

前端开发系列

提供全书
示例代码下载



轻松玩转jQuery

蓝健 编著

本书基于jQuery 1.6.4

 **jQuery**
write less do more

- jQuery 是一个JavaScript 库
- jQuery 极大地简化了JavaScript 编程
- jQuery 很容易学习

多学一点知识，就少写一行代码

 **人民邮电出版社**
POSTS & TELECOM PRESS



郑州大学 *04010783672 *

74

轻松玩转 jQuery

蓝健 编著



TP312JA
L044

人民邮电出版社
北京

图书在版编目 (C I P) 数据

轻松玩转jQuery / 蓝健编著. — 北京 : 人民邮电出版社, 2012. 3
ISBN 978-7-115-27296-6

I. ①轻… II. ①蓝… III. ①JAVA语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2011)第268915号

内 容 提 要

本书围绕着“获取需要的jQuery对象,操作获得的jQuery对象”这条主线,全面讲解了选择器、文档操作、事件处理、动画特效、Ajax交互、jQuery插件、jQuery UI插件等重要知识点。同时,为了进一步完善jQuery的知识体系,一方面对Firebug调试工具进行了详细介绍,以此培养学以致用用的调试技能;另一方面在每一章后面提供了有趣的习题,用来测试和巩固学习的成效。

为了帮助读者轻松掌握这些内容,全书力图“为每一段话举例,让每一段话都明明白白”,所以自行设计了340多个代码示例及配套源程序。它们虽然看似简单,但是却对jQuery的各种函数和方法调用进行了直观地诠释。读者完全可以从中获得启发,去解决更为复杂的实际问题。

相信本书能够让读者深刻地感受到jQuery的易学易用。所以,它既可作为jQuery初学者的入门教材,也适合于网络前端开发人员进一步学习和参考。

轻松玩转 jQuery

-
- ◆ 编 著 蓝 健
责任编辑 蒋 佳
 - ◆ 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>
三河市潮河印业有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 21
字数: 474千字 2012年3月第1版
印数: 1-4000册 2012年3月河北第1次印刷

ISBN 978-7-115-27296-6

定价: 45.00元

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

为什么要学这本书？

当今的时代早已深深刻下了互联网的印记。在互联网中，世界都展现在我们眼前，是那么地触手可及。但是更让人惊喜的是，只要我们愿意，就能够把自己的观点、自己的兴趣、自己的才能……都表达出来让全世界去看、去听。

哪怕你并不是 Web 开发的从业人员，也应该学会如何去制作一个具有自己风格的网站。也许刚开始网站上的内容只有寥寥数笔，网站的外观看上去如同一只丑小鸭，但是没关系，一切都会成长。这个过程中只要记住，把任何想做的事都通过这个平台，去达成部分的实现。

虽然网站的内容可以天马行空、自由发挥，但是网站中每个页面的结构、外观、行为等必须通过规范的技术去构造。目前直接的解决工具就是 XHTML + CSS + JavaScript，它们构成了前端开发不可或缺的三环。下面分别予以介绍。

XHTML 称为可扩展超文本标记语言，包含一整套语义化标签，用于组织页面内容的结构。例如用标签<p>组织一段文字，用标签<table>、<tr>和<td>组织表格和它的每一行、每一个单元格数据等。XHTML 与早期版本的 HTML 基本相似，不过语法上更加严格，最重要的是弃用了以为首的一批纯粹控制样式的标签。

CSS 称为层叠样式表，是一组样式设置规则，用于控制页面元素的外观。它的价值在于能够彻底分离页面的内容与表现形式：页面内容存放在 XHTML 文档中，而用于定义表现形式的 CSS 样式规则存放在另一个文件或 XHTML 文档的文件头部分。这样既便于维护站点的外观，也使得 XHTML 文档代码更加简练。

