

jQuery UI



jQuery UI

开发指南



REILLY®

[美] Eric Sarrion 著
罗晴明 包勇明 译

人民邮电出版社
POSTS & TELECOM PRESS

jQuery UI开发指南

使用jQuery UI库，可以将jQuery的强大功能和统一标准，连同交互元素、动画和主题小组件，应用到用户界面设计中。这本简明的、以代码为主入门指南，展示了如何利用HTML5中缺乏的一些交互功能，其中包括选项卡、折叠菜单和对话框等。通过阅读本书，你还将学会如何为普通但复杂的任务（比如管理拖放和自动补全）编写代码，使用户与网站之间的交互更加容易。

这是一本快速指南，采用每章详细地关注一个组件的方式，介绍了jQuery UI如何改进HTML页面。如果你是一位Web开发人员或者设计师，期望使用新的功能来丰富自己的网站，而又不用深入钻研JavaScript，本书是你的必备之选。

本书中涵盖了jQuery UI 1.8版本中的以下扩展：

- 选项卡管理；
- 折叠菜单；
- 对话框；
- 按钮；
- 进度条；
- 滑块；
- 日历；
- 自动补全；
- 拖放管理；
- 选择元素、缩放和元素调序；
- 新的视觉特效。

Eric Sarrion管理着一个小型的培训与开发公司，曾为O'Reilly法国分部写过多本有关Rails、HTML/CSS、J2EE以及JavaScript方面的书。

O'REILLY®
oreilly.com.cn

O'Reilly Media, Inc.授权人民邮电出版社出版

此简体中文版仅限于中国大陆（不包含中国香港、澳门特别行政区和中国台湾地区）销售发行

This Authorized Edition for sale only in the territory of People's Republic of China
(excluding Hong Kong, Macao and Taiwan)

分类建议：计算机/网络技术

人民邮电出版社网址：www.ptpress.com.cn



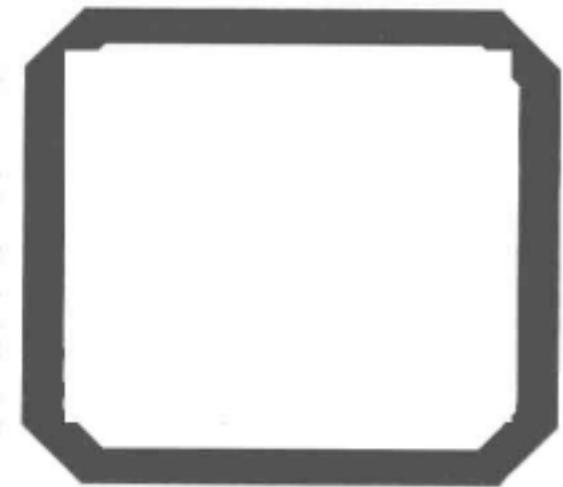
ISBN 978-7-115-29523-1



9 787115 295231 >

ISBN 978-7-115-29523-1

定价：39.00 元



jQuery UI 开发指南

[美] Eric Sarrion 著
罗晴明 包勇明 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

jQuery UI开发指南 / (美) 萨里恩 (Sarrion, E.)
著 ; 罗晴明, 包勇明译. -- 北京 : 人民邮电出版社,
2012.12

书名原文: jQuery UI
ISBN 978-7-115-29523-1

I. ①j… II. ①萨… ②罗… ③包… III. ①
JAVA语言—程序设计—指南 IV. ①TP312-62

中国版本图书馆CIP数据核字(2012)第224669号

版权声明

Copyright ©2012 by O'Reilly Media, Inc.

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

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

本书中文简体版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可，对本书的任何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

jQuery UI 开发指南

-
- ◆ 著 [美] Eric Sarrion
 - 译 罗晴明 包勇明
 - 责任编辑 杨海玲
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 787×1000 1/16
 - 印张: 14.25
 - 字数: 286 千字 2012 年 12 月第 1 版
 - 印数: 1~3 500 册 2012 年 12 月河北第 1 次印刷
 - 著作权合同登记号 图字: 01-2012-6403 号
 - ISBN 978-7-115-29523-1
-

定价: 39.00 元

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223
反盗版热线: (010) 67171154

内 容 提 要

本书是 jQuery UI 的快速开发指南，全面介绍 jQuery UI 如何改进 HTML 页面展示，内容包括 jQuery UI 1.8 版本的下列扩展特性：选项卡管理、折叠菜单、对话框、按钮、进度条、滑块、日历、自动补全、拖放管理、选择元素、缩放、元素调序以及新的视觉特效等。

本书共 14 章，讲述了 jQuery UI 的所有功能。作者用精简的代码展示了 jQuery UI 组件的使用方法，挑选出最为重要的配置选项和组件事件进行详细解释说明。精选的示例循序渐进、由易到难，从假设问题到解决问题，显示了作者挖掘 jQuery UI 功能的思考过程。

本书适合 Web 开发人员及设计人员阅读。无论是 jQuery UI 的初学者还是有一定经验的开发人员，本书都是极好的技术参考书。

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

本书是由 O'Reilly 出版的第一本关于 jQuery UI 的书籍。

前端技术已经受到越来越多程序员的青睐，相信有很多前端工程师，包括我自己，都渴望在短时间内学习并掌握 jQuery UI，特别是在项目开始时犹豫是否使用 jQuery UI 的时候，本书对于学习 jQuery UI 会有极大的帮助。

jQuery UI 的官方文档有些地方写得过于简单，没有详细解释，不易理解。本书作者 Eric Sarrion 在每章的开始先介绍组件的基本使用方法，然后用通俗易懂的语言详细说明组件的配置及事件选项，最后通过精挑细选、循序渐进的示例帮助读者巩固学到的知识点。我在翻译此书的过程中收获颇丰，对 jQuery UI 的细节部分和使用场景都有了更深入的理解。本书是对 jQuery UI 官方文档的强有力的补充和说明，可以作为实际开发时的参考手册。

本书讲解的 jQuery UI 版本是 1.8.16，而非官方最新版本（1.8.23）。可能有朋友担心会不会过时了？几乎所有的技术类书籍都难以避免这样的问题，图书出版的速度永远赶不上技术更新的速度。对于本书（相信还有很多其他书）这种担心是多余的。本书讲解的内容和最新版本的几乎是一样的。本书能够帮助读者掌握 jQuery UI 的使用方法，使初学者能够快速入门，全面掌握 jQuery UI，使有经验的读者在使用 jQuery UI 时能够更加得心应手，提高开发效率，丰富页面展示效果，提升用户体验。

如果你刚刚接触 jQuery UI，希望系统地学习 jQuery UI，本书是极佳的选择。如果你是一位“久经沙场”的开发人员，并已经熟读了 jQuery UI 的官方文档，我依然向你推荐这本书，感受一下别人眼中的 jQuery UI，同时也不失学习的乐趣。

本书也适合非开发人员阅读，比如设计师。不是说让设计师来做开发的工作，而是从中学习“标准”UI 组件的视觉外观和交互逻辑，在设计页面时能开拓思路，设计出更加专业和利于用户交互的界面。

我在一个很偶然的情况下得到了翻译此书的机会。在翻译过程中，我力求在文字与内容上忠实原著，并对每一个技术点都认真进行了验证核对，对每一句话都推敲再三，生怕表达不清而误导了读者。但本人水平有限，加之时间仓促，书中难免有纰漏之处，敬请广大读者批评指正。

本书的翻译由罗晴明和我合作完成，感谢杨海玲老师在翻译过程中给予的充分信任和支持，感谢出版社编辑们认真仔细的编校工作。

包勇明

2012年9月 于上海

包勇明，盛大创新院资深前端工程师，有丰富的前端开发经验，热爱前端和互联网技术，并以无限的热情投身和致力于前端工作，对前端和用户体验相关的一切技术都感兴趣，深知前端在当前互联网中的重要位置及历史使命。罗晴明，资深前端开发者。曾就职于网易和搜狐视频。85后宅男，会说中式英文及“l-n不敏感”普通话。笑点比断点诡异。热衷于在生活的闭包里寻觅泄漏的乐趣。

译者简介

包勇明 盛大创新院资深前端工程师，有丰富的前端开发经验，热爱前端和互联网技术，并以无限的热情投身和致力于前端工作，对前端和用户体验相关的一切技术都感兴趣，深知前端在当前互联网中的重要位置及历史使命。

罗晴明 资深前端开发者。曾就职于网易和搜狐视频。85后宅男，会说中式英文及“l-n不敏感”普通话。笑点比断点诡异。热衷于在生活的闭包里寻觅泄漏的乐趣。

前言

jQuery 是一个很流行的 JavaScript 库，可以通过插件进行功能扩展。有些插件，尤其是那些管理用户界面的插件，都已经被收集到 jQuery UI 库中了。这些插件有助于用户交互，如果你只使用 jQuery，管理这些交互功能会更加简单。

本书涵盖了 jQuery UI 1.8 版本中的以下扩展：

- 选项卡管理；
- 折叠菜单；
- 对话框；
- 按钮；
- 进度条；
- 滑块；
- 日历；
- 自动补全；
- 拖放管理；
- 选择元素、缩放和元素调序；
- 新的视觉特效。

本书的读者对象

所有使用 jQuery 的人都应该阅读这本书！对于那些想改进网站的用户界面和想用新功能来丰富网站的朋友，本书尤其值得一读。

本书的结构

jQuery UI 里面的每个功能组件（选项卡、折叠菜单等）都将单独分章阐述。每章的内容都相互独立，可以根据需要直接使用相应功能。

本书的排版约定

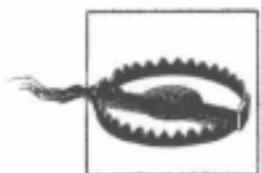
本书使用如下排版约定。

- 等宽代码字体：用于程序清单，以及段落中引用的程序元素，如变量或函数名、数据库、数据类型、环境变量、语句和关键字。
- 加粗的等宽代码字体：展示命令或其他应该由用户输入的文本。
- 斜的等宽代码字体：展示应该用用户提供的值或根据上下文确定的值来替换的文本。



提示

这个图标用来强调一个提示、建议或一般说明。



警告

这个图标用来说明一个警告或注意事项。

代码示例的使用

本书的目标是帮助你完成工作。一般而言，你可以在自己的程序和文档中使用本书中的代码，如果你要复制的不是核心代码，则无须取得我们的许可。例如，你可以在程序中使用本书中的多个代码块，无须获取我们许可。但是，要销售或分发来源于 O'Reilly 图书中的示例的光盘则需要取得我们的许可。通过引用本书中的示例代码来回答问题时，不需要事先获得我们的许可。但是，如果你的产品文档中融合了本书中的大量示例代码，则需要取得我们的许可。

在引用本书中的代码示例时，如果能列出本书的属性信息是最好不过了。属性信息通常包括书名、作者、出版社和 ISBN。例如：“jQuery UI by Eric Sarrion (O'Reilly). Copyright 2012 Eric Sarrion, 978-1-449-31699-0.”

在使用书中的代码时，如果不确定是否属于合理使用，或是否超出了我们的许可，请通过 permissions@oreilly.com 与我们联系。

我们的联系方式

如果你想就本书发表评论或有任何疑问，敬请联系出版社。

美国：

O'Reilly Media Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）

奥莱利技术咨询（北京）有限公司

我们还为本书建立了一个网页，其中包含了勘误表、示例和其他额外的信息。你可以通过如下地址访问该网页：

<http://shop.oreilly.com/product/0636920023159.do>

关于本书的技术性问题或建议，请发邮件到：

bookquestions@oreilly.com

欢迎登录我们的网站（<http://www.oreilly.com>），查看更多我们的书籍、课程、会议和最新动态等信息。

我们的其他联系方式如下：

Facebook: <http://facebook.com/oreilly>

Twitter: <http://twitter.com/oreillymedia>

YouTube: <http://www.youtube.com/oreillymedia>

目录

第 1 章	jQuery UI 介绍	1
1.1	jQuery UI 的安装	1
1.2	jQuery UI 概览	2
1.3	什么是 CSS 主题	3
1.4	在 HTML 页面中应该引入哪些文件	4
1.4.1	未压缩文件	4
1.4.2	压缩文件	6
1.5	更换 CSS 主题	8
1.6	准备开始	8
第 2 章	选项卡	9
2.1	选项卡的基本用法	9
2.2	格式化内容	11
2.3	tabs ()方法	13
2.3.1	tabs (options) 方法	13
2.3.2	tabs ("action", params) 方法	14
2.4	bind ()方法	15
2.5	使用选项卡的示例	16
2.5.1	动态创建选项卡	16
2.5.2	使用 Ajax 修改选项卡的内容	17
2.5.3	通过 Ajax 把信息传输给服务器	18
2.5.4	使用选项卡的 add 方法	20
2.5.5	使用 tabsadd 事件	21
第 3 章	折叠菜单	23
3.1	折叠菜单的基本用法	23
3.2	格式化内容	25
3.3	accordion ()方法	27
3.3.1	accordion (options) 方法	27
3.3.2	accordion ("action", params) 方法	29
3.4	在折叠菜单上使用 bind ()管理事件	29
3.5	使用折叠菜单的示例	30
3.5.1	打开任意菜单	30
3.5.2	用 Ajax 加载菜单内容：使用 options	31

