定版本的元素
的那个数循环处理。
↓
for (i = 0; i <= 4; i++)
{
    document.getElementsByTagName("p")
[1].innerHTML="Change the page";
```

我正在循环
处理的元素。

因为jQuery使用CSS选择器，所以
可以与上面的代码完全相同。

```
$("p").html("Change the page.");
```

利用jQuery，处理DOM时不必了解它的每个细节，这正是jQuery的主要优点之一。在底层，JavaScript会承担具体的工作。通过本书的学习，你会了解如何结合使用JavaScript和jQuery。第6章我们还会更多地了解jQuery与JavaScript的关系。这个过程中将不断增强我们的JavaScript技能。目前，需要处理DOM时都会使用jQuery。

下面带jQuery到DOM旋转一转，好吗？

更多资源请访问稀酷客(www.ckook.com)



成品代码

在一个文本编辑器中输入下面的代码。然后保存，再在浏览器中打开。尝试单击各个按钮(可以看看代码，试着搞清楚它们在做什么，这绝对没有坏处)。

```
<!DOCTYPE html>
<html><head> <title>jQuery goes to DOM-ville</title>
<style>
    #change_me {
        position: absolute;
        top: 100px;
        left: 400px;
        font: 24px arial;
    }

    #move_up #move_down #color #disappear {
        padding: 5px;
    }
</style>
<script src="scripts/jquery-1.6.2.min.js"></script>
</head>
<body>

    <button id="move_up">Move Up</button>
    <button id="move_down">Move Down</button>
    <button id="color">Change Color</button>
    <button id="disappear">Disappear/Re-appear</button>

    <div id="change_me">Make Me Do Stuff!</div>
    <script>
        $(document).ready(function() {
            $("#move_up").click( function() {
                $("#change_me").animate({top:30},200);
            });//end move_up
            $("#move_down").click( function() {
                $("#change_me").animate({top:500},2000);
            });//end move_down
            $("#color").click( function() {
                $("#change_me").css("color", "purple");
            });//end color
            $("#disappear").click( function() {
                $("#change_me").toggle("slow");
            });//end disappear
        });//end doc ready
    </script>
</body>
</html>
```



index.html

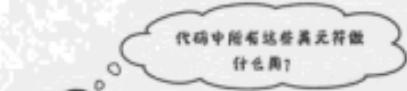
更多资源请访问稀酷客(www.ckook.com)

到底怎么做的？

jQuery对页面的处理真是灵巧，不是吗？重要的是，要记住单击各个按钮时原先的所有HTML和CSS都没有改变。那么到底jQuery是怎么做的呢？下面来看一看：



更多资源请访问稀酷客(www.ckook.com)



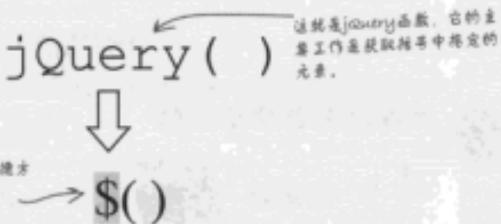
代码中所有这些美元符做什么用？

美元符表示你要用新掌握的jQuery技能赚到的钞票。呵呵，开玩笑的，不过jQuery世界里美元符确实无处不在。

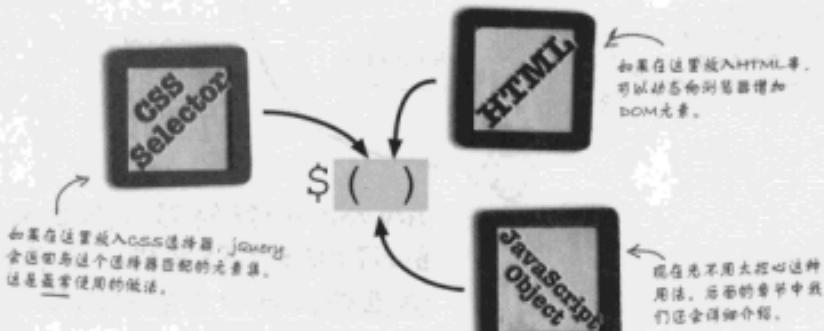
jQuery函数（和快捷方式）

美元符加括号是jQuery函数的简写名。利用这个快捷方式，我们就不需在每次调用jQuery函数时都写上“`jQuery()`”。jQuery函数通常还称为jQuery包装器。

这是jQuery快捷方式。不用输入“`jQuery`”这两个字母，使用快捷方式只需要输入一个字母。

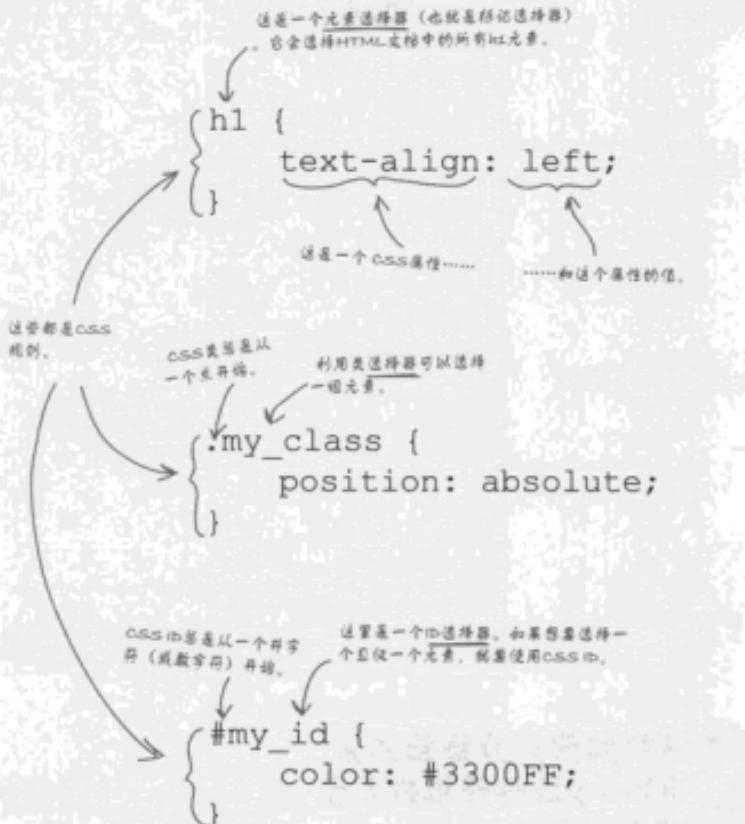


短名和长名指示的东西都是一样的：都是一个名为jQuery的庞大代码块。在这本书中，我们将采用快捷方式。jQuery函数中可以放入3种不同的内容。



JQuery选择元素的方式与CSS完全相同

也许你还没有意识到，实际上你对jQuery已经有不少了解，可能比你自认为的要多得多。要使用jQuery获取元素，主要方法是使用选择器，这正是CSS中使用的同样的选择器。如果你对CSS选择器还不太清楚，没关系。下面来简单复习一下。



样式，来见见脚本

关于jQuery还有一点很棒，它能使用CSS选择器来管理页面上的元素，指定页面样式时所用的也是同样的CSS选择器。

CSS选择器

元素选择器

```
h1 {
    text-align: left;
}
```

类选择器

```
.my_class{
    position: absolute;
}
```

ID选择器

```
#my_id {
    color: #3300FF;
}
```

jQuery选择器

jQuery元素选择器

```
$("h1").hide();
```

这个限制表面上
的所有h1元素。

jQuery类选择器

```
$(".my_class").slideUp();
```

所有作为CSS类以
class成员的元素都向
上滑动。

jQuery ID选择器

```
$("#my_id").fadeOut();
```

这个jQuery语句将CSS ID
为my_id的元素淡出。看到
不可见。

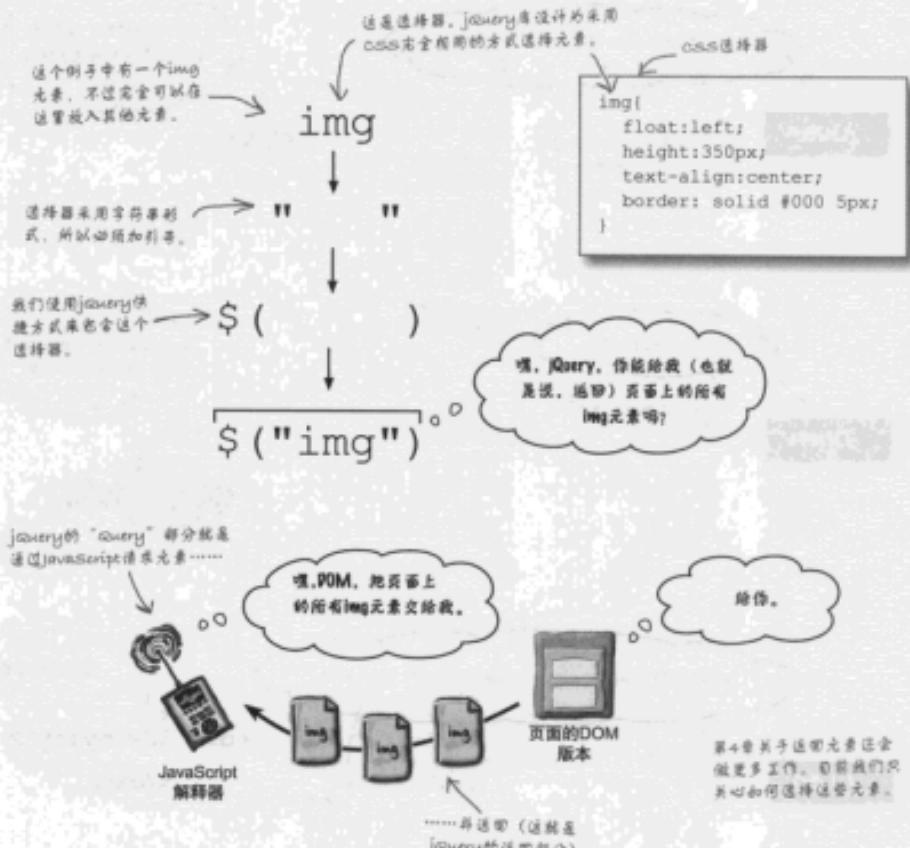
CSS选择器选择元素来为这些元素增加样式；jQuery选择器选择元素来为这些元素增加行为。

第2章和本书后面还会结合选择器和方法完成更多工作。

更多资源请访问稀酷客(www.ckook.com)

jQuery选择器悉听吩咐

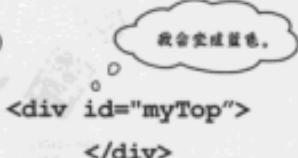
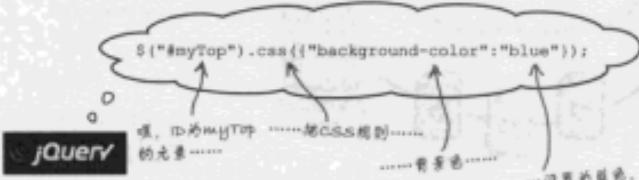
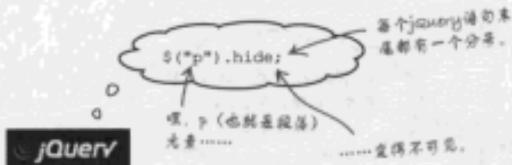
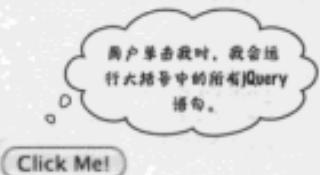
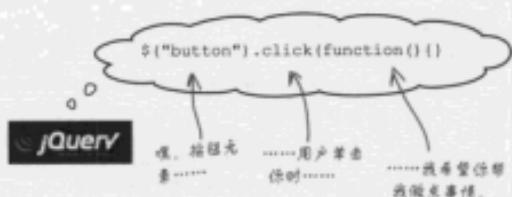
顾名思义，jQuery的重点就是查询。你用选择器请求一个元素时，JavaScript解释器会请求DOM为你获取这个元素。如果你请求一个带嵌套元素的元素，则jQuery还会把嵌套元素也交给你。下面进一步分解一个jQuery选择器，从而能充分了解它是如何工作的。



更多资源请访问稀酷客(www.ckook.com)

jQuery翻译

为了说明学习jQuery是多么容易，下面对DOM王国中使用的一些jQuery语句做个剖析。



扮演浏览器

你的任务是扮演浏览器，输出jQuery语句（左侧）影响的HTML元素（右侧）。



jQuery語句

```
<p>One morning, when Gregor Samsa woke from  
troubled dreams...</p>  
  
<p>He found himself transformed in his bed into  
a horrible vermin.</p>  
  
<p>He lay on his armour-like back, and if he  
lifted his head a little...</p>
```

Nel Mezzo del cammin di nostra vita

In the middle of this road
called "our life"

mi ritrovai per una selva oscura

<p id="mytext">One morning, when Gregor Samsa
woke from troubled dreams...
</p>

`<p id="mytext">he found himself transformed in his bed into a horrible vermin.</p>`

<p>He lay on his armour-like back, and if he lifted his head a little... </p>

更多资源请访问稀酷客(www.ckook.com)

扮演浏览器答案

你的任务是扮演浏览器，指出jQuery语句（左侧）影响的HTML元素（右侧）。
以下是我们答案。



jQuery语句

HTML元素

```
$("p").hide();
```

<p>One morning, when Gregor Samsa woke from troubled dreams...</p>

<p>he found himself transformed in his bed into a horrible vermin.</p>

<p>He lay on his armour-like back, and if he lifted his head a little... </p>

```
$("span.English").toggle();
```

Nel Mezzo del cammin di nostra vita

In the middle of this road called "our life"

mi ritrovai per una selva oscura

```
$("#mytext").show();
```

<p id="mytext">One morning, when Gregor Samsa woke from troubled dreams...</p>

<p id="mytext">he found himself transformed in his bed into a horrible vermin.</p>

<p>He lay on his armour-like back, and if he lifted his head a little... </p>

there are no
Dumb Questions

问：既然jQuery所做的工作无非就是用一用JavaScript，又何必另外创造jQuery呢？JavaScript自己不就足够了吗？

答：JavaScript在很多方面都很出色，特别是管理DOM，不过它太复杂了。即使是底层的DOM管理也绝不简单，正因如此才有了jQuery。它把处理DOM的大量复杂性抽离出来，使得创建效果变得超级容易（jQuery由John Resig始创，可以在这里更多地了解他的情况：<http://ejohn.org/about/>）。

问：这些美元符到底做什么？

答：这只是一个快捷方式。这样你就不用反复地写“jQuery”了！不过，处理其他客户端语言时，使用jQuery()有助于避免命名冲突。

问：之前你也提到过“客户端脚本”，刚才又讲到“客户端”，这到底是什么？

答：Web开发人员通常把Web浏览器称为一个客户端，因为它会消费（Web）服务器的数据。客户端脚本语言可以在后台向浏览器下达指令，服务器端语言则向服务器下达指令。我们将在第8章和第9章介绍有关的更多内容。

问：整个DOM从哪里来的？

答：问得好。Web开发人员和设计人员已经受够了浏览器之间的不一致性，最后决定需要一个标准，能够用来为所有浏览器上的Web页面增加行为，并与之交互。World Wide Web协会（也就是W3C）与相关的组织合作定义了这个标准。可以在这里了解更多有关信息：<http://w3.org/dom>。

问：我要下载jQuery时，会有一个生产版本和一个开发版本。这两个版本之间有什么区别？

答：生产版本是缩减版，这意味着其中删除了大量不必要的字符串和空白符。另外生产版本得到了优化，可以在生产环境中更快地运行。不过要了解发生了什么会比较困难。开发版本中适当地留有空白和间隔，所以更可读。这个版本主要面向那些希望深入jQuery代码来做出修改甚至扩展的人（毕竟，jQuery是开源的）。

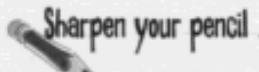
你的第一个jQuery任务

作为Web镇宠物救援基金会新任的Web开发人员，你刚接手一项任务。市场部门希望启动年度募捐活动，改造去年的“帮帮我们的宠物朋友”Web页面。他们给你看了去年的一个截图，还给出了详细说明，告诉你希望这个页面做什么。



嗯，没人希望第一天就让市场部门失望。你可不希望他们对你印象不好！所以下面就来看看这里要做些什么……

更多资源请访问稀酷客(www.ckook.com)



了解如何向页面增加jQuery功能之前，先来看看如何建立HTML和CSS。下面给出了去年活动的相关文件。你认为需要哪些元素，在这些元素旁边写出要提供市场部门所需的功能必须做些什么。我们已经帮你完成了一项。

```
<!DOCTYPE html><html> <head>
<title>Furry Friends Campaign: jQuery
Proof-of-Concept</title>
<link rel="stylesheet" type="text/css"
href="styles/my_style.css">
</head>
<body>
<div id="showfriend">
<a href="#">Our Furry Friends Need
Your Help

</a>
</div>
```



锚标记在CSS中设置了“hover”
和“active”状态。用户将鼠标停留在链接
上时，图像就会出现。

```
a:link img, a:visited img {
display:none;
}
```



```
a:hover img, a:active img {
display:block;
}
a{
text-decoration:none;
color: #000;
}
```



Sharpen your pencil Solution

了解如何向页面增加jQuery功能之前，先来看看如何建立HTML和CSS。下面给出了去年活动的相关文件。你认为需要哪些元素，在这些元素旁边写出要提供市场部门所需的功能必须做些什么。以下是我们答案，不过如果你的答案与我们的答案不完全相同也没有关系。

```
<!DOCTYPE html><html> <head>
<title>Furry Friends Campaign: jQuery
Proof-of-Concept</title>
<link rel="stylesheet" type="text/css"
href="styles/my_style.css">
</head>
<body>
<div id="showfriend">
<a href="#">Our Furry Friends Need
Your Help

</a>
</div>
```



index.html

锚标记在CSS中设置了“hover”
和“active”状态。用户将鼠标停留在链
接上时，图像就会出现。

狗狗图片嵌套在锚标记中。用户单击这个
锚标记中的链接之前图片不应出现。

```
a:link img, a:visited img {
display:none;
}
```

这个CSS选择器将嵌套图像的display属
性改为“none”，使页面第一次加载时这
个图像不可见。

```
a:hover img, a:active img {
display:block;
}
```

用户把鼠标停留在锚标记上时或者单击这
个锚标记时，img元素的display属性会变
为“block”。然后图像立即出现。

```
a {
text-decoration:none;
color: #000;
}
```



my_style.css



OK,现在可以具体深入了。能为我们想要的所有功能编写jQuery了吧?

当然可以，不过情况可能会变得一团糟。

使用jQuery建立市场部门希望的酷炫效果之前，还需要确保jQuery已经做好准备来施展它的“魔法”。现在你已经知道了，jQuery的主要工作之一是管理HTML元素，所以我们需要有一个好的结构。要得到元素，jQuery会使用CSS中使用的相同选择器，所以我们还需要有明确定义的样式。

重新考虑需求

考虑结构时，最好再回过头去想想你要构建什么。市场部门希望当人们单击页面上“Show Me the Furry Friend of the Day”（显示今日宠物朋友）时，图像会下滑并淡入出现。需要对HTML和CSS做哪些改动？



建立HTML和CSS文件

在编写jQuery语句之前，先来考虑在HTML和CSS文件中要做哪些工作。打开第1章的jQuery文件（如果你还没有打开这些文件，则务必返回到之前“如何使用本书”一节了解有关的详细信息）。找到“Chapter 1”中的Begin文件夹。然后在这些文件中增加以下粗体显示的代码，如下所示。

```
<!DOCTYPE html>
<html><head>
  <title>Furry Friends Campaign</title>
  <link rel="stylesheet" type="text/css" href="styles/my_style.css">
</head>
<body>
  <div id="clickMe">Show Me the Furry Friend of the Day</div>
  <div id="picframe">
    
  </div>
  <script src="scripts/jquery-1.6.2.min.js"></script>
  <script>
    $(document).ready(function() {
      $("#clickMe").click(function() {
        // ...
      });
    });
  </script>
</body>
</html>
```

动手做！

这里建立一个可单击的div。我们将在下面的CSS文件中为其增加样式，让它与picframe div有同样的外观。

这里是将要滑动显示的层叠图片的picframe div。



index.html

```
#clickMe {
  background: #D8B36E;
  padding: 20px;
  text-align: center;
  width: 205px;
  display: block;
  border: 2px solid #000;
}

#picframe {
  background: #D8B36E;
  padding: 20px;
  width: 205px;
  display: none;
  border: 2px solid #000;
}
```

这里设置clickMe div的样式，使它与picframe div有相同的外观。

将picframe选择器设置为“display: none”，这样一来，若要加载时不会显示这个div。



更多资源请访问稀酷客(www.ckook.com)

jQuery特写



现在已经建立了HTML和CSS文件，下面来分析<script>标记之间的代码。

嘿，DOM……

……你要是准备好了，
已经完全加载……

……我希望你为
我做点事情。

DOM

`$ (document).ready(function() {`

这是clickMe div的选择器。

`$("#clickMe").click(function()`

{

这个选择器部分与方法部分分开。

将ID为clickMe的按钮与click事件连接，使这个按钮可单击。

接下来时做什么？完成相关工作的代码放在这两个大括号之间（也称为“代码块”）。

分号是一个终止符。它会结束这个jQuery click语句。

`});`

});

这个分号会结束jQuery ready函数。

这里还会有很多新术语。
稍后我们就会更详细地讨论事件、方法和函数。



不过这个页面还是没有新意！

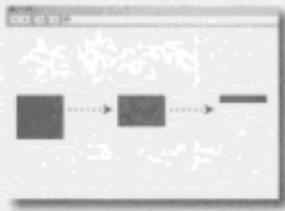
你说对了。我们的HTML和CSS已经准备就绪，现在就需要jQuery了。

我们希望picframe div能够滑动并淡入显示。幸运的是，设计jQuery的人已经建立了一些效果，所以我们可以同时控制这两个视觉动作：滑动和淡入。这本书有一章专门讨论jQuery效果（第5章），所以现在不用担心每个细节。先来实现滑动和淡入。

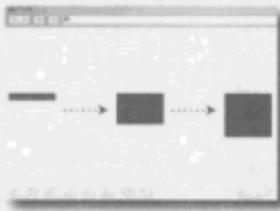
滑入……

我们要实现的第一个效果是让图像滑入视图，这是市场部门经理希望的。处理滑动有3种方式：

```
$("div").slideUp();      $("div").slideDown();      $("div").slideToggle();
```



slideUp方法会改变元素的height属性，直至高度为0，然后将元素隐藏。



slideDown方法将元素的height属性从0设置为CSS样式中设置的那个高度值。



slideToggle动作暗示“如果原先是向上，就向下滑动，如果原来是向下，则向上滑动”。

更多资源请访问稀酷客(www.ckook.com)

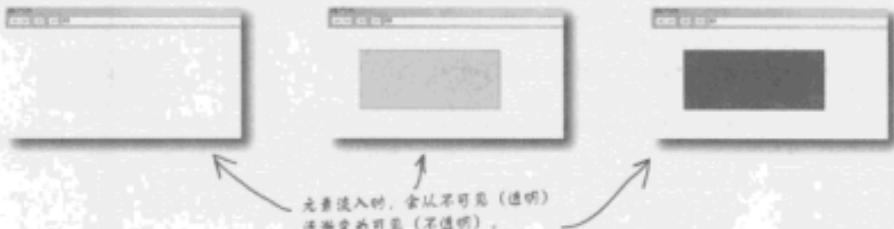
实现淡入

我们还希望图像逐渐出现，从不可见到底完全可见。同样的，jQuery为此提供了一个方法，称为淡入淡出。淡入淡出方法与之前看到的滑动非常类似，包括`fadeIn`、`fadeOut`、`fadeTo`和`fadeToggle`方法。对现在来说，我们只使用`fadeIn`，利用这个方法可以控制HTML元素的`opacity`和`transparency`属性。

这是我们希望淡入显示的元素，在这里是一个图像。

可以在括号中放入一个值来指定以多快的速度淡入，一般单位为毫秒(ms)。

```
$("img").fadeIn();
```



元素淡入时，会从不可见（透明）
逐渐变为可见（不透明）。



你认为完成我们希望的效果需要多少条jQuery语句？

可以试试看，先在一张草稿纸上写出这些语句。如果你还不确定，则可以先用英语写。然后再换个思路，用jQuery的方式来考虑。

更多资源请访问稀酷客(www.ckook.com)

你现在的位置：

如此而已？

太让人惊奇了，只需要写两行jQuery代码就能达到这些效果。现在你可能有点明白了：为什么会有这么多的人喜欢jQuery。将下面粗体显示的代码行增加到你的index.html文件中，你的任务就完成了。

动手做！

```
<!DOCTYPE html>
<html>
  <head>
    <title>Furry Friends Campaign</title>
    <link rel="stylesheet" type="text/css" href="styles/my_style.css">
  </head>
  <body>
    <div id="clickMe">Show me the Furry Friend of the Day</div>
    <div id="picframe">
      
    </div>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script>
      $(document).ready(function() {
        $("#clickMe").click(function() {
          $("img").fadeIn(1000);
          $("#picframe").slideToggle("slow");
        });
      });
    </script>
  </body>
</html>
```

在jQuery中，若用一种适当的方式对元素排序，以免它们相互冲突，这一点很重要。我们将在本书后面解释这个问题。

先在图像上进行淡入效果。

在滑动中增加额外的内容，可以有更丰富的效果。我们还会在第5章深入地讨论这些内容。

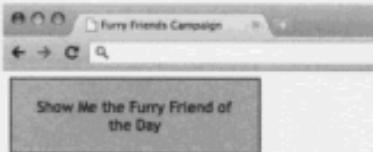


index.html

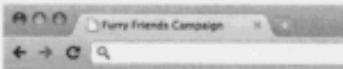


试一试

在你喜欢的浏览器中打开这个页面，确保一切正常。



单击这里



Show Me the Furry Friend of the Day



图像应该滚动
显示向下滑动。



Watch it!

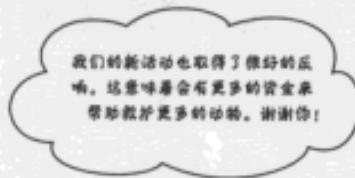
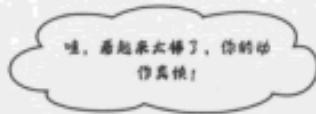
在多个不同的浏览器上检查。

jQuery在所有浏览器上都采用同样的方式工作。但仅凭这一点并不意味着CSS文件中定义的样式或者对页面元素应用的动态样式在所有浏览器中都能得到同样的效果！

更多资源请访问稀酷客(www.ckook.com)

你挽救了宠物朋友活动

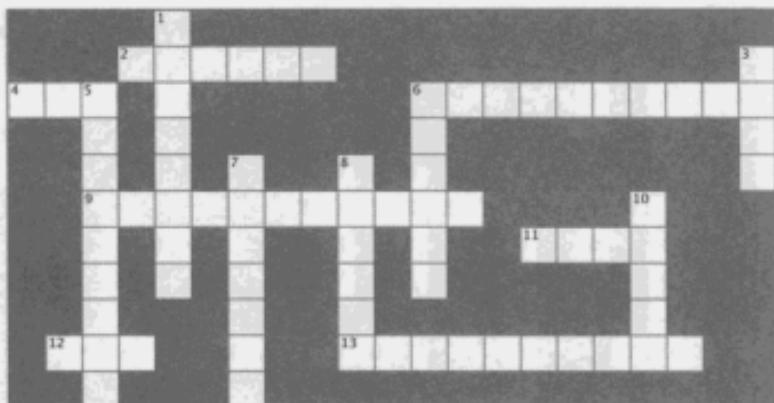
尽管只是对一些HTML和CSS做了细微的改动，而且只写了两行jQuery代码，但你确实已经很好地完成了任务。想想看，你救护了多少小狗狗……





jQuery填字游戏

现在该坐下来，让你的左脑动一动了。下面是一个标准的填字游戏：所有答案都能在本章找到。



横向

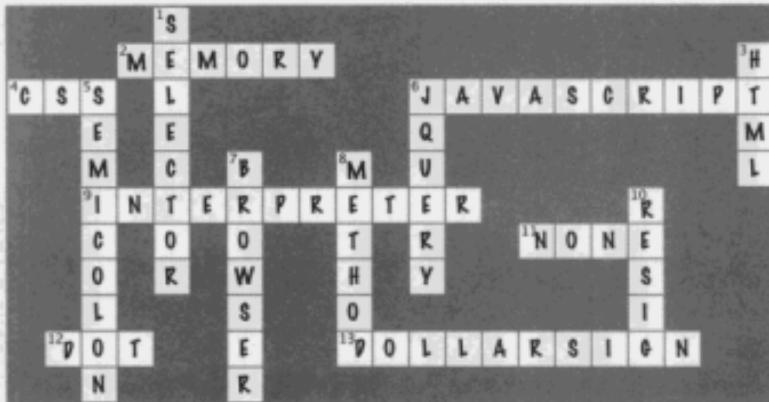
2. 浏览器从Web服务器接收一个Web页面时，它将这个Web页面加载到它的_____。
4. 为Web页面增加样式。
6. 编写jQuery的语言。
9. JavaScript_____将你给出的指令翻译为页面上的不同动作。
11. 确保页面加载时元素不显示的CSS设置。`display:_____`
12. 用于将一个jQuery选择器与jQuery方法分开的字符串。
13. 用做jQuery快捷方式的符号名。
1. jQuery用_____查找并返回web页面上的一个元素。
3. 这种文件构建Web页面的结构。
5. 结束jQuery语句的符号名。
6. 专门用来提供复杂的交互性和富视觉效果的JavaScript库。
7. 使用一个_____来测试jQuery脚本能正常工作。
8. 看到一个关键字后面的一组括号时，可以知道在处理一个_____。
10. jQuery库的创造者，John_____。

纵向

更多资源请访问稀酷客(www.ckook.com)



jQuery填字游戏答案



更多资源请访问稀酷客(www.ckook.com)



JQuery工具箱

你已经掌握了第1章，现在你的工具箱中已经增加了基本jQuery函数、选择器、单击事件和淡入淡出效果。

jquery函数

使用jquery函数可以从HTML页面中选择要处理的元素。

快捷方式意味着你不必应该地输入“jquery”。

jquery函数可以处理选择器、HTML，甚至可以处理JavaScript对象。

选择器

jquery采用CSS的方式选择元素：同样可以利用选择器。

几乎任何HTML元素都可以用jquery选择器来选择。

淡入淡出效果

一旦选择了一个元素，可以采用多种方式让它淡入淡出，包括FadeIn、FadeOut、FadeTo和FadeToggle。

可以将各种元素淡入显示，包括文本、图像等。

控制淡入淡出效果的速度，可以在语句末尾的括号中加入一个时间值（单位为毫秒）。

更多资源请访问稀酷客(www.ckook.com)

2 选择器与方法

获取，行动

嘿，亲爱的，我的选择器和方法可以让你的Web页面元素变魔法……



jQuery可以帮助你获取Web页面元素，用它们完成各种工作。这一章中，我们将深入研究jQuery选择器和方法。利用jQuery选择器可以获取页面上的元素，利用方法可以对这些元素做一些处理。就像一本大部头的魔法书，jQuery库允许我们动态地做出各种各样的变化。我们可以让图像消失，然后突然又重新出现。我们可以选择一段文本，动画地展示文本字体大小的改变。嗯，演出就要开始了，来获取一些Web页面元素，行动吧！

更多资源请访问稀酷客(www.ckook.com)

“Jump for Joy” 需要你的帮助

你收到一个朋友的Email，她是一名专业肖像摄影师，想发起一个“Jump for Joy”促销活动。让用户购买照片时能得到优惠。她需要你的帮助来推进这个促销活动。

来自: Emily

主题: Jump for Joy促销活动！

嘿，

我看了你的微博，你说近来在做更有交互性的Web工作，所以我真的希望你能帮我，在我的网站上为“Jump for Joy”促销活动增加一些交互性的内容。我想为访问者提供一个机会，可以在结帐前打折，鼓励他们在网站上多看看，多点点鼠标（当然希望也能多购买）。

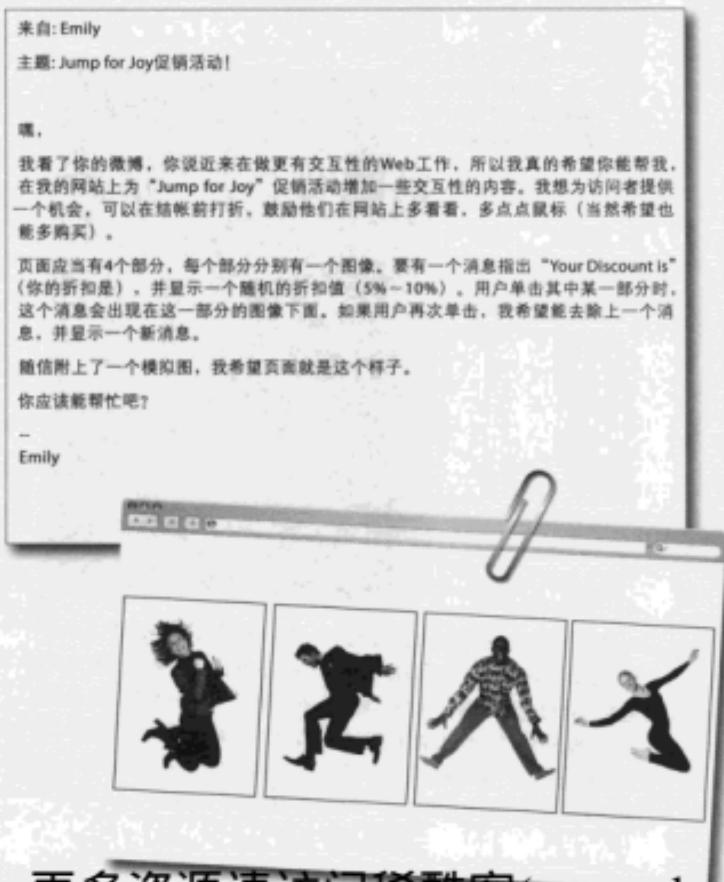
页面应当有4个部分，每个部分分别有一个图像。要有一个消息指出“Your Discount is”（你的折扣是），并显示一个随机的折扣值（5% ~ 10%）。用户单击其中某一部分时，这个消息会出现在这一部分的图像下面。如果用户再次单击，我希望去除上一个消息，并显示一个新消息。

随信附上了一个模版图，我希望页面就是这个样子。

你应该能帮忙吧？

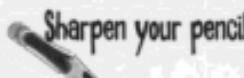
—

Emily



项目需求是什么？

Emily是一个很棒的摄影师。不过她的请求很含糊。下面再仔细看看这封邮件，明确她到底想要些什么。开始编写jQuery之前，你肯定希望能清楚地知道项目需求（或者用户需求）到底是什么。



根据邮件中的请求，把这些请求分解为一个任务列表，我们的Web应用需要完成这些任务。这个列表将作为指南，确保我们的Web应用能够满足客户的需要。

要完成的任务列表：

- 1.
- 2.
- 3.
- 4.
- 5.

要将用户的请求转换为具体的项目需求，这是一个重要的技能，随着实践和时间的积累，你的水平会越来越高。

Make it Stick



更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil Solution

根据邮件中的请求，把这些请求分解为一个任务列表，我们的Web应用需要完成这些任务。这个列表将作为指南，确保我们的Web应用能够满足客户的需要。以下是我们答案。

要完成的任务列表：

1. 首页应当有4个部分，每个部分分别显示一个“jump for joy”图像。
2. 这些部分应当是可单击的。
3. 需要一个消息，指出“Your Discount Is”（你的折扣是），并提供一个随机的折扣值（介于5%~10%）。
4. 用户单击其中一个部分时，这个消息应当出现在这个部分中图像的下方。
5. 如果用户再次点击，应当移除一个消息，并显示一个新消息。

很好，既然已经明确项目需求，下面就动手开始写jQuery吧！



嘿！别这么着急！

完成每一个jQuery项目时，首先明确项目需求是一个好习惯。不过，在具体编写jQuery代码之前，我们还需要做一些工作，先来建立结构和样式。第1章已经做过一点这方面的工作。在领略jQuery的妙处之前，还有一些样式和结构需要建立。

更多资源请访问稀酷客(www.ckook.com)

深入了解div

页面上要有4个可单击的区域，所以先来建立这些可单击区域。要达到这个目的，最有用、最灵活的HTML元素就是

标记。

标记对于建立结构有着重要作用，因为它是一个块级元素。另外很容易为div元素增加样式，让它有我们期望的表现。



打开你喜欢的文本编辑器，创建需要的HTML和CSS文件。下面给出了一些代码。可以由此起步。不过这些代码中少了一些关键的元素。在页面中增加以下各元素，完成后再做个测试：

- 包含jQuery库（版本1.6.2）的标记。
- ID为header的

标记。
- ID为main的

标记。
- 在main div中的4个div元素中，分别放入一个不同的图像（可以从这里得到图像：www.thinkjquery.com/chapter02/images.zip）。

```
<html>
  <head>
    <title>Jump for Joy</title>
    <link href="styles/my_style.css" rel="stylesheet">
  </head>
  <body>

    <h2>Jump for Joy Sale</h2>
    </div>

    <div></div>
    <div>.....</div>
    <div>.....</div>
    <div>.....</div>
  </div>

  <script> </script>
</body>
</html>
```

```
div {
  float:left;
  height:245px;
  text-align:left;
  border: solid #000 3px;
}

#header {
  width:100%;
  border: 0px;
  height:50px;
}

#main {
  background-color: grey;
  height: 500px;
}
```



my_style.css



更多资源请访问稀酷客(www.ckook.com)



打开你喜欢的文本编辑器，创建需要的HTML和CSS文件。下面给出了一些代码，可以由此起步，不过这些代码中少了一些关键的元素。增加了以下元素之后，你的页面应当与我们的答案类似。

- 包含jQuery库（版本1.6.2）的标记。
- ID为header的

标记。
- ID为main的

标记。
- 在main div中的4个div元素中，分别放入一个不同的图像。

你的HTML和CSS文件应该是这样的。

```

<html>
  <head>
    <title>Jump for Joy</title>
    <link href="styles/my_style.css" rel="stylesheet">
  </head>
  <body>
    <div id="header" >
      <h2>Jump for Joy Sale</h2>
    </div>
    <div id="main" >
      <div></div>
      <div></div>
      <div></div>
      <div></div>
    </div>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script> </script>  </body>
  </html>

```

包含jQuery库。

```

div {
  float:left;
  height:24px;
  text-align:left;
  border: solid #000 3px;
}

#header {
  width:100%;
  border: 0px;
  height:50px;
}

#main {
  background-color: grey;
  height: 500px;
}

```



my_style.css

包含div元素。



index.html



试一试

在你喜欢的浏览器中打开这个页面，确保一切正常。可以利用这个机会说明希望如何实现页面功能。

id为introduction的div元素。

id为main的div元素，其中包含……

……4个图像div元素。



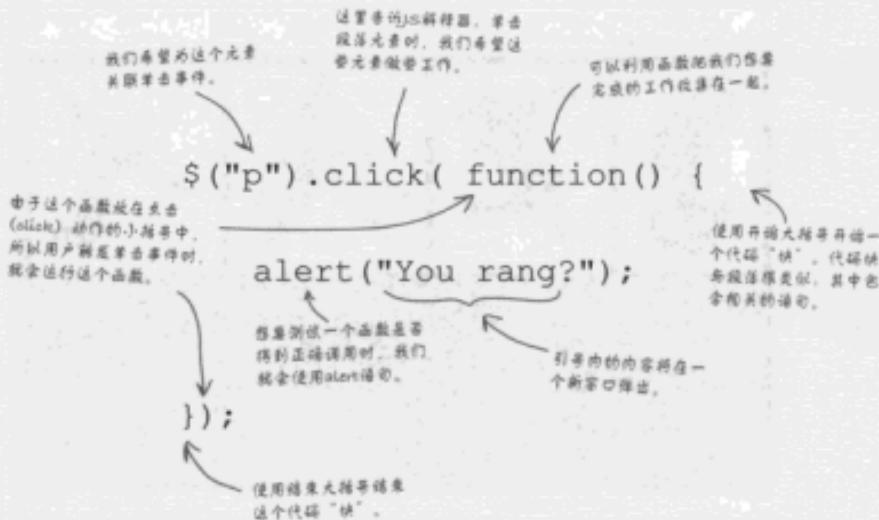
现在页面上有4个区域，分别包含一个图像。如何让这些区域可单击呢？

更多资源请访问稀酷客(www.ckook.com)

单击事件特写

我们已经看到，使用jQuery很容易让一个元素可单击。

单击一个页面元素会在页面上触发一个事件，还可以运行函数。后面还会更深入地讨论事件和函数，不过现在只讨论在段落（或`<div>`）标记上click事件如何作用。





jQuery代码磁贴

移动下面的磁贴来编写代码，让所有div元素都是可单击的。单击一个div元素时，使用JavaScript alert函数显示文本“You clicked me.”。我们已经为你放好了几个磁贴。

```

    alert
    div
    function()
    {
        ${document}
    }
    function()
    </script>
    ").
    });
    $("
    click
    {
    ready(

```

<script>

\$(“

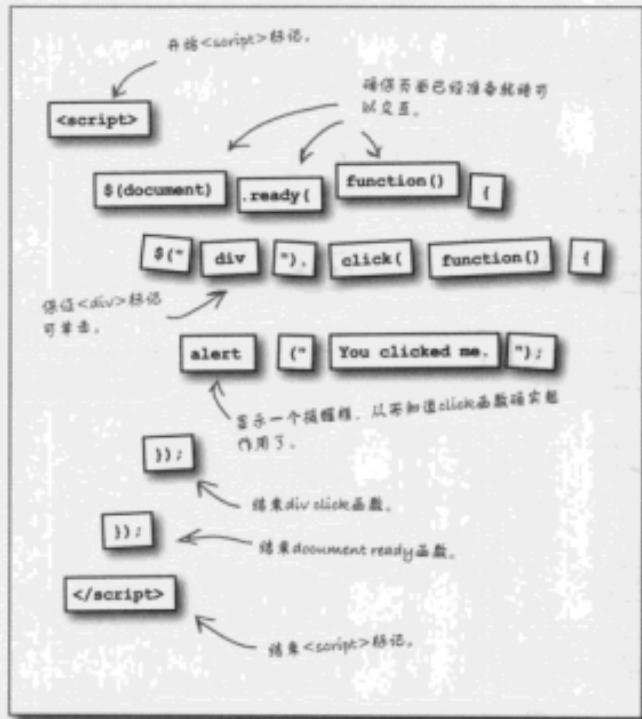
You clicked me.

});



jQuery代码磁贴答案

移动下面的磁贴来编写代码，让所有div元素都是可单击的。单击一个div元素时，使用JavaScript alert函数显示文本“You clicked me.”。



为页面增加click方法

使用上一页磁贴答案中的代码，更新HTML文件来包含这个脚本。不要忘了：要把代码放在一个<script>标记中！

动手做！

alert函数会在浏览器中弹出一个窗口，显示一个消息。如果我们在代码中增加了变量和函数，想看到相应的结果时就全使用这个alert函数。

```
<html>
  <head>
    <title>Jump for Joy</title>
    <link href="styles/my_style.css" rel="stylesheet">
  </head>
  <body>
    <div id="header">
      <h2>Jump for Joy Sale</h2>
    </div>
    <div id="main">
      <div></div>
      <div></div>
      <div></div>
      <div></div>
    </div>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script>
      $(document).ready(function() {
        $("div").click(function() {
          alert("You clicked me.");
        });
      });
    </script>
  </body>
</html>
```

在<script>标记之间增加这些代码行，使div元素可单击。

有些程序员会增加注释来帮助区分小括号和大括号。这是一种编码风格，是否采用完全看你的个人喜好。



index.html



试一试

在你喜欢的浏览器中打开这个页面，确保一切正常。现在，一旦单击页面上的图像，应该就能看到提醒消息。



这是看你增加的级联框，可以看到click函数起作用了。

没错，不过不论我单击哪里，都会得到这个提醒消息。为什么会有这样？



嗯，这确实是个问题。

看来我们单击得有点过头了。下面再来看看单击事件。

jQuery的确会按照我们的要求去做。它选择了所有div……

……因为各个div增加了一个click方法。

`$(“div”).click();`

实际上，甚至不用单击图像也能得到这个消息。对于我们的页面结构，这些div元素嵌套在另一个div中。所以单击这些div时，浏览器会认为你同时单击了这两个div（内部嵌套div和外层的包含div），这种情况下就会得到两个提醒。显然，需要在这里更明确地告诉jQuery要做什么……

更为特定

这里的问题在于，我们的选择没能做到足够特定。那么，该如何得到4个子

而排除那个更大的包含

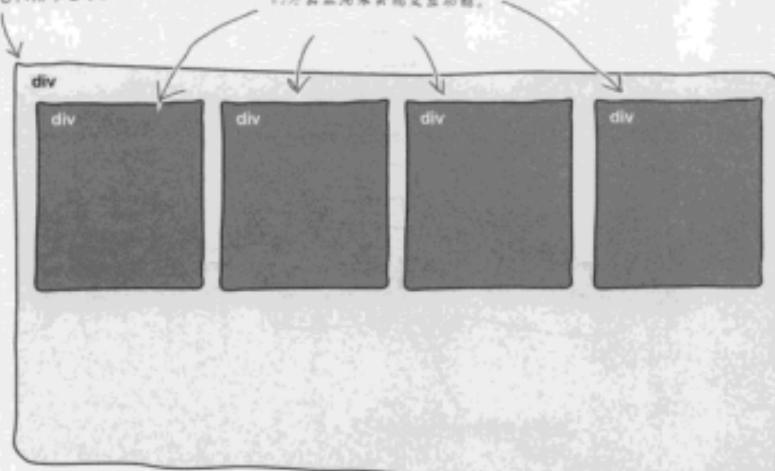
呢？应该还记得，第1章曾经说过，jQuery选择器可以使用CSS类和ID。可以增加元素的类和ID，这样就能更特定地指定希望jQuery获取哪些元素。

我们不希望这个

元素可单击。
它只用于显示。

我们希望这些

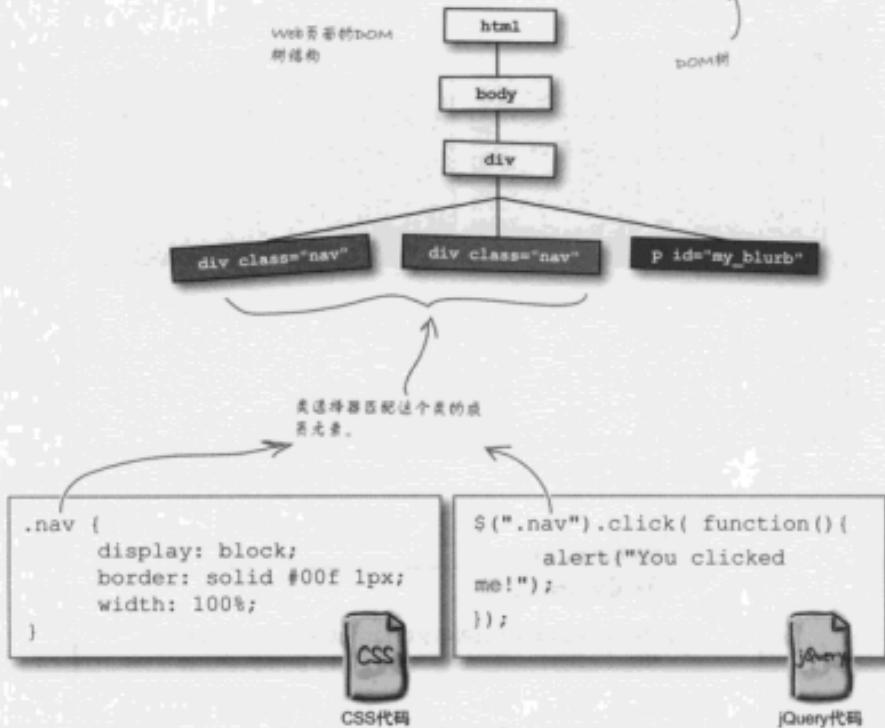
是可单击的。它们才真正用来实现功能。



你能单独使用CSS类或ID或者结合CSS类和ID来指定上面的div元素吗？哪种做法最好？为什么？

元素分类

CSS中，类用于将元素分组，还可以为一类元素提供共同的样式属性。页面上可以有一个或多个同类的元素。jQuery同样可以使用类选择器，而且可以用jQuery方法影响一组元素。CSS和jQuery都使用“.”指示一个类，所以，编写分类代码超级简单。



用ID标识元素

ID选择器用来标识页面上唯一的一个元素。与CSS中类似，jQuery中也使用#号来标识ID选择器。想要特别地指定一个元素时，或者如果页面上只有一个这种元素（如页眉或页脚），ID选择器就很适用。

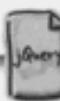
ID选择器匹配唯一的一个元素。

```
#my_blurb {
    display: block;
    border: 0px;
    height: 50%;
}
```



CSS代码

```
$("#my_blurb").slideToggle("slow");
```



jQuery代码

WHO DOES WHAT?

在适当的列中打勾，指出用类选择器能够做什么，另外用ID选择器能做什么。要记住，有时类和ID选择器可能会做同样的事情！

	类	ID
唯一标识页面上的一个元素	<input type="checkbox"/>	<input type="checkbox"/>
可以标识页面上的一个或多个元素	<input type="checkbox"/>	<input type="checkbox"/>
JavaScript方法（跨浏览器）可以用它来标识一个元素	<input type="checkbox"/>	<input type="checkbox"/>
CSS可以用它为元素应用样式	<input type="checkbox"/>	<input type="checkbox"/>
可以为一个元素同时应用多个这种选择器	<input type="checkbox"/>	<input type="checkbox"/>



答案

在适当的列中打勾。指出用类选择器能够做什么，另外用ID选择器能做什么。要记住，有时类和ID选择器可能会做同样的事情！

	类	ID
唯一标识页面上的一个元素	<input type="checkbox"/>	<input checked="" type="checkbox"/>
可以标识页面上的一个或多个元素	<input checked="" type="checkbox"/>	<input type="checkbox"/>
JavaScript方法（跨浏览器）可以用它来标识一个元素	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CSS可以用它为元素应用样式	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
可以为一个元素同时应用多个这种选择器	<input checked="" type="checkbox"/>	<input type="checkbox"/>

there are no Dumb Questions

问：什么是块级元素（block-level element）？

答：块级元素在其父元素中作为矩形对象出现。这些元素跨行也不会中断。块级元素还会有块边距、宽度和高度属性，这些属性可以独立于外层元素单独设置。

问：为什么把<script>标记放在页面最下面的</body>标记的前面？我还以为<script>标记总是要放在<head></head>标记中呢！

答：没错，这曾经是推荐的最佳实践（对于某些人来说，现在仍然是）。不过，脚本会带来一个问题，它们会阻塞浏览器中的并行下载。不同服务器的多个图像可以同时下载，不过一旦浏览器遇到一个<script>标记，就不能再并行地下载多个元素了。把<script>标记放在最下面，这意味着有助于提高页面的加载速度。

问：JavaScript的提醒（alert）到底是怎么回事？

答：我们用到的提醒看上去可能不算太巧妙，不过，提醒在很多方面确实很有用。本质上讲，JavaScript提醒

是一个包含消息的简单窗口。括号之间的文本就是提醒消息所要显示的内容。如果你希望显示一个文本串，则要用引号将这个文本引起来。要显示变量值，可以直接输入变量名而不需要引号。还可以使用+号结合变量值和文本串。你可能经常看到提醒，但从未认真考虑过，对它们已经熟视无睹了。比如在菜单中编辑一个必填域时就会看到提醒。在这里，我们使用提醒更多的是为了完成测试和调试。当然完成测试和调试还有更健壮的方法，本书后面的章节中还会介绍。

Fireside Chats



今晚话题：CSS和jQuery选择器在讨论它们的区别。

CSS选择器：

你好啊，jQuery选择器。真高兴你能来，让大家知道你的存在要完全归功于我。

嗯，我确实只考虑样式，不过我的能力相当强，可以瞬间改变元素的外观。

你的意思是说，那些工作你能做而我做不了？

哈，有意思！你是说，你“返回”元素？

不过，我也可以影响我选择的所有元素。如果我愿意，则可以把它们的背景色都改成紫红色。别忘了，是我的引擎赋予了你现在的能力。

不得不承认，听上去确实很酷。

jQuery选择器：

呵呵，谢谢，也许吧。当然了，在选择元素方面，我的很多能力确实是从你那里得来的。不过，我更多地关注行为而不是样式。你只是坐在那里静静地让一切美观漂亮，而我是实实在在地做具体动作。

你在页面上确实很强，这一点我完全没有异议。你有你的工作，我是说在改变元素外观方面，而我也有自己的工作。这是完全不同的。

我的工作是查找元素并返回这些元素，这样就能利用一个方法对返回的元素集做些处理。

假设有人用我来选择页面上的所有段落元素，我会得到并维护这组段落元素，让jQuery方法对它们做所希望的处理。

你的选择器引擎确实赋予了我很大一部分能力，不过我的另一部分能力来自于JavaScript。别忘了我的名字里有一个“Query”。我可以向浏览器请求一个元素，维护这个元素，把它传递到一个jQuery方法，让它在页面上飞来飞去，甚至可以让它消失。

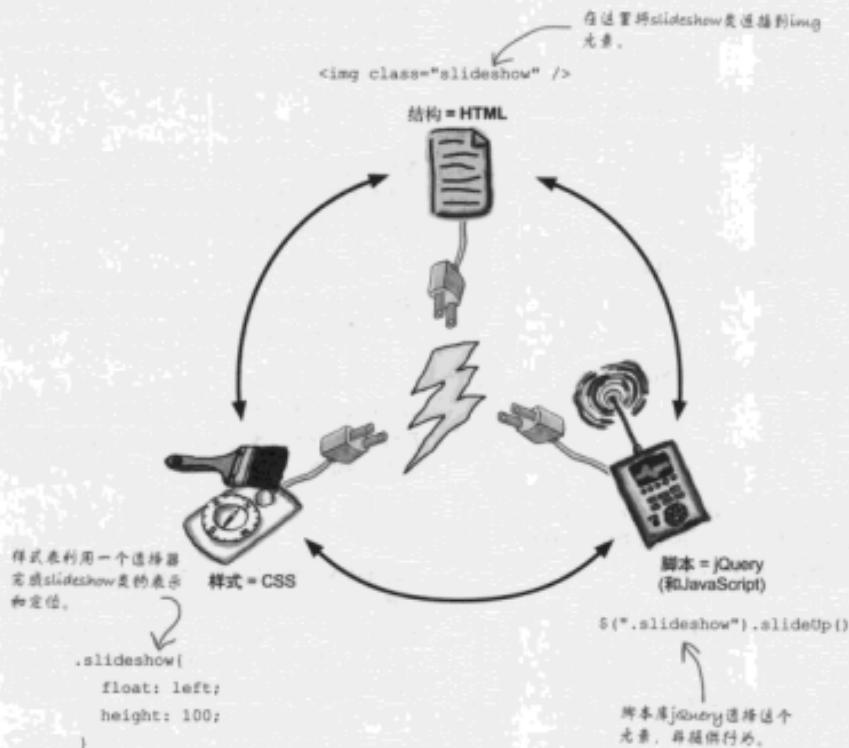
对呀，不过你说的也对，没有你我是做不到这些的。

更多资源请访问稀酷客(www.ckook.com)

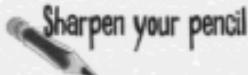
连接Web页面

类和ID是Web页面中3个层次的公共基础。我们在第1章已经了解这3个层次。它们分别是：结构、样式和脚本。选择器将这三个层次连接在一起，使它们能够协同工作。HTML提供了构建模块（即元素及其属性），也就是Web页面的结构。CSS提供样式，也就是这些元素的表示和位置。JavaScript和jQuery提供脚本，来控制这些元素的行为或功能。

假设图像有一个slideshow类，我们希望对这个图像运行slideUp方法：



更多资源请访问稀酷客(www.ckook.com)



更新页面的结构、样式和脚本，仅让4个图像div是可单击的。在CSS文件中，创建一个CSS类（名为guess_box），将它应用到HTML和脚本。看起来其中一个div元素少了ID属性。你能看出是哪一个吗？请为它补上ID属性。

```

<html>
  <head>
    <title>Jump for Joy</title>
    <link href="styles/my_style.css" rel="stylesheet">
  </head>
  <body>
    <div id="header">
      <h2>Jump for Joy Sale</h2>
    </div>
    <div>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
    </div>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script>
      $(document).ready(function() {
        $("").click(function() {
          alert("You clicked me.");
        });
      });
    </script>
  </body>
</html>

```

```

div{
  float:left;
  height:245px;
  text-align:left;
  border: solid #000 3px;
}

#header{
  width:100%;
  border: 0px;
  height:50px;
}

#main{
  background-color: grey;
  height: 500px;
}

```

height: 245px;



my_style.css



Sharpen your pencil

Solution

对于将用来隐藏折扣码的所有div元素，为它们增加guess_box类。更新选择器来使用这个类，并增加到CSS文件中。另外，main div元素需要补充ID属性。

```

<html>
  <head>
    <title>Jump for Joy</title>
    <link href="styles/my_style.css" rel="stylesheet">
  </head>
  <body>
    <div id="header">
      <h2>Jump for Joy Sale</h2>
    </div>
    <div id="main">
      <div class="guess_box"></div>
      <div class="guess_box"></div>
      <div class="guess_box"></div>
      <div class="guess_box"></div>
    </div>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script>
      $(document).ready(function() {
        $("guess_box").click(function() {
          alert("You clicked me.");
        });
      });
    </script>
  </body>
</html>

```

```

div{
  float:left;
  height:245px;
  text-align:left;
  border: solid #000 3px;
}

#header{
  width:100px;
  border: 0px;
  height:50px;
}

#main{
  background-color: grey;
  height: 500px;
}

.guess_box{
  height:245px;
}

```



my_style.css

在这里为“猜猜看”增加一个类。高度与方框中图像的高度一致，以便更好地对齐。



index.html

更多资源请访问稀酷客(www.ckook.com)

再来看我们的列表

再来看我们的需求任务列表，想想看对于Emily的要求我们做到了哪一步：

- 页面应当有4个部分，每个部分分别显示一个“Jump for Joy”图像。
- 这些部分应当是可单击的。
- 需要一个消息，指出“Your Discount is”（你的折扣是），并提供一个随机的折扣值（介于5%~10%）。
- 用户单击其中一个部分时，这个消息应当出现在这个部分中图像的下方。
- 如果用户再次单击，要去除上一个消息，并显示一个新消息。



嘿，太容易了。我们的任务列表已经完成近一半了。剩下的几个任务看起来也不太糟。我们要创建一些文本和一个数。这有什么难的？

确实不算很难。

向用户显示一个消息要考虑几个问题。不要忘了，对于每一个访问网站的用户可能会有一个不同的消息。



你要创建一个消息并把它存储在某个地方，以便显示给访问者。你认为该怎么做呢？

创建一些存储空间

任务列表中的下一个需求是显示一些文本：“Your Discount is”，脚本运行时这个文本保持不变。不过，除此之外，还需要包含一个数，这个数会根据随机量改变或变化。我们需要一种方法在脚本中搭载这个信息，另外页面需要以某种方式保存这个信息。要保存可能变化的信息（或数据），这正是变量最擅长的工作。jQuery中需要用到变量时，可以使用JavaScript变量。

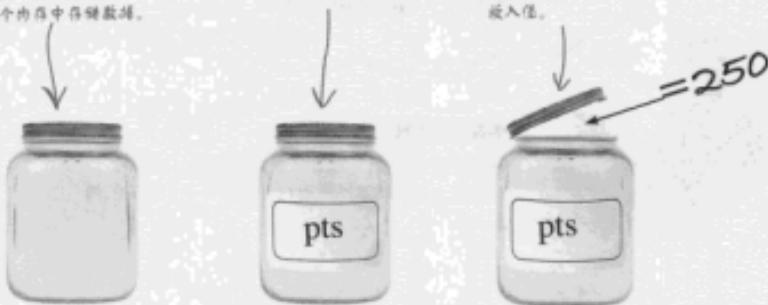
利用var关键字可以声明一个变量。
var关键字之后，必须指定一个名字。
在代码中可以这样设置变量值。

```
var pts = 250;
```

声明一个变量时，JavaScript解释器会提供一些浏览器内存，我们可以在这个内存中存储数据。

对变量命名，以便在脚本中引用这个变量。

值用一个等号在变量声明入值。



现在，如果想得到我们存储的数据，只需要通过它的变量名来获取。

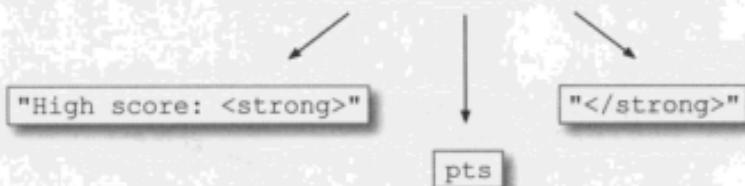
如果你想更多地了解JavaScript变量和math函数，可以找一本《Head First JavaScript》看看！

更多资源请访问稀酷客(www.ckook.com)

通过联接混合在一起

在jQuery脚本中，我们会存储不同类型的数据：数字、文本或true/false值。很多情况下，特别是需要向访问者显示不同的消息时，可能要在HTML中混合这些不同类型的数据。这会让web页面功能更强大。那么，如何将变量与其他值结合在一起呢？我们的方法是使用联接（concatenation）。假设你有一个视频游戏，其中有一个变量名为pts，用来存储最高分，需要为赢家显示这个最高分：

需要把这3部分信息结合（或联接）在一起。



这会得到：

```

var msg = "High score: <strong>" + pts + "</strong>"
  
```

设置一个文本或HTML值时，
要使用引号。

引用一个变量时，
用变量名而不如引号。

还可以在变量中嵌入HTML标记！

利用“+”字符可以联接
(或结合) 文本、数字、
变量等。



我们给出了建立一个discount变量的JavaScript代码，这个变量包含一个从5~10的随机数。
请编写代码来建立一个discount_msg变量，显示消息和这个随机变量。折扣消息需要显示在一个段落元素中。

```
var discount = Math.floor((Math.random() * 5) + 5);
```



我们给出了建立一个discount变量的JavaScript代码，这个变量包含一个5~10的随机数。请编写代码来建立一个discount_msg变量，显示消息和这个随机变量。折扣消息要显示在一个段落元素中。

不用担心，第3章就讲解var和random函数。

```
var discount = Math.floor((Math.random() * 5) + 5);
var discount_msg = "<p>Your Discount is " + discount + "%</p>";
```

再来看代码……

既然已经建立了变量来存储链接的折扣消息，下面只需要更新<script>标记之间的内容，所以现在的重点就是如何更新这个内容。

动手做！

```
<script>
$(document).ready(function() {
    $(".guess_box").click( function() {
        var discount = Math.floor((Math.random() * 5) + 5);
        var discount_msg = "<p>Your Discount is " + discount + "%</p>";
        alert(discount);
    });
});
</script>
```

创建新的
JavaScript变量。

将discount变量放在alert中。
确认一切正常。



index.html

用append插入消息

现在已经有了消息，那么如何把这个消息显示在页面上单击的图像下面呢？要增加一个新消息，可以认为就是将它插入到页面中。为此jQuery提供了很多方法，可以向一个已有的元素中插入内容。我们将在第4章更深入地介绍一些更有用的方法。现在只简要地了解append动作。

```
<p>jQuery lets me add stuff onto my web page  
without having to reload it.</p>
```



这个jQuery语句告诉jQuery将引号中的内容追加到所有段落元素中。

```
$("p").append(" <strong>Like me, for instance.</strong>");
```



如果在脚本中运行这个代码，则将在页面上用粗体显示文本。



jQuery lets me add stuff onto my web page
without having to reload it.

Like me, for instance.

DOM结构的
HTML。

```
<p>jQuery lets me add stuff onto my web page  
without having to reload it.</p> <strong>Like me,  
for instance.</strong>
```



根据你对选择器的了解和新学的append，编写代码将discount_msg变量追加到guess_box元素。

更多资源请访问稀酷客(www.ckook.com)



Exercise Solution

为Web页面增加新消息就是这么简单！

```
$( ".guess_box" ).append(discount_msg);
```

↑
append是一个jQuery方法。jQuery
中使用方法来完成工作。

there are no Dumb Questions

问：对于使用的类名有没有什么限制？

答：类名必须以一个下画线(_)、短横线(-)或字母(a~z)开头，后面可以是任意多个短画线、下画线、字母或数字。有一点要注意：如果第一个字符是一个短横线，那么第二个字符必须是一个字母或下画线，而且类名长度必须至少有两个字符。

问：对于使用的变量名有没有限制呢？

答：当然有！变量不能以数字开头。另外，变量名中不能包含任何数学运算符(+ * - ^ / ! \)，也不能有空格或标点符号。不过，变量名中可以包含下画线。变量不能用任何JavaScript关键字命名（如window、open、array、string、location），另外变量名是区别大小写的。

问：可以为元素指定多少个类？

答：根据标准来讲，并没有明确的最大限制，不过真实世界中实际使用的最大数目大约为每个元素2000个类。

问：有没有办法选择页面上的每一个元素？

答：有啊！只需要在jQuery包装器中输入一个“*”，就能得到所有元素。

问：如果我为元素指定一个类或ID，但是没有任何样式，能影响浏览器中这些元素的外观吗？

答：不会，对于类或ID，并没有任何浏览器默认设置。有些浏览器确实会将元素区别对待，不过，如果只应用一个类或ID，而没有任何CSS，这不会有任何影响。



试一试

在你喜欢的浏览器中打开这个页面，确保一切正常。要特别注意提醒，确认正确地建立了discount变量。

Jump for Joy Sale

Your Discount is 6%

Your Discount is 15%

Your Discount is 5%

Your Discount is 0%

有人点击时，折扣消息会增加到guess_box类的各个div。

一切都很好，不过……

discount变量生成了一个随机数，并按我们期望的方式将消息追加到页面，不过它还有一个我们没有预料到的副作用：各个div中折扣会重复出现。

这可不是我们想要的。哪里出了问题呢？

```
<script>
$(document).ready(function() {
    $(".guess_box").click( function() {
        var discount = Math.floor((Math.random()*5) + 5);
        var discount_msg = "<p>Your Discount is "+discount+"%</p>";
        alert(discount_msg);           ← 这只是用来测试这个
        $(".guess_box").append(discount_msg); 变量。
    });
});
</script>
```

这个函数在遇到click方法时，会把guess_box类的各个成员是可单击的。
选择器已经很特定了，可以用来限制一个类。
但是优惠会影响这个类的所有div。

discount变量只应追加到被单击的那个div。那么，如何只选择被单击的div，并把discount变量追加到那个元素呢？

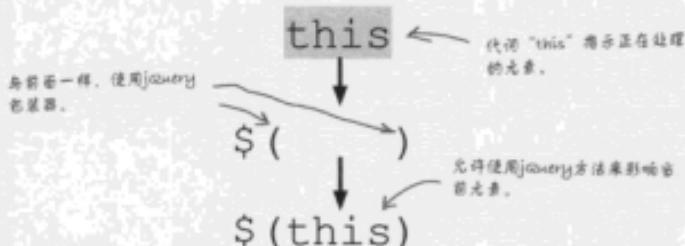
更多资源请访问稀酷客(www.ckook.com)



如果有一种简单的方法可以选择我单击的甜V就好了。这可能只是个美梦吧？
不过我知道这只是异想天开……

用\${this}返回当前元素

这一章中我们一直在考虑jQuery选择器，并讨论它们如何返回将由jQuery方法使用的元素。很多情况下，我们希望非常特定地指出选择哪个元素。谈到特定性，最简单的选择器就是\${this}（毕竟，你只需要记住代词“this”）。\${this}选择器为我们提供了一种简便的方法来指示当前元素。



有一点很重要，可以认为\${this}依赖于上下文。换句话说，\${this}可能表示不同的东西，这取决于在哪里使用或者何时使用。可以在调用jQuery方法时所运行的函数中使用这个选择器，这是最合适的做法之一：

```
这是访问元素的选  
择器。           调用一个jQuery方法。  
$ ("#myImg").click( function() {  
    $(this).slideUp();  
});           调用方法时运行  
这个函数。
```

在函数中访问当前
元素（在这里是
#myImg）。

click和slideUp都是
jQuery方法。看到括
号时可以知道你在让
调用一个方法或函数。

Geek Bits

this与\${this}

在JavaScript中，“this”指示代码中要处理的DOM元素。在this外增加\$()就得到了\${this}。利用这个选择器，我们可以使用jQuery方法与DOM元素交互。

更多资源请访问稀酷客(www.ckook.com)

让\$(this)发挥作用

下面来看\$(this)能不能帮我们解决问题。更新你的代码，现在要使用\$(this)，见下面粗体显示的代码。

动手做！

```
<script type="text/javascript">
$(document).ready(function() {
    $(".guess_box").click( function() {
        var discount = Math.floor((Math.random()*5) + 5);
        var discount_msg = "<p>Your Discount is " + discount +"%</p>";
        alert(discount_msg);
        $(this).append(discount_msg);
    });
}); //end doc ready
</script>
```

现在看来只将折扣添加到被单击的那个“猜折扣”。

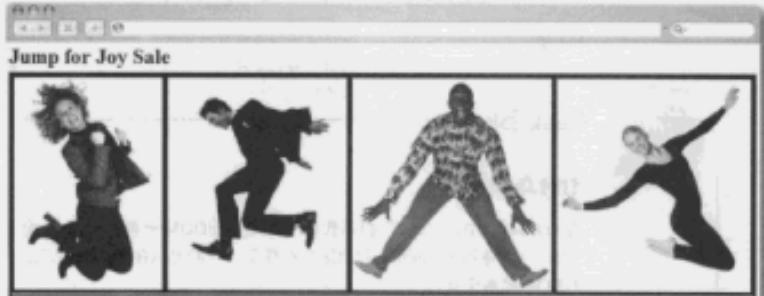


index.html



试一试

在你喜欢的浏览器中打开这个页面，确保一切正常。特别注意提醒，确认得到正确的discount变量。一定要单击多次，检查联接到discount变量的随机数也正常。



更多资源请访问稀酷客(www.ckook.com)



问得好！

这就是我们要完成的任务列表中的最后一步：

- 页面应当有4个部分，每个部分分别显示一个“Jump for Joy”图像。
- 这些部分应当是可单击的。
- 需要一个消息，指出“Your Discount is”（你的折扣是），并提供一个随机的折扣值（介于5%–10%）。
- 用户单击其中一个部分时，这个消息应当出现在这个部分中图像的下方。
- 如果用户再次单击，则要去除上一个消息，并显示一个新消息。

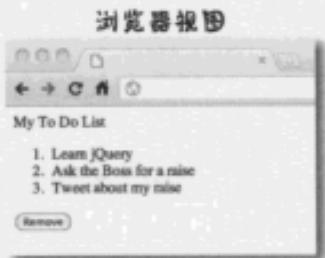


你认为如何删除上一个消息？

用remove妥善删除

如何去除上一个消息并建立一个新消息呢？可以使用remove方法。利用remove方法，我们可以将一个元素或者一组元素从页面上删除。下面来看一个非常简单的页面，页面上只有一个列表和一个按钮。

- ① 这是浏览器中看到的页面，右边是创建这个页面的HTML。



HTML视图

```
<div>My To Do List</div>
<ol>
    <li>Learn jQuery</li>
    <li>Ask the Boss for a raise</li>
    <li>Tweet about my raise</li>
</ol>
<button id="btnRemove">
```

- ② 这是按钮的代码，它会删除列表中的所有列表项：

```
$("#btnRemove").click(function() {
    $("li").remove();
});
```

remove也是一个jQuery方法。
可以使用jQuery方法集或是一个
选择器，就是看需求而定是否需要
参数。

- ③ 再来看浏览器中的页面和HTML（在jQuery完成之后），可以看到所有列表项都不见了。甚至HTML中的列表项也没有了！



HTML视图

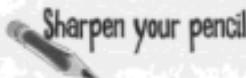
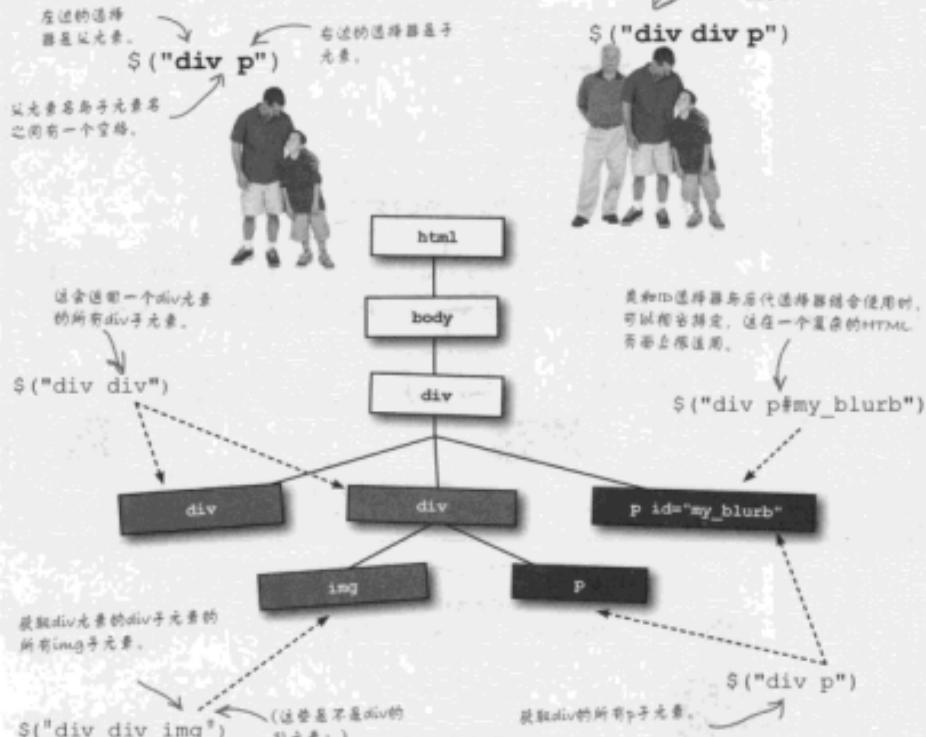
```
<div>My To Do List</div>
<ol>
</ol>
<button id="btnRemove">
```



需要使用哪个选择器来删除页面上的折扣消息？

深入了解后代选择器

后代选择器是jQuery可以使用的另一种选择器，这种选择器恰好适合我们当前的情况。利用后代选择器，可以指定元素之间的关系，可以选择子元素、父元素或兄弟元素。



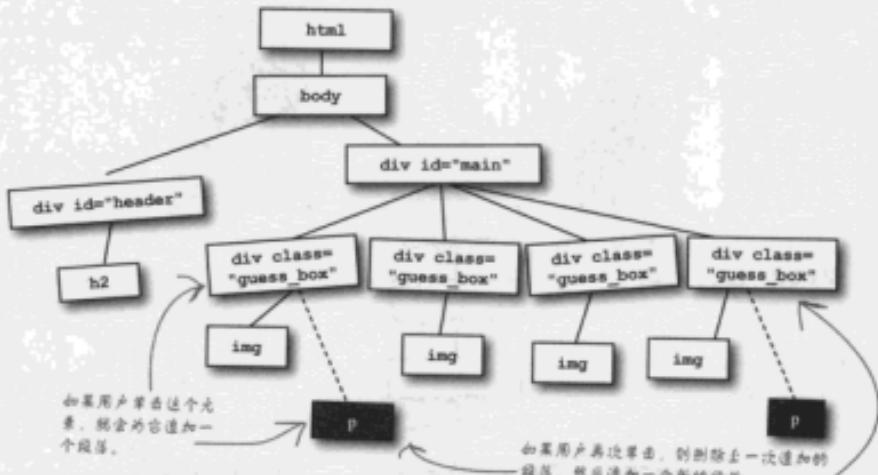
使用后代选择器编写语句，删除“Jump for Joy” DOM中的最后一个段落。

Sharpen your pencil

Solution

首先使用`.guess_box`类选择器，后面是一个`p`后代选择器，用来访问你增加的段落。然后，使用`remove`方法从页面删除`guess_box`类元素中的所有`p`元素。

```
$(".guess_box p").remove();
```

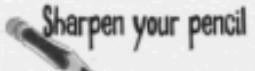


现在我可以往合适的地方追加
和删除元素了。何时或者往哪里追加
和删除会有影响吗？

是的，增加或删除元素的顺序非常重要。

增加元素之前是不能删除这个元素的，另外如
果你刚刚增加一个元素就把它删除，这也没有
意义，对不对？

更多资源请访问稀酷客(www.ckook.com)



你的任务是确定将remove语句放在哪里。将这个语句写在下面编号为1、2或3的行上。然后解释为什么把语句放在那里。考虑什么时候需要删除段落，然后使用排除法选出正确的位置。

```

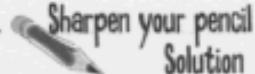
<script>
$(document).ready(function() {
1. .....
    $(".guess_box").click( function() {
2. .....
        var discount = Math.floor((Math.random()*5) + 5);
        var discount_msg = "<p>Your Discount is "+ discount +"%</p>";
        alert(discount_msg);
        $(this).append(discount_msg);
3. .....
    });
});
</script>

```



index.html

为什么我认为语句应该放在那里：



你的任务是确定将remove语句放在哪里。将这个语句写在下面编号为1、2或3的行上，然后解释为什么把语句放在那里。考虑什么时候需要删除段落，然后使用排除法选出正确的位置。

```
<script>
$(document).ready(function() {
1. ....
  $(".guess_box").click( function() {
2.   $(".guess_box p").remove();
    var discount = Math.floor((Math.random()*5) + 5);
    var discount_msg = "<p>Your Discount is " + discount + "%</p>";
    alert(discount_msg);
    $(this).append(discount_msg);
3. });
  });
});
</script>
```



index.html

为什么我认为语句应该放在那里：

*remove语句不能放在第2行，因为这在guess_box的click函数之外。
这个语句也不能放在第三行，因为它会删除我刚刚追加的内容。我希望在生成一个消息之前删除上一个折扣消息，所以把它作为guess_box click代码块（大括号内）的第一行。*



JQuery调用的顺序和时间很重要。

如果要向访问者提供重要的信息，然后再将其删除，则这种情况下jQuery调用的顺序和时间尤其重要。第5章我们还会讨论屏幕效果的时间和顺序问题。

there are no
Dumb Questions

问：有些时候，调用remove方法之后，我在页面的源代码中还能看到已经删除的元素。这是为什么？

答：通常情况下，使用“查看源代码”选项时，浏览器实际上会向服务器发出一个新请求来得到页面的源代码。使用DOM检查工具，如Chrome Developer工具或面向Firefox的Firebug，就可以给出与页面显示相同的DOM。

问：Math.floor和Math.random方法是什么意思？

答：floor方法将一个数取整为最接近的整数，并返回结果。random方法会返回一个介于0和1的随机数。将它乘以一个数时，肯定可以得到介于0和所乘那个数之间的一个数。

问：this是从哪里来的？

答：在很多面向对象编程语言中，this（或self）是一个关键字，可以在实例方法中使用，用来引用调用当前执行方法的那个对象。

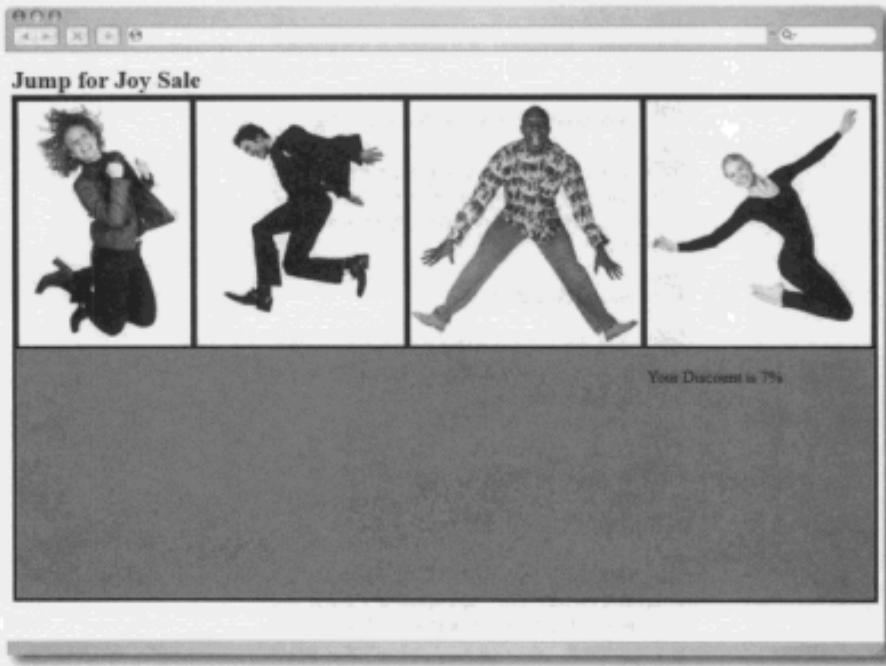
问：这么说，我可以做到有人单击某个图像时才显示一个随机的折扣，当他单击另一个图像时可以将前一个折扣消息删除。不过，实际上要达到我们的最终目的，并不是全部，我们要做的还有很多，是不是？

答：你说的太对了。这确实只是“第一块拼图”。人们结账购买他们的照片时，这个折扣码要一直存在。为了把这个折扣码传送到购物车，需要把这个折扣码再发回给服务器进行处理。我们将在第8章到第10章更详细地研究这种功能。



试一试

将刚才编写的代码行增加到index.html文件。然后在你喜欢的浏览器中打开这个页面，确保一切正常。多单击几次，检查联接到底层discount变量的随机数也正常，上一个折扣值会正确删除，然后才追加新的折扣码。



更多资源请访问稀酷客(www.ckook.com)

回到“Jump for Joy”

祝贺你！你已经满足了所有需求，促销活动已经顺利开展。

- 页面应当有4个部分，每个部分分别显示一个“Jump for Joy”图像。
- 这些部分应当是可单击的。
- 需要一个消息，指出“Your Discount is”（你的折扣是），并提供一个随机的折扣值（介于5%~10%）。
- 用户单击其中一个部分时，这个消息应当出现在这个部分中图像的下方。
- 如果用户再次单击，则要去除上一个消息，并显示一个新消息。

来自: Emily

主题: 回复: Jump for Joy促销活动！

感谢你为我做的一切，实在太谢谢了。我的网站变得好多了。

希望人们能像我喜欢这个新页面一样喜欢我的照片！

--

Emily

> 又及：附上我看到这个新Web页面时用相机拍下的自拍照…… 猜猜我要做什么，不过没有奖励哦！





jQuery工具箱

你已经掌握了第2章，现在你的工具箱里又增加了jQuery选择器和jQuery方法的基础知识。

`$(this)`

选择“当前”元素。

在整个代码中，`$(this)`的含义会改变，取决于它在哪里引用。

jquery方法

方法 — jquery方法是jquery库中定义的可重用的代码。

在jquery和JavaScript中可以使用方法完成工作。可以把方法认为是一个动词，方法完成的就是Web页面动作。

`append` — 将指定的内容插入到DOM。内容将增加到调用这个方法的元素的最后。

`remove` — 从DOM删除元素。

选择器

`$(this)` — 选择当前元素。

`$("div")` — 选择页面上的所有div元素。

`$("div p")` — 选择div元素中的所有p元素。

`$(".my_class")` — 选择所有my_class类的元素。

`$("div.my_class")` — 只选择my_class类的div元素。（不同类型元素可能有相同的类）。

`$("#my_id")` — 选择ID为my_id的元素。

在页面上有所作为



利用jQuery，可以轻松地为Web页面添加动作和交互性。这一章中，我们将了解人们与页面交互时如何让页面做出反应。通过运行代码来响应用户的动作，这会让你的网站更上一层楼。我们还会讨论如何建立可重用的函数，这样就可以编写一次代码，然后多次使用。

你的jQuery技能炙手可热

Emily很满意你为Jump for Joy促销活动所做的工作，不过她刚与会计谈了谈，希望你再做几处修改。

来自：Jump for Joy

主题：回复：Jump for Joy促销活动

你好啊，

你太棒了，多亏了你，我们的Web促销活动正在顺利进行中！最近我和会计碰了个头，聊了聊如何让这个促销活动更加成功。

会计建议对应用做几处修改，这会带来更大销量。

访问者仍有4个选项来得到一个折扣。不过，现在每次的折扣都相同。会计建议在顾客结帐前对访问者购买的照片打个八折。这对他们应该更有吸引力。

访问者找到折扣码的机会应该只有一次。每次访问时折扣码应该是随机的。如果访问者单击时找到了折扣码，继续后面的操作之前要在屏幕上把他找到的折扣码显示出来，否则显示藏有折扣码的方框。

相信你一定能做到，就像第一部分一样出色，没问题吧？

—

Emily Saunders

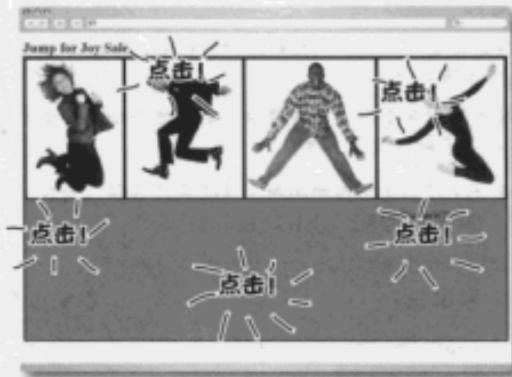
[jumpforjoyphotos.hg](#)

Emily带来了几张会计的照片，不过她没办法让他高兴得跳起来。不知道你对网站的修改能不能让他兴奋起来？



管钱的人总是见解独到……

只有一个方框提供促销，这样Emily就不用分发太多的折扣码，而且人们也没有必要为了找更好的折扣不停地在网站上单击。看起来这些新特性都与单击有关……



Sharpen your pencil



又要整理需求列表了。你应该知道要做些什么，再仔细看看Emily的邮件，找出她要求的所有新特性。用自己的语言写出你认为各个特性分别是什么。

需求：

Sharpen your pencil

Solution

又要整理需求列表了。你应该知道要做些什么，再仔细看看Emily的邮件，找出她要求的所有新特性。

需求：

用自己的语言写出你认为各个特性分别是什么。以下是我们的答案。

- 折扣只应在4个图像框之一中出现，另外每次页面加载时，图像应当出现在不同的（随机）框中。
- 访问者加载页面时只有一次找到折扣的机会，所以要避免他们为了找到一个更好的折扣而多次单击。
- 访问者做出猜测并单击一个方框后，要给出答案，指出猜测是否正确。如果选对了，就要显示他的订单折扣。
- 订单会有一个标准的20%的折扣，不再是可变的。所以不要给出一个百分数，要为访问者提供一个折扣码。

目前的做法



希望这么做



**BRAIN
POWER**

你已经知道如何向页面增加单击事件。不过怎么确保用户只能单击一次呢？

另外你已经在前几章了解到如何在单击时运行代码。你能想办法帮我们完成任务吗？

让页面响应事件

Jump for Joy应用的关键就是单击。在jQuery和JavaScript中，称单击为一个“事件”（当然还有很多其他事件，不过对于我们来说，现在只强调单击事件）。利用事件机制，页面上发生某个事件时（如用户单击了一个按钮），可以运行一段代码。运行的代码就是函数，函数可以让你的jQuery代码更高效、更可重用。稍后我们会更详细地研究函数，不过现在先来看单击事件到底是如何工作的。



更多资源请访问稀酷客(www.ckook.com)

事件监听者的幕后

浏览器利用事件监听者监视页面上做了什么，然后告诉JavaScript解释器是否需要采取行动。

jQuery提供了很多方法，可以很容易地为页面上的任何元素增加事件监听者，所以用户可以不再只是单击链接和按钮了！



绑定事件

为元素增加一个事件时，我们称之为为这个元素绑定一个事件。绑定事件时，事件监听者知道要告诉JavaScript解释器需要调用哪个函数。

为元素绑定事件有两种方法。

方法1

使用这个方法可以在页面加载时为元素增加事件。

这通常称为便利方法。