JavaScript 是一种广泛用于客户端 Web 开发的脚本语言，常用来给 XHTML 网页添加交互功能，使它能够响应用户的各种操作。在传统的 Web 应用程序上，每进行一次操作，用户都需要等待服务器传入新的页面。而 JavaScript 创建的网页具有很快的响应性，给人们的感觉更像是桌面程序，而不像是一连串相互链接着的静态页面。

经过 XHTML、CSS、JavaScript 这 3 个工具的分工，制作网页的基本步骤变得异常清晰。然而即便如此，网页开发依旧不是一件容易的事，原因在于各种浏览器对 CSS 和 JavaScript

的解析差别。

CSS 方面的矛盾主要集中在，IE6 这个曾经辉煌一时的浏览器不能很好地支持 CSS2 技术，然而本该淘汰的它依旧拥有大量的用户群，以致网页设计师不得不想办法编写出兼容 IE6 的代码。于是原本非常简单的 CSS 规则一度让人头大，学习兼容技巧反倒成为了重中之重，测试过程中的艰辛也是罄竹难书。为了解决这个棘手的问题，建议可以直接学习使用 CSS2 规范，并彻底放弃兼容 IE6，毕竟 IE6 的淘汰是大势所趋。

JavaScript 方面的矛盾主要集中在，各个浏览器都有自己的一套标准，让 JavaScript 代码实现跨浏览器兼容简直是个噩梦。早期的 Web 开发人员会将通用的兼容方法放到一个 js 文件中，建立自己的公共类库，以供随时调用。但是每个人都自己重复制造一次“轮子”，来解决跨浏览器兼容这类琐碎的细节，不能不说是一种脑力劳动的无谓浪费。好在，如今这个问题已经有了完美的解决方案：那就是直接使用 jQuery，让一切变得简单。

基于这样的考虑，学习本书，认识 jQuery，掌握 jQuery，势在必行。

为什么要选这本书？

jQuery 的学习资料如今已经非常丰富了。但是对于初学者，并不是每一种学习资料都是适合的。例如 jQuery 的参考文档，不得不说这是一个非常好的备查手册，罗列了全部的语法知识点。但是它给人留下的“全面完整”这个印象恰恰又是站不住脚的。

首先，jQuery 的参考文档虽然把 jQuery 的命令进行了很好的分类，比如选择器、核心、属性、CSS、遍历、文档处理、事件、效果、AJAX、工具。但是却完全没有去说明每类 jQuery 命令之间的关系，这使得读者所学的知识是割裂的。为此，本书使用了一条主线把各类 jQuery 命令给串了起来。与大量的 jQuery 命令相比，这条主线更容易记忆，也更容易指导 jQuery 的编程实践。

其次，“学而不思则罔，思而不学则殆”这句名言在我看来完全可以改为“学而不练则罔，练而不学则殆”。这是因为只有在解决实际问题的过程中，人们才有可能进行深入的、有价值的思考。与 jQuery 的参考文档不同，本书在每一章最后都设置了一道习题。这些习题只需要使用相应章节或之前章节的知识点就能解决，并且解决习题所使用的关键命令是相应章节的重点。

最后，调试程序是异常重要的、也是耗时最多的一环。遗憾的是，jQuery 的参考文档不会提及这个内容，甚至许多 jQuery 图书也只是简单地附上一小段文字，让读者知道可以用什么工具来调试以及这个工具大致长什么样。针对这个问题，在本书的附录部分，对 Firebug 调试工具进行了详细介绍，并且通过丰富的示例一步一步地引导读者真正掌握到能够学以致用用的调试技能。

所以，建立起一套真正完整的、适合学习的知识体系，是本书始终关注的问题。当然了，每个读者自身情况各有不同，本书搭建的知识体系只是抛砖引玉，还需要读者自己进一步地努力。