3.5.3 用Ajax 加载菜单的内容：使用 accordionchange	32
第4章 对话框	34
4.1 对话框的基本用法	34
4.2 格式化内容	36
4.3 dialog ()方法	38
4.3.1 dialog (options)方法	38
4.3.2 dialog ("action", params)方法	42
4.4 在对话框中使用 bind ()处理事件	42
4.5 使用对话框的示例	43
4.5.1 打开和关闭对话框	43
4.5.2 在打开或者关闭对话框时应用特效	45
4.5.3 在关闭对话框时进行验证	45
4.5.4 隐藏关闭按钮	46
4.5.5 在对话框中插入按钮	47
4.5.6 使用 Ajax 插入内容	48
4.5.7 使用特效来更改对话框的行为	50
第5章 按钮	53
5.1 按钮的基本用法	53
5.2 格式化内容	54
5.3 button ()方法	55
5.3.1 button (options)方法	55
5.3.2 button ("action", params)方法	57
5.4 在按钮上使用 bind ()处理事件	57
5.5 单选按钮	57
5.5.1 显示单选按钮	57
5.5.2 使用 buttonset ()方法改进显示	58
5.6 复选框	59
5.6.1 显示复选框	60
5.6.2 使用 buttonset ()方法改进显示	60
5.7 使用按钮的示例	61
5.7.1 在按钮上显示图标	61
5.7.2 创建计算器	64
第6章 进度条	71
6.1 进度条的基本用法	71
6.2 格式化内容	72
6.3 progressbar ()方法	73
6.3.1 progressbar (options)方法	73

6.3.2	progressbar ("action", params)方法	73
6.4	在进度条上使用 bind () 处理事件	74
6.5	使用进度条的示例	74
6.5.1	逐渐填充进度条	74
6.5.2	在完成过程中的不同阶段进行处理	75
第 7 章	滑块	77
7.1	滑块的基本用法	77
7.2	格式化内容	78
7.3	slider () 方法	79
7.3.1	slider (options) 方法	79
7.3.2	slider ("action", params) 方法	81
7.4	在滑块上使用 bind () 管理事件	82
7.5	使用滑块的示例	82
7.5.1	显示单个游标的值	82
7.5.2	显示两个游标的值	84
7.5.3	使用滑块来调整图片的透明度	85
第 8 章	日历	87
8.1	日历的基本用法	87
8.2	格式化内容	88
8.3	datepicker () 方法	90
8.3.1	datepicker (options) 方法	90
8.3.2	datepicker ("action", params) 方法	95
8.4	使用日历的示例	95
8.4.1	用其他语言显示日历	95
8.4.2	在日历中显示多个月份	96
8.4.3	显示静态日历	99
8.4.4	指定最小及最大日期	99
8.4.5	禁止选择特定的日期	100
8.4.6	预先选择任意日期	101
8.4.7	在选择日期时执行 Ajax 请求	104
第 9 章	自动补全	106
9.1	自动补全的基本用法	106
9.2	格式化内容	107
9.3	autocomplete () 方法	109
9.3.1	autocomplete (options) 方法	109
9.3.2	autocomplete ("action", params) 方法	110
9.4	在建议列表上使用 bind () 管理事件	111

9.5 使用自动补全机制的示例.....	112
9.5.1 指定建议列表的宽度.....	112
9.5.2 打开 HTML 页面时即显示建议列表.....	113
9.5.3 在输入框获得焦点时显示建议列表.....	114
9.5.4 为建议列表的显示营造特效.....	114
9.5.5 动态创建建议列表.....	115
9.5.6 基于用户输入的数据动态创建建议列表.....	116
9.5.7 在建议列表中插入图片.....	118
第 10 章 拖放.....	122
10.1 draggable ()方法.....	122
10.1.1 draggable (options) 方法.....	122
10.1.2 draggable ("action", params) 方法.....	127
10.2 在移动元素上使用 bind () 管理事件.....	127
10.3 使用拖动功能的示例.....	128
10.3.1 在移动过程中执行额外处理.....	128
10.3.2 给移动施加一些限制.....	129
10.4 droppable ()方法.....	135
10.4.1 droppable (options) 方法.....	135
10.4.2 droppable ("action", params) 方法.....	137
10.5 在保管元素上使用 bind () 管理事件.....	138
10.6 使用放置功能的示例：购物车.....	138
10.6.1 使用拖放功能来创建一辆购物车.....	138
10.6.2 给购物车的放置过程添加视觉特效.....	141
10.6.3 从购物车中移除图书.....	142
第 11 章 选择元素.....	145
11.1 选择元素的基本用法.....	145
11.2 格式化内容.....	146
11.3 selectable ()方法.....	149
11.3.1 selectable (options) 方法.....	149
11.3.2 selectable ("action", params) 方法.....	150
11.4 在选区中使用 bind () 管理事件.....	151
11.5 使用选择机制的示例.....	151
11.5.1 显示选择序列中的事件发生顺序.....	151
11.5.2 禁止选择某特定元素.....	153
11.5.3 禁止通过点击来选择元素.....	155
11.5.4 管理购物车.....	156

第 12 章	页面内的元素调序	159
12.1	元素调序的基本用法	159
12.2	格式化内容	160
12.3	sortable () 方法	162
12.3.1	sortable (options) 方法	162
12.3.2	sortable ("action", params) 方法	168
12.4	使用 bind () 管理元素调换事件	168
12.5	使用元素调序机制的示例	169
12.5.1	显示事件出现的顺序	169
12.5.2	在列表中放置任何元素	175
第 13 章	缩放	181
13.1	缩放的基本用法	181
13.2	格式化内容	182
13.2	resizable () 方法	183
13.2.1	resizable (options) 方法	183
13.2.2	resizable ("action", params) 方法	186
13.4	在缩放元素上使用 bind () 处理事件	187
13.5	使用缩放机制的示例	187
13.5.1	缩放时显示元素的尺寸	187
13.5.2	缩放时显示元素的位置	188
13.5.3	在缩放过程中执行动画	189
13.5.4	创建一个可缩放的文本框	191
第 14 章	jQuery UI 中的视觉特效	194
14.1	新的视觉特效	194
14.1.1	effect (effectName, options, duration, callback) 方法	194
14.1.2	blind 特效	195
14.1.3	bounce 特效	196
14.1.4	shake 特效	197
14.1.5	clip 特效	197
14.1.6	drop 特效	198
14.1.7	explode 特效	200
14.1.8	fold 特效	201
14.1.9	highlight 特效	201
14.1.10	puff 特效	202
14.1.11	pulsate 特效	203
14.1.12	scale 特效	204
14.1.13	size 特效	204

14.1.14 slide 特效	205
14.2 show ()、hide () 以及 toggle () 方法	206
14.3 jQuery UI 提供的增强的 animate () 方法	207
14.3.1 管理颜色的 CSS 属性	208
14.3.2 新的 easing 选项值	209
14.4 使用 CSS 类生成特效	210
14.4.1 addClass ()、removeClass () 以及 toggleClass () 的 增强型方法	210
14.4.2 switchClass () 方法	211
14.4.3 使用 toggleClass () 方法的示例	211

jQuery UI 介绍

jQuery UI 是 jQuery 的一个插件集，为 jQuery 的核心库添加了新的功能。本章我们将安装 jQuery UI 库，并简略地看一下它的内容。接下来的几章将会详细介绍 jQuery UI 的每一个插件的功能。

1.1 jQuery UI 的安装

jQuery UI 库可以从 <http://jqueryui.com> 下载。单击稳定版本（Stable）的下载链接，此链接直接指向了一个 ZIP 文件，它包含了 jQuery UI 的源代码、示例及文档。文件下载完成后，把它解压缩到一个名为 jqueryui 的目录中。

现在，这个 jqueryui 目录包含了以下内容。

- 一个 css 子目录，包含与 jQuery UI 相关的 CSS 文件。之后你会知道 jQuery UI 是使用 CSS 主题来管理界面元素的自定义样式的。例如，不同主题下的滑块可能显示得不一样，其他组件（如日历和选项卡）也可能显示得不一样。
- 一个 js 子目录，包含 jQuery UI 的 JavaScript 文件。这个目录包含一个拥有 jQuery UI 全部功能的压缩文件和一个 jQuery 库文件。
- 一个 development-bundle 子目录，包含多个不同的子目录——demos（jQuery UI 的示例文件）、docs（jQuery UI 的文档文件）、themes（所有与 jQuery UI 相关联的 CSS 主题文件）以及 ui（jQuery UI 的 JavaScript 文件）。
- 一个 index.html 文件，可以在浏览器中打开这个文件，并查看一些 jQuery UI 的功能。

1.2 jQuery UI 概览

为了简要了解 jQuery UI，请在浏览器中打开 index.html 文件（如图 1-1 所示）。



图 1-1 jQuery UI 首页

在这个文件中，你可以看到 jQuery UI 添加的不同功能，如图 1-2 所示，包括：

- 折叠菜单（accordion menu）；
- 输入框的自动补全机制（autocomplete mechanism）；
- 漂亮的按钮（button）和复选框（checkbox）；
- 便于页面展示的选项卡机制（tab mechanism）；
- 显示在页面最上层的对话框（dialog box）；
- 自定义图标（custom icon）；
- 滑块（slider）；
- 日历（calendar）；
- 进度条（progress bar）。

我们将会在本书的后面章节，讨论这些功能。我们还会讨论其他的机制，比如拖放、新的视觉特效、CSS 主题文件等。

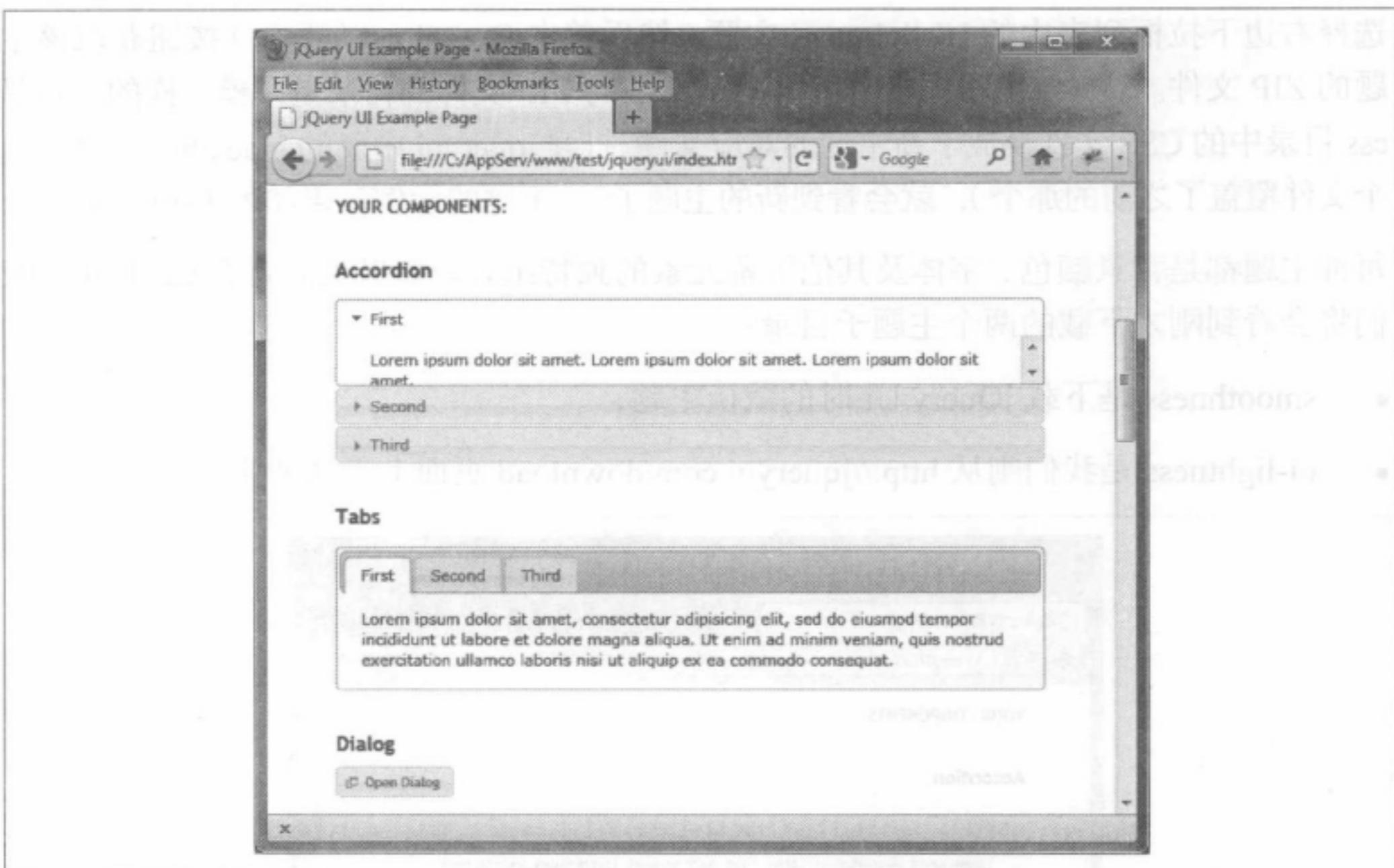


图 1-2 jQuery UI 首页：组件列表

1.3 什么是 CSS 主题

我们之前提到的 CSS 主题是什么呢？为了弄清楚这个，只需按主题选择下载一个新的 jQuery UI 的定制版本。为此，打开 <http://jqueryui.com/download>，显示的页面如图 1-3 所示。