```
$("#myElement").click(function() {
    alert($(this).text());
});
```

方法2

这个方法的用法与方法1类似，不过，利用这种方法，还可以在页面加载之后为元素增加事件，如创建新的DOM元素时。

这两个方法都返回为
myElement的元素增
加一个单击事件监听
者。

```
$("#myElement").bind('click', function() {
    alert($(this).text());
});
```



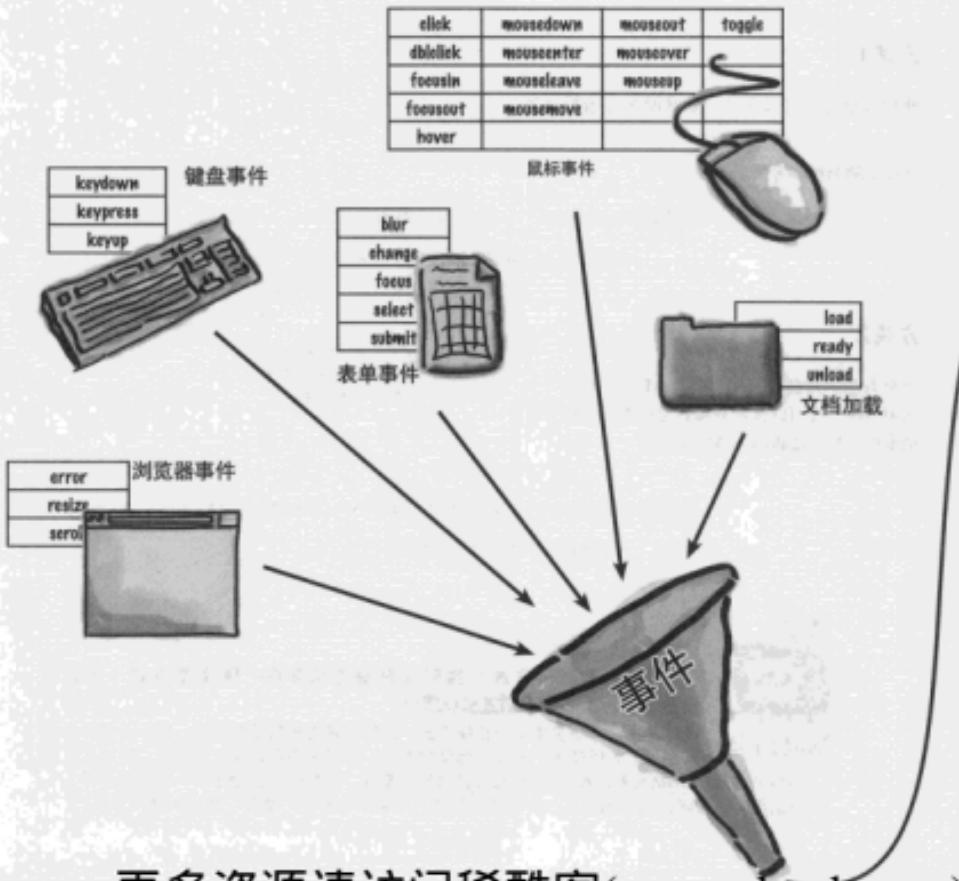
Watch it!

方法1（便利方法）实际上就是方法2的一种快捷方式，不过只有当DOM已经存在时才可以使用。

jQuery提供了很多类似的快捷方式，以便于建立更简洁的代码。由于这些方法只是为了方便使用，所以称为便利方法。不过它们也存在一些限制。对于代码中创建的新DOM元素，就需要使用方法2来增加事件，比如倘若你增加了一个新的可单击的图像，或者向列表增加了一个新的列表项，希望能够与用户交互，就必须使用方法2。

触发事件

任何给定页面上很多情况都可能触发事件。实际上，整个浏览器都是“事件型的”。它的任何部分都可能触发事件！





there are no
Dumb Questions

问：事件中的这些函数用来做什么？

答：它们称为处理函数。处理函数就是事件触发时运行的一个代码块。这一章后面还会更多地了解函数。

问：在哪里可以找到所有不同类型的事件？

答：可以在 jquery.com 网站找到 [Documentation (文档) → Events (事件)]。

问：有多少种不同类别的事件？

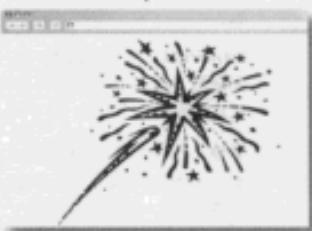
答：jQuery将事件分为5大类：浏览器事件、文档加载事件、表单事件、键盘事件和鼠标事件。

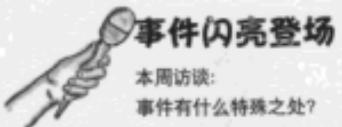
问：到底有多少个不同的事件？

答：如果考虑所有不同类别，大约有30个不同的事件类型。

问：什么能触发页面上的事件（也就是让事件发生）？

答：主要是输入设备（键盘和鼠标）触发不同的事件类型。不过，浏览器、页面文档、jQuery代码，甚至页面上的HTML表单也都可以触发事件。





本周访谈：

事件有什么特殊之处？

HeadFirst：嗨，事件，很高兴能和你聊聊。

事件：能来这里我也很高兴。

HeadFirst：那么，你到底是谁？谁是真正的事件？

事件：现如今，确实到处都有我的身影，不过在jQuery到来之前，我还有些分散。实际上，我就是一个乐于帮助人们与Web页面交互的对象。

HeadFirst：听上去很棒。过一会儿我们再来谈这个话题，现在说说看为什么你有些分散？你到底从哪里来？

事件：说来话长。那还是上个世纪90年代中期，Netscape发布了Navigator 2.0。这个浏览器很棒。当时我还只是一个简单的模型。DOM、JavaScript和我成为新生力量。那时关于我们在浏览器中如何实现还有一个W3C标准呢！

HeadFirst：那可是很久以前了。你走的路可真不短。

事件：没错，我们都走了很长的路。在这个过程中，我们遭遇了Microsoft Internet Explorer与Netscape的浏览器之争。最终Microsoft胜出。不过在分出胜负之前，这两家公司都想用标准中没有规定的一些酷炫技巧超越对方。这些技巧只在它们各自的浏览器中才支持。

HeadFirst：听起来那一段岁月很艰难啊。

事件：确实，不过终于迎来曙光了。1997年，Netscape和Microsoft都发布了各自浏览器的4.0版本。有了很多新的事件，我们对页面可以做更多工作。那真是事件的黄金时光。

HeadFirst：然后呢？

事件：情况失控了。Netscape开始走向开源，后来变成了Mozilla Firefox。不过那时确实还有Netscape，但转瞬即逝。它和Internet Explorer有不同的事件模型。很多事情都只能在某一个浏览器上完成，所以人们访问网站时如果使用了不合适的浏览器，就会看到让他们困惑的结果。Netscape最后终于不复存在了，不过很多其他的浏览器纷纷登场。

HeadFirst：那么为什么现在情况好转了呢？

事件：要说浏览器，情况还不是太理想。Internet Explorer支持的事件与Firefox、Google Chrome、Apple的Safari和Opera不同。不过每个新版本推出时，情况都有改善，它们与标准越来越一致。但是真正的利好消息是jQuery可以为Web开发人员处理这些问题。

HeadFirst：真的吗？太好了！你说过你是……什么对象？

事件：jQuery知道正在使用哪个浏览器，所以它会根据访问网站的人正在使用的浏览器来确定如何处理事件。至于对象，并没有特别之处。实际上，对象就是把变量和函数结合在一个结构中。

更多资源请访问稀酷客(www.ckook.com)

HeadFirst: 从哪里可以更多地了解这些变量和函数呢？

事件: 可以通过jQuery的官方文档来了解我，地址是<http://api.jquery.com/category/events/event-object/>。

HeadFirst: 非常感谢！我肯定会去看看的。我们来谈谈如何在页面上使用事件，好吗？

事件: 嗯，首先，必须把我绑定到某个元素上，这样事件监听者就能知道要监听我。然后，必须有什么来触发我，这样我才能在事件发生时运行所指定的代码。

HeadFirst: OK，不过你怎么知道要运行什么代码呢？

事件: 把我绑定到一个元素时就会知道。为一

个元素绑定事件时可以调用任何代码。正是因为这一点，我才这么有用。另外还可以解除我与元素的绑定。如果解除了绑定，事件监听者不再监听这个元素上的事件，所以触发我时就不会运行指定的代码了。

HeadFirst: 听上去很不错。不过我们的时间不多了。在哪里可以找到更多有关你以及页面上事件类型的资料呢？

事件: 之前给你的链接会解释我为什么是一个对象。jQuery网站上还有更多关于我和所有事件类型的信息，就在Documentation（文档）下面。

HeadFirst: 感谢你的到来。真心希望能在我们的代码中用到你。



这一章后面将深入地了解函数和变量。

另外，我们还会在后面几章进一步学习JavaScript。

删除事件

类似于为元素绑定事件，通常还需要为元素删除事件。例如，如果不希望用户将表单上的一个提交按钮单击两次，或者如果希望用户在页面上做某件事时只能做一次，就要考虑删除事件。这正是新Jump for Joy应用的需求。

元素绑定了一个事件之后，还可以从元素将这个事件删除，使事件不再触发。

删除一个事件：

`unbind`命令告诉Web浏览器不再对该元素监听这个特定事件。

```
多处myElement的  
运行这个代码。
$("#myElement").bind('click', function() {
    alert($(this).text());
});
$("#myElement").unbind('click');
```

为ID为myElement的元素
增加一个click事件监听者。

从myElement删除
click事件。

删除所有事件：

```
$("#myElement").bind('focus', function() {
    alert("I've got focus");
});
$("#myElement").click(function() {
    alert('You clicked me.');
});
$("#myElement").unbind();
```

为ID为myElement的元素
增加一个focus事件监听者。

为ID为myElement的元素
增加一个click事件监听者。

告诉浏览器不再监听
myElement的事件。



这么说，事件监听者就是生活在浏览器里，累积到元素，等待事件发生。然后当事件确实发生时告诉 JavaScript解释器做些事情，对吗？

没错！说得太对了！

下面来看事件如何帮我们解决第一个需求。

- 访问者加载页面时只有一次找到折扣的机会，所以要避免他们为了找到一个更好的折扣而多次单击。

不允许用户再尝试寻找折扣。

Sharpen your pencil

利用你对\$(this)的了解和目前掌握的事件知识，更新上一章代码，加入必要的代码从

元素删除click事件。

```

$(".guess_box").click( function() {
    $(".guess_box p").remove();
    var my_num = Math.floor((Math.random()*5) + 5);
    var discount = "<p>Your Discount is "+my_num+"%</p>";
    $(this).append(discount);
});
```



index.html

更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

Solution

利用你对\$(this)的了解和目前掌握的事件知识，更新上一章代码，加入必要的代码从div元素删除click事件。

```

$( ".guess_box" ).click( function() {
    $(".guess_box p").remove();
    var my_num = Math.floor( (Math.random()*5) + 5 );
    var discount = "<p>Your Discount is "+my_num+"%</p>";
    $(this).append(discount);
    // 首先浏览器不希望听命令元
    // 素的事件。
    $(this).unbind("click");
});

```



index.html



试一试

用这个新代码更新index.html文件，保存，任意单击鼠标，确认一切都
在预想之中。

更多资源请访问稀酷客(www.ckook.com)

Jump for Joy Sale



Your Discount is 30%

参加欢乐跳跃活动，每个顾客都可以获得
30% 的优惠券！

我悲伤的页面的反应可能和我的差
不多。根本不像你这样的那样……



你说的对，现在还没有从所有元素删除单击事
件。

这只是从单击的那个方框删除了单击事件。还可
能单击其他方框。如果能避免单击所有其他元素
就好了……

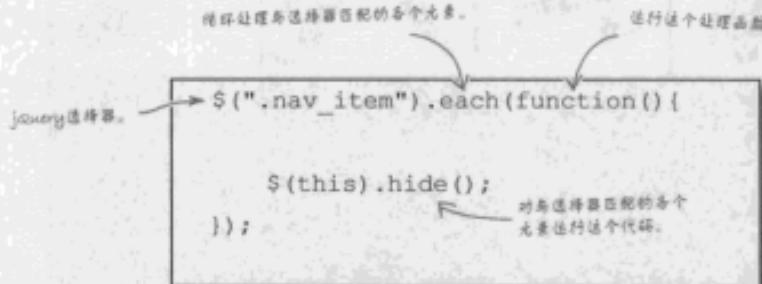


访问者单击一个方框之后如何删除所
有方框的单击事件？是不是得逐个地
处理每一个元素？

遍历处理动作元素

我们经常要与一组元素逐个地交互。

好在jQuery允许我们根据所选择的任何选择器循环处理一组元素。循环，也称为迭代，就是一次一个地遍历一组元素，在这个过程中对各个元素分别做某种处理。



本书后面还会更深入地讨论迭代。

there are no Dumb Questions

问：可以在代码中触发事件吗？

答：当然可以！这种做法很常见。有很多这样的例子，比如提交表单来完成验证、隐藏模式弹出框等。

问：那么，到底如何触发一个事件呢？

答：与jQuery中大多数特性类似，jQuery的创造者会尽可能地让我们便于记忆。可以结合选择器直接使用.trigger方法。例如，`$('.button:first').trigger('click');` 或者`$('#form').trigger('submit');`。

问：可以在Web页面中使用事件而不使用jQuery吗？

答：嗯，这是可以的。使用jQuery只是为了更容易地对元素绑定事件，因为它是跨浏览器兼容的，而且提供了一些

易于使用的函数来帮助为元素绑定事件。

问：.each如何工作？

答：.each会使用调用它的选择器，创建由这个选择器标识的一个元素数组。然后按顺序循环处理这个数组中的各个元素。不过，不用担心，后面就会解释数组和循环！

问：嗯，我知道页面加载之后可以用jQuery创建元素。这些元素也能有事件吗？

答：对。它们也可以有事件。创建一个元素之后，还可以使用.bind方法为它提供一个事件监听者。另外，如果你提前知道一个元素与之前已经创建的其他元素有类似的行为，还可以使用.live方法。这会时与当前选择器匹配的所有元素关联事件处理器程序，包括现有的元素和将来的元素。甚至时还未增加到DOM的元素也同样奏效。

Sharpen your pencil



编写代码，使用迭代删除Jump for Joy页面上所有可单击框的单击事件。另外，仔细查看你的代码，看看有没有不再需要的代码。

```

$( ".guess_box" ).click( function() {
    $(".guess_box p").remove();
    var my_num = Math.floor((Math.random()*5) + 5);
    var discount = "<p>Your Discount is "+my_num+"%</p>";
    $(this).append(discount);

    $(this).unbind('click');

});
```



index.html

更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

Solution

在`.guess_box`类上调用`each`方法会循环处理这个类的所有元素。然后可以依次从各个元素解除`click`方法的绑定。另外不再需要`remove`代码了，因为访问者只能单击一次，所以不再有需要删除的元素。

由于只允许人们单击一次，所以
不再需要删除原来的元素！

```
$(".guess_box").click( function(){
  $(".guess_box p").remove();
  var my_num = Math.floor((Math.random()*5) + 5);
  var discount = "<p>Your Discount is "+my_num+"%</p>";
  $(this).append(discount);
});

$(".guess_box").each( function(){
  $(this).unbind('click');
});
```

循环处理每个`guess_box`元素。
并从各个元素删除`click`事件。



index.html



等一下！
你的HTML文件中脚本是不是有点
太多了？在继续之前，需要想办法
瘦身……



能不能从我们的HTML建立一个单独的脚本文件？对CSS我们就是这么做的……

这个想法很好。实际上，为jQuery代码创建一个单独的文件有很多理由：

- ① 可以包含在多个页面中（代码重用）。
- ② 页面会更快地加载。
- ③ 编写的HTML代码会更简洁，更容易阅读。



Exercise

你已经知道如何包含CSS文件，另外也见过如何包含jQuery库。包含你自己的JavaScript/jQuery文件并没有不同！

你的Web根目录中应该已经有一个名为scripts的文件夹（jQuery库要放在这里）。

- ① 使用你喜欢的文本编辑器，创建一个名为my_scripts.js的文件，保存到scripts文件夹中。
- ② 从index.html文件抽取出所有JavaScript和jQuery代码，转移到这个新文件中。这个新文件中不需要加入<script>和</script>标记。
- ③ 在HTML页面中创建这个文件的链接，把下面的代码放在结束</body>标记前面：

```
<script src="scripts/my_scripts.js"></script>
```



你已经知道如何包含CSS文件，另外也见过如何包含jQuery库。包含你自己的JavaScript/JQuery文件并没有不同！

- ① 使用你喜欢的文本编辑器，创建一个名为my_scripts.js的文件，保存到scripts文件夹中。
- ② 从index.html文件抽取出所有JavaScript和jQuery代码，转移到这个新文件中。这个新文件中不需要加入<script>和</script>标记。

```
$ (document).ready(function() {
    $(".guess_box").click( function() {
        var my_num = Math.floor((Math.random()*5) + 5);
        var discount = "<p>Your Discount is "+my_num+"%</p>";
        $(this).append(discount);

        $(".guess_box").each( function(){
            $(this).unbind('click');
        });
    });
});
});
```



my_scripts.js

现在如果在这个项目的所有HTML文件中包含这个文件，就能访问相同的jQuery代码，而不需要一个文件，就能重用这些代码了。
在每一个文件中重用这些代码了。

⑤

在HTML页面中创建这个文件的链接，把下面的代码放在结束</body>标记前面：

```
<!DOCTYPE html>
<html>
<head>
<title>Jump for Joy</title>
<link href="styles/styles.css" rel="stylesheet">
</head>
<body>
<div id="header">
<h2>Jump for Joy Sale</h2>
</div>
<div id="main ">
<div class="guess_box"></div>
<div class="guess_box"></div>
<div class="guess_box"></div>
<div class="guess_box"></div>
</div>
<script src="scripts/jquery.1.6.2.min.js"></script>
<script src="scripts/my_scripts.js"></script>
</body>
<html>
```



index.html

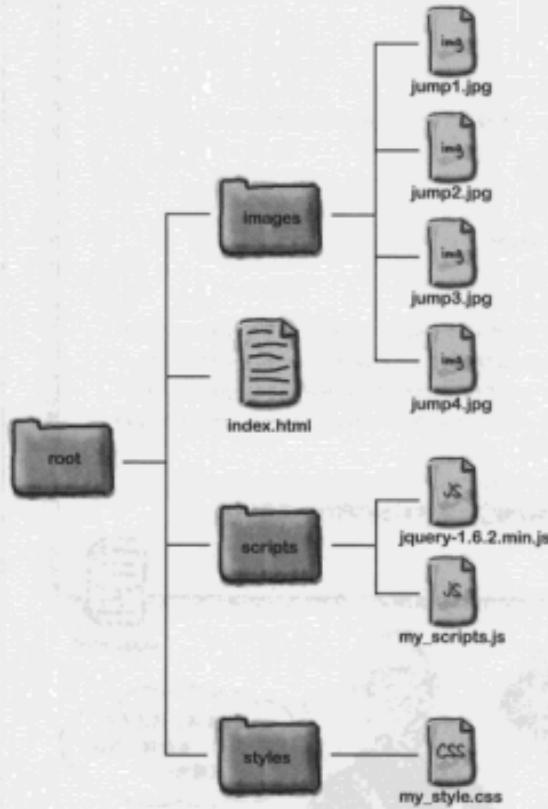


D

大功告成，这还是不多。简洁，有条理。继续
前进吧……

项目结构

你刚刚对文件的结构做了一些重要的改变。
下面来看这些文件如何组织在一起。上一次
查看项目结构之后我们又增加了很多内容。



there are no
Dumb Questions

问：为什么这个文件的扩展名是.js?

答：因为jQuery就是一个JavaScript库，包含我们编写的所有的代码时就像是包含JavaScript代码。

问：这对于提高页面的速度有什么帮助？

答：如果你的.js文件包含在多个HTML文件中，浏览器只会请求一次。它会把这个.js文件存储在浏览器缓存中，这样每次我们访问另一个引用这个脚本文件的HTML页面时，就不用再向服务器请求这个脚本文件了。

问：为什么my_scripts.js文件中不需要<script>和</script>标记？

答：<script>和</script>标记是HTML标记。因为my_scripts.js已经作为一个JavaScript文件包含在页面中，浏览器很清楚这个文件中有些什么。



jQuery磁贴

使用下面的磁贴组织你的项目文件，了解如何分解HTML、CSS和jQuery代码。要保证完全正确。

```
<!DOCTYPE html>
<html>

    <title>Jump for Joy</title>
    <link href="styles/styles.css" rel="stylesheet">
</head>

<div id="header">
    <h2>Jump for Joy Sale</h2>
</div>

<div id="main">
    <div .....></div>
    <div class="guess_box"></div>
    <div class="guess_box"></div>
    <div .....></div>
</div>
<script src="scripts/....."></script>
<..... src="scripts/my_scripts.js"></script>
</body>
<html>
```

my_num

this

117

<head>

".guess_box"

</div>

```
$(document).ready(function() {
    $().click(function() {
        var ..... = Math.floor((Math.random()*5) + 5);
        var discount = "<p>Your Discount is "+my_num+"%</p>";
        $().append(discount);

        $(".guess_box").each(function(){
            $(this).unbind("click");
        });
    });
});
```



index.html

jquery-1.6.2.min.js

<body>

class="guess_box"

script

class="guess_box"

更多资源请访问稀酷客(www.ckook.com)

你现在的位置：



jQuery磁贴答案

使用下面的磁贴组织你的项目文件，了解如何分解HTML、CSS和jQuery代码。要保证完全正确。

```
<!DOCTYPE html>
<html>
  <head>
    <title>Jump for Joy</title>
    <link href="styles/styles.css" rel="stylesheet">
  </head>
  <body>
    <div id="header">
      <h2>Jump for Joy Sale</h2>
    </div>
    <div id="main">
      <div class="guess_box" ></div>
      <div class="guess_box" ></div>
      <div class="guess_box" ></div>
      <div class="guess_box" ></div>
    </div>
    <script src="scripts/ jquery-1.6.2.min.js " ></script>
    <script src="scripts/my_scripts.js"></script>
  </body>
</html>
```



index.html

```
$ (document) .ready(function() {
  $(".guess_box").click( function() {
    var my_num = Math.floor((Math.random()*5) + 5);
    var discount = "<p>Your Discount is "+my_num+"%</p>";
    $(this).append(discount);

    $(".guess_box") .each( function(){
      $(this).unbind('click');
    });
  });
});
```



my_scripts.js



如果只写一次jQuery代码，然后只要需要就能反复使用就好了，这可能只是个美梦吧？不过我知道这只是异想天开……

利用函数实现功能

既然已经了解如何在页面上增加和删除事件，下面来看另一个重要的特性，利用这个特性，可以帮助我们用jQuery掌控我们的网站，这个特性就是函数。

函数是独立于其余代码的一个代码块，可以放在脚本中你希望的任何位置来执行。

不管你是否相信，我们整本书中都在使用函数。还记得这些吗？

```
$(document).ready(function() {  
    $("#clickMe").click(function() {  
        //Do stuff in here!  
    });  
    $(".guess_box").click(function() {  
        //Do stuff in here!  
    });  
});
```

这些都是本书目前
为止的代码中调用
过的。

看这些函数！

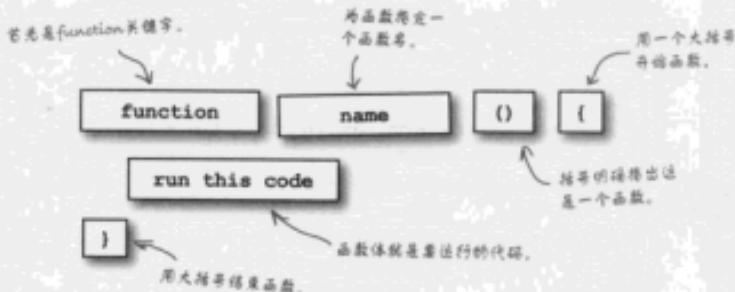
jQuery为我们提供了大量函数，不过你也可以编写自己的定制函数，提供jQuery未提供的特性。通过创建定制函数，可以再次使用自己的代码。要运行函数代码时只需要调用函数名，而无需在脚本中重复全部代码。

利用定制函数可以
按名组织jQuery代
码块，以便重用。

更多资源请访问稀酷客(www.ckook.com)

函数剖析

要创建一个函数，需要使用一致的语法，将函数名与它运行的代码关联起来。这是最基本的JavaScript函数的语法：



指定函数名

指定函数名有两种方法。

函数声明

第一种方法是函数声明，即定义一个命名函数变量，而不需要变量赋值。函数声明以function开头，如下所示：

```
function myFunc1() {  
    $("div").hide();  
}
```

函数表达式

命名的函数表达式将函数定义为一个更大表达式语法（通常是一个变量赋值）的一部分：

```
var myFunc2 = function() {  
    $("div").show();  
}  
执行到这里会有一个变量赋值。
```

函数名？不过我们以前使用的所有函数都没有函数名。为什么现在又要开始指定函数名了呢？

问得好。

通过对函数命名，就能在代码中多次调用函数。未命名的函数（也称为匿名函数）使用很受限。下面会更详细地分析匿名函数。了解没有函数名会有什么限制。



匿名函数

匿名或自执行函数没有函数名。一旦在代码中遇到匿名函数就会立即调用。另外，这些函数中声明的所有变量只在该函数运行时才可用。

不能在代码中的其他位置调用这个函数。

```
$ (document).ready(function() {  
    $(".guess_box").click( function() {  
  
        var my_num = Math.floor((Math.random()*5) + 5);  
        var discount = "<p>Your Discount is "+my_num+"%</p>";  
        $(this).append(discount);  
  
        $(".guess_box").each( function() {  
            $(this).unbind('click');  
        });  
    });  
});
```



my_scripts.js

如果想在别处使用这个代码，就必须重写代码。



Watch it!

由于没有为这个函数指定一个函数名，所以无法在代码中的其他位置调用这个函数。

问：函数声明与命名函数表达式有什么区别？

答：主要区别就在于调用的时间。尽管它们做的事情是一样的，但是如果一个函数声明为命名函数表达式，那么在遇到并定义这个函数之前就不能在代码中使用。另一方面，如果一个函数使用函数声明方法来定义，则可以在页面上你希望的任何时候调用。甚至可以作为一个onload处理程序。

问：对于函数名有没有什么限制？

答：有。函数不能以数字开头，另外不能使用任何数学运算符，也不能使用除下划线以外的任何标点符号。空格不能出现在函数名中，函数名和变量名都是区别大小写的。

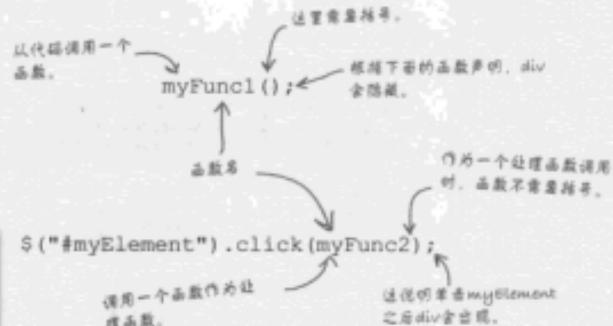
更多资源请访问稀酷客(www.ckook.com)

命名函数作为事件处理程序

之前我们已经看到匿名函数可以用做事件的处理函数。还可以使用我们自己的定制命名函数作为事件处理程序从代码中直接调用。下面更详细地分析前几页命名的两个函数。

函数声明

```
function myFunc1() {
    $("div").hide();
}
```



函数表达式

```
var myFunc2 = function() {
    $("div").show();
}
```



jQuery磁贴

看着你能不整理这些磁贴，把检查折扣码的代码移到一个单独的函数中（名为`checkForCode`），这个函数将用作这些方框上单击事件的处理程序。

```
$ (document).ready(function() {
    $(".guess_box").click( checkForCode );

    var my_num = Math.floor((Math.random()*5) + 5);
    discount = "<p>Your Discount is "+my_num+"%</p>";
    $(this).append(discount);

    $(".guess_box").each( function() {
        $(this).unbind("*");
    });
});
```

"+my_num+"

function

1

discount

0

{ { click

checkForCode

append

)

var





jQuery磁贴答案

看着你能不能整理这些磁贴，把检查折扣码的代码移到一个单独的函数中（名为checkForCode），这个函数将用做这些方框上单击事件的处理程序。

我们的命名函数作为事件
处理程序。

函数声明

```
$ (document).ready(function() {
    $(".guess_box").click( checkForCode );
    function checkForCode () {
        var my_num = Math.floor((Math.random()*5) + 5);
        var discount = "<p>Your Discount is " + my_num + "%</p>";
        $(this).append( discount );
    }
    $(".guess_box").each( function(){
        $(this).unbind('click');
    });
});
```

用分号结束语句。
第二步通过随机生成器。
在屏幕上显示折扣。

my_scripts.js
这些磁贴没有用到。

动手做！

干得不错！用这个代码更新你的代码文件，现在你已经创建了你的第一个函数，并把它用做单击事件的处理函数。



如何创建函数把折扣码隐藏在一个随机的方框中。另外如何创建函数为折扣码本身生成一个随机数？

中身可以使用这个随机数生成器。
checkForCode函数以且将要调用的函数中
的参数为随机数的一个随机的字符串。

更多资源请访问稀酷客(www.ckook.com)



所以，实际上我们希望根据参数的不同方根
法函数做不同的事情……怎么做到呢？

有时我们希望函数反复地完成一个任务，不过结果要根据我们提供的信息有所变化。

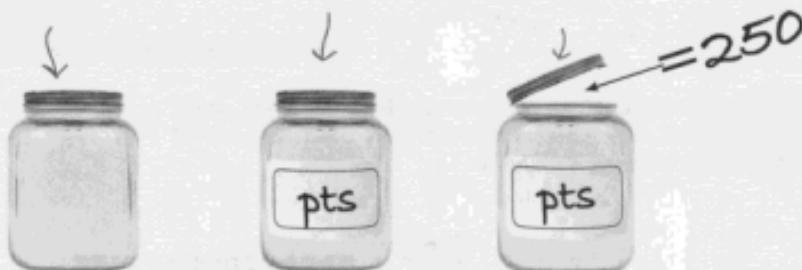
函数可以接受传入的变量。还记得第2章介绍过，变量用来存储可能随时间改变的信息。我们已经讨论过变量，下面再来回顾变量是如何工作的。

利用var关键字声明一个变量。
var pts = 250;
var关键字后声明变量
命名；
代码中要这样设置变量的值。

声明一个变量时，JavaScript脚本会为我们提供一些浏览器内存，可以在其中存储信息。

命名一个变量，以便在脚本中引用。

使用等号将值放入变量。



我们的代码中已经使用了一些变量。还记得吧？

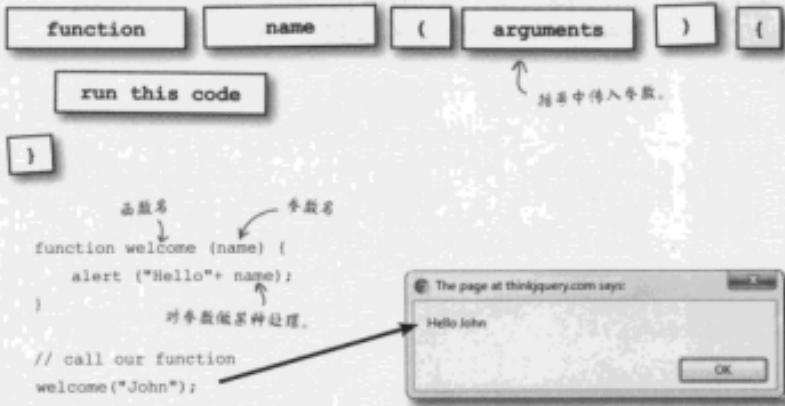
```
var my_num = Math.floor((Math.random() * 5) + 5);

var discount = "Your Discount is "+my_num+"%";
```

更多资源请访问稀酷客(www.ckook.com)

向函数传递变量

将变量增加（或传递）到函数时，这些变量称为参数[有时可能会把它们称为形参（parameter）。下面进一步介绍如何向函数传递参数。



函数不需要知道变量中包含什么；它只是显示变量中当前存储的内容。这样一来，只需要改变变量，你就能改变函数显示的内容，而不必修改函数（如果不修改函数，这肯定不是一个可重用的函数）。



结合变量和函数可能让人有点畏惧。

不过，实际上，可以把函数想成是一个食谱—这里假设要用来制作一种饮料。为制作这种饮料，你已经有一些可重复的基本步骤：一点点这种材料，一点点那种材料，一起搅拌等，这些将构成你的函数，原料就是要传入的变量。来点金汤力鸡尾酒，怎么样？

更多资源请访问稀酷客(www.ckook.com)

函数还可以返回值

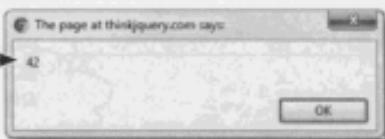
从函数返回信息要使用return关键字，后面是要返回的内容。结果会返回给调用这个函数的代码，以便在其余代码中使用。

```
函数名      参数名
          ↗       ↘
function multiply (num1, num2) {
    var result = num1*num2;
    return result;  ↗ 对参数做某种处理。
}
                                     ↙ 返回一个值。
```

```
// Call our function
var total = multiply (6, 7);

alert (total);
```

返回类型可以是一个数，一个字符串，或者甚至可以是一个DOM元素（对象）。



Sharpen your pencil

既然能够创建你自己的函数，下面在my_scripts.js文件中增加一个新函数，接受一个参数（名为num），然后根据给定的参数返回一个随机数。这个函数命名为getRandom。



你将在哪里
找到它。当你运行它时，你需要先将编辑器的代码。你想要，最好使用这个步骤，在哪里

my_scripts.js

更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

Solution

既然能够创建你自己的函数，下面在my_scripts.js文件中增加一个新函数，接受一个参数（名为num），然后根据给定的参数返回一个随机数。这个函数命名为getRandom。

```

    函数名           参数名
    ↓             ↓
function getRandom(num) {
    var my_num = Math.floor(Math.random()*num);
    return my_num;
}
    )           返回随机数。

```

my_scripts.js

看上去不错！不过，还有很多工作要做。下面来是你的团队对于如何检查“正确的”方框有什么意见……

OK，这些函数和参数都没问题，不过离我们的目标更近了吗？Jump for Joy页面的代码完成了吗？



Jim：嗯，除了这个新的getRandom函数，我们还需要一个……

Frank：……没错，还需要一个函数把折扣码放在一个随机框里，它可以使这个getRandom函数。

Joe：有道理。这样一来，单击发生后，我们就能查看用户是否单击了正确的方框。

Jim：等一下，你说什么？我们怎么判断是不是单击了正确的方框呢？

Frank：条件逻辑！

Jim：什么？

Frank：利用条件，我们可以检查一种特定的情况，然后相应地运行代码。

Joe：所以可以这么说：查看一个变量是否有某个值，或者两个值是否相等，是吗？

Frank：完全正确！甚至可以检查一个元素中是否有另一个元素，我想这会对我们有帮助。

使用条件逻辑做判断

jQuery使用了JavaScript的条件逻辑特性。利用条件逻辑，可以使用已有的信息，根据你希望的代码决策运行不同的代码。下面的代码就是使用JavaScript条件逻辑的一个例子。第6章还会介绍其他条件逻辑。

```

开始语句。
          我们要检查的内容
          ↓
if( myBool == true ) {
    // Do Something!
}

js变量           ↓
else{
    // Otherwise Do something else!
}

}

```

相等性操作符。这可以叫做“等于”。
这是检查结果为真时想要执行的代码。
这是检查结果为假时要执行的代码。
说明，并不需要一个else语句来平衡if语句，不过老常一个else语句没有坏处。



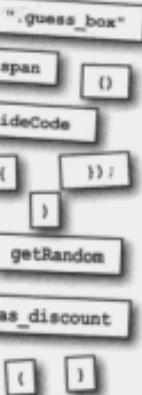
jQuery磁贴

看看能不能组织下面的磁贴，创建一个新的命名函数，名为hideCode，它使用条件逻辑隐藏一个新的span元素，ID为has_discount，每次会随机地把这个元素隐藏在现有的某个可单击的.guess_box div元素中。

```

var ..... = function () {
    var numRand = ..... (4);
    $(...).each(function(index, value) {
        if(numRand == index) {
            $(this).append("<span id='.....'></.....>");
            return false;
        }
    })
}

```



my_scripts.js

你现在的位置：

109

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴答案

看看能不能组织下面的磁贴，创建一个新的命名函数，名为`hideCode`，它使用条件逻辑隐藏一个新的元素，ID为`has_discount`，每次会随机地把这个元素隐藏在现有的某个可单击的`.guess_box`div元素中。以下是我们答案。

我们的命名函数

```
var hideCode = function () {
    var numRand = getRandom(4); // 使用我们的随机数
    $(".guess_box").each(function(index, value) {
        if(numRand == index) {
            $(this).append("<span id='has_discount'></span>"); // 将这个折扣元素增加到.guess_box类元素中。
            return false;
        }
    });
}
```

条件逻辑，将当前列表项置为随机数。
跳出`each()`循环。



my_scripts.js



Watch it!

元素在列表中的索引是指它出现在列表中的位置。

索引总是从0开始。

所以，列表中的第一个元素索引为0，第二个元素索引为1，依此类推。第6章学习数组和循环时还会了解使用索引的更多内容。



Frank：没错，它们是很有用。不过，既然已经把折扣码隐藏起来了，你还能把它再找出来吗？

Jim：噢，是啊，问得好。我真不知道。

Joe：我想我们又得用到这种条件魔法吧？

Frank：完全正确。现在不再从`.guess_box`元素列表中选择一个随机的索引，我们必须再来循环处理，查看其中是否包含我们的`has_discount`元素。

Joe：“包含？”嘿，Frank，你可能得解释一下。

Frank：好的。下面来看看具体的jQuery代码行。

动手做！

更新`checkForCode`函数，包含根据Jim、Frank和Joe的发现增加的新代码。

条件逻辑，查看用户是否找到折扣。

`$("discount")`变量。

它前元素，也就是调用函数的元素。

查找一个ID为`has_discount`的DOM元素。

这是一个jQuery方法，检查第一个参数的内容中是否包含第二个参数的内容。从这个函数名就可以知道大概了。

```
function checkForCode () {
  var discount;
  if ($().contains(this, document.getElementById("has_discount")) ) {
    var my_num = getRandom(5);
    discount = "<p>Your Discount is "+my_num+"%</p>";
  } else {
    discount = "<p>Sorry, no discount this time!</p>";
  }
  $(this).append(discount);
}

$(".guess_box").each( function() {
  $(this).unbind('click');
});
```

设置消息，根据用户是否找到折扣码，消息会有所不同。



my_scripts.js

更多资源请访问稀酷客(www.ckook.com)

再来动手做!

现在来增加一些定制函数：一个用来生成随机数，一个
用来隐藏折扣码，还有一个用来检查折扣码。

```

$(document).ready(function() {
    $(".guess_box").click( checkForCode );
    function getRandom(num) {
        var my_num = Math.floor(Math.random()*num);
        return my_num;
    }
    var hideCode = function() {
        var numRand = getRandom(4);
        $(".guess_box").each(function(index, value) {
            if(numRand == index){
                $(this).append("<span id='has_discount'></span>");
                return false;
            }
        });
        hideCode(); //调用这个命名函数.....      .....它会告诉隐藏折扣码是什么。
    }
    function checkForCode() {
        var discount;
        if($.contains(this, document.getElementById("has_discount")) )
        {
            var my_num = getRandom(5);
            discount = "<p>Your Discount is "+my_num+"%</p>";
        }else{
            discount = "<p>Sorry, no discount this time!</p>";
        }
        $(this).append(discount);
        $(".guess_box").each( function(){
            $(this).unbind('click');
        });
    }
}); //End document.ready()

```



my_scripts.js

更多资源请访问稀酷客(www.ckook.com)

Jump for Joy还需要更多帮助

你可能以为Jump for Joy活动已经大功告成了，不过看起来Emily又有了一些需求……

来自: Jump for Joy

主题: 回复: Jump for Joy促销活动

嗨，

感谢你一直以来所做的。

我想知道，有没有办法在人们单击方框之前让它突出显示？这样他们就能知道目前在哪个方框上。单击之前可以不那么茫然。

另外，可不可以不是简单地弹出折扣码，而是在屏幕上把折扣码放在方框下面一个容易看到的区域中？折扣码可以是一些文本再加一个数吗？我觉得这样可能比较好……唉，对了，那个数不要限制在1到10之间好吗，能不能更大一些？比如最多到100？

如果你觉得能完成这些小改动，那么通知我一声！

--

Emily Saunders

jumpforjoyphotos.hg

Sharpen your pencil

你知道要做什么。从Emily的邮件里找出所有新需求。

需求：



Sharpen your pencil Solution

你知道要做什么。从Emily的邮件里找出所有新需求。

需求：

- 访问者单击之前突出显示他们目前所在的方框，让他们知道当前的选项是什么。
- 把折扣码放在屏幕上单独的区域中。折扣码应当是一些文本和一个1到100之间的数。

there are no Dumb Questions

问：需要为所有函数指定一个返回值吗？

答：理论上讲不需要。不论是否明确指定，所有函数都会返回一个值。如果你没有告诉函数要返回什么值，则它会返回一个未定义的值。如果代码无法处理一个未定义的值，就会导致一个错误。所以，最好指定一个返回值，即使只是return false。

问：对传入函数的参数有没有限制？

答：没有。可以向函数传入任何对象、变量或者值。传入的参数还可以比函数期望的参数多。多余的参数会被忽略。如果传入的参数太少，未传入的其他参数会自动地设置为未定义。

问：\$.contains方法做什么？

答：这是jQuery库的一个静态方法。它有两个参数。这个方法会检查第一个参数的所有子元素，查看其中是否包含第二个参数。然后返回一个true或false。在这里，\$.contains(document.body,document.getElementById("header"))的结果是true；不过\$.contains(document.documentElement.getElementById("header"),document.body)会返回false。

问：jQuery中的静态方法是什么？

答：这表示这是一个与jQuery库关联的函数，而不是与任何特定的对象关联。调用这个方法不需要任何选择器。只需要jQuery名或者它的快捷方式(\$)就可以了。

问：.each处理函数中的索引和值是什么？

答：索引指示处在循环的那个位置。索引从0开始，对应选择器返回的数组中的第一个元素。值指示当前对象。它等于.each循环中的\$(this)。

问：为什么hideCode函数中的.each循环返回false？

答：.each循环返回false时，会告诉它停止执行循环。如果返回一个非false的值，就会移动到列表中的下一项。对我们来说，由于已经知道隐藏了折扣码，所以可以停止处理其余的元素。



你认为有没有办法在用户单击之前告诉她要选择哪个方框？

方法可以改变CSS

要完成这个任务，需要在用户单击之前突出显示他正停在哪个方框上。

要改变一个元素的外观，最容易的办法就是利用CSS和CSS类。

好在jQuery提供了一种简便的办法，可以为元素指定CSS类，还可以删除CSS类。只需要利用几个很容易使用的方法就能做到。下面来看如何在我

还记得第1章和第2章介绍过的这些
内容吧？



成品代码

创建这些文件，在Jump for Joy文件之外单独保存，以便观察这些方法的具体使用。这会帮助你了解如何在用户单击之前突出显示方框。

```
<html>
  <head>
    <link href="styles/test_style.css" rel="stylesheet">
  </head>
  <body>
    <div id="header" class="no_hover"><h1>Header</h1></div>
    <button type="button" id="btn1">Click to Add</button>
    <button type="button" id="btn2">Click to Remove</button>
    <script src="scripts/jquery.1.6.2.js"></script>
    <script src="scripts/my_test_scripts.js"></script>
  </body>
</html>
```



class_test.html

```
.hover{
  border: solid #f00 3px;
}
.no_hover{
  border: solid #000 3px;
}
```



test_style.css

```
$(document).ready(function() {
  $("#btn1").click( function(){
    $("#header").addClass("hover");
    $("#header").removeClass("no_hover");
  });
  $("#btn2").click( function(){
    $("#header").removeClass("hover");
    $("#header").addClass("no_hover");
  });
});
```



my_test_scripts.js

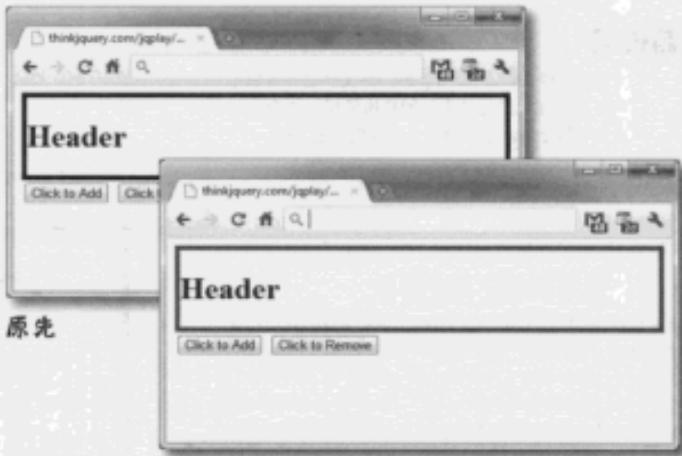
更多资源请访问稀酷客(www.ckook.com)

你现在的位置：



试一试

在浏览器中打开你创建的class_test.html文件。单击Add按钮之后，会为ID为header的div应用这个类。再单击Remove按钮，将这个类删除！



太棒了！如果一切都这么简单就好了。除了
单击事件，这种CSS修改在其他情况下也能
做到吗？

没错，确实如此。而且同样很容易……

你可以对任何事件类型尝试这种CSS修改。不过，
要解决这个问题，还需要另一个事件。再看一看82
页上的列表，看能不能找出需要用到哪个事件。

更多资源请访问稀酷客(www.ckook.com)

增加一个hover事件

hover事件可以有两个处理函数作为参数：一个对应mouseenter事件，另一个对应mouseleave事件。这两个处理函数可以是命名函数，也可以是匿名函数。仔细查看刚才使用的测试脚本，看看当鼠标在元素上时如何使用hover事件对元素应用行为。



my_test_scripts.js

```
$ (document).ready(function() {
    $("#btn1").click( function() {
        $("#header").addClass("hover");
        $("#header").removeClass("no_hover");
    });
    $("#btn2").click( function() {
        $("#header").removeClass("hover");
        $("#header").addClass("no_hover");
    });
});
```

利用removeClass方法可以删除一个元素的CSS类。
利用addClass方法可以为一个元素增加CSS类。它不会影响这个元素已有
的CSS类。



更新my_style.css和my_scripts.js文件，当用户把光标放在图像框上时让该图像框突出显示。需要一个新的CSS类，另外脚本文件中需要两个处理函数（checkForCode函数后面），它们使用addClass和removeClass方法设置这个CSS类。我们先做了一些工作，你只需要在下面写上函数就可以了。



my_style.css

```
$(".guess_box").hover(
    function () {
        // this is the mouseenter event handler
        ...
    },
    function () {
        // this is the mouseleave event handler
        ...
    }
);
```



my_scripts.js



Exercise Solution

用户停在这个方
框上时，使用对
应`mouseenter`事
件的匿名处理函
数应用CSS类。

`removeClass`
jQuery方法允
许从元素删除
一个CSS类。

现在有了一个CSS类，可以用`hover`
事件来管理。

这就是新的CSS类。

```
.my_hover{
    border: solid #00f 3px;
}
```

my_style.css

```
$(".guess_box").hover(
    function () {
        // this is the mouseenter event handler
        $(this).addClass("my_hover");
    },
    function () {
        // this is the mouseleave event handler
        $(this).removeClass("my_hover");
    }
);
```

addClass()方法允许向元素增加一个CSS
类。它不会影响这个元素已有的任何CSS类。

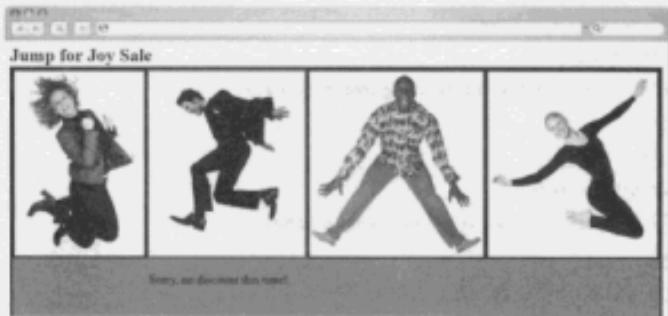
这是对应`mouseleave`事件的
匿名处理函数。

my_scripts.js



试一试

在浏览器中打开index.html文件，其中要包含新的my_scripts.js文件。在图像上移
动鼠标，看看边框有什么变化。



嘿，现在图像
框的颜色变
化了。不过，
还有一个工作
要做……

更多资源请访问稀酷客(www.ckook.com)

就快完成了……

进展很顺利，不过消息出现的位置还不太对，没有按你希望的方式出现。另外，第一封邮件中还有一个需求我们没有解决。目前的需求列表如下：

- 访问者点击它将突出显示他们目前所选的方框，让他知道当前的选项是什么。
- 把折扣码放在屏幕上单独的区域中。折扣码应当是一些文本和一个1到100之间的数。
- 访问者做出猜测并单击一个方框后，要给出答案，指出猜测是否正确。如果选对了，就显示他的订单折扣。

} 这个需求是第一封
邮件中提到的。



更新checkForCode函数，完成最后这3个任务。

1. 把折扣码放在屏幕上单独的区域中。
2. 折扣码是字母与一个介于1到100的数的组合。
3. 向访问者显示折扣码隐藏在哪里（如果她猜错了）。

我们创建了这些CSS类来帮你，可以增加到my_styles.css文件中，指示是否找到了折扣码。

增加了这些CSS类之后，在4个方框下面增加一个ID为result的span元素，用来显示折扣码。

```
.discount{
    border: solid #0f0 3px;
}
.no_discount{
    border: solid #f00 3px;
}
```



my_style.css



Exercise Solution

更新checkForCode函数，完成要求的各项任务：屏幕上有一个单独的位置来显示折扣码，折扣码包括一些文本和一个不超过100的数，另外访问者单击之后要指示折扣码在哪里。

```

function checkForCode() {
    var discount;
    if ($.contains(this, document.getElementById("has_discount")) ) {
        var my_num = getRandom(100);
        discount = "<p>Your Code: CODE" +my_num+"</p>" ;
        // 使用getRandom函数生成折扣码并先加一个
        // 不超过100的数。
    } else{
        discount = "<p>Sorry, no discount this time!</p>" ;
    }
    $(".guess_box").each(function() {
        if ($.contains(this, document.getElementById("has_discount")) )
        {
            $(this).addClass("discount");
        } else{
            $(this).addClass("no_discount");
        }
        $(this).unbind();
    });
    $("#result").append(discount);
} // End checkForCode function

```

使用\$.contains函数。
查看这个对象是否包含折扣码。

如果找到，设置折扣码的外观，告诉人们折扣码是……

……如果没有找到，就直接告诉用户。

使用getRandom函数生成折扣码并先加一个不超过100的数。

设置输出消息，指示是否找到了折扣码。

把输出消息放到上面输入区域中。



my_scripts.js



试一试

既然更新了checkForCode函数，下面来测试Jump for Joy网站上的所有新特性。
(为了便于比较，你的代码应当类似这个文件：http://thinkjquery.com/chapter03/end/scripts/my_scripts.js)。

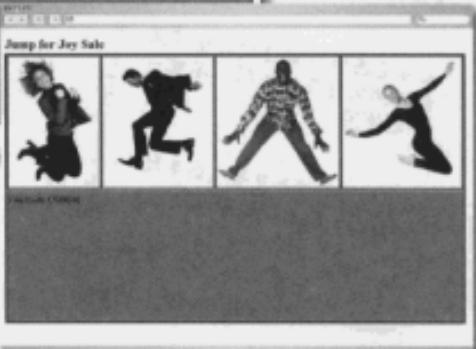


加载的页面



太棒了！这差不多。这肯定对网站大有帮助，还能帮Emily省钱！

没有折扣码



折扣码



更多资源请访问稀酷客(www.ckook.com)



jQuery工具箱

你已经掌握了第3章，现在你的工具箱里又增加了事件、可重用的函数以及条件。

函数

函数是可重用的代码块，可以在代码中的其他地方使用……

……不过前提是这些函数必须是命名的。

未命名函数只能在代码中调用它们的位置正常运行，不能在别处使用。

可以向函数传入变量（或参数）。
函数还可以返回结果。

条件

做具体工作之前测试逻辑条件 ($if xyz = true$)。

如果条件结果为 `false`，通常还有一个 `else` 语句，不过这不是必要的。

事件

事件是帮助用户与Web页面交互的对象。

共有大约30个事件，涉及浏览器上可能触发事件的所有方面。

更多资源请访问稀酷客(www.ckook.com)

修改DOM

我们有相同的父母，可不代表我们完全一样！



页面加载完成并不意味着它会保持相同的结构。第1章中我们已经了解页面加载时如何创建DOM来建立页面的结构。这一章中，我们将介绍如何使用jQuery在DOM结构中上下移动，以及如何处理元素层次结构和父/子关系来动态改变页面结构。

Web镇餐厅需要一个交互式菜单

Alexandra是Web镇餐厅的大厨，现在交给你一个工作。一直以来，她总要为不同版本的菜单维护不同的Web页面：普通菜单和素食菜单。她希望你建立一个页面，能够为素食顾客调整菜单。



墨素食

下面是Alexandra希望你做的工作。

希望有一个“Go Vegetarian”按钮，能自动在我们的Web页面菜单上替换合适的素食选项。

要做以下替换：

— 鱼肉套餐没有替代套餐，所以要把这些套餐去掉。

— 我们为素食顾客提供了大磨菇来替代汉堡。

— 除了汉堡之外，我们为素食顾客提供了豆腐，作为所有其他肉类和蛋类的替代品。

— 需要一个按钮可以把菜单恢复为原来的状态。

又及：如果可以，我们还希望在替换的素食套餐旁边显示一个叶子图标。

我让Web设计人员给你发个邮件，让你看看目前菜单的相关文件。你可以在这个基础上开展工作。

Web设计人员

编写jQuery代码之前，先来看看Web设计人员发给我们的HTML和CSS文件，查看它们的样式和结构是否符合要求。

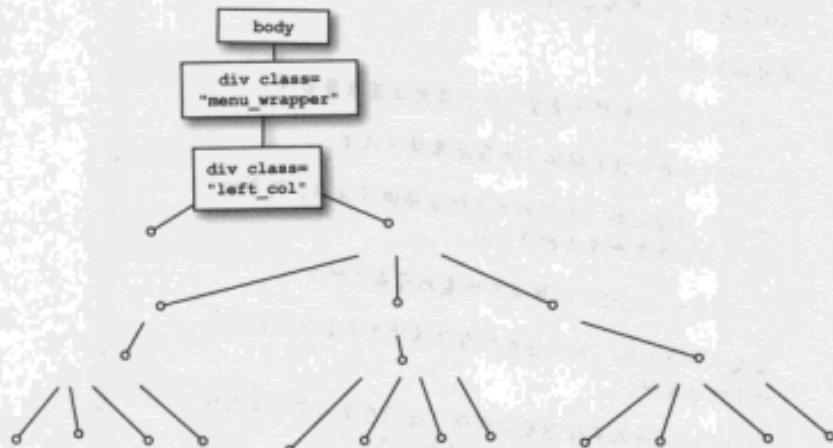
这一次不再提供确定需求的相关练习（因为你可能已经在考虑这个问题了），不过，一定要用你自己的语言把需求写下来，以便之后重构一个什么样的应用。

更多资源请访问稀酷客(www.ckook.com)



DOM树磁贴

仔细研究目前Web菜单的结构，用图表示相应的DOM。你会在下面找到完成这个DOM树所需的全部元素磁贴。使用右页的菜单HTML片段完成这个DOM树。每个磁贴要放在一个空心圆下面。我们已经帮你完成了一部分。



更多资源请访问稀酷客(www.ckook.com)

这只是实际HTML
菜单的一个片段。

```

<body>
  <div id="menu_wrapper">
    <div class="left_col">
      <h4>Dinner Entrees</h4>
      <ul class="menu_entrees">
        <li>Thai-style Halibut
          <ul class="menu_list">
            <li>coconut milk</li>
            <li>pan-fried halibut</li>
            <li>early autumn vegetables</li>
            <li>Thai spices </li>
          </ul>
        </li>
        <li>House Grilled Panini
          <ul class="menu_list">
            <li>prosciutto</li>
            <li>provolone</li>
            <li>avocado</li>
            <li>sourdough roll</li>
          </ul>
        </li>
        <li>Southwest Slider
          <ul class="menu_list">
            <li>whole chiles</li>
            <li>hamburger</li>
            <li>pepperjack cheese</li>
            <li>multigrain roll</li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</body>

```

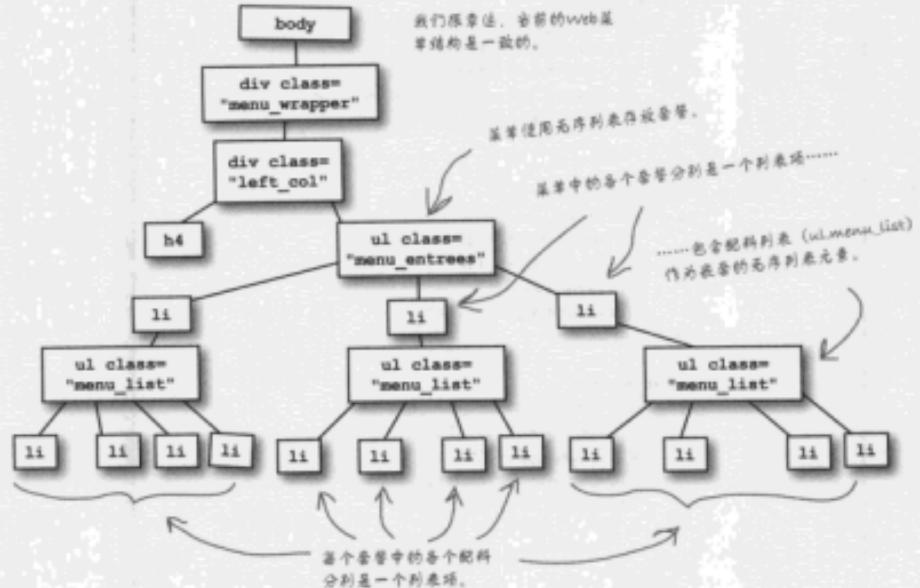


index.html



DOM树磁贴答案

看起来所有配料都作为套餐父列表项的子列表项。原来没有清楚地标出，而且也不唯一，现在呢？



我们甚至写出选择器，查找需要改变的配料。
在这个意义上，它们都是列表元素……

……那么如何区分我们希望替换的配料和其他的配料呢？

```

<body>
  <div id="menu_wrapper">
    <div class="left_col">
      <h4>Dinner Entrees</h4>
      <ul class="menu_entrees">
        <li>Thai-style Halibut
          <ul class="menu_list">
            <li>coconut milk</li>
            <li>pan-fried halibut</li>
            <li>earthy autumn vegetables</li>
            <li>Thai spices </li>
          </ul>
        </li>
        <li>House Grilled Panini
          <ul class="menu_list">
            <li>prosciutto</li>
            <li>provolone</li>
            <li>avocado</li>
            <li>sourdough roll</li>
          </ul>
        </li>
        <li>Southwest Slider
          <ul class="menu_list">
            <li>whole chiles</li>
            <li>hamburger</li>
            <li>pepperjack cheese</li>
            <li>multigrain roll</li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</body>

```

这只是实际
HTML页面的一
个片段。




index.html



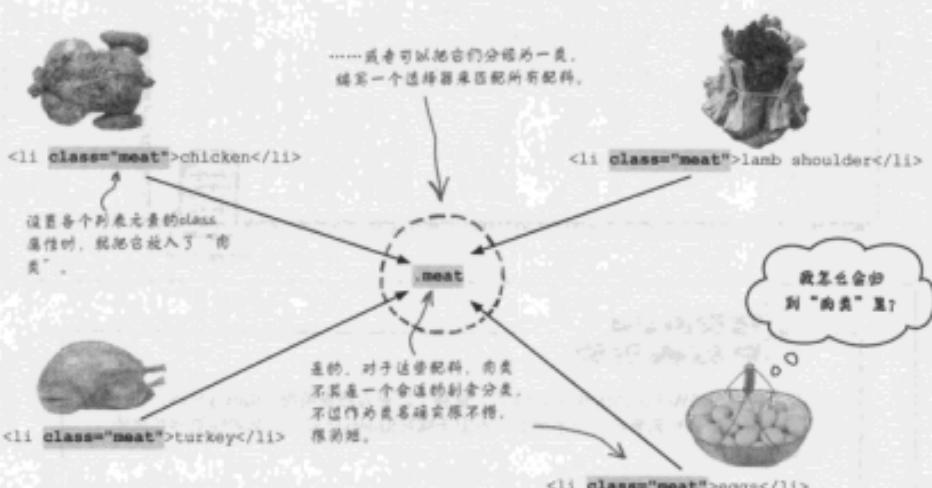
有一个规则的Web页面结构（HTML），这样可以更为容易地编写jQuery代码，不过我们想找的配料元素标记还不明确，不便于编写jQuery代码。怎么才能更容易地选择元素呢？

更多资源请访问稀酷客(www.ckook.com)

对元素分类

到目前为止，前面各章中我们已经看到，通过适当地建立HTML和CSS，可以帮助jQuery更有效地查找Web页面上的元素。为了建立合适的结构，应当为样式表增加类和ID，并对应适当的类和ID设置HTML元素属性。这样选择元素就会更容易，以后还能节省编写代码的时间。

在jQuery中，选择器并不只是用来控制页面的外观。利用选择器，jQuery还能匹配（或查询）页面上的元素。





找出大厨想要替换的配料，分别标上适当的类(鱼、肉或汉堡)。如果一个配料不需要类，可以保留空格。下面根据页面显示的菜单给出HTML。

```

<li>Thai-style Halibut
  <ul class="menu_list">
    <li>.....>coconut milk</li>
    <li>.....>pan-fried halibut</li>
    <li>.....>lemongrass broth</li>
    <li>.....>vegetables</li>
    <li>.....>Thai spices </li>
  </ul>
</li>

<li>Braised Delight
  <ul class="menu_list">
    <li>.....>lamb shoulder</li>
    <li>.....>cippolini onions</li>
    <li>.....>carrots</li>
    <li>.....>baby turnip</li>
    <li>.....>braising jus</li>
  </ul>
</li>

<li>House Grilled Panini
  <ul class="menu_list">
    <li>.....>prosciutto</li>
    <li>.....>provolone</li>
    <li>.....>avocado</li>
    <li>.....>cherry tomatoes</li>
    <li>.....>sourdough roll</li>
    <li>.....>shoestring fries </li>
  </ul>
</li>

<li>House Slider
  <ul class="menu_list">
    <li>.....>eggplant</li>
    <li>.....>zucchini</li>
    <li>.....>hamburger</li>
    <li>.....>balsamic vinegar</li>
    <li>.....>onions</li>
    <li>.....>carrots</li>
    <li>.....>multigrain roll</li>
    <li>.....>goat cheese</li>
  </ul>
</li>
```

```

<li>Frittata
  <ul class="menu_list">
    <li>.....>eggs</li>
    <li>.....>Asiago cheese</li>
    <li>.....>potatoes </li>
  </ul>
</li>

<li>Coconut Soup
  <ul class="menu_list">
    <li>.....>coconut milk</li>
    <li>.....>chicken</li>
    <li>.....>vegetable broth</li>
  </ul>
</li>

<li>Soup Du Jour
  <ul class="menu_list">
    <li>.....>grilled steak</li>
    <li>.....>mushrooms</li>
    <li>.....>vegetables</li>
    <li>.....>vegetable broth </li>
  </ul>
</li>

<li>Hot and Sour Soup
  <ul class="menu_list">
    <li>.....>roasted pork</li>
    <li>.....>carrots</li>
    <li>.....>Chinese mushrooms</li>
    <li>.....>chilli</li>
    <li>.....>vegetable broth </li>
  </ul>
</li>

<li>Avocado Rolls
  <ul class="menu_list">
    <li>.....>avocado</li>
    <li>.....>whole chiles</li>
    <li>.....>sweet red peppers</li>
    <li>.....>ginger sauce</li>
  </ul>
</li>
```



Exercise Solution

找出大厨想要替换的配料，分别标上适当的类(鱼、肉或汉堡)。如果一个配料不需要类，可以保留空格。

```

<li>Thai-style Halibut
    <ul class="menu_list">
        <li>coconut milk</li>
        <li class="fish">pan-fried halibut</li>
        <li>lemongrass broth</li>
        <li>vegetables</li>
        <li>Thai spices </li>
    </ul>
</li>

<li>Braised Delight
    <ul class="menu_list">
        <li class="meat">lamb shoulder</li>
        <li>cippolini onions</li>
        <li>carrots</li>
        <li>baby turnip</li>
        <li>braising jus</li>
    </ul>
</li>

<li>House Grilled Panini
    <ul class="menu_list">
        <li class="meat">prosciutto</li>
        <li>provolone</li>
        <li>avocado</li>
        <li>cherry tomatoes</li>
        <li>sourdough roll</li>
        <li>shoestring fries </li>
    </ul>
</li>

<li>House Slider
    <ul class="menu_list">
        <li>eggplant</li>
        <li>zucchini</li>
        <li class="hamburger">hamburger</li>
        <li>balsamic vinegar</li>
        <li>onion</li>
        <li>carrots</li>
        <li>multigrain roll</li>
        <li>goat cheese</li>
    </ul>
</li>
```

```

<li>Frittata
    <ul class="menu_list">
        <li class="meat">eggs</li>
        <li>Asiago cheese</li>
        <li>potatoes </li>
    </ul>
</li>

<li>Coconut Soup
    <ul class="menu_list">
        <li>coconut milk</li>
        <li class="meat">chicken</li>
        <li>vegetable broth</li>
    </ul>
</li>

<li>Soup Du Jour
    <ul class="menu_list">
        <li class="meat">grilled steak</li>
        <li>mushrooms</li>
        <li>vegetables</li>
        <li>vegetable broth </li>
    </ul>
</li>

<li>Hot and Sour Soup
    <ul class="menu_list">
        <li class="meat">roasted pork</li>
        <li>carrots</li>
        <li>Chinese mushrooms</li>
        <li>chili</li>
        <li>vegetable broth </li>
    </ul>
</li>

<li>Avocado Rolls
    <ul class="menu_list">
        <li>avocado</li>
        <li>whole chiles</li>
        <li>sweet red peppers</li>
        <li>ginger sauce</li>
    </ul>
</li>
```

更多资源请访问稀酷客(www.ckook.com)

增加按钮

内容基本都有了，下面再来看大厨的需求。接下来需要建立两个按钮。

—— 希望有一个“Go Vegetarian”按钮，能自动在我们的web页面菜单上替换合适的素食选项。

—— 还需要一个按钮把菜单恢复为原来的状态。

Sharpen your pencil

更新结构和脚本，具体建立这两个按钮。“Go Vegetarian”按钮的ID指定为vegOn，“Restore Menu”按钮的ID为restoreMe。

```
<div class="topper">
  <h2>Our Menu</h2>
  <ul>
    <li class="nav">.....</li>
    <li class="nav">.....</li>
  </ul>
</div>
```



index.html

```
$ (document).ready(function() {
  var v = false;

  if (v == false) {
    v = true
  } //end button

  if (v == true) {
    v = false;
  } //end button
}) //end document ready
```



my_scripts.js

Sharpen your pencil

Solution

更新结构和脚本，具体建立这两个按钮。“Go Vegetarian”按钮的ID指定为 vegOn，“Restore Menu”按钮的ID为restoreMe。

```

<div class="topper">
  <h2>Our Menu</h2>
  <ul>
    <li class="nav"> <button id="vegOn">Go Vegetarian</button> </li>
    <li class="nav"> <button id="restoreMe">Restore Menu</button> </li>
  </ul>
</div>

```

建立ID为vegOn和restoreMe的按钮元素。

index.html

更具体的逻辑器，这里使用了更多类型的ID。

```

$(document).ready(function() {
  var v = f;
  $("button#vegOn").click(function(){
    if (v == false){
      v = true;
    } //end button
  });
  $("button#restoreMe").click(function(){
    if (v == true){
      v = false;
    } //end button
  });
}); //end document ready

```

为按钮关联click方法。

my_scripts.js

接下来呢？

真是神速！你已经建立了两个按钮。下面把这几项从需求列表中删除，再来讨论“Go Vegetarian”按钮需要完成的工作。

——希望有一个“go vegetarian”按钮，能启动在我们的web页面菜单上替换合适的素食选项。

——还需要一个按钮把菜单恢复为原来的状态。

要做如下替换：

——鱼肉套餐没有替代套餐，所以要把这些套餐去掉。

——我们为素食顾客提供了大磨菇来替代汉堡。

——除了汉堡之外，我们为素食顾客提供了豆腐，作为所有其他肉类和蛋类的替代品。



用你自己的语言，写出“Go Vegetarian”按钮需要做的3件事情。

1.

2.

3.



用你自己的语言，写出“Go Vegetarian”按钮需要做的3件事情。

1. 匹配fish类的li元素，并从菜单中删除这些套餐。
2. 匹配hamburger类的li元素，把它们替换为大薯条。
3. 匹配meat类的li元素，把它们替换为豆腐。

如果你的答案稍有不同也不用担心。无论如何正确地转换为Web应用应完成的工作，这还有很多做练习。

下一个任务是解决上面的第一项任务：匹配fish类的li元素，并从菜单中将其删除。我们可以用类选择器来匹配元素，并用第2章介绍过的remove将元素从DOM树中删除。

jQuery还提供了一个detach方法。detach和remove都可以从DOM删除元素，那么这两个方法有什么区别呢？我们该用哪一个方法？

remove



remove方法从DOM完全删除元素。

```
$( "img#thumbnail" ).remove();
```

detach



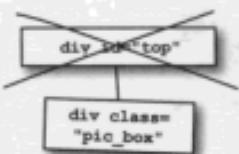
detach方法将选中的元素取出，不过还会维护这些元素，以后还可以重新关联。

```
$( "img#thumbnail" ).detach();
```

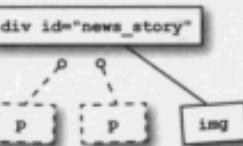
Sharpen your pencil

在下面的空白处，写出选择器以及remove或detach代码来创建左边所示的结果。

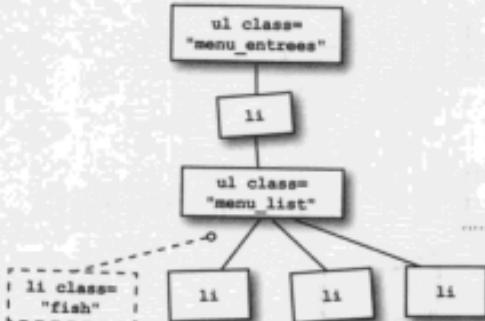
DOM结果



jquery语句



DOM结果



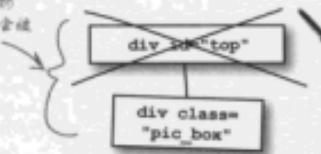
Sharpen your pencil

Solution

在下面的空白处，写出选择器以及remove或detach代码来创建左边所示的结果。

有一个元素上述
行remove或detach
时，这个元素的
所有子元素也会被
删除。

DOM结果



jQuery语句

`$("div#top").remove()`

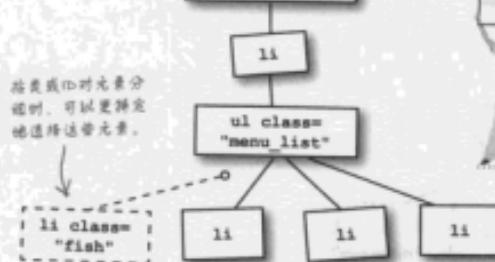
请记住，当
选择器匹配
的所有元素
都会被删除。

div id="news_story"


`$("div p").detach()`

若类或ID对元素分
组时，可以更确定
地选择这些元素。

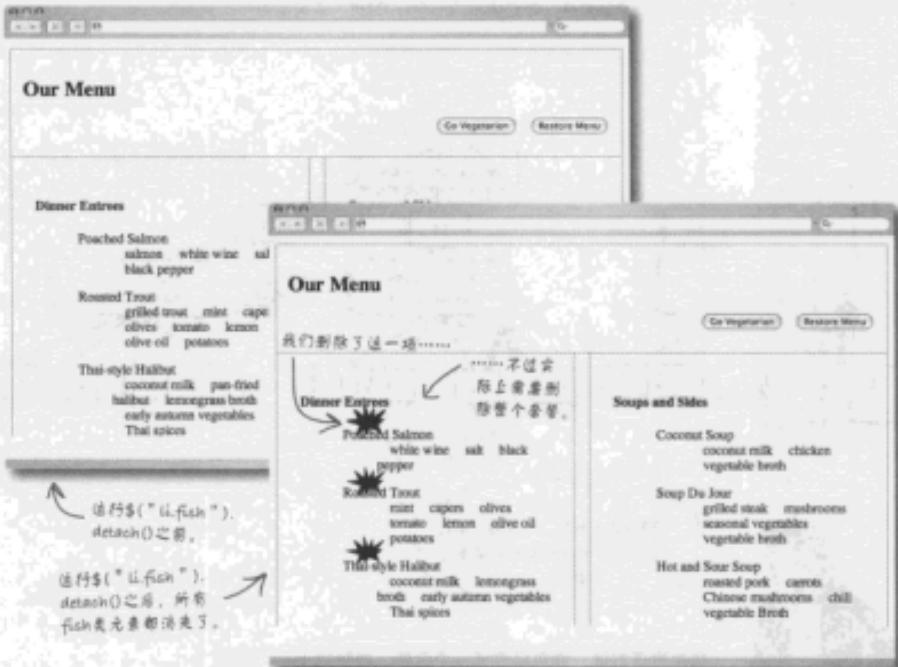
ul class="menu_entrees"


`$("li.fish").detach()`



试一试

在my_scripts.js文件的vegOn按钮click函数中增加第3个练习答案的代码行。
然后在你喜欢的浏览器中打开页面，确保一切正常。



Our Menu

[Go Vegetarian](#)

[Restore Menu](#)

Dinner Entrees

Poached Salmon	white wine	salt
black pepper		
Roasted Trout	grilled trout	mint
capers	olives	lemon
olive oil	potatoes	
Thai-style Halibut	coconut milk	pan-fried
halibut	lemongrass broth	
early autumn vegetables		
Thai spices		

Our Menu

[Go Vegetarian](#)

[Restore Menu](#)

我们删除了这一项……

Dinner Entrees

Poached Salmon
white wine salt black
pepper

Roasted Trout

mint capers olives
tomato lemon olive oil
potatoes

Thai-style Halibut

coconut milk lemongrass
broth early autumn vegetables
Thai spices

Soups and Sides

Coconut Soup
coconut milk chicken
vegetable broth

Soup Du Jour

grilled steak mushrooms
seasonal vegetables
vegetable broth

Hot and Sour Soup

roasted pork carrots
Chinese mushrooms chili
vegetable broth

`detach`方法确实删除了一些内容，不过并不是我们想删除的全部内容。这里删除了套餐的列表元素，但我们真正想删除的是嵌套这个`.fish`列表元素的整个套餐。

怎么告诉DOM删除整个套餐呢？

更多资源请访问稀酷客(www.ckook.com)

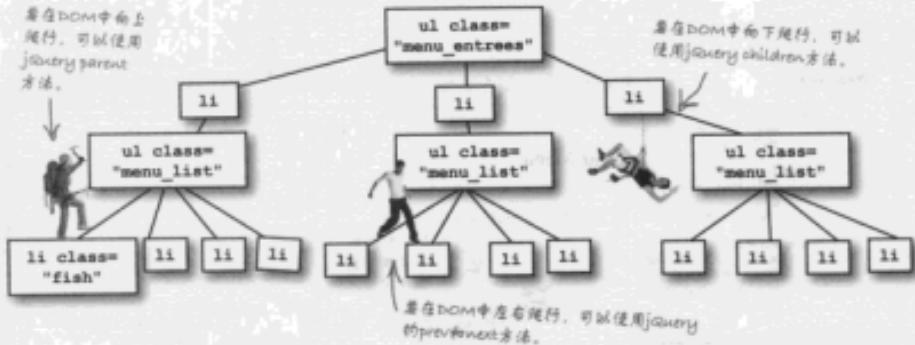
在DOM树上荡来荡去

第1章中，我们已经了解到DOM会构建为一棵树。它有一个根，有分支，还有节点。浏览器的JavaScript解释器可以遍历（并管理）这个DOM，jQuery在这方面尤其擅长。DOM遍历就是在DOM中上下爬行。

从第1章开始我们一直在管理DOM。刚才介绍的`detach`方法就是DOM管理的一个很好的例子（也就是说，可以动态地从DOM取出元素）。

不过遍历到底是什么？下面取一部分菜单，把它表示为一个DOM树，来看看遍历如何工作。

系好你的安全带！DOM遍历就像在DOM中向上、向下、向两边爬行。



“DOM树上爬行。”没错。不过这些遍历对我们删除套餐有什么帮助？

利用遍历方法，你可以选择一个元素，得到它上面、下面或两边的其他元素。

下面会更详细地介绍如何获取这些元素。

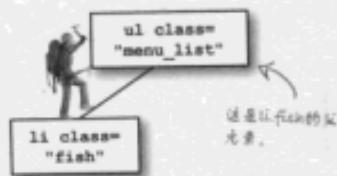
用遍历方法在DOM树上爬行

要告诉DOM我们想删除菜单列表中包含鱼的套餐，必须根据元素的关系引用这些元素。利用jQuery的遍历方法就可以得到这些元素关系。

选择所有fish类的元素。

然后得到这些元素上面的元素。

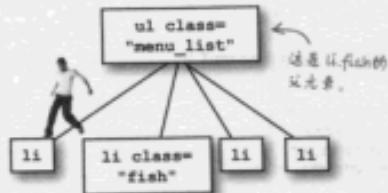
`$(".fish").parent()`



选择所有fish类的元素。

然后得到古董过时你的兄弟元素。

`$(".fish").prev()`



选择所有 menu_list 类的元素。

然后得到这些元素下面的元素。

`$(".menu_list").children()`



选择所有fish类的元素。

然后得到右达宝你的兄弟元素。

`$(".fish").next()`



这些方法中哪一个可以帮助我们删除菜单上包含fish类元素的套餐？

更多资源请访问稀酷客(www.ckook.com)

方法串链，爬得更远

如果我们想爬得更高、更低，或者更深，怎么做呢？抓紧绳子！jQuery为我们提供了方法串链。利用方法串链，我们可以用一种更有效的方式管理和遍历页面。具体如下：

要查找更高一级的父元素，只需
在方法链再加一个方法。

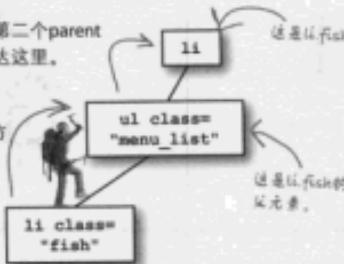
```
$(".fish").parent().parent()
```

- ② 方法链中的第二个parent方法可以到达这里。

这是从fish的父元素的父元素。

- ① 方法链中第一个parent方法可以到达这里。

这是从fish的父元素。



方法还可以连着和匹配。