除了建立一套适合学习 jQuery 的知识体系外，作为计算机类书籍，本书还不可避免地需

要展示大量的示例代码，来增加读者的感性认识。可以说，一个好的例子胜过千言万语。据我的观察，目前对示例代码的处理主要有两种方式。

第一种方式是在书中写出完整的代码，而不提供可以直接运行的源程序文件，如此做对读者的好处是可以强迫他亲自在电脑上实践编程。然而，哪怕书上的代码一个标点都不错，一个字符都不漏，轮到读者自己把书上的代码输到电脑中，也会出现大量的错误，修改起来非常耗费工夫。初期的学习过程如此不顺利，真可谓是万事开头难。相信有过这种亲身体验的读者，对这类书唯恐避之不及吧。

第二种方式是在书中写出主要的代码，同时提供可以直接运行的源程序文件，于是可以轻松在电脑上观看到每个示例的效果。这样做虽然避免了许多麻烦，但是也很自然地让读者丧失了亲身编写和调试程序的机会。也许大家会直观地认为把书上的示例重新完整地在电脑中复现一次，简直就是在做无用功。但是实际上这个过程有一种不可替代的效果，可以说，任何一个教书先生都在孜孜不倦地强迫他的学生干这件事。

权衡利弊，本书还是提供了配套的源程序包。不过，建议每当学完一章内容，一定要去尝试着解决这一章的课后习题。假若能够成功解答，表明基本掌握了已学的内容，可以继续下一章的学习。否则的话，建议把这一章的每个示例也看成一道习题，试着在电脑中重新实现一次，直到出现灵感为止。我相信，只要读者按照这种方法，跟着这本书学和练，必定能在最短的时间内成为一个彻底玩转 jQuery 的高手。

作者

2011年11月

第一部分 jQuery 基础

第 1 章 jQuery 简介	2
1.1 什么是 jQuery	2
1.1.1 JavaScript 的基本概念	2
1.1.2 JavaScript 库的基本概念	2
1.1.3 jQuery 的基本概念	3
1.1.4 jQuery 的代码特色	4
1.2 如何用 jQuery	5
1.2.1 配置 jQuery 开发环境	5
1.2.2 选用 jQuery 开发工具	6
1.2.3 混合使用 jQuery 与 JavaScript	7
1.2.4 混合使用 jQuery 与其他 JavaScript 库	7
第 1 章习题	9
第 2 章 生成 jQuery 对象	11
2.1 jQuery 对象	11
2.1.1 jQuery 对象和 DOM 对象的简介	11
2.1.2 jQuery 对象和 DOM 对象的相互转换	13
2.2 jQuery 选择器	14
2.2.1 基本选择器	17
2.2.2 过滤选择器	18
2.2.3 层次选择器	21
2.2.4 表单选择器	21
第 2 章习题	24
第 3 章 更改 jQuery 对象	27
3.1 更改操作概述	27
3.1.1 更改对象操作	27
3.1.2 链式操作设定	27
3.2 更改 jQuery 对象的方法	28
3.2.1 更改为后代元素集合	31
3.2.2 更改为祖先元素集合	32
3.2.3 更改为兄弟元素集合	33
3.2.4 更改为更多元素集合	34
3.2.5 更改为部分元素集合	35
3.3 还原 jQuery 对象的方法	37

第3章习题	37
第4章 jQuery 对象的文档处理	40
4.1 移动元素	41
4.2 添加元素	45
4.3 替换元素	47
4.4 包裹元素	50
4.5 删除和清空元素	52
第4章习题	53
第5章 jQuery 对象的属性处理	56
5.1 元素的属性处理	57
5.2 元素的 class 属性处理	58
5.3 元素内部的 HTML、文本处理	61
5.4 表单元素的属性处理	63
第5章习题	65
第6章 jQuery 对象的 CSS 处理	67
6.1 CSS 基本属性处理	68
6.2 CSS 尺寸属性处理	70
6.3 CSS 位置属性处理	72
第6章习题	74
第7章 jQuery 对象的动画处理	76
7.1 隐藏与显示动画效果	76
7.2 淡出与淡入动画效果	77
7.3 滑出与滑入动画效果	79
7.4 自定义动画效果	80
7.5 动画队列	82
7.5.1 动画队列概述	82
7.5.2 延迟动画队列	82
7.5.3 中止动画队列	83
7.5.4 循环动画队列	84
7.6 动画回调函数	85
第7章习题	86
第8章 jQuery 对象的事件处理	88
8.1 页面加载	88
8.2 事件方法	89
8.2.1 绑定事件	89
8.2.2 触发事件	93
8.2.3 委派事件	95
8.3 事件类型	98
8.3.1 鼠标点击事件	98
8.3.2 鼠标移动事件	99
8.3.3 鼠标合成事件	101

