

JavaScript by Example Second Edition

# JavaScript 详解 (第2版)

## EXAMPLE

```
<html>
<head>
<title>Using JavaScript alert box</title>
<script type="text/javascript">
1  var message1="Match your Quotes and ";
2  var message2="Beware of Little Bugs ";
3  alert("Welcome to JavaScript Programming!\n" +
4    message1 + message2);
</script>
</head>
</html>
```

using the alert() method

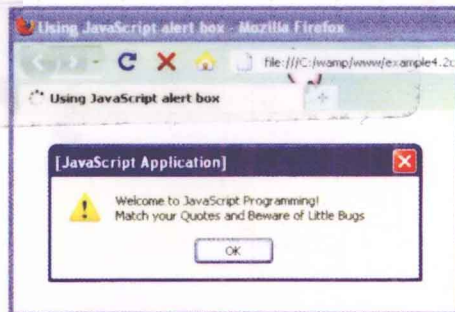
## EXPLANATION

- 1 The JavaScript program starts here with the `<script>` tag.
- 2 Two variables, `message1` and `message2` are assigned text strings.
- 3 The `alert()` method `message1` and `message2` are assigned text strings, backslash sequence, `\n`. There `message1` and `message2` are assigned text strings. JavaScript (see Table 3.1 on page 32), and `message2` are assigned text strings. See the `using the alert() method` escape sequence is.
- 4 The `alert()` method `message1` and `message2` are assigned text strings. The `+` sign concatenates variables. The `+` sign concatenates variables together and variables together as shown in the code on the screen, the program will

[美] Ellie Quigley 著

曹晓立 刘晓娜 译  
赵文婧 杨艳斌

李杰 审校

人民邮电出版社  
POSTS & TELECOM PRESS

JavaScript by Example Second Edition

# JavaScript

## 详解 (第2版)

## EXAMPLE

```
<html>
<head>
<title>Using JavaScript alert box</title>
<script type="text/javascript">
1  var message1="Match your Quotes and ";
2  var message2="Beware of Little Bugs ";
3  alert("Welcome to JavaScript Programming!\n" +
4    message1 + message2);
</script>
</head>
</html>
```

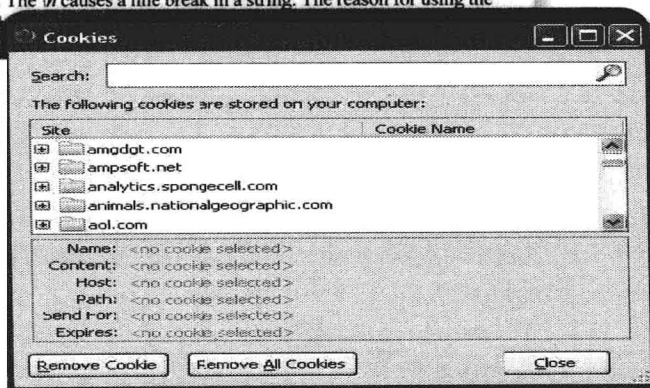
## EXPLANATION

- 1 The JavaScript program starts here with the `<script>` tag.
- 2 Two variables, `message1` and `message2` are assigned text strings.
- 3 The `alert()` method contains a string of text. Buried in the string is a backslash sequence, `\n`. There are a number of these sequences available in JavaScript (see Table 3.1 on page 32). The `\n` causes a line break in a string. The reason for using the `\n` escape sequence is
- 4 The `alert()` method needs two variables. The `+` sign concatenates the two variables together as shown in the code on the screen, the program will

[美] Ellie Quigley 著

曹晓立 刘晓娜 译  
赵文婧 杨艳斌

李杰 审校

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

JavaScript详解: 第2版 / (美) 奎格利  
(Quigley, E.) 著; 曹晓立等译. — 北京: 人民邮电出版社, 2011. 10

(图灵程序设计丛书)

书名原文: JavaScript by Example Second Edition  
ISBN 978-7-115-26291-2

I. ①J… II. ①奎… ②曹… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第184802号

## 内 容 提 要

JavaScript 是目前最流行和最重要的 Web 编程语言。本书从 JavaScript 的编程结构开始, 循序渐进地讲述了 JavaScript 及相关的 CSS、W3C DOM、cookie、正则表达式和匹配模式, 以及功能强大的 Ajax。书中采用了大量课堂实践示例, 附以详尽的代码行解释和实际结果输出, 将 JavaScript 基础知识与高级技巧完美地展现出来。