```
$(".menu_list").parent().next().remove()
```

- ② next方法爬到右边的兄弟元素。

remove方法将遍历到的
元素从DOM中取出。

- ① parent方法向上爬
到包含所选元素的
父元素。





访问<http://www.thinkjquery.com/chapter04/traversal/>，在你喜欢的浏览器开发工具中打开JavaScript控制台。本书最前面的“说明”中介绍过浏览器开发工具。分别运行以上4个遍历方法并串联`detach`方法。如下所示。你认为这对解决我们当前的问题有帮助吗？请写出为什么有帮助或者为什么没有帮助。

重要说明：运行各个语句之后一定要刷新浏览器。

```
$(".menu_entrees").children().detach()
```

```
.....
```

```
$(".menu_list").children().detach()
```

```
.....
```

```
$(".fish").parent().detach()
```

```
.....
```

```
$(".fish").parent().parent().detach()
```

```
.....
```



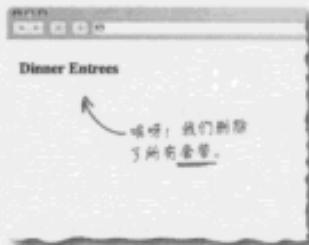
Exercise Solution

访问<http://www.thinkjquery.com/chapter04/traversal/>，在你喜欢的浏览器开发工具中打开JavaScript控制台。本书最前面的“说明”中介绍过浏览器开发工具。分别运行以上4个遍历方法并串链`detach()`方法，如下所示。你认为这对解决我们当前的问题有帮助吗？请写出有帮助或没有帮助的原因。

重要说明：运行各个语句之后一定要刷新浏览器。

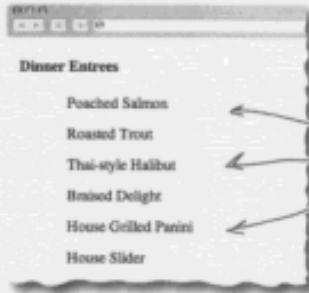
```
$(".menu_entrees").children().detach()
```

这个遍历方法会删除`.menu_entrees`的子元素。它不能用来删除包含肉的套餐，因为这会删除所有套餐列表。真糟糕！这可不是我们想要的。



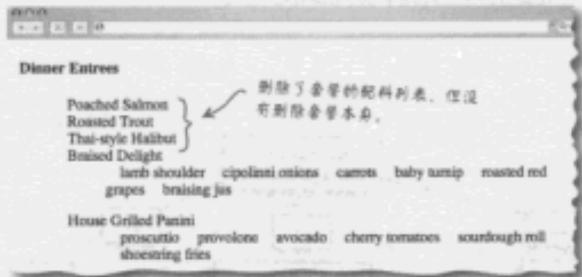
```
$(".menu_list").children().detach()
```

这个遍历方法会删除`.menu_list`的子元素。它不能用来删除包含鱼肉的套餐，因为这会从各个`ul.menu_list`删除配料列表。哦呀！这绝对不是我们想要的。



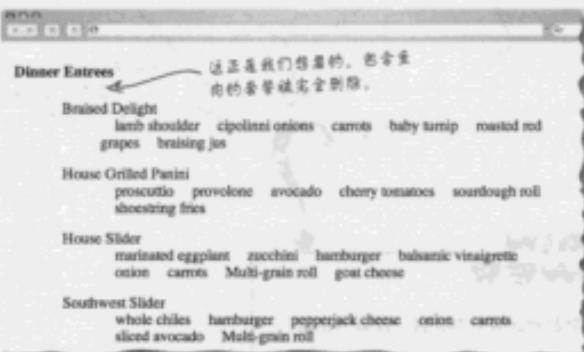
```
$(".fish").parent().detach()
```

这个遍历方法会删除.fish的父元素。它不能用来自删除包含鱼肉的套餐，因为这个方法在DOM树中上行得还不够远。它真正删除的只是ul.menu_list (以及这个元素以下的所有元素)。



```
$(".fish").parent().parent().detach()
```

这个遍历方法会删除.fish的父元素的父元素。这正是我们所做的。



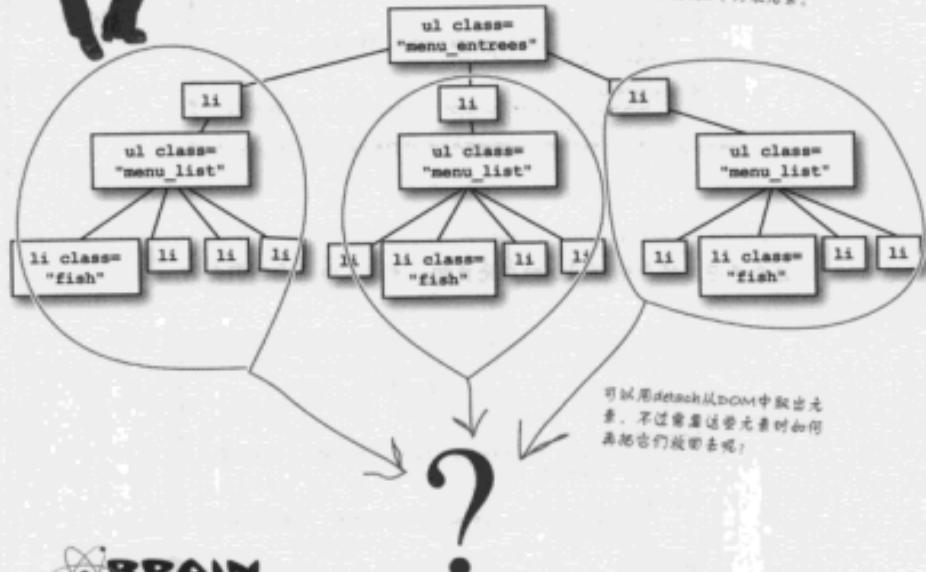


客一下，我们编写“Restore Menu”
按钮的代码时还得恢复这些包含鱼肉
的套餐吧？

没错。不能将这些鱼肉套餐删除后就把它们彻底忘掉。

还要重新考虑我们的代码来恢复这些套餐。

`$(".fish").parent().parent().detach()`会
删除包含鱼肉的3个列表元素。



以后我们还需要放回这些鱼肉套餐。该如何处理？

Sharpen your pencil

到目前为止，我们已经见过不少jQuery和JavaScript构造。需要哪些构造才能做到不至于忘掉这些fish类元素呢？对应各个构造，请在“该用它吗”一列中写上“是”或“否”，并解释你选择或不选择这种构造的原因。我们已经为你完成了一个，你只需要完成其余的3个。

	该用它吗？	为什么？
终止符	否	终止符只是用来结束一个语句。它不能解决记住所删除元素的问题。
变量		
函数		
选择器		



Sharpen your pencil

Solution

到目前为止，我们已经看过不少jQuery和JavaScript构造。需要哪些构造才能做到不至于忘掉这些.js类元素呢？对应各个构造，请在“该用它吗”一列中写上“是”或“否”，并解释你选择或不选择这种构造的原因。我们已经为你完成了一个，你只需要完成其余的3个。以下是我们的答案。

	该用它吗	为什么？
终止符	否	终止符只是用来结束一个语句。它不能解决记住所删除元素的问题。
变量	是	变量可以为我们存储内容。如果保存这些被删除的元素，则以后只要引用变量就可以再把它们找回来。
函数	否	利用函数可以对数据完成处理。对于这些被删除的元素，现在的问题是我要存储数据，而不是处理数据。
选择器	否	选择器会根据DOM中的内容选择元素。我们已经选择了元素。现在只是需要有一种办法来恢复这些元素。

there are no
Dumb Questions

问：我已经了解remove和detach，不过如果我只是想删除一个元素中的某个内容，而不是元素本身。该怎么做到呢？

答：要删除一个元素中的内容，可以使用empty方法。假设你想删除页面上段落中的全部内容，可以这样做：
`$("p").empty();`

问：有没有办法遍历一个元素的所有父元素？

答：有。除了parent方法，jQuery还提供了parents方法。利用这个方法可以遍历所选元素的所有父元素。本章后面还会看到这个方法的使用。

问：如果我想得到离所选元素最近的父元素，该怎么做呢？

答：可以使用closest方法。类似于parents方法，closest方法会上行查找一个元素的父元素，不过它找到一个匹配时就会停止。例如，如果想找到一个列表项上面最近的ul，就可以使用
`$("li").closest("ul")`。

问：我明白next和previous，不过如果我想遍历DOM树同一层上的所有元素该怎么做呢？

答：很幸运，jQuery团队也想到这一点了。siblings方法就可以遍历所选元

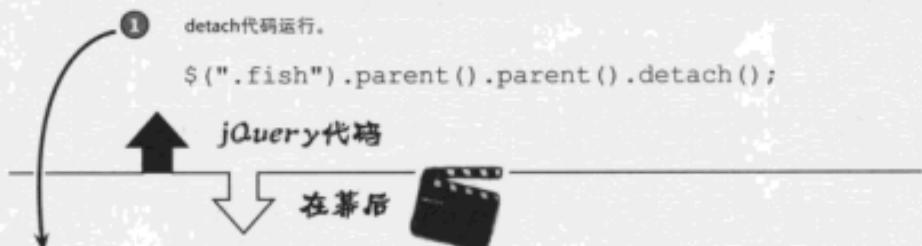
素同一层上的所有元素。

问：Google Chrome内置了jQuery吗？

答：没有。之所以可以在Chrome的浏览器开发工具中运行jQuery，这是因为我们在HTML页面中包含了jQuery。如果访问一个没有使用jQuery的web页面，可不要指望Chrome JavaScript控制台会运行jQuery语句。

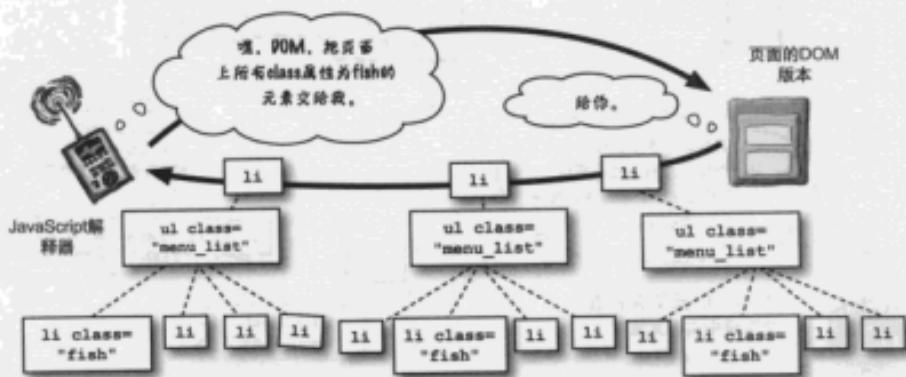
变量还可以存储元素

变量肯定很有用，因为我们发现又要用到它们了。前三章已经用到了变量，不过只是用它们来存储数字和文本串。如果JavaScript变量还能存储元素不是很方便吗？实际上，它们确实可以存储元素。



- ② 利用jQuery库，JavaScript解释器向DOM请求所选的元素。

- ③ DOM获取所选的元素，并把它们返回给解释器。



浏览器会在内存中临时保存这些元素。如果我们希望以后在程序中使用这些元素，则最好把它们放在一个变量中。不过怎么做到呢？

更多资源请访问稀酷客(www.ckook.com)

美元符又出现了……

存储元素很简单。可以创建一个变量，就像存储数字和文本串一样，把这个变量（用等号）设置为一个返回元素的语句。不过，如果能知道变量存储了特殊的内容（比如存储的是元素，而不仅是数字或文本串），那就多好。jQuery编码人员通常采用这种做法：如果一个变量将要用来存储jQuery返回的元素，就在这个变量前面加一个美元符。这样一来，任何查看代码的人都能知道这个变量是用来存储从jQuery得到的内容。

```
$f = $(".fish").parent().parent().detach();
```

在变量前添加一个美元符，表示它是
存储由jQuery返回的元素。

\$f

之前我们使用的变量只存
储一个值。把所有这些不同
的元素都填在一个容器中看起
来太乱了。

把不同的元素存储在一
个变量中实在太乱了。

因为这个原因，jQuery使
用JavaScript数组来存储
元素。下面来看数组是什
么。



更多资源请访问稀酷客(www.ckook.com)

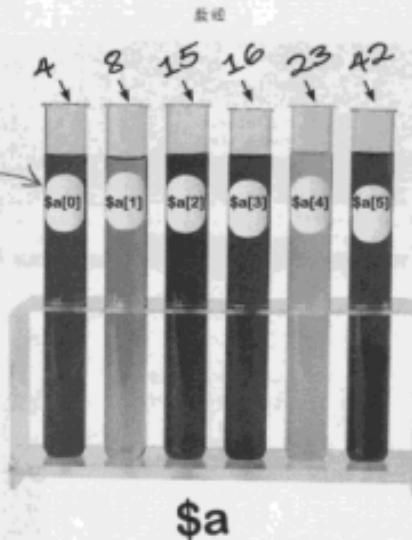
数组扩展存储选项

从DOM选择元素并存储在一个变量中时，jQuery会把这些数据作为一个数组返回。数组就是一个有更多存储选项的变量。

一个基本变量存储一个值。



一个数组存储多个值。



可以在各个存储槽中放入和取出内容。要把值“15”放在第3个槽中，要写为：

\$a [2] = 15;

↑
第3个槽的编号为2，因为
我们总是从0开始编号。



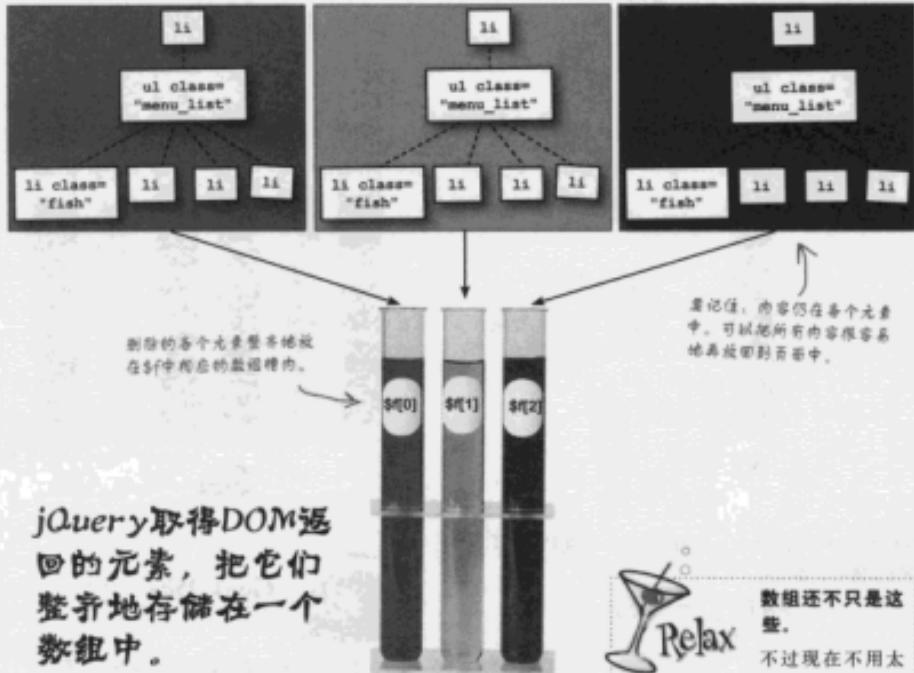
Geek Bits

数组并不是一定要以美元符(\$)开头。用一个美元符指示jQuery数组的做法只是jQuery开发人员的一种编码习惯。

在数组中存储元素

选择li元素并用detach删除，并将一个变量(\$f)设置为detach的结果，此时jQuery会得到DOM返回的元素，把它们整齐地存储在一个JavaScript数组中。这样一来，我们想用restore按钮放回这些元素时就不会那么混乱了。

```
$f = $(".fish").parent().parent().detach();
```



数组还不只是这些。

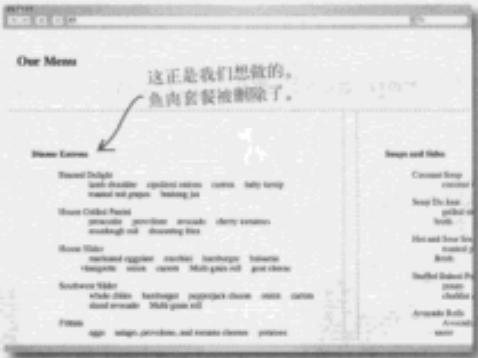
不过现在不用太担心。第6章就会

介绍数组的更多详细信息。



试一试

在my_scripts.js文件的vegOn按钮click函数中增加上一页的代码行。这会删除#fish元素的父元素。然后，在你喜欢的浏览器中打开这个页面，确保一切正常。



做得好。现在来更新任务列表。

- 1. 匹配fish类的li元素，并从菜单中删除这些套餐。
- 2. 匹配hamburger类的li元素，把它们替换为大蘑菇。
- 3. 匹配meat类的li元素，把它们替换为豆腐。

接下来，需要查找包含汉堡的套餐，把汉堡换成大蘑菇。



我们已经见过如何从DOM取出元素，不过怎么动态地将DOM内容替换成不同的内容呢？

更多资源请访问稀酷客(www.ckook.com)

用replaceWith替换元素

利用replaceWith方法可以把选择的元素替换为新元素。想要修改DOM把一个内容换成另一个内容时，就可以使用这个很方便的jQuery方法。假设我们想动态地把2级标题“Our Menu”换成1级标题“My Menu”，用replaceWith方法就能办到：

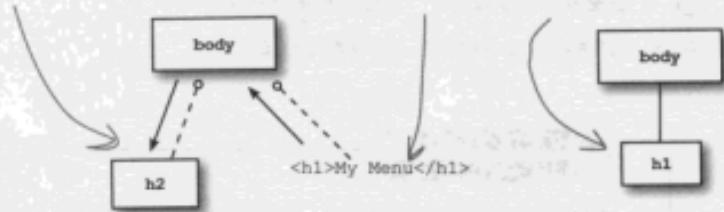
选择所有h2元素。
将选择的元素替换为……
……括号里的内容。

```
$("h2").replaceWith("<h1>My Menu</h1>");
```



- ① JS解释器匹配2级标题元素……

- ② ……把它替换为括号里的内容，用一个新元素和新内容更新DOM。



replaceWith对我们有什么帮助？

现在要匹配hamburger类的li元素，把它们替换为portobello类的li元素。写代码之前先来考虑这个问题。



编写代码，找到hamburger类的li元素，把它们替换为portobello类的元素。下面的图会对你有帮助。我们为你写出了部分答案，剩下的由你来完成。

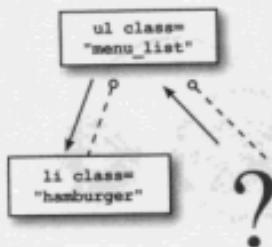


更多资源请访问稀酷客(www.ckook.com)



Exercise Solution

编写代码，找到hamburger类的li元素，把它们替换为portobello类的元素。以下是我们答案。



replaceWith方法动态地将选择的内容替换成插槽中的元素。要记住，可以在插槽里嵌入HTML。

这样hamburger类的所有元素。

```
$(".hamburger").replaceWith($("<li class='portobello'><em>Portobello Mushroom</em></li>");
```



试一试

把replaceWith代码增加到my_scripts.js文件的vegOn按钮click函数中。然后，在你喜欢的浏览器中打开这个页面，单击“Go Vegetarian”按钮，确保一切正常。

House Slider	Southwest Slider
marinated eggplant balsamic vinaigrette cheese	whole chiles onion
zucchini onion carrots	sliced avocado
Portobello Mushroom Multi-grain roll goat cheese	Portobello Mushroom Multi-grain roll

使用replaceWith之前先想想

任务列表中下一步是什么？

- 1. 匹配fish类的li元素，并从菜单中删除这些套餐。
- 2. 匹配hamburger类的li元素，把它们替换为大蘑菇。
- 3. 匹配meat类的li元素，把它们替换为豆腐。

需要找到meat类的套餐，把它们换成豆腐。



jQuery的replaceWith方法很简单，功能也很强大，不过很遗憾，它对我们解决这个问题毫无帮助。为什么不能用呢？

replaceWith并不适用每一种情况

如果是一对一替换，比如把hamburger类替换为portobello类，那么replaceWith方法就很合适。



一对多替换

不过任务列表中下一项任务要完成的元素替换并不是一对一的替换。我们必须把多种不同配料（火鸡、蛋、牛排、羊排）换成一种配料（豆腐）。

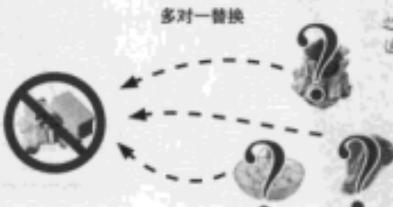


多对一替换

不过，以后选择豆腐并替换时，就会存在一个问题。我们想放回不同的肉类，但DOM已经忘记了原来是什么。

可以把豆腐都换成某一种肉类，不过这不是我们想要的。

所以需要分两步完成这个菜单替换工作：



- ① 将tofu类的li元素插入到DOM中meat类元素之后。

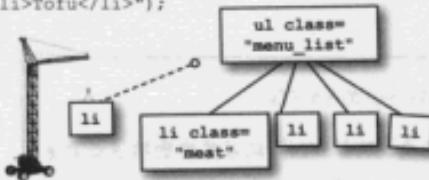
- ② 删除meat类的元素，把它们分别保存到一个变量中。

在DOM中插入HTML内容

到目前为止，我们已经删除或替换了DOM中的元素。很幸运，jQuery库的创建者为我们提供了很多途径可以在DOM中插入内容。现在我们要研究的是before和after。

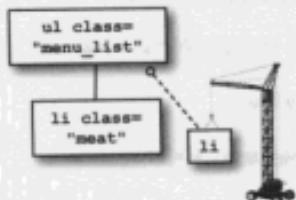
before将新内容插入到选择的元素前面。

```
$(".meat").before("<li>Tofu</li>");
```



after将新内容插入到选择的元素后面。

```
$(".meat").after("<li>Tofu</li>");
```



Sharpen your pencil

编写jQuery代码，完成各个步骤。

- ① 将tofu类的li元素插入到DOM中meat类元素之后。

- ② 删除meat类的元素。把它们分别保存到一个变量中。

Sharpen your pencil

Solution

编写jQuery代码，完成各个步骤。

- ① 将tofu类的li元素插入到DOM中
meat类元素之后。

```
$( ".meat" ).after( "<li class='tofu'><em>Tofu</em></li>" );
```

- ② 删除meat类的元素，把它们分别保存到一个变量中。

```
$m = $( ".meat" ).detach();
```

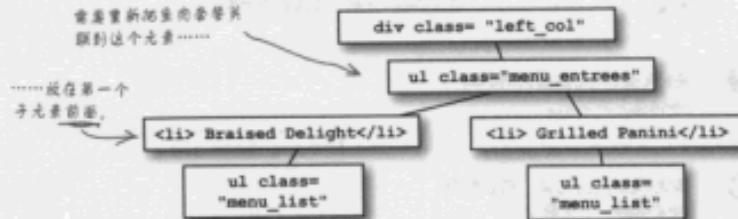
完成“Go Vegetarian”按钮对应的各个步骤：

- 1. 匹配fish类的li元素，并从菜单中删除这些套餐。
- 2. 匹配hamburger类的li元素，把它们替换为大蘑菇。
- 3. 匹配meat类的li元素，把它们替换为豆腐。

接下来，需要建立“Restore Menu”按钮。以下是这个按钮要做的工作。

- 把鱼肉套餐放到菜单中原来的位置（也就是左列中第一个菜单项前面）。
- 找到包含大蘑菇的套餐，把大蘑菇换成汉堡。
- 找到包含豆腐的套餐，把它们（按正确的顺序）换成不同的肉类。

下面来深入分析，看看对于第一个任务需要做什么。



我们已经知道如何使用**before**，不过如何指定第一个子元素呢？

更多资源请访问稀酷客(www.ckook.com)

使用过滤器方法缩小选择范围（第1部分）

幸运的是，jQuery提供了一些过滤方法。可以利用这些方法缩小选择范围，解决查找第一个子元素之类的问题。下面来看其中6个过滤器方法（这一页介绍3个方法，下一页介绍另外3个方法）。

first

除了所选元素中第一个元素以外，first方法会过滤掉所有其他元素。

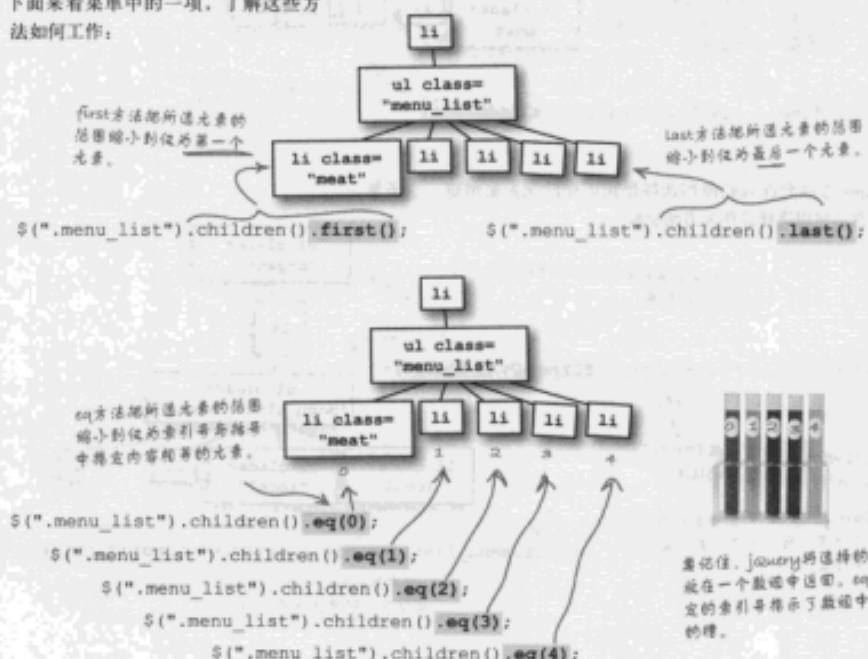
eq

除了所选元素中索引号等于括号中内容的元素之外，eq方法会过滤掉所有其他元素。

last

除了所选元素中最后一个元素以外，last方法会过滤掉所有其他元素。

下面来看菜单中的一项，了解这些方法如何工作：



更多资源请访问稀酷客(www.ckook.com)

你现在的位置：

使用过滤器方法缩小选择范围 (第2部分)

现在来了解slice、filter和not方法，看看它们如何工作。

slice

除了索引介于括号中指定索引号之间
的元素之外，slice方法会过滤掉所有
其他元素。

filter

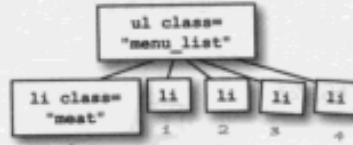
除了与括号中指定选择器匹配
的元素之外，filter方法会过滤掉
所有其他元素。

not

not方法会过滤掉与括号中指定选
择器匹配的所有元素。

slice方法将所选元素的范围缩小到索引介于括号中指定的两个索引之间的那些元素。

```
$(".menu_list").children().slice(1,3);
```



在这里，只含有第一个元
素——第二个元素。

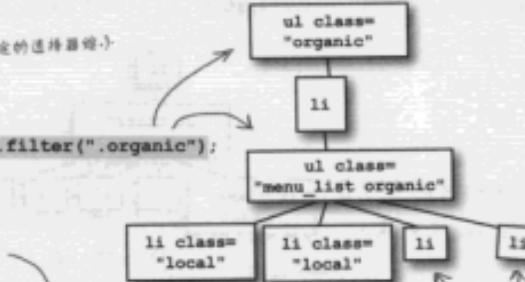
filter和not方法允许我们使用选择器由匹配的元素集创建一个子集。
这两个方法使用选择器作为方法参数。

filter方法指据集中指定的选择器-
所选元素的范围。

```
$(".menu_list").parents().filter(".organic");
```

filter和not方法可嵌套在parents
和children方法很好地使用。

利用parents方法，可以得到作为所选
元素父元素、祖父元素、曾祖父元素
等所有元素。



```
$("ul.menu_list.organic").children().not(".local");
```

not方法将所选元素的范围缩小到与括号中指定
选择器不匹配的所有元素。

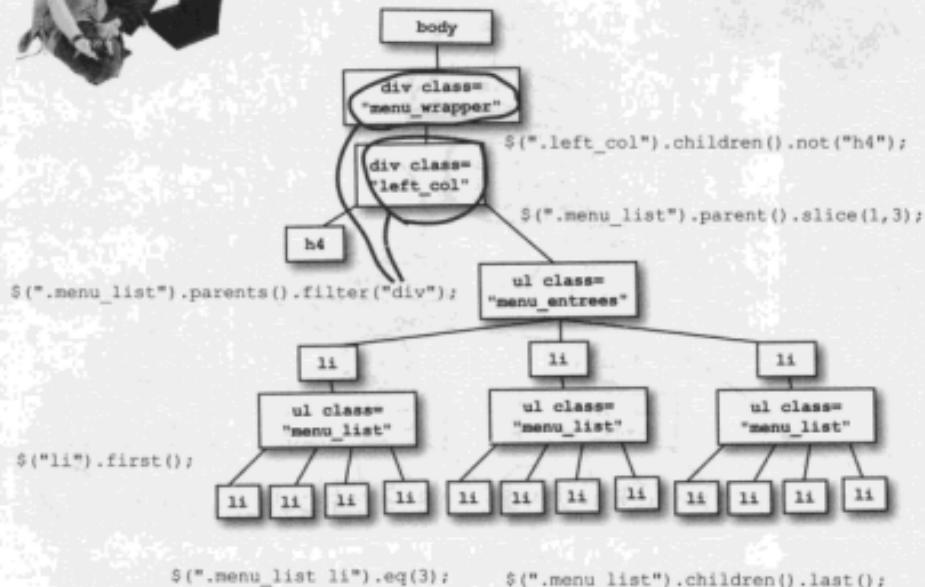
这些方法中，哪一个可以帮助你指定菜单中的第一个子元素？

更多资源请访问稀酷客(www.ckook.com)

扮演DOM

你的任务是扮演DOM。在jQuery语句与选择器将返回的DOM元素之间连线。假设页面上只有这些元素。

我们已经为你完成了第一个。



Sharpen your pencil

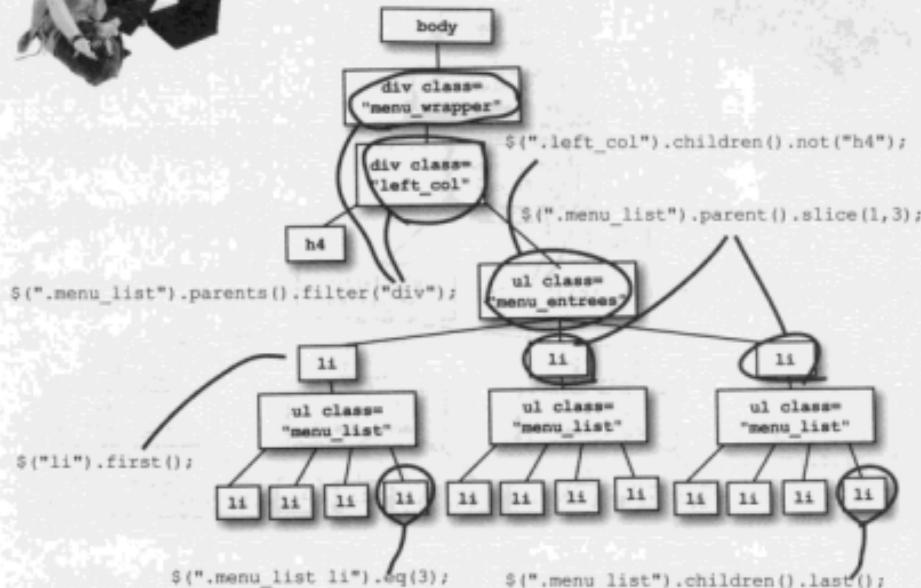
写出jQuery代码行，将鱼肉套餐放回到菜单中原先的位置（也就是menu_entrees下的第一个菜单项）。

.before(\$f);



扮演DOM答案

你的任务是扮演DOM。在jQuery语句与选择器将返回的DOM元素之间连线。假设页面上只有这些元素。我们已经为你完成了第一个。



Sharpen your pencil

Solution

写出jQuery代码行，将鱼肉套餐放回到菜单中原先的位置（也就是menu_entrees下的第一个菜单项）。

```
$“.menu_entrees li”).first().before($f);
```

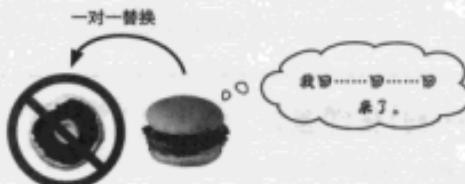
更多资源请访问稀酷客(www.ckook.com)

放回汉堡

到目前为止，我们已经完成了“Restore Menu”按钮需求中的一项任务，还有两项工作需要完成：

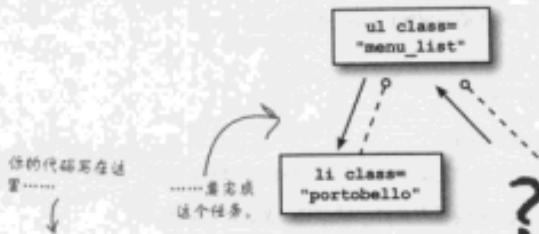
- 把鱼肉套餐放回到菜单中原来的位置（也就是左列中第一个菜单项前面）。
- 找到包含大蘑菇的套餐，把大蘑菇换成汉堡。
- 找到包含豆腐的套餐，把它们（按正确的顺序）换成不同的肉类。

任务列表中下一项任务有些似曾相识，对不对？我们要做的实际上就是把原先的替换反过来。为什么？因为这里处理的是一种一对一替换，我们真喜欢一对一替换，因为它们逻辑上很简单。



Exercise

还记得这个练习吗？现在把它倒过来。编写代码，查找portobello类的li元素，把它们替换为hamburger类的li元素。下面的图会对你有帮助。我们已经为你写出了部分答案，剩下的由你来完成。

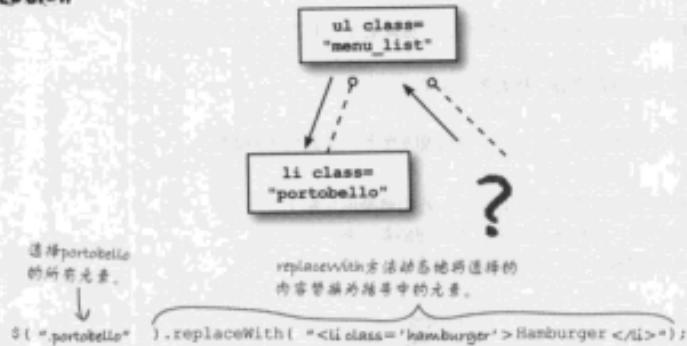


```
$('.....').replaceWith('.....Hamburger.....');
```

更多资源请访问稀酷客(www.ckook.com)

Exercise
SOLUTION

replaceWith只用一步就能把它很快换回来。干得漂亮！



牛肉（噢……肉）在哪里？

现在要完成“Restore Menu”按钮的第三项任务了：

- 把鱼肉套餐放到菜单中原来的位置（也就是左列中第一个菜单项前面）。
- 找到包含大蘑菇的套餐，把大蘑菇换成汉堡。
- 找到包含豆腐的套餐，把它们（按正确的顺序）换成不同的肉类。

我们对这些li.meat元素又做了什么？下面来回顾一下：

我们把li.tofu元素放在DOM中meat元素后面。

```
$('.meat').after('<li class="tofu"><em>Tofu</em></li>');
```

然后，删除li.meat元素，不过通过将它们存储在\$m中，我们还在维护这些元素。

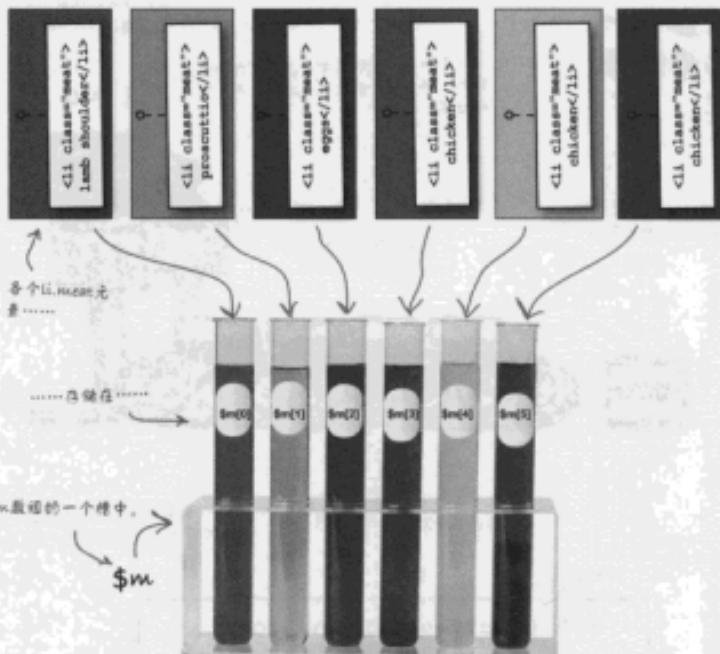
```
$m = $('.meat').detach();
```

那么这些元素在哪里？另外怎么把它们再找回来呢？

更多资源请访问稀酷客(www.ckook.com)

肉类数组

应该记得，存储jQuery元素时，都会为变量增加一个美元符来表示我们使用的变量在存储一种特殊的内容。在这里，变量存储的是一个jQuery数组，\$m中的元素如下存储：



对于每个li.tofu元素，需要分别放入一个li.meat元素。你已经见过很多方法可以在DOM中放入元素。这里要使用哪个方法呢？

each方法循环处理数组

第3章中，你已经见过如何使用each方法循环处理元素。这里仍然可以使用这个方法循环处理\$**m**数组中的所有meat元素，把它们放回原来的位置。不过，为此我们还需要了解each方法是如何工作的。

each方法就像一个
包装元素的流水线机床。

索引（或迭代器）会跟踪函数正在处理的那个元素。
 $i = 0$

each方法为jQuery脚本赋予了强大功能，它允许一次处理数组中的一个元素。

each中加入函数时才会真正体现它的强大能力。利用这个函数可以对各个元素做些处理。



选择这些元素时，jQuery 把我们选择的元素存储在一个数组中。

each方法逐个地处理数组中的元素，并对各个元素做一些工作。

变量i从0开始，处理各个元素时依次计数。

利用\$(this)可以告诉函数它目前正在处理哪个元素。

```
$(".tofu").each(function(i) {  
    $(this).after(" ");  
});
```

这里使用了after方法，不过你也可以使用任何jQuery方法处理元素数组。

我们希望对应各个li.tofu元素放入meat元素。这里应该放什么？

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴

按正确的顺序放入这些代码磁贴来完成restoreMenu按钮。让它能正确工作。
我们已经为你放入了一些磁贴。

\$("button#restoreMe").click(function () {

 if (v == true) {

 v = false;

 }

};



my_scripts.js

<li class="hamburger">Hamburger";

\$m[i]);

\$(".portobello").replaceWith(

.before(\$f);

});

\$(".tofu").remove();

\$(".menu_entrees li").first()

\$(this).after(

\$(".tofu").each(function(i) {

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴答案

按正确的顺序放入这些代码磁贴，来完成restoreMenu按钮，让它能正确工作。我们已经为你放入了一些磁贴。

```

$("button#restoreMe").click(function() {
    if (v == true) {
        $(".portobello").replaceWith("<li class='hamburger'>Hamburger</li>");
        $(".menu_entrees li").first().before($f);
        $(".tofu").each( function(i) {
            $this).after( $m[i]);
        });
        $(".tofu").remove();
        v = false;
    }
});
```

通过利用\$.m数组和索引（对应函数正在处理的tofu元素），嵌入lement元素。



my_scripts.js

这样……对吗？

你已经完成了“Restore Menu”按钮的全部工作。下面来更新我们的文件，完成这个项目。

- 把鱼肉套餐放到菜单中原来的位置（也就是左列中第一个菜单项前面）。
- 找到包含大蘑菇的套餐，把大蘑菇换成汉堡。
- 找到包含豆腐的套餐，把它们（按正确的顺序）换成不同的肉类。



等一下，我们忘记了餐巾纸上“以及”中提到的要求。

以及：如果可以，我们还希望在替换的素食套餐旁边显示一个叶子图标。

嗯，你说得对！

好在Web设计人员已经在my_style.css文件中加入了veg_leaf类。下面来看一下。

```
.veg_leaf{
    list-style-image:url('../images/leaf.png');
}
```



my_style.css



写出语句，将veg_leaf类增加到tofu类父元素的父元素。

“等等，这里可能需要将语句放在哪里？”



只需要一点DOM处理，再加上一点addClass魔法，就大功告成了！

```
$( ".tofu" ).parent().parent().addClass( "veg_leaf" );
```

there are no Dumb Questions

问：其他过滤方法我都明白了，不过对slice还有些搞不懂。能不能再深入地解释一下？

答：slice方法可能会让人有些糊涂，slice最让人迷惑的是它的参数：slice(start, end)。

第一个参数是起始参数，这个参数是必要的，否则slice无法正常工作。start参数指出在数组元素中从哪里开始选择。应该记得，数组中的第一个元素索引为0，这说明必须从0开始计数。start参数还可以取负数。如果放入一个负数，slice会从数组末尾反向开始计数，而不是从数组开头正向计数。

问：OK，那么slice方法的end参数做什么呢？

答：slice方法的第二个参数end不是必要的。如果没有包含这个参数，则slice会从start参数设置的起始位置开始，选择数组中索引大于start参数的所有元素。一定要提醒自己：数组是从0开始计数的，end参数可能与你的直觉印象不同。

问：each方法看起来功能相当强大，each怎么知道在处理哪个元素？

答：真正的威力来自于each与this关键字的结合。each方法会自动跟踪索引，“知道”在处理哪个元素。只有选择了多个元素时才应使用each。要引用当前元素，可以使用this。不过需要用jQuery快捷方式把它包起来：\$(this)。

问：为什么要把“i”或“index”放在each函数里？

答：each函数使用索引变量（通常名为“i”或“index”）来跟踪each正在处理的元素。这样each函数就能知道它何时完成处理。如果each没有索引变量，就无法知道要处理哪个元素，也无法停止。

问：如何找到一个jQuery数组中的元素？

答：可以使用find方法找到jQuery数组中的元素。假设有一个li元素数组：

```
var $my_elements = $( "li" );
```

现在假设你希望找到这个数组中的所有锚元素。可以这样做：

```
$my_elements.find( "a" );
```

问：jQuery有没有提供一种方法把一个元素包在另一个元素中？

答：确实有这样的方法。假设你想把一个ID为oreilly的图像包在一个锚元素中。可以这样做：

```
$( "img#oreilly" ).wrap( "<a href='http://www.oreilly.com'></a>" );
```



试一试

有一段时间没有更新文件了。增加“Restore Menu”按钮的相应代码，另外加入增减叶子图标的功能（在替换素食的位置上增加和删除veg_leaf类）。本章的文件都可以从<http://www.thinkjquery.com/chapter04/>下载，可以与你的代码做个比较。

Our Menu

[Go Home](#) [Restore Menu](#)

这是我们用veg...
Leaf类增加的叶子

Dinner Entrees

- Braised Delight
Tofu, caramelized onions, carrots, baby spinach, roasted red grapes, braising jus
- House Grilled Panini
Tofu, provolone, arugula, cherry tomatoes, sourdough milk, balsamic glaze
- House Slider
smoked eggplant, mozzarella, Portobello Mushrooms, balsamic vinegar, onion, carrots, Multi-grain roll, goat cheese
- Southwest Slider
whole chiles, Portobello Mushrooms, pepperjack cheese, onion, carrots, sliced avocado, Multi-grain roll

Soups and Salads

- Coconut Soup
coconut milk, Tofu, vegetable broth
- Steep Detox
Tofu, mushrooms, seasonal vegetables, vegetable broth
- Hot and Sour Soup
Tofu, carrots, Chinese mushrooms, chili, vegetable broth
- Stuffed Baked Potatoes
potato, Tofu, seasonal veggies, cheddar & jack cheese, creamy Alfredo sauce

Avocado Rolls

Avocado, Whole chiles, Sweet red peppers, Cilantro

太棒了！顾客非常喜欢这个
新Web菜单，最棒的是，现在我们不再需
要维护两个不同的菜单，它们都在同一个
页面上！

现在Alexandra可以专心地制作
美味的新菜单，而不用再担心
网站了。



JQuery工具箱

你已经掌握了第4章，现在你的工具箱里又增加了DOM管理、遍历、数组和过滤器。

DOM管理

可以根据需要在DOM中增加、替换和删除元素。

- `detach`
- `remove`
- `replaceWith`
- `before`
- `after`

DOM遍历

DOM遍历就是在DOM树上爬行来管理DOM。

可以使用元素关系，利用类似`parent`、`child`等关联方法找到想要的元素。

方法串链是一种很高效的方法，可以快速地遍历DOM。

数组

jQuery数组可以存储各种类型的内容（包括元素），以便以后访问。

类似于变量，数组前面加上\$，指示它存储的是特殊的jQuery内容。

过滤器

过滤方法可以帮助缩小所选元素的范围。

`first`

`equal`

`last`

`slice`

`filter`

`not`

更多资源请访问稀酷客(www.ckook.com)

5 jQuery效果与动画

迈出一小步，前进一大步

看我的转身多漂亮，我打赌你
肯定做不到！



页面动作要漂亮。不过如果页面本身不漂亮，人们可能根本不会使用你的网站。这就需要引入jQuery效果和动画。这一章中，你将了解如何让页面上的元素随时间变换、显示或隐藏相关的特定元素部分，以及缩放页面上的元素。所有这些都会在用户面前闪亮进行。你还会看到如何调度这些动画，使它们按不同的间隔发生，为页面赋予十足的动感。

更多资源请访问稀酷客(www.ckook.com)

DoodleStuff需要一个Web应用

DoodleStuff为Web镇的孩子们提供了很多酷炫的艺术作品。多年前，DoodleStuff建立了一个流行网站，为孩子们提供交互式艺术应用。公司的粉丝团迅速扩大，如今要应对如此众多的请求真有些招架不住。

为了满足DoodleStuff更多新用户的需求，Web项目总监希望建立一个不再使用Flash或任何其他浏览器插件的应用。



为孩子们创建的项目重点在于子类更有意思、更便于传播。你能不能为这些6到10岁年龄段的群体构建一个应用？我们重视大量视觉效果，还需要一定的交互性。不过，不要用Flash，拜托！

更多资源请访问稀酷客(www.ckook.com)

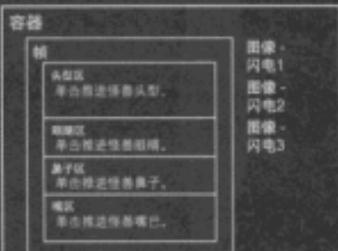
怪兽变脸

下面是web项目总监提供的项目蓝图，另外还给出了图形设计师为这个应用制作的文件。

怪兽变脸项目

这个怪兽变脸应用是专为目标年龄段的孩子们娱乐开发的，孩子们可以通过揍和10种不同的头型、眼睛、鼻子和嘴巴，让他们的怪兽改头换面。怪兽部分面部的变换要通过动画呈现。

用户界面

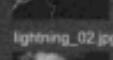


容器

头型区	单击替换怪兽头型。	图像 - 闪电1
眼睛区	单击替换怪兽眼睛。	图像 - 闪电2
鼻子区	单击替换怪兽鼻子。	图像 - 闪电3
嘴巴区	单击替换怪兽嘴巴。	

9次单击之后，各个图片会“凹进”到开始位置。

图形文件

frame.png 宽: 545 像素 高: 629 像素	headstrip.png 宽: 3670 像素, 高: 172 像素	lightning_01.jpg
		
	eyesstrip.png 宽: 3670 像素, 高: 79 像素	lightning_02.jpg
		
	nosesstrip.png 宽: 3670 像素, 高: 86 像素	lightning_03.jpg
		
	mouthstrip.png 宽: 3670 像素, 高: 117 像素	
		

动画

怪兽脸部变化的动画显示过程。



闪电动画显示过程。



闪电图像要淡入，然后快速淡出，就好像瞬间一道闪光。

你已经得到项目需求的很多细节，还得到了需要的图像文件，不过图形设计师没有写任何HTML或CSS，你要从这里开始着手。刚开始需要做些什么？

更多资源请访问稀酷客(www.ckook.com)

怪兽变脸需要布局和定位

在编写任何jQuery代码之前，我们肯定要谈到关于适当建立结构和样式的很多话题。对现在来说，这一点尤其重要。如果没有首先建立适当的布局和位置，效果和动画可能很快就会出问题。那时你可能会盯着你的jQuery代码，奇怪它为什么没有按你希望的那样在浏览器中正常工作，再没有比这更糟糕的了。可以先把你的想法用草图画出来，考虑在屏幕上会发生什么，这是个不错的主意。

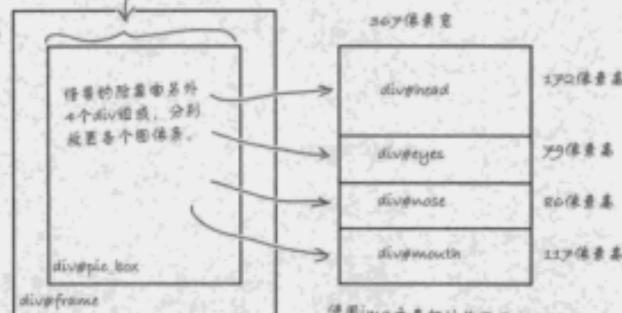


各个怪兽图像条都是340px像素宽，不过我们一次只能显示340px像素。

可以用哪个CSS属性做到这一点？

这应当是一个545像素宽的div，用来设置怪兽的div。

……这是一个340px像素宽的div，用来设置怪兽脸部图像。



使用img元素把这些图像多张海报放在怪兽的脸谱适当的div中。例如，对应headstrip.jpg的img标记应该放在div@head中。

更多资源请访问稀酷客(www.ckook.com)



Exercise

```

body>
<header id="top">
<p>Make your own monster face by clicking on the picture.</p></header>

<div id="frame">
  <div id="pic_box">
    <div id=".....class="face"></div>
    <div id=".....class="face"></div>
    <div id=".....class="face"></div>
    <div id=".....class="face"></div>
  </div>
</div>
<script type="text/javascript" src="scripts/jquery-1.6.2.min.js"></script>
<script type="text/javascript" src="scripts/my_scripts.js"></script>
</body>
```



index.html

```

#frame {
  position: absolute;
  left:100px;
  top:100px;
  width:545px;
  height:629px;
  background-image:url(images/frame.png);
  z-index: 2;
  overflow: hidden;
}

#pic_box{
  position: relative;
  left:91px;
  top:84px;
  height:460px;
  z-index: 1;
  overflow: hidden;
}

.face{
  position: absolute;
  left:0px;
  top:0px;
  z-index: 0;
}

#head{
  height:172px;
}

#eyes{
}

#nose{
}

#mouth{
}
```



my_style.css

更多资源请访问稀酷客(www.ckook.com)



Exercise Solution

在HTML和CSS文件的各个空中填入CSS ID、属性或者能够帮助建立怪兽变脸应用布局和位置的相应设置。如果有疑问，请看前两页的说明，我们已经帮你填了一些空。

```

body>
<header id="top">
<p>Make your own monster face by clicking on the picture.</p></header>

<div id="frame">
  <div id="pic_box">
    <div id="head"...class="face"></div>
    <div id="eyes"...class="face"></div>
    <div id="nose"...class="face"></div>
    <div id="mouth"...class="face"></div>
  </div>
</div>
<script type="text/javascript" src="scripts/jquery-1.6.2.min.js"></script>
<script type="text/javascript" src="scripts/my_scripts.js"></script>
</body>

```



index.html

```

#frame {
  position: absolute;          动画调整元素的位
  left:100px;                  置时，需要使用绝
  top:100px;                  对 (absolute) 或相对
  width:545px;                (relative) 定位。
  height:629px;
  background-image:url(images/frame.png);
  z-index: 2;
  overflow: hidden;           ↓
}                                指overflow属性设置
                                为"hidden"，从而将想
                                被保留条中延伸到pic_box
                                区以外的部分。
#pic_box{
  position: relative;          ↓
  left:91px;
  top:84px;
  width:367px;
  height:460px;
  z-index: 1;
  overflow: hidden;           这可以使用CSS
                                "clip" 属性达到这个
                                目的。
}

```

```

.face{                         .face{
  position: relative;          position: relative;
  left:0px;                     left:0px;
  top:0px;                      top:0px;
  z-index: 0;                   z-index: 0;
}

#head{                         #head{
  height:172px;                height:172px;
}

#eyes{                         #eyes{
  height:79px;                 height:79px;
}

#nose{                         #nose{
  height:86px;                 height:86px;
}

#mouth{                        #mouth{
  height:117px;                height:117px;
}

```



my_style.css

更多资源请访问稀酷客(www.ckook.com)

再来一点结构和样式

接下来是对HTML和CSS文件的结构修改。把下面的代码增加到index.html和my_style.css文件，可以从www.thinkjquery.com/chapter05找到需要的图像文件。

动手做！

增加一个容器，在其中嵌套闪电图像。

```
<div id="container">
  
  
  
<div id="frame">
  <div id="pic_box">
    <div id="head" class="face"></div>
    <div id="eyes" class="face"></div>
    <div id="nose" class="face"></div>
    <div id="mouth" class="face"></div>
  </div>
</div>
</div>
```



index.html

```
#container{
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: 0;          我们希望闪电图像在
}                      嵌套时不可见。
.lightning{
  display: none;      ↓
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: 0;          想看动画显示元素时，是把
}                      它们的position属性设置为
                      absolute, fixed或relative。
```

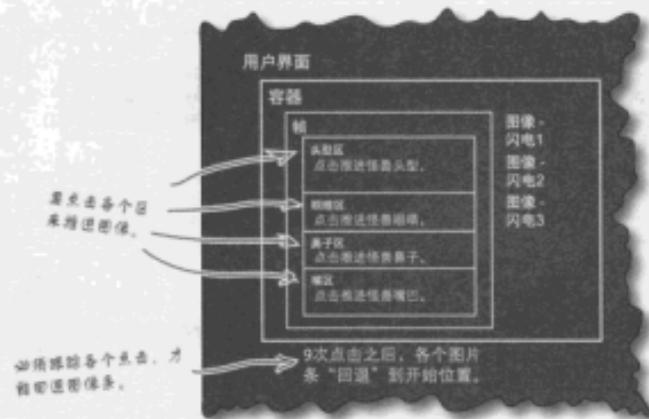
```
body{
  background-color: #000000;
}
p{
  color: #33FF66;
  font-family: Tahoma, Verdana, Arial,
  Helvetica, sans-serif;
  font-size: 12px;
}
#text_top {
  position: relative;
  z-index: 4;
}
```



my_style.css

实现界面点击

既然已经建立了怪兽变脸应用的布局，下面来建立蓝图要求的其他用户界面部分。这一部分都是通过点击来让某件事情发生。实际上你已经在前面4章做过这个工作，现在应该很轻松了。



there are no Dumb Questions

问：我对CSS定位还有点不明白。实现jQuery效果和动画为什么还需要CSS定位呢？

答：position是一个CSS属性。它会控制浏览器布局引擎如何以及在哪里放置元素。jQuery的很多效果都是使用CSS position属性完成的。如果你还不清楚，想复习一下，可以参考Mozilla开发中心提供的一个非常棒的解释：

http://developer.mozilla.org/en/CSS/position#Relative_and_absolute

问：我们想要动画显示元素时为什么要把CSS position属性设置为absolute, fixed或relative?

答：如果保留CSS position属性仍设置为它的默认设置（也就是static），就无法应用上、右、左或下定位。使用animate函数时，我们必须能够设置为些位置，而static不支持这一点。其他位置设置（absolute, fixed和relative）都是允许的。

问：你前面提到有一个浏览器布局引擎。这到底是什么？

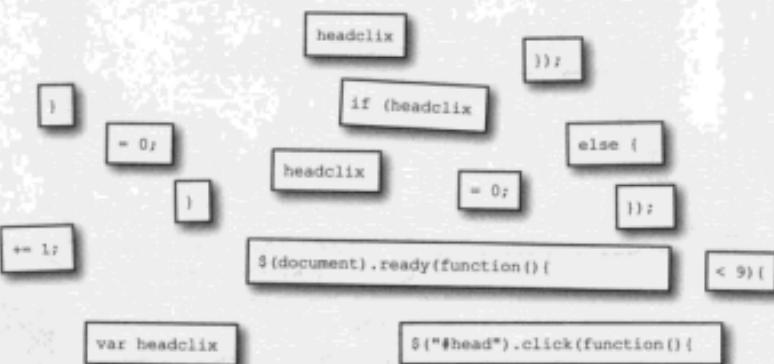
答：浏览器布局引擎是浏览器的核心部分，负责解释HTML和CSS代码，并在浏览器的视窗里显示（视窗就是为访问者显示内容的窗口）。Google的Chrome和Apple的Safari都使用Webkit浏览器布局引擎。Firefox使用Gecko布局引擎。Microsoft Internet Explorer则使用一个名为Trident的布局引擎。

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴

按正确的顺序摆放下面的代码磁贴，使





jQuery磁贴答案

按正确的顺序摆放下面的代码磁贴，使

```

$(document).ready(function() {
    var headclix = 0; // 这个变量初始值为0，因为还没有点击任何元素。
    $("#head").click(function() {
        if (headclix < 9) { // 这个条件限制用户最多点击9次。
            headclix += 1; // 最后要在设置你的动画代码。
            // 将headclix变量设置为原值加1。
        } else { // 最好是在这里设置你通常需要的代码。
            headclix = 0; // 第9次点击之后重置headclix
            // 变量为0。
        }
    });
});

```

如果headclix大于或等于9，就做下面的工作。

能不能采用某种方式重用这个代码，
让其他元素也可以点击？

当然！

各元素都遵循与



更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

填写下面的jQuery脚本，让眼睛、鼻子、嘴元素都可以点击。稍后还会为各个点击增加一些功能。一定要按正确的顺序放置变量和条件语句来检测第9次点击。

```
$ (document) .ready(function() {
    $("#head") .click(function() {
        if (headclix < 9) {
            headclix += 1;
        }
        else {
            headclix = 0;
        }
    });
});
```

```
});
```



my_scripts.js

Sharpen your pencil

Solution

你已经按正确的顺序放置变量和条件语句来检测第9次点击，使眼睛、鼻子、嘴元素都可以点击。

```

$(document).ready(function() {
    var headclix = 0, eyeclix=0, noseclix = 0, mouthclix = 0;
    $("#head").click(function(){
        if (headclix < 9){
            headclix += 1;
        }
        else{
            headclix = 0;
        }
    });
    $("#eyes").click(function(){
        if (eyeclix < 9){
            eyeclix += 1;
        }
        else{
            eyeclix = 0;
        }
    });
    $("#nose").click(function(){
        if (noseclix < 9){
            noseclix += 1;
        }
        else{
            noseclix = 0;
        }
    });
    $("#mouth").click(function(){
        if (mouthclix < 9){
            mouthclix += 1;
        }
        else{
            mouthclix = 0;
        }
    });
});

```

可以在变量之间加进号来声明和设置多个变量。

居住在怪兽身上的各个部分都可以点击，而且在动画图很多之前最多只允许9次点击。

注意，各个点击函数的结构很类似，只是有些细微的差别。这样适合重用。

别担心，别那么惺忪，这个内容我们会在第9章讨论。

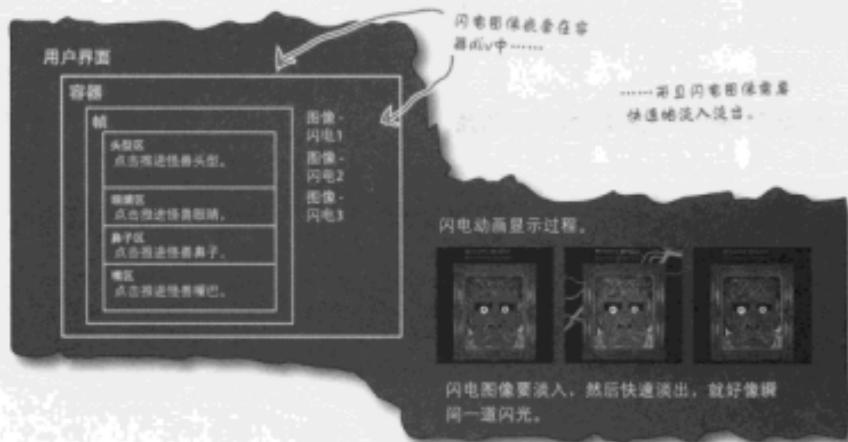


my_scripts.js

更多资源请访问稀酷客(www.ckook.com)

制造闪电效果

接下来实现闪电效果。在制作效果之前先来回顾蓝图的要求。



我们在第1章用slides和fades做过类似的工作。不能直接使用这些方法来实现闪电效果吗？

可能可以，不过还有更好的办法。

我们在第1章研究过jQuery提供的现成效果，不过下面更深入一些。

JQuery如何动画显示元素？

浏览器加载一个CSS文件时，会设置页面上元素的视觉属性。使用jQuery的内置效果时，JS解释器会修改这些CSS属性。在你眼前动画呈现这些改变。不过，这并不是魔法……关键就在于CSS属性。下面再来看之前已经了解的一些内容。

hide, show和toggle改变CSS display属性

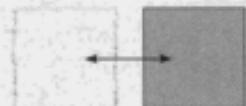
hide



show



toggle

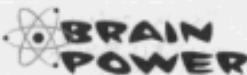


JS解释器将所选元素的CSS display属性改为none，并将它从布局中删除。

JS解释器改变所选元素的CSS display属性，使它可见。

如果一个元素隐藏，JS解释器会让它显示，反之，如果原先显示，则将其隐藏。

jQuery效果动态改变CSS属性，让页面的变化就在用户的眼前发生。



hide, show和toggle的核心都是display属性。不过这次我们需要让脸的组成部分滑动。另外要让闪电淡入淡出。你认为jQuery实现淡入淡出和滑动时要改变哪些CSS属性？

更多资源请访问稀酷客(www.ckook.com)

淡入淡出效果动画要改变CSS opacity属性

fadeIn



利用fadeIn，JavaScript解释器将所选元素的CSS opacity属性从0改为100。

fadeTo



fadeTo会动画显示所选元素，将其改为某个特定的透明度百分比。

fadeOut



利用fadeOut，JavaScript解释器将所选元素的CSS opacity属性从100改为0，不过在页面上仍占据该元素的空间。

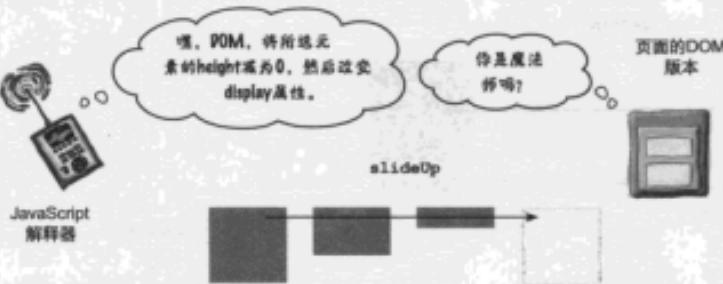


Geek Bits

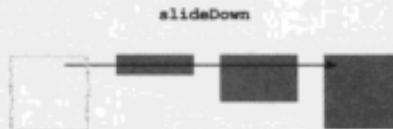
CSS opacity属性并非在所有浏览器上都有相同的表现。幸运的是，jQuery会负责为我们处理这个问题。实际上，你只需要知道这一点就足够了！

更多资源请访问稀酷客(www.ckook.com)

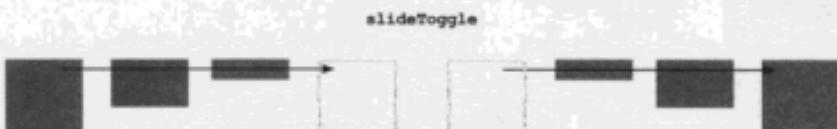
滑动的关键是height属性



JavaScript解释器告诉DOM将所选元素的CSS height属性改为0，然后把display属性设置为none。这实际上就是一个hide外加一个slide。



JavaScript解释器将所选元素的高度从0动画变为CSS样式中设置的高度值，使元素在页面上显示。



JavaScript解释器检查图像的高度是完全高度还是0，并根据发现的结果切换滑动效果。如果元素高度为0，JavaScript解释器会将它下滑。如果元素为完全高度，JavaScript解释器则将元素上滑。



这么说我只能上滑或下滑是吗？如果我想让元素横向左滑或右滑怎么做呢？

jQuery只提供了让元素上滑或下滑的现成效果。

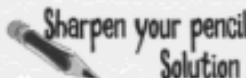
jQuery中是找不到slideRight或slideLeft方法的（至少写这本书时还没有）。不要担心，稍后就会想办法解决这个问题……

jQuery中可找不到slideRight或slideLeft方法。

Sharpen your pencil

jQuery的哪些现成效果可以在这个怪兽变脸应用中运用？对于下面的各组效果，请回答这些效果对我们是否有帮助，并解释选择或不选择这些效果的原因。

效果	可以用吗？	为什么？
显示/隐藏		
滑动		
淡入淡出		



jQuery的哪些现成效果可以在这个怪兽变脸应用中运用？

效果	可以用吗？	为什么？
显示/隐藏	不可以	显示/隐藏效果对这个怪兽变脸应用没有帮助，因为我们不需要动画改变任何元素的display属性。
滑动	不可以	超过了，但不合适。我们需要让图像向左滑动，slideup, slidetdown和slides只允许改变height属性。我们还应当能够改变left属性。
淡入淡出	可以	可以使用fade来满足图片上的一个需求：闪电图像要快速地淡入淡出，就像一道闪光。

实现淡入淡出效果

蓝图要求闪电图像淡入淡出，不过我们需要快速地完成，使它看起来好像一道闪光。下面更深入地研究淡入淡出效果，了解如何实现闪电。

下面是第一个img元素的ID。
`$("#lightning1").fadeIn("fast");`

可以使用某些字符串参数，slow, normal
或fast……

……或者也可以使用一个毫秒值。例如，在这里放入1000，动画效果就需要1秒完成。

`$("#lightning1").fadeIn(1000);`

"duration" 参数要放在这里。它控制完成效果持续多长时间。

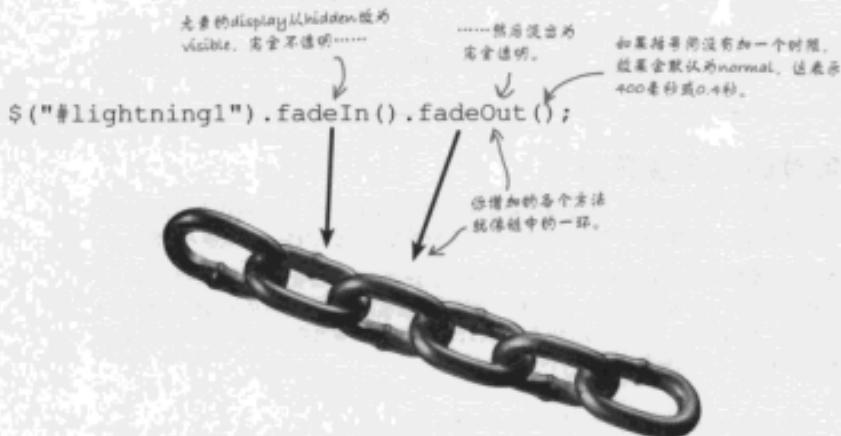
Make it Stick

1秒 = 1000 毫秒



用方法串链结合效果

闪电需要反复地淡入淡出。不要单独地编写这些效果，可以使用串链，这在第4章讨论遍历DOM时已经简单使用过。方法链是jQuery的一个特性，将方法链接在一起在返回的一组元素上分别运行。利用方法串链，闪电效果的编写会更容易也更简洁，下面就来仔细了解。



写出完成以下所示各个步骤的jQuery代码行。

- ① 将#lightning1元素淡入，时限为1/4秒。

- ② 串链另一个效果，使#lightning1元素在1/4秒内淡出。



Exercise Solution

写出完成以下所示各个步骤的jQuery代码行。

- ① 将#lightning1元素淡入，时长为1/4秒。

```
$( "#lightning1" ).fadeIn( "250" );
```

- ② 串联另一个效果，使#lightning1元素在1/4秒内淡出。

```
$( "#lightning1" ).fadeIn( "250" ).fadeOut( "250" );
```

用定时函数反复闪烁

现在已经能够让闪电淡入淡出，不过项目需求中要求闪电反复闪烁。真正的闪电会划破长空，在下一次划过天空之前通常有一个时间间隔。所以我们需要一种方法反复地完成淡入淡出。

再回想前面几章需要完成重复的任务时你是怎么做的？没错，利用函数！函数在第3章中第一次出现，我们建立了一个可重用的click函数和一个随机数生成器，现在可以使用函数来运行淡入淡出，等待一段时间后，按给定的间隔再次运行。这会为怪兽变脸应用提供耀眼的闪电效果。下面来看完成这个工作的函数。

嘿，JavaScript，为我们
构建一个新函数。
嘿，JavaScript，
我们想调用函数时直接
用这个名字。

这是定时参数（名为t的变量）。先了解一下，下面还会使用。

```
function lightning_one(t) {
    $( "#lightning1" ).fadeIn(250).fadeOut(250);
    setTimeout("lightning_one()",t);
}
```

这一行包含
jQuery的基本代
码。

setTimeout方法告诉JavaScript
运行一个函数，另外在
再次运行之前先等待一段
时间。

setTimeout方法告诉JavaScript的一
些非凡能力。可以告诉JavaScript
希望这个函数反覆地调
用它自己。

这是超时时间。单位为毫
秒，类似于前几天看到的效
果时限。

只需要3行代码，就完成了第一个闪电图像的定时闪电函数。现在试着为另
外两个闪电图像编写函数。

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴

按正确的顺序放置下面的代码磁贴，完成另外两个闪电图像的定时闪电函数。

function lightning_two(t){

}

function lightning_three(t){

}

.fadeIn(250)

t);

\$("#lightning2")

setTimeout(

.fadeOut(250),

.fadeIn(250)

"lightning_two()",

.fadeOut(250);

"lightning_three()",

t);

\$("#lightning3")

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴答案

按正确的顺序放置下面的代码磁贴，完成另外两个闪电图像的定时闪电函数。

```

function lightning_two(t) {
    $("#lightning2").fadeIn(250).fadeOut(250);
    setTimeout("lightning_two()", t);
}

function lightning_three(t) {
    $("#lightning3").fadeIn(250).fadeOut(250);
    setTimeout("lightning_three()", t);
}

```

there are no Dumb Questions

问：`fadeIn().fadeOut()`与`toggle`不一样吗？

答：阿得好！它们确实不一样。`toggle`方法只是一个方法，将所选元素从隐藏状态切换为可见状态，或者反之，这取决于所选元素的当前状态。将`fadeIn`和`fadeOut`串连在一起会创建一种顺序效果，先将所选元素淡入，然后当这个效果结束时，再让它淡出视图。

问：`setTimeout`方法是个新方法。这是一个jQuery方法还是一个JavaScript方法？

答：`setTimeout`方法实际上是一个JavaScript方法，可以用来控制jQuery动画的很多方面。在后面的章节中还会更多地了解`setTimeout`函数，特别是第7章。如果你现在就想了解，可以访问Mozilla开发中心：<https://developer.mozilla.org/en/window.setTimeout>，或

找一本David Flanagan编写的关于JavaScript的书：《JavaScript: The Definitive Guide》(O'Reilly; <http://oreilly.com/catalog/9780596805531>)，这本书很棒，也很全面。

问：使用隐藏效果时，元素直接消失了。怎么能让它慢下来呢？

答：要让隐藏、显示或切换效果“慢下来”，只需要在括号中给出一个时限参数。我们可以像这样完成第1章的隐藏效果：

更多资源请访问稀酷客(www.ckook.com)

在脚本中增加闪电函数

使用前一页练习中完成的代码，更新怪物变脸应用的脚本文件。

动手做！

这些代码行
将用星号
来粗体显示
的函数。

为节省这一页的篇幅，这里删
除了一些换行符。如果你的脚
本中换行与这里有所不同，不
用担心。

```

$(document).ready(function(){
    var headclix = 0, eyeclix = 0, noseclix = 0, mouthclix = 0;
    lightning_one(4000); // 指导中的数字是将要传入setTimeout方法的参数，单位为
    lightning_two(5000); // 毫秒。利用这些参数，可以设置闪电闪烁的频率。
    lightning_three(7000);

    $("#head").click(function(){
        if (headclix < 9){headclix+=1;}
        else{headclix = 0;}
    });

    $("#eyes").click(function(){
        if (eyeclix < 9){eyeclix+=1;}
        else{eyeclix = 0;}
    });

    $("#nose").click(function(){
        if (noseclix < 9){noseclix+=1;}
        else{noseclix = 0;}
    });

    $("#mouth").click(function(){
        if (mouthclix < 9){mouthclix+=1;}
        else{mouthclix = 0;}
    });
});

// end doc.onready function

function lightning_one(t){
    $("#container #lightning1").fadeIn(250).fadeOut(250);
    setTimeout("lightning_one()",t);
}
function lightning_two(t){
    $("#container #lightning2").fadeIn("fast").fadeOut("fast");
    setTimeout("lightning_two()",t);
}
function lightning_three(t){
    $("#container #lightning3").fadeIn("fast").fadeOut("fast");
    setTimeout("lightning_three()",t);
}

```

这是闪电
函数定义。





试一试

在你喜欢的浏览器中打开这个页面，看看闪电效果是否成功。



通过结合JavaScript的
setTimeout方法完成闪电
效果的淡入淡出效果。

闪电以不同的时间间隔迅速
淡入和淡出，模拟真实
的闪电效果。

到目前为止，你已经完成了点击函数，而且让3个闪电图像按不同的时间间隔迅速淡入淡出。下面来看蓝图中还有什么任务没有完成。

怪兽变脸项目

这个怪兽变脸应用是专为年龄段的孩子们娱乐开发的，孩子们可以通过揍和10种不同的头型、眼睛、鼻子和嘴巴，让他们的怪兽改头换面。怪兽面部面部的变换要通过动画呈现。

动画

怪兽脸部变化的动画显示过程。



闪电动画显示过程。

在这方面，现在我们希望让这些面部
向左滑动。不过并没有现成的这种
简单效果。有没有其他方法呢？



现成的效果固然很好，不过它们并
不能实现你想要的一切。

现在需要建立一个定制效果，
能够让怪兽面部向左滑动。

蓝图中的这一部分是这个项目最后需
解决的难题。

更多资源请访问稀酷客(www.ckook.com)

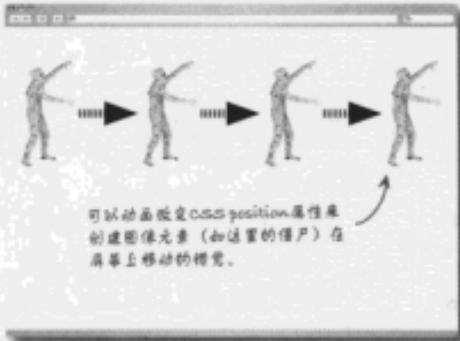
用animate完成DIY效果

jQuery没有slideRight或slideLeft方法。但是在这个项目中，目前必须做到这一点。这是否意味着你的怪兽变脸应用没有成功的希望呢？

别担心！jQuery提供了animate方法，可以用来建立你自己的效果。利用animate，你可以创建定制动画，完成那些现成效果做不到的事情。animate方法允许你动画改变所选元素的CSS属性，另外还允许同时动画调整多个属性。

下面来看利用animate方法可以做到的一些事情。

动作效果



缩放效果



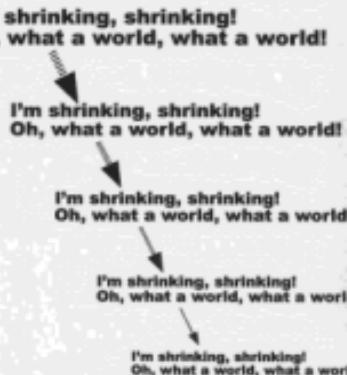
每次点击时要让怪兽脸部向
左滑动需要动画改变哪个CSS
属性？

什么能动画，什么不能？

利用animate方法，还可以动态改变font属性来创建文本效果。另外可以在一个animate调用中动画改变多个CSS属性，这会让你的Web应用更加光彩夺目。

animate方法确实很棒，不过它也有限制。在底层，animate会使用大量数学运算（好在你不必担心这个方面），所以你只能处理使用数值设置的CSS属性。必须知道这个限制。不过这不会影响你的想象力，animate可以提供各种灵活性，让你充分享受其中的乐趣。

文本效果



可以动画改变CSS属性来创建文本飞出、扩大或缩小的感觉。

这些只是一些例子。我们靠很多很多天才和苦干实现所有可能的效果。



Watch it!

animate方法只适用于使用数值设置的CSS属性。

- borders, margin, padding
- bottom, left, right 和 top position
- element height, min-height 和 max-height
- background position
- element width, min-width 和 max-width
- letter spacing, word spacing
- font size
- text indent
- line height



animate方法特写

表面上看，animate与之前用过的其他方法很类似。

选择您想动画的元素。

调用animate方法。

```
$( "#my_div" ).animate({left:"100px"},500);
```

在这个例子中，我们在动画改变CSS属性left……

……并设置这个属性为100px。

第二个参数是时间，单位为毫秒。通过这个参数可以控制完成动画花费的时间。

第一个参数是必选的，这里必须放入这个参数才能完成动画。第二个参数是可选的。

不过，animate最强大的特性之一是它能够同时改变所选元素的多个属性。

```
$( "#my_div" ).animate({
    opacity: 0,
    width: "200",
    height: "800"
}, 5000);
```



Watch it!

CSS的参数要使用**DOM标准**设置，而不是**CSS标准**。



BRAIN
BARBELL

你认为浏览器中在后台发生了什么，使得animate方法可以在用户眼前完成属性的改变？

更多资源请访问稀酷客(www.ckook.com)

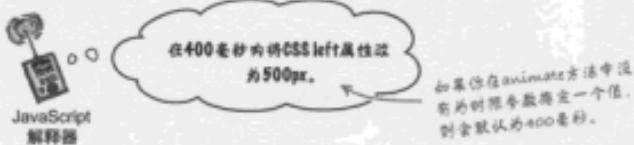
你现在的位置：

随时间动画改变样式

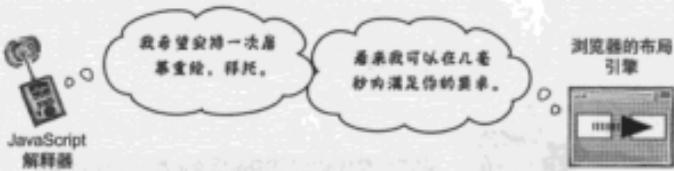
你在电影或电视屏幕上看到的视觉效果和动画都使用了运动错觉。效果技术人员和动画设计会提供一系列图像，按某个特写的速率一次播放一个图像，来实现这种错觉——你可能见过那种技术含量不高的翻页书，你迅速翻动页面时就能达到这种效果。

在浏览器屏幕上也是一样，只不过我们没有一组要处理的图像。实际上，JavaScript解释器会反复运行一个函数，由它改变动画元素的样式。浏览器把这些改变绘制（或重绘）到屏幕上。元素的样式改变时，用户就会看到一种运动错觉或对元素的改变。

- ① animate运行时，JavaScript解释器设置一个定时器的动画时限。



- ② JavaScript解释器告诉浏览器的布局引擎，要改变animate方法参数中指定的CSS属性。浏览器的布局引擎在屏幕上以可视化方式展示这些CSS属性。



- ③ JavaScript解释器反复调用函数来改变元素的CSS属性，直至第一步中设置的定时器到期。每次函数运行时，都会在屏幕上显示所做的改变。

- ④ 浏览器展示对元素的改变时，访问者会有元素正在移动的错觉。



更多资源请访问稀酷客(www.ckook.com)



将各个定制动画代码块与运行时相应的结果连线。

`$("#my_div").animate({top: "150px"}, "slow")`

同时动画修改所有段落的左边距和
右边距。

```
 $("p").animate({
    marginLeft:"150px",
    marginRight:"150px"
});
```

在半秒内将#my_div的right位置改为
0。

`$("#my_div").animate({width: "30%"}, 250)`

在默认的400毫秒时间内动画改变
所有段落的字母间隔。

`$("#my_div").animate({right: "0"}, 500)`

同时动画改变#my_div的padding和
width属性。

`$("p").animate({letterSpacing:"15px"});`

在slow时限内动画改变#my_div的
top位置。

```
 $("#my_div").animate({
    padding: "200px",
    width: "30%"
}, "slow")
```

在fast时限内动画改变所有图像的
height属性。

`$("img").animate({height: "20px"}, "fast")`

在1/4秒内动画改变#my_div的width
属性。



答案

将各个定制动画代码块与运行时相应的结果连线。

```
$( "#my_div" ).animate({top: "150px"}, "slow")
```

同时动画修改所有段落的左边距和
右边距。

```
$("p").animate({
    marginLeft:"150px",
    marginRight:"150px"
});
```

在半秒内将#my_div的right位置改为
0。

```
$( "#my_div" ).animate({width: "30%"}, 250)
```

在默认的400毫秒时间内动画改变所
有段落的字母间隔。

```
$( "#my_div" ).animate({right: "0"}, 500)
```

同时动画改变#my_div的padding和
width属性。

```
$("p").animate({letterSpacing:"15px"});
```

在slow时限内动画改变#my_div的
top位置。

```
$( "#my_div" ).animate({
    padding: "200px",
    width: "30%"
}, "slow")
```

在fast时限内动画改变所有图像的
height属性。

```
$("img").animate({height: "20px"}, "fast")
```

在1/4秒内动画改变#my_div的width
属性。

到底从哪里到哪里？

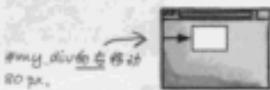
关于animate要记住重要的一点，它会把当前CSS属性改为第一个参数中设置的CSS属性。要让你的定制动画生效，需要仔细考虑目前CSS中的设置。在前面的例子中，我们把#my_div的left位置改为100px。屏幕上完成的具体结果完全取决于#my_div的当前CSS left属性设置。

当前CSS属性