8.3.4	得到焦点和失去焦点事件	102
8.3.5	键盘事件	103
8.3.6	表单事件	105
8.3.7	加载和卸载事件	106
8.4	事件对象	107
8.4.1	阻止事件默认行为的方法	107
8.4.2	阻止事件冒泡的方法	107
8.4.3	阻止事件处理函数执行的方法	109
8.4.4	事件对象的事件类型和命名空间	110
8.4.5	事件对象的当前元素和触发元素	111
8.4.6	事件对象的传递参数	111
8.4.7	事件对象的触发键	112
8.4.8	事件对象的发生位置和发生时间	113
第 8 章	习题	114
第 9 章	jQuery 对象的其他处理	116
9.1	对元素进行遍历操作	116
9.2	对元素进行数据存取	119
9.3	元素的个数和索引	123
第 9 章	习题	125
第 10 章	jQuery 的基本方法	127
10.1	处理字符串的方法	127
10.2	处理对象的方法	128
10.3	处理数组的方法	129
10.4	处理函数的方法	132
第 10 章	习题	133
第 11 章	jQuery 的 Ajax 方法	135
11.1	Ajax 简介	135
11.2	Ajax 底层方法	138
11.3	Ajax 快捷方法	148
11.4	Ajax 跨域方法	150
11.5	全局 Ajax 设置	153
第 11 章	习题	155

第二部分 jQuery 插件

第 12 章	jQuery 插件基础	160
12.1	jQuery 插件简介	160
12.1.1	JavaScript 函数的基本概念	160
12.1.2	jQuery 插件的基本概念	163
12.2	制作原创的 jQuery 插件	165
12.2.1	构造 jQuery 插件	165
12.2.2	优化 jQuery 插件	169
12.3	使用共享的 jQuery 插件	180

12.3.1	Cookie 插件	180
12.3.2	Tablesorter 插件	187
第 12 章	习题	195
第 13 章	jQuery UI 交互功能	198
13.1	jQuery UI 插件简介	198
13.2	使元素具有拖动功能	198
13.3	使元素具有投放功能	205
13.4	使元素具有缩放功能	211
13.5	使元素具有选择功能	216
13.6	使元素具有排序功能	220
第 13 章	习题	229
第 14 章	jQuery UI 页面部件	232
14.1	滑块条部件	232
14.2	进度条部件	237
14.3	按键块部件	240
14.4	对话框部件	244
14.5	折叠面板部件	252
14.6	选项面板部件	256
14.7	自动提示部件	262
14.8	日期选择部件	269
第 14 章	习题	275
第 15 章	jQuery UI 动画效果	277
15.1	颜色动画方法	277
15.2	class 属性动画方法	278
15.3	特效动画方法	281
第 15 章	习题	286
附录 A	jQuery 命令检索表	288
附录 B	Firebug 调试工具	297
B.1	安装 Firebug	297
B.2	启动 Firebug	297
B.3	HTML 面板	297
B.3.1	查看和调试页面的 HTML 实时代码	299
B.3.2	查看和调试元素的 CSS 实时代码	300
B.4	脚本面板	302
B.4.1	使用断点中断脚本的运行	304
B.4.2	使用控件继续脚本的运行	306
B.5	控制台面板	308
B.5.1	查看脚本运行信息	310
B.5.2	使用命令行调试脚本	311
B.5.3	使用日志方法调试脚本	312
附录 C	各章习题解答	314