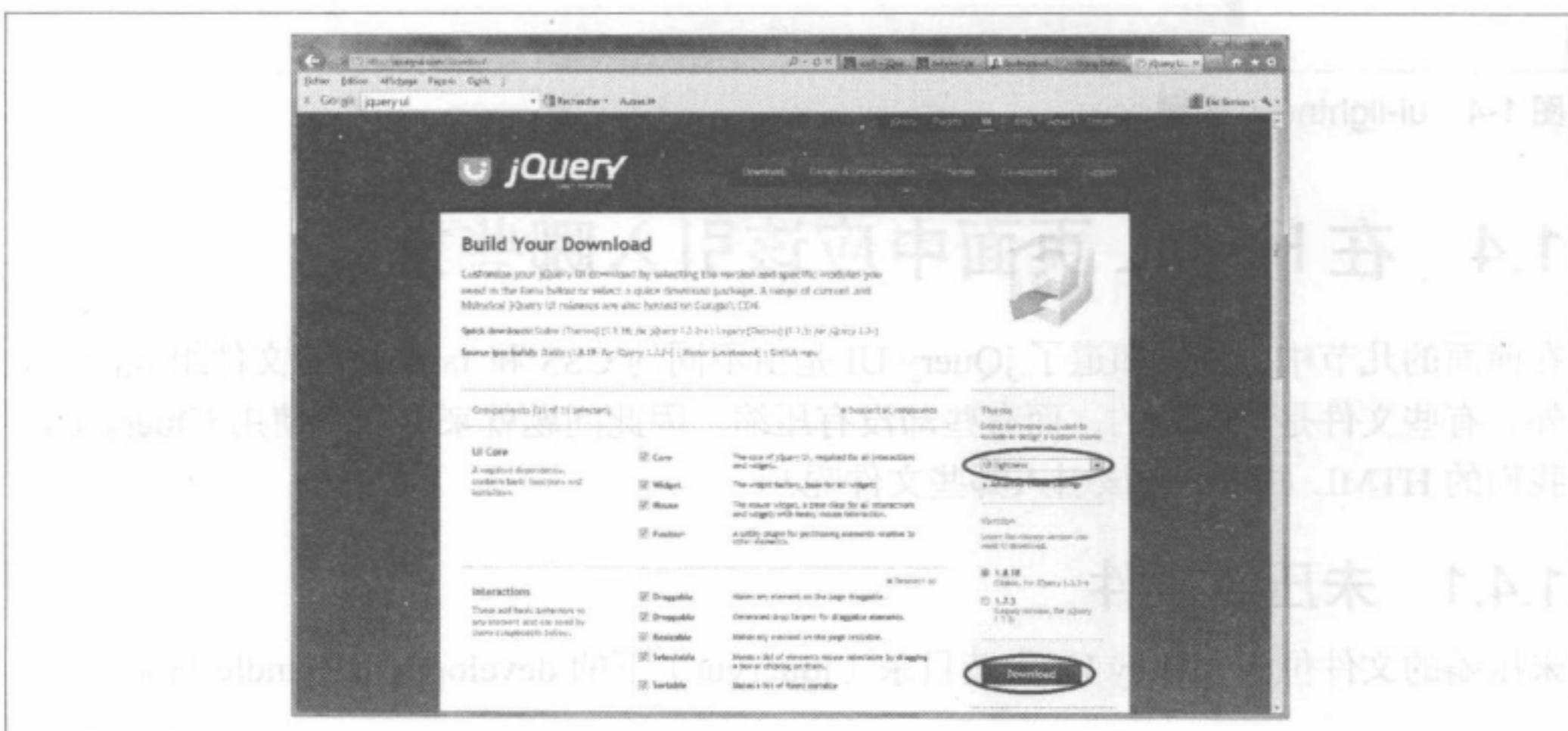


图 1-3 下载定制的 jQuery UI 主题

选择右边下拉框列表中的 UI lightness 主题，然后单击 Download (下载) 按钮获取该主题的 ZIP 文件。这个 ZIP 文件的目录和之前下载的 ZIP 文件的目录是一模一样的，只是 css 目录中的 CSS 文件换成了新主题的对应文件。打开 jqueryui 中新的 index.html 文件(这个文件覆盖了之前的那个)，就会看到新的主题了^①。主题的示例效果如图 1-4 所示。

每种主题都是背景颜色、字体及其他屏幕元素的独特组合。如果我们查看 css 目录，我们将会看到刚才下载的两个主题子目录：

- smoothness 是下载 jQuery UI 时的默认主题；
- ui-lightness 是我们刚从 <http://jqueryui.com/download> 页面上下载的主题。

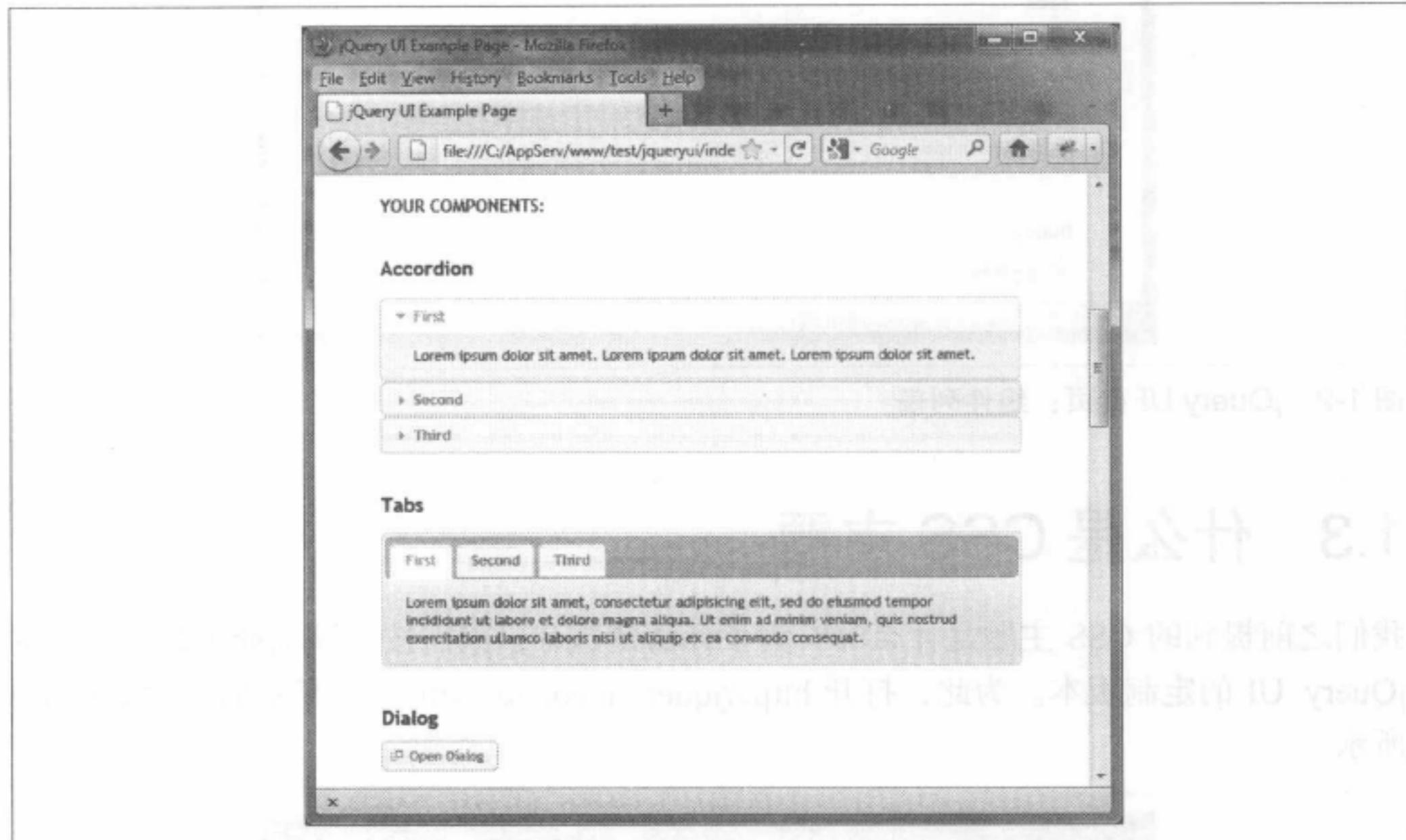


图 1-4 ui-lightness 主题

1.4 在 HTML 页面中应该引入哪些文件

在前面的几节中，我们知道 jQuery UI 是由不同的 CSS 和 JavaScript 文件组成的。此外，有些文件是压缩过的，而有些却没有压缩。因此问题就来了：要使用 jQuery UI，我们的 HTML 页面中应该引入哪些文件呢？

1.4.1 未压缩文件

未压缩的文件位于 jQuery UI 安装目录 (jqueryui) 下的 development-bundle 目录。

① 新下载的主题也是解压缩到 jqueryui 目录。——译者注

1. JavaScript 文件

ui 目录（位于 development-bundle 下）包含了 JavaScript 文件。jquery.ui.core.js 文件包含了基本功能（必须引入），而其他的文件需要时再引入。最终会有一个包含了所有 JavaScript 文件的 custom.js（例如，jquery-ui-1.8.16.custom.js）文件，这样就无须单独引入每个文件了。minified 目录（位于 ui 目录下）包含了压缩格式的相同文件。

2. CSS 文件

themes 目录（位于 development-bundle 目录下）包含了 CSS 文件。它由多个目录组成，每个目录包含一个主题（比如，base、smoothness 及 ui-lightness 目录）。每个主题下有一个 images 目录和其他的 CSS 文件。

jquery.ui.core.css 文件包含了基本功能（必需的），而其他文件需要时再引入。jquery.ui.theme.css 文件定义了主题自身的样式（必需的）。

jquery.ui.base.css 文件包含了 development-bundle 目录下除了 jquery.ui.theme.css 外的所有文件。jquery.ui.all.css 文件包含了所有的文件（即 jquery.ui.base.css 和 jquery.ui.theme.css）。

最终，会有一个包含所有 CSS 文件的 custom.css（例如，jquery-ui-1.8.16.custom.css）文件，而无须单独引入每个文件了（这一点对于 jquery.ui.all.css 也是一样的，只是 jquery.ui.all.css 通过 CSS 指令@import 引入其他文件，而 custom.css 把所有文件的内容都包括在内了）。

3. 引入未压缩文件的 HTML 页面示例

我们想显示一个包含两个选项卡的简单页面。主要的 JavaScript 文件是 jquery.ui.tabs.js，主要的 CSS 文件是 jquery.ui.tabs.css。主页面会引入以下基础文件：^①

```
<script src = "jquery.js"></script>
<script src = "jqueryui/development-bundle/ui/jquery.ui.core.js"></script>
<script src = "jqueryui/development-bundle/ui/jquery.ui.widget.js"></script>
<script src = "jqueryui/development-bundle/ui/jquery.ui.tabs.js"></script>

<link rel=stylesheet type=text/css
      href=jqueryui/development-bundle/themes/smoothness/jquery.ui.core.css />
<link rel=stylesheet type=text/css
      href=jqueryui/development-bundle/themes/smoothness/jquery.ui.theme.css />
<link rel=stylesheet type=text/css
      href=jqueryui/development-bundle/themes/smoothness/jquery.ui.tabs.css />
```

这里的 jquery.js 文件和 jqueryui 目录位于同一层目录级别。它是标准的 jQuery JavaScript 文件。

core.js 文件是必须引入的，tab.js 文件还需要引入依赖的 widget.js（正如 tab.js 文件中指出的）。^①

core.css 文件是必须引入的，theme.css 文件也是必须引入的。tab.css 文件则包含了选项卡的详细样式定义。

既然已经搭建了页面的基本模块，我们再创建并标记两个选项卡，并在每个选项卡里面添加一些文字。下面的代码紧接着之前引入基础文件的那部分代码。

```
<div id=tabs>
  <ul>
    <li><a href="#tab1>Tab 1</a></li>
    <li><a href="#tab2>Tab 2</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
</div>

<script>
  $( "#tabs" ).tabs ();
</script>
```

这段脚本的结果（一个包括两个选项卡的页面）如图 1-5 所示。



图 1-5 我们的第一个使用 jQuery UI 的程序

1.4.2 压缩文件

使用压缩文件能减少 HTML 页面的加载时间。

① 每个 jQuery UI 插件的 js 文件，在最开始的注释区域都会注明它所依赖的 js 文件。——译者注

1. JavaScript 文件

js 目录（位于 jQuery UI 安装目录下面，这里是 jqueryui）包含了 JavaScript 文件。这里只需要 jquery-ui-1.8.16.custom.min.js 文件。目录中的另外一个文件是压缩版本的 jQuery。