```
#my_div{
    left: 20px;
}
```

动画CSS属性



```
$("#my_div").animate({left:"100px"});
```

元素动画调整到一个绝对位置。

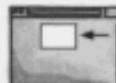
如果当前属性是一个不同的值，我们会得到不同的结果。

当前CSS属性



```
#my_div{
    left: 200px;
}
```

动画CSS属性



```
$("#my_div").animate({left:"100px"});
```

#my_div向左移动 200px. 因为100>子原来的200.

真是太棒了，不过怎么再来完成魔兽变脸应用呢？



都是相对的。

要让魔兽变脸应用中的魔兽脸部分按你希望的方式滑动，必须考虑它们当前的位置是什么，还要确定相对于最后一次动画改变的位置希望如何进行改变。

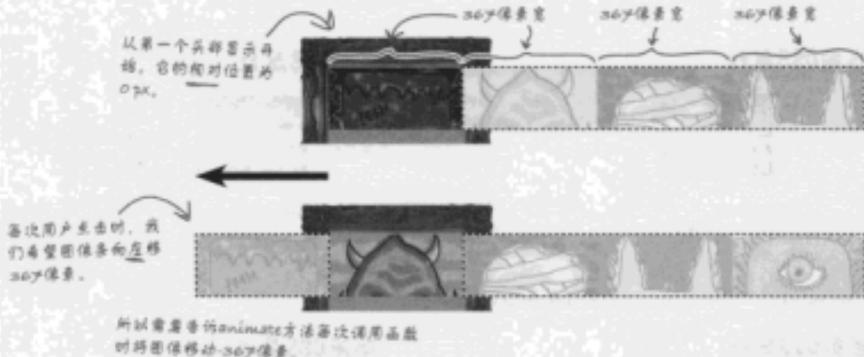
更多资源请访问稀酷客(www.ckook.com)

你现在的位置：

205

绝对 vs. 相对元素移动

应该记得，我们把要显示的图像条嵌套在一个ID为#pic_box的div中。当前CSS中div#pic_box的left属性设置为91px。为了得到我们想要的向左滑动效果，现在来考虑如何移动图像条。



考虑上一页绝对动画的例子。

这会告诉animate方法将#my_div的left位置准确地设置为100像素。
\$ ("#my_div").animate({left:"100px"});

不过如何告诉它每次调用animate方法时将元素移动 -36px 像素呢？
\$ ("#head").animate({left:"-36px"});

相对动画：每次移动这么多

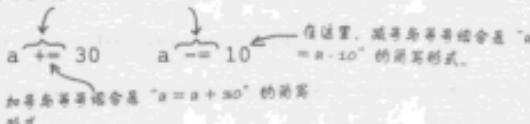
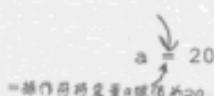
利用绝对动画，会把元素移动到一个绝对位置。采用相对动画时，则会相对于上一次动画移动到的位置来移动元素。

不过如何用animate方法让一个元素相对移动呢？

利用操作符组合实现元素的相对移动

有一些特殊的JavaScript操作符可以在每次调用animate方法时让元素移动相同的量。这些操作符称为赋值操作符，因为它们通常用来为一个变量赋值，将新值增加到变量的当前值上。听上去很复杂，但实际上并没有那么复杂。

等等是一个赋值操作符。结合算术操作符和等等时，就会得到一些有用的东西形式。



这些操作符组合可以帮助你创建一个相对动画，将一个值设置为当前值加上或减去一些像素值。

当在第一次调用animate方法时将box元素移动20像素。

```
$("#box").animate({left: "+=20"});
```

每次调用上面的animate方法时#box会发生下面的变化。

假设left从0开始。

animate运行，并设置left为+= 20。

animate运行，并设置left为 += 20。



通过每次推进元素的left位置，实际上会在浏览器窗口中将其向右移。

BULLET POINTS

下面是一些赋值操作符组合：

- $a *= 5$ 是“将a的当前值乘以5并把结果值赋给a”的简写形式。
- $a /= 2$ 是“将a的当前值除以2并把结果值赋给a”的简写形式。



写出完成以下所示各个步骤的jQuery代码行。

- ① 每次调用animate方法时将#head元素向左移动367像素。时限指定为半秒。
- ② 将#head元素移回到它原来的位置(left:0px)，时限指定为半秒。



Exercise Solution

写出完成以下所示各个步骤的jQuery代码行。

- ① 每次调用animate方法时将#head元素向左移动367像素。时限指定为半秒。

```
$( "#head" ).animate({left: "-=367px"},500);
```

- ② 将#head元素移回到它原来的位置(left:0px)，时限指定为半秒。

```
$( "#head" ).animate({left: "0px"},500);
```

这个动画动画将重置像
怪物头部，让人感觉又
回到了最初状态。

there are no Dumb Questions

问：有些人不希望动画干扰他们的Web页面体验。如果我想允许用户关掉动画该怎么做？

答：这个问题很好。动画可能会让人讨厌，还可能存在可访问性问题。如果你希望允许用户关掉Web页面上的动画，可以将一个点击按钮关联到下面这段代码（你应该知道怎么做）：

```
$fx.off = true;
```

停止动画还有一个有用的方法：jQuery的stop方法。这两个方法的更多信息都可以在jQuery网站上找到：

<http://api.jquery.com/jQuery.fx.off/>
<http://api.jquery.com/stop/>

问：你说“CSS属性的参数必须用DOM标准来设置，而不是CSS标准”。这到底是什么意思？

答：回答好！animate方法的参数要用DOM标准编写（也就是DOM记法）而不是CSS记法。

下面是一个具体的例子，可以展示这两者的差别。要用CSS记法设置一个div的边框宽度，可以这样做：

```
div {  
    border-style:solid;  
    border-width:5px;  
}
```

现在，假设你想动画调整这个边框宽度。在jQuery中要使用DOM记法设置边框宽度属性，如下所示：

```
$( "div" ).animate({borderWidth:30}, "slow");
```

注意在CSS记法中，属性名要写成border-width，而在DOM记法中，要写为borderWidth属性。

对于这两种记法之间的差别，如果你想了解有关的更多信息，可以请读这篇文章：

http://www.oxfordu.net/webdesign/dom/straight_text.html

问：如果我想动画修改一个颜色该怎么做呢？

答：要动画完成颜色变换，需要使用jQuery UI，它会增加jQuery没有的更多效果。我们将在第10章讨论jQuery UI，不过不包括效果。一旦了解如何下载jQuery UI，主题以及如何在web应用中包含jQuery UI，就会发现动画改变颜色相当简单。

在脚本中增加animate函数

使用上一页练习中完成的代码，更新怪兽变脸应用的脚本文件。



动手做！

```

$( "#head" ).click(function() {
    if (headclix < 9) {
        $(this).animate({left:"-=367px"},500);
        headclix+=1;
    }
    else{
        $(this).animate({left:"0px"},500);
        headclix = 0;
    }
    // 这里可以使⽤“this”关键字，因为我们在
    // 通过点击元素的click函数中。
});

$( "#eyes" ).click(function() {
    if (eyeclix < 9) {
        $(this).animate({left:"-=367px"},500);
        eyeclix+=1;
    }
    else{
        $(this).animate({left:"0px"},500);
        eyeclix = 0;
    }
});

$( "#nose" ).click(function() {
    if (noseclix < 9) {
        $(this).animate({left:"-=367px"},500);
        noseclix+=1;
    }
    else{
        $(this).animate({left:"0px"},500);
        noseclix = 0;
    }
});

$( "#mouth" ).click(function() {
    if (mouthclix < 9) {
        $(this).animate({left:"-=367px"},500);
        mouthclix+=1;
    }
    else{
        $(this).animate({left:"0px"},500);
        mouthclix = 0;
    }
});

```



my_scripts.js

更多资源请访问稀酷客(www.ckook.com)

你现在的位置 >



试一试

在你喜欢的浏览器中打开页面，确保一切正常。

尝试了定制的方向左滑动效果。

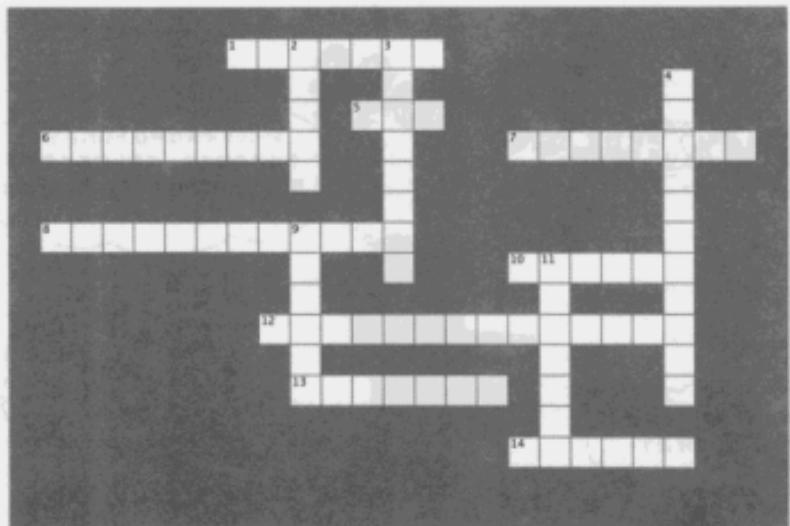
通过几次点击，访问者可以变成自己的怪兽脸。

更多资源请访问稀酷客(www.ckook.com)



jQuery填字游戏

现在可以坐下来，让你的左脑动一动了。这是标准的填字游戏，所有答案都可以在这一章找到。



横向

- 1.hide, show和toggle会动画改变这个CSS属性。
- 5.jQuery效果和动画都是要动态管理_____。
- 6._____ =1000毫秒。
- 7.这个参数会控制效果持续多长时间完成。
- 8.创建元素在屏幕上移动的锚点。
- 10.这个效果方法用于将所选元素动画修改为一个特定的透明度。
- 12.这个jQuery特性允许将希望在一组返回的元素上运行的方法串联起来。

13.animate方法只适用于有_____值的CSS属性。

14.可以动画改变CSS_____和width属性来创建缩放一个元素的

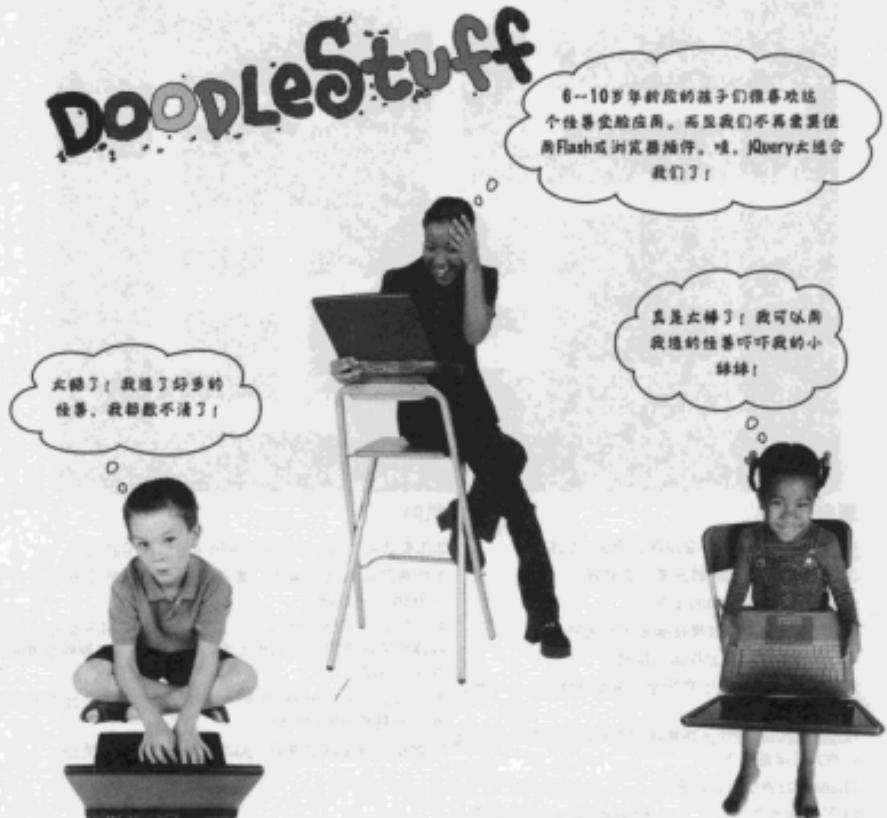
纵向

- 2.想要动画改变一个元素height属性时使用的效果。
- 3.想要动画改变元素时，需要将其位置属性设置为_____，fixed或relative。
- 4.这个效果方法的工作如下：如果所选元素的高度为0，JS解释器将它向下滑动。如果元素为完全高度，JS解释器就将元素向上滑动。
- 9.运行这个jQuery效果时，JS解释器将所选元素的CSS opacity属性从0改为100。
- 11.希望构建定制效果时，jQuery库提供了这个方法。

更多资源请访问稀酷客(www.ckook.com)

看呐，妈妈，不是Flash！

Web项目总监对怪兽变脸应用的成果很满意。你不仅使用了jQuery现成的效果，还结合了专门为满足客户需求建立的定制效果。



更多资源请访问稀酷客(www.ckook.com)



jQuery填字游戏答案

1	P	I	S	P	L	A	Y								
	L			B											S
	I			C	S	S									L
10	O	N	E	S	E	C	O								
11	N	E	S	E	C	O									
12	M	O	T	I	O	N	E	F	E	F	E	T			
	T	O	N	E	F	E	T	A	P	E	T	O			
									N						
13	M	E	T	H	O	P	C	H	A	I	N	I	N	O	
	I								M						
14	N	U	M	E	R	I	C	A	A						
	T								T						
15	H	E	I	G	H	T									



jQuery工具箱

你已经掌握了第5章，现在你的工具箱中又增加了jQuery淡入淡出和滑动效果，还有定制动画。

淡入淡出效果

改变CSS元素的`opacity`属性：

`fadeIn`
`fadeOut`
`fadeTo`

滑动效果

改变CSS元素的`height`属性：

`slideUp`
`slideDown`
`slideToggle`

animate

现成的jQuery效果不能满足要求时，这个方法允许你创建定制效果。

可以在一段时间内动画修改CSS属性。

只适用于使用数值设置的CSS属性。

元素可以绝对或相对移动。

操作符组合（=、+、-）可以更容易地完成相对动画。

6 jQuery与JavaScript

卢克jQuery，
我是你父亲！



jQuery还不能完全独立。尽管这是一个JavaScript库，不过遗憾的是，它还不能完成它的父语言能够做到的所有事情。这一章中，我们将了解创建真正高水平网站需要的一些JavaScript特性，并介绍jQuery如何使用这些特性来创建定制列表和对象，以及如何循环处理这些列表和对象来简化你的工作。

充实 Head First Lounge

你真幸运，你的jQuery能力已经威名远扬了。看看Head First Lounge发来的这个Email。他们需要你的帮助，让访问者得到更好的娱乐体验。

来自: Head First Lounge

主题: 21点应用

嗨！

我们是Head First Lounge的朋友。希望你能帮我们完成一个新应用提供给访问者。

我们希望能在网站上增加一个21点应用。你能完成吗？

理想情况下，玩家点击后会发给他两张牌，他可以选择再要更多的牌。

下面是这个游戏的规则：

1. A总是最大（等于11点，而不是1点）。
2. 如果一个玩家的牌加在一起超过了21，他就输了（输了），必须重新开始。游戏结束。
3. 如果一个玩家的牌加在一起刚好是21点，他就拿到一个“21点”（blackjack），游戏结束。
4. 如果玩家的牌加在一起小于等于21点，但是已经给他发了5张牌，游戏结束，玩家获胜。

如果不满足上述任何条件，玩家可以再要一张牌。

如果满足上述某个规则/条件，结束游戏。

为玩家提供选择，他可以重置或者再玩一次。

不过我们不想要求玩家必须重新加载页面。游戏应该自行重置。

你能否帮我们完成这个应用吗？感激之至！

Head First
Lounge





Jim：嘿，你们看了Head First Lounge发来的Email吗？

Frank：是呀，看来他们希望在网站上增加一个容易上手的21点游戏。应该很简单，我觉得。

Jim：简单？要知道这可是21点！我们需要一副牌，一个发牌人，一个计数器来统计手上的牌，还有好多其他东西。你觉得我们能做到这些吗？

Joe：不会太容易，不过我认为应该办得到。就像你刚才说的，我们需要有“人”来发牌，可以编写一个函数来完成这个任务。之前我们已经写过一个随机函数，现在可以再使用这个函数。

Jim：噢，对啊……不过，那些牌呢？一副牌里可有52张牌呢。

Frank：可以建立一个大列表，每次从列表中随机选择一张。

Jim：不过，怎么避免两次选出同样的牌呢？

Frank：我想我应该知道该怎么做……

Jim：哇，太让人佩服了！那么，怎么记住已经有哪些牌呢？另外在这个过程中如何计数？

Frank：嗯，你难住我了。我还真不太确定该怎么做。

Joe：别担心。有很多JavaScript和jQuery特性可以帮助我们解决这些问题。

Jim：等一下，等一下，你说JavaScript？难道不能用变量或jQuery数组记住我们的牌吗？我原以为如果使用jQuery就不用再掺合JavaScript了呢……

Frank：变量本身可能办不到。它们一次只能存储一个值，比如一个数或者一个文本串，或者页面上的一个特定元素。jQuery数组可以保存多个值，但是只能是选择器返回的DOM元素……

Joe：没错。我们需要更灵活一点。

Frank：比如我们自己的结构或变量类型。

Joe：完全正确！所以我们需要JavaScript来创建自己的结构……

更多资源请访问稀酷客(www.ckook.com)

对象提供更聪明的存储

到目前为止，你使用的数据结构只有变量和数组。变量提供简单存储：为一个名赋一个值。数组允许更高效地存储更多数据，可以用一个变量名赋多个值。

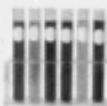
变量



当变量名赋值时，变量会记住一个值。

```
var a = 42;
```

数组



当变量名赋值时，数组会记住多个值。

```
var v = [2, 3, 4];
```

对象能提供更聪明的存储。如果需要存储关于某个特定事物的多个变量，就可以使用对象。在对象中，变量称为属性。对象还可以包含函数，可以用来与对象的属性交互。在对象中建立这样一个函数时，称之为方法。

对象



```
planeObject={  
  engines:"4",  
  type:"passenger",  
  propeller: "No"};
```

数据现在收集在一个对应飞机的分组中。

对象将数据标记为属性。



```
leopardObject={  
  num_spots:"23",  
  color:"brown"};
```

将一个属性名关联到……

……一个值。

可以使用点语法得到对象的任何属性。

```
planeObject.engines;  
leopardObject.color;
```

对象

对象的属性

需要存储关于一个特定事物的多个数据时可以使用对象。



扑克牌对象会有哪些属性？

更多资源请访问稀酷客(www.ckook.com)

建立你自己的对象

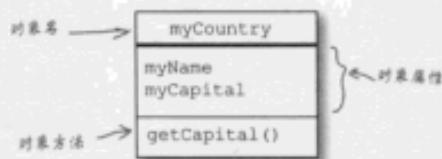
对象实际上就是一种根据需要创建定制变量的方法。可以创建一次性的对象，或者创建你自己的对象蓝图，然后反复使用。稍后会介绍可重用对象，不过现在我们先来讨论如何创建一个一次性对象，以及与对象相关的一些术语和图。

可以用一种标准方式描述对象：使用UML图。UML表示统一建模语言（Unified Modeling Language），是一种全世界通用的标准，可以用来帮助描述面向对象编程中的对象。

对象有一个与之关联的变量时，我们称之为这个对象的一个属性。如果一个对象有一个与之关联的函数，则称它为这个对象的方法。可以使用var关键字创建一次性对象，这类似于目前为止见过的所有其他变量。

对象的UML图

这个结构可以帮助你在编写代码之前查看对象的结构。



可以如下用代码写出这个对象：

```

var myCountry = {
    // 使用var创建一个名为
    // myCountry的对象。
    getCapital : function() {
        // 调用方法时会执
        // 行这个函数。
        alert(this.myCapital);
    },
    // 创建这个对象的一个方法,
    // 名为getCapital。
    // 通过大括号包围对象。
    // 设置属性值。
    myName : 'USA',
    myCapital : 'Washington DC'
}
  
```

创建这个对象的一些属性，分别
名为myName和myCapital。



实际上，jQuery和JavaScript中几乎一切都是对象。

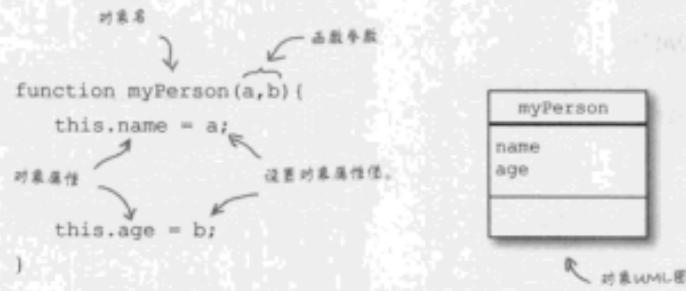
这包括元素、数组、函数、数字，甚至字符串。它们都有属性和方法。

更多资源请访问稀酷客(www.ckook.com)

用对象构造函数创建可重用的对象

对象有一个很好的方面，它们可以有相同的结构，同时有不同的属性值（或变量值）。与创建可重用的函数（见第3章）类似，可以建立一个对象蓝图，或者对象构造函数，这样就能多次使用。可以用对象构造函数来创建对象实例。

构造函数只是一个函数，所以创建对象的构造函数时，要用关键字function而不是关键字var，然后使用new关键字来具体创建对象的一个新实例。



你能想到在哪里使用这些对象吗？

更多资源请访问稀酷客(www.ckook.com)

与对象交互

对象各式各样，大小不同。实例化一个对象（创建对象的一个实例）之后，不论这是你自己创建的对象还是其他人创建的对象，都可以使用点（.）操作符与它交互。为了找点感觉，下面来仔细分析刚才定义的myCountry和myPerson对象。



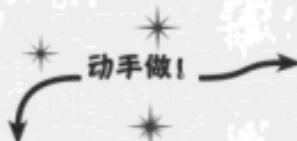
“唉，我想我知道怎么做了……可以为Head First Lounge创建一个对象表示一副扑克中的牌吗？”

对！这是个很好的想法。

下面来建立HTML页面，然后看看如何创建一个扑克牌对象。

建立页面

使用下面的信息创建你的HTML和CSS文件。不要忘记还要在scripts文件夹中创建一个my_scripts.js文件。后面几页我们还会增加大量代码。可以从<http://thinkjquery.com/chapter06/images.zip>下载这一章的所有图像资源。



```
<!DOCTYPE html>
<html>
    <head>
        <title>Head First Black Jack</title>
        <link href="styles/my_style.css" rel="stylesheet">
    </head>
    <body>
        <div id="main">
            <h1>Click to reveal your cards</h1>
            <h3 id="hdrTotal"></h3><h3 id="hdrResult"></h3>
            <div id="my_hand">
            </div>
            <div id="controls">
                <div id="btnDeal">
                    
                </div>
            </div>
        </div>
        <script src="scripts/jquery-1.6.2.min.js"></script>
        <script src="scripts/my_scripts.js"></script>
    </body>
</html>
```



index.html

```
#controls {
    clear:both;
}
#my_hand {
    clear:both;
    border: 1px solid gray;
    height: 250px;
    width: 835px;
}
h3 {
    display: inline;
    padding-right: 40px;
}
.current_hand {
    float:left;
}
```



my_style.css

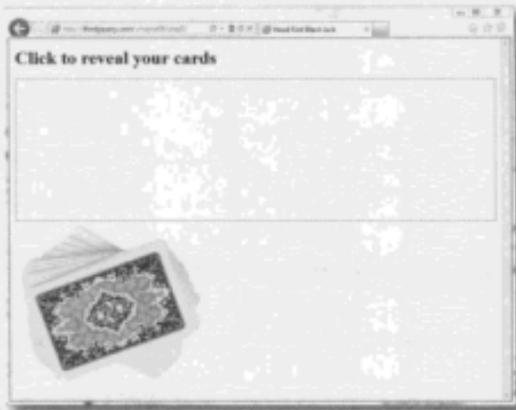


更多资源请访问稀酷客(www.ckook.com)



试一试

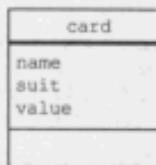
在你喜欢的浏览器中打开刚才创建的index.html页面，看看这个页面的基本结构。



Exercise

使用以下给出的扑克牌对象的UML图，创建一个可重用的对象，名为card，它有3个参数，分别名为name、suit和value。将这些参数设置为对象的属性值。这个对象没有任何方法。我们已经为你填入了一些代码。

```
function card() {  
}  
}  
})
```





Exercise Solution

这就是你的card对象。把它增加到my_scripts.js文件中，放在\$(document).ready(function(){});里。目前这个文件中只有这些代码。

使用function关键字使它可重用。

```
function card( name, suit, value ) {
    this.name = name;
    this.suit = suit;
    this.value = value;
}
```

将参数赋给对象属性。



my_scripts.js

card

name
suit
value

there are no Dumb Questions

问：一次性和可重用对象有什么区别？

答：一次性对象只是一个“有些特别”的变量，定义为可以保存多个信息。可重用对象，怎么说呢，最大的特点就是可重用。为一个可重用对象定义模板指定它的属性/方法之后，你想创建多少对象副本都可以。每个对象实例可以有不同的描述对象的信息。

问：看起来你在使用不同的方法设置属性。这样对吗？

答：没错，我们确实在使用不同的方法，这样完全没有问题。你可以使用赋值操作符 (=) 或冒号 (:) 指定属

性值，就像之前我们为对象赋值一样。这两种方法都是合法的，可以互换。这会得到同样的结果，都能访问my_property的值。

问：关于对象，还有哪些方面你没有告诉我？

答：这个问题比较难回答。对象是JavaScript中一个相当复杂的特性。在本书后面，我们还会使用JavaScript对象记法（JavaScript Object Notation, JSON）。使用JSON时，会用一个稍有不同方法访问属性：即“键”记法。这种方法也同样适用于这里的JavaScript对象。可以不这样：

object.my_property

而是这样：

object['my_property']

问：UML从哪里来？

答：UML诞生于20世纪90年代中期，当时各家公司都在努力寻找一种清晰的方法来描述对象。在那之后已经有过多次迭代。很多私有企业相互竞争，都在努力让它们的版本成为普遍接受的标准。不过，好在现在有了一个标准，使用UML的任何人都能阅读和理解其他UML来源提供的图和信息。



达么提card对象非常有用，不过我们还要另一种方法在发牌时跟踪每张牌，对吗？

你说得对。

我们需要一种方法在发牌时存储和访问这些牌。有一个好消息告诉你，之前我们已经见过怎么做到这一点……

又是我！之前在第4章
你已经用过我。

数组归来

你已经看到，可以把多个元素归组在一个称为数组的结构中。数组中的元素没有必要相互关联，不过如果数组元素相互关联，访问元素值会更为容易。还记得在第4章中，我们见过jQuery选择器如何返回元素并把它们存储在一个数组中。现在我们要使用JavaScript更充分地利用数组。

数组中的变量可以是任何类型，包括串、数字、对象，甚至可以是HTML元素！可以采用多种不同方法来创建你自己的数组：

```
使用 "new" →  
关键字创建一个空数组。  
var my_arr1 = new Array();      使用 "new" 关键字创建一个数组，同时指出数组中的值。  
                                ↓  
var my_arr2 = new Array('USA', 'China', 'Japan', 'Ireland');
```

```
var my_arr3 = ['USA', 'China', 'Japan', 'Ireland'];  
创造一个数组，通常用 "new" 关键字，若要用中括号 ([]) 也未尝为数  
不该更佳。
```

正如前面提到的，数组也是对象，这说明它们有方法和属性。有一个常用的数组属性：length，它指示数组中有多少个元素。可以用array_name.length来访问length属性。



Watch it!

方法。

创建数组的各种方式没有任何区别。
混合和搭配使用这些方式很常见，
这取决于数组的用途。在你喜欢的
搜索引擎中查找“JavaScript数组
方法”，就能找到数组对象的所有

访问数组

与创建数组不同，要访问数组中的信息只有一种方法。数组是从0开始索引的，也就是说，元素在列表中的位置（或索引）从0开始。我们在第3章用过索引，所以如果你不太记得了，可以再翻回到第3章复习一下。

```
浏览器中第一个元素的索引为0，第  
二个元素的索引为1，依此类推。  
alert( my_arr2[0] );  
// USA appears in the alert box  
  
这是数组名。  
这是我们在前面  
创造的。  
alert( my_arr3[2] );  
// Japan appears in the alert box  
  
alert( my_arr1[1] );  
// Results in an error because it is empty
```

数组使用中括号[]访问数组中
的元素。
索引访问数组时不需要加引号。
试图访问一个不存在的索引会导
致一个“未定义索引”错误。

元素在列表中的
索引是指它出现
在列表中的位
置。

OK，现在我们有了一个数组，它已经
插入了一些内容。不过是不是只能使用这
些最初放入的元素？

当然不是！

增加、修改和删除数组中的元素相当
容易。下面就来看看如何做到。



在数组中增加和更新元素

可以根据需要向数组增加更多的元素。在前一页的例子中，我们预先在my_arr2和my_arr3数组中填入了一些元素，不过my_arr1数组仍为空。可以在数组中增加或更新元素，关键还是索引。下面是更新数组的一些不同的方法：

```

    ↗          设置my_arr1数组中第一个元
    ↗          索引的值。
my_arr1[0] = "France";
alert( my_arr1[0] );
// Displays 'France' on the screen

    ↗          和my_arr2数组一样
    ↗          如果两个值。
my_arr1[1] = "Spain" ;
    ↗          更新my_arr1数组第
    ↗          一个元素的值。
my_arr1[0] = "Italy" ;
alert( my_arr1[0] );
// Displays 'Italy' on the screen

    ↗          更新my_arr3数组
    ↗          第三个元素的值。
my_arr3[2] = "Canada";
alert( my_arr3[2] );
// Displays 'Canada' on the screen

```



Exercise

在my_scripts.js文件的card对象代码后面，创建一个名为deck的数组，其中包含标准的一副扑克的52张牌。

可以使用之前创建的card对象，每次要调用构造函数并提供正确的参数来创建每一张牌，包括4个花色（黑桃、红桃、方块和梅花）从A到K，对于每张牌的值，A对应11，“2”对应2，“3”对应3，依此类推。

更多资源请访问稀酷客(www.ckook.com)



Exercise Solution

设置数组名。

为每张牌传入3个参数。

```
var deck = [
    new card('Ace', 'Hearts', 1),
    new card('Two', 'Hearts', 2),
    new card('Three', 'Hearts', 3),
    new card('Four', 'Hearts', 4),
```

```
new card('King', 'Hearts', 10),
new card('Ace', 'Diamonds', 1),
new card('Two', 'Diamonds', 2),
new card('Three', 'Diamonds', 3),
```

```
new card('Queen', 'Diamonds', 10),
new card('King', 'Diamonds', 10),
new card('Ace', 'Clubs', 11),
new card('Two', 'Clubs', 2),
```

```
new card('King', 'Clubs', 10),
new card('Ace', 'Spades', 11),
new card('Two', 'Spades', 2),
new card('Three', 'Spades', 3),
```

```
new card('Jack', 'Spades', 10),
new card('Queen', 'Spades', 10),
new card('King', 'Spades', 10)
```

}

记住是花括号中的值

不是空格



不过，现在数组中有许多牌。看来
我们要写大量代码再把它们取出来。
太让人头疼了！

没有必要。

我们还是要利用元素的索引访问各个元素，不过可以使用一种类似each的技术依次访问各个元素（这在第3章已经见过），而无需为每一张牌编写大量代码。

现在该看看循环了……



反复（再反复……）地完成动作

这个21点游戏中你要经常在数组中放入牌，还要从数组中取出牌。好在JavaScript已经对这种情况做好了准备，提供了循环特性。还有一个更好的消息，你之前实际上已经用过循环，还记得在第3章中，我们曾经用jQuery的each方法根据jQuery选择器来循环处理元素。不过，这一次我们会有很多选择，因为JavaScript提供了很多不同类型的循环，每种循环的话语稍有不同，分别有各自不同的用途。

for循环非常适合将代码重复指定的次数。开始循环之前你应该知道这个次数，否则它会一直循环下去。for循环根据变量值可以运行0或多次。

for循环周期：



for循环允许将代码重复指定的次数。

每一种循环（不论是哪种类型）都有4个部分：

- ① 初始化
这会在循环刚开始时发生一次。
- ② 测试条件
检查循环应当停止还是继续下一轮运行代码，通常会检查某个变量的值。
- ③ 动作
这是每次进入循环反复执行的代码。
- ④ 更新
这部分更新测试条件中使用的变量，这个变量要用来查看是否继续循环。

do…while循环会运行一次代码，然后不断运行相同的代码，直至满足某个特定的条件，如一个值从true变为false（或者反之），或者代码中计数达到某个特定的数。do…while循环可以运行一次或多次。

do…while 循环周期：



do…while循环会运行一次代码，然后重复运行直至满足某个特定的条件。

我最喜欢反复做同样
的事情！

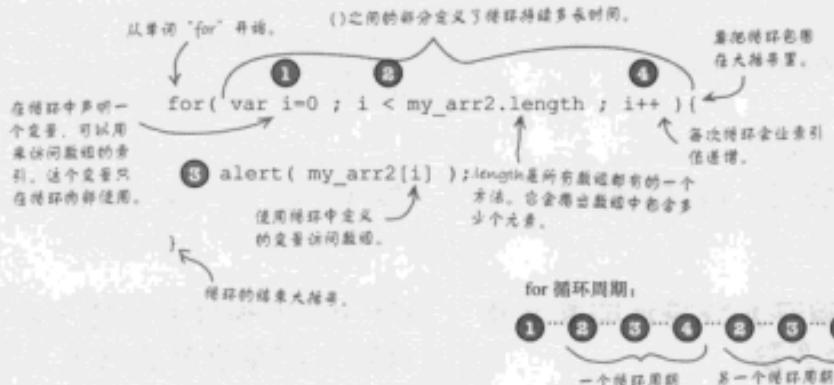




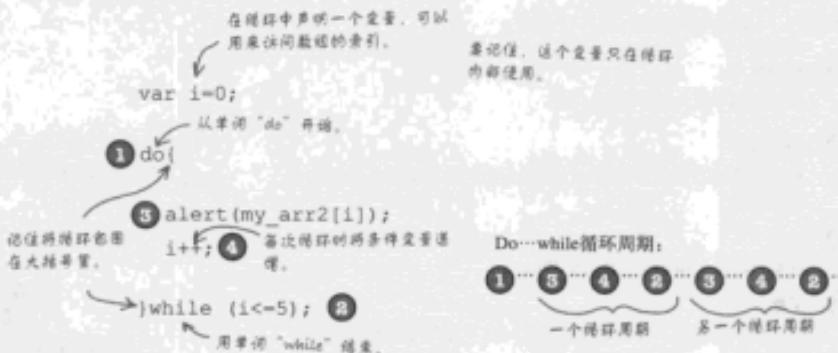
循环特写

仔细查看前面提到的各种不同的循环类型，可以看到它们都包含4个主要部分，不过顺序稍有不同。这个顺序反映了不同循环类型之间的主要差别。

for循环



do...while循环





Frank: 嗯，我们已经有了card对象数组，不过还需要在发牌时随机地抽出一张牌，对吗？

Joe: 是的，好在我们已经在第3章写过一个getRandom函数。这个函数每次可以提供一个随机数，从数组中抽出随机的一张牌。

Jim: 不过然后怎么处理这张牌呢？

Frank: 我们要把它记住。必须能够累加牌的总点数，查看玩家是否超过了21点。

Joe: 记住牌还有一个原因。我们不能两次给出同一张牌，所以还必须确保这张牌之前没有发过。

Jim: 可以用一个变量来记住牌吗？

Frank: 我们可以使用一个数组变量……

Joe: 想法很好！我们甚至不用把这些牌存储起来，只需要存储它们的索引值，这样就能测试它是否在我们的used_cards数组中。

Jim: 哇，太震撼了！怎么判断一个值是否在一个数组中呢？

Frank: 可以使用一个jQuery实用方法：inArray。

Joe: 听起来很方便。不过我想可能需要很多函数来完成这个任务。首先要得到一个介于0到51之间的随机数，还必须检查之前是否已经用过这个数。如果用过，就要再次尝试。如果没有用过，则需要从一副牌中取出这张牌，并记住这张牌的索引。然后还要把这张牌显示给玩家。

Jim: 听上去工作很多啊！怎么向玩家显示扑克牌呢？

Frank: 嗯，我们已经有图像，并已经按花色和命名类型来组织，所以可以使用card对象的这些属性在屏幕上放置图像。

Joe: 完全正确。我们可以创建一个DOM元素，把它追加到页面上已经有的my_hand div元素。

Frank: card对象已经开始施展才能了……下面就开始吧！

于草堆里找根针

经常需要查看一个变量是否已经在数组中，以防重复数据，或者要避免代码向数组中多次增加同样的元素。如果使用数组存储购物车或者愿望列表之类的东西，这尤其有用。

为于草堆里的东西创建一个数组。

不过如果我们想找出这个值在 haystack 数组中的位置呢？

```
var haystack = new Array('hay', 'mouse', 'needle', 'pitchfork')
```

jQuery提供了很多实用方法来帮助我们更高效地完成一些特定的任务。这包括大量函数，有些用来检查网站访问者使用的浏览器类型，有些可以返回当前时间，有些用来合并数组，还有一些可以从数组中删除重复值。

对于这里的特殊情况，我们可以使用实用方法 `inArray`。它会返回你要查找的值在数组中的位置（索引），当然前提是数组中存在这个值。如果数组中找不到这个值，就会返回 -1。与其他实用方法一样，`inArray` 不需要选择器。它由 jQuery 函数或 jQuery 快捷方式直接调用。

创建一个变量来保存你想要找的值。

你要查找的值。

是在这个数组中

查找。

```
var index = $.inArray( value, array );
```

jQuery 快捷方式。

`inArray` 需用方法调用。

```
var needle_index = $.inArray( 'needle', haystack );
```

这是你要查找的值。

是在这个数组中

搜索。



21点应用的哪个特性需要检查是否已经用过某个值？



jQuery 21点应用代码磁贴

组织下面的磁贴，写出以下函数的代码，这些函数可以帮助你完成这个21点游戏。完成后的代码将创建两个函数deal和hit，还会为ID为btnDeal的元素创建一个点击事件监听者，另外会创建一个新的数组变量，名为used_cards，用来记住已经发过哪些牌。

```

var used_cards = new _____();
function _____() {
    for(var i=0;i<2;i++){
        hit();
    }
}
function getRandom(num) {
    var my_num = Math.floor(_____ * num);
    return my_num;
}
function _____() {
    var good_card = false;
    do{
        var index = _____(52);
        if( !$inArray(index, _____) > -1 ){
            good_card = true;
            var c = deck[ index ];
            _____[used_cards.length] = index;
            hand.cards[hand.cards.length] = c;
            var $d = $(<div>);
            $d.addClass("current_hand")
            .appendTo(_____);
            $(<img>).appendTo($d)
            .attr(_____, 'images/cards/' + c.suit + '/' + c.name + '.jpg')
            .fadeOut('slow')
            .fadeIn('slow');
        }
    }_____(!good_card);
    good_card = false;
}
$("#btnDeal").click(_____() {
    deal();
    $(this).toggle();
});

```

getRandom

deal()

used_cards

'src'

hit()

while

Array

#my_hand

function

used_cards

Math.random()



my_scripts.js

更多资源请访问稀酷客(www.ckook.com)



jQuery 21点应用代码磁贴答案

下面就是deal和hit函数的代码，以及ID为btnDeal的元素的点击事件监听者，另外还有一个新的数据变量，名为used_cards，用来记住已经发过哪些牌。

```

var used_cards = new Array(); // 创建一个数组，来保存用过的扑克牌。
function deal() {
    for(var i=0;i<2;i++) {
        hit(); // 用一个for循环调用hit函数两次。
    }
}
// 这是getRandom函数：
function getRandom(num) {
    var my_num = Math.floor(Math.random() * num);
    return my_num;
}
function hit() {
    var good_card = false;
    do{
        var index = getRandom(52); // 使用inArray函数，检查这张牌是否已经被选过。
        if(index > -1){ // 条件变量。
            good_card = true;
            var c = deck[index]; // 从deck数组中取牌。
            used_cards[used_cards.length] = index; // 将这张牌的数据添加到used_cards数组。
            hand.cards(hand.cards.length) = c; // 将这张牌的数据添加到hand数组。
            $d = $(<div>); // 使用card对象的属性建立指向图像的路径。
            $d.addClass("current_hand")
            .appendTo("#my_hand");
            $("<img>").append($d)
            .attr("src", "images/cards/" + c.suit + "/" + c.name + ".jpg");
            .fadeOut('slow');
            .fadeIn('slow');
        }
    }while(!good_card); // 如果这张牌已经被用过，就再试一次。
}
$("#btnDeal").click(function() {
    deal(); // 点击时调用deal函数。
    $(this).toggle();
});

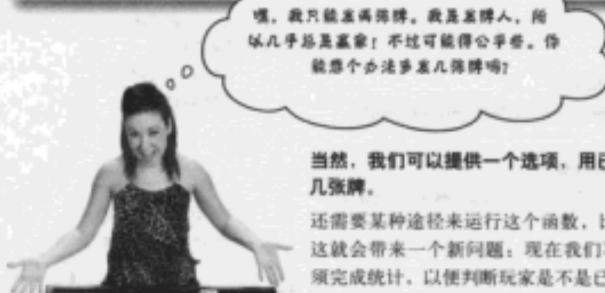
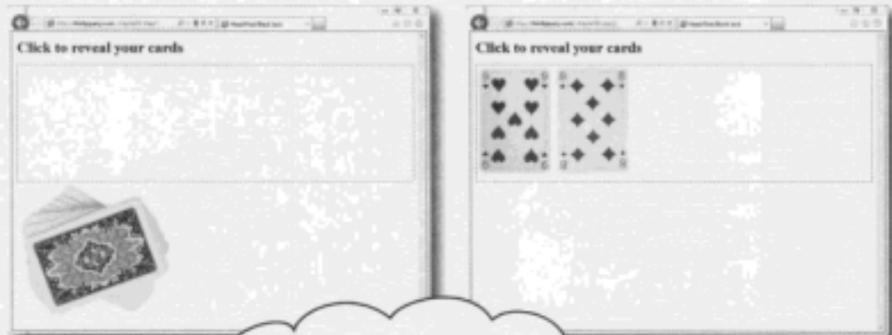
```





试一试

把上一个磁点练习中的所有代码增加到my_scripts.js文件，放在deck数组后面。在你的浏览器中试一试。点击这副牌，发一手牌。



当然，我们可以提供一个选项，用已经创建的hit函数从一副牌中多取几张牌。

还需要某种途径来运行这个函数，比如点击一个按钮或者类似的方式。这就会带来一个新问题：现在我们不仅要记住已经发过哪些牌，还必须完成统计，以便判断玩家是不是已经超了。



你能想出用什么来记住所有这些不同的信息吗？



更多资源请访问稀酷客(www.ckook.com)



成品 HTML & CSS

由于你在样式和结构方面已经是专家了，我们只给出index.html和my_style.css文件的更新代码，以便你比较。加入这些新的HTML和CSS代码之后，你会看到页面的一些变化。稍后我们会把所有部分连接起来。

```
<!DOCTYPE html>
<html>
<head>
    <title>Head First Black Jack</title>
    <link href="styles/my_style.css" rel="stylesheet">
</head>
<body>
    <div id="main">
        <h1>Click to reveal your cards</h1>
        <h3 id="hdrTotal"></h3>
        <h3 id="hdrResult"></h3>
        <div id="my_hand">
        </div>
        <div id="controls">          为新增按钮增加一些
            <div id="btnDeal">          CSS。
                
            </div>
            <div id="btnHit">
                
            </div>
            <div id="btnStick">
                
            </div>
        </div>
        <div id="result">          为21点游戏增加一些
            <div id="msg">          新控件。
                <h3>You're out! Better luck next time!</h3>
            </div>
            <div id="msg2">
                <h3>Great hand! You win!</h3>
            </div>
        </div>
        <script src="scripts/jquery-1.6.2.min.js"></script>
        <script src="scripts/my_scripts.js"></script>
    </div>
</body>
</html>
```

```
#controls{
    clear:both;
}

.current_hand{
    float:left;
}

#my_hand{
    clear:both;
    border: 1px solid gray;
    height: 250px;
    width: 835px;
}

#btnHit, #btnStick,
#btnRestart{
    display:none;
    float:left;
}
```

```
h3 {
    display: inline;
    padding-right: 40px;
}
```



my_style.css



index.html

更多资源请访问稀酷客(www.ckook.com)

there are no
Dumb Questions

问：还有没有其他类型的循环需要了解？

答：是的，确实还有其他类型的循环，有一个while循环，它与do...while循环非常相似，只不过会在开始时进行条件检查。另外还有一个for……in循环，这会循环处理一个对象的属性，并取出各个属性的值。

问：好吧，我已经开始了一个循环，能在中间停止循环吗？

答：可以，只需要使用一个简单的命令：break。在循环中的某个地方调用这个命令就会让循环停止，转而执行循环后的下一段代码。

问：appendTo是什么？我以及只见过append。有什么不同吗？

答：调用append方法的选择器是容器，内容将插入到这个容器中。appendTo则不同，内容写在这个方法前面，而不是作为一个选择器表达式或者作为动态创建的HTML标记，它将插入到目标容器。



Exercise

使用下面给出的UML图，创建一个一次性对象，名为hand。cards属性应当是一个新的空数组。current_total属性要设置为0。sumCardTotal方法应循环处理cards属性中的所有牌，把它们的值累加在一起，并将这个数设置为current_total属性的值。然后，使用current_total值设置ID为hdrtotal的元素的值。我们已经开了头，为你写出了部分对象代码。

```
var hand = {
    cards : new Array(),
    current_total : 0,
    sumCardTotal: function(){
        }
}
```

hand
cards
current_total
sumCardTotal()



Exercise Solution

现在你已经有了一个hand对象，它包括一个card属性（这是一个数组），还有一个函数可以循环处理card数组、得到当前牌并更新当前总点数。

```
var hand = {
  cards : new Array(),
  current_total : 0, ← 将current_total
  sumCardTotal: function() { ← 属性设置为0。
    this.current_total = 0; ← 循环处理card
    for(var i=0;i<this.cards.length;i++){ ← 数组。
      var c = this.cards[i]; ← 从数组得到它的牌。
      this.current_total += c.value; ← 将这个值增加到
    }                                current_total。
    $( "#hdrtotal").html( "Total: " + this.current_total );
  }
}
```

hand
cards
current_total
sumCardTotal()

不过我看不出怎么能说明是不是赢了。难道我得掌定所有的牌吗？



我们真不想这么做。

这样没有办法看出谁赢了。根据Head First Lounge邮件中给出的规则，我们要根据几个不同条件来确定输赢。下面再来看看这些规则。

- ① 如果玩家的牌加在一起超过了21点，她就超了，必须重新开始，游戏结束。
- ② 如果玩家的牌加在一起刚好是21点，她就拿到一个“21点”，游戏结束。
- ③ 如果玩家的牌加在一起小于等于21点，但是已经给她发过5张牌，游戏结束，她赢了。
- ④ 否则，玩家可以选择再要一张牌或者停止游戏。

又到了……判断时间！

在前面的第3章中，我们已经了解到，给定已有的信息，可以使用条件逻辑根据你希望代码做出的判断来运行不同的代码。

```
开始对话吧。
if( myBool == true ) { 检查的条件
    javascript变量 // Do Something!
} else{           相等。这可以读作“等于”。
    // Otherwise Do something else!
}
```

这是检查结果为true时将要运行的代码。

实际上，还有一种选择，可以一次做多个判断。通过把if和else语句结合到一个称为else if的复合语句中，就可以在一个语句中检查多个条件。下面就来看一看。

```
我们想要检查的
条件。
if( myNumber < 10 ){
    // Do Something!
} else if( myNumber > 20 ){ 检查的另一个
    // Do something else! 条件。
} else{
    // Finally, Do something even different!
}
```

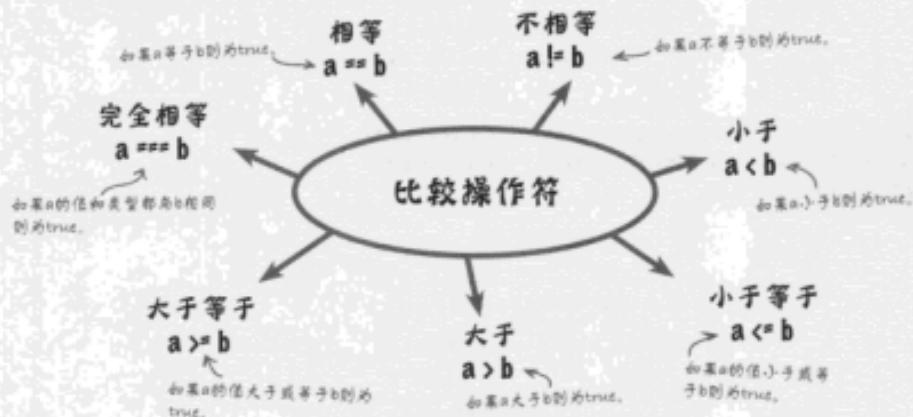


你认为代码中哪里可以使用if / else if / else语句？

更多资源请访问稀酷客(www.ckook.com)

比较和逻辑操作符

要让条件逻辑语句（如if/else或do…while）正常工作，它们必须能根据所检查的条件做出正确的判断。为此，条件逻辑语句要使用一系列比较和逻辑操作符来帮助做出判断。在JavaScript中，共有7个不同的比较操作符和3个逻辑操作符。另外还有一个对应if/else语句的简写操作符，称为三元操作符。我们已经见过这样一些比较和逻辑操作符，不过下面将完整地给出。



逻辑操作符

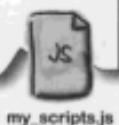


更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

更新hand对象，检查current_total属性的值是否满足游戏规则（如果你记不清全部规则，可以翻到本章最前面查看之前收到的Email）。下面是原来的对象。另外给出了你需要编写的部分新代码。

```
var hand = {
    cards : new Array(),
    current_total : 0,
    sumCardTotal: function(){
        this.current_total = 0;
        for(var i=0;i<this.cards.length;i++){
            var c = this.cards[i];
            this.current_total += c.value;
        }
        $("#hdrTotal").html("Total: " + this.current_total );
        if(this._____ > 21){
            $("#btnStick").trigger("click");
            $("#hdrResult").html("BUST!");
        } _____ (this.current_total _____ ){
            $("#btnStick").trigger("click");
            $("#hdrResult").html("BlackJack!");
        } else if( _____ .current_total _____ 21 _____ this.cards.length == 5){
            $("#btnStick").trigger("click");
            $("#hdrResult").html("5 card trick!");
        } _____
            // Keep playing! :)
        }
    }
};
```



my_scripts.js

Sharpen your pencil

Solution

现在你已经更新了sumCardTotal方法，其中包含相应逻辑来检查已发的一手牌的值。尽管这只是一个非常简单的应用，但也用到了大量条件和逻辑操作符。

```

var hand = (
    cards : new Array(),
    current_total : 0,
    sumCardTotal: function(){
        this.current_total = 0;
        for(var i=0;i<this.cards.length;i++){
            var c = this.cards[i];
            this.current_total += c.value;
        }
        $("#hdrTotal").html("Total: " + this.current_total );
        if(this.current_total > 21){ ← 检查current_total是否大于21。
            $("#btnStick").trigger("click");
            $("#hdrResult").html("BUST!");
        }else if(this.current_total == 21){ ← 检查current_total是否等于21。
            $("#btnStick").trigger("click");
            $("#hdrResult").html("BlackJack!");
        }else if(this.current_total <= 21 && this.cards.length == 5){
            $("#btnStick").trigger("click");
            $("#hdrResult").html("BlackJack - 5 card trick!");
        }else{ ← 检查current_total是否小于等于21，另外查看是否超过了5张牌。
            // Keep playing!
        }
    }
);

```



my_scripts.js



不过我们还是不能执行这些新函数，因为没有对象来调用它们，对吧？

是的，我们还没有完成呢。

你的HTML代码已经包含了所需的全部内容，包括最初的发牌，再要一张牌，以及结束游戏。不过还没有把它们连起来。另外不要忘了，每次发一张新牌时，都需要调用方法累加牌的总点数。



jQuery 21点游戏代码磁贴

移动下面的磁贴，写出一个函数的代码，这个函数将为21点应用增加多个事件监听者。这些监听者要放在ID为btnHit和btnStick的元素上，btnHit事件要发另一张牌。另一个事件则应停止游戏。另外，发牌后要调用sumCardTotal方法。这里还给出了hit函数末尾的一些代码，以便你更新。

```

    }while(!_____);
good_card = false;
hand._____();
}

$("#btnDeal").click(_____){
deal();
$(this).toggle();
$("#btnHit")._____();
$("#btnStick").toggle();
});

$("#_____").click( function(){
hit();
});

$("#btnStick").click( function(){
$( "#hdrResult" ).html(_____);
});
}

```



my_scripts.js

function

toggle

good_card

"Stick!"

#btnHit

sumCardTotal

更多资源请访问稀酷客(www.ckook.com)



jQuery 21点游戏代码磁贴答案

这个代码创建的函数会在hit和stick按钮上增加一些事件监听者。另外每次发牌后会调用sumCardTotal方法。

直接开始初始化。

最后一张牌。

```

while(!_good_card); // 不断循环，直至找到一张好牌。
good_card = false;
hand.sumCardTotal(); // 在hit函数的末尾得到一手牌的总点数。

$("#btnDeal").click(function() {
    deal();
    $(this).toggle();
    $("#btnHit").toggle();
    $("#btnStick").toggle();
});

$("#btnHit").click(function() {
    hit();
}); // 弹出消息设置为“Stick!”。

$("#btnStick").click(function() {
    $("#hdrResult").html('Stick!');
});

```



my_scripts.js

there are no
Dumb Questions

问：JavaScript中有没有其他办法来完成值的比较？

问：你说 if/else语句还有一种简写形式，是什么？

答：事实上，完成值的比较别无其他方法，不过还有一个方法可以根据变量的值做出判断。这个方法名为switch。可以有多个不同的条件。通常情况下，如果你发现要写很庞大的if / else if / else语句，就可以使用switch语句。这样可能更为容易。

答：这称为三元操作符，它用一个问号将逻辑操作和结果动作分开，如下：

`a > b ? if_true_code : if_false_code`



试一试

将刚才创建的所有代码增加到my_scripts.js文件，放在hit函数后面。别忘了hit函数本身的最后部分也要更新，然后在你喜欢的浏览器中整体浏览看看。



Frank：还没那么快。我们还要增加他们要求的重置特性。这样一旦游戏结束，玩家可以重新开始而不用重新加载页面。

Joe：另外我们还要确保人们不会拿到以前游戏发过的同样的牌，要保证清除之前的一切信息。

Jim：不过怎么做到呢？我们动态增加了HTML元素，另外数组中还增加了新的数据元素。要把它们都清除吗？

Frank：是的。我们要分别使用稍有不同的技术。不过，你说的没错，我们必须把它们都清除。

Joe：我知道一些小技巧！对于jQuery，可以使用empty方法来清空。对于数组，也有很多方法，不过并不是所有方法都是跨浏览器的。下面来看最好的办法是什么。

用jQuery清空……

第2章中，我们曾使用jQuery `remove`方法从DOM清除一个特定的元素和它的所有子元素，然后它们再也看不见了，还记得吧？如果你想删除父元素，这个方法就非常合适。不过，如果你想保留主元素，只清空它的内容，那么可以使用jQuery `empty`方法。与`remove`类似，这需要一个选择器，不过会保留调用这个方法的元素。

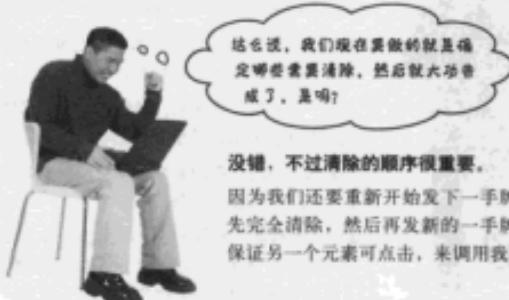
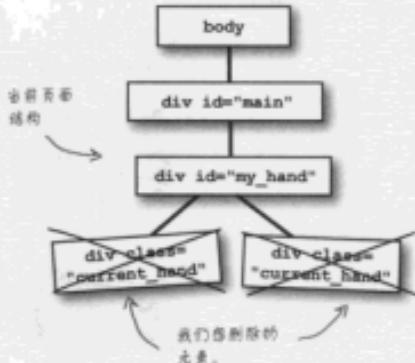
……用JavaScript更容易

通常我们可能更愿意编写jQuery代码而不想写很多行JavaScript代码。好在，有些情况下用JavaScript与使用jQuery同样容易。这里就属于这种情况。尽管语法稍有不同，但最终结果是一样的，你不用跟踪目前在DOM中的位置。要用JavaScript真正清空一个数组，只需要将它的长度设置为0：

```
used_cards.length = 0;
```

没有比这更容易的了，是吧？

```
$("#my_hand").empty();
```



没错，不过清除的顺序很重要。

因为我们还要重新开始发一手牌，所以应当先完全清除，然后再发新的一手牌。另外还要保证另一个元素可点击，来调用我们的代码。

Sharpen your pencil

更新index.html文件，增加一个元素（类似于controls div元素中的所有其他元素）。这个可点击div的ID指定为btnRestart。在这个div中，放置一个图像，来源为images文件夹中的restart_small.jpg。

另外，更新my_scripts.js文件，为btnRestart元素增加一个点击事件监听者。这会清空my_hand元素、used_cards数组和hand对象中的cards数组。另外还会切换一个新div元素（ID为result）和它自身（btnRestart）的状态。它还会清空hdrResult元素的html。最后，还要切换并触发btnDeal元素的点击事件。

```
<div id="btnStick">
  
</div>
```



index.html

```
<div id="_____><img src="" id="imgResult">_____
</div>
</div>
<script src="scripts/jquery-1.6.2.min.js"></script>
```

```
$( "#hdrResult" ).html('Stick!');
})
$( "#btnRestart" ).click( function() {
  _____ .toggle();
  $(this) ._____
  $( "#my_hand" )_____
  $( "#hdrResult" ).html('');
  used_cards._____ = 0;
  _____ .length = 0;
  hand._____ = 0;

  $( "#btnDeal" ).toggle()
  _____ ('click');
}) ;
```



my_scripts.js

Sharpen your pencil

Solution

现在你有了一个重置按钮，可以重新开始游戏了，它会把所有元素重置回游戏开始之前的状态。利用length属性还可以增加一点JavaScript魔法，大功告成。

```
<div id="btnStick">
  
</div>          ↗ 这是一个重置按钮用来重
<div id="btnRestart">      新开始游戏。
  
</div>
<div id="result"><img src="" id="imgResult"></div>
</div>          ↗ 使结果更明显。
</div>
<script src="scripts/jquery-1.6.2.min.js"></script>
```



index.html

```
$( "#hdrResult" ).html('Stick!');
$( "result" ).toggle();
});

$( "#btnRestart" ).click( function() {
  $( "#result" ).toggle();
  $( this ).toggle();
  $( "#my_hand" ).empty();
  $( "#hdrResult" ).html('');
  used_cards.length = 0;
  hand.cards.length = 0;
  hand.current_total = 0;
  ↗ 模拟btnDeal元素
  ↗ 的一次点击。
  $( "#btnDeal" ).trigger('click');
});
```

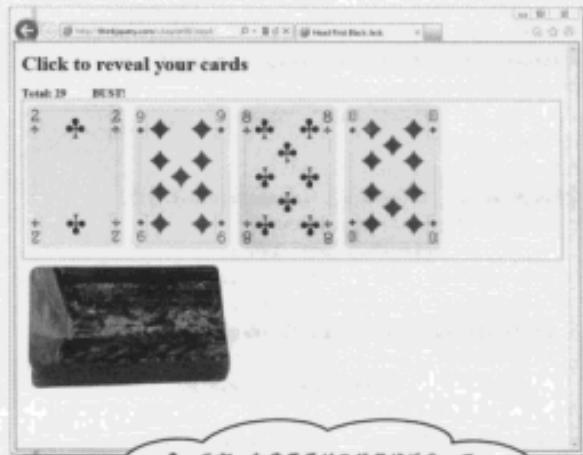


my_scripts.js



试一试

在my_scripts.js文件中为btnRestart元素增加点击事件。另外，不要忘记在index.html文件中增加补充的HTML代码。



啧。天啊。和我想要的已经很接近了。不过至少一样：能不能让输赢结果更明显一些，或者更让人兴奋呢？实在不好意思，总是麻烦你。



更多资源请访问稀酷客(www.ckook.com)

增加兴奋点

用一个新的end函数更新my_scripts.js函数，这个函数将由btnStick调用。另外要对sumCardTotal中的计算逻辑做些更新。可以在这里找到最新的my_style.css文件：
http://thinkjquery.com/chapter06/end/styles/my_style.css

动手做！

```

if(this.current_total > 21){
    $("#btnStick").trigger("click");
    $("#imgResult").attr('src','images/x2.png'); ← 根据结果将imgResult的
    $("#hdrResult").html("BUST!");   attr设置为不同的图像。
    .attr('class', 'lose');
} else if(this.current_total == 21){
    $("#btnStick").trigger("click");
    $("#imgResult").attr('src','images/check.png'); ← 根据结果设置不同的类。
    $("#hdrResult").html("BlackJack!");
    .attr('class', 'win');
} else if(this.current_total <= 21 && this.cards.length == 5){
    $("#btnStick").trigger("click");
    $("#imgResult").attr('src','images/check.png');
    $("#hdrResult").html("BlackJack - 5 card trick!");
    .attr('class', 'win');
} else{
    $("#hdrTotal").html("Total: " + this.current_total );
}

};

function end(){
    $("#btnHit").toggle();
    $("#btnStick").toggle(); } 切换所有控件的状态，结束
    $("#btnRestart").toggle(); 游戏。
}

$("#btnStick").click( function(){
    $("#hdrResult").html('Stick!');
    .attr('class', 'win');

    $("#result").toggle();
    end(); ← 在stick之后调用end函数结束游戏。
});
```



my_scripts.js

更多资源请访问稀酷客(www.ckook.com)



试一试

更新my_scripts.js文件中hand函数的sumCardTotal方法。另外，不要忘记找到最新的my_style.css文件，用这个新版本替换替换你当前的版本。

The image is a composite of several elements. At the bottom left, a woman in a black dress stands behind a roulette table, smiling. A thought bubble from her contains Chinese text: "太棒了！实在太完美了：现在Head First Lounge的访问者到来时可以享有一个绝妙的21点游戏了。" Above her are two screenshots of a blackjack game interface. Both screenshots show a hand total of 21. The top screenshot shows a hand with cards Ace, 8 of diamonds, and 2 of diamonds, with a dealer's hand partially visible. The bottom screenshot shows a hand with cards Ace, 8 of diamonds, and 2 of diamonds, with a dealer's hand showing a 2 of clubs. Below the screenshots is a large, bold, blue banner with white text that reads "更多资源请访问稀酷客(www.ckook.com)".



jQuery/JavaScript工具箱

你已经掌握了第6章，现在你的工具箱里又增加了JavaScript对象、数组和循环。

JavaScript对象

创建单个对象和创建构造函数
使用对象和调用构造函数

数组

创建数组
为数组赋值
为数组增加更多元素
更新现有数组元素

循环

for循环
do...while循环
逻辑操作符
比较操作符

jquery

.empty
.inArray —实用方法
.attr
.trigger

更多资源请访问稀酷客(www.ckook.com)

7 定制函数提供定制效果



结合jQuery的定制效果和JavaScript函数，可以让你的代码和Web应用更高效、更有用、更强大。这一章中，将通过处理浏览器事件、使用定时函数，以及优化定制JavaScript函数的组织和可重用性，进一步改进jQuery效果。

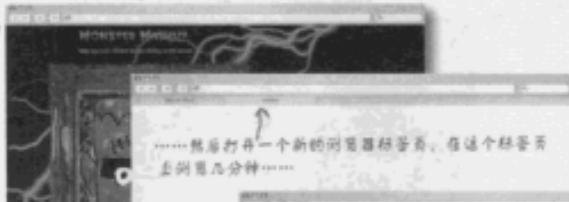
暴风雨来了

你在第5章构建的怪兽变脸应用很受孩子们和他们父母的欢迎。不过，听说好像有一个bug，应用中的闪电有些毛病。DoodleStuff的质量保证经理与你联系，指出了一些问题，还提出一个特性需求。希望能让这个怪兽变脸应用更出色。

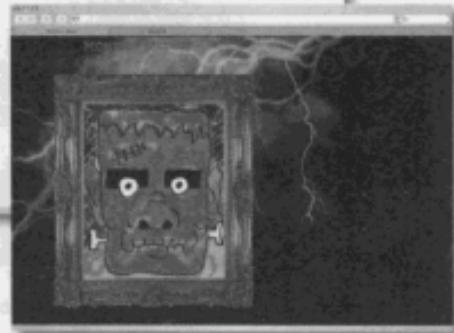
Jill，开着DoodleStuff
的质量保证。

访问者启动怪兽变脸应
用……

我们发现，如果用户在他的浏览器中打开一个新的标签页，同时仍在另一个标签页中打开怪兽变脸应用。
等他再返回到怪兽变脸应用时，闪电会迅速消失，中间毫元
停顿。看起来应用好像有毛病了！



……然后返回怪兽变脸应用的
标签页，闪电会迅速消失，就好像
饭菜煮熟一样。



尝试让这个问题再次出现。然后考虑闪电函数哪里出了问题。
为什么人们从一个标签页切换到另一个标签页时会相互冲突？

我们创建了一个怪兽……函数

第5章创建的闪电函数确实有些问题。它会一直运行，即使用户离开这个页面仍会继续运行。用户返回到这个标签页时，定时器不得不“赶时间”，尝试在屏幕上飞快地重绘闪电。看起来定时器没有像我们希望的那样工作，那么到底发生了什么？

```
function lightning_one(t) {
    $("#lightning1").fadeIn(250).fadeOut(250);
    setTimeout("lightning_one()", t);
```

超时时间为毫秒为单位。

这会告诉JS解释器，这个函数会反复地调用自身。

在JavaScript中，通常会定义一个函数，然后在代码中的另外某个位置调用这个函数。在这里，则是从函数本身调用。

setTimeout方法告诉JS解释器运行一个函数，并在再次运行之前等待一段时间。

第5章中，我们需要反复地调用这个方法，每次调用之间有一个超时时间间隔。解决这个问题时，我们无意识地引入了一个新的问题：窗口失去访问者的焦点时（也就是说，访问者打开一个新的标签页，离开了当前活动窗口），这个函数仍在继续运行。

函数会一直执行下去，失去控制？这太复杂，太可怕了！怎么才能重新施加控制？



Watch it!

对调用自身的函数一定要当心。

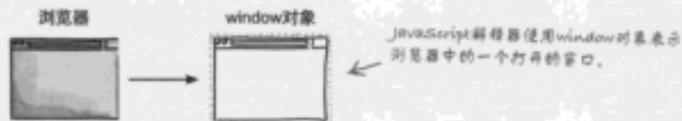
产生无限循环可能会耗尽CPU资源，使访问者的浏览器崩溃。



更多资源请访问稀酷客(www.ckook.com)

用window对象控制定时效果

幸运的是，可以使用JavaScript的window对象控制闪电动画。每次访问者在浏览器中打开一个新窗口时，就会创建一个window对象，它能提供jQuery和JavaScript的大量强大功能。在JavaScript世界里，window对象是统领全局的对象。换句话说，window是JavaScript世界中最高层次的对象。



假设你在浏览器中打开了3个标签页，浏览器会为各个标签页分别创建一个window对象。window对象与第6章使用的其他对象类似，它也有属性、事件处理器程序和方法，而且极其方便，我们可以使用window对象的onblur和onfocus事件处理程序来了解访问者正在浏览器上做什么。



Window对象还提供了定时器方法，可以利用这些方法运行定制的定时函数。window还有很多其他方法，不过修正闪电函数只需要用到上述方法。



Watch it!

不要把JavaScript window对象的onblur和onfocus事件处理程序与jQuery的blur和focus方法混淆。

jQuery blur和focus方法关联到HTML表单域和其他元素，而不是window对象。

将window对象的各个属性、事件处理程序或方法与相应作用连线。

window.name

检测窗口何时接收到一个点击、键盘输入或其他类型的输入。

window.history

这是window对象的一个属性，指示所加载文档的主要内容。

window.document

检测窗口何时失去焦点。

window.onfocus

这是window对象的一个方法，用于设置调用一个函数或其他语句之前所等待的时间。

window.setTimeout()

这是window对象的一个方法，用于取消重复调用之间等待的时间。

window.clearTimeout()

这是window对象的一个方法，用于设置重复函数调用或其他语句之间需要等待的时间。

window.setInterval()

这是window对象的一个属性，可以用来访问一段时间以来窗口加载的不同URL。

window.clearInterval()

这是window对象的一个方法，用于取消需要等待的时间。

window.onblur

这是window对象的一个属性，可以用来访问或设置窗口名。



WHO DOES WHAT?

答案

window.name

检测窗口何时接收到一个点击、键盘输入或其他类型的输入。

window.history

这是window对象的一个属性，指示所加载文档的主要内容。

window.document

检测窗口何时失去焦点。

window.onfocus

这是window对象的一个方法，用于设置调用一个函数或其他语句之前所等待的时间。

window.setTimeout()

这是window对象的一个方法，用于取消重复调用之间等待的时间。

window.clearTimeout()

这是window对象的一个方法，用于设置重复函数调用或其他语句之间需要等待的时间。

window.setInterval()

这是window对象的一个属性。可以用来访问一段时间以来窗口加载的不同URL。

window.clearInterval()

这是window对象的一个方法，用于取消需要等待的时间。

window.onblur

这是window对象的一个属性，可以用来访问或设置窗口名。



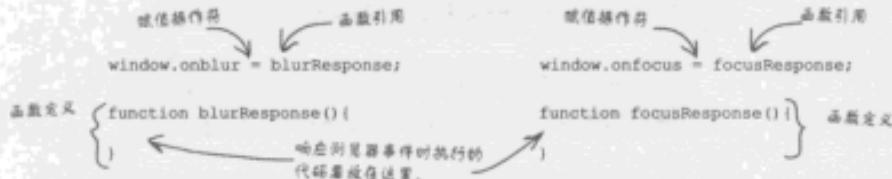
BRAIN
POWER

Window对象的onfocus和onblur事件处理程序可以检测到窗口焦点的改变，不过如何对这些事件做出响应呢？

更多资源请访问稀酷客(www.ckook.com)

用onblur和onfocus响应浏览器事件

我们知道，利用window.onfocus可以判断窗口何时得到焦点（也就是访问者激活了页面，或者访问者对窗口有鼠标或键盘输入），利用window.onblur可以判断活动的浏览器窗口何时失去焦点。不过如何对这些事件做出响应呢？可以为onfocus或onblur指定一个函数引用。



这里才逐步显示出编写定制函数的强大威力。现在有了一个window对象，它能提供大量信息，告诉你用户在浏览器中做什么，可以根据这个对象告诉你的信息指定你自己的定制函数。所以，实际上，你想做什么都可以，只要能编写相应的定制函数……



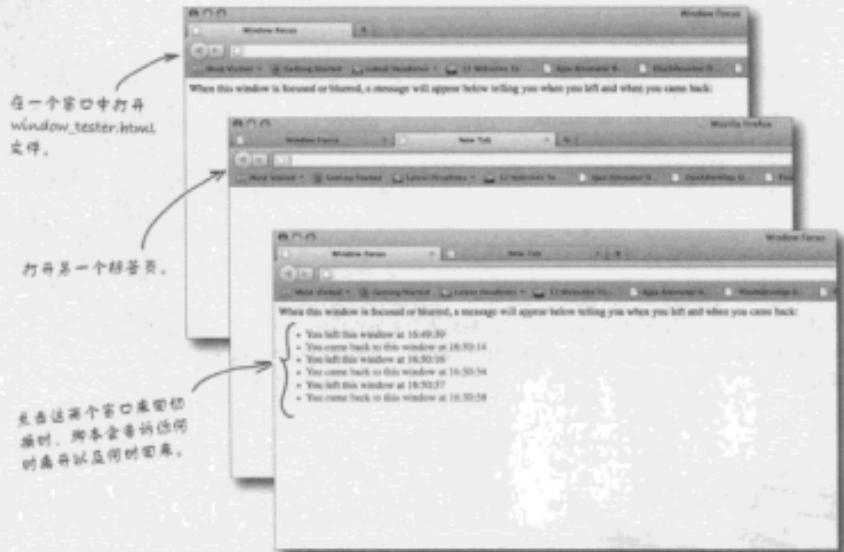
试一试

下面来试一试window对象的onfocus和onblur事件处理程序。在你下载的第7章代码文件夹中，可以找到一个名为window_tester的文件夹。在你喜欢的浏览器中打开这个文件夹中的window_tester.html文件。再打开另一个标签页，尝试着来回点击这两个浏览器窗口。



试一试

打开window_tester.html文件，再打开另一个标签页，交替点击切换这两个窗口时，会看到以下结果。



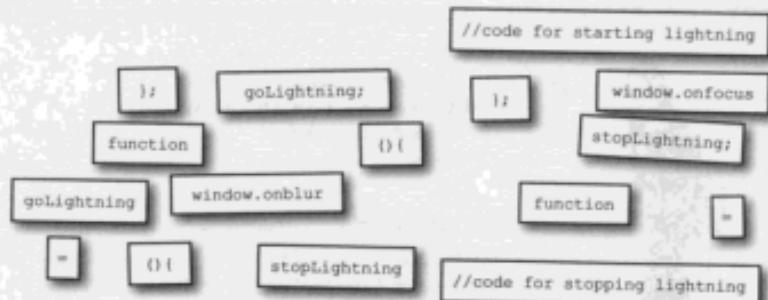
使用从**window**对象得到的信息，访问者离开怪兽变脸窗口时可以停止闪电，等访问者返回时再重新开始。

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴

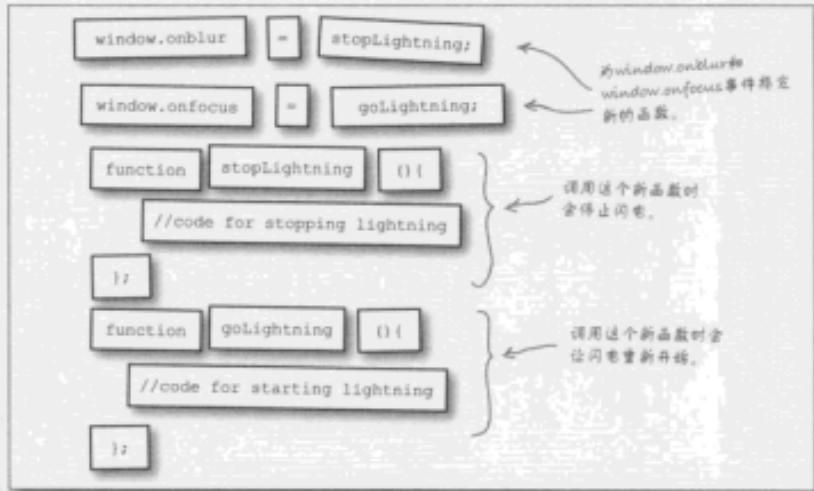
按正确的顺序摆放代码磁贴。为onblur和onfocus处理程序指定函数定义。其中一个函数定义应当在浏览器失去焦点时停止闪电（这个函数命名为stopLightning）。另一个函数定义要在浏览器重新得到焦点时重新开始闪电（这个函数名为goLightning）。现在你不用为这些函数编写代码，只需要在各个函数中放入以下磁贴和注释（以//开头）。





jQuery磁贴答案

现在你已经完成了这两个window对象事件处理程序的函数声明。



不过这些只是函数中的注释罢了。函数应当
做点什么才对啊！是不是只要能把定制闪电
函数复制粘贴进来？

没错，这些函数目前……什么也不做。

不过不是复制粘贴我们原来的代码，下面来看window对象的另外一个方法：定时器方法。利用这个定时器方法，我们可以更好地处理闪电的定时效果。

更多资源请访问稀酷客(www.ckook.com)

定时器方法告诉函数何时运行

JavaScript和jQuery都提供了定时器方法，可以根据经过多少时间来调用函数具体运行。JavaScript的window对象有4个定时器方法来完成定时控制：setTimeout、clearTimeout、setInterval和clearInterval。jQuery提供了delay方法。下面来进一步了解这些方法，看看它们分别有什么作用。

JavaScript定时器方法

setTimeout



如果你设置一个等待时间，等经过这段时间之后才让一个函数运行，就可以离我。

```
setTimeout(myFunction, 4000);
```

达到预定时间时调用的函数。

定时器延时（单位为毫秒）。

setInterval



我会告诉一个函数反复地运行。每次重复之前都有一个时间间隔。

```
setInterval(repeatMe, 1000);
```

各个时间间隔后重复运行的函数。

函数调用之间的间隔（单位为毫秒）。

jQuery的delay方法

delay



我会在效果链中的效果之间加入一个停顿。

```
slideDown().delay(5000).slideUp();
```

运行这个效果链，在jQuery中这称为一个效果队列。

在这个例子中，delay方法会在slideUp和slideDown效果之间加入一个5秒的停顿。

Sharpen your pencil

这些定时器方法中哪些最适合用来修正goLightning函数？请回答以下各个定时器方法是否有帮助，并解释你为什么选择或者不选择这个方法。

定时器	要用它吗？	为什么？
setTimeout		
setInterval		
delay		

更多资源请访问稀酷客(www.ckook.com)



Sharpen your pencil

Solution

这些定时器方法中哪些最适合用来修正goLightning函数？以下是我们答案。

定时器	要用它吗？	为什么？
setTimeout	No	如果希望等待一个特定的时间之后才运行一个函数，这种情况下才适用setTimeout方法。
setInterval	Yes	setInterval方法特别适用于希望函数反复运行的情况。这里恰恰需要闪电反复运行。
delay	No	delay方法对于排队的效果很适用，但是没有提供反复运行的机制。



这节课，setInterval就是完成`goLightning`函数的最佳方案。不过还需要`stopLightning`函数停止定时器。`clearInterval`方法会为我们做这个工作吗？

问得好！

可以使用`clearInterval`方法停止定时器，不再反复运行由`setInterval`创建的函数调用。为此，需要向`clearInterval`传入一个变量作为参数。下面来进一步了解它是如何工作的。

先定义一个变量，标记
setInterval方法。

```
myInterval = setInterval(repeatMe, 1000);
```

向`clearInterval`传入这个变量作为参数。

`clearInterval`方法返回`setInterval`清除定时器，并停止重复动作。

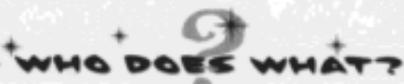
there are no
Dumb Questions

问：所有浏览器都采用同样的方式处理`setTimeout`方法吗？

答：并非如此。在Mozilla Firefox和Google Chrome中，就像前面所介绍的，这些浏览器会完成函数调用的堆叠。不过，Internet Explorer 9则不同，它会像第5章中那样不断调用函数。由此可以说明，并不只是Web设计人员会遇到跨浏览器兼容性问题。

问：`setInterval`和`setTimeout`之类的定时函数能不能用于`window`对象之外的其他对象？

答：这个问题问得很好。遗憾的是，答案是否定的。`setInterval`和`setTimeout`之类的方法是`window`对象的特定方法，只能由`window`对象调用。不过，调用时可以不加前缀“`window`”，即使没有这个前缀，浏览器也会知道你希望这个方法与当前的`window`对象关联。不过，加上前缀是一个很好的实践做法。



将各个定时器方法与运行这个方法完成的工作连线。

```
window.clearInterval(int1);

window.onfocus = goLightning;

setTimeout(wakeUp(), 4000);

$( "#container #lightning1" ).  
fadeIn(250).delay(5000).fadeOut(250).;

int1 = setInterval(function() {
    lightning_one();
},  
4000
);

window.onblur = stopLightning;
```

检测当前窗口何时得到焦点并调用`goLightning`方法。

设置`lightning_one`函数每4秒运行一次，并把它赋至变量`int1`。

检测当前窗口何时失去焦点并调用`stopLightning`函数。

清除定时器，停止反复运行对应`int1`的`setInterval`。

调用一个名为`wakeUp`的函数之前设置4秒的等待时间。

在`fadeIn`和`fadeOut`效果之间创建一个5秒的停顿。

更多资源请访问稀酷客(www.ckook.com)

* WHO DOES WHAT? *

答案

```

window.clearInterval(intl);
window.onfocus = goLightning;
setTimeout(wakeUp(), 4000);
$("#container #lightning1").fadeIn(250).delay(5000).fadeOut(250)..}
intl = setInterval(function() {
    lightning_one();
}, 4000);
window.onblur = stopLightning;

```

检测当前窗口何时得到焦点并调用goLightning方法。

设置lightning_one函数每4秒运行一次，并把它赋至变量intl。

检测当前窗口何时失去焦点并调用stopLightning函数。

清除定时器，停止反复运行对应intl的setInterval。

调用一个名为wakeUp的函数之前设置4秒的等待时间。

在fadeIn和fadeOut效果之间创建一个5秒的停顿。

编写stopLightning和goLightning函数

既然已经对定时器方法有了更多了解，下面来考虑哪里需要用到这些方法。

goLightning();

浏览器加载时开始闪光。

window.onblur = stopLightning;

浏览器失去焦点时调用stopLightning函数。

window.onfocus = goLightning;

浏览器得到焦点时调用goLightning函数。

```

function stopLightning () {
    //code for stopping lightning
}

```

清除对应3个闪电间隔的定时器。这里我们常常用3个 clearInterval。知道为什么吗？

```

function goLightning () {
    //code for starting lightning
}

```

为3个闪电间隔设置3个定时器。记得，这里需要3个setInterval。



在文件中填空，填入能够帮助修正怪兽变脸应用的变量、函数或方法。如果有疑问，可以查看前两页寻求帮助。我们已经帮你填了一些空。

```
goLightning();
window.onblur = stopLightning;
window.onfocus = goLightning;
var int1, int2, int3 ;
function goLightning() {
    int1 = .....(function() {
        .....,
        4000
    });

    ..... = .....(function() {
        .....,
        5000
    );
    ..... = .....(function() {
        lightning_three();
        .....,
        7000
    );
}

function stopLightning()
{
    window.....(int1);
    window.....(int2);
    window.....(int3);
}
function lightning_one()
{
    $("#container #lightning1").fadeIn(250).fadeOut(250);
}

function ..... {
    $("#container #lightning2").fadeIn(250).fadeOut(250);
};

function ..... {
    $("#container #lightning3").fadeIn(250).fadeOut(250);
}
```



my_scripts.js



Exercise Solution

现在你有了两个定制函数，分别引用了第5章写的函数，这两个定制函数要用来响应window对象的onfocus和onblur事件。