第一部分 jQuery 基础

- 第 1 章 jQuery 简介
- 第 2 章 生成 jQuery 对象
- 第 3 章 更改 jQuery 对象
- 第 4 章 jQuery 对象的文档处理
- 第 5 章 jQuery 对象的属性处理
- 第 6 章 jQuery 对象的 CSS 处理
- 第 7 章 jQuery 对象的动画处理
- 第 8 章 jQuery 对象的事件处理
- 第 9 章 jQuery 对象的其他处理
- 第 10 章 jQuery 的基本方法
- 第 11 章 jQuery 的 Ajax 方法

jQuery 简介

1.1 什么是 jQuery

1.1.1 JavaScript 的基本概念

为了认识 jQuery，就必须从 JavaScript 说起。JavaScript 诞生于 1995 年，是网景公司为其 Netscape 浏览器开发的一种脚本语言，可以插入到 HTML 代码中，由浏览器解析并运行。从那时起，这种语言越来越普及，很快几乎所有的浏览器都能够支持。

在 JavaScript 的发展过程中，它曾一度受到许多人的误解，但越来越多的人认识到，JavaScript 不是一种玩具语言。一方面，JavaScript 可以减少浏览器与服务器的通信次数，典型的例子是在浏览器端进行表单验证。另一方面，JavaScript 可以提高浏览器与服务器的通信效率，其代表是基于 JavaScript 的 Web 技术——Ajax。这些功能不仅减少了与服务器的通信成本，同时让网页具有和桌面计算机程序一样的响应性，大大提升了用户体验。

总之，JavaScript 已经是 Web 开发的必备组件，它能够让 HTML 网页包含更多活跃的交互元素。而随着现代浏览器不断地提升 JavaScript 的性能和扩充 JavaScript 的功能，相信它将来会有更多的用武之地。

1.1.2 JavaScript 库的基本概念

虽然仅凭 JavaScript 自身就可以创建具有高度交互性的动态页面，但是使用 JavaScript 并不是一件容易的事。和 CSS 一样，各种浏览器对 JavaScript 的解析方式都有所区别，实现 JavaScript 代码跨浏览器兼容对开发者来说是一个恶梦。

为了解决开发过程中的浏览器兼容性问题，许多 JavaScript 库应运而生。这些 JavaScript 库由富有经验、了解系统特性的开发者编写，其中封装了很多预定义的对象方法，直接

调用它们就能编写出兼容各种不同的浏览器的程序。于是，我们的关注点可以集中在功能需求而非实现细节上，从而提高了项目的开发速度。

为了让大家更直观地认识到 JavaScript 库在实际开发中的地位，图 1.1 展示了一张国外科技网站 w3techs.com 对近 100 万个网站的统计图表。其中的数据显示超过 40% 的网站都使用了 JavaScript 库，被频繁使用的 JavaScript 库包括 jQuery、MooTools、Prototype、Script.aculo.us、YUI Library、ASP.NET Ajax、Spry、Dojo、Ext JS。而最让人印象深刻的是 jQuery，在使用 JavaScript 库的网站中，它占到 76.4% 的市场份额，成为这一领域不可动摇的霸主。

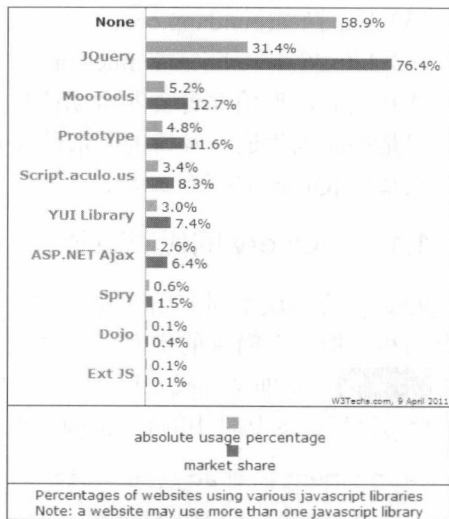


图 1.1 JavaScript 及其 JavaScript 库的市场份额

1.1.3 jQuery 的基本概念

jQuery 是由美国人 John Resig 于 2006 年创建的一个 JavaScript 库，从最初的 1.0 版本发展到 2011 年 3 月面世的 1.5.2 版本。作为一个 JavaScript 库，jQuery 极大程度上解决了浏览器的兼容性问题，能够在 IE 6.0 +、FF 2.0 +、Safari 3.0 +、Opera 9.0 + 以及 Chrome 下正常运行。

