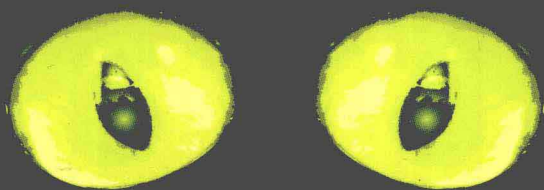


■ 快速、简单、免费和开源，改变你编写JavaScript的方式



Write Less,
Do More

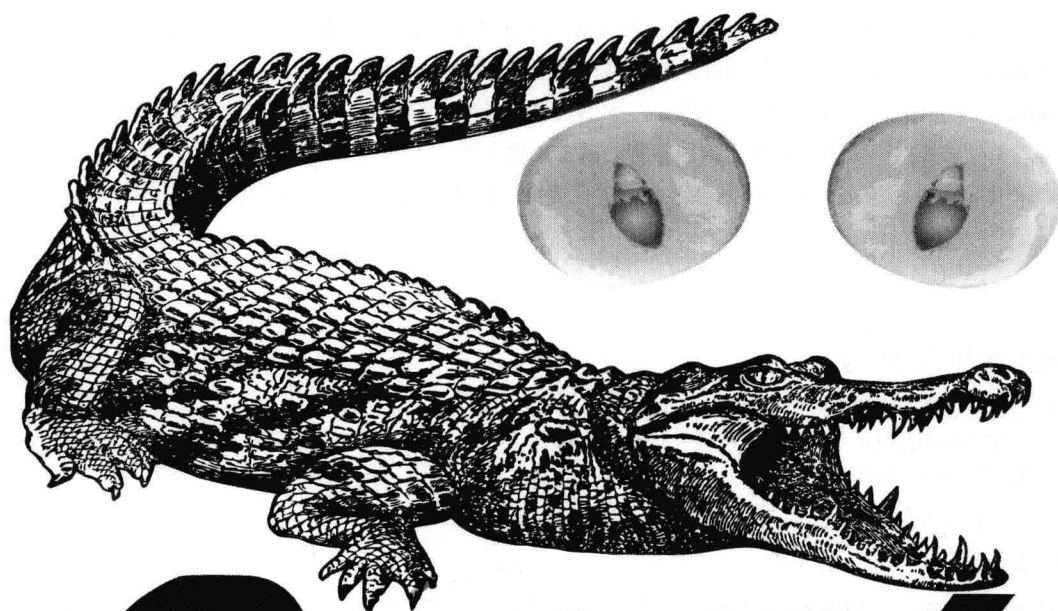
jQuery

开发完全技术宝典

一个优秀的“JavaScript框架”，宗旨是“写更少的代码，做更多的事情”
jQuery development of complete technical collection

朱育发 岳阳 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



jQuery **开发完全技术宝典**

朱育发 岳阳 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书从零起步,理论结合实践,详细讲解了 jQuery 基础知识和开发技巧,内容包括 jQuery 基础知识、设计思路、使用选择器、DOM 文档操作、事件处理、动画设计、Ajax 异步通信、插件扩展、辅助工具等,同时针对当前网页开发中几个主要技术板块的开发进行实践演练,最后通过一个综合案例帮助读者快速把握 jQuery 开发的完整过程。

本书在详细讲解 jQuery 基础知识和技巧的同时,重点演示了各种 Web 应用开发的实践途径。由于 jQuery 是 JavaScript 语言的高级封装,因此本书在讲解 jQuery 的同时,还介绍了如何混合使用 jQuery 和 JavaScript 进行开发。本书是广大网页设计爱好者和前端开发人员必读的参考书目。全书理论结合实践,大量生动的示例能够帮助读者快速上手。

本书适合 Web 开发人员阅读和参考,同时也适合广大网页制作和设计的学生阅读和学习。

图书在版编目(CIP)数据

jQuery 开发完全技术宝典/朱育发,岳阳编著. --
北京:中国铁道出版社,2012.2
ISBN 978-7-113-13845-5

I. ①j… II. ①朱…② 岳…III. ①JAVA 语言—程序
设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 230623 号

书 名: jQuery 开发完全技术宝典
作 者: 朱育发 岳 阳 编著

策 划: 武文斌
责任编辑: 苏 茜
封面设计: 张 丽

读者热线电话: 010-63560056
特邀编辑: 赵树刚
责任印制: 李 佳

出版发行: 中国铁道出版社(北京市西城区右安门西街 8 号 邮政编码: 100054)

印 刷: 北京新魏印刷厂

版 次: 2012 年 2 月第 1 版 2012 年 2 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 36 字数: 849 千

书 号: ISBN 978-7-113-13845-5

定 价: 65.00 元(附赠光盘)

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社发行部联系调换。

前言

jQuery 是一个非常优秀的 JavaScript 框架，由美国人 John Resig 创建。

jQuery 优势

与 Prototype、YUI、Mootools 等众多的 JS 框架相比，它剑走偏锋，从 Web 开发实用的角度出发，摒弃了其他框架中一些不实用的东西，为开发者提供了短小精悍的类库。

其短小精悍，使用简单方便，性能高效，能极大地提高开发效率，是开发 Web 应用最佳辅助工具之一。因此大部分开发者抛弃 Prototype 而选择 jQuery 来进行 Web 开发。

jQuery 的宗旨是“Write Less, Do More”，即写更少的代码，做更多的事情。它是轻量级的 JS 框架，这是其他 JS 框架所不及的，它兼容 CSS 3，还兼容各种浏览器（IE 6.0+，FF 1.5+，Safari 2.0+，Opera 9.0+）。

jQuery 使用户能更方便地处理 HTML DOM、Events，实现动画效果，为网站提供 Ajax 交互。jQuery 还有一个比较大的优势，即它的文档说明很全，而且各种应用也说得详细，同时还有许多成熟的插件可供选择。jQuery 能够使用户的网页保持代码内容的分离，也就是说不用在页面中插入一堆 JavaScript 调用命令，只需要定义 ID 即可。

jQuery 经过多个版本的磨炼，特别是新的独立开发的 Sizzle 选择器引擎，使得 jQuery 匹配元素的速度已经达到了新的高度，毫不夸张地说，它已是目前业界第一。