2. CSS 文件

css 目录（位于 jQuery UI 安装目录下面，这里是 jqueryui）包含了每个已安装主题的子目录（例如，smoothness 和 ui-lightness 目录）。

每个主题都包括了一个 images 目录和一个会被引入到 HTML 页面中的 CSS 文件。这和最终的 custom.css（比如，jquery.ui-1.8.16.custom.css）是一样的，是个压缩版本。

3. 引入压缩文件的 HTML 示例页面

我们想显示一个有两个选项卡的页面（和之前的一样）：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />
```

除了 jquery.js 文件外，现在只需要两个文件了：

- jQuery UI 的 JavaScript 全局文件（jquery-ui-1.8.16.custom.min.js）；
- 与使用的样式相关的整个 jQuery UI CSS 文件（smoothness/jquery-ui-1.8.16.custom.css，与 smoothness 主题相关的文件）。

现在再添加与之前一样的 HTML 代码来创建、标记和填充选项卡：

```
<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
</div>

<script>
  $(#tabs).tabs();
</script>
```

结果和之前的是一样的。

1.5 更换 CSS 主题

CSS 主题最好的地方是允许你方便地更换页面的外观——仅需更换你想要使用的主题目录名字就可以了。比如，把 smoothness 换成 ui-lightness。

在使用未压缩版本文件的页面中，只需把页面中每个 smoothness 目录改成 ui-lightness 即可（粗体部分所示）：

```
<link rel=stylesheet type=text/css  
      href=jqueryui/development-bundle/themes/ui-lightness/jquery.ui.core.css />  
<link rel=stylesheet type=text/css  
      href=jqueryui/development-bundle/themes/ui-lightness/jquery.ui.theme.css />  
<link rel=stylesheet type=text/css  
      href=jqueryui/development-bundle/themes/ui-lightness/jquery.ui.tabs.css />
```

在使用压缩文件的页面中，把 smoothness 目录改成 ui-lightness（粗体部分所示）：

```
<link rel=stylesheet type=text/css  
      href=jqueryui/css/ui-lightness/jquery-ui-1.8.16.custom.css />
```

HTML 页面将使用新主题（如图 1-6 所示）。

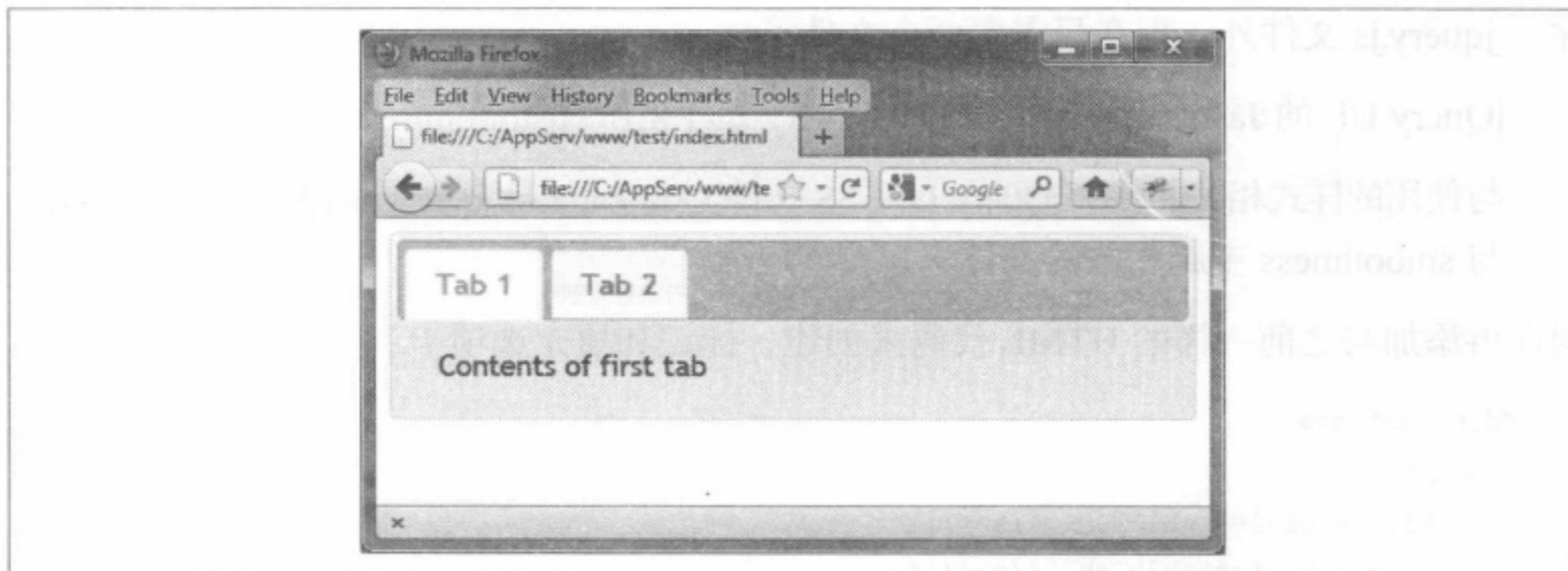


图 1-6 使用 ui-lightness 主题的 HTML 页面

1.6 准备开始

在快速浏览了 jQuery UI 能为我们的 HTML 页面做什么之后，接下来我们将深入了解每个组件的细节，首先是选项卡。

选项卡

在目前的网站中，使用选项卡（tab）的HTML页面已经是很常见了。你可以使用选项卡按话题来组织一个站点的信息，这样用户选择相关的选项卡就可以快速方便地找到相关信息了。

2.1 选项卡的基本用法

假设我们想要编写一段HTML代码来显示图2-1所示的选项卡。它有一个选项卡栏（包含3个选项卡）并且每个选项卡里面的内容是不同的。

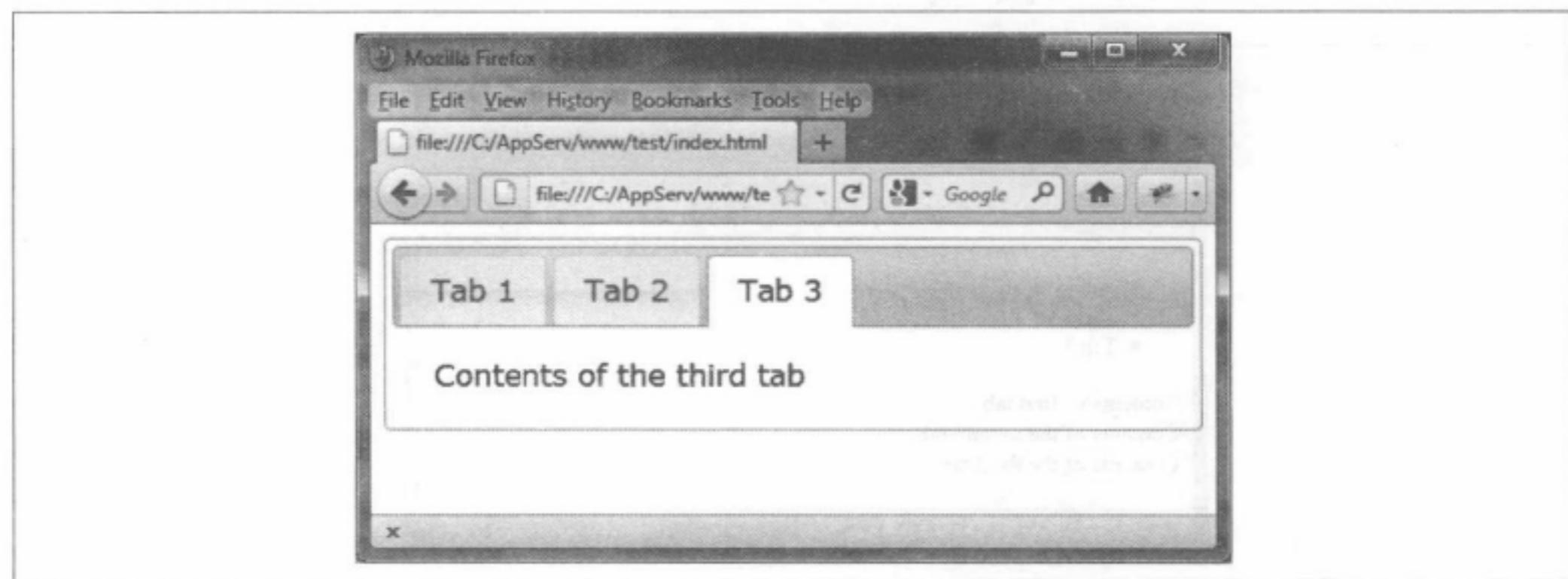


图2-1 HTML页面中的选项卡

要用jQuery UI创建这种类型的页面，需要以下内容：

- 一个包含整个选项卡的

块；
- 一个构成选项卡栏的
元素；

- 每个选项卡对应的一个元素；
- 每个选项卡的窗口对应的一个<div>元素。

下面是创建如图 2-1 所示页面的代码：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
  <div id=tab3>Contents of the third tab</div>
</div>

<script>
</script>
```

如果在浏览器中打开此页面（如图 2-2 所示），你会看到结果和预期的很不一样。为了得到想要的结果，还必须使用 jQuery UI 的 tabs () 方法。

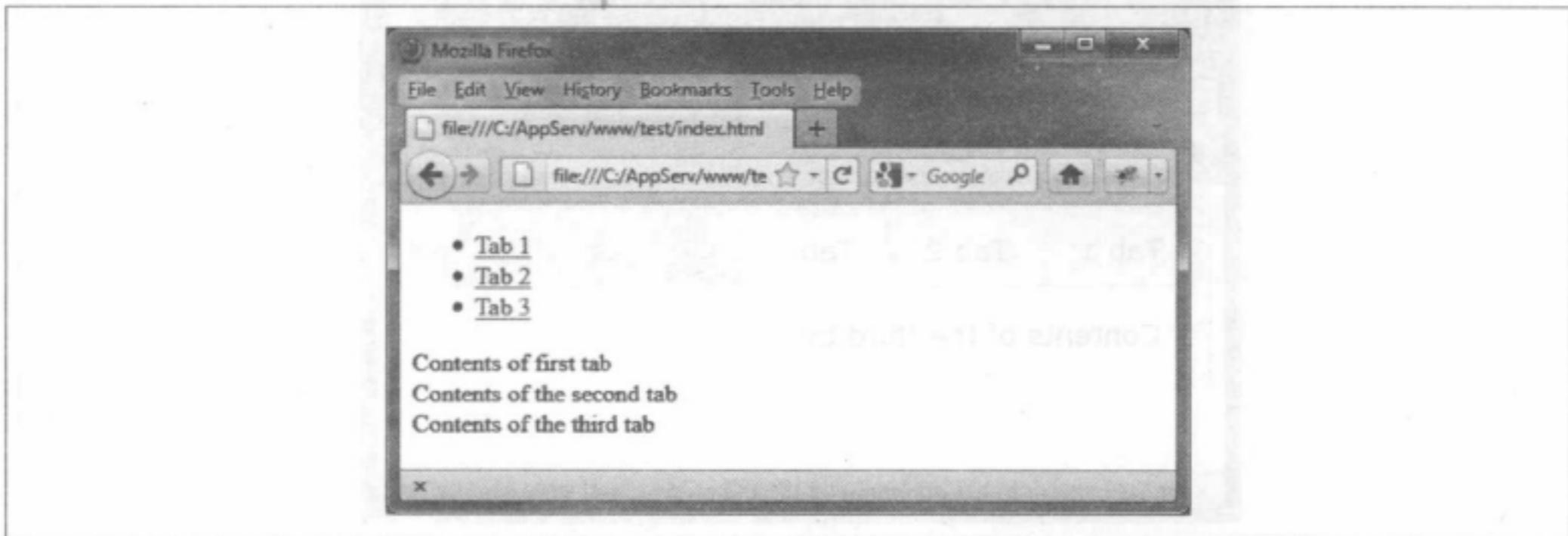


图 2-2 选项卡在 HTML 页面中没有按预期的那样显示

在页面的<script>标签里面添加下面这行代码(粗体部分所示)，调用 jQuery UI 的 tabs 方法，使用选项卡来管理 HTML 元素：

```
<script>
  \$("\#tabs") . tabs();
</script>
```

```
</script>
```

这样页面就会出现选项卡了。

tabs ()方法是jQuery UI的多个方法中的一个，可以在由jQuery ()函数返回的jQuery类对象上调用该方法，然后由选择器选取的结果集中的元素就被转换成选项卡了。当用户单击某个选项卡时，会自动切换到该选项卡，这一切都是jQuery UI“透明地”在管理的。

2.2 格式化内容

使用 tabs ()方法会彻底改变页面中HTML元素的外观。实际上，该方法遍历了（jQuery UI内部实现）HTML代码，并给相关的元素（这里是选项卡）添加了新的CSS类，赋予了它们适当的样式。