虽然 jQuery 并不是唯一的 JavaScript 库，也不是最早出现的 JavaScript 库，但是 jQuery 的影响力却是最大的，它不断地吸引着全球技术爱好者和专家加入其阵营，几乎已经成为 Web 开发领域的事实标准。如今，哪怕是坚持不愿意使用 jQuery 的 JavaScript 工程师也必须能够看懂 jQuery 代码，否则他就无法方便地与其他人进行技术交流合作。所以，jQuery 已成为前端开发从业人员必须掌握的基本技能。

按理来说，jQuery 只不过是基于 JavaScript 建立起来的一组代码，它的能力不可能超出 JavaScript，甚至某些功能的实现依旧要依靠 JavaScript 本身。也许很多人想当然地认为学习使用 JavaScript 就可以了，jQuery 还是留给那些投机取巧的人吧。但是事实证明，jQuery 不是可有可无的，它的出现彻底解放了开发者的创造力，使得大量优秀的网络应用爆发式地涌现，令人目不暇接，可谓是开创了原先 JavaScript 没有做到的繁荣局面。

所谓闻名不如见面，为了近距离地去体验 jQuery 的实际应用，读者可以去访问下面的几个国外网站。这些炫丽的网站隐藏了一个共同的特点，那就是都使用 jQuery 在网页上创造出丰富的交互元素和动态效果。

- (1) <http://www.indofolio.com/>
- (2) <http://www.head2heart.us/>
- (3) <http://modabots.com/>
- (4) <http://duplos.org/>

(5) <http://tearroundapp.com/>

(6) <http://www.slicemachine.com/>

欣赏完后，恐怕读者会非常期待自己也能制作出如此效果的网页。确实，无限风光在顶峰，但是不能因为迫不及待的心情就妄想速成。如果非要找一条捷径，那么就静下心来学习和彻底掌握 jQuery 的基础知识。

1.1.4 jQuery 的代码特色

jQuery 的与众不同突出表现在它的代码特色上，为了让读者直观地感受到这一点，下面将介绍和分析一个简单的实例，来看看 jQuery 是如何改变 JavaScript 的编程方式。

示例 1 页面文档载入后，为所有的<h3>元素绑定 click 事件。当鼠标单击某个<h3>元素时，会切换其 class 属性中的 highlight 属性值，同时它后面的兄弟元素会切换其显示隐藏状态。

```
$(document).ready(function(){
    $("h3").click(function(){
        $(this).toggleClass("highlight")
            .next("ul").toggle();
    });
});
```

1. \$符号标志

示例 1 的代码里反复出现\$符号，它是变量名 jQuery 的简写形式，\$()等效于 jQuery()。实际上，几乎每一行 jQuery 命令的开头都需要使用该符号，可以说它是 jQuery 程序的标志，甚至能够靠它迅速区分出一个页面是否使用了 jQuery。

2. 隐式循环

当用 jQuery 找到符合要求的全部元素后，无需显式地循环遍历每一个元素进行操作，比如示例 1 中所有<h3>元素都会一一绑定 click 事件。依靠这种特性，使用 jQuery 命令的同时就可以实现循环结构的功能，从而大幅减少了代码量。

3. 链式书写

示例 1 中，toggleClass()和 next()这两个操作都是针对同一个对象——当前被鼠标单击的<h3>元素。jQuery 程序里允许把发生在同一个对象上的一组命令直接连写在一个语句中，其中还可以插入换行和空格。这种链式书写方式使 jQuery 的代码无比简洁和优雅。