人们都有一种习惯，用惯了 Java 作为开发语言，就不习惯 .NET 的编程环境。同样，如果你习惯了 CSS 的用法，以及其选择网页标签的方式，再来学习 jQuery 的选择器，就会倍感亲切，甚至入迷。

jQuery 完全仿制了 CSS 选择器的设计思路和用法，让广大的初学者和整日为获取网页节点而搔首的开发人员快速上手。很多读者和开发人员接触 jQuery 之后，身不由己地迷恋于 jQuery 的优雅和便捷。

关于本书

本书从初学者的角度进行选材和编写，在编写过程中考虑到读者的实际技术水平和战术能力，注重基础知识和实战应用的结合，主要讲解 jQuery 框架的一般用法、扩展应用及其实战演练。全书共分为 18 章，简单介绍如下。

第 1 章 jQuery 开始。本章重点介绍了 JavaScript 及其框架发展概述，如何使用 jQuery，如何编写自己的第一个示例，以及 jQuery 框架的基本特性。

第 2 章 jQuery 框架实现：选择和操作。本章重点介绍了 jQuery 框架的设计思路，详细分解了选择器接口和选择器引擎 Sizzle 的工作机制。

第 3 章使用选择器：选择 DOM 中的特定元素。本章重点讲解 jQuery 选择器的用法。

第 4 章 DOM 开发详解。本章重点讲解 HTML 文档对象模型的基础知识，以及如何使用 JavaScript 操作 HTML 文档。

第 5 章使用 jQuery 操作 DOM。本章重点讲解 jQuery 如何操作文档结构，如何控制页面 CSS 的样式，从而更加深刻地理解 jQuery 的操作机制。

第 6 章事件模型。本章重点讲解 DOM 事件机制，并介绍不同浏览器对 DOM 事件的支持情

况，以及如何使用 JavaScript 操作事件。

第 7 章管理事件。本章重点讲解 jQuery 如何封装 JavaScript 的事件处理机制，并介绍了如何使用 JavaScript 来定义 jQuery 的事件处理方法。

第 8 章操作 CSS。本章重点讲解如何使用 jQuery 设计动画，并就一些常见动画形式进行案例讲解。

第 9 章使用 Ajax。本章讲解 Ajax 通信原理，并介绍 jQuery 实现异步通信的方法。

第 10 章使用插件。本章重点讲解 jQuery 插件定义的常用方法，并结合几个案例进行演示。

第 11 章使用辅助工具。本章重点讲解 jQuery 辅助工具的使用，帮助读者更加全面地认识 jQuery 的功能。

第 12 章浏览器开发。本章重点介绍如何使用 jQuery 开发设计浏览器的相关应用。

第 13 章导航开发。本章介绍导航器的各种形式的开发实战。

第 14 章表格开发。本章重点介绍网页表格的控制和设计。

第 15 章表单开发。本章重点介绍网页表单的控制和设计。

第 16 章异步通信开发。本章重点介绍异步通信技术的原理和实现途径，帮助读者更深入地理解 Ajax 技术的实现方法。

第 17 章数据存储开发。本章重点介绍如何使用 JavaScript 读写 XML 和 JSON 外部数据。

第 18 章使用 jQuery 打造个性网站。本章重点通过一个动态演示网站的架设帮助读者认识 jQuery 在网站开发中的综合应用。

读者对象

本书适合网页制作的初学者、广大网页设计师和前端技术人员阅读参考。读者在阅读本书之前，应该初步了解 HTML、CSS 和 JavaScript 语言基础知识。

本书约定

- 本书代码以灰色背景进行显示，以方便读者阅读。读者在练习时，应该把这些代码输入到网页，在输入时应该注意格式和规则。例如：

```
<script src="jquery-1.4.2.js" type="text/javascript" ></script>
<script type="text/javascript" >
$(function(){
    $("div").after("<p>文本 2</p>");
})
</script>

<div>
    <p>文本 1</p>
</div>
```

读者应该按如下代码格式输入上面示例代码，然后另存为 HTML 网页文件，在浏览器中预览即可。

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="jquery-1.4.2.js" type="text/javascript"></script>
<script type="text/javascript">
$(function(){
    $("div").after("<p>文本 2</p>");
})
</script>
</head>
<body>
<div>
    <p>文本 1</p>
</div>
</body>
</html>

```

- 本书以 jQuery 1.4.2 版本为基础进行介绍和演示，程序可以兼容 jQuery 3.0 及其以上任何版本。
- 在默认情况下，jQuery 1.4.2 库文件都会自动导入文档，如果没有特别说明，我们会在示例中省略该行命令。jquery-1.4.2.js 存放在 images 文件夹或者 jquery 文件夹中。

```

<script src="images/jquery-1.4.2.js" type="text/javascript"></script>

```

- 在默认情况下，示例会使用 jQuery 的别名“\$”来表示 jQuery 名字空间，同时直接把调用的函数放在\$()函数中，实际上它是\$(“document”).ready()方法的简写，相当于 JavaScript 中的 window.onload =function(){ }事件处理函数。例如：

```

<script type="text/javascript">
$(function(){
    $("div").after("<p>文本</p>");
})
</script>

```

实际上，上面示例代码应该是下面形式：

```

<script type="text/javascript">
$(“document”).ready (function){
    $("div").after("<p>文本</p>");
})
</script>

```

它相当于 JavaScript 中下面的示例代码：