图2-3展示了jQuery UI生成的HTML，tabs ()方法改变了HTML DOM（文档对象模型）树（代码是用Firefox的Firebug扩展查看的）。



The screenshot shows the Firebug DOM Inspector with the 'HTML' tab selected. The tree view displays the following structure:

```
<html>
  <head>
  <body>
    <div id="tabs" class="ui-tabs ui-widget ui-widget-content ui-corner-all">
      <ul class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-all">
        <li class="ui-state-default ui-corner-top ui-tabs-selected ui-state-active">
          <a href="#tab1">Tab 1</a>
        </li>
        <li class="ui-state-default ui-corner-top">
          <a href="#tab2">Tab 2</a>
        </li>
        <li class="ui-state-default ui-corner-top">
          <a href="#tab3">Tab 3</a>
        </li>
      </ul>
      <div id="tab1" class="ui-tabs-panel ui-widget-content ui-corner-bottom">Contents of first tab</div>
      <div id="tab2" class="ui-tabs-panel ui-widget-content ui-corner-bottom ui-tabs-hide">Contents of the second tab</div>
      <div id="tab3" class="ui-tabs-panel ui-widget-content ui-corner-bottom ui-tabs-hide">Contents of the third tab</div>
    </div>
    <script>
    </body>
</html>
```

图2-3 tab ()方法产生的HTML代码

可以修改元素的CSS类来自定义元素的显示样式。比如，如果修改了与元素关联的ui-state-default CSS类，将会得到新的选项卡外观。同样地，如果修改了与<div>元素关联的ui-tabs-panel CSS类，则选项卡的内容外观就改变了。

在HTML中添加一个<style>标签，来修改这些元素的样式（粗体部分所示）：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>
```

```

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<style type=text/css>
  li.ui-state-default {
    font-size : 10px;
  }
  div.ui-tabs-panel {
    font-size : 15px;
    font-family : georgia;
    font-style : italic;
  }
</style>

<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
  <div id=tab3>Contents of the third tab</div>
</div>

<script>
$( "#tabs" ).tabs();
</script>

```

必须在 jQuery UI 样式的后面添加自己的样式，否则更改是会被“忽略”的。^①

如图 2-4 所示，应用新的样式后，选项卡及其内容的外观改变了。

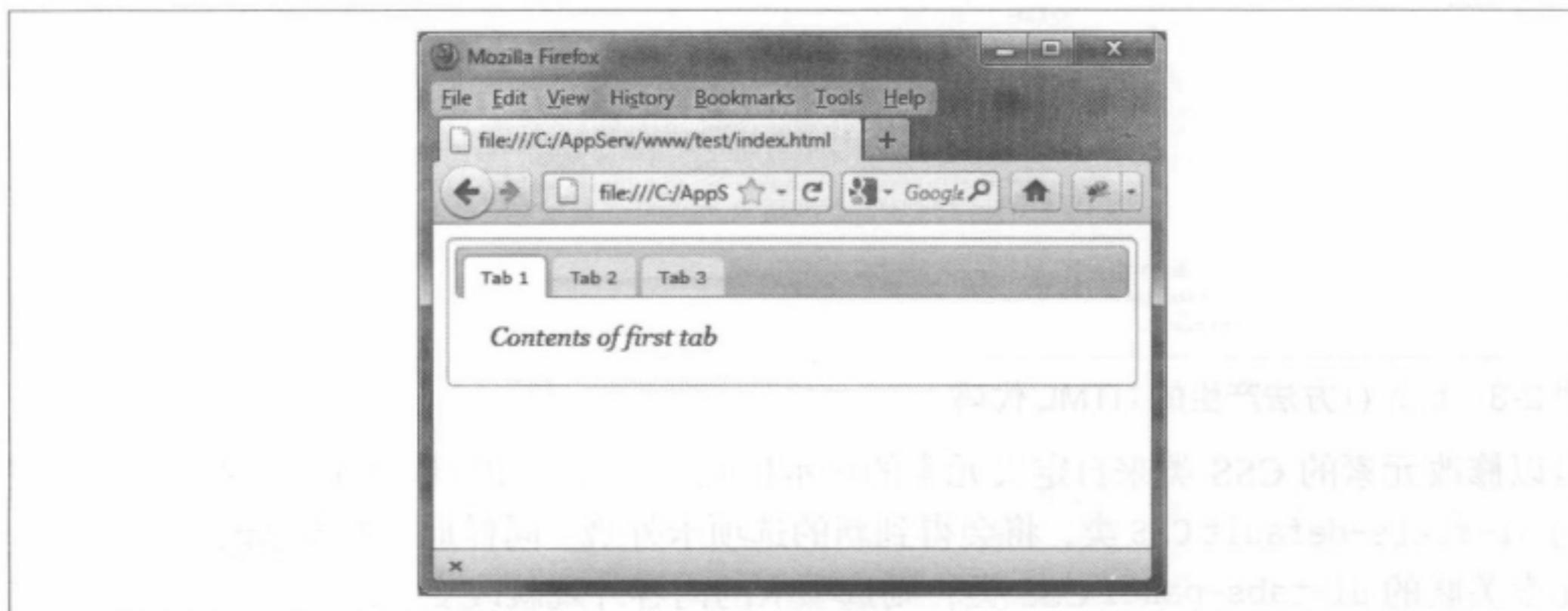


图 2-4 自定义样式的选项卡

^① 如果自己添加的样式写在 jQuery UI 的前面，则根据 CSS 的优先级规则，自己添加的样式会被 jQuery UI 的样式覆盖，就像被“忽略”了一样。——译者注

2.3 tabs () 方法

tabs ()方法可以按以下两种形式使用：

- `$(selector, context).tabs (options)`
- `$(selector, context).tabs ("action", params)`

2.3.1 tabs (options) 方法

tabs (options)方法声明把一个 HTML 元素（及其内容）当作选项卡来管理。options 参数是一个对象，用来指定选项卡的外观及行为。options 参数有不同类型的值，它们要么是直接用于管理选项卡，要么就是管理和选项卡有关的事件。

1. 选项卡的外观和行为

表 2-1 描述了修改选项卡的外观和行为的选项。

表 2-1 管理选项卡的外观和行为的选项

选 项	功 能
<code>options.collapsible</code>	当设置为 <code>true</code> 时，允许取消选中选项卡。当设置为 <code>false</code> （默认值）时，单击选中的选项卡不会取消选中（它仍然被选中）
<code>options.disabled</code>	使用一个数组来指定禁用的选项卡的索引（因此不能选中）。比如，用 <code>[0,1]</code> 来禁用前两个选项卡
<code>options.selected</code>	指定首次选中的选项卡索引。默认为 0，表示页面中的第一个选项卡
<code>options.event</code>	让用户选中新选项卡的事件的名称（默认是"click"）。例如，如果此选项设置为"mouseover"，则当鼠标光标移至选项卡时，就会选中该选项卡
<code>options.fx</code>	指定伴随选择选项卡时的特效，比如渐进地显示选项卡及其内容（指定 <code>options.fx = {{opacity:"toggle"}}</code> ）
<code>options.ajaxOptions</code>	指定 Ajax 的选项（通过 Ajax 来更新选项卡的内容）。比如， <code>options.ajaxOptions.data</code> 可以指定发送给服务器的参数

2. 管理与选项卡相关的事件

有些选项可用于管理选项卡，如选择、添加和删除选项卡。这些选项（表 2-2 列出

的)接收 event 参数, 随后的参数是发生事件的 tab 对象。这个 tab 对象由以下属性组成:

- index, 发生事件的选项卡的索引(0 表示第一个选项卡);
- panel, 对应选项卡内容的<div>元素。

表 2-2 管理选项卡的选项

选 项	功 能
options.select	select (event, tab) 方法在选中选项卡时会被调用(要么手动选中, 要么调用 tabs ("select") 方法)
options.show	show (event, tab) 方法在选项卡的内容变得可见时会被调用(手动选中、首次显示选中的选项卡时或者调用 tabs ("select") 方法)
options.add	add (event, tab) 方法在添加选项卡时会被调用(调用 tabs ("add") 方法)
options.remove	remove (event, tab) 方法在删除选项卡时会被调用(调用 tabs ("remove") 方法)
options.enable	enable (event, tab) 方法在激活选项卡时会被调用(调用 tabs ("enable") 方法)
options.disable	disable (event, tab) 方法在禁用选项卡时会被调用(调用 tabs ("disable") 方法)
options.load	load (event, tab) 方法在使用 Ajax 加载或查看选项卡的内容时会被调用(调用 tab ("load") 方法)

2.3.2 tabs ("action", params) 方法

和前面的 tabs (options) 方法不同, 这个新形式的方法能在创建选项卡后修改选项卡的行为。

tabs ("action", params) 方法允许通过 JavaScript 程序操作选项卡, 比如选择、禁用、添加或者删除选项卡。第一个参数是一个字符串, 指定是什么操作(例如, "add"表示添加选项卡), 随后的是和这个操作有关的参数(例如, 选项卡的索引)。

调用这些方法有时会引起一个和操作同名的事件发生(add 事件由"add"这个操作触发)。这些在 options 中处理的事件已经讨论过了, 表 2-3 列出了可以执行的操作。

表 2-3 tabs ("action", params) 方法的操作

方 法	功 能
tabs ("add", "#id", title, index)	在 index 指定的位置（从 0 开始）添加选项卡。该添加的选项卡后面的选项卡的索引号增 1。"#id" 是和这个选项卡内容相关的 <div>元素的 id(<div>由 jQuery UI 创建时，它的内容应稍后再添加)；title 参数是选项卡的标题；如果未指定 index 参数，选项卡会被添加在列表的末尾</div>
tabs ("remove", index)	移除指定的选项卡及其内容
tabs ("disable", index)	禁用指定的选项卡
tabs ("enable", index)	激活指定的选项卡
tabs ("select", index)	选中指定的选项卡，该选项卡的内容变得可见
tabs ("url", index, url)	将选项卡的内容和由 url 参数指定的 URL 联系起来。这会调用 tabs ("load", index)方法，通过 Ajax 方式来获取选项卡的内容
tabs ("load", index)	根据 tabs ("url", index, url)中的 url 参数指定的 URL，通过 Ajax 方式获取选项卡的内容
tabs ("rotate", duration, repeat)	按指定的时间周期 duration (以毫秒为单位)，定期选中每个选项卡。如果 repeat 为 true，则重复周期，否则只执行一遍（默认）
tabs ("destroy")	移除选项卡管理。选项卡再次成了没有 CSS 类和事件管理的简单 HTML
tabs ("length")	返回由选择器选取的结果集中的第一个元素中的选项卡个数

2.4 bind () 方法

除了在 tabs (options) 方法的选项中使用事件方法外，jQuery UI 还支持使用 bind () 方法来管理这些事件。jQuery UI 为选项卡创建了不同的事件，在表 2-4 列出。

表 2-4 管理选项卡的 jQuery UI 事件

事 件	功 能
tabsselect	选中了选项卡（手动选中或者使用 tabs ("select")方法）
tabsshow	选项卡内容变得可见了（手动选中、首次显示选中的选项卡时或者调用 tabs("select")方法）

续表

事 件	功 能
tabsadd	添加了选项卡（调用 tabs ("add")方法）
tabsremove	移除了选项卡（调用 tabs ("remove")方法）
tabsenable	激活了选项卡（调用 tabs ("enable")方法）
tabsdisable	禁用了选项卡（调用 tabs ("disable")方法）
tabsload	选项卡的内容已通过 Ajax 加载完成了（调用 tabs ("load")方法）

有了这些事件，就可以利用 bind (eventName, callback) 提供的回调方法，完成一些处理工作。

2.5 使用选项卡的示例

在本节中，我们将运用所学到的有关选项卡的知识。

2.5.1 动态创建选项卡

我们想用 JavaScript 动态创建选项卡（及其内容）。在下面显示的代码中，首先用 HTML 代码创建了 3 个选项卡，然后用 JavaScript 添加了第 4 个：

```

<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
  <div id=tab3>Contents of the third tab</div>
</div>

<script>

  $("#tabs").tabs ({
    fx : { opacity : "toggle" },
  }).tabs ("add", "#tab4", "Tab 4");
  $("<i>Contents of the fourth tab</i>").appendTo ("#tab4");

```