注意

jQuery 需要配合 HTML、CSS 一起使用，比如示例 1 的<h3>、元素是在 HTML 代码里定义的，class 属性值 highlight 是在 CSS 代码里定义的。为强调重点，部分示例在书中并没有列出相应的 HTML 和 CSS 代码。如果有需要，读者可以在本书配套的示例源程序包中查看。

1.2 如何用 jQuery

1.2.1 配置 jQuery 开发环境

为了使页面中的 jQuery 代码生效，必须在 jQuery 代码前引入 jQuery 库文件。jQuery 是一个免费开源项目，任何人都可以在 jQuery 的官方网站 <http://jquery.com> 下载到最新版本的 jQuery 库文件。本书所有示例都是基于 1.6.4 版本进行编写的，所以建议选择下载该版本。

jQuery 库文件有两种类型：完整版和压缩版。前者主要用于测试开发，后者主要用于项目应用。例如 jQuery 1.6.4 版本有 `jquery-1.6.4.js` (233KB) 和 `jquery-1.6.4.min.js` (90KB，服务器开启 Gzip 压缩后变为 31KB) 两个文件，它们分别对应完整版和压缩版。

本书配套的示例源程序包中，就有 `jquery-1.6.4.min.js` 可供使用。在 HTML 页面引入该 jQuery 库文件的代码如下：

```
<script src="../../jquery-1.5.2.min.js" type="text/javascript"></script>
```

可以看出，在 HTML 页面上引入 jQuery 库文件和引入外部的 JavaScript 程序文件，形式上没有任何区别。同时，在 HTML 页面直接插入 jQuery 代码或引入外部 jQuery 程序文件，需要符合的格式也跟 JavaScript 一样。

值得一提的是，外部 jQuery 程序文件是不同页面共享相同 jQuery 代码的一种高效方式。这样当修改 jQuery 代码时，只需要编辑一个外部文件，操作更为方便。此外，一旦载入某个外部 jQuery 文件，它就会存储在浏览器的缓存中，因此不同页面重复使用它时无须再次下载，从而加快了网页的访问速度。

下面在前文示例 1 的基础上，配置 jQuery 代码的开发环境。

```
<script src="../../jquery-1.6.4.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function() {
    $("h3").click(function() {
        $(this).toggleClass("highlight")
            .next("ul").toggle();
    });
});
</script>
```



注意 为节省篇幅，本书的所有 jQuery 代码示例，默认导入了 jQuery 库文件，且默认设置了插入 jQuery 代码的 `<script>` 标签格式。

在实际项目中，我们还可以使用 google 提供的 CDN 内容分发网络来加载 jQuery 库文件，这样既能大大节省服务器的带宽，又能加快客户访问网站的速度。此时引入 jQuery 库文件的代码需改变为：