```
<script type="text/javascript">
window.onload =function(){
    $("div").after("<p>文本</p>");
}
</script>
```

附赠光盘说明

1. 光盘内容

(1) Readme.txt 为光盘说明文件。

(2) “书稿实例”文件夹下存放的是与书中章节对应的操作实例文件和源代码。

(3) “电子教程”文件夹下存放的是 JavaScript 基础教程。如果读者对于 JavaScript 语言不是很熟悉,建议先阅读这些电子教程,这样可以帮助您快速入门。

(4) “参考手册”文件夹下存放的是各种技术参考手册,包括 JavaScript 参考手册、jQuery 参考手册、HTML 参考手册、CSS 参考手册、Ajax 参考手册。这样在学习和开发过程中,可以随时查询相关技术资料,方便学习和实践。

2. 使用注意事项

(1) 建议将光盘复制到计算机硬盘上再进行操作。

(2) 如果实例文件中的网页无法预览,请直接使用记事本打开来学习和研究其中的代码。

(3) 电子教程是 PDF 格式,需要在电脑上安装 Adobe Reader 阅读器软件,读者可以访问 <http://www.adobe.com/cn/下载>。

3. 光盘运行环境

本光盘可以运行于 Windows 98/2000/XP/Server 2003/Vista/Server 2008 及 Windows 7 的操作系统下。

注意:本书配套光盘中的文件仅供个人学习和练习时使用,未经许可不得用于任何商业行为。

关于作者

本书由朱育发、岳阳编写,参与资料整理及编写的还有常才英、袁祚寿、袁衍明、张敏、袁江、田明学、唐荣华、毛荣辉、卢敬孝、刘玉凤、李坤伟、旷晓军、陈万林、陈锐、钱佩林、苏敬波、冉东林、杨龙贵、张炜、王慧明、涂怀清、卢国才、苏恢定、司成向、胡体清、陈宗亮、徐清银、周秀成、颜昌学、王幼平、冉原洲、李经键、胡厚成等,在此对大家的辛勤工作表示衷心的感谢。

由于水平有限,书中难免会有疏漏之处,恳请广大读者提出宝贵意见,并将个人的意见、建议或问题发送到 wwb_beijing@163.com,以便我们与您进行交流。

编 者

2011 年 11 月

目 录

第 1 章 jQuery 开始 1

- 1.1 jQuery: 改变 JavaScript 代码编写方式 1
- 1.2 jQuery 项目及框架 1
 - 1.2.1 jQuery 项目 4 个部分 1
 - 1.2.2 jQuery 框架优势 1
- 1.3 使用 jQuery 前的准备工作 3
 - 1.3.1 安装 jQuery 库 3
 - 1.3.2 导入 jQuery 库 4
 - 1.3.3 编写 jQuery 代码 4
 - 1.3.4 解析 jQuery 重点代码 4
- 1.4 jQuery 必知的重要特征 5
 - 1.4.1 jQuery 代码和 JavaScript 代码的区别 5
 - 1.4.2 jQuery 对象和 DOM 对象的区别 5
 - 1.4.3 把 jQuery 对象转换为 DOM 对象 7
 - 1.4.4 把 DOM 对象转换为 jQuery 对象 8
 - 1.4.5 jQuery 构造器 8
 - 1.4.6 jQuery 语法 9
 - 1.4.7 链式语法注意问题 11
 - 1.4.8 jQuery 选择器 11
 - 1.4.9 jQuery 事件 13
 - 1.4.10 jQuery 扩展 14
 - 1.4.11 jQuery 类数组: 定义类数组 14
 - 1.4.12 jQuery 类数组操作: 元素定位、查找、复制和删除.. 17

第 2 章 jQuery 框架实现: 选择和操作 22

- 2.1 jQuery 框架解决: 选择和操作问题 22
- 2.2 jQuery 框架核心构建过程 23
 - 2.2.1 定义 jQuery 类 23
 - 2.2.2 扩展类原型 23
 - 2.2.3 添加原型属性和原型方法 24

- 2.2.4 调用原型属性和原型方法 24
- 2.2.5 返回 jQuery 实例 25
- 2.2.6 解决作用域问题 25
- 2.2.7 解决跨域问题 27
- 2.3 解析 jQuery 对象 28
- 2.4 操作 jQuery 对象中的 DOM 元素 29
- 2.5 jQuery 扩展 31
 - 2.5.1 通过 extend() 函数实现功能扩展 31
 - 2.5.2 应用 extend() 函数的好处 31
 - 2.5.3 通过 extend() 函数实现对象合并功能 32
- 2.6 jQuery 参数 33
- 2.7 jQuery 名字空间 34
 - 2.7.1 封闭问题 35
 - 2.7.2 为什么要用到匿名函数 ... 35
 - 2.7.3 JavaScript 函数最核心概念: 闭包 36
- 2.8 jQuery 对象是一个选择器 36
- 2.9 jQuery 如何分辨不同类型的参数 37
 - 2.9.1 初始化处理 37
 - 2.9.2 类型处理 38
 - 2.9.3 选择器 39
- 2.10 选择器引擎 Sizzle 工作原理 40
 - 2.10.1 回顾 CSS 选择器 40
 - 2.10.2 拆分选择器的思路 40
 - 2.10.3 不同版本的 jQuery 选择器引擎区别 41
 - 2.10.4 选择器与过滤器 41
 - 2.10.5 判断选择符适合类型 41
 - 2.10.6 jQuery 选择器逻辑流程 42

第 3 章 使用选择器: 选择 DOM 中的特定元素 48

- 3.1 jQuery 选择器的优势 48
- 3.2 基本选择器 49
 - 3.2.1 ID 选择器 49

3.2.2	类型选择器	49
3.2.3	类选择器	50
3.2.4	通配选择器	51
3.2.5	分组选择器	52
3.3	层级选择器	54