```
</script>
```

请注意我们是如何将第一个 tabs ()方法和第二个 tabs ()方法链接起来的。第一个 tabs ()方法是必需的，因为是它把编写的 HTML 代码转变成了显示选项卡的代码（使用jQuery UI 自动添加的 CSS 类），第二个 tabs ()方法执行了“add”操作，这将在列表的末尾添加选项卡。

jQuery UI 创建了选项卡，也创建了与之内容相应的<div>。由于这内容是空的，所以添加了最后的 appendTo ()语句来添加内容。

结果如图 2-5 所示。

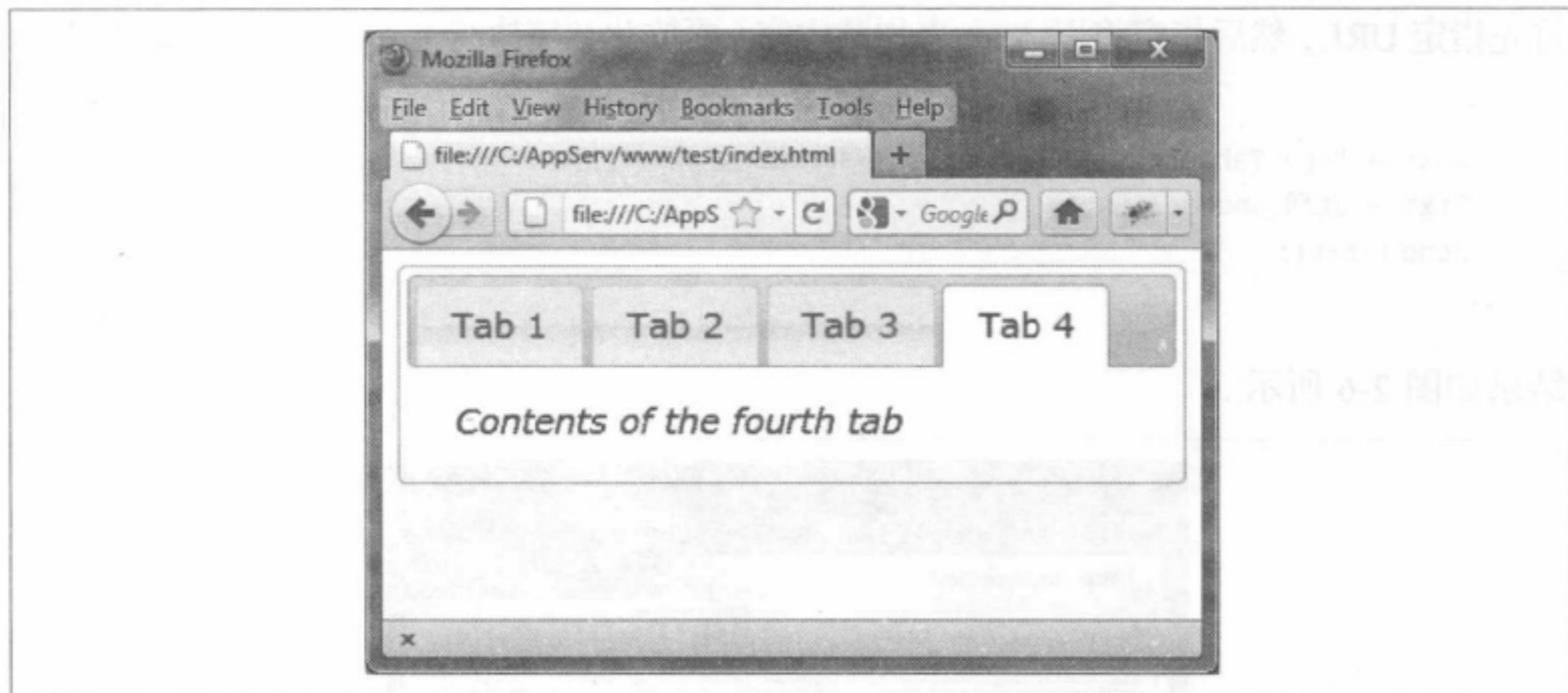


图 2-5 第 4 个选项卡是动态创建的

2.5.2 使用 Ajax 修改选项卡的内容

我们现在想使用服务器返回的 HTML 来初始化选项卡的内容。我们将使用 PHP 服务器。

我们将使用"url"和"load"操作来修改第一个选项卡（索引为 0）的内容。增加的代码如粗体所示：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>
<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
```

```

</ul>
<div id=tab1>Contents of first tab</div>
<div id=tab2>Contents of the second tab</div>
<div id=tab3>Contents of the third tab</div>
</div>

<script>

$("#tabs").tabs({
  fx : { opacity : "toggle" },
}).tabs ("url", 0, "action.php").tabs ("load", 0);

</script>

```

首先指定 URL, 然后指定将以 Ajax 来加载内容(要按此顺序执行)。action.php 文件如下:

```

<?
$txt = "<p> Tab content sent by the server </p>";
$txt = utf8_encode($txt);
echo ($txt);
?>

```

结果如图 2-6 所示。

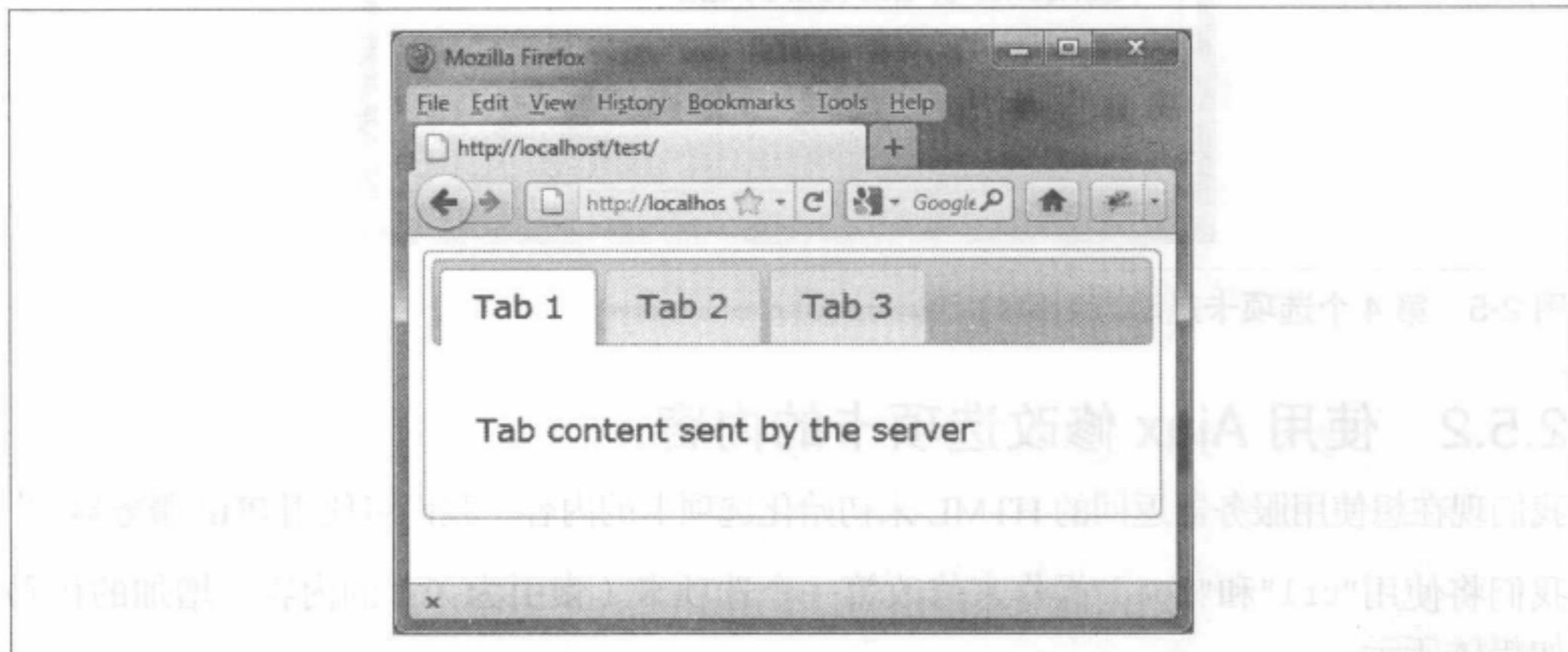


图 2-6 用 Ajax 动态创建的选项卡内容

请注意, 因为在 HTML 页面里使用了 Ajax 请求, 这个 HTML 页面应该使用 HTTP 协议(即地址栏中的 URL 要以 http 开头)来显示, 而不是简单地把 HTML 文件拖放到浏览器中就行了。

接下来看一下如何把信息通过 Ajax 传输给服务器。

2.5.3 通过 Ajax 把信息传输给服务器

这个例子展示了如何通过 Ajax 把信息传输给服务器, 来修改相应选项卡的内容。在这

个例子中，把一个人的名字和姓氏发送给服务器，然后把服务器返回的内容显示在选项卡的内容中。传输的信息（名字和姓氏）是写在 ajaxOptions 选项中的 data 属性里的（如粗体部分所示）：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />
<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
  <div id=tab3>Contents of the third tab</div>
</div>
<script>
  $( "#tabs" ).tabs ({
    fx : { opacity : "toggle" },
    ajaxOptions : { data : { name : "Sarrion", surname : "Eric" } }
  }).tabs ("url", 0, "action.php").tabs ("load", 0);
</script>
```

接收发送参数和显示选项卡内容的 action.php 文件如下：

```
<?
$name = $_REQUEST["name"];
$surname = $_REQUEST["surname"];
$name = utf8_decode ($name);
$surname = utf8_decode ($surname);

$txt = "<p> Tab content sent by the server </p>";
$txt .= "Name : " . $name . "<br />";
$txt .= "Surname : " . $surname . "<br />";
$txt = utf8_encode ($txt);
echo ($txt);
?>
```

结果如图 2-7 所示。

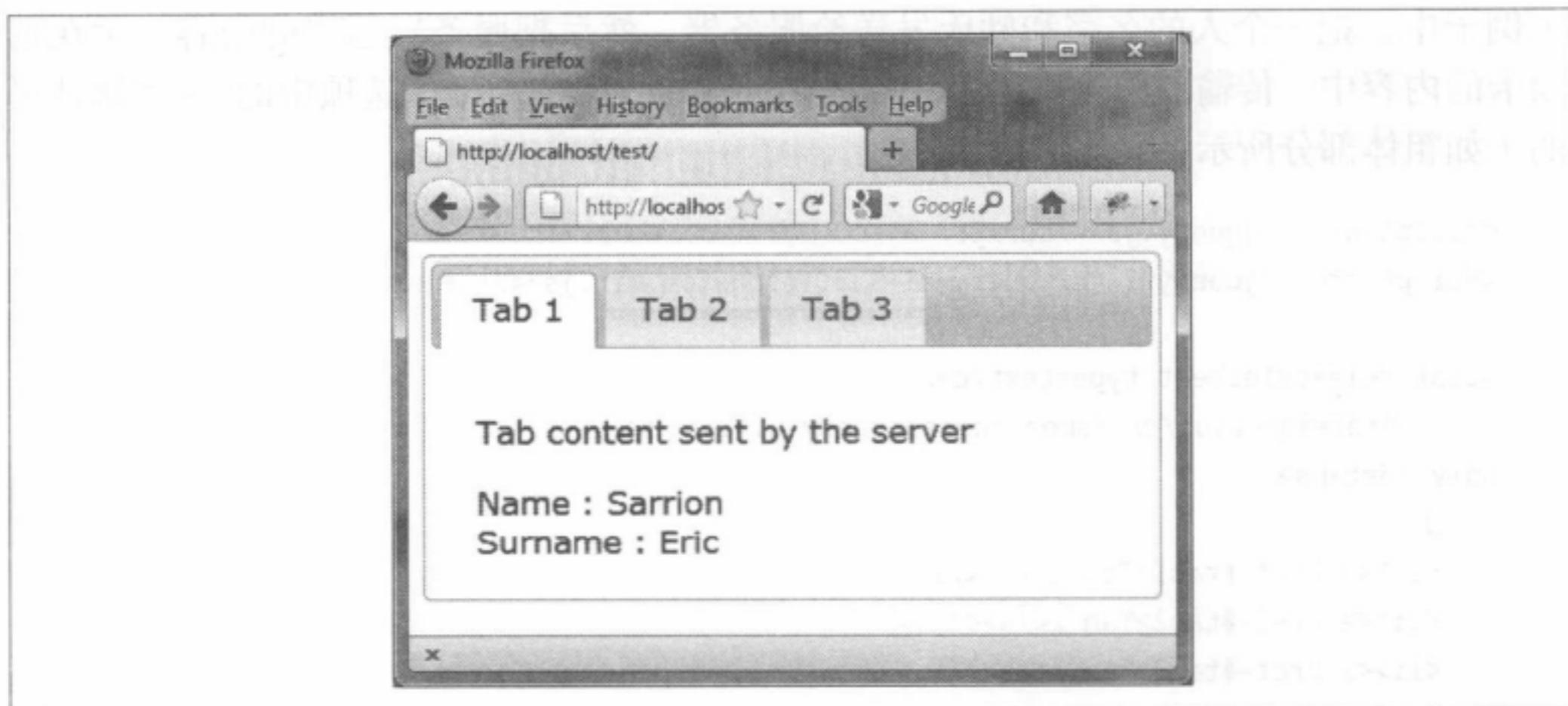


图 2-7 使用 Ajax 发送的选项卡内容

2.5.4 使用选项卡的 add 方法

我们想动态创建新的选项卡，并由 Ajax 动态初始化该选项卡的内容。add 事件（插入新选项卡时会触发）可以完成这件事，比如发起插入内容的 Ajax 请求（如粗体部分所示）：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>
<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
  </ul>
  <div id=tab1>Contents of first tab</div>
  <div id=tab2>Contents of the second tab</div>
  <div id=tab3>Contents of the third tab</div>
</div>

<script>
  $("#tabs").tabs ({
    fx : { opacity : "toggle" },
    add : function (event, tab)
    {
      $(tab.panel).load ("action.php");
    }
  });
</script>
```

```
        }
    }).tabs("add", "#tab4", "Tab 4");