```

goLightning();
window.onblur = stopLightning;
window.onfocus = goLightning;
var int1, int2, int3; ← 声明3个变量来记住定时器，以便浏览器以后清除这些定时器。
function goLightning(){
    int1 = setInterval( function() { ←
        lightning_one(); ← 为3个闪电间隔设置3个不同的定时器。
    }, 4000, ← 然后调用lightning_one函数。
);
    int2 = setInterval( function() { ←
        lightning_two(); ← 然后调用lightning_two函数。
    }, 5000, ←
);
    int3 = setInterval( function() { ←
        lightning_three(); ← 或者使用lightning_three函数。
    }, 7000, ←
);
}

function stopLightning()
{
    window.clearInterval( int1 ); } ← 清除时在3个闪电间隔的定时器。
    window.clearInterval( int2 );
    window.clearInterval( int3 );
}

function lightning_one(){
    $("#container #lightning1").fadeIn(250).fadeOut(250); } ← 3个闪电函数定义。
}

function lightning_two(){
    $("#container #lightning2").fadeIn(250).fadeOut(250); }

function lightning_three(){
    $("#container #lightning3").fadeIn(250).fadeOut(250); }
}

```

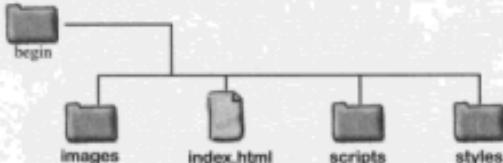


my_scripts.js

更多资源请访问稀酷客(www.ckook.com)

动手做!

需要更新大量代码来修正和改进第5章构建的应用，下面我们从一个空脚本文件着手。本书下载代码文件中包含对应第7章的一个文件夹。在这个文件夹中，可以找到一个begin文件夹，结构如下：



地上一页的代码……

……增加到脚本文件的\$(document).ready()代码块中。

```

$(document).ready(function() {
    //...//end doc.onready function
  
```

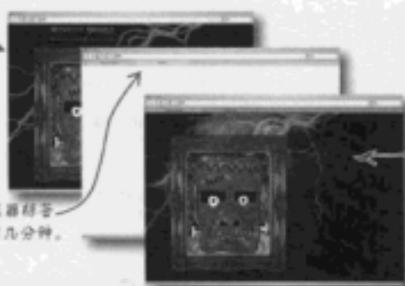


试一试

把上一页的代码增加到脚本文件中，然后在你喜欢的浏览器中打开这个页面，查看是否成功地修正了闪电效果。

在一个浏览器窗口中打开你想要测试的应用。

接下来，打开一个新的浏览器标签页，并在这个标签页上停留几分钟。



然后返回到运行改善应用的那个标签页。返回时，第一个效果应该不会运行。它应该在4秒之后才开始。

更多资源请访问稀酷客(www.ckook.com)

既然我们在做出改进，为什么不利用这个机会修正第5步构建的那些重复函数呢？

这里采用一种不同的数据结构可能会更好。常是一种能够处理多个变量的结构。

好主意。我们有很多与点击相关的重复函数，完全可以合并为一个
多用途的函数。

```
var headclix = 0, eyeclix = 0, noseclix = 0, mouthclix = 0;

$( "#head" ).click(function(){
    if (headclix < 9){
        $( "#head" ).animate({left:"-=367px"},500);
        headclix+=1;
    } else{
        $( "#head" ).animate({left:"0px"},500);
        headclix = 0;
    }
});

$( "#eyes" ).click(function(){
    if (eyeclix < 9){
        $( "#eyes" ).animate({left:"-=367px"},500);
        eyeclix+=1;
    } else{
        $( "#eyes" ).animate({left:"0px"},500);
        eyeclix = 0;
    }
});

$( "#nose" ).click(function(){
    if (noseclix < 9){
        $( "#nose" ).animate({left:"-=367px"},500);
        noseclix+=1;
    } else{
        $( "#nose" ).animate({left:"0px"},500);
        noseclix = 0;
    }
}); //end click

$( "#mouth" ).click(function(){
    if (mouthclix < 9){
        $( "#mouth" ).animate({left:"-=367px"},500);
        mouthclix+=1;
    } else{
        $( "#mouth" ).animate({left:"0px"},500);
        mouthclix = 0;
    }
}); //end click
```

你能否写出一个函数能
处理所有这些情况呢？





jQuery 磁贴

从上一页找出应用中所有不同方面共同的代码片段。使用下面的磁贴创建一个通用函数，名为moveMe，只要用户点击任何一个可移动的图像就会调用这个函数。对于moveMe函数，第一个参数是clix数组中的相应索引，第二个参数是所点击对象的引用。

```

var clix = [ ]; // head,eyes,nose,mouth

$( "#head" ).click( function() {
    moveMe (2, this);
}

); // end click function

$( "#eyes" ).click( function() {
    moveMe (0, this);
}

); // end click function

$( "#nose" ).click( function() {
    moveMe (3, this);
}

); // end click function

$( "#mouth" ).click( function() {
    moveMe (1, this);
}

); // end click function

function moveMe(____) {
    if (_____ < 9) {
        $(obj).animate({left:"-=367px"},500);
        clix[i] = clix[i]+1;
    }else{
        clix[i] = 0;
        $(____).animate({left:"0px"},500);
    }
}

```

obj

moveMe (0, this);

i, obj

moveMe (3, this);

clix[i]

(0,0,0,0)

moveMe (1, this);

JS

jQuery磁贴答案

现在你有了一个可重用的函数，它要利用一个数组，这样一来，需要维护的代码大大减少，而且现在更容易跟踪和调试可能遇到的问题。

```

var clix = [0,0,0,0] ; // head,eyes,nose,mouth      将clix转换为一个数组，这有助于简化你的代码。
    $("#head").click( function(){
        moveMe(0, this);                            ← moveMe函数将传入clix数组中一个槽的引用。然后可以使用这个槽来跟踪各个元素已被点击多少次。
    });//end click function

    $("#eyes").click( function(){
        moveMe(1, this);                          ← 正是向moveMe函数传入省略对象，来完成动画。
    });//end click function

    $("#nose").click( function(){
        moveMe(2, this);
    });//end click function

    $("#mouth").click( function(){
        moveMe(3, this);
    });//end click function

    function moveMe( i, obj ) {
        if ( clix[i] < 9 ){
            $(obj).animate({left:"-=367px"},500);
            clix[i] = clix[i]+1;
        }else{
            clix[i] = 0;
            $(obj).animate({left:"0px"},500);
        }
    }
}

```

通过创建一个多用途的moveMe函数，可以减少出现代码错误的可能性，同时还能减少需要维护的函数个数。

之前的重复逻辑现在统一放在一个地方，这样就能更容易地修正可能出现的问题。



更多资源请访问稀酷客(www.ckook.com)



试一试

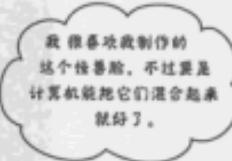
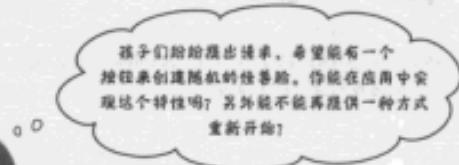
将上一页磁贴练习的代码增加到my_scripts.js文件，并保存。然后在你喜欢的浏览器中打开index.html页面，确保点击脸部的所有部分时你重写的函数不会引入任何问题。



与之前相比，怪兽笑脸页面看上去没有任何变化。不过要知道现在你的代码更高级，减少了重复，而且更容易维护。

怪兽变脸应用的特性需求

销售和质量保证小组对你所做的修正很满意。他们很欣赏你的工作，所以想把原来交给产品小组完成的一个怪兽变脸应用特性需求转交给你来实现。



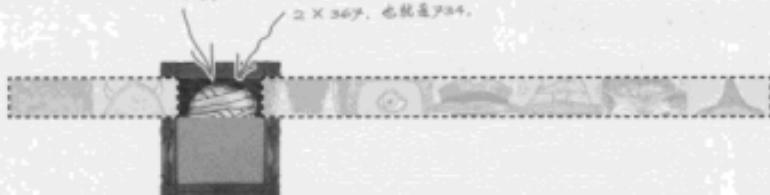
更多资源请访问稀酷客(www.ckook.com)

下面（再）来点随机性

这本书一直在构建随机函数。所以现在你可能已经是这方面的专家了。这里需要创建一个函数，随机地动画创建怪兽的脸。下面用分而治之的方法解决这个问题，把它划分为更小的步骤。首先确定每个图像条的当前位置。

需要跟踪各个怪兽面部的当前位置。假设动画正在查看这个图像。

当前位置是点击次数乘以脸部各个部分之间的距离（ 34μ 像素）。举例来说，当前位置就是 $2 \times 34\mu$ ，也就是 68μ 。



根据当前位置，我们需要确定目标位置，这实际上是屏幕上一个随机位置。可以分两部分来考虑：

- ① 得到一个随机数。

我们在第2章和第3章就是这样得到一个随机数。

```
var my_num = Math.floor((Math.random()*5) + 5);
```

不过现在需要一个介于 ± 10 的数（因为每个怪兽图像包含 ± 10 个部分）。

- ② 根据这个随机数将脸部各个部分移动到一个随机的位置。

需要将怪兽脸上各个部分移到`random`位置乘以怪兽各个部分的宽度。对于脸的数， ± 10 位置就是 $\pm 34\mu$ ，也就是 ± 349 。



嘿，这样听上去确实容易一些了。不过得到当前位置是不是很麻烦？我们如何知道图像条上的位置？特别是如果有人移除了图像条，又该怎么知道现在的位置呢？

没有你想象中那么困难。

实际上，只需要翻开下一页，你就会知道该怎么做。

你已经知道当前位置……

幸运的是，这里不需要给出任何新变量或函数。clix数组的索引值就能提供当前位置，因为它会告诉我们用户在怪兽脸上各个部分点击了多少次。所以你只需要一行代码：

设置当前位置为
clix[index]的值。
`var current_position = clix[index];`

……以及getRandom函数

我们在第2章、第3章和第6章构建了一个函数来获得随机数。这里可以重用这个函数，只需要稍加调整。

`function getRandom(num) {
 var my_random_num = Math.floor(Math.random() * num);
 return my_random_num;
}`

可以向getRandom函数传入一个数字为参数……

……生成并返回一个整数。在这里，我们能得到一个介于0到10的数。

将Math.random乘以传入的数
入参数，可以创建一个介于0到
num变量值之间的随机数。

- ① 设置变量，并把它传入函数：

向函数传
入一个值。
逐一个值。
`num = 10;
getRandom(num);`

下面是这个函数的核心操作：

有些人可能更喜欢把它称为一个实用函数。
这种函数只做一件事，但做得很棒。

`var my_random_num = Math.floor(Math.random() * num);`

- ② 然后是函数的结果（或输出）：

`return my_random_num;` 有些人喜欢将有返回值的函数称为“被
取函数”，因为它们会为你提供一个值。

下集预告：target_position（也就是我们想要滑动到的目
标位置，这是脸上的一个随机的部分）。

更多资源请访问稀酷客(www.ckook.com)



成品代码

在index.html和my_scripts.js文件中增加以下粗体代码。这会建立随机函数，并给出提醒，显示目标位置（一个随机数）和当前位置（由访问者点击的次数确定）。

```
<header id="top">
  
  <button id="btnRandom">Randomize</button>
  <button id="btnReset">Reset</button>
  <p>Make your own monster face by clicking on the picture.</p>
</header>
```



index.html

```
var w = 367; //width of the face strip
var m = 10; //number of the monster face strip we're on

$("#btnRandom").click( randomize );
$("#btnReset").click( );

function getRandom(num) {
  var my_random_num = Math.floor(Math.random() * num);
  return my_random_num;
}

function randomize() {
  // 随机確定脸上各个部分的位置。
  $(".face").each(function(index) {
    var target_position = getRandom(m);
    var current_position = clix[index];
    clix[index] = target_position; ← 将target_position设置为
    var move_to = target_position * w; ← getRandom函数的结果。
    $this.animate({left:"-= "+move_to+"px"},500);
  });
}
```



my_scripts.js



试一试

将上一页的代码输入到文件后，在你喜欢的浏览器中打开index.html页面，测试randomize函数。点击Randomize按钮10到20次，完成一个完整的测试。

随机数生成器开始工作……

前几次点击时，随机数
函数会按你要求的去做。



……前几次点击

点击几次后，随机数函数会
继续按你要求的去做。



我们没有编写相应的代码来避免脸上出现空白部分，这里只是位置为到达一个随机的位置！我们是不是走得太远了？

你说对了……

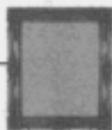
这些定制函数会产生一些预料之外的结果，不过它们确实在做代码中要求的工作。下面来看我们还有哪些方面没有考虑到。

`animate`语句将图像条不断向左滑动，最终超出用户在图片框内能看到的范围。可以说这么说，它会“越界”：

```
$(this).animate({left: -= " +move_to+"px"}, 500);
```

救命！我们要掉出去了！

如果用户不断点击`randomize`按钮，最终图像条会向左移动得太过，而无法出现在屏幕上。

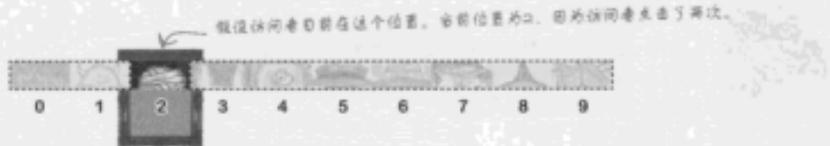


BRAIN
BARBELL

要保证图像条不会出界，总是对应怪兽脸上一个随机的部分。我们需要做什么？

相对于当前位置移动

要避免图像条出界，保证正确地落在怪兽脸上一个随机的部分，为此需要让它相对于当前位置移动。这意味着要包含当前位置和一些条件逻辑。下面分别来讨论。

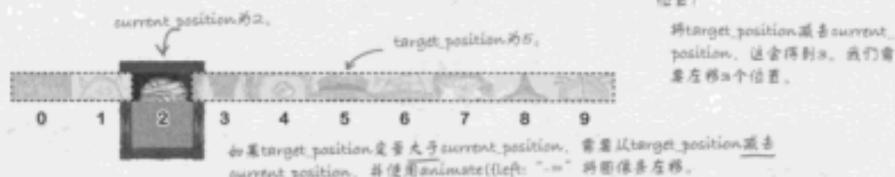


然后用户点击Randomize按钮，这会提供一个介于0到9的随机数。下面来看可能出现的两种不同场景：

场景1：目标位置>当前位置

getRandom函数返回一个值5。所以target_position变量设置为5。这说明它大于current_position变量。我们需要编写条件逻辑来处理这种情况。

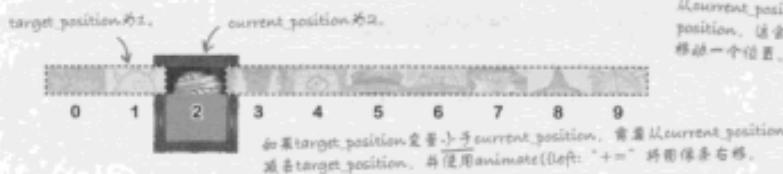
需要向右偏移移动多少个位置？



场景2：目标位置<当前位置

getRandom函数返回值1。target_position变量为1，这说明它小于current_position变量。根据场景1的条件逻辑，你能得出这里需要什么逻辑吗？

从current_position减去target_position，这会得到1。需要向右移动一个位置。



池塘谜题



你的任务是从池塘里取出代码片段，把它们放在代码中的空格里。
同一个代码片段不能多次使用，另外不是所有代码片段都必须用到。目标是让随机数函数按我们原计划的方式工作，保证脸上的各个部分不会变成空白。

```

var w = 367;
var m = 10;

function getRandom(num) {
    var my_random_num = Math.floor(Math.random() * num);
    return my_random_num;
}

function randomize(){
    $(".face").each(function(index) {
        var target_position = getRandom(m);
        var current_position = clix[index];
        clix[index] = target_position;

        if(... > current_position) {
            var move_to = (... - current_position) * w;
            $(this).animate(..., move_to);
        } else if(... < current_position) {
            var move_to = (... - current_position) * w;
            $(this).animate(..., move_to);
        } else {
            // They are the same - Don't move it.
        }
    });
}

```

注意：池塘里的所有片段只能使用一次！

```

        {left:"-= "+move_to+"px"},500
        {left:"+= "+move_to+"px"},500
        target_position > current_position {left:"-= "+move_to+"px"},500
        target_position - current_position
        current_position = target_position
        target_position == current_position
        target_position + current_position
        target_position < current_position
        each(function(index)

```

池塘谜题答案

你的任务是从池塘里取出代码片段，把它们放在代码中的空格里。
同一个代码片段不能多次使用。另外不是所有代码片段都必须用到。目标是让随机数函数按我们原计划的方式工作，保证脸上的各个部分不会变成空白。



```

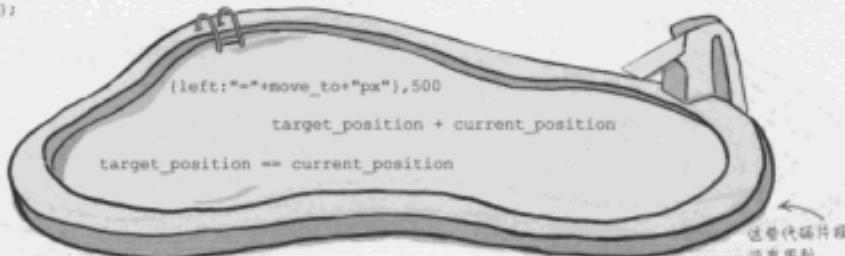
var w = 367;
var m = 10;

function getRandom(num) {
    var my_random_num = Math.floor(Math.random() * num);
    return my_random_num;
}

function randomize() {
    $(".face").each(function(index) {
        var target_position = getRandom(m);
        if(target_position === current_position) {
            var current_position = clix[index];
            clix[index] = target_position;
            if(target_position > current_position) {
                var move_to = (target_position - current_position) * w;
                $(this).animate({left: "=" + move_to + "px"}, 500);
            } else if(target_position < current_position) {
                var move_to = (current_position - target_position) * w;
                $(this).animate({left: "=" + move_to + "px"}, 500);
            } else {
                // They are the same - Don't move it.
                animate({left: "="});
            }
        }
    });
}

$(document).ready(function() {
    randomize();
});

```



这些代码片段
很有用到。

更多资源请访问稀酷客(www.ckook.com)

现在随机数函数能很好地工作了。我认为还需要另外一个定制函数重置所有元素。对吗？



非常正确。

还记得前几页index.html文件中的重置按钮吗？

现在只需要把它关联到一个定制的reset函数。



jQuery 磁贴

按正确的顺序摆放代码磁贴，完成重置按钮的代码，并构建你的定制reset函数。我们已经为你做了一些工作。

`$("#btnReset").click(reset);`

`function reset() {`

`}`

`$(".face")`

`)`

`$("this")`

`.each(function(index) {`

`clix[index] = 0;`

`.animate({left:"0px"},500);`

更多资源请访问稀酷客(www.ckook.com)



jQuery磁贴答案

太棒了！只需要几行简单的代码就能将一切还原。

```
$("#btnReset").click( reset );
```

调reset定制函数并原封重置
按钮。

```
function reset(){
```

定义定制reset函数。

```
$(".face")
```

```
.each(function(index){
```

使用each，将脸上所有各部分重置为0。

```
clix[index] = 0;
```

重置clix数组为0。

```
$(this).animate({left:"0px"},500);
```

便利各个图像多，将其
CSS left属性设置为一
个绝对值：0px。

```
});
```

```
}
```

there are no Dumb Questions

问： 所有浏览器都有window对象吗？

答： 是的，所有现代浏览器都有一个window对象，可以与之交互。每个window对象（浏览器中的每一个标签页）还有一个单独的document对象，Web页面就要加载到这个document对象中。

问： 那么，为什么必须相对于当前位置移动呢？不能直接移动到随机数指定的位置吗？

答： 这样也是可以的，不过你必须首先将图像重置回起始位置，然后再移动到随机函数指定的位置。这样你要编写和维护的代码量就会加倍，而且会使应用的速度大大减慢。

问： reset函数如何工作？

答： reset函数遍历处理类为face的所有元素，将这些元素的CSS属性left设置为0。它还会将clix数组中的各个元素设置为0，就像加载页面时一样。

动手做！

下面你会看到前几页构建的所有代码。请把粗体的代码增加到my_scripts.js文件中（如果之前还没有增加），准备测试你构建的所有新函数吧。

```

var w = 367; //width of the face strip
var m = 10; //number of the monster face strip we're on
$("#btnRandom").click( randomize );
$("#btnReset").click( reset );

function getRandom(num) {
    var my_random_num = Math.floor(Math.random() * num);
    return my_random_num;
}

function randomize(){
    $(".face").each(function(index){
        var target_position = getRandom(m);
        var current_position = clix[index];
        clix[index] = target_position;
        if( target_position > current_position ) {
            var move_to = (target_position - current_position) * w;
            $(this).animate({left:"-= "+move_to+"px"},500);
        }else if( target_position < current_position ){
            var move_to = (current_position - target_position) * w;
            $(this).animate({left:"+=" +move_to+"px"},500);
        }else{
            // They are the same - Don't move it.
        }
    });
}

function reset(){
    $(".face").each(function(index){
        clix[index] = 0;
        $(this).animate({left:"0px"},500);
    });
}

```



my_scripts.js



试一试

输入上一页的代码之后，在你喜欢的浏览器中打开index.html页面，测试randomize和reset函数。点击Randomize按钮10到20次，做一个完整的测试。另外间隔地点击重置按钮，确保它会正常工作。

成功了！

怪兽脸的各个部分现在可以同时右移和左移，这会为用户增加更多视觉效果。

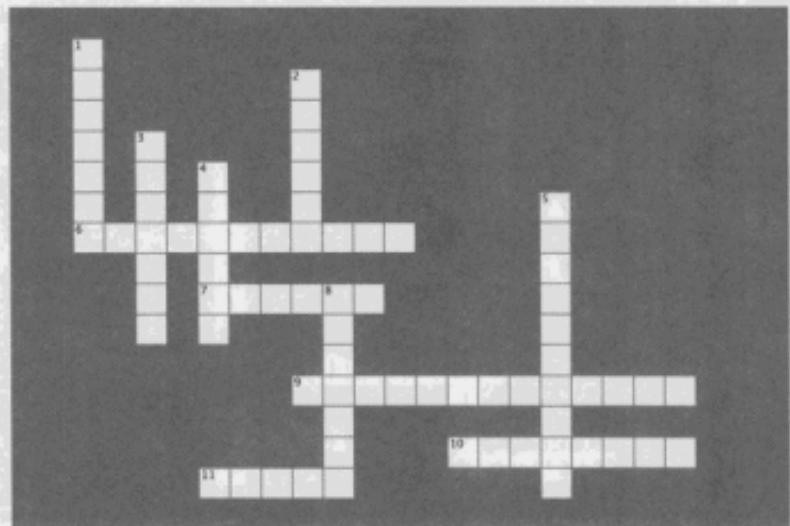
重置按钮将一切还原到起始状态。





jQuery填字游戏

现在该坐下来让你的左脑动一动了。这是一个标准的填字游戏，所有答案都可以在这一章中找到。



横向

- 6.让函数反复运行，每次重复有一个时间间隔。
- 7.这个JavaScript事件处理器可以检测窗口何时失去焦点。
- 9.这个JavaScript方法用于取消重复运行之间的等待时间。
- 10.用于向函数“传递”变量或对象。提示：想想括号。
- 11.这个jQuery方法可以在方法链中的效果之间加入停顿。

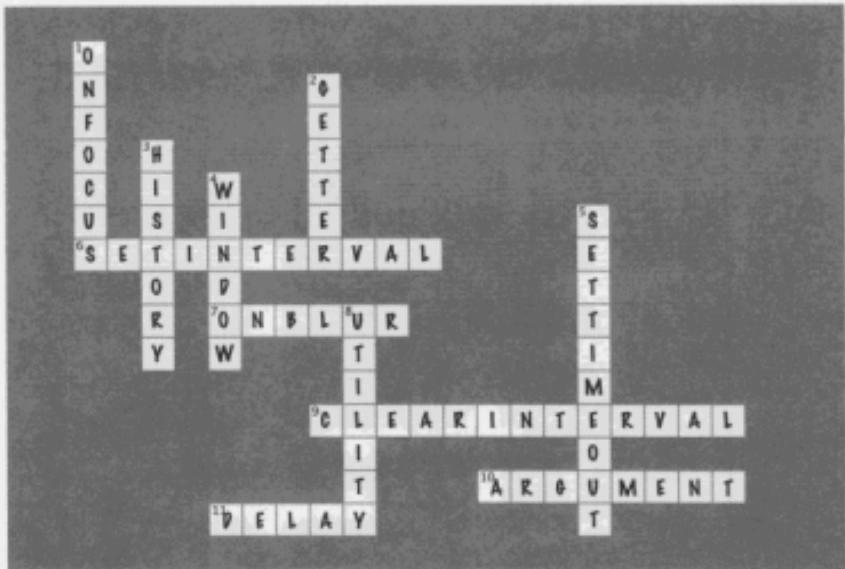
纵向

- 1.这个事件处理器用来检测窗口何时接收一个点击、键盘输入或另外某种输入。
- 2.返回值的函数有时称为_____函数。
- 3.window对象的一个属性，允许访问窗口之前访问的不同URL。
- 4.每次访问者在浏览器中打开一个新窗口时就会创建这个全局对象。
- 5.想要设置运行一个函数前要等待的时间时就要使用这个JavaScript定时器方法。
- 8.这种函数只做一件事而且做得很好，有人把这种函数称作_____。

更多资源请访问稀酷客(www.ckook.com)



jQuery填字游戏答案



更多资源请访问稀酷客(www.ckook.com)

怪兽变脸2大获成功！

你的辛苦工作让这个怪兽变脸应用成为孩子们的最爱！你解决了些问题，还在指定的时间内增加了一个新的特性！为了对你的卓越工作表示鼓励，我已经向老板提出要涨高你的报酬。



看呐，我做了一个僵尸巫师！



更多资源请访问稀酷客(www.ckook.com)

你现在的位罝 >



JQuery工具箱

你已经掌握了第7章，你的工具箱中又增加了
window对象、定时函数和定制函数。

window对象

这是JavaScript中最顶层的对象。它包括属性、事件处理程序和方法，可以帮助检测和响应浏览器事件。

onFocus指出一个浏览器窗口何时激活。

onBlur检测一个窗口何时失去焦点。

定时函数

定时函数是window对象提供的一些方法。

setTimeout通知一个函数要等待设定的一个时间后才能运行。

setInterval会反复运行一个函数，运行函数之间有一个特定的时间间隔。

clearInterval会清除重复的函数调用。

优化的定制函数

编写你自己的定制函数，真正开始建立人们想要的交互式Web页面。

不过也可能已经过正，一定要了解如何最好地结合和优化你的函数，从而能编写更少的代码，而且代码更易于维护和调试，这一点很重要。

8 jQuery与Ajax

请把数据传给我

一点点Ajax，一滴jQuery，7
小杯炼奶油。亲爱的，你确定把
食谱写下来了吗？



用jQuery做些酷炫的CSS和DOM处理很有意思，不过接下来可能还需要从一个服务器读取和显示信息（或数据）。甚至可能还要用从服务器读取的信息更新页面的某些部分，而无需重新加载页面。来认识认识Ajax。结合jQuery和JavaScript，它能很好地完成这些工作。这一章中，我们将学习jQuery如何向服务器做出Ajax调用，还会了解利用返回的信息能够做什么。

Bit to Byte长跑大赛与日俱进

来自: Web镇MegaCorp公司市场部

主题: 第42届年度Bit to Byte长跑大赛结果页面

嗨，Web设计小组。

你们可能注意到了，每年我们都会赞助Web镇的年度Bit to Byte 10公里长跑大赛。为他们提供大赛结果页面。不过我们的页面有些过时。因为现在只是在所有结果都得到之后才更新页面。人们希望随时都能了解。参加比赛的人要求我们利用Twitter和Facebook提供实时的结果。所以现在交给你一个任务，报酬会很诱人。如果你能在下周之前更新我们的Web镇结果页面，提供实时的结果，你会得到赛道终点附近的VIP专区贵宾票（噢，对了，我们提到过今年大赛在毛伊岛举办吗）。

以下是我们需求：

- (1) 页面应当提供选项只显示男运动员或女运动员，或者同时显示所有参赛者。
- (2) 参赛者冲过终点线时应当提供自动更新。
- (3) 结果更新时人们无需刷新页面。
- (4) 最后一点，我们希望在页面上指出最后一次更新是什么时间，并给出更新的频率，如果人们有要求，还应当允许他们开始和停止更新。

看起来和去年的页面相比并没有太大差别，所以你有一个不错的起点。这可是一个重要赛事，真想马上看到你的作品！

Dionah C. Housney
市场部首席
Web镇MegaCorp公司

哈，毛伊岛！如果我们能坐在
VIP专区就太棒了！

看起来www小姐已经
整理得乱七八糟了……
收拾应该没问
题吧！



看看去年的页面

下面来看去年的页面，了解它是如何建立的，另外看看页面到底是什么样子，这样我们才能更好地理解市场部门提出的要求。



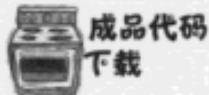
配置插件

插件是对基本jQuery库的扩展，可以改进功能，或者更容易地完成某些特定的函数或任务。在上面的例子中，通过结合CSS，idTabs插件可以把ul元素转换为可点击的标签页，指出点击li中的链接时要显示哪些div元素。这个插件为页面提供了一个易于使用的导航结构，使我们能够在相同的显示区中区别显示不同类型的信息。



不要太担心插件。

插件可以为默认jQuery库提供额外的功能。我们将第10章进一步研究插件，不过现在先来看看这个插件对这个项目有什么帮助……



成品代码

下载

继续学习后面的内容之前，先来看看去年的文件，了解它们是如何建立的。这些代码应该放在last_year.zip文件中（本章的所有其他文件可以从<http://thinkjquery.com/chapter08>下载）。下面先来看我们需要的3个主要文件：my_style.css、index.html和my_scripts.js。

```
body{
    background-color: #000;
    color: white;
}
/* CSS注释 */

/* Style for tabs */
#main {
    color:#111;           /* 从这里开始，下面的
    width:500px;          CSS将用于构建页面
    margin:8px auto;      上的标签页。
}
#main > li, #main > ul > li
{ list-style:none; float:left; }
#main ul a {
    display:block;
    padding:6px 10px;
    text-decoration:none!important;
    margin:1px 1px 1px 0;
    color:#FFF;
    background:#444;
}
```

```
#main ul a:hover {
    color:#FFF;
    background:#111;
}

#main ul a.selected {
    margin-bottom:0;
    color:#000;
    background:snow;
    border-bottom:1px solid snow;
    cursor:default;
}

#main div {
    padding:10px 10px 8px 10px;
    *padding-top:3px;
    *margin-top:-15px;
    clear:left;
    background:snow;
    height: 300px ;
}

#main div a {
    color:#000; font-weight:bold;
}
```



my_style.css

更多资源请访问稀酷客(www.ckook.com)

创建链接，它们
指向该文件并成为根
页面。

包含嵌套页内容
的div元素。

```
<div id="main">
  <ul class="idTabs">
    <li><a href="#about">About the Race</a></li>
    <li><a href="#finishers">All Finishers</a></li>
  </ul>
  <div id="about">
    <h4>About the race</h4>This race Bit to Byte Campaign!
  </div>
  <div id="finishers">
    <h4>All Finishers</h4>
    <ul id="finishers_all">
      <li>Name: Bob Hope. Time: 25:30</li>
      <li>Name: John Smith. Time: 25:31</li>
      <li>Name: Jane Smith. Time: 25:44</li>
      ...
    </ul>
  </div>
  ...
<script src="scripts/jquery-1.6.2.min.js"></script>
<script src="scripts/my_scripts.js"></script>
<script src="scripts/jquery.idTabs.min.js"></script>
```



index.html

`$ (document).ready(function() {`

`getTime();` 使用我们的定制getTime函数。

`function getTime() {` JavaScript Date对象的一个新实例。

`var a_p = "";` Date对象的方法。

`var d = new Date();` JavaScript三元操作符（稍后会
做更多说明）。

`var curr_hour = d.getHours();` Date对象的方法。

`(curr_hour < 12) ? a_p = "AM" : a_p = "PM";` Date对象的方法。

`(curr_hour == 0) ? curr_hour = 12 : curr_hour = curr_hour;` Date对象的方法。

`(curr_hour > 12) ? curr_hour = curr_hour - 12 : curr_hour = curr_hour;` Date对象的方法。

`var curr_min = d.getMinutes().toString();` Date对象的方法。

`var curr_sec = d.getSeconds().toString();` Date对象的方法。

`if (curr_min.length == 1) { curr_min = "0" + curr_min; }` Date对象的方法。

`if (curr_sec.length == 1) { curr_sec = "0" + curr_sec; }` Date对象的方法。

`$('#updatedTime').html(curr_hour + ":" + curr_min + ";" + curr_sec + " " + a_p);`

});



my_scripts.js

力求动态

市场部希望页面近实时地更新，所以HTML文件中这些硬编码的结果必须删除。而且原来只是使用JavaScript更新页面上的时间！这是一个绝好的机会，可以借此机会让你的jQuery水平更上一层楼。欢迎来到下一代Web应用，在这里，利用jQuery和JavaScript，再加一点Ajax和XML，可以让Web应用就像是动态的（不同于静态）、响应性的桌面应用。

Ajax代表“异步JavaScript和XML”（Asynchronous JavaScript and XML），这是一种采用结构化格式在web服务器和浏览器之间传递数据的方式，采用这种方式传递数据不会影响网站访问者。利用Ajax，页面和应用只向服务器请求真正需要的内容一只涉及需要改变的页面部分，以及服务器提供那些部分所需的数据。这说明数据流量会更小、更新更少，访问者呆坐着等待页面刷新的时间也更短。

最棒的是，Ajax页面是用标准Internet技术构建的。都是本书中已经见过或者你已经知道如何使用的那些技术，比如：

- HTML
- CSS
- JavaScript
- DOM

要使用Ajax，我们需要认识一种数据格式（XML），这种格式已经存在一段时间了，另外还要了解处理Ajax请求的jQuery方法ajax。

使用Ajax时，Web页面只在
需要时（以及需要的地方）
向服务器请求它们真正需要
的内容。

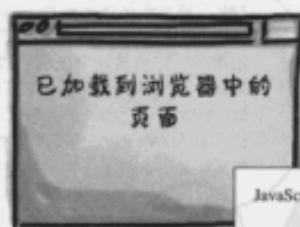
老Web，来认识一下新Web

尽管现在我们了解了一些jQuery，不过处理数据还是会把我们拖回到老Web的年代，那时要想得到要一些或全部更新的数据，必须刷新整个页面，或者链接到另外一个页面。我们面对的网站看起来行动相当缓慢，因为每次都不得不从服务器请求整个页面。如果处理数据还是这么慢，我们学习jQuery又有什么意义呢？

认识Ajax

利用Ajax，可以用一种动态的方式与服务器交换数据。使用Ajax和一些DOM管理，可以用jQuery和JavaScript只加载（或重新加载）页面的一部分。

①



jQuery或JavaScript向服务器发起一个Ajax请求。

发送到服务器的
Ajax请求



jQuery或JavaScript接收结果，完成解析，只更新页面的某一部分。

从服务器
接收的响应



更新页面的一部分

了解Ajax

前面已经提到，Ajax是一种采用结构化格式在服务器和浏览器之间传递数据的方式，采用这种方式传递数据不会影响网站访问者。不过实际上，它并不是一样东西，这是多种不同技术的组合，可以用来构建超炫的交互式Web应用。利用JavaScript，可以与页面的DOM结构交互。异步意味着它可以在后台进行，不会影响页面或用户与页面的交互。X则与数据有关。

什么是Ajax?

异步

JavaScript向服务器发出请求，不过你仍然可以与页面自由地交互，填写Web表单，甚至可以点击按钮，与此同时Web服务器一直在后台工作。服务器工作完成时，你的代码可以只更新页面中有变化的部分。这个过程中你根本不需要干等着。这正是异步请求的强大之处！

XML

XML，或可扩展标记语言（eXtensible Markup Language）是一个存储信息的规范。这也是一个描述信息结构的规范。尽管XML是一个标记语言（类似于HTML），但XML并没有自己的标记。编写XML的人可以创建他们需要的标记。



JavaScript

你现在应该已经很了解了，JavaScript是一个用于Web内容开发的脚本语言，主要用来创建函数。这些函数可以嵌入或包含在HTML文档中，并与DOM交互。

但是，不能直接使用HTML吗？为什么还需要另外一种标记语言呢？



没错，确实可以使用HTML。不过，对于信息的传输，相对于她的姊妹语言HTML，XML还有一些特别的好处。下面就来看这些好处到底是什么。

更多资源请访问稀酷客(www.ckook.com)

X是什么

XML是可扩展标记语言（Extensible Markup Language）的缩写。它提供了一种得到广泛接受的标准方式，采用一种可处理的格式表示文本和数据，而无需太多人为介入。采用XML格式的信息可以跨平台、跨应用、甚至跨编程和书写语言进行交换，可用于大量开发工具和实用工具。XML很容易创建和编辑，只需要一个简单的文本编辑器，另外只要在文件最前面加上XML声明，剩下的全由你来决定！

XML什么也不做

听起来可能有点奇怪，不过XML本身并不做什么。XML可以建立信息的结构，存储信息以便传输。实际上，XML是一种描述标记语言的元语言。换句话说，XML提供了一种功能，可以定义标记以及标记之间的结构关系。XML并不是HTML的替代品，而是HTML的补充，了解这一点很重要。在很多Web应用中，XML用来格式化数据完成传输，而HTML用于格式化和显示数据。下面进一步查看一个XML文件，这个文件包含关于一些书的数据。

XML声明，必须包含这个声明，它定义了文档的XML版本。

根节点子节点（书籍和读者标记）。这里在描述某物信息。

```
<?xml version="1.0" encoding="utf-8"?>
<books> ← 根节点或根标记。
  <book>
    <title>The Hitchhiker's Guide to the Galaxy</title>
    <author>Douglas Adams</author>
    <year>1980</year> ← 其他标记，用于包含数据
  </book>
  <book>
    <title>The Color of Magic</title>
    <author>Terry Pratchett</author>
    <year>1983</year>
  </book>
  <book>
    <title>Mort</title>
    <author>Terry Pratchett</author>
    <year>1987</year>
  </book>
  <book>
    <title>And Another Thing...</title>
    <author>Eoin Colfer</author>
    <year>2009</year>
  </book>
</books> ← 结束根节点或标记。
```



**XML用来格式化
数据进行传输，而
HTML用于格式化
和显示数据。**

更多资源请访问稀酷客(www.ckook.com)

there are no Dumb Questions

问：这么说，XML的关键就是可以创建你自己的标记？

答：完全正确！能定义适用于你的元素和结构会相当方便。更棒的是，XML是一个标准。所以很多人都知道如何使用XML。这说明，你的词汇表对很多程序员都可用，包括编写客户端程序和服务器端程序的开发人员。

问：建立我们自己的数据格式不是更容易吗？

答：乍一看可能如此，不过专用的数据格式（你为自己的特定用途专门建立的数据格式）实际上会带来很多问题。如果你没有提供文档，人们可能会忘记如何使用这些格式。如果格式有变化，你还要确保一切都随之相应更新：客户端、服务器、数据库、文档……这实在让人很头疼。

问：人们真的像你说的那样在广泛使用XML吗？

答：XML可以创建你需要的任何数据结构。由于这种灵活性，Web上很多不同类型的标记语言都以XML

为基础。有超过150种不同类型的语言使用XML。从面向新闻或音频/视频摘要的RSS[RDF站点摘要（RDF Site Summary），也称为真正简单合成（Real Simple Syndication）]，到KML[销眼标记语言（Keyhole Markup Language）]，它面向Google Earth中使用的地理信息；再到OOXML（Office Open XML），这是微软为字处理文件、样式表、演示文稿等等提交的一个标准；再到SVG（可扩展矢量图形），这用来描述二维图像；以及SOAP[简单对象访问协议（Simple Object Access Protocol）]，它定义了在Web服务上交换信息的方法。哇，使用XML的用户确实不少！

问：OK，我知道了为什么要使用XML，不过当我们开始声明元素名时，它会不会变成一个“专用的数据格式”呢？

答：不，绝对不会。这正是XML的妙处：它很灵活。服务器和客户端需要查我相同的元素名，不过通常在运行时就可以确定。

问：不是所有Web页面都是异步的吗？比如浏览器加载一个图像时我已经在看页面了。

答：浏览器是异步的，不过标准Web页面并不是。通常Web页面需要服务器端程序的信息时，所有一切都会完全停顿，直至服务器响应……除非页面发出一个异步请求。这正是Ajax的工作。

问：我怎么知道什么时候使用Ajax和异步请求，什么时候不用呢？

答：可以这样想：如果你希望用户工作时一切继续进行，可能就需要一个异步请求。不过，如果用户在继续下一步之前需要从你的应用得到信息或响应，就应当让用户等待。这往往意味着需要一个同步请求。

问：难道我不能使用XHTML与XML交互吗？

答：有意思的是：XHTML就是XML，究其本质。XHTML并不像人们想象的那样类似于HTML。在解析方面，XHTML是一个更严格的语言。与XML出自于同一系列。不过，这并不意味着它能比HTML更好地解析XML或者能与XML更好地交互。这本书中的标记实际上使用了HTML5。标准规范发布时它将涵盖XHTML5。



我很愿意使用Ajax，不过
首先要建立相应的结构，不是吗？以前每
次都是这样做的……

你说得对……

下面就要建立这个结构，以便
在页面上增加一些Ajax……



HTML代码磁贴

重新组织磁贴，完成下面的代码，创建两个新的标签页。可以显示两个不同的信息：一个对应男运动员（ID为male），另一个对应女运动员（ID为female）。可以把About标签页删除，不过要保留All Finishers标签页（全部运动员）。在每一部分放入一个空的ul元素来包含运动员。另外，从finishers_all ul元素删除所有现有内容。

```

<body>
    <header>
        <h2> _____ </h2>
    </header>
    <div id="main">
        <ul class="idTabs">
            <li><a href="#">Male Finishers</a></li>
            <li><a href="#female"> _____ </a></li>
            <li><a href="#all">All Finishers</a></li>
        </ul>
        <div id="male">
            <h4>Male Finishers</h4>
            <ul id=" _____ "></ul>
        </div>
        <div _____ >
            <h4>Female Finishers</h4>
            <ul id="finishers_f"></ul>
        </div>
        <div _____ >
            <h4>All Finishers</h4>
            <ul id=" _____ "></ul>
        </div>
    </div>
    <footer>
        <h4>Congratulations to all our finishers!</h4>
        <br>Last Updated: <div id=" _____ "></div>
    </footer>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script src=" _____ "></script>
    <script src="scripts/jquery.idTabs.min.js"></script>
</body>

```



index.html

"scripts/my_scripts.js"

#male

"finishers_all"

2011 Race Finishers!

"updatedTime"

Female Finishers

finishers_f

id="female"

id="all"



HTML代码磁贴答案

现在你应该有了两个新标签页，分别对应男运动员和女运动员。

```

<body>
    <header>
        <h2> 2011 Race Finishers! </h2>
    </header>
    <div id="main">
        <ul class="idTable">
            <li><a href="#male">Male Finishers</a></li>
            <li><a href="#female">Female Finishers </a></li>
            <li><a href="#all">All Finishers</a></li>
        </ul>
        <div id="male">
            <h4>Male Finishers</h4>
            <ul id="finishers_m"></ul>
        </div>
        <div id="female">
            <h4>Female Finishers</h4>
            <ul id="finishers_f"></ul>
        </div>
        <div id="all">
            <h4>All Finishers</h4>
            <ul id="finishers_all"></ul>
        </div>
    </div>
    <footer>
        <h4>Congratulations to all our finishers!</h4>
        <br>Last Updated: <div id="updatedTime"></div>
    </footer>
    <script src="scripts/jquery-1.6.2.min.js"></script>
    <script src="scripts/my_scripts.js"></script>
    <script src="scripts/jquery.idTable.min.js"></script>
</body>

```



index.html



运动员列表

包含一个jQuery
插件来建立格
签页。



试一试

用磁贴练习中完成的代码更新index.html文件。在你喜欢的浏览器中打开这个文件。



真不错！

页面开始初具雏形了。现在来看如何从服务器得到数据，用一些真正的比赛信息填充各个标签页。



更多资源请访问稀酷客(www.ckook.com)

用Ajax方法获取数据

想要些数据？jQuery和Ajax已经做好了准备。jQuery的Ajax方法会返回一个对象（还记得第6章介绍的对象吧），其中包含要完成的特定动作的相关数据。Ajax方法可以接受很多不同的参数，可以用POST向服务器提交数据，或者用GET从服务器获取数据。

```
jquery快捷方式 → $ .ajax({ → jquery ajax方法
    想通过Ajax获取的URL → url: "my_page.html"
    success: function(data){ → 如果Ajax方法成功则运行这个函数。
        } → 然后我们会在那里增加更多代码。
    });
    };
```

Ajax调用返回的数据。

要了解这个方法的所有参数，可以访问jQuery文档网站 (<http://api.jquery.com/jQuery.ajax/>)。另外还有一系列jQuery便利方法可以用来处理Ajax调用。别担心，后面还会介绍这些方法。

对现在来说，只需要用这个代码更新my_scripts.js文件，只包含下面用粗体显示的新代码。

```
$ (document) . ready(function() {
    $ . ajax({
        url: "finishers.xml",
        cache: false,
        dataType: "xml",
        success: function(xml){
    }
});
    });

    getTime();

    function getTime(){
        var a_p = "";
        var d = new Date();
    }
});
```

通过Ajax加载finishers.xml文件。

这个参数表示在本地缓存结果，这样可以减少对服务器的调用。

希望从服务器返回的数据类型。



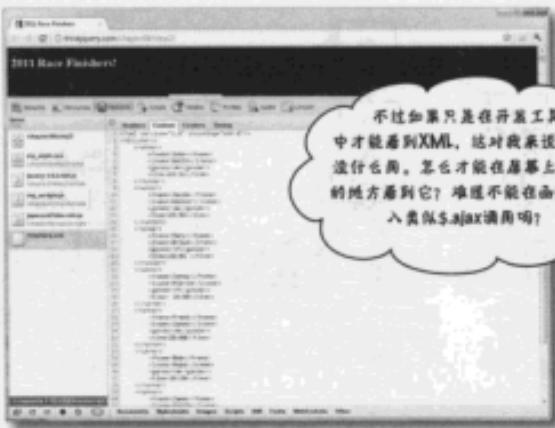
my_scripts.js

更多资源请访问稀酷客(www.ckook.com)



试一试

用上一页的代码更新my_scripts.js文件。然后从<http://thinkjquery.com/chapter08/step2/finishers.xml>下载这一章的示例XML文件，把它保存在index.html文件的同一目录下。完成后，在浏览器中打开index.html，再打开“Network”标签页（如果使用Google Chrome的开发工具），或者打开“Net”标签页（如果使用Firefox的Firebug）。这里会列出你的XML文件，还会列出页面的其他文件。



不过如果只是在开发者工具中才能看到XML，这对我来说可没什么用。怎么才能在屏幕上我需要的地方看到它？难道不能在函数中加入类似\$.ajax调用吗？



问得好。

既然知道可以在浏览器中加载XML文件，我们需要选出必要的文本，在屏幕上显示。要找到各个运动员，将他（或她）增加到页面上正确的列表中。另外，你说得没错，在函数中加入ajax调用是一种很好的做法，这样一来，需要时就可以调用。



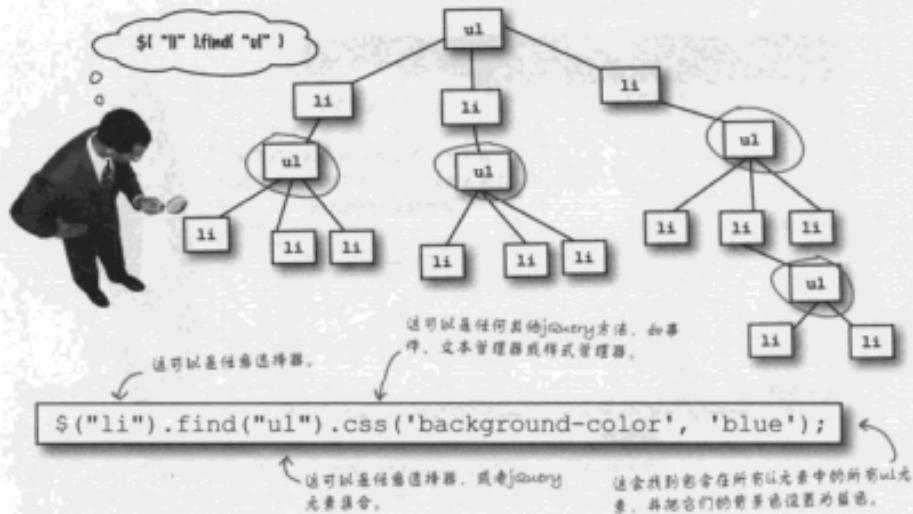
Watch it!

Ajax调用受到同源策略的限制！

同源策略是有关JavaScript和其他客户端脚本语言的一个安全概念。允许在页面上运行脚本，来访问来自相同服务器的资源，如元素属性和方法，但不允许脚本访问其他服务器上页面中的元素。由于遗留兼容性，JavaScript的includes不会做这些检查，不过这个例子中的XML文件会受到这个限制。这说明，XML文件与加载它的页面必须在同一个服务器上。

解析XML数据

我们需要从XML文件中选出各个运动员，并在屏幕上显示。很幸运，jQuery为我们提供了find方法。它的任务就是根据我们提供的规则选出匹配的元素。find允许我们在一个层次结构的信息集（如DOM树或XML文档）中搜索元素的后代，并由匹配的元素构造一个新数组。find与children方法类似（我们在第4章为Web镇餐厅构建菜单时讨论过children方法），只不过后者只在DOM树中向下遍历一层。我们可能需要走得更远……



通过结合使用find方法和each方法，可以搜索一组元素，使用一个循环分别与各个元素交互。



要在屏幕上显示各个运动员，你能想出需要与XML文档中的哪些部分交互吗？

更多资源请访问稀酷客(www.ckook.com)



jQuery代码磁贴

重新组织代码磁贴，创建一个名为getXMLRacers的函数。它要调用ajax方法，并加载finishers.xml文件。一旦加载成功，清空所有包含信息的列表，然后查找XML文件中的各个运动员，确定是男运动员还是女运动员。根据性别把运动员追加到适当的列表中，还要把他们全部增加到finishers_all列表中。最后调用getTime函数更新页面上的时间。

```

function _____ {
    $.ajax({
        url: _____,
        cache: false,
        dataType: "xml",
        _____ function(xml) {
            $(_____.empty());
            $("#finishers_f")._____();
            $("#finishers_all").empty();
            $(xml).find_____(function() {
                var info = "<li>Name: " + $(this).find_____.text() + " " + $(this).
                find("lname").text() + ". Time: " + ______.text() + "</li>";
                if( $(this).find("gender").text() == "m" ) {
                    $("#finishers_m").append_____
                } else if( $(this).find("gender").text() == "f" ) {
                    ______.append(info);
                } else{
                    ______.append(info);
                }
            });
        }
    });
}

```

"finishers.xml"

getXMLRacers()

getTime()

success()

(info)

\$("#finishers_all")

'#finishers_m'

("runner").each

\$("#finishers_f")

.empty()

("fname").text()



my_scripts.js



jQuery代码磁贴答案

使用find和each，可以循环处理finishers.xml文件，检查性别，并把各个运动员增加到这个Web应用适当的标签页中。

```

function getXMLRacers() {
    $.ajax({
        url: "finishers.xml",
        cache: false,
        dataType: "xml",
        success: function(xml) {
            $('#finishers_m').empty();
            $('#finishers_f').empty();
            $('#finishers_all').empty();
            $(xml).find("runner").each(function() {
                var info = '<li>Name: ' + $(this).find("fname").text() + ' ' + $(this).
                find("lname").text() + '. Time: ' + $(this).find("time").text() + '</li>';
                if ($(this).find("gender").text() == "m") {
                    $('#finishers_m').append(info);
                } else if ($(this).find("gender").text() == "f") {
                    $('#finishers_f').append(info);
                } else {
                    $('#finishers_all').append(info);
                }
            });
            getTime();
        }
    });
}

```

清空所有的ul元素，使它们能读取更新后的数据。

循环处理XML文件中的各个runner元素。

检查各个运动员的性别，将他（她）增加到适当列表中。

另外，还要把所有运动员都增加到finishers_all列表中。

调用getTime函数，用最先读取的XML文件中的时间更新页面。



my_scripts.js

这个例子中，“var info = ...”开头的代码行太长了，一行放不下，所以必须放到下一行。你的代码中不需要这样运行。

更多资源请访问稀酷客(www.ckook.com)



试一试

更新my_scripts.js文件，加入getXMLRacers函数。另外，把getTime函数调用（`document.ready`中）替换为getXMLRacers函数调用。现在会在这个新函数中调用getTime函数。一定要通过web服务器运行你的所有代码，所以URL应当是`http://`，而不是`file://`。另外，要保证XML文件与HTML文件在同一个服务器上，否则会遇到讨厌的同源许可问题。

The screenshot shows a web page titled "2011 Race Finishers". Below the title are three tabs: "Male Finishers", "Female Finishers", and "All Finishers". The "All Finishers" tab is selected, displaying a list of names and times:

- Name: John Smith, Time: 25.32
- Name: Jacob Walker, Time: 25.54
- Name: Mary Stevens, Time: 26.02
- Name: Sam Johnson, Time: 26.08
- Name: Frank Lewis, Time: 26.08
- Name: Bob Hope, Time: 26.16
- Name: Jane Smith, Time: 26.04
- Name: Ryan Blue, Time: 26.24

Below the list, a message says "Congratulations to all our finishers!" and the date "Last Updated: 11/27/04 AM". A thought bubble from a character at the bottom left contains the following text:

太棒了，我有了一个函数，可以调用这个函数得到XML数据。不过如果需要自动更新，这个函数是不是需要运行多次？



对，确实如此。

很幸运，上一章我们已经见过如何调度事件，让它们在页面上定期发生。下面再简单回顾一下，并了解这一次我们有哪些选择……

更多资源请访问稀酷客(www.ckook.com)

调度页面上的事件

上一章中，我们已经了解JavaScript和jQuery都提供了定时器方法，可以根据经过了多长时间调用函数运行。JavaScript的window对象有4个定时器方法用于定时控制：`setTimeout`、`clearTimeout`、`setInterval`和`clearInterval`。jQuery提供了`delay`方法，不过它只强调效果，不支持调度或重复动作。所以，在这里这个方法对我们没有帮助……

JavaScript定时器方法



`setTimeout`

如果你想设置一个等待时间，等经过指定时间之后才让一个函数运行，就可以用我。

```
setTimeout(myFunction, 4000);
```

达到超时时间后调用的函数。
定时器延迟（单位为毫秒）。

jQuery的delay方法



`delay`

我会告诉一个函数反反复复运行，每次重复之前有一个时间间隔。

```
slideDown().delay(5000).slideUp();
```

运行这个效果链时，在jQuery中通常将一个效果队列。
这个例子中，delay方法会在slideDown和slideUp效果之间加入一个5秒的停顿。

很显然，不是吗？我们要使用`setInterval`，就像上一节一样。对吗？

别那么快下结论！

我们不能总那么有把握。`setInterval`通常用来调度页面上定期发生的事件，不过如果要依赖于外部资源（如我们的数据文件），可能就会出问题。



即使`setInterval`调用的函数还未结束，`setInterval`也会运行。

Watch it! 如果你在等待来自另一个服务器的信息，或者在等待用户交互，在你准备就绪之前`setInterval`可能会再次调用函数。函数可能不按原先调用的顺序返回。

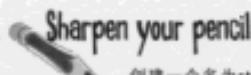
更多资源请访问稀酷客(www.ckook.com)

310

第8章

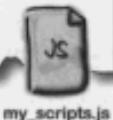
自引用函数

自引用函数会在正常操作期间调用自身。如果要等一个函数当前运行结束之后才允许它再次运行，自引用函数就特别有用。结合自引用函数和setTimeout调用，可以适当调度函数，使它只有在该函数前一次调用成功的情况下才会运行。否则，不会执行到该函数的调用代码，因此也不会再次调用。



创建一个名为startAJAXcalls的函数，每次页面加载时要调用这个函数，它会每10秒调用一次getXMLRacers函数。在脚本文件最前面定义一个变量，放在\$(document).ready函数中，名为FREQ。将这个变量设置为一个毫秒数，我们需要把这个毫秒数作为getXMLRacers函数重复调用的频率参数。使用setTimeout调用startAJAXcalls函数，使它成为一个自引用函数，getXMLRacers函数完成之后。还需要在代码中直接调用startAJAXcalls函数来启动定时器。

```
$ (document) . ready(function () {
    .....
    function startAJAXcalls(){
        .....
        .....
        .....
        .....
        .....
    }
    getXMLRacers();
    .....
    function getXMLRacers(){
        $.ajax({
            url: "finishers.xml",
            cache: false,
    
```



Sharpen your pencil

Solution

下面是动动笔练习的答案，startAJAXcalls中使用setTimeout调用getXMLRacers函数来得到XML。另外还包括了startAJAXcalls自身的调用。这个自调用可以确保上一个调用完成之后才会发生下一个调用。这样就能保证即使网络很慢，或者即使调度下一个调用之前服务器的响应尚未返回，也不会堆积服务器请求。

将FREQ变量设置为10000，因为setTimeout函数需要参数为毫秒数。

由于我们是等待上一个函数调用完成，所以使用setTimeout函数。

使用getXMLRacers函数确保页面加载时页面有内容。

```
$ (document).ready(function () {
    var FREQ = 10000;

    function startAJAXcalls() {
        setTimeout(function () {
            getXMLRacers();
            startAJAXcalls();
        }, FREQ);
    }

    getXMLRacers();
    startAJAXcalls();
}

function getXMLRacers() {
    $.ajax({
        url: "finishers.xml",
        cache: false,
    });
}
```



在setTimeout中应该调用getXMLRacers函数。

10秒后再次调用函数执行。

将这个变量作为参数传入。

调用新创建的函数执行。

there are no
Dumb Questions

问：关于Ajax，我听说需要使用XMLHttpRequest对象，这种说法对吗？

答：是的。不过jQuery不同。作为一个Web程序员，你不需要使用这个对象。使用Ajax方法时jQuery会为你做这件事。另外，由于每个浏览器的Ajax调用可能不同，jQuery会为每个网站访问者找到完成Ajax请求的最佳方法。

问：如果服务器返回一个错误，或者根本没有响应，会怎么样？会一直等下去吗？

答：不会。请求不会永远等待。可以在Ajax调用中设置一个超时时间参数。另外，类似于success事件参数，它可以运行一个函数，还有另外一些参数来处理error, complete等事件。调用ajax方法时这些事件可以设置为本地事件，或者作为全局事件来触发可能正在监听这些事件的处理程序。

更多资源请访问稀酷客(www.ckook.com)



试一试

用刚才创建的新代码更新my_scripts.js文件。另外，不要忘记在脚本最后getXMLRacers函数调用后面增加这个新函数的一个调用。然后，在浏览器中查看页面，使用Google Chrome的“Network”（网络）特性，或者Firefox Firebug的“Net”（网络）特性，查看是否每10秒调用一次文件。如果确实如此。使用你喜欢的文本编辑器，用下面所列的数据项更新XML文件，你会在页面上看到这个新的运动员（不要忘记更新之后保存XML文件）！

把这个运动员增加到
XML文件，并保存。可
以看到它会自动地加载
到你的页面中。

```
<runner>
  <fname>Justin</fname>
  <lname>Jones</lname>
  <gender>m</gender>
  <time>29:14</time>
</runner>
```

哇，看起来越来越棒了！
不过，可以使更简单的方法
得到页面的更新时间吗？



更多资源请访问稀酷客(www.ckook.com)

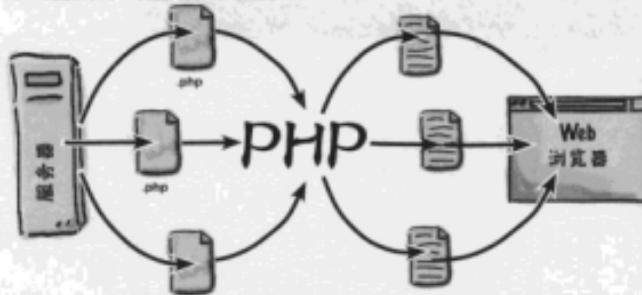
从服务器得到更多信息

到目前为止，我们已经看到，HTML很擅长在页面上显示信息，XML则擅长于格式化数据传输到页面。不过如果需要页面真正做点什么，比如明确时间，或者从数据库获取数据，该怎么做呢？当然，利用jQuery和JavaScript我们还能做更多有意思的事情，不过为什么不使用一些专门设计用来完成这些任务的工具呢？

服务器端语言来帮忙！

有很多不同类型的服务器端语言，比如JSP、ASP或Cold Fusion，不过我们只想强调一种语言：PHP。

PHP（代表PHP: Hypertext Processor。没错，这是缩略语中嵌套的一个缩略语，不要问我为什么这样取名）。这是一个免费的通用服务器端脚本语言，可以用来制作Web页面。包含PHP代码的文件在服务器上运行，将生成HTML，这些HTML再提供到浏览器显示。我们将在下一章更详细了解PHP，不过现在先来看它对我们的“更新时间”特性有什么帮助。



PHP用来动态
生成HTML，
然后在浏览器
上显示。



等等！如果没有按照第II部分安装PHP
和MySQL，你可能没办法制作下一页。
这一部分需要使用PHP。
所以在继续之前先要安装PHP。

什么时间？

必须承认，已经有一个JavaScript函数可以用来得到时间。不过，这是一个很大、很复杂的函数，对于完成这样一个简单的任务来说有些大材小用了。幸运的是，PHP为我们提供了一个很容易的获取时间的方法：可以使用date函数。就像目前创建的其他函数一样，date函数也有多个参数，可以根据你传入的参数返回不同版本的日期。其中的主要参数用于确定希望如何显示日期。

下面更仔细地分析这个函数：

```
使用PHP date函数。
date (string $format [, int $timestamp = time() ]);
```

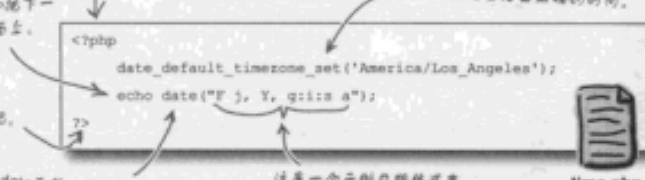
字符串 ([和]) 内的部分是可选参数。
 PHP也使用美元符。传入一个参数，作为返回日期的格式。这是一个字符串。
 不过仅用于变量。PHP各行代码都通过一个分号结尾。

date函数的完整参数列表请访问

<http://php.net/manual/en/function.date.php>

动手做！

在index.html文件所在的文件夹中创建一个新文件，命名为time.php，在这个新的time.php文件中增加以下代码。



PHP开始标记。这会告诉服务器哪里是需要解析的PHP代码。

echo命令告诉PHP把下面一部分信息写到页面上。

结束PHP标记。

使用date函数，指定特定的参数来返回与JavaScript函数同样的日期格式。

对date函数设置默认时区，从而可确保你所在的地区给出正确的时间。

time.php

这是一个示例日期格式串：
 F — 完整之本月份
 J — 0，不加前导0
 Y — 4位年份
 g — 小时 - 12小时格式
 i — 分钟，有前导0
 s — 秒，有前导0
 a — 小写am或pm



试一试

保存time.php文件后，在浏览器中打开这个文件，确保日期的格式正确。PHP代码必须通过Web服务器运行，所以URL应当是http://而不是file://。另外，要确保URL指向开发代码所在的服务器。

只需要利用一点点PHP代码，
就可以在屏幕上显示时间。比
JavaScript容易多了！



是啊，我们还没完成呢。来看看还剩下什么：

- ① 我们希望在页面上指出最近页面何时更新。
- ② 我们希望指出更新的频率。
- ③ 我们希望人们能自己选择停止和开始更新。

下面来处理这个列表中的第一项和第二项任务。因为它们是相关的，
所以这两个任务一起解决。

更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil



在index.html页面的页脚增加一个标记，ID为freq，用来指示一个名为showFrequency的新函数的结果。这个函数用于显示页面多久更新一次。另外，创建一个名为getTimeAjax的函数，它使用load方法加载time.php文件，load是jQuery为Ajax提供的一个便利方法。这个方法取一个URL为参数，将结果自动写至ID为updatedTime的div。最后，用这个新的getTimeAjax函数替换getXMLRacers中的getTime函数调用。

```
<footer>
  <h4>Congratulations to all our finishers!</h4>
  .....
  <br><br>
  Last Updated: <div id="updatedTime"></div>
</footer>
<script src="scripts/jquery-1.6.2.min.js"></script>
<script src="scripts/my_scripts.js"></script>
```



index.html

```
function ..... "Page refreshes every " + FREQ/1000 + " second(s).")
}

.

.

function ..... $(.....).load(.....);
}
```



my_scripts.js



Sharpen your pencil

Solution

现在你在index.html页面的页脚增加了一个标记，ID为freq。用来显示页面多久更新一次。你还创建了一个新的getTimeAjax的函数，它使用load Ajax便利方法加载time.php文件，然后将结果写至updatedTime div。你还更新了getXMLRacers函数，其中使用这个新的getTimeAjax函数而不是JavaScript getTime函数。

```
<footer>
  <h4>Congratulations to all our finishers!</h4>
  <span id="freq"></span> ← 用这个span元素来显示频率。
  <br><br>
  Last Updated: <div id="updatedTime"></div>
</footer>
<script src="scripts/jquery-1.6.2.min.js"></script>
<script src="scripts/my_scripts.js"></script>
```



index.html

```
function showFrequency() {
  $('#freq').html('... "Page refreshes every ' + FREQ/1000 + " second(s).");
```

↑ 创建两个新函数，一个显示频率，另一个通过Ajax从服务器得到时间。

```
function getTimeAjax() {
  $('#updatedTime').load("time.php");
```

↑ 将结果输出到底部上的updatedTime元素中。



my_scripts.js



试一试

用你刚创建的新代码更新my_scripts.js文件。另外，不要忘记在index.html文件中增加新的span元素，还要把getTime函数替换为getTimeAjax函数。



- ① 我们希望在页面上指出最近页面何时更新。
- ② 我们希望指出更新的频率。
- ③ 我们希望人们能自己选择停止和开始更新。



不过怎么让一个调用自身的函数停止呢？

这个问题有些麻烦。

我们需要修改函数，让它仅在满足某些条件的情况下才运行。



目前我们见过哪些特性能检查是否满足条件？

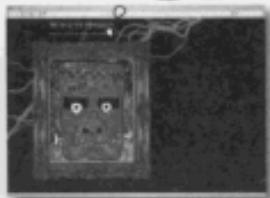
关闭页面上的调度事件

再来看第5章和第7章，我们创建了一个“怪物”函数，它使用`setTimeout`不断调用会产生闪电效果的函数。这会得到一些意想不到的结果—页面有可能失去焦点，人们返回这个应用时视觉效果会出现叠加。

不过，由于已经确定需要等到前一个函数调用完成才能运行下一个函数调用，所以不能对这些调用使用`setInterval`。

我们需要给出一个更好的答案。有没有比之前更好的方法？这里不能使用`window.onblur`和`window.onfocus`浏览器事件。因为我们不希望人们只有在离开时才能停止更新。不过，前几章中已经见过如何根据条件逻辑运行代码，所以下面也会使用条件逻辑。

查看更多怪物？



你能想到这个问题可以用哪个条件逻辑吗（提示：我们已经用过这个条件逻辑检查XML文件中运动员的性别）？

问：除了XML，Ajax能在页面中加载其他内容吗？

答：使用jQuery可以在页面中加载各种信息。正如刚才见到的，使用`load`方法可以把一个PHP文件的结果直接加载到HTML元素中。另外，还可以加载其他HTML文件、JavaScript文件、纯文本和JSON（JavaScript对象表示法）对象。下一章将介绍JSON。

there are no
Dumb Questions

问：jQuery中还为Ajax提供了其他便利方法吗？

答：jQuery为Ajax提供了5个便利方法（快捷方法）：`get`、`getJSON`、`getScript`、`post`和`load`。前4个要通过jQuery对象调用。不过`load`可以从任何元素调用—这个元素将成为返回数据的目标。

问：什么时候要使用`load`方法，什么时候使用Ajax呢？

答：`load`方法设计用来将一些特定的数据加载到一个特定位置，就像我们在`getTimeAjax`函数中所做的。Ajax方法要复杂得多，它的用途和参数也更多。这个方法可以用来加载其他信息，或者向服务器发送数据来进行处理。有关的更多内容将在下一章介绍。

更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

创建一个全局变量，名为repeat，默认值为true。再创建一个函数，点击一个新按钮（这个按钮的ID指定为btnStop）时要修改这个repeat变量。将ID为freq的span元素的HTML内容设置为“Updates paused”。另外，创建一个名为btnStart的按钮，它将repeat全局变量设置为true，如果repeat变量为true，还要调用startAJAXcalls和setTimeout函数。把这些新按钮增加到页面的页脚区。

```

$(document).ready(function(){

    .....
    var FREQ = 10000;
    function startAJAXcalls(){

        .....
        setTimeout( function() {
            getXMLRacers();
            startAJAXcalls();
        }, FREQ
        .....
    };

    $("#btnStop").click(function(){

        .....
        $("#freq").html("Updates paused");
    });

    .....
    , function(){

        startAJAXcalls();
    };
});

<footer>
    <h4>Congratulations to all our finishers!</h4>
    .....
<br>
<span id="freq"></span> <br><br>

```



index.html



my_scripts.js

Sharpen your pencil

Solution

完成后，会有一个名为repeat的变量，它将控制函数是否再次调用自身来得到XML文件完成更新。这个变量的值由btnStop和btnStart按钮控制。这两个按钮都增加到页面的页脚区。这些按钮还要设置freq span元素，根据页面是否更新显示不同的消息。

将这个变量设置为默认值true。
这样页面加载时会更新。

检查repeat变量
是否为true。

点击btnStop按钮时将这
个变量设置为false。

点击btnStart按钮时再将这
个变量设置为true。变量调
用startAJAXcalls函数并指
向再次刷新文件。

在页面的页脚区增加
这些新按钮。

```
<footer>
    <h4>Congratulations to all our finishers!</h4>
    <button id="btnStart">Start Page Updates</button>
    <button id="btnStop">Stop Page Updates</button>
    <br>
    <span id="freq"></span> <br><br>
```

```
$ (document).ready(function () {
    var repeat = true;
    var FREQ = 10000;
    function startAJAXcalls () {
        if(repeat) {
            setTimeout( function () {
                getXMLRacers();
                startAJAXcalls();
            }, FREQ
        )
    }
    $("#btnStop").click(function () {
        repeat = false;
        $("#freq").html( "Updates paused." );
    });
    $("#btnStart").click(function () {
        repeat = true;
        startAJAXcalls();
        showFrequency();
    });
});
```



my_scripts.js



index.html



试一试

用刚才创建的新代码更新my_scripts.js文件和index.html文件。然后，在你喜欢的浏览器中加载页面，确保能正常工作。试着点击这些新按钮，确认它们确实会停止Ajax请求。可以在Google Chrome的“Network”标签页或者Firebug for Firefox的“Net”标签页中查看。

2011 Race Finishers!

[Male Finishers](#) [Female Finishers](#) [All Finishers](#)

All Finishers

- Name: John Smith. Time: 25:31
- Name: Jacob Walker. Time: 25:54
- Name: Mary Brown. Time: 26:01
- Name: Jerry Pierce. Time: 26:04
- Name: Frank Jones. Time: 26:08
- Name: Bob Hope. Time: 26:38
- Name: Jane Smith. Time: 28:04
- Name: Ryan Rice. Time: 28:24

Congratulations to all our Finishers!

[Start Page Updates](#) [Stop Page Updates](#)
Page refreshes every 10 seconds.

Last Updated:
July 14, 2011, 11:08:55 pm

成功了！

现在你有了一个能够实时更新的页面（通过更新XML文件），而且允许用户开始和停止更新。

更多资源请访问稀酷客(www.ckook.com)

你现在的位置 >

323



jQuery/Ajax工具箱

你已经掌握了第8章，你的工具箱里又增加了一点PHP，一些XML，还有一堆Ajax。

Ajax

这是一种技术的组合，允许更新web页面的一部分，而无需重新加载整个页面。

对一个后台服务器做出调用，这个服务器会在返回数据之前先处理数据。

jquery通过一个`ajax`方法实现Ajax功能。

ajax()快捷方法

jquery中为Ajax提供了5个快捷方法，默认地配置有不同的参数，不过最后都会调用`ajax`方法：

- `$.get`
- `$.getJSON`
- `$.getScript`
- `$.post`
- `$.load`

XML

这是一个严格但很灵活的标记语言，用来描述数据和数据结构。可以用于存储信息，或者格式化数据进行传输。

很多常用的web技术中都用到了XML，如RSS，SOAP/web服务和SVG。

PHP

这是一个服务器端脚本语言，允许在向客户浏览器传递页面之前先处理web页面内容。

9 处理JSON数据



从XML文件读数据尽管很有用，不过并不总符合要求。利用一种更高效的数据交换格式（JavaScript对象记法，JavaScript Object Notation，简称为JSON），可以更容易地从服务器端获取数据。而且，与XML相比，JSON更容易生成和读取。通过使用jQuery、PHP和SQL，你将了解如何创建一个数据库存储信息，以便以后利用JSON获取以及使用jQuery在屏幕上显示。真正的Web应用绝对能力超凡！

更多资源请访问稀酷客(www.ckook.com)

Web镇MegaCorp公司市场部不了解XML

来自：Web镇MegaCorp公司市场部

主题：回复：第42届年度Bit to Byte长跑大赛结果页面

你们好，Web设计小组，

我们非常满意你们对网站的更新。

不过还有一个问题：我们办公室里没人懂XML！所以我们不知道如何向这个大赛网站增加新的参赛者。

我们试过，不过每次都不对。总会让网站做一些奇怪的事情…… 比如不显示参赛者，或者有些字段会从页面上消失。尽管它们都还在XML文件中。真是太怪异了。

我们希望能直接在一些方框里键入参赛者的信息。然后点击一个按钮来增加参赛者。

你能做到吗？

如果我们输入有误，你能不能保证我们的输入错误不会破坏整个网站？

我知道我们很快都要飞往夏威夷，现在只剩下3天时间了。不过我们真的希望在我们出发之前能搞定。你能及时完成吗？

Dionah C. Housney

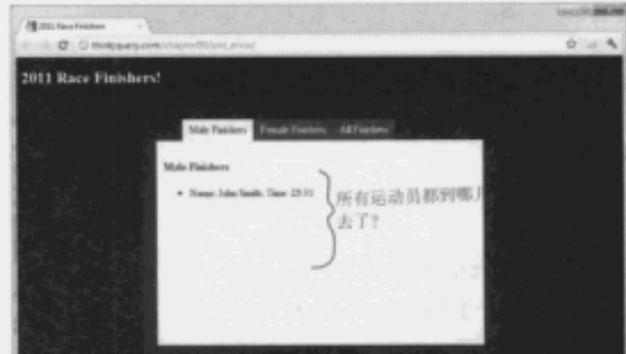
市场部首席

Web镇MegaCorp公司



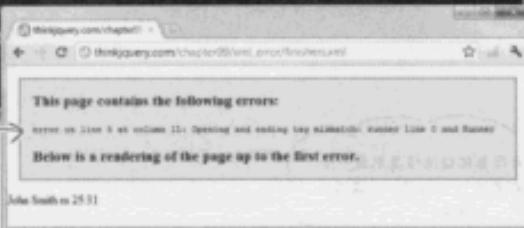
XML错误会破坏页面

如果XML中有错误，我们编写的读取和解析XML的逻辑就会失败。这些错误主要是因为标记有问题，比如一个标记没有结束，或者标记名的大小写不正确。不过，除了这些以外，如果标记中的数据未能正确编码，不能合法地在XML中使用，这些数据也可能导致XML出现问题。



Congratulations!
Last Page Updated
Page updated every

Last Updated
July 30, 2013, 7:21



John Smith vs 25:31

在Web浏览器中打开这个XML文件，
你会发现我们哪里有错误。

<runner>标记中大小
写错误。

错误的开
始标记。

```

<runners>
  <runner>
    <name>John Smith</name>
    <time>25:31</time>
  </runner>
  <runner>
    <name>Maryann O'Farrell</name>
    <gender>F</gender>
    <time>26:08</time>
  </runner>
  <runner>
    <name>Frank Johnson</name>
    <gender>M</gender>
    <time>26:08</time>
  </runner>
  <runner>
    <name>Peter Parker</name>
    <gender>M</gender>
    <time>26:08</time>
  </runner>
  <runner>
    <firstname>Smit</firstname>
    <lastname>Smit</lastname>
    <gender>F</gender>
    <time>26:08</time>
  </runner>
</runners>
  
```

另一个<runner>标记大-
小写问题。

看起来XML不太满足我们的需要。你还有什么选择可以让市场部增加新的运动员？



从Web页面收集数据

你可能已经想到可以使用一个HTML表单。利用表单，可以收集各种数据，并把它们发送给服务器进行处理。表单上可以有多个不同类型的元素，用来收集各种类型的数据。我们将在第10章更详细地介绍表单，不过现在只使用两种最基本的表单元素：文本框和下拉列表。你在表单方面可能早已经是高手了，这里只做一个简单的回顾，来了解现在要处理的问题。

```

<input> 标记告诉浏览器从
这个元素得到信息。
    ↓
<input type="text" name="txtEmail" />
    ↑
    type属性告诉浏览器显示时需如何做
    出反应。
    ↑
    这个元素的
    唯一标记。
<select> 标记告诉浏览器
显示一个下拉列表。
    ↓
<select name="truthiness">
    <option value="1">True</option>
    <option value="0">False</option>
</select>
    ↑
    元素名发送给服务
    器来处理。
    ↓
    选项信息发送给服务器。
    ↑
    option元素显示下拉
    列表中的一项选择。

```



现在最简单的方法就是创建一个
新的标签页，上面有一个表单。
对吗？

这种做法应该是可以的，而且很容易做到，因为我们已
经知道如何增加标签页。

然后可以研究如何存储/获取这个数据，在参赛者列表中
显示。

更多资源请访问稀酷客(www.ckook.com)



成品

HTML & CSS

更新index.html文件，增加一个标签页，从而可以通过表单增加新的参赛者。另外更新my_style.css文件中的数据项，让ID为main的元素更宽一些。



```

<ul class="idTable">
    <li><a href="#male">Male Finishers</a></li>
    <li><a href="#female">Female Finishers</a></li>
    <li><a href="#all">All Finishers</a></li>
    <li><a href="#new">Add New Finisher</a></li> ← 增加新的标签页，名为“Add New Finisher”。
</ul>

<div id="male">
    <h4>Male Finishers</h4><ul id="finishers_m"></ul>
</div>

<div id="female">
    <h4>Female Finishers</h4><ul id="finishers_f"></ul>
</div>

<div id="all">
    <h4>All Finishers</h4> <ul id="finishers_all"></ul>
</div>

<div id="new">
    <h4>Add New Finisher</h4>
    <form id="addRunner" name="addRunner" action="service.php" method="POST"> ← 增加一个新的HTML表单，用来收集数据并且回服务器提交数据。
        First Name: <input type="text" name="txtFirstName" id="txtFirstName" /> <br>
        Last Name: <input type="text" name="txtLastName" id="txtLastName" /> <br>
        Gender: <select id="ddlGender" name="ddlGender"> ← 这个action告诉表单要把数据发送到哪里进行处理。
            <option value="">--Please Select--</option> ← 这个方法确定数据如何发送到哪里去。
            <option value="f">Female</option>
            <option value="m">Male</option>
        </select> <br>
        Finish Time:
        <input type="text" name="txtMinutes" id="txtMinutes" size="10" maxlength="2" /> (Minutes)
        <input type="text" name="txtSeconds" id="txtSeconds" size="10" maxlength="2" /> (Seconds)
        <br><br>
        <button type="submit" name="btnSave" id="btnSave">Add Runner</button>
        <input type="hidden" name="action" value="addRunner" id="action">
    </form> ← 这是一个隐藏的HTML值，然后我们还会使用这个输入值。
</div>

```



my_style.css

```

#main {
    background:#181818;
    color:#111;
    padding:15px 20px;
    width:600px;
    border:1px solid #222;
    margin:8px auto;
}

```



试一试

在浏览器中打开index.html，选择Add New Finisher标签页，
查看增加到页面的新表单和输入域。

2011 Race Finishers!

Male Finishers Female Finishers All Finishers Add New Finisher

All Finishers

First Name:
 Last Name:
 Gender:
 Finish Time: (Minutes) (Seconds)

Add Runner

Congratulations to all our finishers!

Show Page Updates Start Page Updates
 Page refreshes every 10 second(s).

Last Updated: July 31, 2011, 10:09:14 am



现在你已经有了一个表单，可以用来收集数据，你认为该如何存储和获取数据呢？

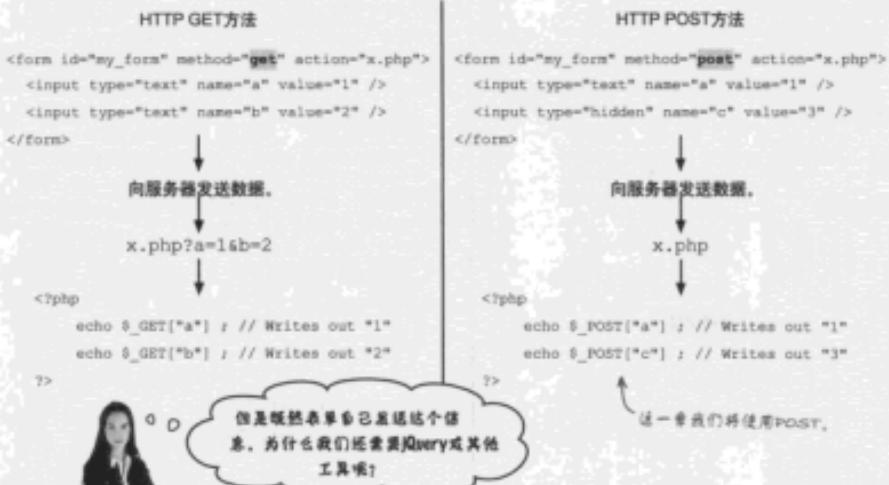
更多资源请访问稀酷客(www.ckook.com)

如何处理数据

现在要把由表单收集的数据发送到服务器，并以某种方式存储。为此，我们要用另外一种语言（PHP）将数据插入到一个数据库中。不用担心！稍后就会让你迅速了解PHP和数据库，不过先来了解如何把表单数据发送到服务器。

使用HTTP向服务器发送数据有两种方法：GET和POST。GET和POST的主要区别是数据以何种方式发送到服务器。GET将表单域名和值作为键/值对追加到URL的末尾。PHP可以从一个关联数组\$_GET[]读取这个信息。这个关联数组会在提交表单时发送到服务器。URL中?后面的参数是可见的。

POST也会发送数据，同样通过一个关联数组发送，但是编码方式不同。另外URL中的数据对最终用户不可见。\$_POST[]关联数组包含由表单元素得到的所有信息。与\$_GET[]数组类似，也是由表单元素名和值组成的一系列名/值对。



是的，表单可以发送信息……

不过，上一章我们曾经提到过jQuery和Ajax的好处，你不用重新加载整个页面就可以获取或发送数据，这样一来用户不会每次都看到整个页面重新加载。不过，在使用jQuery和Ajax向服务器发送数据之前，你要做好发送数据的准备。

更多资源请访问稀酷客(www.ckook.com)

发送数据前先格式化

向服务器发送信息之前（使用Ajax），需要先做一些准备工作，让它采用Ajax能够发送而且服务器可以理解的一种格式。为此，我们要把数据串行化为一个对象。这样Ajax调用就可以把它作为一个数据包发送。jQuery提供了两个表单辅助方法来完成数据的串行化：`serialize`和`serializeArray`。前一个方法会把所有表单输入连接起来构成一个键/值对组成的串，各个键/值对之间用&号分隔。后一个方法将创建一个键/值对关联数组，这仍是一个对象，但是与简单的`serialize`方法的结果相比，结构更为清晰。我们将介绍这两个方法，不过对于长跑数据将使用`serializeArray`方法。

serialize