3.3.1	关系选择器	54
3.3.2	子元素选择器	55
3.4	过滤选择器	56
3.4.1	定位过滤器	57
3.4.2	内容过滤器	59
3.4.3	可见过滤器	60
3.5	属性选择器	60
3.5.1	七类属性选择器	60
3.5.2	使用属性选择器示例	61
3.6	表单选择器	62
3.6.1	基本表单选择器	62
3.6.2	高级表单选择器	63
3.7	使用过滤器	65

第4章 DOM 开发详解 67

4.1	认识 DOM	67
4.1.1	DOM 规范不同版本	67
4.1.2	DOM 兼容性	69
4.1.3	DOM 分解	71
4.1.4	DOM 结点	72
4.2	使用 JavaScript 操作 DOM	73
4.2.1	HTML DOM 加载和处理	74
4.2.2	DOM 遍历指针	76
4.2.3	文档遍历缺陷及解决方法	78
4.2.4	文档遍历演示	79
4.2.5	DOM 指针函数	80
4.2.6	DOM 原型方法	82
4.2.7	读写文档内容	84
4.3	使用 JavaScript 操作结点	86
4.3.1	获取结点	86
4.3.2	创建结点	88
4.3.3	复制结点	90
4.3.4	插入结点	91
4.3.5	删除结点	93
4.3.6	替换结点	95
4.4	使用 JavaScript 操作属性	96

4.4.1	获取结点属性	97
4.4.2	设置结点属性	98
4.4.3	删除结点属性	99
4.4.4	存取 JavaScript 属性	99
4.5	使用 JavaScript 操作文本内容	100
4.5.1	获取元素包含的文本	100
4.5.2	获取元素包含的 HTML 信息	101

第5章 使用 jQuery 操作 DOM ... 102

5.1	使用 jQuery 创建结点	102
5.1.1	创建元素	102
5.1.2	创建文本	103
5.1.3	创建属性	103
5.2	使用 jQuery 插入元素	104
5.2.1	在结点内部插入内容	104
5.2.2	在结点外部插入内容	105
5.2.3	插入内容的破坏性	106
5.3	使用 jQuery 删除元素	107
5.4	使用 jQuery 复制元素	108
5.5	使用 jQuery 替换元素	109
5.6	使用 jQuery 包裹元素	110
5.7	使用 jQuery 操作属性	111
5.7.1	设置属性	111
5.7.2	获取属性	111
5.7.3	删除属性	111
5.8	使用 jQuery 操作类样式	112
5.8.1	追加样式	112
5.8.2	移出样式	113
5.8.3	切换样式	114
5.8.4	判断样式	115
5.9	使用 jQuery 操作 HTML、文本和值	115
5.9.1	读写 HTML 字符串	115
5.9.2	读写文本内容	116
5.9.3	读写表单值	117
5.10	使用 jQuery 操作样式表	118
5.10.1	通用 CSS 样式读写方法	118
5.10.2	绝对偏移位置	120
5.10.3	相对偏移位置	121
5.10.4	元素的宽和高	122
5.11	使用 jQuery 遍历文档树	124

第 6 章 事件模型	126
6.1 认识事件	126
6.1.1 什么是事件	126
6.1.2 事件驱动模型	127
6.1.3 事件监听函数	128
6.1.4 异步回调	128
6.2 事件流	129
6.2.1 事件传播过程	129
6.2.2 冒泡型事件流	130
6.2.3 捕获型事件流	132
6.2.4 DOM 标准事件流	133
6.3 DOM 0 级事件模型	134
6.3.1 事件处理方式	134
6.3.2 定义事件	135
6.3.3 调用事件处理函数	136
6.3.4 事件处理函数返回值	137
6.3.5 事件处理函数参数	138
6.3.6 事件中的 this 关键字	139
6.3.7 事件处理函数作用域	140
6.4 DOM 2 级事件模型	141
6.4.1 认识 DOM 2 级事件模型	142
6.4.2 事件模块与接口	142
6.4.3 DOM 2 级事件模型工作机制	143
6.4.4 Event 接口	143
6.4.5 UIEvent 接口	143
6.4.6 MouseEvent 与 MutationEvent 接口	144
6.4.7 绑定事件	144
6.4.8 销毁事件	145
6.4.9 注册多个事件	146
6.4.10 兼容 0 级事件模型的处理方法	146
6.4.11 兼容 IE 的事件处理方法	147
6.5 IE 事件模型	147
6.5.1 认识 IE 事件模型	148
6.5.2 绑定和销毁事件	148
6.5.3 设计可兼容的 Event 对象	150
6.6 事件类型	150
6.6.1 DOM 标准定义下的几组事件	150

6.6.2 HTML 4.0 定义的事件类型	151
6.6.3 原始事件和语义事件	152
6.7 鼠标事件	153
6.7.1 鼠标单击事件	154
6.7.2 鼠标移动事件	154
6.7.3 鼠标经过事件	156
6.7.4 鼠标来源	157
6.7.5 鼠标指针定位	158
6.7.6 封装鼠标指针定位	159
6.7.7 元素内定位鼠标指针	160
6.7.8 鼠标按键	161
6.7.9 鼠标响应顺序	161
6.8 键盘事件	162
6.8.1 键盘事件属性	162
6.8.2 ctrlKey 和 shiftKey 属性	163
6.8.3 keyCode 和 charCode 属性	163
6.8.4 键盘响应顺序	165
6.9 页面事件	166
6.9.1 load 事件类型	166
6.9.2 DOMContentLoaded 事件类型	169
6.9.3 unload 事件类型	171
6.9.4 beforeunload 事件类型	171
6.9.5 resize 事件类型	171