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.min.js" type="text/javascript"></script>
```

1.2.2 选用 jQuery 开发工具

由于 jQuery 本身就是 JavaScript 代码，属于不需要事先编译的脚本语言，所以可以使用任意的文本编辑器进行开发。不过为了提高开发效率，这里推荐使用 Dreamweaver CS4 中文版软件。

Dreamweaver 是由 Macromedia 公司开发的一款所见即所得的网页编辑器。作为一个前端开发平台，它提供了编写 HTML、CSS、JavaScript 代码的配套支持。除了基本的代码编辑功能，语法着色、智能提示等功能也非常的有用。

为了让 Dreamweaver 进一步支持 jQuery，需要安装一个名为“jQuery_API_for_dw4.mxp”的插件，该插件可以直接从本书配套的示例源程序包中获取。具体安装步骤为：在 Dream weaver 中单击“命令”菜单下的“扩展管理”命令，弹出“Adobe Extension Manager”界面后选择安装“jQuery_API_for_dw4.mxp”，成功后重启 Dreamweaver，就可以让 Dreamweaver 拥有 jQuery 自动提示代码功能，如图 1.2 所示。

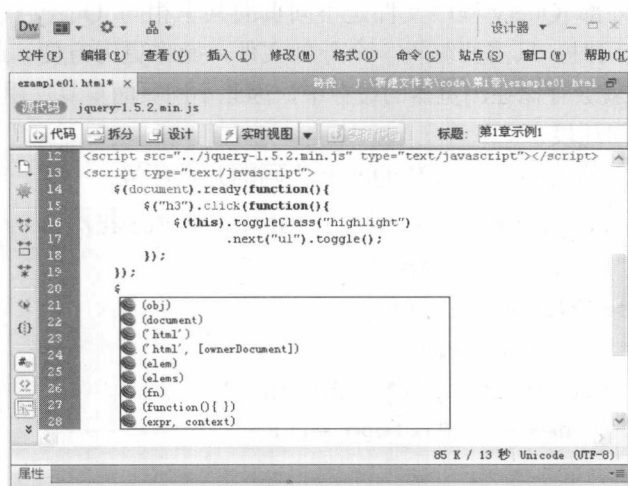


图 1.2 Dreamweaver 的自动提示 jQuery 代码功能

如果用户安装的是精简版的 Dreamweaver，“扩展管理”命令会使用无效，那么需要先下载安装 Adobe Extension Manager 插件管理器。

虽然 Dreamweaver 的功能相当的强大，完全学会掌握颇需一番功夫，但是这里我们只需

要把它作为一个文本编辑器使用，输入所有的代码，并预览运行效果即可。所以，读者切记不要花费精力专门学习 Dreamweaver，毕竟它只是一个网页开发的辅助工具。

注意 本书的源程序代码均已在 Firefox 4.0 + 和 IE 7.0 +、chrome 下获得测试通过，所以读者需要安装或升级该类浏览器，此后在 Dreamweaver 中执行预览命令时就可以选择使用。

1.2.3 混合使用 jQuery 与 JavaScript

网页中引入 jQuery 库文件后，依旧可以使用 JavaScript 编写代码，两者并不冲突。不过为了充分发挥 jQuery 的优势，一旦涉及文档操作、事件处理、动画特效、Ajax 交互等内容，就应该尽可能使用 jQuery 命令。

虽然 jQuery 命令成为了程序的主体部分，但是“数据类型”、“变量”、“运算符”、“表达式”、“控制语句”、“对象”、“数组”、“函数”这些 JavaScript 基本语法依旧会穿插在整个程序中，构成不可缺少的一环。

例如，在前文示例 1 中，单击<h3>元素时会调用一个函数。这个函数的构造，完全符合 JavaScript 语法，并不是 jQuery 特有的东西。为了让大家看清这个事实，这里把示例 1 进一步等效改写为：

```
fun=function(){
    $(this).toggleClass("highlight").next("ul").toggle();
};
$(document).ready(function(){
    $("h3").click(fun);
});
```

为了学好 jQuery，很有必要对 JavaScript 基本语法了解一番。事实上，这些语法概念是每种编程语言都有的，无非在形式规范上有些差异。所以，即使读者对 JavaScript 并不了解，但是只要曾经学过某种编程语言，就可以无障碍地阅读本书。

注意 为避免混淆 jQuery 命令和 JavaScript 命令，本书所有示例如果使用了 JavaScript 全局函数或内置对象方法，都会有简单地说明。而对于 JavaScript 基本语法，默认读者已掌握。

1.2.4 混合使用 jQuery 与其他 JavaScript 库

JavaScript 库之间往往都具备一些相似的功能，所以一般只需要调用一个库即可。但是有一些情况需要混用 2 个或 2 个以上的 JavaScript 库，比如过去使用其他的库编写了许多代码，希望能够继续使用。

为了不妨碍其他库，需要限制对全局命名空间的侵入，所以 jQuery 在库文件中只设定了 jQuery 变量名。虽然 jQuery 变量名不会和其他 JavaScript 库发生冲突，但是它的简写形式 \$，在其他 JavaScript 库也会被使用，这就不可避免地引发了问题。

具体地说，当页面导入多个库时，只有最后导入的库拥有简写记号 \$ 的控制权。如果 jQuery