本书适合对 Web 开发和设计感兴趣的人员阅读, 也可作为高校相关课程的教材。

图灵程序设计丛书

## JavaScript详解 (第2版)

- 
- ◆ 著 [美] Ellie Quigley  
译 曹晓立 刘晓娜 赵文婧 杨艳斌  
审校 李 杰  
责任编辑 朱 巍  
执行编辑 刘美英
- ◆ 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>  
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16  
印张: 42.25  
字数: 1004千字 2011年10月第1版  
印数: 1-3 500册 2011年10月北京第1次印刷

著作权合同登记号 图字: 01-2011-1849号

ISBN 978-7-115-26291-2

定价: 99.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154



# 译者序

自1995年出现以来,JavaScript已经走过了16个年头。如今不论是资深的前端开发人员,还是初入行稚气未脱的新手,都将JavaScript视为一门必修课。而JavaScript又像是哆啦A梦的宝贝口袋,简洁易用的代码总能让枯燥的页面创意百现。

如今互联网新技术层出不穷,JavaScript却能依然保持其旺盛的生命力,并在Web开发中发挥出越来越关键的作用。除了与JavaScript自身的纯文本免编译的特性有关外,还要归功于其衍生的jQuery、MooTools、Prototype等JavaScript框架。这些框架以及其他JavaScript类库大大降低了我们的开发强度,也极大地提升了前端交互性。在互联网应用越来越注重UCD的潮流下,书写更强大、更坚实可靠的JavaScript变得日益迫切。因此,我们希望读者能通过本书,扎实地打好JavaScript基础。

本书原著已经是第2版。作者不仅是Web领域的资深技术专家,也同时执教JavaScript,在JavaScript方面拥有极其丰富的实战经验和坚若磐石的理论基础。作者在撰写本书的过程中倾尽所能,设身处地以前端开发和设计两种角色为出发点,致力于缩小两者之间的鸿沟。粗略翻阅本书,即可看到示例代码占据了大量的篇幅。本书的示例代码全部经过作者的深思熟虑,由浅入深,循循善诱,也恰恰验证了本书正是一本贴近实际开发的教材,这在同类书籍中是难能可贵的。

本书由大连广播电视大学的曹晓立、中国科学院计算技术研究所的刘晓娜、中国移动通信集团山西有限公司的赵文婧,以及空军航空大学的杨艳斌协作译成,并由中国移动通信集团山西有限公司的李杰负责最终统稿、润色及审校。在翻译过程中,我们深深体会到作者清晰的教学思路和缜密的程序逻辑,为保证最大程度传递作者思想,不惜耗费近一年的时间对译稿进行反复斟酌,以便读者能够快速融入,并从中获益。

感谢各位译者的家人和同伴,感谢你们一如既往的支持,也请原谅我们在生活中的种种任性。

感谢人民邮电出版社图灵公司的杨海玲老师和刘美英老师,没有你们的信任与指导,我们不可能完成本书的翻译。

感谢FreeWheel的柳靖和中国移动通信集团山西有限公司的杨琳、王雪、王仲妮、张贝贝等专家为本书的出版倾注了大量的智慧和汗水。

由于译者水平有限,译文中难免会有错漏,望广大同仁批评指正。

李杰

2011年7月1日于太原

# 前 言

本书作为第 2 版不仅只是版本的更新，实际上完全可以算作一本新书了。自从 2002 年发行第 1 版以来，变化是如此之多，再加上现在有了新兴的 Ajax 等流行元素，JavaScript 的发展有如快马加鞭！几乎所有电脑上都安装和运行着 JavaScript。尽管 JavaScript 有很多别名，如 Mocha、LiveScript、JScript、ECMAScript 等，但最流行的 Web 脚本语言非它莫属。市面上有很多关于 JavaScript 的书，它们的关注点各不相同，而对于初学者来说，面对纷繁复杂的选择难免感觉无从下手。本书可谓人人适合，尤其要献给那些欲在这门语言的技术性和趣味性之间寻求平衡的读者。这是一本注重跨平台问题的书，一本低起点的书。通过从易到难、接连不断的例子，本书循序渐进地讲述了 JavaScript 语言的工作机制，在潜移默化的过程中，引领读者熟练掌握基本编程技巧，进而足以应对更深一步的内容：CSS、DOM、Ajax 等。

作为一名教师，我发现在课堂上使用第 1 版进行教学效果非常不错。不过我也发现，为了达到理想效果，还需要更多更好的示例。我的大部分学生不是不懂编码的设计人员，就是不懂设计的编程人员。所以急需编写一本书，在不让双方感觉到无聊或备受打击的前提下，同时满足他们的需求。更新第 1 版所需要付出的巨大努力远远超出了我的预期。在这个过程中我收获颇多，希望大家能分享我的研究成果，跟随我的脚步，在愉悦的氛围中详细了解全球最流行和最重要的 Web 编程语言。