6.9.6 scroll 事件类型	172
6.9.7 error 事件类型	173
6.10 UI 事件	173
6.10.1 focus 事件类型	173
6.10.2 blur 事件类型	173
6.11 表单事件	174
6.11.1 select 事件类型	174
6.11.2 change 事件类型	174
6.11.3 submit 事件类型	175
6.11.4 reset 事件类型	176

第 7 章 管理事件 178

7.1 jQuery 事件模型	178
7.1.1 绑定事件: 使用 bind()	178
7.1.2 绑定事件: 快速绑定	179

7.1.3	绑定事件: 使用 one()	
	绑定	180
7.1.4	注销事件	180
7.1.5	Event 对象	182
7.1.6	事件触发	182
7.1.7	事件切换: 使用 toggle()	184
7.1.8	事件切换: 使用 hover()	185
7.1.9	事件委派	187
7.1.10	事件命名空间	189
7.1.11	绑定多个事件	190
7.1.12	自定义事件	192
7.2	jQuery 页面初始化	192
7.2.1	使用 ready() 方法	192
7.2.2	ready 触发时机	194
7.2.3	ready 重复调用	194
7.2.4	自定义 addLoadEvent()	195

第 8 章 操作 CSS.....197

8.1	CSS 2 级脚本模型	197
8.1.1	CSSStyleSheet 接口	197
8.1.2	CSSStyleRule 接口	198
8.1.3	CSSStyleDeclaration 接口	199
8.2	CSS 0 级脚本模型	199
8.2.1	Style 对象	200
8.2.2	读写行内样式	201
8.2.3	styleSheets 集合	204
8.2.4	访问 CSS 样式表	206
8.2.5	读写 CSS 规则	207
8.2.6	访问最终样式	210
8.2.7	修改 CSS 样式方法总结	211
8.3	设计显隐动态效果	213
8.3.1	简单的显隐效果	213
8.3.2	折叠效果	215
8.3.3	树形结构	216
8.3.4	Tab 选项卡	218
8.3.5	显隐切换	219
8.4	设计滑动动态效果	220
8.4.1	滑动显隐效果	220
8.4.2	滑动显隐切换	221

8.5	设计渐进动态效果	222
8.6	设计自定义动画	224
8.6.1	自定义动画	224
8.6.2	停止动画	227
8.6.3	关闭动画	227

第 9 章 使用 Ajax.....229

9.1	jQuery 对 Ajax 技术进行封装	229
9.1.1	为什么要采用 Ajax 技术	229
9.1.2	安装虚拟服务器	230
9.1.3	创建一个简单的实例	230
9.2	使用 GET 请求	232
9.3	使用 POST 请求	234
9.4	使用 ajax() 请求	235
9.5	跟踪状态	237
9.6	载入文件	238
9.7	预设 Ajax 选项	240
9.8	预处理字符串	242

第 10 章 使用插件245

10.1	如何开发插件	245
10.1.1	jQuery 插件的三种类型	245
10.1.2	jQuery 开发团队制定的插件规范	246
10.1.3	jQuery 插件工作原理	247
10.1.4	添加全局函数	249
10.1.5	添加 jQuery 方法	250
10.1.6	如何准确指定当前元素对象	251
10.1.7	使用 extend () 添加方法	253
10.1.8	jQuery 选择器的设计思路	254
10.1.9	创建自定义的选择器	255
10.1.10	优化默认选择器	256
10.1.11	封装插件	259
10.1.12	开放公共参数	261
10.1.13	开放插件功能	262
10.1.14	保留插件隐私	264
10.1.15	非破坏性实现	265
10.1.16	添加事件日志	267
10.2	创建插件	272
10.2.1	简化式插件设计	272

10.2.2	定宽输出插件设计	274
10.2.3	Tab 选项卡插件设计	276
10.3	jQuery UI 插件设计	282
10.3.1	使用插件	283
10.3.2	认识 UI 插件	285
10.3.3	拖曳元素大小	285
10.3.4	日期面板	287

第 11 章 使用辅助工具..... 289

11.1	浏览器检测	289
11.1.1	检测浏览器类型	289
11.1.2	检测浏览器版本号	291
11.1.3	检测浏览器渲染方式	291
11.1.4	浏览器综合测试	291
11.2	管理页面中的多个库	292
11.2.1	jQuery 库兼容性	292
11.2.2	如何在同一个页面中联合 使用多个库	294
11.3	使用 jQuery 操作 JavaScript	295
11.3.1	修剪字符串	295
11.3.2	序列化字符串	295
11.3.3	检测数组	296
11.3.4	遍历数组或集合对象	296
11.3.5	转换数组	298
11.3.6	过滤数组	299
11.3.7	映射数组	300
11.3.8	合并数组	301
11.3.9	删除数组重复项	301
11.3.10	遍历 jQuery 对象	302
11.3.11	获取 jQuery 对象 长度	303
11.3.12	获取选择器和选择 范围	304
11.3.13	获取 jQuery 对象 成员	304
11.4	缓存和队列	305
11.4.1	认识缓存	305
11.4.2	定义缓存	307
11.4.3	获取缓存	308
11.4.4	删除缓存	309
11.4.5	jQuery 缓存规范	309
11.4.6	认识队列	310
11.4.7	定义队列	310
11.4.8	获取队列	312