```
<form id="my_form">
  <input type="text" name="a" value="1" />
  <input type="text" name="b" value="2" />
  <input type="hidden" name="c" value="3" />
</form>

$( "#my_form" ).serialize();
```

表单ID选择器

serialize方法

最终结果

a=1&b=2&c=3

serializeArray

```
<form id="my_form">
  <input type="text" name="a" value="1" />
  <input type="hidden" name="c" value="3" />
</form>
```

\$("#my_form:input").serializeArray();

表单ID选择器，后面是HTML元素输入过滤器。这就告诉过滤器只要找类型为“input”的HTML元素。
调用serializeArray方法。

最终结果

```
[
  {
    name: "a",
    value: "1"
  },
  {
    name: "c",
    value: "3"
  }
]
```

向服务器发送数据

jQuery提供了一个快捷方法post，这个方法专门用来向服务器发送数据。post方法有多个参数，包括一个URL（将向这个URL发送信息）、想要发送的信息，还有一个处理函数，POST完成时将运行这个处理函数。

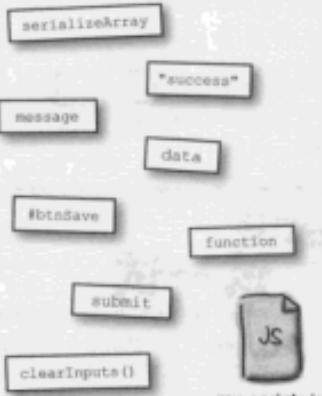
```
jquery快  
速方法。  
$.post(url_to_send, data, function(json) {  
    // URL, 数据是  
    // 要发送的数据。  
    // 已经串行化。  
    // 运行这个  
    // 回调函数。  
    // 送回的数据，现在一个名为  
    // JSON的对象中。现在不用  
    // 担心这个对象。字符串后面还  
    // 会介。  
});
```



jQuery代码磁贴

在#btnSave上创建一个点击事件监听者，得到表单中的所有数据，并串行化。然后使用jQuery post方法将这些信息发送到服务器。从表单的action属性得到数据的目标URL。另外，创建一个clearInputs函数。如果post成功，则将所有表单域的值都设置为空，还需要利用表单上ID为addRunner的submit监听者取消表单的默认提交动作（返回false）。

```
$("____").click(function() {  
    var data = $("#addRunner :input").____();  
    $.post($("#addRunner").attr('action'), _____, _____(json) {  
        if (json.status == "fail") {  
            alert(json._____);  
        }  
        if (json.status == _____) {  
            alert(json.message);  
            clearInputs();  
        }  
    }, "json");  
});  
function _____ {  
    $("#addRunner :input").each(function() {  
        $(this).val("");  
    });  
}  
$("#addRunner").____(function() {  
    return false;  
});
```





jQuery代码磁贴答案

在#btnSave上创建一个点击事件监听者。得到表单中的所有数据，并串行化。然后使用jQuery post方法将这些信息发送到服务器。从表单的action属性得到数据的目标URL。另外，创建一个clearInputs函数。如果post成功，则将所有表单域的值都设置为空。还需要利用表单上ID为addRunner的submit监听者取消表单的默认提交动作（返回false）。

```

$( "#btnSave" ).click(function() {
    var data = $( "#addRunner :input" ).serializeArray();
    $.post( $( "#addRunner" ).attr("action"), data, function (json) {
        if (json.status == "fail") {
            alert(json.message);
        }
        if (json.status == "success") {
            alert(json.message);
            clearInputs();
        }
    }, "json");
});

function clearInputs() {
    $( "#addRunner :input" ).each(function() {
        $(this).val("");
    });
}

$( "#addRunner" ).submit(function() {
    return false;
});

```

使所有表单值做好转换并返回到服务器的准备。

得到想要提交的表单的action属性。

检查服务器的返回值（设置在PHP代码中），来查看POST是否成功。

使用一个HTML元素过滤器访问表单中的所有输入值，并将它们都设置为空。

取消表单的默认提交动作，当按钮点击事件中的jQuery代码负责发送数据。



my_scripts.js



试一试

增加了这些内容之后，你的页面看上去并没有不同。不过，要用刚才创建的代码更新my_scripts.js文件。然后在一个浏览器中打开index.html页面，打开“Network”标签页（Chrome）或“Net”标签页（Firebug），每次按下btnSubmit按钮时应该能看到对service.php文件的POST请求。Headers（头部）标签页Request Method（请求方法）部分中会列出一个POST，另外还会列出Form Data（表单的数据）。现在只需要一个地方放置这些数据……

更多资源请访问稀酷客(www.ckook.com)

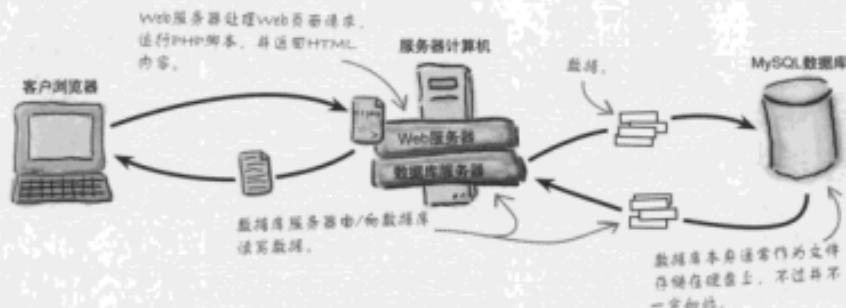
将数据存储在MySQL数据库中

关系数据库管理系统（Relational Database Management Systems, RDBMS）是极有条理的一种应用，这些应用设计用来存储、组织和记住各部分数据之间的关系。

RDBMS通常称为数据库服务器，它们的结构不同、规模不同（当然价格也不同）。在这里我们将使用一个免费的数据库服务器，名为MySQL。你要用数据库服务器能理解的语言与它通信，这就是SQL。数据库服务器通常在Web服务器旁运行，有时还可能在同一个服务器上，它们相互协调来读写数据和发布Web页面。

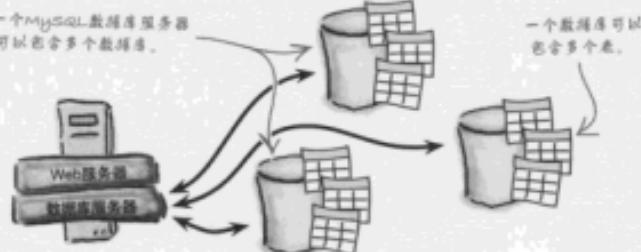
MySQL中的“SQL”代表结构化查询语言（Structured Query Language）。

MySQL将数据存储在数据库表中。



MySQL数据库组织为数据库表，表通过相关数据的行和列存储信息。大多数Web应用会使用一个数据库中的一个或多个表，这有点像一个文件柜里的不同文件夹。

数据本身通常认为存储在硬盘上，不过并不一定如此。



SQL是用来与MySQL数据库通信的查询语言。

创建数据库存储运动员信息



嘿！你配置了MySQL和PHP吗？继续下一步之前一定要按附录II的说明安装和配置PHP和MySQL。

好了，继续吧。现在你能完成这一章的工作了。



成品
SQL

为了建立数据库、表和用户，我们已经为你编写了SQL。打开MySQL Workbench，再打开一个新的连接，运行以下SQL。

```
create database hfjq_race_info; ← 创建一个名为hfjq_race_info的数据库。
```

```
CREATE USER 'runner_db_user'@'localhost' IDENTIFIED BY 'runner_db_password';
GRANT SELECT, INSERT, UPDATE, DELETE ON hfjq_race_info.* TO 'runner_db_user'@'localhost';
```

```
use hfjq_race_info; ← 告诉脚本下一部分是你
                           的新数据库名。
```

创建一个用户，名为runner_db_user，为它提供一个登录密码，使这个用户能够获取、设置、更新和删除数据库中的数据。

```
CREATE TABLE runners {
    runner_id INT not null AUTO_INCREMENT,
    first_name VARCHAR(100) not null,
    last_name VARCHAR(100) not null,
    gender VARCHAR(1) not null,
    finish_time VARCHAR(10),
    PRIMARY KEY (runner_id)
};
```

创建一个名为runners的表，其中包含我们想要记住的长跑运动员信息。

更多资源请访问稀酷客(www.ckook.com)



试一试

打开MySQL Workbench，并打开一个服务器连接。将上一页的SQL代码粘贴到查询面板，点击闪电图标运行这些SQL代码。你会在下方的输出面板中得到成功消息。

```

CREATE DATABASE race_info;
CREATE USER 'runner_00_user'@'localhost' IDENTIFIED BY 'runner_00_password';
GRANT SELECT,INSERT,UPDATE,DELETE ON race_info.* TO 'runner_00_user'@'localhost';
USE race_info;
CREATE TABLE runner(
    runner_id INT NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(20) NOT NULL,
    last_name VARCHAR(20) NOT NULL,
    gender VARCHAR(10) NOT NULL,
    TIRIIN_CODE VARCHAR(10),
    PRIMARY KEY (runner_id)
);

```

Actions	Time	Action	Result	Duration / Fetch Time
1	22:36:02	create database race_info	0 rows affected	0.000 sec
2	22:36:02	CREATE USER 'runner_00_user'@'localhost' IDENTIFIED BY 'runner_00_password'	0 rows affected	0.001 sec
3	22:36:02	GRANT SELECT,INSERT,UPDATE,DELETE ON race_info.* TO 'runner_00_user'@'localhost'	0 rows affected	0.000 sec
4	22:36:02	use race_info	0 rows affected	0.000 sec
5	22:36:02	CREATE TABLE runner(runner_id INT NOT NULL AUTO_INCREMENT);	0 rows affected	0.001 sec



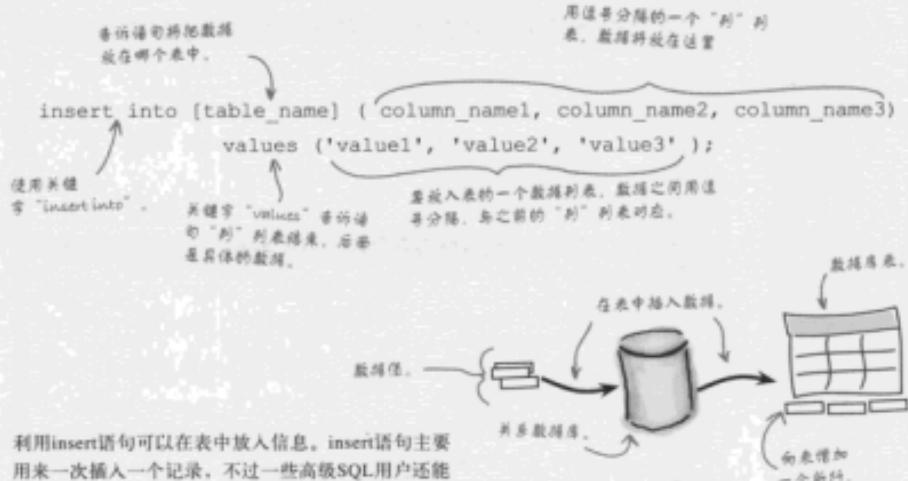
接下来就会谈到这个问题。

你刚才运行的SQL创建了一个数据库，创建了一个用户，并授权这个新用户访问这个数据库，另外还创建了一个表来存储运动员信息。下面来看如何在其中放入一些数据。

insert语句剖析

有一种方法可以在数据库中放入数据，另一种方法可以修改/更新数据，还有一种方法可以再次取回数据。稍后我们将介绍如何获取数据，不过现在先重点介绍如何在数据库表中放入数据。

要把数据放入数据库表，需要使用一个insert语句。



利用`insert`语句可以在表中放入信息。`insert`语句主要用来一次插入一个记录，不过一些高级SQL用户还能创建一些复杂的`insert`语句，可以向表中增加多行数据。不过，对我们来说，还是采用这种一次插入一个记录的语法。

建议按照你希望的插入数据的顺序来指定列，不过并不要求一定如此。如果没有指定列，可能会导致数据问题，因为它会自动将第一个值放在第一列，第二个值放在第二列，依此类推。你要对数据表非常了解才能有信心使用这个方法。



列名和值的顺序很重要！

值必须完全按照列的顺序列出。数据库就是利用这一点才能知道数据放在哪里。



编写SQL Insert语句，将XML文件中已有的数据插入到数据库中。在之前创建的runner表中一次插入一个记录。我们已经为你完成了第一条语句。

```
insert into runners (first_name, last_name, gender, finish_time)
values ('John','Smith','m','25:31') ;
```



既然已经编写了所有必要的SQL语句，可以将运动员插入到数据库表中，下面打开MySQL Workbench，运行你的代码。

```
insert into runners (first_name, last_name, gender, finish_time)
values ('John', 'Smith', 'm', '25:31') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Jacob', 'Walker', 'm', '25:54') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Mary', 'Brown', 'f', '26:01') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Jenny', 'Pierce', 'f', '26:04') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Frank', 'Jones', 'm', '26:08') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Bob', 'Hope', 'm', '26:32') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Jane', 'Smith', 'f', '28:04') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Ryan', 'Reese', 'm', '28:24') ;

insert into runners (first_name, last_name, gender, finish_time)
values ('Justin', 'Jones', 'm', '29:14') ;
```



现在我们已经把运动员信息放入数据表。不过，如何再把它取出来，在Web应用中使用呢？

来认识一个新语言：PHP。

别担心！我们会为你提供充分的PHP代码来完成所需的全部服务器端通信（包括与数据库服务器通信），非常简单。

更多资源请访问稀酷客(www.ckook.com)

使用PHP访问数据

PHP是一种编程语言，它需要一个运行环境：提供PHP支持的Web服务器。PHP脚本和依赖于这些脚本的Web页面必须放在一个真正的Web服务器上，而不能只是从一个本地文件系统直接打开脚本。

如果你確實在本地安装了一个Web服务器，而且提供PHP支持，就可以在你的本地计算机上直接测试PHP脚本。

Web浏览器对PHP一无所知，因此，无法运行PHP脚本。



与HTML Web页面不同，Web页面可以在Web浏览器上本地打开，PHP脚本则必须从Web服务器通过一个URL“打开”。

这个PHP脚本对Web浏览器来说只是一堆毫无意义的乱码。

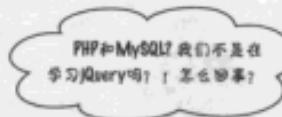
Web服务器理解这个PHP代码，并运行脚本：



提供PHP支持的Web服务器能够运行PHP脚本，并将它们转换为浏览器能理解的HTML Web页面。

若想知道一个Web页面是否由Web服务器提供（而不是本地提供），有一种简便方法：查看URL是否以“http://”开头。作为本地文件打开的Web页面总是以“file://”开头。

PHP脚本必须在一个Web服务器上运行，否则将无法运行。



我保证，会有jQuery的。

不过，首先来看如何让我们的PHP文件处理POST数据，以便将数据写入数据库。我们还会讨论向服务器发送信息时要记住的一些重要问题。



在服务器上处理POST数据

我们已经介绍过一个特殊对象：`$_POST`对象。可以创建这个对象来处理从浏览器表单向服务器传送信息。这是一个关联数组，其中包含发送的所有信息，使用HTML元素的名（而非ID）作为关联数组的键，HTML元素中的信息作为关联数组的值。服务器上的PHP代码会读取`$_POST`对象，并确定已向服务器发送哪些信息。

可以利用数组发送的键（HTML元素名）从数组中取回信息。这会在PHP脚本中返回值。

```
echo $_POST["txtFirstName"];
```

将值写到屏幕上。

收集表单数据的HTML元素的名。

自动创建的数据名，用来处理使用POST方法发送到PHP文件的数据。

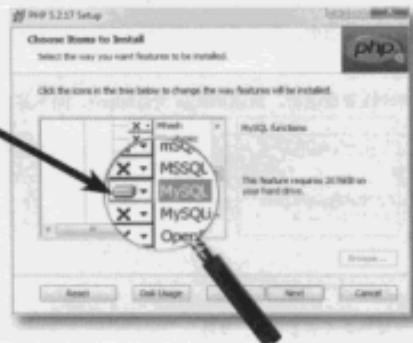


嘿，接下来要做什么？
我想去海滩！

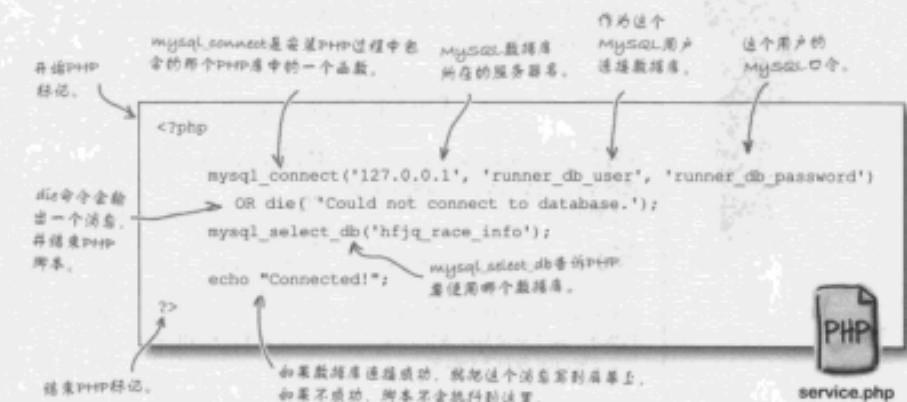
就快要完成了，要从数据库取回数据，并确定如何在Finishers列表中显示。不过，首先我们需要再来一点PHP代码连接数据库……

用PHP连接数据库

安装PHP时，快结束的时候你选择了一个特定的库，还记得吧？



这个库允许PHP与MySQL数据库通信。我们将使用这个库连接之前创建的数据库，来读出运动员数据。

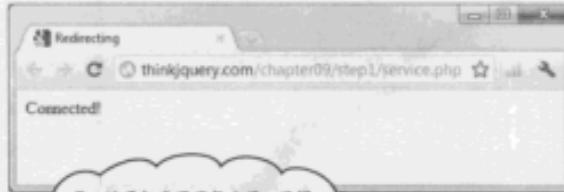




试一试

打开你喜欢的文本编辑器，增加上一页的代码。将这个文件保存为service.php，并放在这一章index.html文件所在的同一个目录中。在浏览器中打开service.php，查看数据库查询的结果。

不要忘记，PHP代码必须通过Web服务器运行，所以URL应当是http://，而不是file://。





老，这看起来可不怎么样。没错，
我们确实可以连接……不过我还
是看不到任何数据！

你说的没错。

类似于使用INSERT插入数据，读回数据也有一种特殊的话语法。
下面来看如何做到。

*there are no
Dumb Questions*

问：MySQL Workbench是与MySQL数据库交互或管理MySQL数据库的唯一途径吗？

答：当然不是！还有很多其他途径和工具。PHPMyAdmin是一个常用的基于Web的工具，可以用来管理MySQL数据库。还可以使用终端窗口从命令行管理MySQL数据库和数据。

问：还有哪些PHP库？

答：还有很多不同的PHP库，用于各种不同的用途。包括SSL、邮件(SMTP或IMAP)、压缩、认证、其他数据库连接等等。在你喜欢的搜索引擎中键入“PHP libraries”（PHP库），就会看到可用的PHP库的一个列表。

使用select从数据库读取数据

要从数据库读取数据，可以使用select语句，数据会返回到一个结果集中。结果集就是select查询中请求的所有数据的一个集合。使用select语句还可以连接多个表，这样就可以从多个表中获取数据放在同一个结果集中。

要了解有关PHP、SQL、数据仓库和表的更多信息，可以找一本《Head First PHP & MySQL》看看。

```

    select column_name1, column_name2 from table_name order by column_name1 asc
    ^           ^           ^
    "select" 关键字放在语句最前面。
    "from" 关键字告诉语句从哪个表
    获得数据。
    "order by" 关键字告诉语句如何对结果排序
    (asc表示升序, desc表示降序)。
  
```

道名分隔的一个“列”列表。
我们希望从这些列获取数据。

告诉语句从哪个表
获取数据。

“asc”关键字的“order by”
如何对结果排序(asc表示升
序, desc表示降序)。

SQL select语句从一个或多个表获取数据列，并返回一个结果集。



Sharpen your pencil

要在网站上显示运动员，创建一个select语句来获取我们需要的数据。需要从runners表读出first_name、last_name、gender和finish_time。对这些数据排序（按finish_time从低到高排序）。如果需要，可以再看看第336页，回顾这个表如何创建，以便得到正确的列名。



Sharpen your pencil

Solution

你已经创建了SQL语句从数据库获取运动员数据。

希望选择的“列”列表。

希望对数据如何排序。

`SELECT first_name, last_name, gender, finish_time FROM runners ORDER BY finish_time ASC`

希望从哪个表获取数据。



试一试

使用MySQL Workbench运行你的select语句，查看结果集中返回的所有数据。

The screenshot shows the MySQL Workbench interface. On the left is the 'Object Navigator' pane with a tree view of database structures. In the center is the 'Results' pane displaying a table titled 'runners'. The table has columns: id, first_name, last_name, gender, and finish_time. The data shows several rows of runner information, such as 'John', 'Jane', 'Mike', etc., with their respective genders and finish times.

太棒了。我可以在Workbench中看到数据了。不过难道不是要在Web页面上显示这些数据吗？



不错，确实如此。

下面来看如何从数据库获取信息在页面上显示。

用PHP获取数据

到目前为止，我们已经见过一些非常基本的PHP和一些不那么简单的PHP代码。我们了解了如何将一些基本信息写到屏幕上，以及如何连接到一个数据库，还编写了一个select语句从数据库获取信息。现在来看如何从数据库获取信息，并把这个信息写到屏幕上。



PHP代码磁贴

重新组织下面的磁贴，完成PHP代码来创建一个名为db_connection的函数。这个函数将管理数据库连接。另外创建一个\$query变量，将这个变量的值设置为之前编写的select语句（从数据库获取所有运动员）。然后创建一个\$result变量，它将调用这个db_connection函数，将\$query变量作为参数传入。最后，使用一个while循环，循环处理结果集中的各行（结果集是一个关联数组），并写到屏幕上。

```
<?php

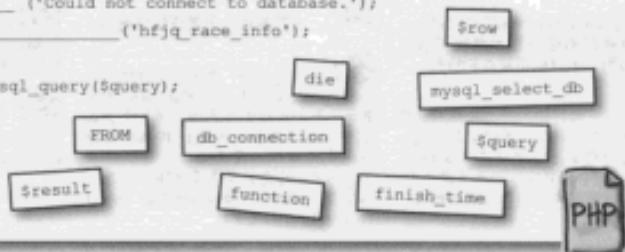
    $query = "SELECT first_name, last_name, gender, finish_time ____ runners
order by _____ ASC ";
    $result = _____ ($query);

    while ($row = mysql_fetch_array(_____, MYSQL_ASSOC)) {
        print_r(_____);
    }

    _____ db_connection(_____) ;
    mysql_connect('127.0.0.1', 'runner_db_user', 'runner_db_password')
        OR _____ ("Could not connect to database.");
        _____ ("hfhjq_race_info");

    return mysql_query($query);
}

?>
```



更多资源请访问稀酷客(www.ckook.com)



PHP代码磁贴答案

只需要利用一点点PHP，现在可以从数据库获取数据，并通过一个数据得到返回的结果，从而可以在Web页面上显示。

```
<?php

    $query = "SELECT first_name, last_name, gender, finish_time
              FROM runners
          ORDER BY finish_time ASC";
    $result = db_connection($query);

    while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
        print_r($row);
    }

    function db_connection($query) {
        mysql_connect('127.0.0.1', 'runner_db_user', 'runner_db_password')
            OR die ('Could not connect to database.');
        mysql_select_db('hfjq_race_info');

        return mysql_query($query);
    }
?>
```



service.php

问：这么说，select是不是只能从一个表中选择所有信息？我发现可以限制返回的列，不过能不能限制行呢？

答：可以，对返回的行也能限制，只需要使用where子句。第11章还会进一步讨论这个内容，不过现在应该知道，可以在where子句中输入一个过滤条件，在select语句中只返回与条件匹配的行。

问：一次只能从一个表获取数据吗？

答：不是这样的，查询中可以根据你的需要连接多个表，通常在where子句中由一个公共标识符连接。将多个表连接在一起可能会大大减慢数据库查询的速度，所以一定要当心。有关这个内容的更多信息，请参考《Head First PHP & MySQL》的第8章或者《Head First SQL》的第2章。

问：127.0.0.1上是什么数据库？我看到我的网站在“localhost”上。有什么区别吗？

答：问得好，它们没有任何区别。127.0.0.1和localhost都是指同一个东西，你目前正在用的计算机/服务器。



试一试

用刚才创建的代码更新service.php文件，然后在浏览器中打开，查看数据库查询的结果。不要忘记，PHP代码必须通过web服务器运行，所以URL应当是http://，而不是file://。

```

Array ( [first_name] => John [last_name] => Smith [gender] => m [finish_time] => 25.31 ) Array ( [first_name] => Jacob [last_name] => Walker [gender] => m [finish_time] => 25.54 ) Array ( [first_name] => Mary [last_name] => Brown [gender] => f [finish_time] => 26.01 ) Array ( [first_name] => Jenny [last_name] => Pierce [gender] => f [finish_time] => 26.04 ) Array ( [first_name] => Frank [last_name] => Jones [gender] => m [finish_time] => 26.08 ) Array ( [first_name] => Bob [last_name] => Hope [gender] => m [finish_time] => 26.38 ) Array ( [first_name] => Jane [last_name] => Smith [gender] => f [finish_time] => 28.04 ) Array ( [first_name] => Ryan [last_name] => Rice [gender] => m [finish_time] => 28.24 ) Array ( [first_name] => Justin [last_name] => Jones [gender] => m [finish_time] => 29.14 )
  
```

将所有数据写到
屏幕上的最好
结果。



嗯，确实是数据没错，不过看起来太
复杂，太乱了。能不能做些处理，让
它漂亮一点？

完全可以。

我们看到的只是写到屏幕的一堆数组。它们确实包含我们需要的数据，不过格式可不是我们想要的。幸运的是，我们可以利用一种高效的方法组织数据，这种格式非常适合定义数据结构。

更多资源请访问稀酷客(www.ckook.com)

JSON来拯救！

JSON是JavaScript对象记法（JavaScript Object Notation）的缩写，这是一种轻量级数据交换格式。JSON格式对人们来说很容易读写，对计算机则很容易解析和生成。正因如此，JSON非常适合建立数据结构和传输数据。JSON建立在JavaScript标准的一个子集基础上，它与语言无关，这说明JSON可以用于几乎任何编程语言。JSON传输数据比XML更高效，类似于关联数组，它也基于名/值对。JSON中的值可以是字符串、数字、数组、对象、布尔值（true或false）或null。

这会很方便。
不是吗？

除了根元素，元素之间没有任何逻辑连接。正确的说，必须使用“find()”查找各个元素。

XML 与 JSON

```
<?xml version="1.0" encoding="utf-8"?>
<books>
  <book>
    <title>The Color of Magic</title>
    <author>Terry Pratchett</author>
    <year>1983</year>
  </book>
  <book>
    <title>Mort</title>
    <author>Terry Pratchett</author>
    <year>1987</year>
  </book>
  <book>
    <title>And Another thing...</title>
    <author>Eoin Colfer</author>
    <year>2009</year>
  </book>
</books>
```

标记有多个副本，这会增加需要传输的数据量。

根元素。

{
 books:
 book:
 title:'The Color of Magic',
 author:'Terry Pratchett',
 year:1983
 },
 {
 title:'Mort',
 author:'Terry Pratchett',
 year:1987
 },
 {
 title:'And Another thing...',
 author:'Eoin Colfer',
 year:2009
 }
}
各键数据的数组。
名/值对，用冒号分隔。
各个属性用逗号分隔。
字符串值包围在引号中。
数字不需要引号。

jQuery + JSON = 超级棒

由于JSON如此流行，而且很容易使用，开发jQuery的好人们建立了一个特殊的快捷方法，专门处理JSON数据的获取：getJSON方法。

```
jQuery快捷方法。           使用getJSON方法。           运行这个回调函数。
$.getJSON(url_to_load, function(json) {
    ↑
    URL, 希望从这里加载
    数据
}) ;                         ↑
                                返回的数据，放在一个名为json
                                的对象中（稍后会详细介绍）。
```

看起来可能很熟悉，因为这与之前从表单获取数据使用的post方法几乎完全相同。这个简单的方法是ajax方法的一个快捷方式，已经为你设置了很多参数。如果不用快捷方法调用，原本会是这样：

```
$.ajax({
    url: url_to_load,
    dataType: 'json',
    data: json,
    success: function(json) {

    };
});
```

不过我们的数据并没有采用JSON格式，它们只是一系列数组。能将这些数组转换为JSON吗？



可以！

真是很幸运，创建PHP的人已经想到了这一点。下面再来了解一些PHP基本规则，然后介绍如何结合这些规则和其他PHP函数将数据转换为JSON格式。

一些PHP规则……

说实在的，没有人会真的喜欢一大堆编码规则，不过关于PHP还有几点要强调。其中很多都有关于语法，必须了解这些规则，才能帮助你为jQuery整理数据。好在我们已经看到其中很多概念都与JavaScript有关，所以这里会尽可能简短，力求简单……

PHP基本规则

- 所有PHP代码都需要用<?php和?>标记包围。
- 可以混合使用PHP和HTML，只需在PHP代码周围使用<?php和?>标记。
- 所有PHP代码行都必须用一个分号();结束。

```
<div><span> Hello
<?php
    echo "Bob";
?>
</span></div>
```

变量规则

- 所有变量都必须用一个美元符(\$)开头。
- 在美元符后面，必须包含至少一个字母或下划线，然后是字母、数字或下划线的任意组合。
- 短横线(-)、空格()和所有特殊字符（除\$和_外）都不允许出现在变量名中。

```
<?php
$a = "USA"; // OK
$bhome_country = "Ireland"; // OK
$canother-var = "Canada"; // Causes an error
?>
```

循环规则

- PHP也包含for、while和do…while循环—同样使用JavaScript的语法。
- PHP还包含一个额外的循环机制，名为foreach循环，它会使用as关键字逐个地遍历数组中的所有元素，直至到达末尾，然后自动停止。

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo $i;
}
while ($j <= 10) {
    echo $j++;
}
$a = array(1, 2, 3, 17);
foreach ($a as $v) {
    echo "Current value: $v.\n";
}
?>
```

更多PHP规则……

还有一些规则可以帮助我们得到需要的数据，对它们正确地完成格式化，并放在Web页面上。

数组规则

1. 可以使用array关键字创建新数组，这与JavaScript中类似。
2. 类似于JavaScript，可以使用数组元素的索引访问数组值，索引放在中括号([])中。与JavaScript一样，索引同样是从0开始。
3. 数组还可以是关联的，这说明可以使用键访问数组中的元素，而不是使用索引。这称为键/值对。
4. 要为关联数组中的键赋一个值，可以使用 => 操作符。

```
<?php
$my_arr2 = array('USA', 'China',
'Ireland');
echo $my_arr2[2]; // Prints "Ireland"

$arr = array("foo" => "bar", 12 => true);
echo $arr["foo"]; // Prints "bar"
echo $arr[12]; // Prints true
?>
```

条件规则

1. if语句与JavaScript中的语法相同，else子句以及else if子句也采用JavaScript同样的语法。
2. 比较操作符都与JavaScript中相同。
3. 逻辑操作符也与JavaScript中相同，不过还增加了描述字and、or和not，可以使用这些描述字取代操作符。

```
<?php
if ($x > $y) {
    echo "x is greater than y";
}
elseif ($x == $y) {
    echo "x is equal to y";
}
else {
    echo "x is smaller than y";
}
?>
```

写屏幕的规则

1. 关键字echo和print都可以写到屏幕上。
2. 可以使用print_r命令写出一个数组的内容。

```
<?php
echo "Bob";
print_r($my_arr2);
?>
```

使用PHP格式化输出

OK，既然已经了解这些规则，下面来看PHP能为我们做什么！PHP的json_encode函数可以将一个关联数组转换为JSON编码的值串。

将值塞进到调用这个文件的任何对象。也就是说，这可以是一个浏览器，或者是一个ajax请求等。

调用这个PHP函数，用JSON样式对数据编码。

传入一个普通的数组。

```
echo json_encode(array_name);
```

不过，在对数据编码之前，数据必须在一个关联数组中。我们已经见过一种方法可以循环处理结果集，并查看其中的各个关联数组。现在我们需要一种方法取出各个数组，将它们结合到一个数组中。通过使用PHP函数array_push，可以将新元素增加到数组的末尾。

创建一个新的空数组。

传入想增加到数组的所有信息，在这里，是把另一个关联数组增加到my_array数组。

```
$my_array = array();
```

```
array_push($my_array, array('my_key' => 'my_val'));
```

调用array_push函数，并提供参数。

目标数组内部一个参数传入。

增加到这个数组的一个键/值对。



Geek Bits

json_encode函数只在PHP 5.2版本中才可用。如果你使用的是之前的某个版本，可以更新你的PHP版本，或者在你喜欢的搜索引擎中键入“json_encode PHP alternatives”，你将了解发明PHP的人们是如何创建这个函数的。这样一来，就能创建你自己的函数，充分利用它的优秀特性。

there are no
Dumb Questions

问：是创建jQuery的人提出的JSON吗？

答：不是。实际上是Douglas Crockford（Yahoo!的一位JavaScript建构师）发明了JSON，他把它称为“fat-free alternative to XML”。至于为什么这么取名，他在<http://www.json.org/fatfree.html>给出了解释。

问：JSON不就是JavaScript吗？

答：也是，也不是。JSON建立在JavaScript的一个子集之上，也就是ECMA 262第3版，不过很多语言都可以用它来完成数据传输。要查看哪些语言可以使用JSON，请访问<http://www.json.org/>。

问：这么说来，既然JavaScript和PHP有类似的语法，为什么不能直接使用JavaScript来完成我要做的事情呢？

答：正如前面提到的，PHP是一个服务器端脚本语言，可以代表你与Web服务器和数据库交互。代码在服务器上执

行，生成HTML，然后再把生成的HTML发送到客户端。与它不同，JavaScript只在浏览器上运行。它只与客户端交互。

问：好吧，再问一下，PHP是什么？

答：PEP（PHP: Hypertext Preprocessor的递归缩略语）是一种广泛使用的开源、通用的脚本语言，尤其适合完成web开发。可以嵌入在HTML中。

问：那么，PHP是从哪里来的呢？

答：问得好。PHP最早出现在1994年。它是Rasmus Lerdorf创建的，当时用它来显示他的在线简历。他在1995年6月发布了源代码。这使得其他开发人员可以更新和修正PHP的bug。在此以后，PHP得到飞速发展，已经在全世界超过2000万个网站中使用。



你刚才在大脑里灌入了有关PHP、MySQL和JSON的一些新知识。下面我们要完成一个大练习，把所有这些内容联系在一起，所以现在稍事休息，来杯咖啡，出去走走，或者做点其他事情让你的大脑放松一下。为接下来的工作做好准备。如果休息好了，就可以翻开下一页，投入下一步的工作吧。



Long Exercise

更新my_scripts.js文件，增加新函数getDBRacers，它会调用service.php文件。这个调用应当返回一个JSON对象，然后指示返回的运动员个数。另外还要更新startAJAXCalls定时器来调用这个新函数，而不是getXMLRunners函数。接下来更新service.php，从数据库发回获取到的运动员数据，并用JSON编码。

```

function startAJAXcalls() {
    if(repeat) {
        setTimeout( function() {
            -----
            startAJAXcalls();
        }, FREQ
    );
}
}

function getDBRacers() {
    $.getJSON('.....', function(....) {
        ....(json.runners.....);
    });
    getTimeAjax();
}

```



my_scripts.js

```

<?php

$query = "SELECT first_name, last_name, gender, finish_time FROM runners
order by finish_time ASC ";
$result = _____($query);

$runners = array();

while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
    _____($runners, array('fname' => $row['first_name'], 'lname' =>
$row['last_name'], 'gender' => $row['gender'], 'time' => $row['finish_time']));
}

echo _____(array("runners" => _____));
exit;

function db_connection($query) {
    mysql_connect('127.0.0.1', 'runner_db_user', 'runner_db_password')
        OR die(fail('Could not connect to database.'));
    mysql_select_db_____;
}

return mysql_query($query);
}

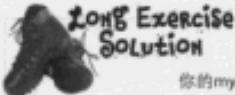
function fail($message) {
    die(json_encode(array('status' => 'fail', 'message' => $message)));
}

function success($message) {
    die(json_encode(array('status' => 'success', 'message' => $message)));
}
    
```

?>



service.php



Long Exercise Solution

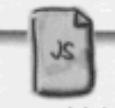
你的my_scripts.js文件现在有了一个新函数，名为getDBRacers。它调用service.php文件。原来的getXMLRunners函数已经不再需要，所以可以将它删除。这个新函数接收从service.php文件返回的JSON数据，并指示返回的运动员个数。startAJAXCalls定时器函数也得到更新来调用这个新函数。service.php文件已经更新为发回从数据库获取到的运动员数据，采用JSON编码，并按finish_time从低到高排序。

```
function startAJAXcalls() {
    if(repeat) {
        setTimeout( function() {
            getDBRacers(); // 根据计划调用这个新函数。
        }, FREQ
    );
    }
}
// 使用getJSON调用返回的数据
```

使用getJSON方法
调用service.php文件。

```
function getDBRacers() {
    // 使用getJSON方法调用service.php，返回JSON数据。
    $.getJSON("service.php", function(json) {
        alert(json.runners.length); // 其他数组类似，这个数组也有一个length属性。
        getTimeAjax();
    });
}
```

JSON对象包含一个名为
runners的数据。这个名字
让PHP中的json_encode方法
得来。



my_scripts.js

```

<?php
    // 提取运动员的数据库查询语句。
    $query = "SELECT first_name, last_name, gender, finish_time FROM runners
    ORDER BY finish_time ASC";
    $result = db_connection($query);

    // 创建一个新数组来包含运动员的值。
    $runners = array();
    // 将处理结果集，返回关联数组。
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
        $array_push($runners, array('fname' => $row['first_name'], 'lname' =>
        $row['last_name'], 'gender' => $row['gender'], 'time' => $row['finish_time']));
    }
    echo json_encode(array("runners" => $runners));
    exit;

    function db_connection($query) {
        mysql_connect('127.0.0.1', 'runner_db_user', 'runner_db_password')
        OR die("Could not connect to database.");
        mysql_select_db('hfjq_race_info');
        // 处理数据库连接的函数。
        return mysql_query($query);
    }

    // 处理脚本中错误和成功情况的处理函数。
    function fail($message) {
        die(json_encode(array('status' => 'fail', 'message' => $message)));
    }

    function success($message) {
        die(json_encode(array('status' => 'success', 'message' => $message)));
    }
?>

```



service.php



试一试

用刚才创建的代码更新service.php和tiny_scripts.js文件，在浏览器中打开index.html。打开开发工具中的“Network”标签页，可以看到JSON信息已经载入。

这个品质指出JSON对象中返回的数据量。

浏览器工具的“Network”部分中，可以在JSON标签页看到JSON信息。

不错。需要像处理XML那样使用find和each显示这些数据吗？

没必要。

我们很清楚现在数据采用的是JSON格式，因为这些数据是我们自己的PHP代码创建的。之前一直在讲我们要介绍JSON对象，现在（终于）可以进一步分析JSON对象取回数据了。

更多资源请访问稀酷客(www.ckook.com)

访问JSON对象中的数据

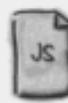
利用PHP中的`json_encode`函数，可以把关联数组转换为JSON编码的值串。这些值还可以再作为关联数组用JavaScript访问，所以我们可以循环处理关联数组，采用与其他数组类似的方式与关联数组交互。

使用XML时，必须扫描数据，查找下一个运动员。一旦找到一个运动员，还必须查看这个运动员是男是女。还记得从`json_encode`返回的JSON对象吗？我们可以使用点(.)记法直接访问JSON对象的属性。另外，一旦得到数组，还可以使用关联数组的键来了解这个运动员是男还是女—这比每一次单独查找要高效得多。

Sharpen your pencil

更新`getDBRunners`函数，从`service.php`读取JSON对象。然后使用条件逻辑确定这个运动员放入哪一个列表。不过要当心！只有当JSON对象确实返回了运动员时才能这么做。

```
function getDBRacers() {
    $.getJSON('service.php', function(json) {
        if (json.runners.length > 0) {
            $('#finishers_m').empty();
            $('#finishers_f').empty();
            $('#finishers_all').empty();
            $.each(json.runners, function() {
                var info = '<li>Name: ' + this['fname'] + ' ' + this['lname'] + '. Time: ' +
this['time'] + '</li>';
                if(this['gender'] == 'm'){
                    $('#finishers_m').append( info );
                }else if(this['gender'] == 'f'){
                    $('#finishers_f').append( info );
                }else{
                    $('#finishers_all').append( info );
                }
            });
        }
        getTimeAjax();
    });
}
```



my_scripts.js



Sharpen your pencil Solution

使用条件逻辑以及JSON对象中返回的信息，可以确定这个运动员应当放到哪一个列表。与前面一样，所有运动员都还要增加到all_finishers列表。

```

function getDBRacers() {
    $.getJSON("service.php", function(json) {
        if (json.runners.length > 0) {
            if (!$('#finishers_m').empty()) {
                if (!$('#finishers_f').empty()) {
                    if (!$('#finishers_all').empty())
                        $().each(json.runners, function() {
                            var info = '<li>Name: ' + this['fname'] + ' ' + this['lname'] + '. Time: ' +
                            this['time'] + '</li>';
                            if (this['gender'] == 'm') {
                                $('#finishers_m').append(info);
                            } else if (this['gender'] == 'f') {
                                $('#finishers_f').append(info);
                            } else {
                                $('#finishers_all').append(info);
                            }
                        });
                    // 将这个运动员增加到all_runners列表。
                }
            }
        }
    });
    getTimeAjax();
}

```



my_scripts.js



Geek Bits

可以使用each方法循环处理JSON对象所返回数据中的所有元素。这个方法与（选择器）。each方法稍有不同，因为它可以迭代处理非jQuery数组，比如这里的runners数组。



试一试

更新 my_scripts.js文件中的getDBRacers函数。然后打开index.html，可以看到，通过使用Ajax、JSON和PHP，现在可以从MySQL数据库加载运动员了。

Male Finishers

- Name: John Smith, Time: 25.11
- Name: Jacob Walker, Time: 25.54
- Name: Fred Jones, Time: 26.08
- Name: Bob Hope, Time: 26.98
- Name: Ryan Rice, Time: 28.24
- Name: Austin Jones, Time: 29.14

从数据库获取的运动员信息。

Congratulations to all our finishers!

Page refreshes every 10 seconds.

Last Updated
July 31, 2011, 10:04:25 ms

看呐，能这样得到数据确实很不错。不过我们真正需要的是要把这些数据放在我们为市场部创建的表单中，对吧？

说得对。

一旦完成这一部分，啊哈，夏威夷，我们来了！下面首先确保市场部放入的数据不会给我们带来麻烦。



PHP的数据清理和验证

由于越来越多的垃圾邮件机器人和黑客试图非法得到你的数据集的所有权，来达到邪恶的目的，所以永远不要贸然相信输入到Web表单的数据。将发送到服务器的任何数据插入到数据库之前，一定要对这些数据进行验证和清理，这绝对没有坏处。这样可以确保得到特定输入域期望的数据类型（验证），另外还能确保接收到的数据不会包含任何可能对服务器或数据库造成危险的内容（清理）。这有助于保护你免于遇到类似SQL注入攻击、跨站点脚本问题以及网上常见的很多其他棘手问题。对于这个应用，我们将使用一些巧妙的PHP方法清理数据，确保使用正确的数据。

```
<?php
    // Encode the strings into safer web and database values
    htmlspecialchars($_POST["a"]); // The "empty" method checks if the value is empty
    empty($_POST["b"]); // This is a "Regular Expression". It checks $var against a set pattern
    preg_match('', $var); // This is a "Regular Expression". It checks $var against a set pattern
    ?> // 检查一个字符串是否为空。
        这是一个正则表达式匹配函数。使用正则表达式模式匹配可以指定很具体的条件，从而能控制输入数据的类型。

```

还有很多其他函数也可以用来完成数据清理：包括htmlentities、trim、stripslashes、mysql_real_escape_string等。可以在《Head First PHP & MySQL》的第6章中找到更多此类函数。

同一个PHP文件用于多种用途

我们已经了解了向服务器发送数据并由PHP处理的两种方法：POST和GET。通过结合条件逻辑，可以检测是利用POST请求还是GET请求来得到我们的PHP文件，并相应做出反应。还记得前面我们增加到表单的隐藏域吗？

```
<input type="hidden" name="action" value="addRunner" id="action">
    // 检查POST中的这个值，从而知道在通过POST提交表单。然后可以运行一些数据验证和清理函数，确保所有数据确实都是我们真正想要的。类似地，如果更新getJSON调用。利用一个URL参数（对应$_GET PHP对象）从数据库获取运动员，就能指定PHP文件中那一部分代码运行。这样一来，我们只需要维护一个PHP文件。

```

```
$.getJSON("service.php?action=getRunners", function(json) {
```

使用这个调用告诉PHP函数运行相应的代码从数据库获取运动员。

好了！所有人都在收拾东西，准备去机场了！现在已经知道自己完成名单了，是吗？

O_o



成品
PHP

用下面的代码更新你的service.php文件。它会处理信息GET和POST。另外还需要包含前面的db_connection、success和fail函数。

```

<?php
if ($_POST['action'] == 'addRunner') {
    $fname = htmlspecialchars($_POST['txtFirstName']);
    $lname = htmlspecialchars($_POST['txtLastName']);
    $gender = htmlspecialchars($_POST['ddlGender']);
    $minutes = htmlspecialchars($_POST['txtMinutes']);
    $seconds = htmlspecialchars($_POST['TxtSeconds']);
    if (preg_match('/[^\\w\\s]/i', $fname) || preg_match('/[^\\w\\s]/i', $lname)) {
        fail('Invalid name provided.');
    }
    if (empty($fname) || empty($lname)) {
        fail('Please enter a first and last name.');
    }
    if (empty($gender)) {
        fail('Please select a gender.');
    }
    $time = $minutes.":".$seconds;
    $query = "INSERT INTO runners SET first_name='".$fname."', last_name='".$lname."', gender='".$gender."', finish_time='".$time."'";
    $result = db_connection($query);
    if ($result) {
        $msg = "Runner: '".$fname."' ".$lname." added successfully";
        success($msg);
    } else {
        fail("Insert failed.");
    }
} elseif ($_GET['action'] == 'getRunners') {
    $query = "SELECT first_name, last_name, gender, finish_time FROM runners ORDER BY finish_time ASC";
    $result = db_connection($query);
    $runners = array();
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
        array_push($runners, array('fname' => $row['first_name'], 'lname' => $row['last_name'], 'gender' => $row['gender'], 'time' => $row['finish_time']));
    }
    echo json_encode(array("runners" => $runners));
}
exit;
}

```

检查是否有一个addRunner 通过 POST 提交到服务器。这是它将增加的数据。

对\$_POST数据信息完成数据清理。

数据验证确保确实输入了信息。

如果验证失败则调用fail函数。

检查数据库插入一个新记录……

……并检查是否成功。

检查URL串中是否存在getRunners值。

得到并返回结果。



```
function getDBRunners() {
    $.getJSON("service.php?action=getRunners", function(json) {
        if (json.runners.length > 0) {
            $('#finishers_m').empty();
            $.each(json.runners, function(i, runner) {
                var row = '|  |  |  |  |
| --- | --- | --- | --- |
| ' + runner.first_name + ' ' + runner.last_name + ' | ' + runner.gender + ' | ' + runner.rank + ' | ' + runner.time + ' |
';
                $('#finishers_m').append(row);
            });
        }
    });
    getTimeAjax();
}

```

动手做！



my_scripts.js



试一试

更新service.php和my_scripts.js文件后，在浏览器中打开index.html。应该能看到载入了运动员。还可以使用之前创建的新标签页上的表单，在列表中增加运动员。

The screenshot shows a web browser window with the title "2011 Race Finishers". The page content includes:

- A navigation bar with tabs: Male Finishers, Female Finishers, All Finishers, and Add New Finisher.
- A search form titled "All Finishers" with fields for First Name, Last Name, Gender (a dropdown menu), Rank Time (a dropdown menu), and a "Add Runner" button.
- A table with columns: First Name, Last Name, Gender, Rank, and Time.
- A message at the bottom: "Congratulations to all our finishers!"
- Buttons for "Start Page Updates" and "Stop Page Updates" with the note "Page refreshes every 10 seconds".
- Information about the last update: "Last Updated August 2, 2011, 10:57:59 pm".

A thought bubble from a character in the foreground contains the text: "太棒了！快点登机吧，别忘了在薄油上多喝几杯……"

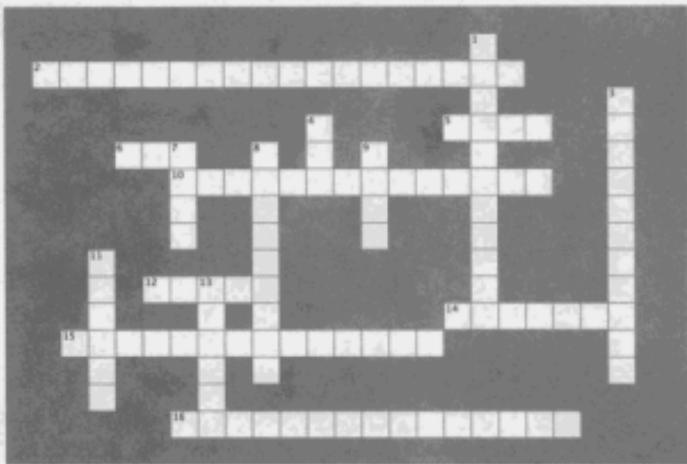


更多资源请访问稀酷客(www.ckook.com)



jQuery填字游戏

终于可以坐下来，让你的左脑动一动了。这是一个标准的填字游戏，所有答案都可以在本章中找到。



横向

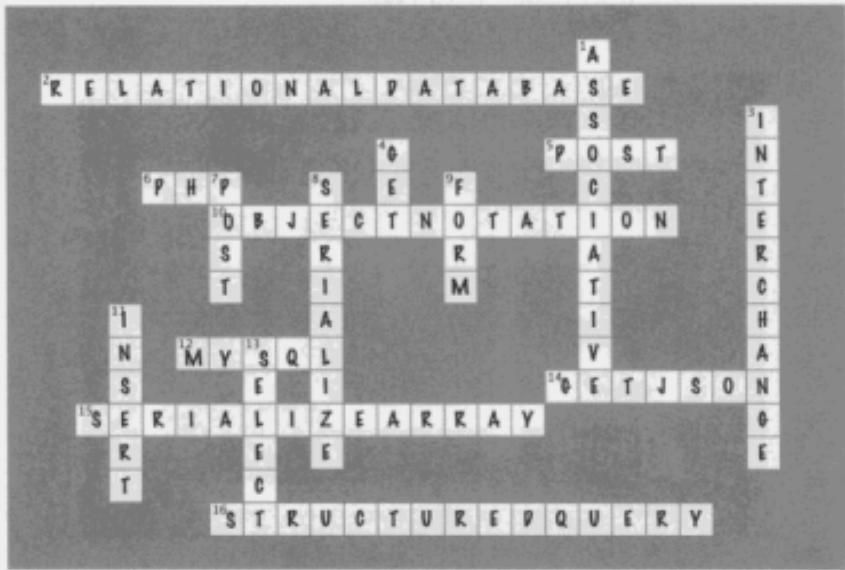
2. _____ 管理系统（RDBMS）是一些极有条理的应用，专门设计用来存储、组织和记住不同数据之间的关系。
- 5.这个jQuery快捷方法用于向服务器发送数据。这个方法有多个参数：URL，想要发送的信息以及一个处理函数，数据发送之后会运行这个函数。
- 6.这个服务器端脚本语言通常用来将表单中收集的数据发送到服务器。
- 10.JSON = Java脚本_____。
- 12.这是一个流行的开源数据库服务器，可以免费下载。
- 14.这个jQuery快捷方法用于从服务器获取JSON信息。
- 15.这是另外一个jQuery表单辅助方法，会创建键/值对的一个关联数组（非常适用于结构化存储）。
- 16.SQL _____。

纵向

- 1.一种特殊类型的数组，将信息存储在名/值对中。
- 3.JSON和XML是两种数据_____格式，jQuery和PHP可以相互来回传递。
- 4.用来从HTML表单向服务器发送数据的两个方法之一。这个方法将表单域名和值追加到URL的末尾。
- 7.用来从HTML表单向服务器发送数据的两个方法之一。这个方法也会发送数据，但是不会让数据在URL串中可见。
- 8.这是一个jQuery表单辅助方法，将表单中的所有输入连接为一个键/值对的串，各个键/值对之间用&字符分隔。
- 9.想要收集一个Web页面上的数据并发送到服务器进行处理时会使用这个HTML元素。
- 11.这个SQL命令用来将数据放入表中。



jQuery填字游戏答案





jQuery/Ajax/PHP/MySQL 工具箱

你已经掌握了第9章，现在工具箱里又增加了一些基本PHP、MySQL、JSON和更多Ajax。

MySQL

允许将数据存储在数据库和表中，可以使用SQL语言插入和获取信息。

SQL

这是一种查询语言，用来与数据库应用（如MySQL）交互。

JSON

使用getJSON函数可以从服务器得到JSON编码的数据。这会返回一个JSON对象。

数据可以使用POST方法从表单发送。发送数据之前，需要使用serializeArray格式化数据。

PHP

这是一种服务器端语言，允许将页面传递到浏览器之前在服务器上处理Web页面内容。
PHP脚本

这是一个包含PHP代码的文本文件，可以在Web服务器上完成任务。

<?php ?>

PHP脚本中的所有PHP代码都必须包围在这些标记之间。

这个PHP命令用来向浏览器窗口输出。语法如下：
echo 'Hello World';

\$_POST

这是一个包含表单数据的特殊变量。

json_encode

这个命令取一个数组，将它转换为jQuery请求的JSON编码数据。

更多资源请访问稀酷客(www.ckook.com)

表单改头换面



Web的生死全由用户和数据决定。从用户那里收集数据是很重要的事情，对Web开发人员来说可能是一个很耗费时间的难题。你已经看到jQuery可以帮助Ajax、PHP和MySQL。Web应用更有效地工作。现在来看jQuery对于建立用户界面来创建表单收集用户数据又有什么帮助。在这个过程中，你会得到一剂jQuery UI良药，这就是jQuery的官方用户界面库。

Cryptozoologists.org需要改头换面

Pattershy博士和Gimli博士专门致力于从世界各地的用户那里收集尽可能多的神秘生物观测数据。他们的网站cryptozoologists.org得到全世界专业和业余神秘生物学者的推崇。两位博士向你提出了一个要求：更新已经相当过时的Cryptid Sightings（神秘生物观测信息）表单。



Submit Your Cryptid Sighting

Please provide details of your cryptid sighting below along with your contact information.

Fill in the form below and then "Enter" to Submit

Describe Sighting: _____

When Did You See the Creature:

Creature Description:

Mammal Reptile Bird

这个表单看上去真不怎么样，就像20世纪30年代末的作品。

这里没有任何互动的内容来帮助用户输入高质量的数据，在这方面HTML也没有提供更多这样。所以博士们得到了很多批评质问。

美化HTML表单

下面是神秘生物学者希望看到的新表单的样板，这里还给出了一些附加说明。

Submit Your Cryptid Sighting

CRYPTID SIGHTING DATA

Date of Sighting:

Month	
1	2
3	4
5	6
7	8
9	10
11	12

在当前的表单上，这是一个文本域，可能带来很多乱七八糟的数据。我们希望用户从一个弹出的日历中选择日期，保证数据尽可能准确。

Creature Type:

当前表单上这里使用了单选框。
能不能让它们看起来更像亮堂？

Distance from Creature (estimate in feet):

Creature Weight:

Creature Height:

Color of Creature Hide/Fur/Covering:

A

B

C

这里放一个颜色
混合器怎么样？

SIGHTING LOCATION DATA

Latitude of Sighting Location:

Longitude of Sighting Location:

Enter your sighting data:

能不能让这个按钮更好看一些：目前表单上这种按钮式按钮显得我们很老土。



神秘生物学者的要求很苛刻。他们希望你建立一个漂亮的用户界面，就像桌面应用中常见的用户界面一样。你认为jQuery对完成这个任务有什么帮助？

更多资源请访问稀酷客(www.ckook.com)



Frank

Jim

Joe

你们见过“Submit Your Cryptid Sighting”
 (提交神秘生物观测数据) 表单的样子了吗?

Frank：我看到了。现在的表单是一个HTML表单，但是光靠HTML和CSS可没办法得到神秘生物学者们想要的新表单。

Jim：说说看。你还试图用CSS建立表单元素的样式？那太让人头疼了。

Frank：是啊，说到jQuery……嗯，我看不出jQuery中有什么可以帮助我们建立那样的界面组件。

Joe：伙计们，我们已经发现了，人们总喜欢这种花哨的玩艺，所以我们也要想办法构建这种组件。

Frank：可能需要结合JavaScript、jQuery和CSS来达到目的。

Jim：要写一大堆的逻辑。光是实现他们想要的弹出日历就需要编写一大堆代码和大量复杂的CSS。

Joe：嗯，实际上，可能有一个jQuery插件能完成这种工作。

Jim：插件，对呀！前几章我们就用过一个插件为Bit to Byte长跑大赛结果页面创建了标签页。所以肯定还有更多插件，是吗？

Joe：是啊，如果jQuery没有提供开发人员需要的某个功能，开发人员可能会构建一个插件。然后在jQuery社区发布这个插件，供大家使用。这样可以大大节省其他开发人员的时间。

Jim：这么说，可能有开发人员或开发团队已经解决了这个问题，是吗？

Frank：这样我们就省事了。

Joe：我们到jQuery.com看看，看能找到些什么。



如果能有一个jQuery界面插件库
该多好啊，这只是一个美梦吧？
我知道这只是异想天开……

用jQuery UI减轻编码负担（还能节约时间）

对所有开发人员来说，非常幸运，jQuery为此提供了一个官方的用户界面插件库。这个库名为jQuery UI，它为jQuery内核提供了3种主要插件：效果插件、交互插件和部件插件。

效果插件

jQuery UI扩展了jQuery，它增加了更多效果。可以让你的元素弹跳、爆炸、有规律地跳动，还可以摇晃。jQuery UI还包括一些松弛函数和复杂的数学运算，使动画效果更真实。

交互插件

交互插件可以为Web应用增加更复杂的行为。通过实现可拖放或可排序，或者支持其他一些选项，允许用户与元素自然地交互。

部件插件

Web部件是一种自包含的组件，可以为Web应用增加功能。利用部件，可以大大节省你的编码时间，降低复杂性，创建可用的响应式用户界面元素。

这一章将主要讲解部件来完成我们的用户界面。

jQuery提供了一个插件体系结构，允许Web开发人员扩展核心jQuery库（或者增加核心jQuery库的功能）。



试一试

可以尝试一些jQuery UI效果、交互和部件插件，访问以下URL，并按使用说明来操作。

URL	使用说明
http://jqueryui.com/demos/animate/#default	点击Toggle Effect（切换效果）按钮。
http://jqueryui.com/demos/effect/default.html	从下拉列表选择一个效果，然后点击Run Effect（运行效果）。
http://jqueryui.com/demos/draggable/#default	点击并保持在“Drag me around”方框上，然后拖动鼠标，将这个方框拖动到指定的屏幕区域内。
http://jqueryui.com/demos/accordion/#default	点击不同的区域，查看折叠区的扩展和收缩。
http://jqueryui.com/demos/dialog/#animated	点击Open Dialog（打开对话框）按钮，查看一个定制jQuery UI对话框。这个对话框比沉闷的普通JavaScript提醒框好多了，不是吗？



WHO DOES WHAT?

将各个jQuery UI插件与插件类型和插件作用连线。提示：如果你不太确定，可以在上一页“试一试”练习中给出的演示网站上多花点时间来了解。

Puff

交互插件：使一个DOM元素作为可拖放元素的目标。

Autocomplete

部件插件：显示某个事件当前完成的百分比。

Droppable

效果插件：让一个元素看起来像烟雾一样扩散并消散至透明。

Explode

部件插件：用户在一个输入域中键入时，提供一组可取值的列表。

Sortable

效果插件：使元素看起来像窗户一样上滑或下滑。

Progressbar

部件插件：创建折叠区来组织Web内容。

Resizable

效果插件：使元素看起来像破碎成片，四处分散。

Blind

交互插件：使元素可以通过拖动进行排序。

Accordion

交互插件：为元素提供一个可拖动的手柄，从而允许用户扩展这个元素。



答案

将各个jQuery UI插件与插件类型和插件作用连线。



动手做! → 使用jQuery UI之前，首先需要配置我们想要的组件。选择一个主题，并下载一个jQuery UI副本。步骤如下：

- ① 将浏览器指向jQuery UI下载页面：

<http://jqueryui.com/download>

- ② 选择想要下载的组件。

UI Core
A required dependency, contains basic functions and initializers.

Core
 Widget
 Mouse
 Position

Widgets
Full-featured UI Controls - each has a range of options and is fully themable.
这个项目中，我们只需要UI核心组件（UI Core）和部件（Widgets），所以只选择这两个所示的组件。

Accordion
 Autocomplete
 Button
 Dialog
 Slider
 Tab
 Tree
 Uploader
 Progressbar

- ③ 为下载选择一个主题。

Theme
Select the theme you want to include or design a custom theme

为下载选择Sunny主题。

jQuery UI有很多优点，其中之一就是它的主题。jQuery UI的开发团队提供了用来建立漂亮界面的所有CSS，你甚至可以用jQuery UI的“theme roller”创建自己的主题。要查看全部jQuery UI主题库，请访问以下URL：

<http://jqueryui.com/themeroller/#themeGallery>

- ④ 按下Download（下载）按钮。

Download

现在我已经准备好jQuery UI了！怎么开始使用呢？

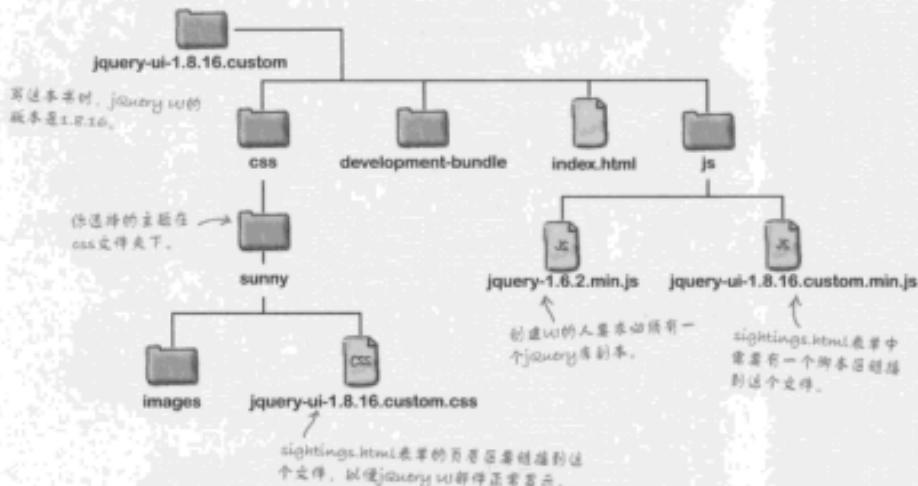
只需要解压缩文件夹，把这个库包含在一个项目文件夹中。

翻开下一页，我们来看看jQuery UI的内部。

更多资源请访问稀酷客(www.ckook.com)

JQuery UI包的内部

下载并解压缩jQuery UI后，你会发现它的结构如下：



在本书最前面下载的代码文件夹中，我们已经包含了jQuery UI文件夹。可以在ch10下的end文件夹中找到这个文件夹。

项目任务列表

jQuery UI可以为你做很多事情，不过建立这个新表单还需要解决一些问题。下面是
我们需要完成的任务列表：

- 1. 为用户构建一个日期选择工具来输入观测日期。
- 2. 为用户建立更有吸引力的按钮来选择生物类型。
- 3. 为用户构建一些录入数字的滑动条，用来输入与生物的距离、生物重量、生物高度、纬度和经度。
- 4. 为用户构建一个颜色混合器界面组件来输入生物颜色。
- 5. 为观测表单建立一个更漂亮的提交按钮。

更多资源请访问稀酷客(www.ckook.com)

观测表单中建立一个日期选择工具

在HTML表单中加入jQuery UI部件极其容易。下面先来看这个日历选择工具：

- ① 创建jQuery UI CSS文件的一个链接：

```
<link type="text/css" href="jquery-ui-1.8.16.custom/css/sunny/
jquery-ui-1.8.16.custom.css" rel="stylesheet" />
```

- ② 创建一个<script>标记，指向jQuery UI。

```
<script src="jquery-ui-1.8.16.custom/js/jquery-ui-1.8.16.
custom.min.js"></script>
```

- ③ 建立一个普通的HTML输入域。

```
<input type="text" name="sighting_date">
```



- ④ 指定这个<input>标记的ID为datepicker。

```
<input type="text" name="sighting_date" id="datepicker">
```

- ⑤ 创建一个JavaScript文件，将以下代码放在\$(document).ready(function(){})的大括号之间。

```
$('#datepicker').datepicker();
```

- ⑥ 在你喜欢的浏览器中打开这个文件，点击输入域。



幕后的jQuery UI

看起来简直有魔法一样，不过jQuery UI实际上只是一些合理设计、合理编写的jQuery代码，有了jQuery UI，你就不必再自己编写这些代码。下面来进一步讨论它是如何工作的。

- ① 类似于之前编写的所有其他jQuery代码，这个日期选择工具要使用一个选择器和一个方法。



- ② 使用jQuery和jQuery UI库，JavaScript解释器向DOM请求ID为datepicker的元素。

- ③ DOM获取选择的元素，运行这个元素的datepicker方法，再返回给解释器。



只需要写一点点HTML和jQuery就能完成所有这一切，真是太棒了。不过这是不是意味着我只能接受目前现成的外观和功能而没有其他选择？如果我想有所谓的选择怎么办？



不用担心，你还是有选择的。下面就来看看这些选择。

部件有可定制的选项

如果深入研究datepicker部件，会发现它提供了丰富的特性和可配置的选项。



用选项定制datepicker

由于jQuery UI基于jQuery构建，所以不用编写太多代码，就能定制这个datepicker部件来满足你的需要。写这本书时，datepicker有46个不同选项可以设置。

```
$("#datepicker").datepicker({
    stepMonths: 3
});
```

datepicker部件提供了大量可配置的选项。stepMonths选项允许改变步长，也就是每次跨过多少月数。

```
$("#datepicker").datepicker({
    changeMonth: true
});
```

如果设置changeMonth选项为true，用户可以从一个下拉列表选择月份。



编写代码，允许用户使用一个下拉列表改变datepicker部件的月份和年份。提示：设置多个选项项时，要在选项之间加一个逗号。



Exercise Solution

编写代码，允许用户使用一个下拉列表改变datepicker部件的月份和年份。提示：设置多个选项时，要在选项之间加一个逗号。

```
$( '#datepicker' ).datepicker({
    changeMonth: true, changeYear: true
});
```

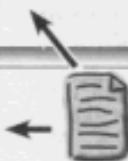


成品代码

在ch10下的begin文件夹中找到文件sightings_begin.html。把它保存为第10章end文件夹中的sightings_end.html。把以下加粗的代码增加到sightings_end.html和my_scripts.js文件中。

```
<head>
    <title>Submit Your Cryptid Sighting</title>
    <link rel="stylesheet" type="text/css" href="style/form.css" />
    <link type="text/css" href="jquery-ui-1.8.16.custom/css/sunny/jquery-ui-1.8.16.
    custom.css" rel="stylesheet" />
</head>
```

靠近sightings_end.html文件最上面的位置。
将链接放到jquery ui CSS文件。
使部件能正常显示。



```
<h3>Date of Sighting:</h3>
    <input type="text" name="sighting_date" id="datepicker" />
```

sightings_end.html

```
<script src="scripts/jquery-1.6.2.min.js"></script>
<script src="scripts/my_scripts.js"></script>
<script src="jquery-ui-1.8.16.custom/js/jquery-ui-1.8.16.custom.min.js"></
script>
</body>    需要包含jquery ui库，从而可以利用所有属性。
</html>    的ui特性。
```

靠近sightings_end.html
文件最下面的位置。



```
$ (document) .ready(function () {
    $( '#datepicker' ).datepicker({ changeMonth: true, changeYear: true});
}); //end doc ready
```

↑ 日期选择工具代码。



my_scripts.js

更多资源请访问稀酷客(www.ckook.com)



试一试

输入上一页的代码后，在你喜欢的浏览器中打开sightings_end.html文件。测试datepicker部件。点击“下一个”和“上一个”按钮，还可以点击月份和年份下拉列表，确保一切正常。

日期选择工具成功了！

datepicker部件能够按我们预期的方式工作。

datepicker部件能够按我们预期的方式工作。

datepicker部件能够按我们预期的方式工作。

勾掉第一项

任务列表中第一项已经完成。下面继续完成第二项。

- 1. 为用户构建一个日期选择工具来输入观测日期。
- 2. 为用户建立更有吸引力的按钮来选择生物类型。
- 3. 为用户构建一些录入数字的滑块条，用来输入与生物的距离、生物重量、生物高度、纬度和经度。
- 4. 为用户构建一个颜色混合器界面组件，来输入生物颜色。
- 5. 为观测表单建立一个更漂亮的提交按钮。

更多资源请访问稀酷客(www.ckook.com)

建立按钮样式

“更有吸引力”到底是什么意思？这主要是一个关于样式的问题：就是要建立一个更好看的按钮，让人们更愿意点击它。jQuery UI库中有一个超级有用的部件：按钮部件。按钮部件提供了一个button方法，来帮助你创建更漂亮的表单元素，如提交按钮、单选钮和复选框。

这是一个按钮的HTML：

```
<input type="radio" id="radiol" name="radio" />
<label for="radiol">Choice 1</label>
```

这个输入会在用户点击时更新。

按钮部件将标签的样式设置到表单元素上，从而形成一个按钮。

相应的jQuery语句：

```
$("#radiol").button();
```

button方法将普通的HTML单选钮转换为一个漂亮的、更具交互性的按钮。

不要忘记，输入按钮的标记代码要放在一个HTML表单标记中。

按钮部件成组

为了建立一个成组的按钮集合，jQuery UI提供了buttonset方法，这个方法通过引用按钮组的容器元素，将单个按钮元素归为一组。

将单选钮组放在一个容器元素中。

```
<div id="radio">
  <input type="radio" id="radiol" name="radio" />
  <label for="radiol">Choice 1</label>
  <input type="radio" id="radio2" name="radio" />
  <label for="radio2">Choice 2</label>
  <input type="radio" id="radio3" name="radio" />
  <label for="radio3">Choice 3</label>
</div>
```

在jQuery代码中：

选择容器元素。

```
$("#radio").buttonset();
```

buttonset方法将按钮成组，并对各个元素分别运行button方法。



jQuery UI磁贴

按正确的顺序摆放代码磁贴，为用户创建一组按钮，用来选择他们看到的生物的类型。我们已经为你放了几个磁贴。



sightings_end.html

```
<div id="type_select">
  <input type="radio" id="radio1" name="creature_type" />
  <label for="radio1">Chupacabras</label>
```



my_scripts.js

```
<input type="radio" id="radio2" name="creature_type" />
</div>
$( "#type_select" ).buttonset();
<label for="radio3">Loch Ness Monster</label>
<label for="radio4">Sasquatch</label>
<label for="radio2">Jersey Devil</label>
<input type="radio" id="radio4" name="creature_type" />
<input type="radio" id="radio3" name="creature_type" />
```

更多资源请访问稀酷客(www.ckook.com)



jQuery UI磁贴答案

现在你已经有了一组漂亮的按钮，能适应表单的整体主题。

```
<div id="type_select">
```

```
    <input type="radio" id="radio1" name="creature_type" />
```

```
        <label for="radio1">Chupacabras</label>
```

```
    <input type="radio" id="radio2" name="creature_type" />
```

```
        <label for="radio2">Jersey Devil</label>
```

```
    <input type="radio" id="radio3" name="creature_type" />
```

```
        <label for="radio3">Loch Ness Monster</label>
```

```
    <input type="radio" id="radio4" name="creature_type" />
```

```
        <label for="radio4">Sasquatch</label>
```

```
</div>
```



sightings_end.html

在code文件夹中，可以看到我们还增加了Yeti和Other按钮。这里没有给出主要是因为这个练习的篇幅不够。

```
$( "#type_select" ).buttonset();
```



my_scripts.js



试一试

将上面的代码行增加到sightings_end.html和my_scripts.js文件中。然后，在你喜欢的浏览器中打开这个页面，确保一切正常。

Creature Type:

Chupacabras Jersey Devil Loch Ness Monster Sasquatch Yeti Other

你把这些普通的
HTML单选框.....

.....变成了一组
漂亮的按钮。

Submit Your Cryptid Sighting
Fill out the form below and click "Save" to submit.