```

"add"操作触发了 add 事件，该事件会更新新建的选项卡的内容：

```
<?
$	txt = "<p> Tab content sent by the server </p>";
$	txt = utf8_encode($txt);
echo($txt);
?>
```

结果如图 2-8 所示。



图 2-8 使用 add 事件

2.5.5 使用 tabsadd 事件

使用同样的例子，只是这次用 bind ()方法来管理事件。在添加选项卡时，jQuery UI 会触发 tabsadd 事件（如粗体部分所示）：

```
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id=tabs>
  <ul>
    <li><a href=#tab1>Tab 1</a></li>
    <li><a href=#tab2>Tab 2</a></li>
    <li><a href=#tab3>Tab 3</a></li>
```

```

</ul>
<div id=tab1>Contents of first tab</div>
<div id=tab2>Contents of the second tab</div>
<div id=tab3>Contents of the third tab</div>
</div>

<script>
$( "#tabs" ).tabs ({
  fx : { opacity : "toggle" }
}) .bind ("tabsadd", function (event, tab)
{
  $( tab.panel ).load ("action.php");
}) .tabs ("add", "#tab4", "Tab 4");

</script>

```

首先创建选项卡，然后监听 tabsadd 事件，最后向列表中插入一个选项卡。要小心，这个顺序是很重要的，否则没有任何效果。



折叠菜单

和选项卡一样，可以使用折叠菜单（accordion menu）来组织 HTML 页面上的信息。菜单中的内容是显示的还是隐藏的，取决于该菜单是否被选中。折叠菜单的意思是当显示一个菜单时，另外的菜单会隐藏，产生的动画看起来就像手风琴演奏一样。

3.1 折叠菜单的基本用法

假设我们想要编写一段 HTML 代码来显示图 3-1 所示的折叠菜单。它有 3 个菜单栏，每一个里面的内容都是不一样的。

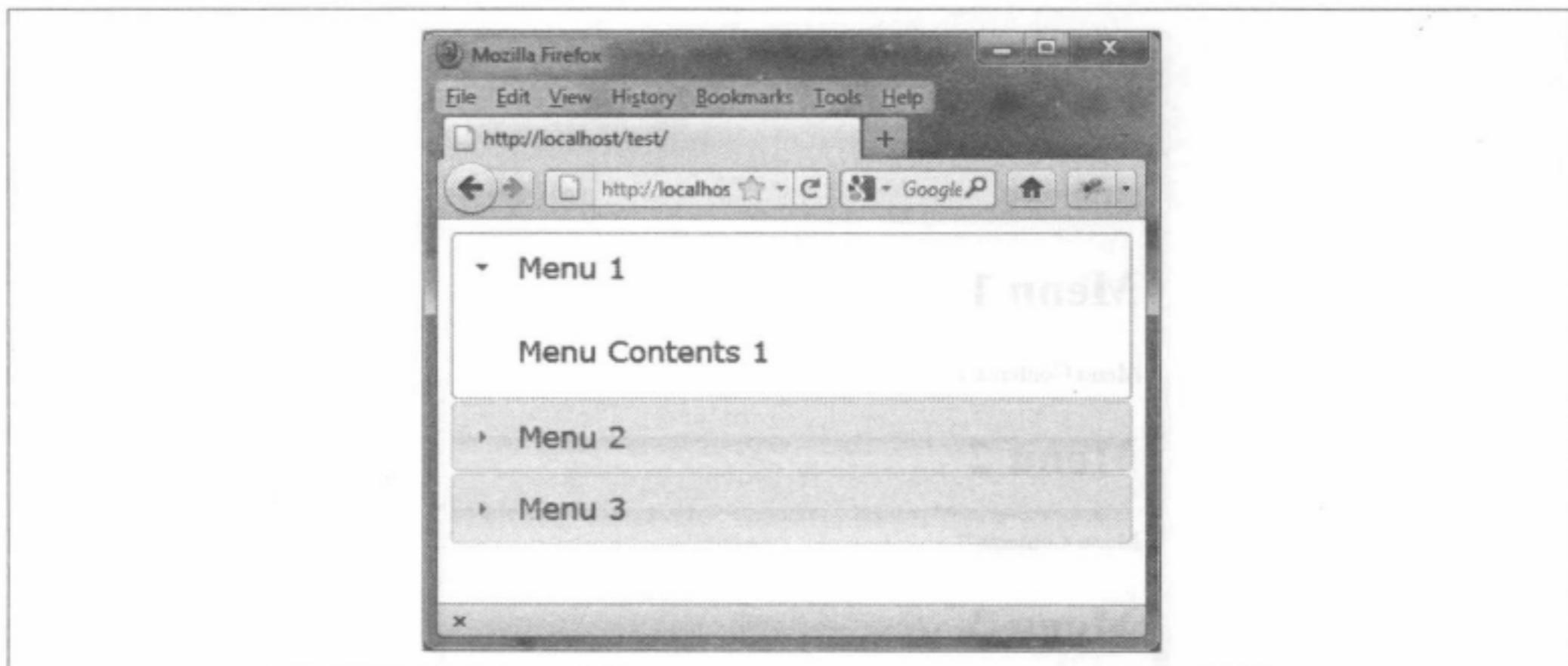


图 3-1 折叠菜单

按照 jQuery UI 的要求，需要挨个（标题和内容元素挨个重复）为每一个菜单编写以下内容：

- 一个包含整个折叠菜单的

元素；
- 一个作为菜单标题的元素，可以是

、、…、，它包含了一个链接，是菜单的名称（该链接不需要 href 属性，因为没用到）；
- 一个作为菜单内容的

元素。

```
<!DOCTYPE html>
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id="accordion">
<h1><a>Menu 1</a></h1>
<div>Menu Contents 1</div>
<h1><a>Menu 2</a></h1>
<div>Menu Contents 2</div>
<h1><a>Menu 3</a></h1>
<div>Menu Contents 3</div>
</div>

<script>
</script>
```

在浏览器中显示这个页面，它只是一张包含了菜单标题和内容的简单 HTML 页面，结果又不是我们预期的（如图 3-2 所示）。

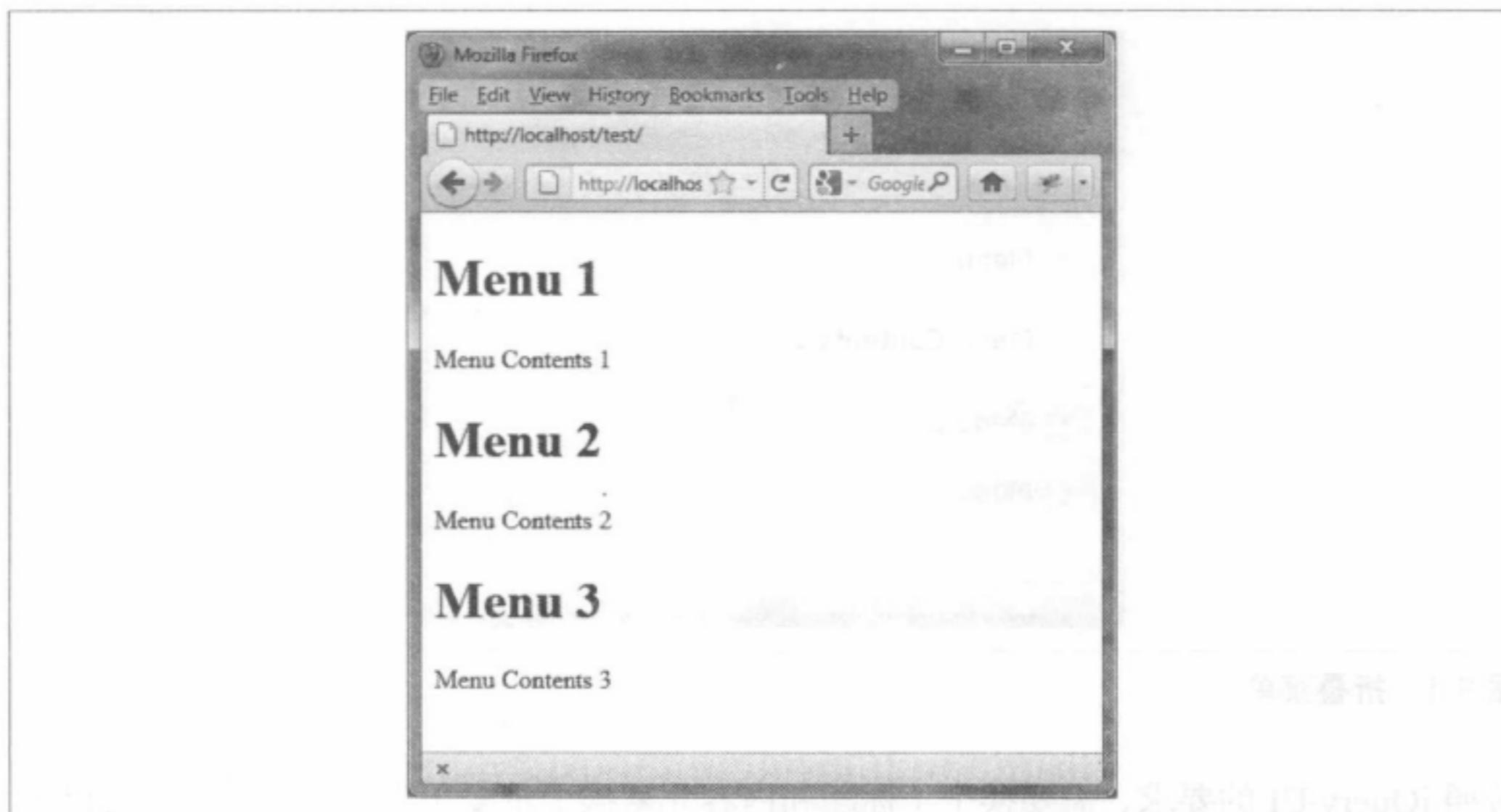


图 3-2 简单折叠菜单的 HTML 预览页面

正如在第 2 章中我们就知道的，要想得到预期的显示结果，在 HTML 页面中要按照 jQuery UI 的约定来做。为此，仅需调用 jQuery UI 的 accordion() 方法来管理全局的<div>即可。在<script>标签中添加下面这行代码：

```
<script>
    $("#accordion").accordion();
</script>
```

请注意 HTML 中的<!DOCTYPE html>语句。如果没有这条语句，则在 IE 浏览器中折叠菜单会被管理得很差劲。

3.2 格式化内容

使用 accordion() 方法会彻底改变已渲染页面中 HTML 元素的外观。实际上，该方法遍历了 HTML 代码并给相关的元素（这里是折叠菜单）增加了新的 CSS 类，赋予了它们适当的样式。如图 3.3 所示，在 accordion() 方法执行后，HTML 代码已经改变了。该代码是用 Firefox 的 Firebug 扩展查看的。



The screenshot shows the Firebug extension's HTML panel displaying the generated HTML code for an accordion menu. The code is as follows:

```
<!DOCTYPE html>
<html>
    <head>
        <body>
            <div id="accordion" class="ui-accordion ui-widget ui-helper-reset ui-accordion-icons" role="tablist">
                <h1 class="ui-accordion-header ui-helper-reset ui-state-default ui-state-active ui-corner-top" role="tab" aria-expanded="true" aria-selected="true" tabindex="0">
                    <span class="ui-icon ui-icon-triangle-1-s"></span>
                    <a tabindex="-1">Menu 1</a>
                </h1>
                <div class="ui-accordion-content ui-helper-reset ui-widget-content ui-corner-bottom ui-accordion-content-active" style="height: 22.8px;" role="tabpanel">Menu Contents 1</div>
                <h1 class="ui-accordion-header ui-helper-reset ui-state-default ui-corner-all" role="tab" aria-expanded="false" aria-selected="false" tabindex="-1">
                    <span class="ui-icon ui-icon-triangle-1-e"></span>
                    <a tabindex="-1">Menu 2</a>
                </h1>
                <div class="ui-accordion-content ui-helper-reset ui-widget-content ui-corner-bottom" style="height: 22.8px; display: none;" role="tabpanel">Menu Contents 2</div>
                <h1 class="ui-accordion-header ui-helper-reset ui-state-default ui-corner-all" role="tab" aria-expanded="false" aria-selected="false" tabindex="-1">
                    <span class="ui-icon ui-icon-triangle-1-e"></span>
                    <a tabindex="-1">Menu 3</a>
                </h1>
                <div class="ui-accordion-content ui-helper-reset ui-widget-content ui-corner-bottom" style="height: 22.8px; display: none;" role="tabpanel">Menu Contents 3</div>
            </div>
            <script>
            </script>
        </body>
    </html>
```

图 3-3 用 accordion() 方法生成的 HTML 代码

也可以修改元素的 CSS 类来自定义元素的显示样式。比如，如果修改了与<h1>元素关联的 ui-accordion-header CSS 类，将会得到新的菜单标题外观。同样地，如果修改了与<div>元素关联的 ui-accordion-content CSS 类，就会得到菜单内容的新外观了。