11.4.9	替换队列	312
11.4.10	删除队列	314

第 12 章 浏览器开发..... 316

12.1	客户端开发基础	316
12.1.1	客户端全局对象	317
12.1.2	全局作用域	318
12.1.3	全局生命周期	319
12.1.4	JavaScript 人机交互	319
12.1.5	控制弹出窗口	320
12.1.6	创建框架集页面	321
12.1.7	窗口不等于框架	322
12.1.8	框架之间相互访问的 方式	322
12.1.9	框架之间的 JavaScript 交互	323
12.1.10	检测浏览器信息	324
12.1.11	检测浏览器类型和 版本	325
12.1.12	检测客户端操作系统 ...	327
12.1.13	客户端导航: Location 对象	327
12.1.14	客户端导航: History 对象	329
12.1.15	客户端导航: 使用 JavaScript 访问历史记录	329
12.1.16	客户端导航: 使用 JavaScript 访问帧的历史记录	330
12.1.17	控制窗口大小	330
12.1.18	文档对象模型	331
12.1.19	生成文档的两种方式	334
12.2	浏览器调整实战演练	335
12.2.1	高级浏览器检测	335
12.2.2	演示 jQuery Browser 插件的 实际应用	336
12.2.3	强制浏览器重新渲染 页面	339
12.2.4	解决 IE 浏览器下拉菜单固 定宽度解析问题	340
12.2.5	浏览器插件检测	343

第 13 章 导航开发..... 345

13.1	常用菜单	345
------	------------	-----

- 13.1.1 jMenu 多级菜单: 插件
下载 345
- 13.1.2 jMenu 多级菜单: 构建菜单
结构 346
- 13.1.3 如何简单调用 jMenu
插件 346
- 13.1.4 jMenu 插件包含的多个
参数 348
- 13.1.5 自定义 jMenu 多级
菜单 349
- 13.1.6 uiSelect 下拉菜单: 插件下载
与参数 350
- 13.1.7 uiSelect 下拉菜单: 插件的
用法 350
- 13.1.8 Apycom 弹出菜单: 插件
下载 352
- 13.1.9 Apycom 弹出菜单: 插件具
体用法 352
- 13.1.10 Smartango 内容菜单 ... 355
- 13.2 创意的导航 359
- 13.2.1 jqDock 码头导航 359
- 13.2.2 File Tree 树形导航 361
- 13.2.3 AnythingSlider 预览导航/366

第 14 章 表格开发 370

- 14.1 功能性表格开发 370
- 14.1.1 uiTableFilter 表格行过滤/370
- 14.1.2 tablesorter 表格排序 372
- 14.1.3 HeatColor 表格颜色 377
- 14.1.4 Pagination 表格分页显示/379
- 14.1.5 tableRowCheckboxToggle
表格行复选框 382
- 14.1.6 TableDnD 拖曳表格行 .. 385
- 14.1.7 uiTableEdit 表格编辑 386
- 14.2 表格系统功能开发 388
- 14.2.1 DataTables 表格组件 388
- 14.2.2 Flexigrid 表格组件 391

第 15 章 表单开发 397

- 15.1 功能性表单开发 397
- 15.1.1 Elastic 弹性文本域 397
- 15.1.2 Autotab 自动 Tab 文
本框 398
- 15.1.3 passwordStrength 密码强度
指标 400

- 15.1.4 formToWizard 表单填充
向导 403
- 15.1.5 maskedInput 屏蔽输入框/406
- 15.2 表单皮肤设计 409
- 15.2.1 checkbox 复选框 409
- 15.2.2 jqTransform 表单样式 .. 412
- 15.2.3 niceForms 表单样式 415
- 15.3 表单验证 416
- 15.3.1 jQuery Validation 表单
验证 416
- 15.3.2 validationEngine 表单
验证 417
- 15.4 表单 UI 插件 419
- 15.4.1 datePicker 日期选择器 .. 419
- 15.4.2 jqUploader 文件上传器/ 421

第 16 章 异步通信开发 424

- 16.1 HTTP 解析 424
- 16.1.1 认识 HTTP 424
- 16.1.2 HTTP 请求 424
- 16.1.3 HTTP 响应 426
- 16.2 使用隐藏框架实现异步通信 ... 428
- 16.2.1 隐藏框架 428
- 16.2.2 异步交互与回调处理 ... 430
- 16.2.3 使用 iframe 处理异步
交互 433
- 16.3 使用 JSONP 实现跨域异步
通信 435
- 16.3.1 认识<script>标签 435
- 16.3.2 使用 script 元素实现异步
通信 437
- 16.3.3 异步通信中的参数传递/438
- 16.3.4 异步通信中的响应数据
类型 442
- 16.3.5 认识 JSONP 异步通信
协议 444
- 16.3.6 了解异步通信的安全性/447
- 16.4 使用 Ajax 实现异步通信 448
- 16.4.1 认识 XMLHttpRequest
对象 448
- 16.4.2 创建 XMLHttpRequest
对象 449
- 16.4.3 建立连接 451
- 16.5 使用 Ajax 发送请求 452

16.5.1	发送 GET 请求.....	452