CRYPTID SIGHTING DATA

Date of sighting: _____

Creature Type: Chupacabra Jersey Devil Loch Ness Monster Sasquatch Yeti Other

Distance from Creature (in ft.): _____

Creature Height (in feet): _____

Creature Height (in ft.): _____

Color of Creature: _____

CRYPTID LOCATION DATA

Latitude of sighting: _____

Longitude of sighting: _____

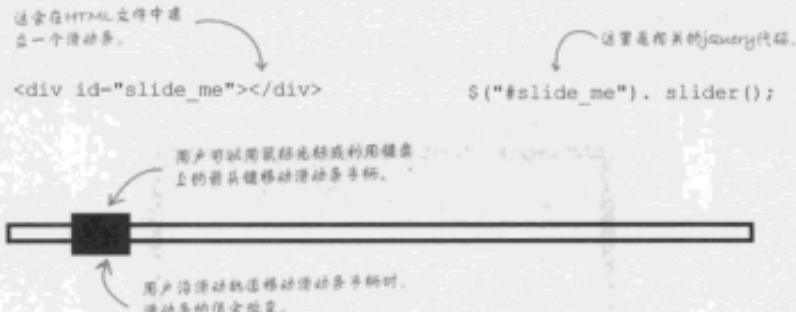
哇，真是很容易。表单任务列表的下一项是什么呢？

- 1. 为用户构建一个日期选择工具来输入观测日期。
- 2. 为用户建立更有吸引力的按钮来选择生物类型。
- 3. 为用户构建一些录入数字的滑动条，用来输入与生物的距离、生物重量、生物高度、纬度和经度。
- 4. 为用户构建一个颜色混合器界面组件，来输入生物颜色。
- 5. 为观测表单建立一个更漂亮的提交按钮。

更多资源请访问稀酷客(www.ckook.com)
你现在的位置：389

用滑动条控制数值输入

利用jQuery UI滑动条插件，你可以构建一个滑动条界面，允许用户用他们的鼠标或键盘控制。滑动条还可以帮助你控制用户输入的数值。前面已经看到，一旦有了jQuery UI库，建立部件就是小菜一碟。建立滑动条部件也同样很简单。



滑动条也提供了很多定制选项。假设需要用户输入一组数。我们希望用户输入的最小值是0，最大值是100。另外我们希望用户以5为增量输入数值。就可以如下设置滑动条部件的选项：

```
$("#slide_me").slider({
    value: 0,           // value选项告诉滑动条当前的值。
    min: 0,             // min选项向滑动条指出用户可以输入的最小值。
    max: 100,            // max选项向滑动条指出用户可以输入的最大值。
    step: 5,             // step选项告诉滑动条我们希望采用多大的增量。
    orientation: 'vertical' // orientation选项可以设置为horizontal(水平)或vertical(垂直)。
});
```

orientation选项可以设置为horizontal(水平)或vertical(垂直)。



更多资源请访问稀酷客(www.ckook.com)



滑动条部件也有一些很棒的选项，
不过我们怎么从滑动条得到值并放
在表单输入域中呢？

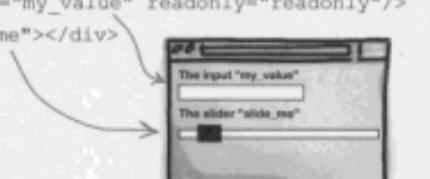
必须用滑动条部件的某个事件处理程序来连接滑动条。

我们已经见过一些部件选项，不过还没有研究过jQuery UI的另一个强大特性。很多jQuery组件都提供了事件处理程序，滑动条也不例外。写这本书时，jQuery UI的滑动条部件提供了5个事件处理程序：create、start、slide、change和stop。要把滑动条连接到一个表单输入域，下面来尝试slide事件处理程序。

如果不想用户键入或输入
一个数，可以把这个属性设
置为“readonly”（只读）。

滑动条部件的HTML

```
<input type="text" id="my_value" readonly="readonly"/>
    <div id="slide_me"></div>
```



滑动条部件的jQuery脚本

```
$("#slide_me").slider({
    slide: function( event, ui ) {
        $("#my_value").val( ui.value );
    }
});
$("my_value").val( $("#slide_me").slider( "value" ));
```

这是slide事件处理程
序。用户移动滑动条手
柄时会触发slide事件。

slide事件关联到一个函
数回调。这个函数运行
时，会用jQuery val方
法设置输入。

用户移动滑动条手柄时，会调用这个函
数，输入将更新为滑动条的值。





Long Exercise

下面是神秘生物学者们需要的各个输入域的代码。请在这些代码中填空。他们对需要设置的滑动条选项给出了一些提示。

与生物的距离 (单位为英尺) :

起始值应当为0。

最小距离应当为0。

最大距离应当是500。

增量为10英尺。

```
<h3>Distance from Creature (in ft.):</h3>
<input type="text" id="....." class="just_display" name="creature_distance"
       readonly="readonly"/>
    <div id="....."></div>
</div>
```



sightings_end.html

```
$( "#slide_dist" ).slider({
    .....
    .....
    .....
    slide: function( event, ui ) {
        $( "#distance" ..... )
    }
});
```



my_scripts.js

生物重量（单位为磅）：

初始值应为0。

最小重量应为0。

最大重量应为5000。

增量为5磅。

生物高度（单位为英尺）：

初始值应为0。

最小高度应为0。

最大高度应为20。

增量为1英尺。

```
<h3>Creature Weight (in lbs.):</h3>
<input type="text" id="weight" class="just_display" name="creature_weight"
       readonly="readonly"/>
<div id="slide_weight"></div>

<h3>Creature Height (in ft.):</h3>
<input type="text" id="height" class="just_display" name="creature_height"
       readonly="readonly"/>
<div id="slide_height"></div>
```



sightings_end.html

```
$(".....").slider({
    .....
    .....
    .....
    slide: function( event, ui ) {
        $(".....").val( ui.value);
    }
});
```

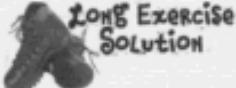


my_scripts.js

```
$("#slide_weight").slider({
    .....
    .....
    .....
    .....
    $("#weight").val( ui.value);
});
```



my_scripts.js



下面是神秘生物学者们需要的各个输入域的代码，请在这些代码中填空。他们对需要设置的滑动条选项给出了一些提示。

与生物的距离（单位为英尺）：

初始值应当为0。

最小距离应当为0。

最大距离应当是500。

增量为10英尺。

```
<h3>Distance from Creature (in ft.):</h3>
<input type="text" id="distance" class="just_display" name="creature_distance"
       readonly="readonly"/>
    <div id="#slide_dist"></div>
</div>
</div>
```



sightings_end.html

```
$( "#slide_dist" ).slider({
    value: 0,
    min: 0,
    max: 500,
    step: 10,
    slide: function( event, ui ) {
        $( "#distance" ).val(ui.value);
    }
});
```



my_scripts.js

生物重量（单位为磅）：

起始值应为0。

最小重量应为0。

最大重量应为5000。

增量为5磅。

生物高度（单位为英尺）：

起始值应为0。

最小高度应为0。

最大高度应为20。

增量为1英尺。

```
<h3>Creature Weight (in lbs.):</h3>
<input type="text" id="weight" class="just_display" name="creature_weight"
       readonly="readonly"/>
<div id="slide_weight"></div>

<h3>Creature Height (in ft.):</h3>
<input type="text" id="height" class="just_display" name="creature_height"
       readonly="readonly"/>
<div id="slide_height"></div>
```



sightings_end.html

```
$( "#slide_height" ).slider({
    value:0,
    min:0,
    max:20,
    step: 1,
    slide: function( event, ui ) {
        $( "#height" ).val( ui.value );
    }
});
```



my_scripts.js

```
$( "#slide_weight" ).slider({
    value:0,
    min:0,
    max:5000,
    step: 10,
    slide: function( event, ui ) {
        $( "#weight" ).val( ui.value );
    }
});
```



my_scripts.js



试一试

把前几页大练习的代码增加到你的文件中，然后在你喜欢的浏览器中打开sightings_end.html。要用鼠标和键盘测试滑动条（左右箭头键会让滑动条按其增量滑动）。

Submit Your Cryptid Sighting
Fill out the form below and click "Enter" to Submit

CRYPTID SIGHTING DATA

Date of Sighting:

Creature Type:

- Chupacabra
- Jersey Devil
- Loch Ness Monster
- Sasquatch
- Yeti
- Other

Distance from Creature (in ft.):

120

Creature Weight (in lbs.):

3400

Creature Height (in ft.):

8

Color of Creature:

CRYPTID LOCATION DATA

Latitude of Sighting:

用户现在可以通过滑动条手动输入数据，神秘生物学家们得到更好的数据质量控制。

还有两个数值滑动条需要构建：纬度和经度。然后就可以从项目的任务列表中再去掉一项了！

更多资源请访问稀酷客(www.ckook.com)



不过纬度和经度有负值，而且间隔是小数，
比如0.00001。滑动部件能处理这些数据吗？

这种情况jQuery UI也能处理。

滑动部件可以处理负数和小数。你可以输入负数作为值，作为最小值，也可以作为最大值。可以试试看负数的实际使用。

Sharpen your pencil

填入jQuery代码，让纬度和经度滑动条正常工作。纬度滑动条范围为-90 - 90，增量为0.00001。经度滑动条范围为-180 - 180，增量为0.00001。

```
value:0,
.....
```

```
slide: function( event, ui ) {
    $( "....." ).val( ui.value );
}
});
```



```
value:0,
.....
```

```
slide: function( event, ui ) {
    $( "....." ).val( ui.value );
}
});
```



my_scripts.js

Sharpen your pencil

Solution

填入jQuery代码，让纬度和经度滑动条正常工作。纬度滑动条范围为-90~90，增量为0.00001。经度滑动条范围为-180~180，增量为0.00001。

```

$( "#slide_lat" ).slider({
    value:0,
    min:-90,
    max:90,
    step: 0.00001,
    slide: function( event, ui ) {
        $( "#latitude" ).val( ui.value);
    }
});

$( "#slide_long" ).slider({
    value:0,
    min:-180,
    max:180,
    step: 0.00001,
    slide: function( event, ui ) {
        $( "#longitude" ).val( ui.value);
    }
});

```



my_scripts.js

到目前为止，任务列表中的很多任务都已经完成了。

- 1. 为用户构建一个日期选择工具来输入观测日期。
- 2. 为用户建立更有吸引力的按钮来选择生物类型。
- 3. 为用户构建一些录入数字的滑动条，用来输入鸟生物的距离、生物重量、生物高度、纬度和经度。

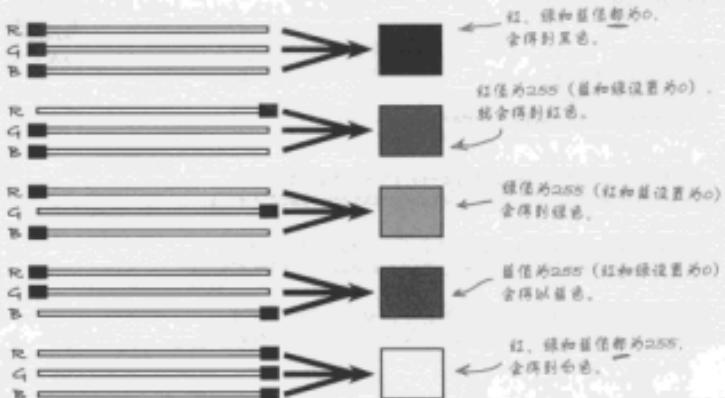
接下来是什么？现在需要为神秘生物学者们构建一个界面，允许用户使用表示红、绿和蓝的滑动条，来输入他们看到的生物的颜色。

- 4. 为用户构建一个颜色混合器界面组件，来输入生物颜色。

更多资源请访问稀酷客(www.ckook.com)

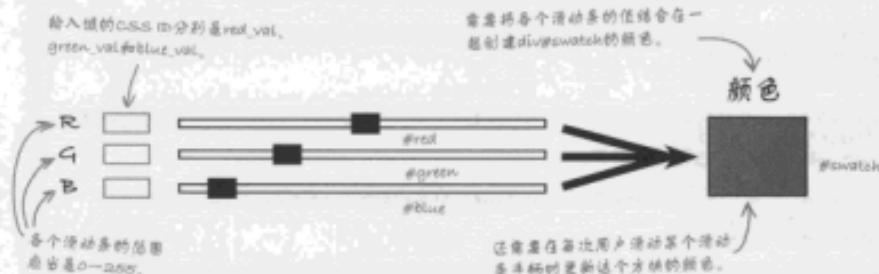
计算机使用红、绿和蓝混合颜色

红、绿和蓝值分别有最小值0和最大值255。各个颜色都取最小值时，也就是说，红值为0，绿值为0，蓝值也为0，你就会得到黑色。如果各个值都取其最大值，红值为255，绿值为255，蓝值也为255，你就会得到白色。



滑动条要做的工作是一样的

所以需要建立3个不同的滑动条：一个对应红色，一个对应绿色，另一个对应蓝色。然后要把各个滑动条的值组合为一个颜色。下面来看需要各个滑动条部件做些什么。





成品代码

将粗体的代码增加到sightings_end.html和form.css文件。这会完善文件，建立选色工具。

```
*color slider styles*
  #slide_dist, #slide_weight, #slide_height {
    margin-bottom:14px;
  }
  #swatch {
    width: 75px;
    height: 75px;
    background-image: none;
  }
```

} 这个CSS ID样式定义了用户设置滑动条时显示颜色的样貌。

```
#red .ui-slider-range { background: #ef2929; }
#red .ui-slider-handle { border-color: #ef2929; }
```

← 这些样式分别将在滑动条轨道中嵌入滑动条颜色。

```
#green .ui-slider-range { background: #8ae234; }
#green .ui-slider-handle { border-color: #8ae234; }
```

```
#blue .ui-slider-range { background: #729fcf; }
#blue .ui-slider-handle { border-color: #729fcf; }
```



form.css

<h3>Color of Creature (use the color sliders to enter):</h3> 这个输入域包含颜色的十六进制值。

Color (in hexadecimal):<input type="text" class="just_display" name="creature_color" id="color_val" readonly="readonly"/>

<div id="swatch" class="ui-widget-content ui-corner-all"></div>

Red:<input type="text" class="just_display" name="creature_color" id="red_val" readonly="readonly"/>
<div id="red"></div> ← 红色滑动条的div。

Green:<input type="text" class="just_display" name="creature_color" id="green_val" readonly="readonly"/>
<div id="green"></div> ← 绿色滑动条的div。

Blue:<input type="text" class="just_display" name="creature_color" id="blue_val" readonly="readonly"/>
<div id="blue"></div> ← 蓝色滑动条的div。



sightings_end.html

Sharpen your pencil

下面的脚本建立了红色、绿色和蓝色滑动条。阅读各行代码，根据你对jQuery和jQuery UI的了解，考虑这些代码分别用来做什么。然后把你考虑的结果写下来。如果你不确定这些代码做什么，可以大胆去猜。我们已经为你完成了一个。

```
$("#red, #green, #blue").slider({
    orientation: "horizontal",
    range: "min",
    max: 255,
    value: 127,
    slide: refreshSwatch,
    change: refreshSwatch
});
```

设置滑动条范围，使用户只能选择一个最大值。

there are no Dumb Questions

问：jQuery UI提供的那些CSS都是什么？

答：jQuery UI的一大好处，开发人员已经考虑了很多复杂的CSS。所以无需你再考虑。可以查看jQuery UI色中的文件（jquery-ui-1.8.16.custom.css/sunny/jquery-ui-1.8.16.custom.css）来了解这些CSS。要想对这些CSS有更多的了解，请访问以下URL的jQuery UI文档：

<http://jqueryui.com/docs/Theming/API>

问：你说过可以为jQuery UI建立我们自己的主题。怎么做到呢？

答：通过使用jQuery UI的theme roller，很容易创建你自己的主题。首先，将浏览器指向theme roller应用：

通过使用Draggable部件，可以把元素变成可拖动的组件。

Droppable部件可以作为一个可拖动部件的目标。

Resizable部件将元素变成一个可扩展的对象，可以拖动它的四角或边框改变它的大小。

Selectable部件可以将元素变为可选择的组件。访问网站的人可以在这些元素上拖动鼠标来选择这些元素，就像选择桌面上的文件一样。

Sortable部件允许用户交叉式地重新排序。

问：我还不太了解交互部件。这些部件用来做什么？

答：利用交互部件可以创建桌面应用中常见的那种交互功能。

更多资源请访问稀酷客(www.ckook.com)



Sharpen your pencil Solution

下面的脚本建立了红色、绿色和蓝色滑动条。阅读各行代码，根据你对jQuery和jQuery UI的了解，考虑这些代码分别用来做什么。然后把你考虑的结果写下来。如果你不确定这些代码做什么，可以大胆去猜。以下是我们答案。

```
$("#red, #green, #blue").slider({
    orientation: "horizontal",
    range: "min",
    max: 255,
    value: 127,
    slide: refreshSwatch,
    change: refreshSwatch
});
```

将对应#red,#green,#blue的

转换为滑动条部件。
使它们成为水平滑动条而不是垂直滑动条。
设置滑动条范围，使用户只能选择一个最大值。
设置最大值为255，保证在颜色限制范围内。
设置值，使滑动条手柄大概在中间位置。
用户滑动时调用一个名为refreshSwatch的函数。
任何值改变时同样调用这个函数。

建立refreshSwatch函数

要完成我们的颜色混合器，需要一个JavaScript函数设置和刷新样块。下面是这个函数的骨架，另外给出了一些需要考虑的关键问题，在具体完成这个函数实现你想要的功能之前，需要先仔细考虑这些问题。

```
function refreshSwatch() {
    var red = ??? } 如何把各个滑动条的值分
    var green = ??? 别放在这些变量中？
    var blue = ??? } 常常在这个变量中设置rgb值。
    var my_rgb = ??? ← 以便设置颜色样块。
    $("#swatch").???; ← 哪个jQuery方法允许设置
    $("#red_val").val(red); 颜色样块的颜色？
    $("#blue_val").val(blue);
    $("#green_val").val(green);
    $("#color_val").val(my_rgb); } 这里没有太大问题。只要使用
}                                     jQuery的val方法设置输入值。滑
                                         动条改变时显示它们的值。这样
                                         用户就能知道值是什么。
```

更多资源请访问稀酷客(www.ckook.com)



我们已经了解如何从滑动条得到值，
不过怎么创建颜色呢？难道不需要一堆代
码来转换十六进制的Web颜色吗？

问得好！我们可以编写一个十进制到十六进制的转换函数，或者也可以直接使用滑动条提供的十进制值。

应该记得，CSS `background-color`属性允许我们采用下面的方式指定颜色：

R G B
 ↓ ↓ ↓
`background-color:rgb(255,0,255)`

不过这个提示只回答了其中一个问题。要编写整个函数，还要再动动脑筋。



填入代码行，完成这个`refreshSwatch`函数。

```
function refreshSwatch() {
    var red =
    var green =
    var blue =
    var my_rgb =
    $( "#swatch" ).css("background-color", "#");
    $( "#red_val" ).val(red);
    $( "#blue_val" ).val(blue);
    $( "#green_val" ).val(green);
    $( "#color_val" ).val(my_rgb);
}
```



填写refreshSwatch函数。

```
function refreshSwatch() {
    var red = $("#red").slider("value");
    var green = $("#green").slider("value");
    var blue = $("#blue").slider("value");
    var my_rgb = "rgb(" + red + "," + green + "," + blue + ")";
    $("#swatch").$( "#swatch" ).css("background-color", my_rgb);
    $("#red_val").val(red);
    $("#blue_val").val(blue);
    $("#green_val").val(green);
    $("#color_val").val(my_rgb);
}
```

把RGB值放入这个变量……

……可以将样块的CSS值设
置为三个颜色的组合值。



成品代码

更新my_scripts.js文件，包含前几页颜色滑动条和refreshSwatch函数的代码。另外还要增加下面的代码行。这些代码会触发slider值，在Web页面加载时运行refreshSwatch函数，这使得Web页面最开始有一个有颜色的样块，而不是一个空页面。

```
$( "#red" ).slider("value", 127);
$( "#green" ).slider("value", 127);
$( "#blue" ).slider("value", 127);
```



试一试

在你喜欢的浏览器中打开sightings_end.html。要用鼠标和键盘测试滑动条（左右箭头键会让滑动条按其增量滑动）。

Submit Your Cryptid Sighting
Fill out the form below and click "Enter" to submit.

CRYPTID SIGHTING DATA

Date of Sighting:

Creature Type: Chupacabra Jersey Devil Loch Ness Monster Sasquatch Yeti Other

Distance from Creature (in ft.)

Creature Weight (in lbs.)

Color of Creature (use the color sliders to enter):
Color is hexadecimal: #FF0000

Red: 127

Green: 127

Blue: 127

这个颜色滑动条简直
炸裂了！现在我们的研究渠
容易多了。



用户现在可以通过滑动这个滑动条的手柄来输入值，神秘生物学者们也能得到更好的数据质量控制。

更多资源请访问稀酷客(www.ckook.com)

还有最后一个小问题……

现在任务列表只剩下一项了。

- 1. 为用户构建一个日期选择工具来输入观测日期。
- 2. 为用户建立更有吸引力的按钮来选择生物类型。
- 3. 为用户构建一些录入数字的滑动条，用来输入与生物的距离、生物重量、生物高度、纬度和经度。
- 4. 为用户构建一个颜色选择器界面组件，来输入生物颜色。
- 5. 为观测表单建立一个更漂亮的提交按钮。

这一章已经使用了按钮，而且从第1章开始你一直都在使用按钮，所以这个任务对你来说应该相当容易！

there are no Dumb Questions

问：我们建立的颜色滑动条中background-color使用了rgb设置。如果我想建立另外一个颜色滑动条，颜色使用标准十六进制记法。该怎么做呢？

答：我们希望为颜色滑动条创建最精巧、最简单的代码。如果需要一个采用十六进制颜色格式收集数据的颜色滑动条，那么你很幸运。

可以使用jQuery UI网站提供的一个简单的颜色选择工具的示例代码。将浏览器指向以下URL：

<http://jqueryui.com/demos/slider/#colorpicker>

然后找到这个示例颜色选择工具下的View Source（查看源代码）链接。选择并复制文本域中的HTML、CSS和

jQuery代码，保存到一个新文档中。将这个文档链接到jQuery包中适当的CSS和jQuery文件。搞定了，现在你已经有了一个使用十六进制颜色值的颜色选择工具。

问：我需要这样一个表单域：它能够在用户键入时给出搜索项建议。jQuery UI提供了这方面的功能吗？

答：是的！jQuery 提供了Autocomplete部件，这是写这本书时（2011年8月）最新的特性之一。用户在表单域中键入时这个部件会显示搜索建议。

这些搜索项从哪里来呢？可以由一个JavaScript数组提供，或者来自一个URL，也可以使用一个回调函数，利用你在第8章和第9章学到的Ajax方法

从一个服务器获取数据。关于这个部件的更多信息请访问jQuery UI的演示页面：

<http://jqueryui.com/demos/#autocomplete/>

问：jQuery UI提供了某种表单验证吗？

答：没有。很遗憾，jQuery UI并没有提供表单验证，不过有一个jQuery插件可以很好地完成验证。可以在以下地址找到这个插件：

<http://bassistance.de/jquery-plugins/jQuery-validation/>

我们将在附录i中更多地讨论这个插件。相关信息请见附录i。

等一下。当然，我们确实从第1章就在使用按钮，这一点不假，不过这是一个jQuery UI按钮，而且是一个HTML表单元素。

没错，说得对。

我们用过jQuery的click方法和jQuery UI的button方法，不过还没有处理过选择表单元素（比如输入提交按钮）。下面给出一个选择表单元素的简单指南。

BULLET POINTS



- `$(":input")` = 选择所有输入元素
- `$(":text")` = 选择所有类型为text的输入元素
- `$(":radio")` = 选择所有类型为radio的输入元素
- `$(":checkbox")` = 选择所有类型为checkbox的元素
- `$(":submit")` = 选择所有类型为submit的输入元素
- `$(":reset")` = 选择所有类型为reset的输入元素
- `$(":checked")` = 选择所有未选中的输入元素
- `$(":selected")` = 选择所有已选择的输入元素
- `$(":enabled")` = 选择所有启用的输入元素
- `$(":disabled")` = 选择所有禁用的输入元素
- `$(":password")` = 选择所有用于输入密码的输入元素



写出my_scripts.js中要增加的一行代码，将普通的药片式按钮转换为指定主题的jQuery UI按钮。



一定要在my_scripts.js文件中增加这行代码，在你喜欢的浏览器中打开sightings_end.html进行测试。

```
$( "button:submit" ).button();
```

Submit Your Cryptid Sighting
Fill out the form below and click "Submit" to Submit.

CRYPTO SIGHTING DATA

Date of Sighting:
2013/01/01

Creature Type:
 Chupacabra Jersey Devil Loch Ness Monster Sasquatch Yeti Other

Distance From Creature (in ft):
100

Creature Weight (in lbs):
1000

Creature Height (in ft):
8

Color of Creature (use the color sliders to select):
Creature Head/Body: #A9A9A9
Red: 100 Green: 100 Blue: 100

Creature Head/Body: #A9A9A9
Red: 100 Green: 100 Blue: 100

CRYPTO LOCATION DATA

Latitude of Sighting:
41.9403

Longitude of Sighting:
-73.9891

你工作很让人满意，
下一周我们还会接待你呢！



呵，真是感激之至！
现在我们再无空日了！



附加练习

你的表单看上去很不错，但目前还不能真正提交任何数据。不过，你在第9章已经了解提交数据需要做什么。所以下面花点时间来考虑如何让这个漂亮的表单真正起作用。

这一章end文件夹中包含了完成这个工作所需的全部代码：sightings.html、service.php和sightings.sql。你要自己来完成初始工作。不过这正是Web开发人员应该做的，不是吗？我们包含了第9章学习的Ajax和JSON方法，这样刚才建立的表单就能提交数据了。

所有初始工作都要由你完成（也就是说，需要运行sightings.sql脚本，或者你自己创建数据库，其中包含这个脚本中的字段，并向my_scripts.js文件增加Ajax和JSON方法）。在继续学习第11章之前，还需要向数据库增加一些记录。如果需要复习MySQL、PHP和Ajax，可以再返回去看看第8章和第9章。

“www.ckook.com”是CKOOK公司旗下一个专注于前端设计与开发的资源类网站。
更多精彩资源尽在“www.ckook.com”，欢迎光临！



JQuery工具箱

你已经掌握了第10章，现在工具箱里又增加了jQuery UI。

jquery UI

这是一个官方jQuery库，主要是为jQuery核心库提供三类插件：效果、交互和部件。

部件

这是一个自包含的组件，可以为web应用增加功能。能够为你节省大量编码时间，降低复杂性，创建实用的响应式用户界面元素。

按钮部件

提供了一个button方法，可以帮助你建立更漂亮的表单元素，如提交按钮、单选框和复选框。

日期选择部件

datepicker方法将jQuery解释器与建立日期选择工具，以及大量的HTML、CSS和内置交互性。另外还提供了丰富的可定制选项。

滑动条

用户可以用鼠标或键盘管理这些UI元素，控制他们输入的数据。

有5个事件处理程序可以用来自将滑动条连接到表单输入域：create、start、slide、change和stop。

对象，无处不在



尽管你是颇有天赋的开发人员，但也不能一直单枪匹马……我们已经看到，可以包含jQuery插件（比如jQuery UI或标签页导航），这样不需要花太多功夫就能大大提升我们的jQuery应用。为了让我们的应用更上一个台阶，互联网上已经有一些相当酷的工具可以使用，还可以使用一些软件巨头提供的信息。比如Google、Twitter或Yahoo！，我们还需要……更多。这些公司为他们的服务提供了API（应用编程接口），可以在你的网站中包含这些API。这一章中，我们将介绍一些API基础知识，还会使用一个相当常用的API：Google Maps API。

更多资源请访问稀酷客(www.ckook.com)

~~沃利~~ 大脚怪在哪里？

Pattersby博士和Gimli博士希望再为他们的网站增加另外一些酷炫特性，他们认为你应该能胜任这个任务。

来自: Gimli博士 [gimli@cryptozoologists.org]

主题: 对网站再做一些更新

嗨，

有了你的帮忙，现在我们的网站对用户更友好，而且更容易收集神秘生物的观测数据，真是太感谢了。我们的流量有了大幅提升，看到收集到的所有这些数据实在让人兴奋。很多人对我们提出请求，希望大家能更容易地访问这些信息。很多人都对这些观测数据感兴趣。所以我们希望能提供一个途径，方便他们查看收集到的信息。

以下是我们想要的：

- (1) 希望能选择一个观测记录，查看与它关联的信息。除了关于神秘生物的信息，我们还希望看到显示在Google地图上的经/纬度信息。
- (2) 在Google地图上显示一个点时，我们希望能够点击这个点，查看关于观测记录的更多信息。
- (3) 希望能够从一个列表选择生物类型，并显示数据库中与这个类型关联的所有生物。我们还希望在一个Google地图上看到这个类型的所有生物，这样就能找到观测热点地区，做更深入的检查。希望所有这些点都是可点击的，另外生物列表也应该 是可点击的，这样用户就能看到彼此的更多信息了。

既然我们已经存储了所有信息，应该没有太多问题吧？

期待你的回音！

--
Pattersby博士和Gimli博士
cryptozoologists.org



Joe：你是说我们要构建的Google地图吗？

Frank：是啊，非常简单……

Jim：简单？他们想要一个完整的Google地图！

Frank：是呀，那又怎样？

Jim：地图上有很多点，每个点分别对应一个神秘生物——各个点还应当可点击，来得到更多信息……

Frank：是的，我知道该怎么做。

Jim：另外还要定制对列表的点击，与这些点交互，还要在地图上弹出显示“更多信息”。

Frank：噢，嗯，这方面我没有太大把握了。

Joe：别担心这一部分。Google Maps的开发人员提供了一个API，我们可以用这个API来完成工作。

Jim：API什么？

Frank：API。这是应用编程接口（Application Programming Interface）的缩写。像Google、Netflix和Yahoo!之类的公司就是通过提供API，允许我们在自己的网站上使用他们那些超棒的工具。

Jim：听上去真的很酷，不过它能为我们提供所有这些弹出窗口和需要在地图上放置的其他内容吗？

Joe：嗯，可以。也许我们该看看Google Maps API，了解它是如何工作的。

Google Maps API

要使用某个API时，可以查看它的文档，得到一些联机示例代码。我们从<http://code.google.com/apis/maps>得到了下面这个例子。

```

var map;
function initialize() {
  var myLatlng = new google.maps.LatLng(40.720721,-74.005966);
  var myOptions = {
    zoom: 13,
    center: myLatlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
  google.maps.event.addListener(map, 'zoom_changed', function() {
    setTimeout(moveToNewYork, 3000);
  });
  var marker = new google.maps.Marker({
    position: myLatlng,
    map: map,
    title:"Hello World!"
  });
  google.maps.event.addListener(marker, 'click', function() {
    map.setZoom(8);
  });
}
function moveToNewYork() {
  var NewYork = new google.maps.LatLng(45.526585, -122.642612);
  map.setCenter(NewYork);
}

```

OK，我发现这里有一些变量和函数。
不过另外那些东西是什么？

那是Google的代码。

不过没有看上去那么糟。稍后我就会详细介绍这个代码要做些什么。

更多资源请访问稀酷客(www.ckook.com)

API使用对象

第6章中我们创建了自己的JavaScript对象，其中包含属性和方法来存储和使用我们认为合适的信息。很多公司比如Google、Netflix、Microsoft和Yahoo!，也创建了API对象，利用这些API，允许我们与他们的数据交互。如果你有点忘了，完全可以返回到第6章，再复习一下对象以及它们是如何工作的。

```

    // 声明变量。
    var map;
    // 将这个变量视为一个新的
    // Google LatLng对象。
    var myLatlng = new google.maps.LatLng(40.720721, -74.005966);

    var myOptions = { ← 创建一个地图选项
        zoom: 13,           // 地图的一些选项。
        center: myLatlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    // 将map参数放在map_canvas
    // 元素中。
    map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);

    google.maps.event.addListener(map, 'zoom_changed', function() {
        setTimeout(moveToNewYork, 3000) ← 为这个Google地图增加一个
    });
}

```

```

    // 声明一个变量为
    // marker对象。
    var marker = new google.maps.Marker();
    // 将myLatlng参数
    // 传递给marker对象。
    { position: myLatlng, ← 前面声明的LatLng对象。
      map: map,           ← 之前声明的Map对象。
      title: "Hello World!" }
    };

    // 声明一个函数调用名
    // 外一节Google代码。
    function moveToNewYork() {
        var NewYork = new google.maps.LatLng(45.526585, -122.642612);
        map.setCenter(NewYork); ← 将map参数居中。
    }
}

```



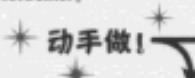
这也很API的一切工作都要用对象来完成，是吗？为什么要使用其他人创建的对象呢？

因为这会加快速度。

正如我们在第6章看到的，对象提供了更“聪明”的存储。如果需要存储关于一个特定事物的多个变量，就可以使用对象。API就是一系列对象构造函数，允许你为其他人的对象创建自己的实例。一旦有了实例，就能在你的代码中使用与这些对象关联的所有属性和方法！

创建一个地图应用供所有人使用，这会相当复杂。正因如此，Google在API中使用了对象。这样一来，它能创建包含很多不同方法和属性的对象，这些对象分别对应构建地图时需要的所有不同的元素。

从上一页的代码可以看到map对象有zoom、center和mapTypeId属性，另外还有很多属性在这里没有给出。除此以外，map对象还包含很多方法，如setCenter。



创建一个新页面，名为display_one.html，把它保存在对应这一章的项目文件夹中。

```
<!DOCTYPE html>
<html>
  <head>
    <title>View Cryptid Sightings</title>
    <link type="text/css" href="style/form.css" rel="stylesheet"/>
    <link type="text/css" href="jquery-ui-1.8.16.custom/css/sunny/jquery-ui-1.8.16.custom.css" />
  </head>
  <body>
    <div class="ui-widget-header ui-corner-top form_pad">
      <h2>View Cryptid Sightings</h2>
    </div>
    <div class="ui-widget-content form_pad"> 创建一个放置来放置
      <div id="map_canvas"></div> 地图。
    </div>
    <script src="http://maps.google.com/maps/api/js?sensor=false"></script> 包含 Google Maps API。
    <script src="scripts/jquery-1.6.2.min.js"></script> 包含jQuery。
    <script src="scripts/maps.js"></script>
  </body>
</html>
```



display_one.html

在页面中包含Google地图

首先，复制上一章最后完成的所有文件。这里我们要使用同样的文件，哪里缺就补哪里，所有那些代码再加上新的display_one.html文件，将提供两个重要的新内容：

- ID为map_canvas的div。
- Google Maps API代码（通过增加<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script>）。

要在这个页面上包含一个Google地图，需要创建一个maps.js文件，其中增加一个函数来调用API代码，在页面上构建一个地图。



jQuery代码磁贴

重新组织磁贴完成下面的代码，创建一个名为initialize的函数。这个函数要用代码中定义的参数创建Google Maps map对象的一个新实例。这个新map对象再应用到页面上的map_canvas元素。另外，更新现有的form.css文件，在其中包含地图窗口的一些样式定义。

```
#map_canvas {
    position: absolute;
    left: 10px;
    height: 450px;
    width: 450px;
}

$(document).ready(function() {
    var map;
    function initialize() {
        var latlng = new google.maps.LatLng(45.519098, -122.672138);
        var mapOptions = {
            zoom: 13,
            center: latlng,
            mapTypeId: google.maps.MapTypeId.HYBRID
        };
        map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
    }
    initialize();
});

```

#map_canvas {
position: absolute;
left: 10px;
height: 450px;
width: 450px;
}



form.css



maps.js

(45.519098, -122.672138)

new

450px

initialize()

(function() {

#map_canvas

更多资源请访问稀酷客(www.ckook.com)



jQuery代码磁贴答案

重新组织磁贴完成下面的代码，创建一个名为initialize的函数。这个函数要用代码中定义的参数创建Google Maps map对象的一个新实例。这个新map对象再应用到页面上的map_canvas元素。另外，更新现有的form.css文件，在其中包含地图窗口的一些样式定义。

```

$(document).ready(function() {
    var map;
    function initialize() {
        var latlng = new google.maps.LatLng(45.519098,-122.672138);
        var mapOpts = {
            zoom: 13,
            center: latlng,
            mapTypeId: google.maps.MapTypeId.HYBRID
        };
        map = new google.maps.Map(document.getElementById("map_canvas"), mapOpts);
    }
    initialize();
});

```

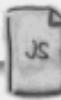
```

#map_canvas {
    float: left;
    height: 450px;
    width: 450px;
}

```



form.css



maps.js

there are no
Dumb Questions

问：LatLng对象和mapOpts属性是什么？

答：在<http://code.google.com/apis/maps/documentation/javascript/reference.html>上可以找到关于这个API的对象、方法等更多信息。这个网站提供了一些示例代码，还提供了需要与之交互的所有不同对象和方法的更多详细信息。

问：Google Maps是创建地图的唯一选择吗？

答：当然不是！不过这可能是最流行的，正是因为这个原因，这里我们将使用Google Maps API。其他公司（比如Yahoo!、Microsoft、MapQuest和OpenLayers）也提供了地图API。



试一试

使用之前由磁贴组织得到的initialize函数更新maps.js文件。另外，确保display_one.html中包含maps.js文件。然后在浏览器中打开display_one.html。要通过Web服务器运行所有代码，所以URL应当是http://而不是file://。



真是太棒了，在页面上得到一个地图居然这么容易，不过我还没看到任何物种和生物数据。我们需要从数据库读取这些数据，对吗？

完全正确。

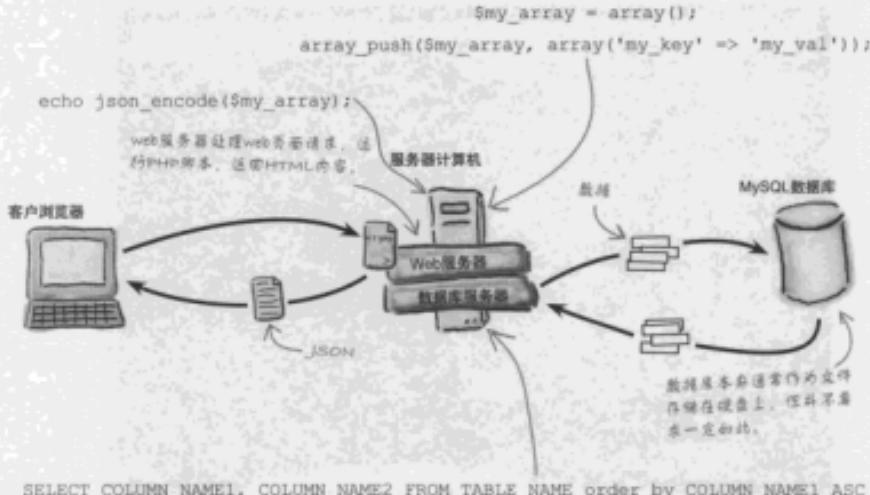
在第9章中，你已经了解了如何使用jQuery、Ajax、JSON、PHP和MySQL从一个MySQL数据库获取信息。这涉及很多技术，不过确实能很好地完成我们要求的工作。下面来看在这里如何应用。

更多资源请访问稀酷客(www.ckook.com)

用SQL和PHP获取JSON数据

第9章展示了SQL select语句如何从数据库取出你想要的信息，然后由一个PHP文件将它转换为JSON，并使用Ajax返回到页面。

你还了解了如何使用Ajax从一个PHP文件得到JSON编码的信息。要让PHP返回JSON数据，这很容易，只需要使用`json_encode`函数，它接受一个数组作为参数，返回JSON编码的数据，以便jQuery与之交互。



```
SELECT COLUMN_NAME1, COLUMN_NAME2 FROM TABLE_NAME order by COLUMN_NAME1 ASC
```

我们已经为你编写了这一章的所有PHP和SQL代码。只要有第10章的MySQL数据库，就可以开始下面的工作了！其余的SQL和PHP代码都在本章的下载代码中。可以在你自己的服务器上运行这些代码。所有PHP和SQL代码可以从<http://thinkjquery.com/chapter11/end/service.zip>下载（都压缩在一个文件中）。

更多资源请访问稀酷客(www.ckook.com)



jQuery、HTML和CSS代码磁贴

重新组织磁贴，更新你的display_one.html、forms.css和maps.js文件，通过JSON获取数据，并在屏幕上显示。增加一个div和一个ul来包含这些数据。另外增加一些CSS为列表建立样式，还要增加一个函数获取数据（通过JSON），并向列表增加各个神秘生物。

```
function getAllSightings() {
    $.getJSON("service.php?action=getAllSightings", _____) {
        if (_____length > 0) {
            $('#sight_list')_____.each(function() {
                var info = 'Date: ' + this['date'] + ', Type: ' + this['type'];
                var $li = $("<li>" + info + "</li>");
                $li.addClass(_____);
                $li.attr('id', this['id']);
                $li.appendTo(_____);
            });
        }
    }
}
```



maps.js

"map_canvas"

"sightings"

json.sightings.

#sight_navi

float:left;

)
ul#sight_list{
width:150px;
padding:0px;
margin:0px;
}
li.sightings{
padding:4px;
background:#7B7382;
border:1px #000 solid;
color:#fff;
}

}
background:#eee;
color:#000;
}

```
<div class="ui-widget-content form_pad">
<div id="sight_navi"></div>
<div id="sight_list">
<ul id="sight_list">
</ul>
</div>
</div>
```



display_one.html

\$.empty()

\$.each

list-style:none;

"sight_list"

"#sight_list"

function(json)



form.css

jQuery、HTML和CSS代码磁贴答案

重新组织磁贴，更新你的display_one.html、forms.css和maps.js文件，通过JSON获取数据，并在屏幕上显示。增加一个div和一个ul来包含这些数据，另外增加一些CSS为列表建立样式，还要增加一个函数获取数据（通过JSON），并向列表增加各个神秘生物。

```
function getAllSightings(){
    $.getJSON("service.php?action=getAllSightings", function(json) {
        if (json.sightings.length > 0) {
            $("#sight_list").empty();
            $.each(json.sightings, function() {
                var info = "Date: " + this['date'] + ", Type: " + this['type'];
                var $li = $(<li />);
                $li.html(info);
                $li.addClass("sightings");
                $li.attr('id', this['id']);
                $li.appendTo("#sight_list");
            });
        }
    });
}
```

```
<div class="ui-widget-content form_pad">
    <div id="map_canvas"></div>
    <div id="sight_nav">
        <ul id="sight_list">
        </ul>
    </div>
</div>
```

display_one.html

```
#sight_nav {
    float:left;
}
ul#sight_list{
    width:150px;
    padding:0px;
    margin:0px;
}
li.sightings {
    padding:4px;
    background:#7B7382;
    border:1px #000 solid;
    color:#fff;
    list-style:none;
}
li.sightings:hover {
    background:#eee;
    color:#000;
}
```



maps.js



form.css



试一试

用刚才完成的getAllSightings函数更新你的maps.js文件。另外，在initialize函数的最后增加这个新函数调用。然后在浏览器中打开display_one.html。这里假设你已经在第10章向数据库中增加了一些神秘生物。如果还没有，请现在着手增加。要记住，一定要通过web服务器运行你的所有代码，所以URL应当是http://而不是file://。

Date	Type	Name
2013-09-04	Loch Ness Monster	
1967-10-20	Bigfoot	
2011-08-01	Chupacabra	
2011-08-04	Chupacabra	
2011-08-12	Loch Ness Monster	
2011-08-13	Bigfoot	
2011-08-13	Loch Ness Monster	
2011-08-13	Yeti	
2011-08-18	Sasquatch	

看得很不错，不过尽管增加了所有这些代码，地图上还是没有具体显示任何东西。

是的。需要把我们的神秘生物增加到地图上。

Google Maps提供了一个非常容易的方法来完成这个任务。下面就来看看这个方法如何工作。

更多资源请访问稀酷客(www.ckook.com)

地图上的点是标记

要在地图上放置点，Google就可以搞定。不过，Google并不把它们叫做点，而是称之为标记（marker）。标记都是对象，就像Google Maps API中的所有其他对象一样，它们也有自己的方法和属性来完成交互和管理。

```
从Google Maps API定义一个新的
LatLang对象。
var myLatLang = new google.maps.LatLng(45.519098,-122.672138);
```

定义一个名为my_marker的对象，这是
Google Marker对象的一个实例。

```
传递具体的纬度和
经度值作为参数。
var my_marker = new google.maps.Marker({
```

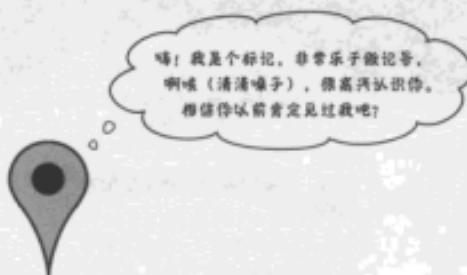
调用Google Marker
对象的构造函数。

```
position: myLatLang,           // 将刚才的LatLang对象，它作为一个参数。
map: map,                      // 定义的map
title:"Hi! I'm a marker. Seen me around before?"
```

将刚才的map
对象。你已经
见过这个对象
的使用。
});

将参数传递给my_marker对象的属性。

在这里，我们设置了他的位置，把他
增加到那个地图，以及这个点的标题。



Sharpen your pencil

更新getAllSightings函数，将列表项增加到列表之前，要向列表项增加一个点击事件监听者。这个点击事件要调用一个定制函数，名为getSingleSighting，它取一个参数，也就是所点击的观测信息的ID。这个新函数再加载所点击元素的有关信息，使用它的纬度和经度属性，并把它作为一个标记增加到地图上。

```

function getAllSightings() {
    $.getJSON("service.php?action=getAllSightings", function(json) {
        if (json.sightings.length > 0) {
            $("#sight_list").empty();
            $.each(json.sightings, function() {
                var info = "Date: " + this['date'] + ", Type: " + this['type'];
                var $li = $(<li />');
                $li.html(info);
                $li.addClass("sightings");
                $li.attr('id', this['id']);
                $li.click(function() {
                    _____ this['id']);
                });
                $li.appendTo("#sight_list");
            });
        }
    });
}

function getSingleSighting(_____) {
    $.getJSON("service.php?action=getSingleSighting&id=" + id, function(json) {
        if (json.sightings.length > 0) {
            var loc = new google.maps.LatLng(this['lat'], this['long']);
            var my_marker = new google.maps_____({
                _____
                loc,
                map: map,
                title: this['type']
            });
            _____setCenter(loc, 20);
        }
    });
}

```



maps.js

更多资源请访问稀酷客(www.ckook.com)



Sharpen your pencil

Solution

完成代码后，你的列表项就是可点击的了，这会加载所点击神秘生物的有关数据，并把数据增加到地图上。

```

function getAllSightings() {
    $.getJSON("service.php?action=getAllSightings", function(json) {
        if (json.sightings.length > 0) {
            $("#sight_list").empty();
            $.each(json.sightings, function() {
                var info = 'Date: ' + this['date'] + ', Type: ' + this['type'];
                var $li = $( "<li>" + info + "</li>" );
                $li.html(info);
                $li.addClass("sightings");
                $li.attr('id', this['id']);
                $li.click(function(){
                    getSingleSighting(this['id']);
                });
                $li.appendTo("#sight_list");
            });
        }
    });
}

function getSingleSighting(id) {
    $.getJSON("service.php?action=getSingleSighting&id=" + id, function(json) {
        if (json.sightings.length > 0) {
            $.each(json.sightings, function() {
                var loc = new google.maps.LatLng(this['lat'], this['long']);
                var my_marker = new google.maps.Marker({
                    position: loc,
                    map: map,
                    title: this['type']
                });
                map.setCenter(loc, 20);
            });
        }
    });
}

```



maps.js



试一试

用你刚才完成的getAllSightings和getSingleSighting函数更新maps.js文件。然后在浏览器中打开display_one.html。要像前面一样使用`http://`打开这个页面。

Date	Type	Name
2011-08-04	Unknown	Monster
2011-08-09	Unknown	
2011-08-11	Unknown	
2011-08-14	Unknown	
2011-08-15	Unknown	



你已经完成了前两项需求，下面来看博士们最后一个要求。

- (3) 希望能够从一个列表选择生物类型，并显示数据库中与这个类型关联的所有生物。我们还希望在一个Google地图上看到这个类型的所有生物，这样就能找到观测热点地区，做更深入的检查。希望所有这些点都是可点击的，另外生物列表也应该是可点击的，这样用户就能看到彼此的更多信息了。

更多资源请访问稀酷客(www.ckook.com)

多生物列表

要完成这最后一项需求，我们需要：

1. 一个下拉列表，包含（从数据库中选择的）生物类型的列表。
2. 这个下拉列表改变时，从数据库得到一个与选择类型匹配的生物列表。
3. 在一个列表中显示从数据库返回的所有生物，并显示在地图上。
4. 列表和地图表示都应该是可点击的，能够在地图上的用户弹出更多信息。



成品代码

创建一个新页面，名为display_type.html，把它保存在这个项目其他HTML文件所在的同一个目录中。这个新文件会显示可选择的生物类型列表。一旦选择，会在地图上显示这种类型的所有生物。这个新页面的结构和样式与原来的页面非常类似，只是增加了一个ID为ddlTypes的select元素。

```
<!DOCTYPE html>
<html>
    <head>
        <title>View Cryptid Sightings</title>
        <link type="text/css" href="style/form.css" rel="stylesheet" />
        <link type="text/css" href="jquery-ui-1.8.16.custom/css/sunny/
jquery-ui-1.8.16.custom.css"/>
    </head>
    <body>
        <div class="ui-widget-header ui-corner-top form_pad">
            <h2>View Cryptid Sightings</h2>
        </div>
        <div class="ui-widget-content form_pad">
            <div id="map_canvas"></div>
            <div id="sight_nav">
                <select id="ddlTypes"> ↗ 增加一个下拉列表，其中将填充
                    <option value="">-- Please Select --</option>
                </select>
                <ul id="sight_list"></ul>
            </div> ↗ 包含Google Maps API。
            <script src="http://maps.google.com/maps/api/js?sensor=false"></script>
            <script src="scripts/jquery-1.6.2.min.js"></script>
            <script src="scripts/maps.js"></script> ↗ 包含maps.js文件。
        </body>
    </html>
```



更多资源请访问稀酷客(www.ckook.com)



jQuery代码磁贴

重新组织磁贴完成getAllTypes函数。这个函数会调用service.php文件（应该包含在这一章的下载文件中），得到数据库中不同生物类型的一个列表。这些生物类型再增加到ID为ddlTypes的下拉列表中。另外，为下拉列表增加一个事件监听者，监听change事件，并提示所选择的值。最后，由于我们对两个HTML文件使用了同样的maps.js文件，要在initialize函数中增加一些逻辑检查下拉列表是否存在。如果存在，则调用getAllTypes函数，否则就调用getAllSightings函数。

```

function initialize() {
    ...

    map = new google.maps.Map(document.getElementById("map_canvas"), mapOpts);
    if ($("#ddlTypes").length) {
        ...
    } else {
        ...
    }
}

function getAllTypes() {
    $.getJSON("service.php?action=getSightingsTypes", function(json_types) {
        if (creature_types.length > 0) {
            $.each(json_types.creature_types, function(i, creature_type) {
                var info = this['type'];
                var $li = $("- " + creature_type + "$li.html(info);
");
                $li.appendTo($("#ddlTypes"));
            });
        }
    });
}

```

map_canvas
getAllTypes();
.appendTo
function() {
 \$("#ddlTypes").
 change(function() {
 if(\$("#this").val() != ""){
 alert(\$("#this").val());
 }
 });
}





jQuery代码磁贴答案

现在已经有了一个与Google地图关联的下拉列表，一个从数据库获取所选择生物类型相关数据（JSON格式）的函数，还有一个提醒，会显示选择了哪个生物类型。

```
function initialize(){
    ...

    map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
    if ($('#ddlTypes').length) {
        getAllTypes();
    } else {
        getAllSightings();
    }
}

function getAllTypes() {
    $.getJSON("service.php?action=getSightingsTypes", function(json_types) {
        if (json_types.creature_types.length > 0) {
            $.each(json_types.creature_types, function() {
                var info = this["type"];
                var $li = $("");
                $li.html(info);
                $li.appendTo("#ddlTypes");
            });
        }
        $('#ddlTypes').change(function() {
            if($(this).val() != "") {
                alert( $(this).val() );
            }
        });
    });
}
```

使用Length属性检测元素是否存在。

使用JSON和PHP以数组读取到类型。

设置下拉列表项的文本，→\$li.html(info);
为下拉列表追加选择项。

为下拉列表的change事件增加一个监听者。

向表单中所选项目的值。

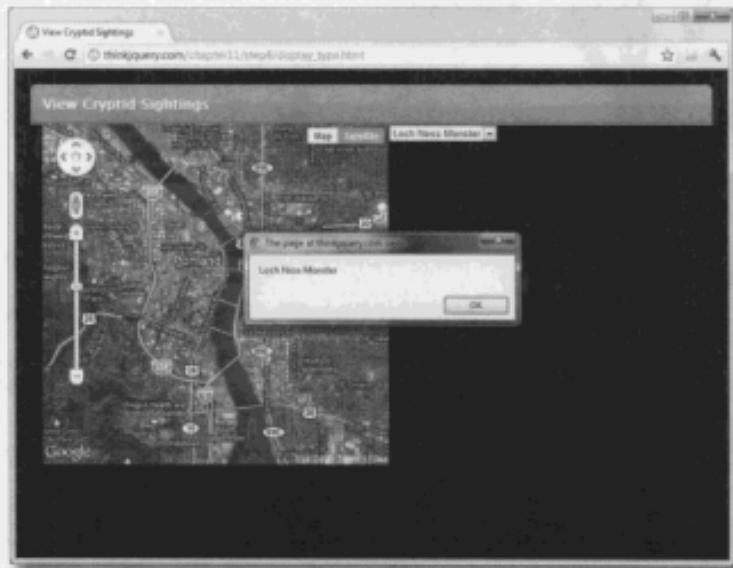


maps.js



试一试

增加getAllTypes函数和下拉列表change事件监听者，更新maps.js文件。另外，采用新逻辑更新initialize函数。然后在浏览器中打开display_type.html，同样要使用http://。



你真棒。这么快就完成了这个地图！现在划掉其中两个需求。

1. 一个下拉列表，包含（从数据库中选择的）生物类型的列表。
2. 当一个下拉列表改变时，从数据库得到一个与该类型匹配的生物列表。
3. 在一个列表中显示从数据库返回的所有生物，并显示在地图上。
4. 列表和地图指标都应该是可点击的，能够在地图上的用户弹出更多信息。

更多资源请访问稀酷客(www.ckook.com)



嘿！别太着急！从下拉列表选择一项时，我没有看到任何生物。我只看到弹出一个提醒。我不认为第2个菜单已经完成！

你说的有道理。

我们需要更新代码，在下拉列表改变时从数据库获取数据，而不只是在一个提醒框中显示生物类型。

在此之后我们才能把这个要求从列表中划去。不过，一旦完成这一点，就能把列表中的第3个要求也去掉了。加油—接下来就要深入了，我们会把这些都集成在一起。



LONG Exercise

填入缺少的代码行，创建一个getSightingsByType函数。这个函数要接收一个参数，也就是你查看的生物类型。这个函数要以JSON格式获取数据，循环处理返回的所有生物（如果存在），并为每个生物在地图上增加相应的点。另外创建两个全局变量：一个是名为markerArray的数据，还有一个是名为bounds的新Google Maps LatLngBounds对象。另外，创建一个函数，如果下拉列表有变化，在增加新的点之前先清除之前的点。

```
var markersArray = [];
var bounds = new google.maps_____;
function getSightingsByType(type) {
  $.getJSON("service.php?action=getSightingsByType&type="+type, function(json)
  {
    if (_____sightings.length > 0) {
      $('#sight_list').empty();
      $.each(json.sightings, function() {
        var loc = new google.maps_____([this['lat'], this['long']]);
        var opts = {
          map: map,
          position: _____
        };
    }
  });
}
```

更多资源请访问稀酷客(www.ckook.com)

```

var marker = new google.maps.Marker({opts});
    markersArray.push(marker);
    var $li = $("- 
");
    $li.html("Date: " + this["date"] + ", Type: " + this["type"]);
    $li.append("sightings");
    $li.appendTo("#sight_list");
    bounds.extend(loc);
});
map.fitBounds(bounds);
}
});
}

$("#ddlTypes").change(function() {
if($(this).val() != "") {
    clearOverlays();
    $(this).val();
}
});
function clearOverlays() {
if(markersArray) {
    for(i in markersArray) {
        markersArray[i].setMap(null);
    }
    markersArray.length = 0;
    bounds = null;
    bounds = new google.maps.LatLngBounds();
}
}
}

```



maps.js



LONG Exercise Solution

通过增加两个新的全局变量和另外一些Google Maps函数，现在下拉列表改变时可以向地图增加和删除标记了。标记增加到地图上的同时，还会增加到markerArray数组，并用于扩展地图的边界。这样一来，地图可以使用fitBounds函数自动缩放来包含所有的点。现在下拉列表有任何改变时就会调用getSightingsByType函数，向地图增加标记，并为页面上的列表增加生物。

```

var markersArray = [];
var bounds = new google.maps.LatLngBounds();
function getSightingsByType(type) {
    $.getJSON("service.php?action=getSightingsByType&type=" + type, function(json) {
        if (json.sightings.length > 0) {
            $('#sight_list').empty();
            $.each(json.sightings, function() {
                var loc = new google.maps.LatLng(this['lat'], this['long']);
                var opts = {
                    map: map,
                    position: loc
                };
                // 在地图上的各个点创建一个新Marker对象。
                var marker = new google.maps.Marker(opts);
                markersArray.push(marker);
                var $li = $(<li />);
                $li.html('Date: ' + this['date'] + ', Type: ' + this['type']);
                $li.addClass("sightings");
                $li.appendTo("#sight_list");
                bounds.extend(loc); // 加到bounds对象。
            });
            map.fitBounds(bounds);
        }
    });
}

```

↑ 一个新的LatLngBounds对象。
↑ 用JSON数组数据。
↑ 在地图上的各个点创建一个新Marker对象。
↑ 加到bounds对象。
↑ 通知地图使用bounds正确地缩放，以便看到所有标记。

```

$( "#ddlTypes" ).change(function() {
    if( $(this).val() != "" ){
        clearOverlays();
        getSightingsByType( $(this).val() );
    }
});
function clearOverlays(){
    if (markersArray) {
        for (i in markersArray) {
            markersArray[i].setMap(null);
        }
        markersArray.length = 0;
        bounds = null;
        bounds = new google.maps.LatLngBounds();
    }
}

```

将下拉列表的值作为一个参数传入我们的函数。

从地图删除这个标记。

还原重置bounds变量。



maps.js



Geek Bits

我们包含了一个名为clearOverlays的函数。它会在增加新标记之前删除之前增加的所有标记。Google将增加到地图上的所有内容称为一个重叠（overlay），这可以是一个标记、一条线、一个多边形，或者很多其他类型的对象。



试一试

用刚创建的新代码更新maps.js文件。然后在浏览器中打开display_type.html，从列表中选择一些生物。



1. 一个下拉列表，包含（从数组中选择的）生物类型的列表。
2. 这个下拉列表被选中时，从数组中得到一个值选择类型对应的生物列表。
3. 在一个列表中显示从数组中选出的所有生物，并显示在地图上。
4. 列表和地图表示都应该是可点击的，能够在地图上为用户弹出更多信息。

就快完成了……只剩下一个任务！



你已经知道如何使用jQuery使元素可点击。这对于完成任务列表中的最后一个需求有帮助吗？

there are no Dumb Questions

问：那么，我是不是可以在我的网站上免费使用Google Maps API？

答：是的！无论什么人想使用这个API，Google都允许他免费使用—不论是否个人使用还是商业使用—只要他们遵守服务条款。

问：我不清楚我是不是遵守这些条款。在哪里可以看到这些服务条款？

答：只需要浏览<http://www.google.com/apis/maps/terms.html>，就可以查看完整的服务条款。

问：Google Maps API覆盖全世界吗？

答：不全是。不过很接近了。只有少数几个国家没有覆盖到。可以访问Google Maps API网站查看这个列表。

问：我能在移动设备上使用Google Maps API显示地图吗？

答：没错，完全可以。这本书出版时，Google Maps API版本3已经发布，这个版本专为完全使用手机设备

只要它们的浏览器能够运行JavaScript。
问：不过我还是更愿意编写一个应用。Google Maps API适合我吗？

答：如果你在为Android或iPhone平台编写应用，答案是肯定的。Google针对这两个平台提供了特定的框架，可以包含在你的应用中。如果你在为其他移动平台编写应用，还没有特定的框架，就将使用与网站相同的框架。

问：Google Maps的完整版本还允许获取方向。这个API能做到吗？

答：不行。这个API还不能做到这一点。Google还提供了另一个API，名为Google Directions API，可以用它来查找方向。

问：要按地址查找地图上的位置呢？

答：业界称之为地理编码。为此Google还提供了另一个API：Geocoding API。这些都是Google Maps API家族的组成部分。另外，你很幸运，只要从Google请求数据，它就会向你返回JSON数据。对于JSON数据你应当已经有所了解了吧？

问：Google Maps API家族中还有什么？

答：Directions和Geocoding API都属于一个名为Maps API Web服务的子部分。除此以外，在这个子部分中还有Distance API、Elevation API和Places API。

问：还有吗？

答：嗯，对于不能完全支持JavaScript的浏览器还有Static Maps API和面向Flash的Maps API，甚至还有一个Earth API，允许你在页面中加载一个Google Earth浏览器，得到地球的3D图像，还可以做虚拟旅行，在地形上绘制形状等等。不过，这个API还要求安装Google Earth插件。

问：那么除了JavaScript，还可以得到面向其他语言的API吗？

答：没问题！这种API不计其数—有些免费，有些需要授权许可。看情况了，如果要找一个特定功能，而你不想自己来编写，很可能其他地方已经有人提供了这样一个API。

监听地图事件

就快完成了，现在你已经看到jQuery和JavaScript提供了大量事件来建立有趣的交互式web页面。由于Google Maps API就是JavaScript（不过是精心编写的高质量、高效的JavaScript），它也可以利用浏览器的功能来监听事件，并相应地做出反应。

jQuery增加了自己的事件监听者创建函数，出于同样的原因，Google Maps的开发人员也提供了类似功能，可以通过这个API增加事件监听者。这是因为，并不是所有浏览器都采用同样的方式处理事件监听者，所以这样能确保这个API可以控制页面如何增加监听者。下面来看如何为点击事件增加一个事件监听者来创建一个Google Maps弹出窗口（也称为InfoWindow）：

用将要显示的内容定
义一个变量。

```
var contentString = "This is an InfoWindow";
```

创建Google Maps
infowindow对象
的一个实例。 });

content: contentString

设置infowindow对象
content属性的值。

告诉地图监听marker对
象的点击事件。

```
google.maps.event.addListener(my_marker, 'click', function() {  
    my_infowindow.open(map,my_marker);  
});
```

点名标记时运行这个代码（在地
图上打开一个输出窗口）。



Geek Bits

在Google Maps API中，几乎所有不同的对象类型（Map, Marker, Line, InfoWindow, TrafficOverlay, Polygon等）都有与之关联的事件。不过，尽管不同对象的事件可能有相同的事件名，但它们也许有不同的参数！想要使用某个对象时，一定要检查这个对象的相关文档。

更多资源请访问稀酷客(www.ckook.com)

Sharpen your pencil

填入缺少的代码，完成getSightingsByType函数，它会为地图上的标记和列表项增加点击功能。另外，创建一个全局变量，名为info_window，这是Google Maps InfoWindow对象的一个新实例，其默认内容设置为一个空串。

```

var info_window = new google.maps _____ ([content: ''});
function _____ (type) {
  $.getJSON("_____ action=getSightingsByType&type=" + type, function(json) {
    if (json.sightings.length > 0) {
      $('#sight_list').empty();
      $.each(json.sightings, function() {
        var info = 'Distance: ' + this[_____] + '<br>' + ' Height: ' + this['height'];
        info += ', Weight: ' + this['weight'] + ', Color: ' + this['color'] + '<br>';
        info += 'Latitude: ' + this['lat'] + ', Longitude: ' + this[_____];
        var loc = new _____ (this['lat'], this["long"]);
        var opts = {
          map: map,
          position: _____
        };
        var marker = new google.maps _____ (opts);
        markersArray.push(marker);
        google.maps.event_____(marker, 'click', function() {
          info_window.content = info;
          info_window.open(map, marker);
        });
        var $li = $(<li />);
        $li.html('Date: ' + this['date'] + ', Type: ' + this['type']);
        $li.addClass("sightings");
        $li_____(function() {
          info_window.content = info;
          info_window.open(map, _____);
        });
        $li.appendTo("#sight_list");
        _____ extend(loc);
      });
      map _____ (bounds);
    }
  });
}

```



Sharpen your pencil

Solution

你已经更新了getSightingsByType函数，将采用列表格式显示，还会在地图上显示，另外会使地图标记和列表可点击。

```

var info_window = new google.maps.InfoWindow({content: ''});
function getSightingsByType(type) {
  $.getJSON("service.php? action=getSightingsByType&type=" + type, function(json) {
    if (json.sightings.length > 0) {
      $('#sight_list').empty();
      $.each(json.sightings, function() {
        var info = 'Distance: ' + this['distance'] + '<br>' + ' Height: ' + this['height'];
        info += ', Weight: ' + this['weight'] + ', Color: ' + this['color'] + '<br>';
        info += 'Latitude: ' + this['lat'] + ', Longitude: ' + this['long'];
        var loc = new google.maps.LatLng(this['lat'], this['long']);
        var opts = {
          map: map,
          position: loc
        };
        var marker = new google.maps.Marker(opts);
        markersArray.push(marker);
        google.maps.event.addListener(marker, 'click', function() {
          info_window.content = info;
          info_window.open(map, marker);
        });
        var $li = $(/*li */);
        $li.html('Date: ' + this['date'] + ', Type: ' + this['type']);
        $li.addClass("sightings");
        $li.click(function() {
          info_window.content = info;
          info_window.open(map, marker);
        });
        $li.appendTo("#sight_list");
        bounds.extend(loc);
      });
      map.fitBounds(bounds);
    }
  });
}

```

为地图上的marker增加一个对应点击事件的事件监听者。

为列表项增加一个事件监听者，在地图上打开infoWindow。





试一试

用上一页创建的新代码更新maps.js文件。然后，在浏览器中打开display_type.html，从列表中选择一个生物。点击列表中的标记来查看关于这个生物的更多信息。



1. 一个下拉列表，包含（从数据库中选择的）生物类型的列表。
2. 这个下拉列表被选中时，从数据库得到一个与选择类型匹配的生物列表。
3. 在一个列表中显示从数据库返回的所有生物，并显示在地图上。
4. 列表和地图显示都是可交互的，能够单击地图并为用户提供更多信息。

大功告成！

只需短短几个页面，你就组建了一个功能完备的网站，这里使用了不同语言的代码（PHP、SQL、JavaScript和jQuery），另外还结合了Google Maps API以及Ajax和JSON显示一些非常复杂的数据。功劳不小！

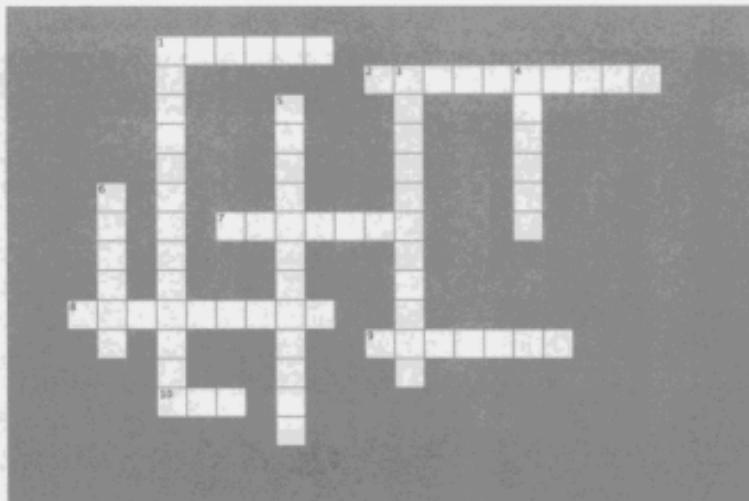


谢谢莫非带来的辛苦
一五一十



API填字游戏

现在该坐下来让你的左脑动一动了。这是一个标准的填字游戏。所有答案都可以在这一章中找到。



横向

1. Google Maps API构造函数代码。允许在google地图上放置一个点。google.maps: new google.maps._____。
2. 第11章中使用的Google Maps API用这种语言编写。
- 7.一旦有了一个API对象的实例，可以在代码中使用与这些对象关联的所有属性和_____。
8. API = 应用编程_____。
9. 我们利用这个jQuery方法使用JSON和PHP从观测数据库获取数据。
10. 这个Google Maps API对象有以下属性：zoom, center和mapTypeId。

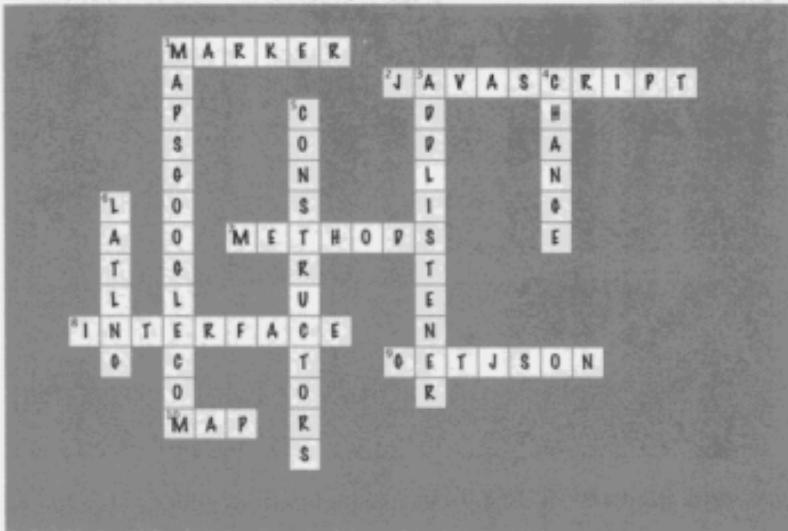
纵向

- 1.要在你的Web页面中包含Google Maps API，需要使用以下脚本标记：`<script src="http://_____maps.api/js?sensor=false"></script>`。
- 3.这个Google Maps API方法监听一个标记上的点击事件：google.maps.event._____。
- 4.这一章中使用的jQuery事件监听者。一个输入域中的数据值改变时就会触发这个监听者。
5. API是一系列方法_____，允许你为API提供者的对象创建自己的实例。
- 6.这个Google Maps API对象构造函数允许传入纬度和经度作为参数。

更多资源请访问稀酷客(www.ckook.com)



API填字游戏答案

更多资源请访问稀酷客(www.ckook.com)



jQuery API工具箱

你已经掌握了第11章，现在了解了如何将jQuery和各种API（以及JavaScript、PHP、MySQL、Ajax、JSON等）完美集成。

API

应用编程接口（Application Programming Interfaces）就是其他人（或公司）提供的代码，可以利用这些代码来处理他们的数据、对象和其他服务。

API提供了一系列对象构造函数，允许你为其他对象创造你自己的实例。一旦有了实例，就可以在代码中使用与这些对象关联的所有属性和方法。

离开jQuery镇……



能在jQuery镇遇到你们真是太好了！

你们要离开真让人难过，不过现在你们已经具备必要的全部技能，可以使用jQuery自己构建超酷的网站了。相信你肯定会做点什么，而不只是随便逛逛。很高兴带你参观jQuery世界。可以随时联系我们，或者在Head First Labs告诉我们你开发的新网站：<http://headfirstlabs.com>

更多资源请访问稀酷客(www.ckook.com)

附录 i：其他

十大主题(我们没有谈到的)



毕竟，还有很多内容我们没有谈到。还有很多其他jQuery和JavaScript小点心我们没办法都塞在这本书里。如果完全不提及有些不太公平，所以这里会简单做个介绍，让你能做更好的准备，应对旅程中可能遇到的其他jQuery问题。

更多资源请访问稀酷客(www.ckook.com)

#1. jQuery库全貌大观

你现在可能已经认识到jQuery是一个相当庞大的库。我们已经很努力了，希望能够覆盖最初接触jQuery的人需要了解的主要内容。现在你已经掌握了所有这些知识，以后可以进一步了解这个库的其他内容。

jQuery方法

.add()	deferred.pipe()	.fadeOut()
.addClass()	deferred.promise()	.fadeTo()
.after()	deferred.reject()	.fadeToggle()
jQuery.ajax()	deferred.rejectWith()	.filter()
.ajaxComplete()	deferred.resolve()	.find()
.ajaxError()	deferred.resolveWith()	.first()
jQuery.ajaxPrefilter()	deferred.then()	.focus()
.ajaxSend()	.delay()	.focusin()
jQuery.ajaxSetup()	.delegated()	.focusout()
.ajaxStart()	.dequeue()	jQuery.fx.interval
.ajaxStop()	jQuery.dequeue()	jQuery.fx.off
.ajaxSuccess()	.detach()	jQuery.get()
.andSelf()	.die()	.get()
.animated()	jQuery.each()	jQuery.getJSON()
.append()	.each()	jQuery.getScript()
.appendTo()	.empty()	jQuery.globalEval()
.attr()	.end()	jQuery grep()
.before()	.eq()	.has()
.bind()	.error()	.hasClass()
.blur()	jQuery.error	jQuery.hasData()
jQuery.browser	event.currentTarget	.height()
.change()	event.data	.hide()
.children()	event.isDefaultPrevented()	jQuery.holdReady()
.clearQueue()	event.	.hover()
.click()	isImmediatePropagationStopped()	.html()
.clone()	event.isPropagationStopped()	jQuery.inArray()
.closest()	event.namespace	.index()
jQuery.contains()	event.pageX	.innerHeight()
.contents()	event.pageY	.innerWidth()
.context	event.preventDefault()	.insertAfter()
.css()	event.relatedTarget	.insertBefore()
jQuery.cssHooks	event.result	.is()
.data()	event.stopImmediatePropagation()	jQuery.isArray()
jQuery.data()	event.stopPropagation()	jQuery.isEmptyObject()
.dblclick()	event.target	jQueryisFunction()
deferred.always()	event.timeStamp	jQuery.isPlainObject()
deferred.done()	event.type	jQuery.isWindow()
deferred.fail()	event.which	jQuery.isXMLDoc()
deferred.isRejected()	jQuery.extend()	jQuery()
deferred.isResolved()	.fadeIn()	.jquery

#1. jQuery库全貌大观（续）

jQuery方法（续）

```

.keydown()
.keypressed()
.keyup()
.last()
.length
.live()
.load()
.load()
jQuery.makeArray()
.map()
jQuery.map()
jQuery.merge()
.mousedown()
.mouseenter()
.mouseleave()
.mousemove()
.mouseout()
.mouseover()
.mouseup()
.next()
.nextAll()
.nextUntil()
jQuery.noConflict()
jQuery.noop()
.not()
jQuery.now()
.offset()
.offsetParent()
.one()
.outerHeight()
.outerWidth()
jQuery.param()
.parent()
.parents()
.parentsUntil()
jQuery.parseJSON()
jQuery.parseXML()
.position()
jQuery.post()
.prepend()
.prependTo()
.prev()
.prevAll()
.prevUntil()
.promise()
.prop()
jQuery.proxy()
.pushStack()
.queue()
.ready()
.remove()
.removeAttr()
.removeClass()
.removeData()
jQuery.removeData()
.removeProp()
.replaceAll()
.replaceWith()
.resize()
.scroll()
.scrollLeft()
.scrollTop()
.select()
.serialize()
.serializeArray()
.show()
.siblings()
.size()
.slice()
.slideDown()
.slideToggle()
.slideUp()
.stop()
jQuery.submit()
.submit()
jQuery.support()
.text()
.toArray()
.toggle()
.toggle()
.toggleClass()
.trigger()
.triggerHandler()
jQuery.trim()
jQuery.type()
.unbind()
.undelegate()
jQuery.unique()
.unload()
.unwrap()
.val()
jQuery.when()
.width()
.wrap()
.wrapAll()
.wrapInner()

```

#1. jQuery库（续）

jQuery选择器

所有选择器 (*)	属性选择器 [name]
属性包含前缀选择器 [name^="value"]	:header选择器
属性包含选择器 [name*="value"]	:hidden选择器
属性包含字选择器 [name~="value"]	:ID选择器 (#id)
属性结束选择器 [name\$="value"]	:image选择器
属性相等选择器 [name=="value"]	:input选择器
属性不相等选择器 [name!="value"]	:last-child选择器
属性开始选择器 [name^=="value"]	:last选择器
:animated选择器	:lt()选择器
:button选择器	多属性选择器 [name=="value"] [name2=="value2"]
:checkbox选择器	:多选器 ("selector1, selector2, selector3")
:checked选择器	下一个相邻选择器 ("prev + next")
子选择器 ("parent > child")	下一个兄弟选择器 ("prev ~ siblings")
类选择器 (*.class*)	:not()选择器
:contains()选择器	:nth-child()选择器
后代选择器 ("ancestor descendant")	:odd选择器
:disabled选择器	:only-child选择器
元素选择器 (:element)	:parent选择器
:empty选择器	:password选择器
:enabled选择器	:radio选择器
:eq()选择器	:reset选择器
:even选择器	:selected选择器
:first选择器	:submit选择器
:first-child选择器	:text选择器
:first选择器	:visible选择器
:focus selector	
:gt()选择器	

#2. jQuery CDNs

CDNs（内容分发网络，或内容发布网络）是一个大规模的服务器网络，专门设计用来存储和分发信息，包括数据、软件、API代码、媒体文件或视频等等。从而可以很容易地在Web上访问这些信息。节点中的各个服务器包含所提供数据的一个副本。这些节点采用一定策略放置在一个网络上，如互联网，向使用数据的很多人发布信息时可以提高速度。Windows Azure 和Amazon CloudFront就是传统CDN的例子。

很多大企业都在CDN网络上提供了托管的jQuery副本，可供公开使用。以下是可以访问的一些CDN托管jQuery副本链接。

Google Ajax API CDN

- <http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js>

Microsoft CDN

- <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.6.2.min.js>

jQuery CDN (通过Media Temple)

- <http://code.jquery.com/jquery-1.6.2.min.js> (缩减版)

- <http://code.jquery.com/jquery-1.6.2.js> (源代码版本)

可以把这些包含在你的jQuery应用中，而不用每一次都下载jQuery。

#3. jQuery命名空间: noConflict方法

很多JavaScript库都使用\$作为函数或变量名, jQuery也是如此。对于jQuery, \$只是jQuery的一个别名, 所以即使不用\$也可以使用所有功能。如果除了jQuery还需要使用另外一个JavaScript库, 可以用一个\$.noConflict调用把\$的控制交还给另外那个库:

```
<script type="text/javascript" src="other_lib.js"></script>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    $.noConflict();
    //Code that uses other libraries $ can follow here.
</script>
```

结合.ready方法还可以对jQuery对象加别名, 这个技术特别有用, 因为如果希望以后不用担心出现冲突, 就可以在传递到.ready的回调中使用\$:

```
<script type="text/javascript" src="other_lib.js"></script>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    $.noConflict();
    jQuery(document).ready(function($) {
        // Code that uses jQuery's $ can follow here.
    });
    // Code that uses other libraries $ can follow here.
</script>
```

只有当你计划使用其他JavaScript库, 而且这个库中使用了\$作为引用, 此时才需要这个技术。如果只是在你的页面上使用jQuery, 就没有必要使用这个技术。即使包含了多个插件, 也不需要这个技术。

#4. 调试jQuery代码

调试代码通常很有用—特别是如果你在完成一个大规模的项目，其中有很多种不同类型的对象，包含或API。很多情况下你需要知道发回的某个对象或变量的内容，但是不希望通过提醒来显示，或者想要确定如何得到对象的所有属性。

这就引入了一些调试插件。这些插件可以帮助你查看对象内部，确定它们的属性值何时改变，或者跟踪一段时间内变量的变化。还可以查看对象在应用中的变化，或者是否会无意地得到null值。调试JavaScript或jQuery代码错误时这会很有用。

我们在编写JavaScript和jQuery代码时发现有两个调试插件尤其有用，它们是Dump和VariableDebugger。

<http://plugins.jquery.com/project/Dump>(查看对象中包含什么)。

<http://plugins.jquery.com/project/VariableDebugger>（与前面类似，不过会在弹出窗口中显示信息）。

还有很多其他插件，而且以后还会有越来越多的调试插件以及对这些插件的改进。这些插件很有用，不过也许你有更喜欢的其他插件，要查看这些插件的信息，请访问jQuery插件网站(<http://plugins.jquery.com/>)，搜索“debug”。

当然，不论怎样都可以使用这本书中一直在用的浏览器工具。

#5. 高级动画：队列

jQuery中的队列主要用于动画，可以用它们实现你想完成的任何工作。队列就是使用jQuery.data逐元素存储的一个函数队列，有先进先出（First-In-First-Out, FIFO）的特性。可以通过调用.queue为队列增加一个函数，还可以使用.dequeue删除（调用）一个函数。

jQuery可能会为各个元素关联一个或多个函数队列。大多数应用中只使用一个队列（名为fx）。队列允许异步地对一个元素调用一个动作序列，而不会中止程序的执行。这方面的典型例子就在一个元素上调用多个动画。例如：

```
$('#my_element').slideUp().fadeIn();
```

执行这个语句时，元素会立即开始滑动动画，不过只有在滑动转换完成时才会调用fx队列上的淡入转换。

利用.queue方法可以直接管理这个函数队列。结合回调来调用.queue尤其有用：这使我们可以在队尾放置一个新函数。

这个特性类似于用一个动画方法提供一个回调函数，不过不需要在完成动画时指定回调。

```
$('#my_element').slideUp();
$('#my_element').queue(function() {
    alert('Animation complete.');
    $(this).dequeue();
});
```

这等价于：

```
$('#my_element').slideUp(function() {
    alert('Animation complete.');
});
```

需要注意，用.queue增加函数时，最后一定要调用.dequeue，这样队列中的下一个函数才有可能执行。

在jQuery 1.4中，所调用的函数会作为第一个参数传入另一个函数，调用时自动将下一个函数出队。保证队列前移。可以如下使用：

```
$("#test").queue(function(next) {
    // Do some stuff...
    next();
});
```

jQuery的默认队列是fx。.animate和所有默认调用它的函数都使用这个默认队列。

注意：如果你在使用一个定制队列，必须手动地将函数出队。它们不会像默认fx队列中那样自动出队！

#6. 表单验证

由于篇幅原因，还有一个非常重要的特性我们没有介绍。这就是在客户/浏览器端使用jQuery完成表单验证。第9章、第10章和第11章中，我们见过一些服务器端验证的内容：在数据插入到数据库之前使用PHP完成验证。这也非常重要，而且值得强烈推荐。对数据库执行非法的插入或选择语句可能最终会暴露你原本不想暴露的数据。

不过我们还是来谈谈客户端验证……

有很多jQuery插件专门用来完成表单验证。其中我们最喜欢的插件之一就叫做“验证”插件，这个名字很贴切。可以在这里找到这个插件：<http://docs.jquery.com/Plugins/validation>。

利用这个插件，可以为表单上的各个元素创建一系列规则来定制验证，对希望在表单中接收的数据做出约束。这包括最小或最大域长度、检查必要域、查看是否输入了一个合法的Email地址等等所有方面。下面是jQuery网站提供的一些例子：

指定一个name元素为必要元素，另外指定一个email元素也是必要的（使用快捷方式表示简单规则），而且要有一个合法的Email地址（使用一个对象字面量）。

```
$( ".selector" ).validate({
    rules: {
        // simple rule, converted to {required:true}
        name: "required",
        // compound rule
        email: {
            required: true,
            email: true
        }
    }
});
```

指定一个元素为必要，而且最小长度（minlength）为2，并指定定制消息。

```
$("#myinput").rules("add", {
    required: true,
    minlength: 2,
    messages: {
        required: "Required input",
        minlength: jQuery.format("Please, at least {0} characters are
necessary")
    }
});
```

#7. jQuery UI效果

jQuery UI Effects库提供了另外一些动画，这是常规jQuery库所没有的。这些动画可以分为3种不同功能：

① 颜色动画

颜色动画扩展了animate函数，使它还能对颜色完成动画。颜色动画在类变换特性中大量使用，可以对以下属性完成颜色动画：

```
backgroundColor  
borderBottomColor  
borderLeftColor  
borderRightColor  
borderTopColor  
color  
outlineColor
```

② 类变换

类变换扩展了基本类API，能够在两个不同的类之间实现动画。以下jQuery方法由jQuery UI修改为可以接受另外3个参数：speed, easing（可选）和callback。

`addClass(class)`

向各组匹配的元素增加指定的类。

`removeClass(class)`

从一组匹配的元素删除所有或指定的类。

`toggleClass(class)`

如果没有指定的类，就增加该类；如果存在这个类则将指定的类删除。

`switchClass(currentClass, newClass)`

允许采用可视的方式从一个类变换为另一个类。

③ 高级松弛

高级松弛包含在Effects核心库中，这是Robert Penner编写的松弛函数的一个jQuery移植，原来是用ActionScript为Flash编写。这些函数就是一系列数学公式，用来使对象的动画更平滑、更准确。以下是所有松弛函数列表：

`linear`

`easeInQuart`

`easeInExpo`

`easeInBack`

`swing`

`easeOutQuart`

`easeOutExpo`

`easeOutBack`

`jswing`

`easeInOutQuart`

`easeInOutExpo`

`easeInOutBack`

`easeInQuad`

`easeInQuint`

`easeInCirc`

`easeInBounce`

`easeOutQuad`

`easeOutQuint`

`easeOutCirc`

`easeOutBounce`

`easeInOutQuad`

`easeInOutQuint`

`easeInOutCirc`

`easeInOutBounce`

`easeInCubic`

`easeInSine`

`easeInElastic`

`easeInElastic`

`easeOutCubic`

`easeOutSine`

`easeOutElastic`

`easeInOutCubic`

`easeInOutSine`

`easeInOutElastic`

更多资源请访问稀酷客(www.ckook.com)

#8. 创建你自己的jQuery插件

用插件和方法扩展jQuery能得到非常强大的功能，而且通过将最巧妙的函数抽象到插件中，还可以节省你和同事们的大量开发时间。

我们不打算用太多文字介绍如何创建一个jQuery插件，这个工作最好留给jQuery专家。他们提供了一个信息含量很丰富、非常详实的教程，可以在这里找到：<http://docs.jquery.com/Plugins/Authoring>。

以下是一个简短的总结，开发你的下一个jQuery插件时一定要谨记这些原则：

- 一定要把你的插件包围在(function(\$){ // 插件放在这里 })(jQuery);中。
- 不要在插件函数的直接作用域中冗余地包含this关键字。
- 除非从插件返回一个内部值，否则一定要让插件函数返回this关键字，以保持可串链性。
- 无需大量参数，可以把插件设置传入一个对象字面量，在插件默认设置基础上扩展这个对象。
- jQuery.fn对象中每个插件不要有多于一个命名空间。
- 要为方法、事件和数据指定命名空间。
- jQuery.fn读作“jQuery effin”。

#9. 高级JavaScript：闭包

闭包（Closures）是JavaScript中非常复杂的一个主题，而且这个主题过于闭合，在这本书里放在哪里都不太合适。尽管这个主题并非特别重要，但我们认为还是需要对它有所了解，所以这里简要做个介绍。

实际上一旦掌握了核心概念，闭包并不难理解。不过，如果你读了一些过于详细的技术描述，可能反而很困惑。

首先，来看下面的几个定义：

- 闭包是对应函数的局部变量，函数返回后仍然存活。
- 只要看到函数中又出现function关键字，内部函数可以访问外部函数中的所有变量。

是不是让人莫名其妙？

闭包完全依赖于变量和对象的作用域。作用域是指对象、变量和函数在哪里创建，可以在哪里访问，以及可以在哪个上下文中调用。基本说来，对象、变量和函数可以在局部或全局作用域中定义。

局部作用域：局部作用域是指，对象（变量和函数）仅在代码的某个部分中定义和访问，如一个函数内部。

全局作用域：与局部作用域不同，这是指全局的对象（变量和函数），可以从代码中任何位置访问。

考虑以下代码：

```
function func1(x) {  
    var tmp = 3;  
    function func2(y) {  
        alert(x + y + (++tmp));  
    }  
    func2(10);  
}  
func1(2);
```

#9. 高级JavaScript：闭包（续）

tmp变量在一个局部作用域中声明，即func1函数中。这总会提示16，因为func2可以访问x（它定义为func1的一个参数），还可以从func1访问tmp。

这不是一个闭包。闭包是指返回内部函数。内部函数在离开前会对func1的变量闭合。

现在来考虑以下代码：

```
function func1(x) {
    var tmp = 3;
    return function (y) {
        alert(x + y + (++tmp));
    }
}
var func2 = func1(2); // 现在func2是一个闭包。
func2(10);
```

同样的，tmp在局部作用域声明，不过func2函数在全局作用域中。上面的函数也会提示16，因为func2仍能引用x和tmp，尽管它不再直接位于这个局部作用域内。

不过，由于tmp仍存在于func2的闭包中，所以仍会递增。每次调用func2时它都会递增。

通过返回一个闭包函数列表，或者将它们设置为全局变量，可以创建多个闭包函数。所有这些都会引用相同的x和相同的tmp，它们不会建立自己的副本。

#10. 模板

jQuery模板仍为beta版，不过这是一个将要推出的很棒的特性，可以帮助你构建更为灵活的网站，而不需要太多HTML或jQuery。模板设计用来获取数据，将数据绑定到某个模板标记，从而能使用相同的标记一致地显示以类似方式关联的数据。

可以参考：<http://api.jquery.com/category/plugins/templates/>。

更多资源请访问稀酷客(www.ckook.com)

附录II：建立开发环境



你需要一个环境来实践刚刚掌握的PHP技能，但又不希望影响Web上的实际数据。最好能有一个安全的地方开发你的PHP应用，然后再（在Web上）对外发布。这是一个很好的想法。本附录包含了安装Web服务器、MySQL和PHP的操作说明，能提供一个安全的场所让你开展工作、着手实践。

更多资源请访问稀酷客(www.ckook.com)

创建一个PHP开发环境

利用刚刚掌握的jQuery和Ajax技能完成应用后，要把它放在Web上，不过在此之前，首先需要开发这个应用。在所有人都能看到的Web上直接开发Web应用可不是一个好主意。你可以在本地安装软件，应用上线之前先在本地进行构建和测试。

要构建和测试PHP和MySQL应用，需要在你的本地计算机上安装3个软件：

1. Web服务器
2. PHP
3. MySQL数据库服务器

PHP不是一个服务器，这是Web服务器了解的一组规则，使它能解释PHP代码。Web服务器和MySQL服务器都是计算机上运行的可执行程序。

我们现在讨论如何把你的本地计算机构建成一个Web服务器来完成PHP开发。不过要记住，最终还是需要一个在线Web服务器，要把你完成的应用上传到这个在线Web服务器，以便其他人访问和使用。



看看你有些什么

尝试安装PHP开发环境所需的各个软件之前，最好先检查一下你已经安装了什么。下面来查看这3个软件，了解如何确定系统上已经安装了哪些软件。

在已安装的软件方面，不同的本地计算机平台可能有很大差别。例如，Mac OS X往往已经默认安装一个Web服务器，但大多数Windows计算机没有默认安装的Web服务器。

说明：这个附录涵盖**Windows XP, Vista, Windows 7**和**Windows Server 2003/2008**，对于**Mac**，本附录适用于**Mac OS X 10.3.x或更新版本**。