在 HTML 中添加一个<style>标签，来修改这些元素的样式（如粗体部分所示）：

```
<!DOCTYPE html>
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<style type=text/css>
  h1.ui-accordion-header {
    font-size : 10px;
  }
  div.ui-accordion-content {
    font-size : 15px;
    font-family : georgia;
    font-style : italic;
  }
</style>

<div id="accordion">
  <h1><a>Menu 1</a></h1>
  <div>Menu Contents 1</div>
  <h1><a>Menu 2</a></h1>
  <div>Menu Contents 2</div>
  <h1><a>Menu 3</a></h1>
  <div>Menu Contents 3</div>
</div>

<script>
  $("#accordion").accordion();
</script>
```

我们在引入 jQuery UI 的样式之后添加了<style>标签，除此之外，这段 HTML 代码和前面的是一模一样的。必须在 jQuery UI 样式的后面添加自己的样式，否则更改是会被“忽略”的。

现在菜单及其内容呈现了新的自定义风格（如图 3-4 所示）。

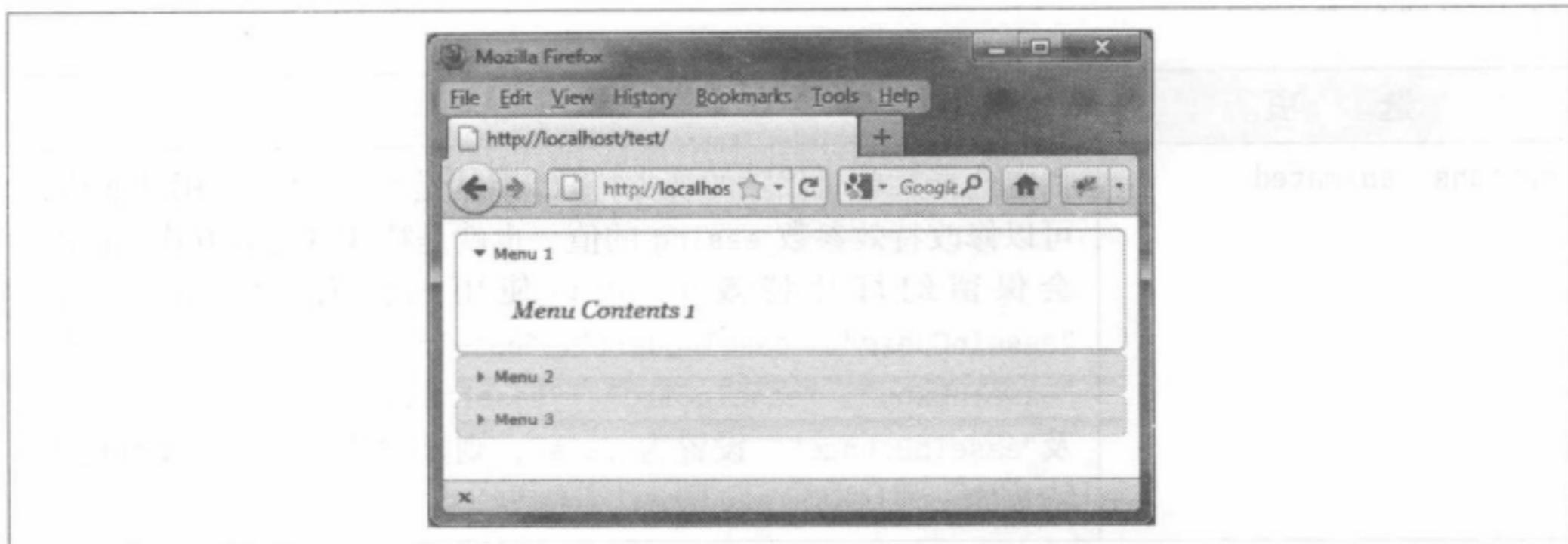


图 3-4 自定义的折叠菜单

3.3 accordion () 方法

accordion ()方法可以按以下两种形式使用：

- `$(selector, context).accordion (options)`
- `$(selector, context).accordion ("action", params)`

3.3.1 accordion (options) 方法

accordion (options)方法指定使用折叠菜单来管理 HTML 元素（及其内容）。options 参数是一个对象，用来指定菜单的外观及行为。这些选项与菜单的行为、菜单内容的高度或者菜单的事件有关。

1. 管理折叠菜单的选项

表 3-1 描述了管理折叠菜单行为的选项。

表 3-1

管理菜单行为的选项

选 项	功 能
<code>options.collapsible</code>	当设置为 <code>true</code> 时，允许用户通过单击来关闭菜单。当设置为 <code>false</code> 时（默认），在打开的菜单上单击不会关闭该菜单
<code>options.active</code>	指定第一次访问页面时被打开菜单的索引。默认是 0（第一个菜单）。若要指定在启动时不打开菜单，使用 <code>false</code>
<code>options.event</code>	让用户选择新菜单的事件的名称（默认是"click"）。比如，如果指定为"mouseover"，则用户移动鼠标到菜单上就能选中该菜单

续表

选 项	功 能
options.animated	指定选择菜单时伴随的视觉特效。默认是"slide"。 ^① 要用其他值，可以修改特效参数 easing 的值，也就是特效进行的方式（同时会保留幻灯片特效）。可以使用的值有 "easeInQuad"、"easeInCubic"、"easeInQuart"、"easeInQuint"、"easeInSine"、"easeInExpo"、"easeInCirc"、"easeInElastic"、"easeInBack" 及"easeInBounce"。设置为 false，则显示菜单内容是没有过渡特效的

单击垂钓的飞宝店 1.6.1

2. 管理菜单内容的高度

默认情况下，菜单会自动调整高度，以适应内容的高度。也可以手动设定高度。表 3-2 列出了管理菜单内容高度的选项。

表 3-2 管理菜单高度的选项

选 项	功 能
options.autoHeight	当设置为 true 时（默认），最高内容的高度会被应用到所有其他的菜单内容上。当设置为 false 时，每个菜单的高度等于其内容的实际高度，所以每个菜单都可能会有不同的高度
options.fillSpace	当设置为 true 时，所有的菜单内容拥有和全局<div>元素的父元素一样的高度和宽度。默认为 false

3. 管理与菜单相关的事件

也有管理选择菜单的方法。这些方法接收 event 事件参数，随后是描述与该事件相关的 menus 对象（打开的菜单及关闭的菜单）。这个 menus 对象包含了以下属性。

- oldHeader，正在关闭的菜单的 jQuery 类对象。
- oldContent，正在关闭的菜单内容的 jQuery 类对象。
- newHeader，正在打开的菜单的 jQuery 类对象。
- newContent，正在打开的菜单内容的 jQuery 类对象。

表 3-3 描述了管理菜单事件的选项。

① 即幻灯片特效。——译者注

表 3-3

管理菜单事件的选项

选 项	功 能
options.change	当选择菜单时（要么手动调用，要么调用 accordion ("activate") 方法），在动画发生后（选择的菜单已经打开且先前打开的菜单已经关闭）会调用 change (event, menus) 方法
options.changestart	当选择菜单时（要么手动调用，要么调用 accordion ("activate") 方法），在动画发生前（应该打开的菜单尚未打开，应该关闭的菜单也尚未关闭）会调用 changestart (event, menus) 方法

3.3.2 accordion ("action", params) 方法

accordion ("action", params) 方法允许操作菜单，比如选中或取消选中菜单。第一个参数是一个字符串，指定是什么操作（例如，“activate”表示选中新菜单），随后的是和该操作有关的参数（例如，菜单的索引）。表 3-4 列出了该方法的一些操作。

表 3-4

accordion ("action", params) 方法的操作

方 法	功 能
accordion ("activate", index)	选中指定的菜单
accordion ("disable")	禁用所有菜单。任何单击都是无效的
accordion ("enable")	重新激活所有菜单。单击又有效了
accordion ("destroy")	移除菜单管理。菜单再次成了没有 CSS 类和事件管理的简单 HTML

3.4 在折叠菜单上使用 bind () 管理事件

除了在 accordion (options) 方法的选项中使用事件方法外，jQuery UI 还支持使用 bind () 方法来管理这些事件。jQuery UI 为折叠菜单创建了不同的事件，如表 3-5 所示。

表 3-5

管理 accordion 菜单的 jQuery UI 事件

事 件	功 能
accordionchange	同 options.change（见表 3-3）
accordionchangestart	同 options.changestart（见表 3-3）

有了这些事件，就可以利用 bind (eventName, callback) 提供的回调方法，执行处理一些事情。

3.5 使用折叠菜单的示例

在本节中，我们将在 UI 中整合折叠菜单的用法。

3.5.1 打开任意菜单

当创建折叠菜单时，默认第一个菜单（索引为 0）是打开的。通过配置页面中的选项，换成打开第二个菜单（索引为 1）。可以使用 accordion ("activate") 来完成：

```
<!DOCTYPE html>
<script src = jquery.js></script>
<script src = jqueryui/js/jquery-ui-1.8.16.custom.min.js></script>

<link rel=stylesheet type=text/css
      href=jqueryui/css/smoothness/jquery-ui-1.8.16.custom.css />

<div id="accordion">
  <h1><a>Menu 1</a></h1>
  <div>Menu Contents 1</div>
  <h1><a>Menu 2</a></h1>
  <div>Menu Contents 2</div>
  <h1><a>Menu 3</a></h1>
  <div>Menu Contents 3</div>
</div>

<script>
  $("#accordion").accordion () .accordion ("activate", 1);
</script>
```

第一个 accordion () 方法创建了折叠菜单，是必需的。第二个 accordion () 方法打开了索引为 1 的菜单。

当启动应用时第二个菜单是打开的，如图 3-5 所示。

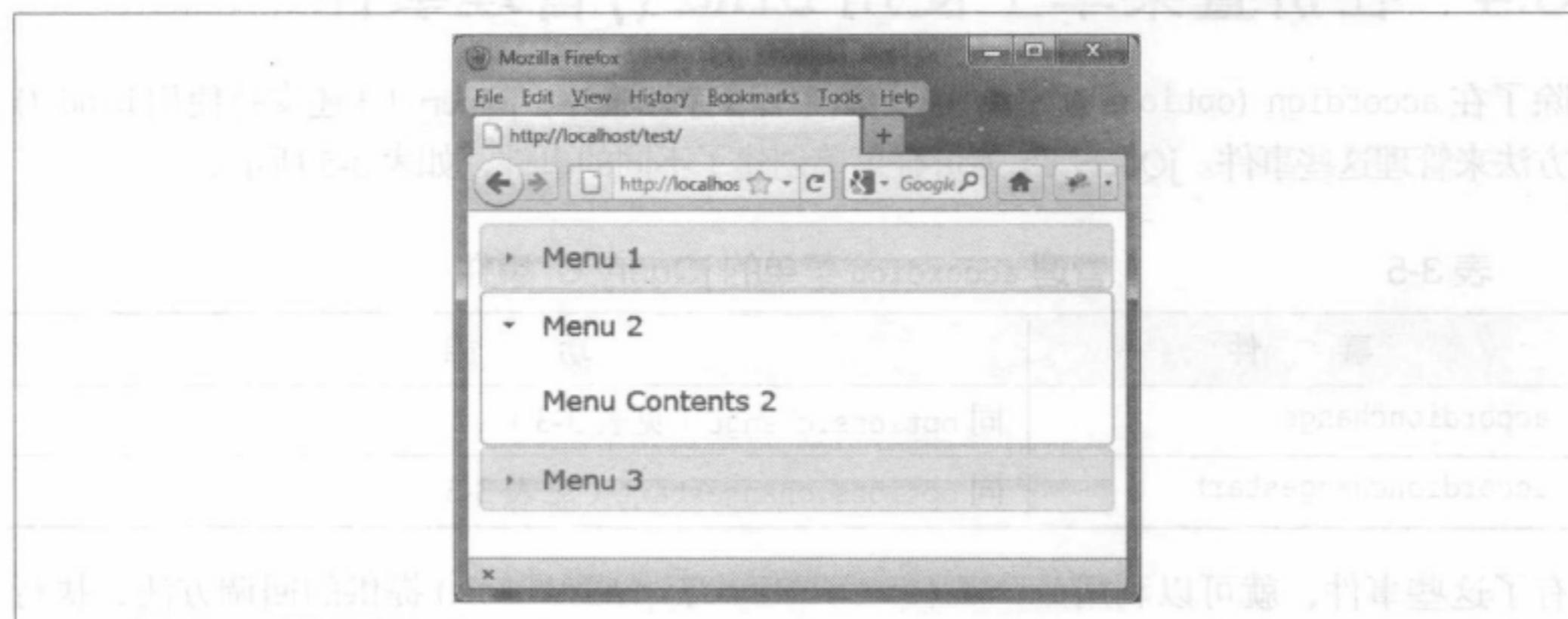


图 3-5 用"activate"操作打开第二个菜单

完整内容下载地址：<http://www.400gb.com/file/83652651>