16.5.2	发送 POST 请求.....	453
16.5.3	把数据转换为串行 字符串.....	454
16.6	异步监控.....	456
16.6.1	跟踪 Ajax 异步 通信状态.....	456
16.6.2	中止异步请求.....	457
16.7	使用 Ajax 获取响应信息.....	458
16.7.1	获取 XML 结构数据.....	458
16.7.2	获取 HTML 结构片段..	459
16.7.3	获取 Code 代码片段.....	460
16.7.4	获取 JSON 格式数据.....	461
16.7.5	获取 Text 字符串.....	462
16.7.6	获取和设置头部消息.....	463
16.8	异步通信功能封装和应用.....	463
16.8.1	封装异步请求函数.....	464
16.8.2	使用 Ajax 改进 Tab 显示.....	464

第 17 章 数据存储开发..... 469

17.1	XML 数据格式.....	469
17.1.1	认识 XML.....	469
17.1.2	XML 基本结构.....	470
17.1.3	XML 声明.....	471
17.1.4	XML 处理指令.....	471
17.1.5	XML 注释.....	471
17.1.6	XML 元素.....	472
17.1.7	XML 属性.....	473
17.2	XML 文档对象模型.....	474
17.2.1	认识 XML DOM.....	474
17.2.2	读取数据.....	476
17.3	在 IE 中存储 XML 数据.....	477
17.3.1	创建 XML DOM 对象..	478
17.3.2	加载 XML 数据.....	478
17.3.3	数据加载错误处理.....	480
17.4	在 Mozilla 中存储 XML 数据..	481
17.4.1	创建 XML DOM 对象..	481
17.4.2	加载 XML 数据.....	481
17.4.3	读取 XML 数据.....	482
17.5	XPath 数据查询.....	483
17.5.1	XPath 语法.....	484

17.5.2	在 IE 中使用 XPath.....	485
17.5.3	在 Mozilla 中使用 XPath.....	486

17.6	JSON 数据格式.....	487
17.6.1	认识 JSON.....	488
17.6.2	JSON 数据结构.....	490

第 18 章 使用 jQuery 打造个性 网站.....493

18.1	案例背景介绍.....	493
18.1.1	案例分析.....	493
18.1.2	案例六大具体功能和效果 演示.....	494
18.2	网站设计思路.....	496
18.3	数据结构设计.....	498
18.3.1	网站数据结构概述.....	498
18.3.2	数据表结构简介.....	500
18.4	网站结构和样式.....	503
18.4.1	网站结构.....	503
18.4.2	网站动态布景.....	504
18.4.3	表单结构.....	505
18.4.4	网站布局.....	507
18.5	公用函数开发.....	507
18.5.1	数据库连接.....	508
18.5.2	数据库操作基本函数.....	508
18.5.3	网站门禁系统.....	509
18.5.4	DOM 操作补充工具.....	510
18.6	动态表单及初始化.....	511
18.6.1	生成表单.....	512
18.6.2	表单初始化配置.....	513
18.7	数据库操作.....	515
18.7.1	数据录入.....	515
18.7.2	数据查询.....	517
18.8	数据显示.....	522
18.8.1	把数据行转换为可编辑的 表单域行.....	524
18.8.2	更新数据.....	529
18.8.3	添加数据.....	532
18.8.4	删除数据.....	535
18.8.5	恢复数据.....	538
18.8.6	显示数据.....	540
18.8.7	数据显示对外主函数.....	544
18.9	页面初始化.....	555

jQuery is a new kind of JavaScript Library.

第 1 章 jQuery 开始

JavaScript 是一种脚本语言，主要功能是设计网页交互、动态效果。由于 JavaScript 版本之间的差异性，以及各大浏览器对于 JavaScript 和 DOM 解析的不统一，这些都给开发人员带来了麻烦。

为了简化 JavaScript 开发，一些 JavaScript 代码库诞生了。这些代码库封装了很多预定义的对象和实用函数，能够简化开发人员的工作，提高代码执行效率。JavaScript 库的诞生，标志着真正的 Web 2.0 应用开发到来，为丰富客户端开发奠定基础。

1.1 jQuery: 改变 JavaScript 代码编写方式

jQuery (<http://jquery.com/>) 是一个快速、简洁的 JavaScript 框架，是继 Prototype 之后又一个优秀的 JavaScript 代码库（或 JavaScript 框架）。jQuery 设计的宗旨是“Write Less, Do More”，即倡导写更少的代码，做更多的事情。它封装 JavaScript 常用的功能代码，提供一种简洁、快捷的 JavaScript 设计模式，优化 HTML 文档操作、事件处理、动画设计和 Ajax 交互。

可以说，jQuery 改变用户编写 JavaScript 代码的方式。由于 jQuery 能够兼容 CSS 3，兼容各种主流浏览器，如 IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0+ 等，因此它正被越来越多的开发人员喜爱和选用。