更多资源请访问稀酷客(www.ckook.com)

有Web服务器吗？

如果你使用的是一个较新的PC或Mac，可能已经安装有一个Web服务器。在这些系统上，想要快速查出是否安装有Web服务器，可以打开一个浏览器窗口，在地址栏中键入`http://localhost`。如果看到一个介绍页面，这就说明你的本地机器上已经安装和运行着Web服务器。



如果你有一个安装了
Apache Web服务器的
Mac或Windows机器，
可能会看到这样结果。



如果你有一个安装了IIS的
Windows机器，可能会看
到类似这样的页面。



有PHP吗？哪个版本？

如果已经有Web服务器，可以很容易地查看是否安装有PHP，还能知道安装了哪个版本。创建一个名为info.php的新脚本，在其中键入下面的语句：

```
<?php phpinfo(); ?>
```

把这个文件保存到你的Web服务器使用的目录中。在Windows上，这通常是：

C:\inetpub\wwwroot\ (IIS)

或者：

C:\Program files (x86)\Apache Software Foundation\Apache2.2\htdocs (Apache)

在Mac上，往往是：

/Users/yourname/sites/

在浏览器中打开这个文件，键入`http://localhost/info.php`，如果安装了PHP，你会看到类似这样的页面：

这里是所安装的
PHP版本。



有MySQL吗？哪个版本？

在Windows上，右键点击Windows任务栏，选择任务管理器，再选择服务标签页，就可以确定是否安装有MySQL。要了解更多信息，在Windows 7中可以点击“services”（服务）按钮。



在这里可以看到
MySQL。

在Mac上要确定是否安装有MySQL，可以打开终端，并键入：

```
cd /usr/local/mysql
```

MySQL终端也称为
MySQL“控制台”。

如果这个命令成功执行，说明安装有MySQL。要查看版本，可以键入：

```
mysql
```

如果这个命令
成功执行，说
明已经安装
MySQL。

```
File Edit Window Help HelpPHP
$ cd /usr/local/mysql
$ mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.51b MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

这是已安装的
MySQL版本。

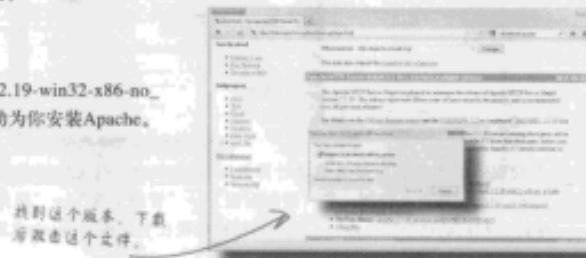
开始安装Web服务器

取决于你的Windows版本，可以下载Microsoft的Internet Information Server (IIS)，也可以下载开源的Apache Web服务器。如果你在使用Mac，需要一个服务器时往往会选择Apache，因为Mac上已经安装了Apache。

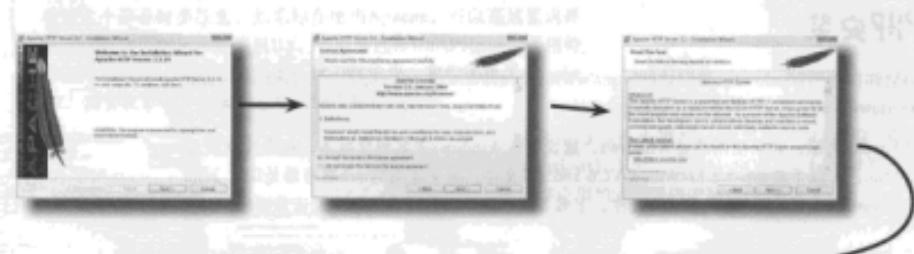
下面将简要介绍如何在Windows上安装Apache：

访问`http://httpd.apache.org/download.cgi`。

如果你在使用Windows，建议下载`apache_2.2.19-win32-no-ssl.msi`文件。下载并双击这个文件后，会自动为你安装Apache。



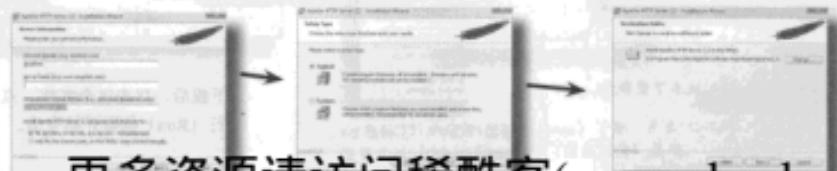
接下来你会看到安装向导。大多数操作说明都
很简单，可以直接接受默认选择。



选择你的计算机所在的域。如果没有，可以输入`localhost`。

最好选择典型安装选项。

通常可以选择安装软件的默认目录。



更多资源请访问稀酷客(www.ckook.com)

Apache安装……完毕

基本上完成了。点击安装（Install），等一分钟左右，安装就完成了。就这么简单！



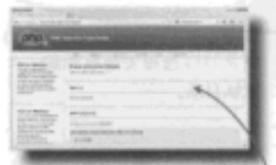
你的Web服务器会设置为启动计算机时自动启动。不过也可以使用服务（Services）面板来控制，在“控制面板”→“管理工具”→“服务对话框”中停止和启动这个服务器，这个对话框中现在它显示为Apache2.2。

如果这些说明对你不适用，可以再试一次，或者在你喜欢的搜索引擎中键入“Windows上安装Apache”（Installing Apache on Windows）寻求帮助。

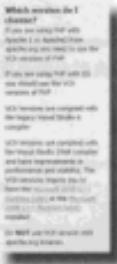
PHP安装

如果你在使用Windows，可以访问<http://www.php.net/downloads.php>或<http://windows.php.net/download/>。

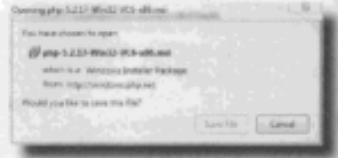
与Apache类似，如果使用的是Windows，建议下载Windows安装程序版本。如果你在使用Apache，可以下载php-5.2.17-Win32-VC6-x86.msi文件。如果使用的web服务器是IIS，可以下载php-5.3.6-Win32-VC9-x86.msi文件。下载并双击这个文件后会为你自动安装PHP。



这是.msi Windows版本下载部分。



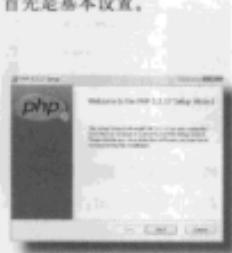
阅读操作说明，了解应当
下载哪个版本。



下载后，双击这个文件。点击运行（Run）按钮开始安装。

PHP安装步骤

首先是基本设置。

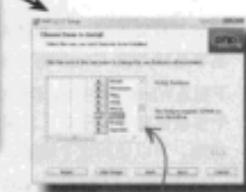
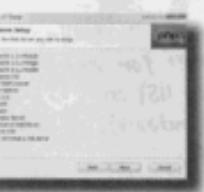


接受许可条款，继续。

通常最好选择默认安装文件夹，
不过这取决于你的个人喜好。
在这里我们选择了C:\PHP。



看到这个屏幕时要注意。如果你在使用Apache，可以在这里选择适当的版本。如果你在使用IIS，可能要选择IIS API模块。要用特定软件来检查，确定究竟需要什么。在这里，我们选择了Apache 2.2，需要在下一个屏幕指定Apache安装的路径。



下一个屏幕也有点麻烦。需要向下滚动到扩展包(Extensions)下面，选择MySQL。这样你就能使用这本书中用到的内置PHP MySQL函数了！

向下滚动到扩展包(Extensions)下面，
注意点击“Enable feature”(启用特性)选项。

更多资源请访问稀酷客(www.ckook.com)

PHP安装步骤……完毕

搞定了。点击Install（安装），然后点击Done（完成），结束安装程序。



现在请创建一个名为info.php的新脚本（如果之前还没有创建这个脚本）。在其中键入下面的语句：

```
<?php phpinfo(); ?>
```

把这个文件保存到你的Web服务器使用的目录中。在Windows上，这通常是：

C:\inetpub\wwwroot\ (IIS)

或者：

C:\Program files (x86)\Apache Software Foundation\Apache2.2\htdocs (Apache)

在Mac上，往往是：

/Users/yourname/sites/

要在浏览器中打开这个文件，可以键入

http://localhost/info.php。

如果已经安装了PHP，会看到类似这样的页面。

安装MySQL

安装说明和排错

还需要有MySQL，下面就来下载和安装MySQL。目前，MySQL RDBMS服务器免费版本的官方名字是MySQL Community Server。

下面是Windows和Mac OS X上安装MySQL的步骤清单。这些并不能替代MySQL网站提供的安装说明（那个安装说明更为完备），强烈建议你访问MySQL网站，阅读有关文档！要了解更详细的安装说明和排错指南，可以访问这个地址：

<http://dev.mysql.com/doc/refman/5.5/en/windows-installation.html>

你可能还希望有一个MySQL查询浏览器，可以在其中键入你的查询，通过软件界面查看结果，而不是在一个控制台窗口中。

更多资源请访问稀酷客(www.ckook.com)

如果这些操作说明
对你不适用，可以
再试一次，或者在
你喜欢的搜索引擎
中键入“Windows
上为Apache[或IIS]
安装”

PHP”(Installing
PHP for Apache
[or IIS] on
Windows)，寻求帮
助。



在Windows上安装MySQL的步骤

①

访问：

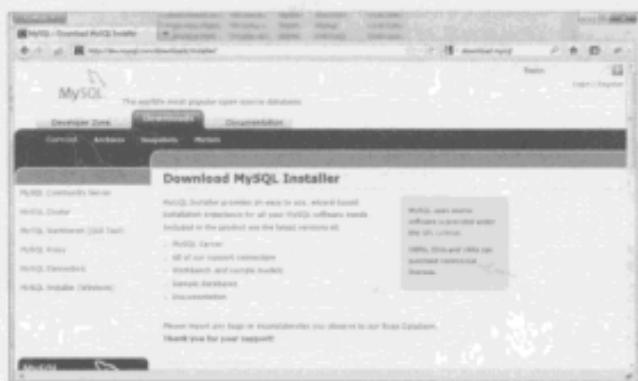
<http://dev.mysql.com/downloads/>

点击面向Windows的MySQL安装程序“Download the Beta”下载按钮（注意：写本书时，还是“Beta”版本）。



②

从列表中选择Microsoft Windows。



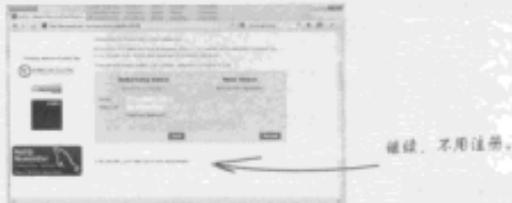
更多资源请访问稀酷客(www.ckook.com)

下载安装程序

- ③ 从列表中选择“Windows(x86, 32-bit), MSI Installer”。

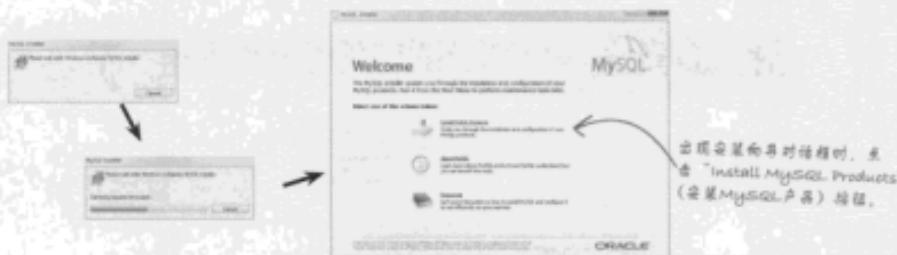


点击“No thanks, just take me to the downloads!”（不用了，谢谢，我要直接下载！），除非你想在网站上注册一个帐户，或者你已经有一个帐户。



- ④ 你会看到一个位置列表，这些位置分别提供了一个副本可供下载；选择离你最近的位置。

- ⑤ 文件下载结束后，如果启用了Windows UAC，右键点击这个文件，选择“Run as Administrator”（作为管理员运行）开始运行。现在就要跟随安装向导一步一步完成安装了。点击Next（下一步）。



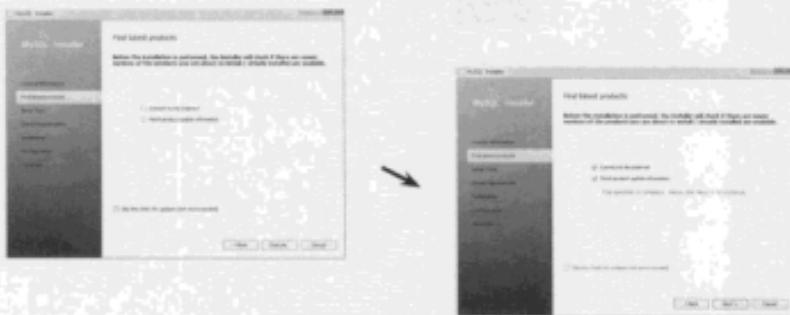
更多资源请访问稀酷客(www.ckook.com)

选择目标文件夹

- 6 阅读并接受许可条款，点击Next（下一步）。



- 7 下一步会运行自动更新，确保没有更新的版本。可以选择Skip（跳过）复选框跳过这一步，不过最好确保应用是最新的。完成更新之后，点击Next（下一步）继续。



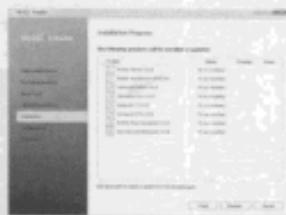
- ⑧ 接下来要求你选择一个安装类型。对你来说，Developer Default就很理想。另外，保留已经设置的默认安装路径，点击Next（下一步）。



- ⑨ 接下来，安装程序会检查系统与Microsoft .NET Framework 4 Client Profile的兼容性。这是为了运行MySQL Workbench应用。如果不兼容，可以在<http://update.microsoft.com>更新你的Windows实例。



- ⑩ 下一个屏幕会列出要安装的所有特性。点击执行（Execute）开始安装。

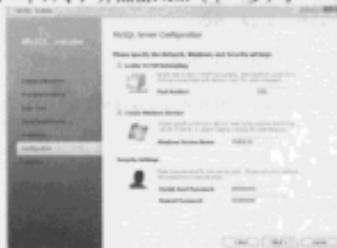


更多资源请访问稀酷客(www.ckook.com)

- ⑪ 所有服务都显示成功安装后，点击Next（下一步），访问MySQL服务的安装选项。选择Developer Machine（开发人员机器），并点击Next（下一步）。



- ⑫ 确保Enable TCP/IP Networking（启用TCP/IP网络）和Create Windows Service（创建Windows服务）选项都已经选中，保留原来的默认值。在最下面的方框中为根MySQL用户输入一个口令，并点击Next（下一步）。



- ⑬ 现在安装就完成了。如果没有自动启动，从“Start（开始）”→“All Programs（所有程序）”→“MySQL”菜单打开MySQL Workbench。



在Mac OS X上启用PHP

Mac OS X 10.5+ (Leopard)版本已经包含有PHP，不过默认情况下没有启用。必须访问主Apache配置文件，取消其中一行代码的注释才能启用PHP。这个文件名为http.conf，它藏在Apache安装文件夹中。

找到下面这行代码，它前面有一个井字符(#)，这使它作为一个注释：

```
#LoadModule php5_module      libexec/apache2/libphp5.so
```

要把这个井字符去掉，重启服务器来启用PHP。http.conf文档由“root”所有，这说明必须输入命令才能修改这个文档。你可能还想调整php.ini文件，以便Apache使用。关于如何完成这些步骤和启用PHP的更多详细信息，请访问http://foundationphp.com/tutorials/php_leopard.php。

在Mac OS X上安装MySQL的步骤

如果运行Mac OS X服务器，应该已经安装了一个MySQL版本。

开始之前，查看是否已经安装有一个版本。可以查看Applications/Server/MySQL Manager来确定。

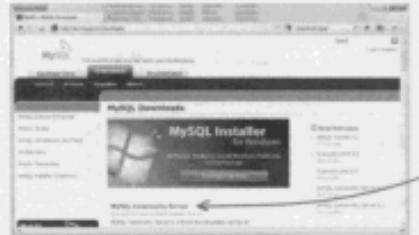


访问：

<http://dev.mysql.com/downloads/>

点击MySQL Community Server链接。

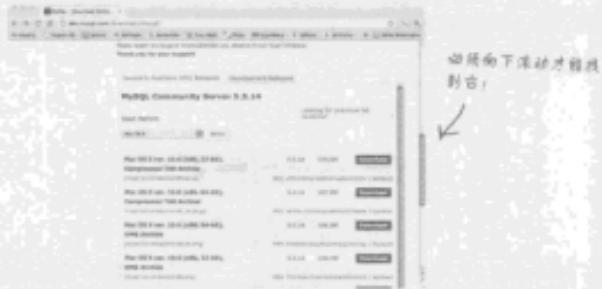
这里的安装说明主要针对32位版本的下载。要确保下载适用于你的操作系统版本。



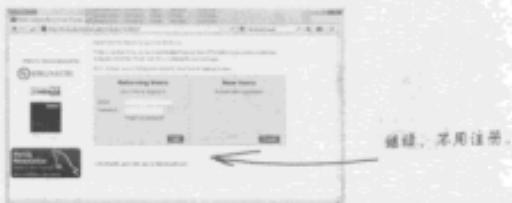
可能得向下滑动一点。

更多资源请访问稀酷客(www.ckook.com)

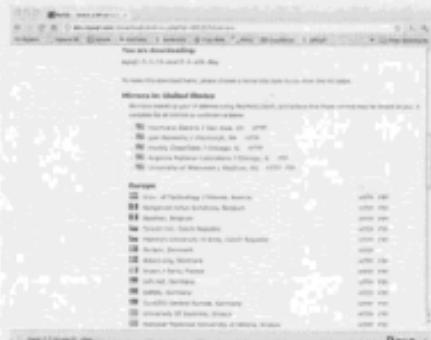
- ② 点击“Mac OS X v10.6 (x86, 32-bit), DMG Archive” download（下载）按钮。



- ③ 点击“No thanks, just take me to the downloads!”（不用了，谢谢，我要直接下载！），除非你想在网站上注册一个帐户，或者你已经有一个帐户。



点击离你最近的镜像，这样能更快地下载。



更多资源请访问稀酷客(www.ckook.com)

下载安装程序

④

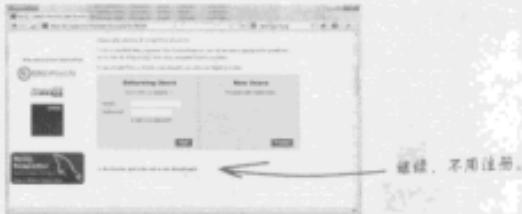
返回到：

<http://dev.mysql.com/downloads/>

点击MySQL Workbench (GUI工具)。



点击“*No thanks, just take me to the downloads!*”（不用了，谢谢，我要直接下载！），除非你想在网站上注册一个帐户，或者你已经有一个帐户，然后再选择一个镜像。



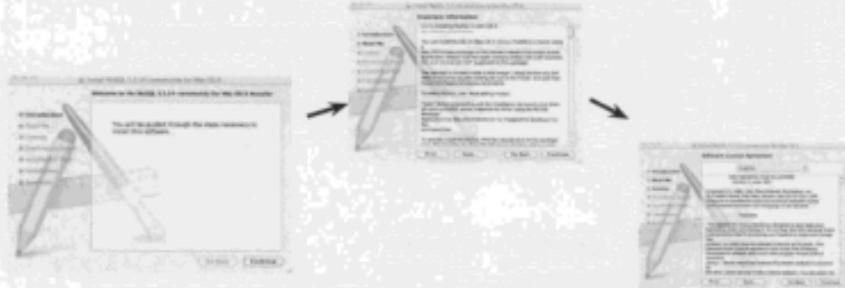
⑤

两个文件都完成下载后，双击mysql-5.5.14-osx10.6-x86.dmg文件装载这个安装程序，然后双击mysql-5.5.14-osx10.6-x86.pkg文件开始运行包安装程序。

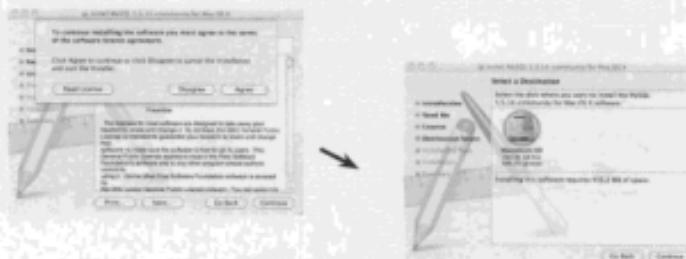


运行包安装程序

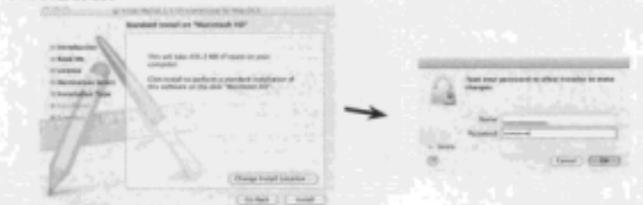
- 6 现在包安装程序会启动。点击Next（下一步），继续前进到Read Me（重要说明）页面。点击Continue（继续）进入许可页面。



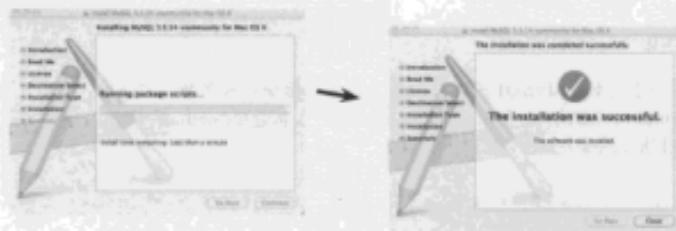
- 7 下一步会显示MySQL的许可信息。如果你接受这些条款，点击Continue（继续），然后点击Agree（同意）。再选择Continue（继续）把它安装在默认位置上。



- ⑧ 点击Install (安装)，输入管理员用户的用户名和口令，按OK开始安装。

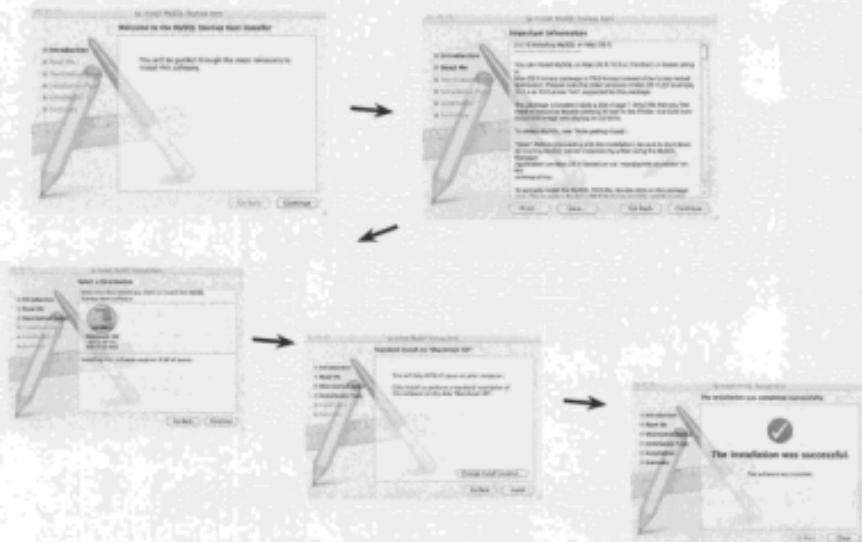


然后会开始安装，完成时将给出一个成功消息。



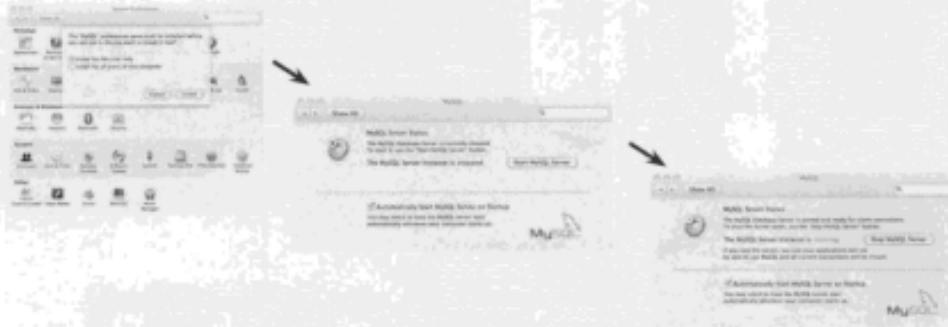
- ⑨ 对MySQLStartupItem.pkg文件重复同样的步骤。





- 10 双击MySQL.prefPane（也在mysql-5.5.14-osx10.6-x86.dmg中），安装MySQL Preference pane（首选项面板），然后，点击Start MySQL Server（启动MySQL服务器）。





- ⑪ 双击下载的mysql-workbench-gpl-5.2.34-osx-i686.dmg文件，启动MySQL Workbench工具的安装程序。



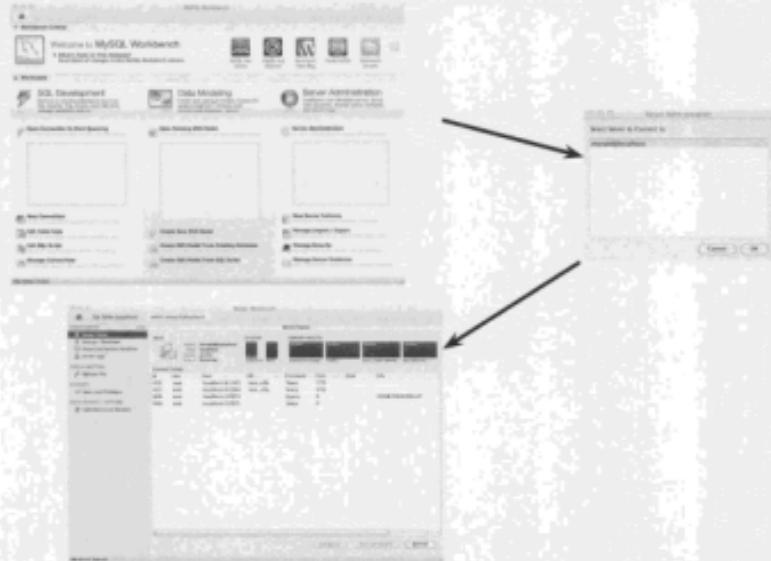
To install the applications drag the MySQL Workbench icon to your Applications folder.
You may then "Eject" and throw away this disk image.

把MySQLWorkbench.app拖到Applications文件夹中。

从Application列表打开Workbench工具。

更多资源请访问稀酷客(www.ckook.com)

- ⑭ 使用服务器管理面板 (Server Administration panel)，确保服务器正在运行。如果这个面板为空，点击New Server Instance (新建服务器实例) 选项，并选择运行的服务器。



- ⑮ 在SQL Development (SQL开发) 面板中创建一个新连接，点击New Connection (新建连接)，完成弹出的屏幕。然后，双击新建的连接将它打开。



有关MySQL或MySQL Workbench的更多帮助，请访问<http://dev.mysql.com/doc/>。

更多资源请访问稀酷客(www.ckook.com)

更多资源请访问稀酷客(www.ckook.com)

索引

符号

\$contains() method, \$contains()方法 114
\$ (dollar sign), \$ (美元符)
 and array names, 数组名 151
 for PHP variables, PHP变量 352
 for jQuery function shortcut, jQuery函数快捷方式 12, 19, 33
 for variable storing elements, 变量存储元素 150
\$_GET [associative array, \$_GET] 关联数组 331
\$_POST [associative array, \$_POST] 关联数组 331
\$_POST object, \$_POST 对象 342
\$_POST variable, \$_POST 变量 369
\$(this) selector, \$(this) 选择器 63–64, 74, 88
&& (and) operator, && (与) 操作符 240
{ } (curly braces), { } (大括号)
 for code block, 代码块 25, 42
 for function block, 函数模块 101
 for loops, 循环 230
. (dot) operator, . (点) 操作符 221. 参见 . (period)
== (equality operator), == (相等性操作符) 109, 240
 and if statement, if语句 239
= (equals sign), for setting variable value, = (等号), 用于设置变量值 56, 105, 207
===(exact equality operator), == (完全相等性操作符) 240
> (greater than) operator, > (大于) 操作符 240
>= (greater than or equal to) operator, >= (大于或等于) 操作符 240
(hash mark), # (井字符)
 for CSS id 13
 for id selector, id选择器 49
!= (inequality operator), != (不等) 操作符 240
<= (less than or equal to) operator, <= (小于或等于) 操作符 240
! (negation) operator, ! (否定) 操作符 240
-= operator, -= 操作符 207
*= operator, *= 操作符 207

/= operator, /= 操作符 207
+= operator, += 操作符 207
=> operator, => 操作符, PHP中 353
|| (or) operator, || (或) 操作符 240
() (parentheses), for click method, () (括号), click方法 42
. (period), . (点)
 to separate selector from method, 分隔方法与选择器 25
 to start CSS class, 开始CSS类 13, 48
<?php and ?> tags, <?php和?>标记 343, 352, 369
+ (plus), for concatenation, + (加), 用于链接 57
“ ” (quotation marks), “ ” (引号)
 for text or HTML value, 文本或HTML值 57
 for selectors, 选择器 15
; (semicolon), ; (分号)
 to end jQuery statement, 结束jQuery语句 16, 25
 to end PHP line, 结束PHP行 315, 352
[] (square brackets), [] (中括号)
 for array, 数组 225

A

absolute movement of elements, 元素的绝对移动 206
absolute positioning of elements, 元素的绝对定位 180
Accordion widget, Accordion部件 378
actions, 动作
 of forms, 表单 329
 repeating, 重复 229–230
active state, for anchor element, 活动状态, 锚元素 21
addClass() method, addClass()方法 117, 118, 172
addition operator (+=), 加法操作符 (+=) 207
advanced easing, 高级松驰 456
a element, 参见 anchor element (HTML)
after() method, after()方法 159–160
Ajax (Asynchronous JavaScript and XML), Ajax (异步JavaScript与XML) 291, 296~298, 324, 420

- components (组件) 298
for loading information, 加载信息 320
time for using, 使用时间 300
- ajax() method (jQuery), ajax()方法 (jQuery) 296, 304
getXMLRacers function call, getXMLRacers函数调用 307
parameters, 参数 304
testing, 测试 305
timeout setting in, 超时设置 312
- alert function, alert函数 42, 45, 50
for checking discount variable, 检查折扣变量 58
for click event, click事件 80
- Amazon CloudFront 431
- anchor element (HTML), 锚元素 (HTML)
nested image, 嵌套图像 22
states, 状态 21
- and (&&) operator, 与 (&&) 操作符 240
- animate element, animate元素, position property and, 属性 181
- animate() method, animate()方法 182, 208, 214
adding to script, 增加到脚本 209
do-it-yourself effects with, DIY效果 199
limitations, 限制 200
style changes over time, 样式随时间改变 202–204
syntax, 语法 201
with relative movement, 相对移动 208
- animation, 动画
color, 颜色 208, 456
of elements, 元素 188
method chains for combining, 方法链结合 193
option for turning off, 关闭选项 208
plug-ins for, 插件 376
queues for, 队列 454
- anonymous functions, 匿名函数 101, 102
- Apache web server, Apache web服务器 462
installing on Windows, Windows上安装 465–466
- APIs (application programming interfaces), API (应用编程接口) 411, 413, 445
Google Maps, 参见 Google Maps
sample code, 示例代码 414
object use by, 对象使用 415–416
- append() method, append()方法 59, 74, 237
- appendTo() method, appendTo()方法 237
- Apple's Safari 84
layout engine, 布局引擎 182
- apps, building without Flash, 应用, 不用Flash构建 176
- arguments, 参数 106, 114
- arithmetic operators, 算术操作符 207
- array_push() function (PHP), array_push()函数 (PHP) 354
- arrays, 数组 91, 151–152, 167, 174, 218, 252
accessing, 访问 226
adding and updating items, 增加和更新元素 227–228
blackjack data in, 21点游戏数据 225–228
creating, 创建 354
each() method for loop through, each()方法用于循环处理 168
finding elements in, 查找元素 172
JavaScript for emptying, JavaScript用于清空 246
PHP rules for, PHP规则 353
setting length to zero, 设置长度为0 246
storing elements in, 存储元素 150, 152
variable existence in, 变量存在 232
writing contents of, 写内容 353
- ASC keyword (SQL), ASC关键字 (SQL) 345
- assignment operator, = (equals sign) as, 赋值操作符, = (等号) 207
- associative arrays, 关联数组 331, 353
combining into single array, 结合为一个数组 354
converting to JSON-encoded string, 转换为JSON编码字符串 361
- asynchronous browsers, 异构浏览器 300
- Autocomplete widget, Autocomplete部件 378, 406

B

- background color, changing, 背景色, 改变 16
- background-color property (CSS), background-color属性 (CSS) 403
- before() method, before()方法 159–160
- binding events to element, 为元素绑定事件 81, 85
- bind() method, bind()方法 91
- Bit to Byte Race project, Bit to Byte Race项目
getXMLRacers function, getXMLRacers函数 307–308
repeated calls, 重复调用 312
testing, 测试 309
- initial appearance, 初始出现 293–295
- requirements, 需求 292, 326
- startAJAXcalls() function, startAJAXcalls()函数 311–312
- testing, 测试 305, 363

blackjack application, 21点应用 216–250
 adding excitement, 增加兴奋点 250–251
 array for card deck, 一副牌数组 227–228
 arrays for organizing data, 数组组织数据 225–228
 code, 代码 233–234
 end() function, end()函数 250
 function for event listeners, 函数作为事件监听者 243–244
 hand object, hand对象 237–238
 HTML and CSS code, HTML和CSS代码 236
 page setup, 页面设置 222
 reset feature for new game, 新游戏的重置特性 245–248
 rules, 规则 216, 238

Blind effect, Blind效果 378

block-level element, 块级元素 50

blur() method, blur()方法 256

body element (HTML), body元素 (HTML) 8

break command, for loops, break命令, 循环 237

browser cache, 浏览器缓存 96

browser layout engine, 浏览器布局引擎 182

browsers, 浏览器

- Ajax for passing data to, Ajax传递数据 298
- asynchronous, 异步 300
- events, 事件 82
 - listeners, 监听者 79
- JavaScript interpreter in, JavaScript解释器 5
- onblur and onfocus to respond to events, onblur和onfocus响应事件 259
- and PHP 341
- testing page in multiple, 多个浏览器中测试页面 29
- viewport, 视窗 7
- Web page display, Web页面显示 7
- and window object, window对象 256

browser wars, 浏览器之争 84

button elements, 按钮元素 16, 79

buttons, 按钮

- for interactive menu, 交互式菜单 133–134
- actions, 动作 135–138
- styling, 样式 386–389

buttonset() method, buttonset()方法 (jQuery UI) 386

button widget, button部件 386, 410

C

case sensitivity, 区分大小写

- of function names, 函数名 102
- of variable names, 变量名 60
- of XML tags, XML标记 327

CDNs (content delivery networks), CDN (内容分发网络) 451

chaining methods, 链接方法 142

checkForCode function, checkForCode函数 119–120
 testing, 测试 121

child of child element, selecting, 子元素的元素, 选择 67

children() method, children()方法 140, 141, 144
 filter methods and, 过滤器方法 162

child selector, 子选择器 67

Chrome Developer 71

classes, 类

- in CSS, CSS中 13, 48, 119
- for interactive menu, 交互式菜单 130
- maximum, 最大 60
- name restrictions, 命名限制 60
- and selectors, 选择器 52–54
- to specify div element for action, 为动作指定div元素 47

class selector, 类选择器 13, 14, 48
 combining with descendant selectors, 结合后代选择器 67

class transitions, 类变换 456

clearInputs() function, clearInputs()函数 333

clearInterval() method, clearInterval()方法 258, 263, 264, 310

clearOverlays() function, clearOverlays()函数 435

clearTimeout() method, clearTimeout()方法 258, 263, 310

clickable div element, 可点击div元素 24

clickable tabs, plug-in to convert ul element to, 可点击标签页, 插件来转换ul元素 293

click event, click事件 42, 79

- adding to page, 增加到页面 45
- handler function in, 处理函数 104
- limiting number, 限制数 184
- limiting user to one, 限制用户为一个 78
- for Monster Mashup, 怪兽变脸应用 182–184
- testing, 测试 88–89

click event listener, click事件监听者

- adding to getAllSightings() function, 增加到getAllSightings()函数 425–426
- for sending form data, 发送表单数据 333–334
- clickMe div element, style for, clickMe div元素, 样式 24

- click-related functions, combining, 与点击相关的函数, 结合 270–272
- client-side scripting, 客户端脚本 19
- client-side validation, 客户端验证 455
- clip property (CSS), clip属性 (CSS) 180
- closest method, closest方法 148
- closures, 包含 458–459
- code block, {} (curly braces) for, 代码块, {} (大括号) 25, 42
- color, 颜色
- animation, 动画 208, 456
 - jQuery to change background, jQuery改变背景 16
- color mixer, 颜色混合器
- refreshSwatch function, refreshSwatch函数 402–404
 - slider for, 滑动条 399–402, 406
- columns in SQL database, SQL数据库中的列 338
- comments, 注释 45
 - in CSS, CSS中 294
- comparing values, 比较值 244
- comparison operators, 比较操作符 240–242
 - in PHP, PHP中 353
- concatenation, 联接 57
 - testing, 测试 72
- conditional logic, 条件逻辑 108, 109–111, 122, 239, 280
 - for numbers list, 运动员列表 362
 - operators for decision process, 决策过程操作符 240–242
 - PHP rules for, PHP规则 353
 - for turning off scheduled events, 关闭调度事件 320–322
- contains method, contains方法 111
- Content Delivery Networks (CDNs), 内容分发网络 (CDN) 451
- convenience methods, 便利方法
 - for Ajax in jQuery, jQuery中的Ajax 320
 - for binding events, 绑定事件 81
- counter, for array processing, 计数器, 数组处理 168
- create database statement (SQL), create database语句 (SQL) 336
- CREATE TABLE statement (SQL), CREATE TABLE语句 (SQL) 336
- CREATE USER statement (SQL), CREATE USER语句 (SQL) 336
- Cryptid Sightings form, 参见 sightings form
- CSS (cascading style sheets), CSS (层叠样式表) 4
- background-color property, background-color属性 403 vs. DOM notation, DOM记法 208
 - file example for page tabs, 页面标签页文件示例 294 in jQuery UI, jQuery UI中 401
 - jQuery UI themes and, jQuery UI主题 379
 - limitations, 限制 3
 - link to jQuery UI file, 链接到jQuery UI文件 381
 - methods to change, 改变方法 115–116
 - properties, 属性 13, 188
- CSS class, CSS类 13, 48, 119
- CSS id 13
- CSS selectors, CSS选择器 13
 - in jQuery function, jQuery函数中 12
 - vs. jQuery selector, jQuery选择器 14, 51
- curly braces ({}), 大括号 ({})
- for code block, 代码块 25, 42
 - for function block, 函数模块 101
 - for loops, 循环 230
- current position, 当前位置
- movement relative to, 相对移动 280
 - setting, 设置 276
- current property, animate() impact and, current属性, animate()影响 205
- custom functions, 定制函数 100
 - optimizing, 优化 290
 - reset, 重置 283–285

D

- data, 数据 57
 - adding to Google Maps, 增加到Google Maps 423–426
 - Ajax for passing in structured format, Ajax用于传入结构化格式 298
 - arrays for storing, 数组用于存储 151–152
 - collecting from Web page, 从Web页面收集 328–330
 - formatting before sending to server, 发送到服务器之前格式化 332
 - getting from server, 从服务器获取 325
 - MySQL database for storing, MySQL数据库用于存储 335
 - objects for storing, 对象用于存储 218
 - PHP to access, PHP用于访问 341, 347–349
 - sanitization and validation in PHP, PHP中的清理和验证 364–366
 - sending to server, 发送到服务器 333–334
- database, 参见 MySQL database

- connecting with PHP, 用PHP连接 343
 inserting data with PHP, 用PHP插入数据 331
 database query, 数据库查询 359
 database servers, 数据库服务器 335
 data structures, 数据结构 218
 date() function (PHP), date()函数 (PHP) 315~316
 Date object (JavaScript), Date对象(JavaScript) 295
 datepicker, 日期选择工具 410
 customizing with options, 用选项定制 383
 for form, 表单 381
 testing, 测试 385
 datepicker() method, datepicker()方法 382
 db_connection() function (PHP), db_connection()函数 (PHP) 347~349, 357, 359
 deal() function, deal()函数 234
 debugging jQuery code, 调试jQuery代码 453
 decimal numbers, for slider widget, 小数, 滑动部件 397
 declaration for XML, XML声明 299
 declaring function, 声明函数 101, 103
 vs. named function expression, 命名函数表达式 102
 declaring variables, 声明变量 56, 186
 in loop, 循环中 230
 delay() method, delay()方法 263, 264, 310
 dequeue() method, dequeue()方法 454
 descendant selectors, 后代选择器 67
 detach() method, detach()方法
 testing for interactive menu, 测试交互式菜单 153
 vs. remove() 136~138
 developer, 开发人员
 jQuery version for, vs. production version, jQuery版本, 生产版本 19
 plug-in from, 插件 374
 Developer Default setup type for MySQL install, MySQL安装的开发人员默认设置类型 472
 development environment setup, 开发环境设置 xxxii~xxxiii, 461~482
 MySQL database, MySQL数据库
 checking for install, 检查安装 464
 installing on Mac OS X, Mac OS X上安装 474~482
 installing on Windows, Windows上安装 468~473
 PHP 462~463
 install, 安装 466~468
- Web server, Web服务器 465~466
 die command (PHP), 命令 (PHP) 343
 discount code, 折扣码
 custom function to check for, 定制函数检查 112
 testing, 测试 121
 discount variable, 折扣变量
 testing concatenation, 测试联接 72
 testing display, 测试显示 61~63, 64
 display property (CSS), display属性 (CSS) 188
 of img element, img元素 22
 Distance API 437
 div element (HTML), div元素 (HTML) 8, 39~40, 178, 179, 181
 action for specific, 特定div元素的动作 47~49
 clickable, 可点击 24, 45, 182~184
 clickable tabs to display, 可点击标签页显示 293
 click event for, click事件 42
 for Google map, Google地图 416
 for images, 图像 40
 left property, left属性 206
 for radio button group, 单选组 386
 for tab content, 标签页内容 295
 divide operator (/), 除法操作符 (/) 207
 document loading, events for, 文件加载, 事件 82
 document object, document对象 284
 Document Object Model (DOM), 文档对象模型 (DOM) 7, 8
 creating elements and adding events, 创建事件和增加事件 81
 inserting HTML content into, 插入HTML内容 159~160
 jQuery and 9~11, 19
 manipulating, 管理 174
 and selected elements, 选择的元素 149
 traversal, 遍历 140~144, 174
 vs. CSS notation, CSS记法 208
 web menu structure, web菜单结构 126~129
 document property of window object, window对象的document属性 258
 document structure, HTML and, 文档结构, HTML 4
 dollar sign (\$), 美元符 (\$) and array names, 数组名 151
 for PHP variables, PHP变量 352
 for jQuery function shortcut, jQuery函数快捷方式 12, 15, 19, 33
 for variable storing elements, 变量存储元素 150

- DOM. 参见 Document Object Model (DOM)
 domain, for Apache install, 域。Apache安装 465
 DOM inspectors, DOM检查工具 71
 dot(.) notation, 点(.)记法
 for JSON object, JSON对象 350
 for getting properties, 获取属性 218
 dot(.) operator, 操作符 221. 参见 period(.)
 do...while loops, do...while循环 229, 234, 237
 syntax, 语法 230
 downloading, 下载
 IIS (Internet Information Server), IIS (Internet信息服务
 器) 465
 jQuery UI 379
 MySQL installer, MySQL安装程序 470
 for Mac 476
 script blocking parallel, 脚本阻塞并行 50
 Draggable widgets, Draggable部件 401
 drop-down list in HTML form, HTML表单中的下拉列表 328
 Droppable interaction, Droppable交互 378
 Droppable widget, Droppable部件 401
 Dump plug-in, Dump插件 453
 dynamic Web pages, 动态Web页面
 PHP to create, PHP来创建 314
 updates, 更新 296
- ## E
- each() method, each()方法 91, 92, 114, 168
 and find() method, find()方法 306
 looping through array elements with, 用来循环处理数组元
 素 362
 and this keyword, this关键字 172
- Earth API 437
- easing functions, 松弛函数 456
- echo command (PHP), echo命令 (PHP) 315, 342, 353, 369
- effects in jQuery UI, jQuery UI中效果 456
- effects plug-ins, 效果插件 376
- elements, 元素
 absolute vs. relative movement, 绝对与相对移动 206
 adding event listeners to, 增加事件监听者 80
 animating, 动画 188
 binding events to, 绑定事件 81, 85
 block-level, 块级 50
 emptying content, 清空内容 148
 id selector to match single, id选择器来匹配单个元素 49
 index for list items, 列表项索引 110
 length property to check for existence, length属性检查存在
 性 430
 loop through group of, 循环处理元素组 90
 order of adding or removing, 增加或删除元素顺序 68
 positioning, absolute or relative, 定位, 绝对或相对 180
 relationships, and DOM traversal, 关系, DOM遍历 141
 – 146
 relative movement of, 相对移动 207
 removing from page, 从页面删除 66
 replaceWith() method for changing, replaceWith()方法用于
 改变 154–158
 selecting all on page, 选择页面上的全部元素 60
 storing, 存储
 in array, 数组中 152, 167
 in variables, 变量中 149–150
 wrapping inside another element, 包裹在另一个元素中
 172
- element selector, 元素选择器 13, 14
- Elevation API 437
- else clause (PHP), else子句 (PHP) 353
- else if clause (PHP), else if子句 (PHP) 353
- else if statement, else if语句 239
- else statement, else语句 109, 239
- empty() function (PHP), empty()函数 (PHP) 364
- emptying array, JavaScript for, 清空数组, JavaScript 246
- empty method, empty方法 148, 246
- end() function, for blackjack, end()函数, 21点应用 250
- end parameter, of slice method, end参数, slice方法 172
- eq() method, eq()方法 161
- equality operator (==), 相等性操作符 (==) 109, 240
 and if statement, if语句 239
- equals sign (=), as assignment operator, 等号 (=), 赋值操作
 符 56, 105, 207
- error, 错误
 in XML, and broken page, XML, 破坏的页面 327
 server return of, 服务器返回 312
- event handlers, 事件处理器
- named functions as, 命名函数 103
- for slider widget, 滑动条部件 391

- event listeners, 事件监听者 79, 85, 87
 behind the scene, 在幕后 80
 function for, 函数 243–244
- events, 事件 79, 122
 binding to elements, 绑定到元素 81, 85
 categories of, 种类 83
 development history, 开发历史 84–85
 functions in, 函数 83
 removing, 删除 86
 scheduling, 调度 310
 triggering, 触发 82–83, 91
 turning off scheduled, 关闭计划的事件 320–322
- exact equality operator (==), 完全相等性操作符 (==) 240
- Explode effect, 爆炸效果 378
- eXtensible Markup Language. 参见 XML (eXtensible Markup Language)
- ## F
- fade effects, 淡入淡出效果 192, 214
- fadeIn() method, fadeIn()方法
 in method chain, 方法链中 193, 196
 parameters for, 参数 192
- fade methods, 淡入淡出方法 27–28, 33
- fadeOut() method, fadeOut()方法 189
 in method chain, 方法链中 193, 196
- fadeTo() method, fadeTo()方法 189
- fail function, fail函数 357, 365
- fields in HTML, hidden, HTML中的域, 隐藏 329, 364
- file, for jQuery code, 文件, jQuery码 93–95
- filter() method, filter()方法 162
- filter methods, 过滤器方法 161–164, 174
- find() method, find()方法 172, 306, 361
- Firebug for Firefox 71
 Net tab 305
- Firefox (Mozilla) 84, 265
- first() method, first()方法 161
- Flash, building app without, Flash, 不用来构建应用 176
- floor() method, floor()方法 71
- focus, 焦点
 change, and running functions, 改变, 运行函数 255
 window with, 有焦点的窗口 256
- focus() method, focus()方法 256
- folders, for jQuery library, 文件夹, jQuery库 93–95
- font properties, font属性 200
- footer of Web page, Web页面页脚 317
- foreach loop (PHP), foreach循环 (PHP) 352
- for……in loop, for……in循环 237
- for loop, for循环 229, 234
 in PHP, PHP中 352
 syntax, 语法 230
- formatting data, 格式化数据
 before sending to server, 发送到服务器之前 332
 PHP for 354
- form element (HTML), form元素 (HTML) 328–330
 HTTP GET or POST methods, HTTP GET或POST方法 331
 input buttons in, 输入按钮 386
 selecting input elements on, 选择输入元素 407
- form events, 表单事件 82
- forms, 表单 328–330
 Autocomplete suggestions for field, 域的自动完成建议 406
 distrust data in, 不信任数据 364
 planning, 计划 373–374
 validation, 验证 406, 455
- FROM keyword (SQL), FROM关键字 (SQL) 345
- function callback, slide event attached to, 函数回调, slide事件关联到 391
- function expression, 函数表达式 101, 103
- function keyword, function关键字 101, 194, 220, 224
 within another function, and variable access, 另一个函数中, 变量访问 458
- functions, 函数 79, 100–108, 122, 148
 anonymous, 匿名 101, 102
 call to itself, 调用自身 194
 for click event, click事件 42
 code line length, 代码行长度 308
 combining click-related结合与click相关的 270–272
 creating, 创建 101
 custom, 定制
 optimizing, 优化 290
 reset, 重置 283–285
 declaring, 声明 101, 103
 vs. named function expression, 命名函数表达式 102
- in events, 事件中 83
- inside each() method, each()方法中 168

named, as event handlers, 命名, 事件处理器 103
 passing variables to, 传递变量 106
 repeated calls, 重复调用 312
 return value for, 返回值 107, 114
 running, focus change and, 运行, 焦点改变 255
 self-referencing, 自引用 311–312
 syntax, 语法 101
 timed, for repeated action, 定时, 重复动作 194
 variable for return value, 对应返回值的变量 232
 fx queue, fx队列 454

G

Gecko layout engine, Gecko布局引擎 182
 Geocoding API 437
 getAllSightings() function, getAllSightings()函数 422, 423, 429
 – 430
 adding click event listener, 增加click事件监听者 425
 – 426
 getAllTypes() function, getAllTypes()函数 429–430
 GET data, from ajax() method, GET数据, ajax()方法 304
 getDBRacers() function, getDBRacers()函数 356–360, 358
 to read JSON object, 读取JSON对象 361–362
 getJSON() method, getJSON()方法 320, 351, 358, 369
 GET method, GET方法
 for Ajax 320
 for HTTP, to send data to server, 向服务器发送数据 331
 getRandom() function, getRandom()函数 120, 231, 234, 276,
 277, 280
 creating, 创建 108
 getScript() method for Ajax, Ajax的getScript()方法 320
 getSightingsByType() function, getSightingsByType()函数 432
 – 435, 439–440
 getSingleSighting() function, getSingleSighting()函数 425
 – 426
 “getters”, “获取方法” 276
 getTimeAjax() function, getTimeAjax()函数 317–318
 getTime() function, getTime()函数 293
 calling, 调用 295
 getXMLRacers function, getXMLRacers函数 307
 repeated calls, 重复调用 312
 testing, 测试 309
 global scope, 全局作用域 458
 goLightning() function, goLightning()函数 275
 highlighting, hovering and, 突出, 悬浮 115–116

Google Ajax API CDN 451
 Google Chrome 84, 265
 Developer tools, 开发工具 305
 and jQuery 148
 layout engine, 布局引擎 182
 Google Directions API 437
 Google Earth plug-in, Google Earth插件 437
 Google Maps
 adding data to, 增加数据 423–426
 documentation, 文档 418, 438
 event listeners, 事件监听者 438
 including in Web page, 包含在Web页面中 417–418
 overlays, 重叠 435
 points as markers, 点作为标记 424
 sample code, 示例代码 414
 Terms of Service, 服务条款 437
 testing, 测试 419
 graphics, 参见 images
 greater than operator (>), 大于操作符 (>) 240
 greater than or equal to operator (>=), 大于或等于操作符
 (>=) 240
 grouping button widgets, 成组按钮部件 386

H

handler functions, 处理函数 83
 hash mark (#), 井号 (#)
 for CSS id 13
 for id selector, id选择器 49
 head element (HTML), head元素 (HTML) 8
 height property (CSS), height属性 (CSS) 190, 214
 hidden HTML field, 隐藏HTML域 329, 364
 hidden value for overflow property, overflow属性的隐藏值 180
 hide effect, slowing down, 隐藏效果, 减慢 196
 hide() method, hide()方法 188
 hiding span element, conditional logic for, 隐藏span元素, 条件逻辑 109–110
 highlighting, hovering and, 突出, 悬浮 115–116
 history property of window object, window对象的history属性
 258
 hit() function, hit()函数 234
 hover event, adding, hover事件, 增加 117–118

- hover state, for anchor element, 悬浮状态, 锚元素 21
- HTML. 参见 Document Object Model (DOM)
- and document structure, 文档结构 4
 - for slider widget, 滑动部件 391
 - inserting content into DOM, 向DOM插入内容 159–160
 - limitations, 限制 3
 - mixing with data, 混合数据 57
 - PHP and 352
 - separating jQuery code from, 与jQuery代码分离 97–98
 - tags inside variables, 变量中的标记 57
 - XML and 299
- html element, html元素 8
- htmlspecialchars() function (PHP), htmlspecialchars()函数 (PHP) 364
- HTML string, in jQuery function, HTML串, jQuery函数中 12
- I**
- IDs
- in CSS, CSS中 13
 - and selectors, 选择器 52–54
 - to specify div element for action, 为动作指定div元素 47
 - id selector, id选择器 13, 14, 49
 - combining with descendant selectors, 结合后代选择器 67
 - idTabs plug-in, idTabs插件 293
 - if statement, if语句 109–111, 239
 - == (equality operator) and, == (相等性操作符) 239
 - ternary operator as alternative, 三元操作符作为替代 244
 - if statement (PHP), if语句(PHP) 353
- IIS (Internet Information Server), IIS (Internet信息服务器) 462
- downloading, 下载 465
- images, 图像
- basing display on game result, 显示基于游戏结果 250
 - div element for, div元素 40
 - invisible, 不可见 181
 - properties for building path to, 属性用于建立路径 234
- img element (HTML), img元素 (HTML) 8
- in anchor element, 锚元素中 22
 - display property, display属性 22
 - nesting, 嵌套 178
 - sliding into view, 滑动到视图 26
- inArray() method, inArray()方法 231, 232
- index of element, 元素索引 110, 114
- in array, 数组中 151, 168, 226
- and eq() method, eq()方法 161
 - in loop, 循环中 230
- index variable, for each function, 索引变量, each函数 172
- inequality operator (!=), 不相等操作符 (!=) 240
- infinite loops, 无限循环 255
- info.php script, info.php脚本 463
- initialization of loop, 循环初始化 229
- initialize() function, initialize()函数 417–418
- input elements (HTML), 输入元素 (HTML) 328
 - datepicker for, 日期选择工具 381
 - selecting on form, 在表单上选择 407
- inserting, 插入
- data into database, PHP for, 数据插入数据库, PHP 331
 - HTML content into DOM, HTML内容插入DOM 159–160
- insert statement (SQL), insert语句 (SQL) 338–340
- installing, 安装
- Apache on Windows 465–466
 - MySQL database server, MySQL数据库服务器
 - on Mac OS X, Mac OS X上 474–482
 - on Windows, Windows上 468–473
 - PHP 466–468
- instances of object, 对象实例 220
 - creating, 创建 221
- interaction plug-ins, 交互插件 376, 401
- interactive menu, 交互式菜单 124–138
 - buttons, 按钮 133–134
 - actions, 动作 135–138
 - “Restore Menu” button, “Restore Menu”按钮 160
 - restoring entries, 恢复输入 146–152, 163–170
 - testing, 测试 139
- interactivity, 交互性 2
- example, 示例 20–30
- Internet Explorer (Microsoft) 84, 265
 - layout engine, 布局引擎 182
- invisibility, CSS for, 不可见性, CSS 22, 181
- iteration 90. 参见 loops
- J**
- JavaScript 5
- and Ajax 298

emptying array, 清空数组 246
 for datepicker 381
 including files, 包含文件 295
 interpreter, 解释器
 and animation process, 动画过程 188, 202
 and DOM 190
 jQuery and 9, 19
 JSON and 355
 keywords, and variable names, 关键字, 变量名 60
 objects, 对象 252
 vs. PHP 355
 variables, 变量 56
 window object, window对象 284, 290
 for timed effects control, 定时效果控制 256–259

jQuery 4, 5
 debugging code, 调试代码 453
 and DOM 9–11
 hosted copies on CDNs, 在CDN上托管副本 451

jQuery (续)
 vs. JavaScript 19
 and JSON 351, 355
 methods, 方法 448–449
 production vs. developer versions, 生产和开发版本 19
 selectors, 选择器 13, 15, 33, 450
 vs. CSS selector, CSS选择器 14, 51
 ; (semicolon) to end statement, ; (分号) 结束语句 16, 25
 separating code from HTML, 代码与HTML分离 93–95,
 97–98
 for slider widget, 滑动条部件 391
 translation, 转换 16–18
 web resources on, web资源 84

jQuery CDN 451

jquery.com website, jquery.com网站 83

jQuery function, jQuery函数
 \$ (美元符) for shortcut, \$ (美元符) 作为快捷方式 12,
 15, 19, 33
 contents, 内容 12

jQuery library, jQuery库
 folder for, 文件夹 93–95
 including, 包含 40

jQuery namespace, noConflict() method, jQuery命名空间
 noConflict()方法 452

jQuery plug-ins, creating, jQuery插件, 创建 457

jQuery UI 371, 410
 behind the scenes, 在幕后 382
 CSS in 401

download page, 下载页面 379
 effects, 效果 456
 package contents, 包内容 380
 plug-ins, 插件 374, 376
 button widget, 按钮部件 386
 customizable options for widgets, 部件的可定制选项
 383–384
 slider widget, 滑动条部件 390–398
 testing effects, 测试效果 376
 themes, 主题 379
 creating, 创建 401

jQuery wrapper, jQuery包装器 12

js extension, js扩展名 96

JSON (JavaScript Object Notation), JSON (JavaScript对象记法) 224, 325, 350–351, 369

jQuery and 351, 355
 with SQL and PHP, SQL和PHP 420–422

json_encode() function (PHP), json_encode()函数 (PHP)
 354, 358, 361, 369, 420

K

keyboard events, 键盘事件 82

key/value pairs, 健/值对
 for HTTP GET 331
 in PHP, PHP中 353

keywords in JavaScript, and variable names, JavaScript中关键字, 变量名 60

KML (Keyhole Markup Language), KML (Keyhole标记语言) 300

L

languages, 语言
 markup, 标记 299
 server-side, 服务器端 19, 314

last() method, last()方法 161

latitude, slider for, 纬度, 滑动条 397–398

LatLang object (Google), LatLang对象(Google) 415, 418
 defining, 定义 424

left property, of div element, left属性, div元素 206

length of array, setting to zero, 数组长度, 设置为0 246

length property, length属性
 of array, 数组 225
 to check for element existence, 检查元素存在性 430

- Lendorf, Rasmus 355
 less than or equal to (\leq) operator, 小于或等于 (\leq) 操作符 240
 libraries, 库
 jQuery
 folder for, 文件夹 93–95
 including, 包含 40
 PHP 344
 li element (HTML), li元素 (HTML) 8
 assigning class to, 指定类 131–132
 replacing in unordered list, 无序列表中替换 155–156
 lightning effect, 闪电效果 187
 adding functions to script, 向脚本增加函数 197
 image fade effect, 图像淡入淡出效果 192
 testing, 测试 269
 troubleshooting, 排错 254–255
 link to jQuery UI CSS file, 链接到jQuery UI CSS文件 381
 lists, 列表
 of elements, index of, 元素, 索引 110
 unorder, 无序 128–129
 .live() method, .live()方法 91
 loading information, using Ajax for, 加载信息, 使用Ajax 320
 local files, for web pages, vs. web server, 本地文件, web页面, web服务器 341
 localhost 348, 463
 local scope, 局部作用域 458
 logical operators, 逻辑操作符 240–242
 longitude, slider for, 经度, 滑动条 397–398
 loops, 循环 229–230, 252
 to change multiple elements, 改变多个元素 9, 90
 declaring variables in, 声明变量 230
 each() method for, each()方法 168
 and array elements, 数组元素 362
 infinite, 无限 255
 in PHP, PHP中 352
 syntax, 语法 230
 types, 类型 237
- ## M
- Mac OS X
 default web server, 默认web服务器 462
 determining MySQL install status, 确定MySQL安装状态 464
 enabling PHP, 启用PHP 474
 installing MySQL on, 安装MySQL 474–482
 many-to-one substitution, 多对一替换 158
 map_canvas element, map_canvas元素 417–418
 map events, listening for, 地图事件, 监听 438
 map object, map对象
 creating instance, 创建实例 417–418
 properties and methods, 属性和方法 416
 mapOpts property, mapOpts属性 418
 MapQuest 418
 Marker object (Google), Marker对象 (Google) 424
 markup languages, 标记语言 299
 max option for slider widget, 滑动条部件的max选项 390
 menus, 参见 interactive menu
 messages, 消息
 append() to insert, append() 插入 59
 concatenation when creating, 创建时联接 57
 displaying for user, 为用户显示 55–60
 removing, 删除 66
 methods, 方法
 chaining, 串联 142
 for combining effects, 结合效果 193
 for CSS changes, CSS改变 115–116
 of objects, 对象 219
 static, 静态 114
 Microsoft CDN 451
 Microsoft IIS (Internet Information Server), Microsoft IIS (Internet信息服务器) 462
 downloading, 下载 465
 Microsoft Internet Explorer 84, 265
 layout engine, 布局引擎 182
 Microsoft, maps 418
 Microsoft .NET Framework 4 Client Profile 472
 milliseconds setting, for fade effect, 毫秒设置, 淡入淡出效果 192
 min option for slider widget, 滑动条部件的min选项 390
 mobile device, maps on, 移动设备, 地图 437
 Monster Mashup, 怪兽变脸
 click event, click事件 182–184
 do-it-yourself effects, DIY效果 199
 layout and positioning, 布局和定位 178–180
 lightning effect, 闪电效果 187

- testing, 测试 269
 troubleshooting, 排错 254~255
 project blueprint, 项目蓝图 177
 randomize feature request, 随机特性请求 274~277
 testing, 测试 278~280
 testing, 测试 273, 286
 motion effects, 动作效果
 with animate() method, animate()方法 199
 relative to current position, 相对于当前位置 280
 mouseenter event, mouseenter事件 117
 mouse events, mouse事件 82
 mouseleave event, mouseleave事件 117
 moveMe function, moveMe函数 271~272
 moving through DOM, 在DOM中移动 140~144
 Mozilla Firefox 84, 265
 multiply operator (*=), 乘法操作符 (*=) 207
 MySQL 369
 MySQL Community Server 468
 mysql_connect function (PHP), mysql_connect函数 (PHP) 343
 MySQL database, MySQL数据库
 for storing data, 存储数据 335
 managing, 管理 344
 MySQL database server, MySQL数据库服务器 462
 checking for install, 检查安装 464
 downloading installer, 下载安装程序 470
 installing, 安装
 destination folder for, 目标文件夹 471
 on Mac OS X, Mac OS X上 474
 on Windows, Windows上 468~473
 troubleshooting guide, 排错指南 468
 MySQL functions (PHP), MySQL 函数 (PHP) 467
 MySQL query browser, MySQL查询浏览器 468
 MySQL Workbench 336, 337, 472
 opening, 打开 473
 select statement to see resultset, 选择语句查看结果集 346
- N**
- named functions, 命名函数
 vs. declaring functions, 声明函数 102
 as event handlers, 事件处理程序 103
 name property of window object, window对象的name属性 258
 names, 命名
- O**
- \$ for array, 数组 151
 of classes, 类 60
 of functions, 函数 101, 102
 of variables, 变量 56, 60, 105
 namespace, 命名空间 457
 name/value pairs, for JSON, 名/值对, JSON 350
 negation (!) operator, 否(!)操作符 240
 negative numbers, for slider widget, 负数, 滑动部件 397
 nested elements, 嵌套元素 15
 nested image, in anchor element, 嵌套图像, 锚元素 22
 Netscape Navigator 84
 new keyword, new关键字 220
 for array creation, 数组创建 225
 new() method, new()方法 221
 next() method, next()方法 140, 141
 in method chain, 方法链中 142
 noConflict() method, noConflict()方法 452
 not equal to operator (!=), 不等于操作符 (!=) 240
 not() method, not()方法 162
 numbers, 数字
 in CSS property settings, animate() method, CSS属性设置, animate()方法和 200
 slider widget for entry control, 滑动部件用于输入控制 390~398

- testing, 测试 259~260
 OOXML (Office Open XML) 300
 opacity of HTML elements, HTML元素的不透明性 27
 opacity property (CSS), opacity属性 (CSS) 189, 214
 OpenLayers, mapping API 418
 optimizing custom functions, 优化定制函数 290
 option element (HTML), for drop-down list, 选项元素 (HTML), 下拉列表 328
 order by keyword (SQL), order by关键字 (SQL) 345
 orientation option for slider widget, 滑动部件的orientation属性 390
 or (||) operator, 或(||)操作符 240
 overflow property, overflow属性 180
 overlays in Google Maps, Google Maps中的重叠 435
- ## P
- package installer, for MySQL for Mac, 包安装程序, 向在Mac的MySQL 477~479
 page load time, speeding up, 页面加载时间, 加速 50
 parallel downloads, script blocking, 并行下载, 脚本阻塞 50
 parameters, 参数 106, 114, 197
 parentheses, for click method, 括号, click方法 42
 parent() method, parent()方法 140, 141, 145, 148
 in method chain, 方法链中 142
 parent selector, 父选择器 67
 parents() method, parents()方法 148
 filter methods and, 过滤器方法 162
 passing variables to function, 向函数传递变量 106
 Penners, Robert 456
 period (.), 点(.)
 to separate selector from method, 分隔方法与选择器 25
 to start CSS class, 开始CSS类 13, 48
 PHP (PHP: Hypertext Processor), PHP (PHP: 超文本处理器) 314, 324, 369
 to access data, 访问数据 341, 347~349
 data sanitization and validation, 数据清理和验证 364
 – 366
 development setup, 开发设置 462~463
 enabling on Mac OS X, Mac OS X上启用 474
 for connecting to database, 连接数据库 343
 for inserting data into database, 向数据库插入数据 331
 formatting output, 格式化输出 354
 installation, 安装 466~468
 checking for, 检查 463
 library selection, 库选择 343
 vs. JavaScript 355
 JSON with SQL and 420~422
 libraries, 库 344
 MySQL functions, MySQL函数 467
 rules, 规则 352~353
 PHPMyAdmin 344
 picframe div element, picframe div元素 24
 pictures, 参见 images
 Places API 437
 planning, 计划 178
 plug-ins, 插件 302, 376
 configuring, 配置 293
 creating, 创建 457
 debugging, 调试 453
 plus sign (+), for concatenation, 加号 (+), 联接 57
 pop-up window, for alert statement, 弹出窗口, alert语句 42
 position of elements, 元素位置
 absolute or relative, 绝对或相对 180
 setting current, 设置当前位置 276
 position property (CSS), position属性 (CSS) 182
 of animate element, animate元素 181
 POST data, POST数据
 Ajax() method for, Ajax()方法 304
 on Web server, Web服务器 342
 POST method, POST方法 333
 for HTTP, to send data to server, 向服务器发送数据 331
 in Ajax 320
 p (paragraph) element (HTML), p (段落)元素 (HTML) 8, 16
 changing HTML inside, 改变HTML 9
 click event for, click事件 42
 preg_match function (PHP), preg_match函数 (PHP) 364
 prev() method, prev()方法 140, 141
 print command (PHP), print命令 (PHP) 353
 print_r() command, print_r()命令 353
 production version of jQuery, vs. developer version, jQuery生产版本与开发版本 19
 Progressbar widget, Progressbar部件 378

project, 项目

- determining requirements, 确定需求 37~38
- planning, 计划 178
- structure, 结构 96

properties, 属性

- animate() method to change, animate()方法来改变 201
- assigning value, 赋值 224
- for building path to image建立图像路径 234
- of objects, 对象 218, 219
- accessing, 访问 221

Puff effect, Puff效果 378

Q

querying, 查询 15

queues for animation, 队列用于动画 454

quotation marks (" "), 引号 (" ")

- for text or HTML value, 文本或HTML值 57
- for selectors, 选择器 15

R

radio buttons, div element for group, 单选钮, div元素用于分组 386

random animation in Monster Mashup, 怪兽变脸应用中的随机动画 275~277

randomize() function, randomize()函数 277, 281~282, 285

random() method, random()方法 71

random number, 随机数

- testing concatenation into discount variable, 测试与折扣变量链接 72

- variable to hold, 变量用于保存 57~58

random number generator function, 随机数生成函数 112

RDBMS (Relational Database Management Systems), RDBMS (关系数据库管理系统) 335

ready() method, ready()方法 184~185, 452

referencing variables, 引用变量 57

refreshSwatch() function (JavaScript), for color
mixer, refreshSwatch()函数(JavaScript), 用于颜色混合器 402~404

regular expression matching function, 正则表达式匹配函数 364

relationships of elements, and DOM traversal, 元素关系, DOM 遍历 141~146

relative movement of elements, 元素相对移动 206, 207

to current position, 到当前位置 280

relative positioning of elements, 元素的相对定位 180

reloading, changing web pages without, 重新加载, 改变web
页面 4

removeClass method, removeClass方法 117

remove() method, remove()方法 74, 92

in method chain, 方法链中 142

for message, 消息 66

order of calls and, 调用顺序 68, 70

troubleshooting, 排错 71

vs. detach() 136~138

removing events, 删除事件 86

repeated actions, 重复动作 229~230

timed function for, 定时函数 194

repeated function calls, 重复函数调用 312

replaceWith() method, replaceWith()方法 154~158, 165~166
limitations, 限制 158

replacing li elements in unordered list, 无序列表中替换li元素
155~156

reset() function, reset()函数 283~285

Resig, John 19

Resizable widget, Resizable部件 378, 401

restoring interactive menu entries, 存储交互式菜单输入 146
– 152, 163~170

resultset for SELECT statement, SELECT语句结果集 345, 346

return keyword, return关键字 107

return value of function, 函数的返回值 114

variable for, 变量 232

reusable objects, 可重用对象

building, 构建 220, 223~224

vs. one-time, 一次性 224

root node in XML, XML中根节点 299

RSS (RDF Site Summary), RSS (RDF站点摘要) 300

RSS (Real Simple Syndication), RSS (真正简单合成) 300

running functions, focus change and, 运行函数, 焦点改变 255

S

Safari (Apple) 84

layout engine, 布局引擎 182

- same-origin policy, and Ajax calls, 同源策略, Ajax调用 305
 sanitizing data in PHP, 用PHP清理数据 364–366
 scalable object, Resizable widget for, 可扩展对象, Resizable 部件 401
 scale effects, with animate() method, 扩展效果, animate()方法 199
 scheduled events, 调度事件
 creating, 创建 310
 turning off, 关闭 320–322
 scope, global vs. local, 作用域, 全局与局部 458
 screen display, PHP for writing to, 屏幕显示, PHP用于写入 353
 script element (HTML), 脚本元素 (HTML) 4, 45
 for Google Maps API code, Google Maps API代码 417
 for jQuery UI 381
 for link to script file, 链接到脚本文件 93
 location of, 位置 50
 scripts, 脚本
 adding animate() functions, 增加animate()函数 209
 adding lightning effect functions to增加闪电效果函数 197
 and web page changes, web页面改变 5–6
 folder for, 文件夹 93
 Selectables widgets, Selectables部件 401
 selecting, 选择
 all elements on page, 页面上所有元素 60
 elements using traversal methods, 使用遍历方法选择元素 141–144
 filter methods to narrow, 过滤器方法缩小选择范围 161–164
 input elements on HTML form, HTML表单上的输入元素 407
 selector engine for jQuery, jQuery的选择器引擎 9
 selectors, 选择器 33, 74, 130, 148
 \$ (this) 63–64
 CSS vs. jQuery 51
 for datepicker 382
 descendant, 后代 67
 in jQuery 13, 450
 .(period) to separate from method, .(点) 分隔方法选择器 25
 select statement (SQL), select语句 (SQL) 345–346
 where clause, where子句 348
 <select> tag (HTML), <select>标记 (HTML) 328
 self-executing functions, 自执行函数 102
 self-referencing functions, 自引用函数 311–312
 semicolon (;)
 to end jQuery statement, 结束jQuery语句 16, 25
 to end PHP line, 结束PHP行 315, 352
 serializeArray() method, serializeArray()方法 332, 369
 serialize() method, serialize()方法 332
 servers, 参见 web server
 CDNs and 451
 database, 数据库 335
 error returned from, 返回的错误 312
 formatting data before sending to, 发送前格式化数据 332
 sending data to, 发送数据 329, 333–334
 server-side languages, 服务器端语言 19, 314
 setInterval() method, setInterval()方法 (JavaScript) 258, 263,
 264, 265, 290, 310
 browser variations, 浏览器验证 265
 setTimeout() method, setTimeout()方法 (JavaScript) 196, 198,
 255, 258, 263, 264, 290, 310, 312
 browser processing of, 浏览器处理 265
 showFrequency() method, showFrequency()方法 317–318
 show() method, show()方法 188
 siblings method, siblings方法 148
 sightings form, 观测表单 372–374
 code, 代码 384
 datepicker for 381
 getSightingsByType() function, getSightingsByType()函数 432–435
 Google map, Google地图 412–413
 multicreature checklist, 多生物列表 428
 project checklist, 项目任务列表 380
 testing, 测试 389, 405, 427, 431
 slice() method, slice()方法 162, 172
 slideDown() method, slideDown()方法 26, 190
 slide effects, 滑动效果 214
 and element height, 元素高度 190
 slider widget, 滑动条部件 410
 for color mixing, 颜色混合 399–402
 for controlling numerical entries, 控制数字输入 390–398
 for longitude and latitude, 经度和纬度 397–398
 negative numbers and decimal numbers, 负数和小数 397
 planning, 计划 373
 testing increments, 测试递增 396

slideToggle() method, slideToggle()方法 26, 28, 190
 slideUp() method, slideUp()方法 190
 for image, 图像 26, 52
 SOAP (Simple Object Access Protocol), SOAP (简单对象访问协议) 300
 Sortable widgets, Scatable部件 378, 401
 span element (HTML), span元素 (HTML) 317~318
 conditional logic to hide, 条件逻辑隐藏 109~110
 speed of fadeIn(), fadeIn速度 27
 SQL (Structured Query Language), SQL (结构化查询语言) 335, 369. 参见 MySQL
 for database, table and users setup, 数据库, 表和用户设置 336~337
 insert statement, insert语句 338~340
 JSON with PHP and 420~422
 SELECT to read data from, SELECT从中读取数据 345~346
 testing connection, 测试连接 344
 square brackets ([]), 中括号 ([])
 for array, 数组 225
 for array item index, 数组项索引 226
 in PHP function, PHP函数中 315
 standards for browsers, 浏览器标准 84
 startAJAXcalls() function, startAJAXcalls()函数 311~312, 322, 356, 358
 start parameter, of slice method, start参数, slice方法 172
 Static Maps API 437
 static methods, 静态方法 114
 static Web pages, 静态Web页面 3
 step option for slider widget, 滑动部件的step选项 390
 stopLightning() function, stopLightning()函数 266, 268
 stop method, stop方法 208
 storage space, creating, 存储空间, 创建 56
 storing elements in array, 数组中存储元素 152
 structure of project, 项目结构 96
 styles, 样式 23
 class or ID and, 类或ID 60
 submit() listener, submit()监听者 333
 substitution, 替换
 one-to-many or many-to-one, 一对多或多对一 158
 one-to-one, 一对一 165~166
 subtraction operator (-), 减法操作符 (-) 207

success function, success函数 357
 sumCardTotal method, sumCardTotal方法 242
 SVG (Scalable Vector Graphics), SVG (可扩展矢量图像) 300
 switch method, switch方法 244

T

tables in MySQL databases, MySQL数据库中的表 335
 tags in XML, XML中标记 299
 Task Manager, Services tab, 任务管理器, 服务标签页 464
 templates for jQuery, jQuery模板 459
 terminator, 终止符 148
 ternary operator, 三元操作符 244
 test condition in loop, 循环中测试条件 229
 testing, 测试
 Bit to Byte race project, Bit to Byte长跑比赛项目 305
 click event, click事件 46, 88~89
 draggable 385
 detach() method for interactive menu, 交互式菜单的 detach()方法 153
 discount variable display, 折扣变量显示 61~63, 64
 getXMLLRacers function, getXMLLRacers函数 309
 Google Maps API 419
 interactive menu, 交互式菜单 139
 lightning effect, 闪电效果 269
 Monster Mashup, 怪兽变脸应用 286
 onfocus and onblur event handlers, onfocus和onblur事件处理器 259~260
 random number concatenated into discount variable, 随机数链接到折扣变量中 72
 sightings form, 观测表单 389, 405
 slider widget increments, 滑动部件递增 396
 SQL database connection, SQL数据库连接 344
 web page in multiple browsers, 多个浏览器中web页面 29
 test box in HTML form, HTML表单中文本框 328
 test effects, 文本效果 200
 themes in jQuery UI, jQuery UI中主题 379, 380
 creating, 创建 401
 this keyword, this关键字 63, 168, 209
 each() method and, each()方法 172
 origin of use, 使用源 71
 plug-ins and, 插件 457

timed effects, window object to control, 定时效果, window对象控制 256~259
 timed functions, 定时函数 290
 for repeated action, 重复动作 194
 timeout, 超时
 interval, 间隔 255
 setting in ajax() call, ajax()调用中设置 312
 Timeout() method, Timeout()方法 194
 timer for animation, 动画定时器 202
 timer methods, 定时器方法 263~265, 310
 of window object, Timeout对象 256
 time zone, for PHP date function, 时区, PHP日期函数 315
 title element (HTML), title元素 (HTML) 8
 toggle() method, toggle()方法 188, 196
 transparency of HTML elements, HTML元素的透明性 27
 traversing DOM, 遍历DOM 140~144, 174
 Trident layout engine, Trident布局引擎 182
 triggering events, 触发事件 82~83, 91
 .trigger() method, trigger()方法 91
 troubleshooting lightning effect, 闪电效果排错 254~255
 turning off animation, option for, 关闭动画, 选项 208

U

ul element (HTML), ul元素 (HTML) 8
 plug-in to convert to clickable tabs, 插件用来转换为可点击标签页 293
 UML (Unified Modeling Language) diagram, UML (统一建模语言) 图 219, 223~224
 unbind command, unbind命令 86
 "Undefined Index" error, "未定义索引" 错误 226
 "undefined" value, function return of, "未定义" 值, 函数返回 114
 unnamed functions, 未命名函数 101
 unordered list 128~129. 参见 ul element (HTML)
 assigning class to items, 为元素指定类 131~132
 method chain to traverse, 遍历的方法链 142
 replacing li elements in, 其中替换ul元素 155~156
 user interface. 参见 forms; jQuery UI
 slider widget for controlling numerical entries, 滑动条部件控制数值输入 390~398

user requirements, determining, 用户需求, 确定 37~38
 utility function, 实用工具函数 276

V

validating, 验证
 data in PHP, PHP中数据 364~366
 forms, 表单 406, 455
 val() method, val()方法 391
 value, 值
 for object property, 对象属性 218
 option for slider widget, 滑动条部件选项 390
 VariableDebugger plug-in, VariableDebugger插件 453
 variables, 变量 56, 148
 in anonymous functions, 匿名函数中 102
 vs. arrays, 数组 218
 assigning value to, 赋值 207
 for clearInterval() method, clearInterval()方法 264
 counter for array processing, 计数器用于数组处理 168
 declaring, 声明 186
 in loop, 循环中 230
 for return value of function, 函数的返回值 232
 for storing elements, 存储元素 149~150
 function keyword within another function and, 另一个函数中的function关键字 458
 in array, 数组中 225, 232
 in functions, 函数中 105
 name restrictions, 命名限制 60
 passing to function, 传入函数 106
 in PHP, PHP中 352
 referencing, 引用 57
 var keyword, var关键字 56, 105, 219
 viewport of browser, 浏览器视窗 7
 visibility of img element, img元素可见性 22
 visual effects, example, 视觉效果, 示例 20~30
 visual properties of elements, 元素的视觉属性 188

W

W3C (World Wide Web Consortium) 19
 Webkit browser layout engine, Webkit浏览器布局引擎 182
 Web page power, Web页面能力 2
 Web pages, Web页面
 adding message to, 增加消息 60

broken from XML errors, 由XML错误中断 327
 browser display, 浏览器显示 7
 browser request for, 浏览器请求 3
 changing without reloading, 不重新加载就完成改变 4
 collecting data from, 从中收集数据 328~330
 delivery by web server, vs. local files, 由web服务器与本地文件发布 341
 dynamic, PHP to create, 动态, PHP来创建 314
 dynamic updates, 动态更新 296
 footer, 页脚 317
 hidden structure, 隐藏结构 8
 including Google Maps, 包含 Google Maps 417~418
 PHP to write to, PHP写入 315
 scheduling events, 调度事件 310
 turning off, 关闭 320~322
 scripts to change, 脚本来修改 5~6
 static, 静态 3
 tabs, 标签页
 adding, 增加 329~330
 CSS file example, CSS文件示例 294
 testing in multiple browsers, 多个浏览器中测试 29
 updating new data only, 只更新新数据 297
 Web resources on jQuery, jQuery上Web资源 84
 Web server, Web服务器
 determining existence of, 确定存在性 463
 for PHP scripts, PHP脚本 341
 for Web pages, vs. local files, Web页面与本地文件 341
 local computer as, 本地计算机 462
 POST data on, POST数据 342
 setup, 建立 465~466
 where clause in SQL SELECT statement, SQL SELECT语句中
 where子句 348
 while loop, while循环 237, 359
 in PHP, PHP中 352
 widget plug-ins, 部件插件 376, 410

customizable options, 可定制选项 383~384
 for interactions, 交互 401

Window, lost focus, and running functions, 窗口, 失去焦点, 运行函数 255

Window object (JavaScript), Window对象(JavaScript) 284, 290
 for timed effects control, 定时效果控制 256~259

Windows Azure 451

World Wide Web Consortium (W3C) 19

X

XHTML 300

XML (eXtensible Markup Language), XML (可扩展标记语言) 298, 299~300, 324

declaration, 声明 299

errors and broken pages, 错误和破坏的页面 327

file listing, 文件列表 305

vs. JSON 350

parsing data, 解析数据 306

server for file storage, 服务器用于文件存储 305

XMLHttpRequest object, XMLHttpRequest对象 312

Y

Yahoo! 418

Z

zero-indexed arrays, 从0开始索引的数组 226

zero, setting array length to, 0, 设置数组长度为0 246