1.2 jQuery 项目及框架

jQuery 诞生于 2006 年初，属于一个开源项目。目前，jQuery 框架也被微软完整封装到 Visual Studio 2008 中，读者在 VS 2008 中开发 JavaScript 脚本时，可以直接调用 jQuery 框架。

1.2.1 jQuery 项目 4 个部分

jQuery 项目主要包括 jQuery Core（核心库）、jQuery UI（界面库）、Sizzle（CSS 选择器）和 QUnit（测试套件）4 个部分，参考网址如表 1.1 所示。

表 1.1 jQuery 参考网址

类 型	网 址
jQuery 框架官网	http://jquery.com/
jQuery 项目组官网	http://jquery.org/
John Resign 个人网站	http://ejohn.org/

1.2.2 jQuery 框架优势

在众多流行的 JavaScript 框架中，jQuery 能够脱颖而出，说明它拥有很多优势，简单概括如下。

1. 代码小巧

小巧是 jQuery 先天优势,任何框架都无法相比。如果采用 Dean Edwards 的 Packer(<http://dean.edwards.name/packer/>) 压缩后, jQuery 源代码仅有 30KB 左右,如果服务器端启用 gzip 压缩后,甚至只有 16KB 大小。

2. 操作比较方便

jQuery 提供了强大的 HTML 元素选择功能。Sizzle 引擎功能强大,能够支持 CSS 1 到 CSS 3 所有的选择器、XPath 选择器以及 jQuery 自定义选择器。但是用户只需要调用 jQuery() 函数 (或 \$()) 即可,使用非常方便,不需要记忆很多访问的方法。

3. 兼容主流浏览器

jQuery 能够在 IE 6.0+、FF 2+、Safari 2.0+ 和 Opera 9.0+ 下正常运行,解决了 JavaScript 在不同浏览器中的差异性。

4. 优雅的语法规则

jQuery 中最有特色的莫过于它的链式操作方式,即对发生在同一个 jQuery 对象上的一组动作,可直接连写而无须重复获取对象。这一点使 jQuery 的代码无比优雅。

5. 支持扩展功能

jQuery 提供了丰富的插件支持。jQuery 的易扩展性可以方便任何用户扩展 jQuery 的功能,因此也吸引了来自全球各地的开发者来共同编写 jQuery 的扩展插件。目前已经有超过几百种的官方插件支持。

6. 封装烦琐的底层技术

jQuery 封装了 DOM 操作,并定义了大量的方法,使用户在编写 DOM 操作相关程序时能够得心应手,优雅地完成各种原本非常复杂的操作。jQuery 还封装了大量的事件处理函数,使得 jQuery 处理事件绑定时非常稳定。jQuery 将所有的 Ajax 操作封装到一个函数 \$.ajax() 中,使得用户在处理 Ajax 时能够专心处理业务逻辑而无须关心复杂的浏览器兼容性以及 XMLHttpRequest 对象的创建和使用的问题。

7. 框架封闭

jQuery 只建立一个名为 jQuery 的对象,其所有的方法都在这个对象之下。另外的一个别名 \$ 也是可以随时交出控制权的,不会污染其他的对象,也不会与其他 JavaScript 框架发生冲突。

8. 自成体系

jQuery 完全摆脱 JavaScript 的设计模式,可以允许用户在 jQuery 环境下自由开发程序,调用 jQuery() 函数选择相关匹配元素,然后直接在 jQuery 对象上完成操作,而不用在 jQuery 和 JavaScript 两种设计模式中来切换。

9. 丰富完善的学习和应用环境

jQuery 框架能够后来者居上,主要得益于它的全新设计模式,并产生了滚雪球效应。网上到处充斥着 jQuery 相关的学习资料和开发应用案例,同时 jQuery 团队也提供了大量的帮助文档,这些都为 jQuery 框架的普及和推广奠定了基础。

1.3 使用 jQuery 前的准备工作

jQuery 如此受欢迎，有很大一部分原因是它使得用户用最少的代码就能非常优雅地找出和操作 HTML 元素，且非常有效。下面我们就一起走入 jQuery 的世界，去感受它的敏捷。

1.3.1 安装 jQuery 库

学习和使用 jQuery 框架的第一步应该是安装 jQuery 代码库。

进入 jQuery 官方网站 (<http://jquery.com/>) 下载最新版本的 jQuery 源代码，如图 1.1 所示。目前的最新版本是 1.4.2。

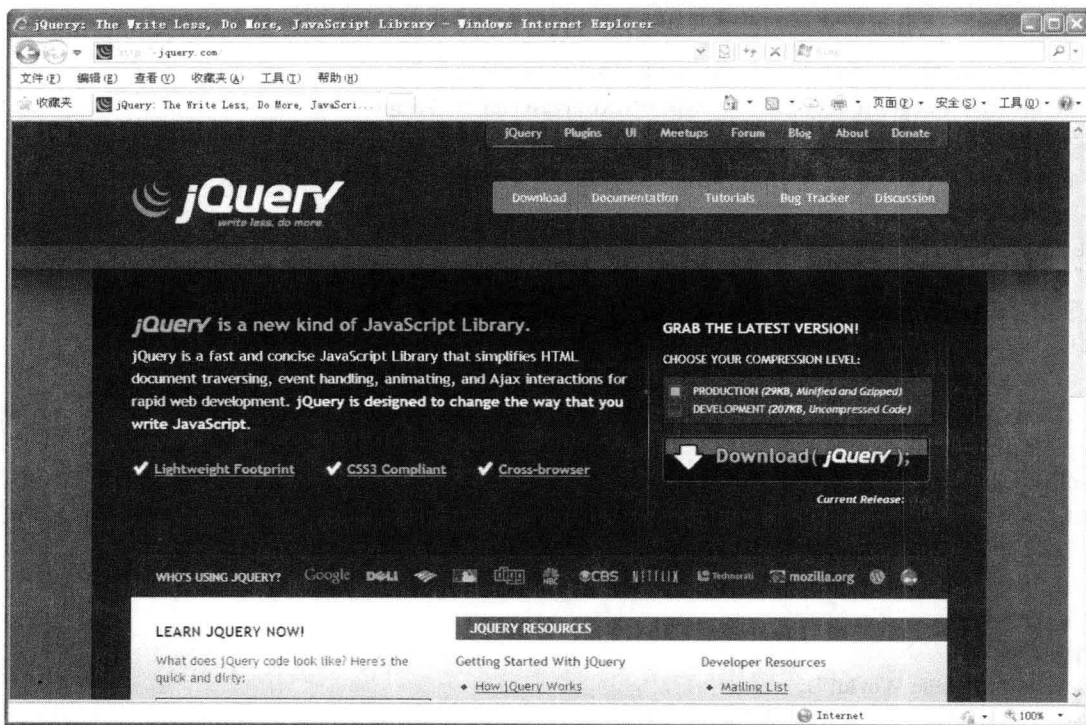


图 1.1 下载 jQuery 最新版本

 **注意：**本书主要根据 1.3.2 版本进行讲解，由于 jQuery 1.4.2 仅增加和完善了部分功能，所以读者使用 jQuery 1.4.2 版本也可以进行学习和上机测试。

在 jQuery 官网首页右侧，先勾选要下载的 jQuery 框架类型，主要包括发布版和测试版。

- Production (24KB, Minified and Gzipped)。
- jquery-1.4.2.js Development (155KB, Uncompressed Code)。

如果选择 Production (发布) 选项，则可以下载代码压缩版本，此时 jQuery 框架源代码被压缩到了 24KB，下载的文件为 jquery-1.4.2.min.js。

如果选择 Development (开发) 选项，则可以下载包含注释的未被压缩的版本，大小为 155KB，下载的文件为 jquery-1.4.2.js。然后，单击 Download(jQuery)按钮即可。

如果要下载 jQuery 其他版本的源代码，可以访问下面网址进行下载：