## 致谢

对 Prentice Hall 的各位友人致以深深的谢意！包括给予我最大支持的 Mark L. Taub 总编，生产部主任 Julie Nahil，责任编辑 John Fuller 以及封面设计 Ann Jones。同时感谢 Techne Group 的制作编辑 Dmitri Korzh。最后，特别感谢 Thomas Bishop 专门抽出时间来审阅我的稿件，并提出建设性意见，进而大大提升了本书的质量。还要感谢 Brendon Crawford 的审阅，感谢 Elizabeth Triplett 绘制的插图让每章都有一个愉快的开始。

Ellie Quigley

2010 年 9 月

# 目 录

|                                       |    |                                       |    |
|---------------------------------------|----|---------------------------------------|----|
| 第 1 章 JavaScript 简介 .....             | 1  | 2.2.4 注释 .....                        | 28 |
| 1.1 JavaScript 是什么 .....              | 1  | 2.2.5 <script> 标签 .....               | 29 |
| 1.2 JavaScript 不是什么 .....             | 2  | 2.3 生成 HTML 并输出显示 .....               | 30 |
| 1.3 JavaScript 的用处 .....              | 3  | 2.3.1 字符串及字符串串联 .....                 | 30 |
| 1.4 JavaScript 及其在 Web 页面中的位置 .....   | 3  | 2.3.2 write() 及 writeln()<br>方法 ..... | 31 |
| 1.5 Ajax 是什么 .....                    | 5  | 2.4 关于调试 .....                        | 33 |
| 1.6 JavaScript 是什么样子的 .....           | 6  | 2.5 调试工具 .....                        | 33 |
| 1.7 JavaScript 及其在 Web 开发中承担的角色 ..... | 7  | 2.5.1 Firefox .....                   | 33 |
| 1.8 JavaScript 和事件 .....              | 9  | 2.5.2 在 Internet Explorer 8 中调试 ..... | 36 |
| 1.9 标准化 JavaScript 和 W3C .....        | 11 | 2.5.3 JavaScript:URL 协议 .....         | 37 |
| 1.9.1 JavaScript 对象 .....             | 12 | 2.6 JavaScript 与旧浏览器或受限的浏览器 .....     | 37 |
| 1.9.2 文档对象模型 .....                    | 12 | 2.7 应知应会 .....                        | 40 |
| 1.10 关于浏览器 .....                      | 13 | 练习 .....                              | 41 |
| 1.10.1 JavaScript 的版本 .....           | 14 | 第 3 章 数据类型、字面量和变量 .....               | 42 |
| 1.10.2 你的浏览器遵循标准吗 .....               | 16 | 3.1 数据类型 .....                        | 42 |
| 1.10.3 浏览器是否已启用<br>JavaScript .....   | 16 | 3.1.1 基本数据类型 .....                    | 42 |
| 1.11 JavaScript 代码放在哪儿 .....          | 17 | 3.1.2 复合数据类型 .....                    | 47 |
| 1.12 验证标签 .....                       | 20 | 3.2 变量 .....                          | 47 |
| 1.12.1 W3C 验证工具 .....                 | 21 | 3.2.1 有效变量名 .....                     | 48 |
| 1.12.2 Validome 验证工具 .....            | 21 | 3.2.2 声明和初始化变量 .....                  | 48 |
| 1.13 应知应会 .....                       | 22 | 3.2.3 动态或宽松类型语言 .....                 | 50 |
| 练习 .....                              | 23 | 3.2.4 变量作用域 .....                     | 52 |
| 第 2 章 脚本的安装 .....                     | 24 | 3.2.5 变量的拼接 .....                     | 52 |
| 2.1 HTML 文档与 JavaScript .....         | 24 | 3.3 常量 .....                          | 54 |
| 2.2 语法细节 .....                        | 27 | 3.4 要注意的 bug .....                    | 55 |
| 2.2.1 区分大小写问题 .....                   | 27 | 3.5 应知应会 .....                        | 55 |
| 2.2.2 自由形式及保留字 .....                  | 27 | 练习 .....                              | 56 |
| 2.2.3 语句及分号 .....                     | 28 |                                       |    |

|                                  |     |                                         |     |
|----------------------------------|-----|-----------------------------------------|-----|
| 第4章 对话框.....                     | 57  | 6.3.4 for/in 循环.....                    | 105 |
| 4.1 与用户交互.....                   | 57  | 6.3.5 使用 break 和 continue<br>控制循环.....  | 105 |
| 4.1.1 alert() 方法.....            | 57  | 6.3.6 嵌套循环及标签.....                      | 107 |
| 4.1.2 prompt() 方法.....           | 59  | 6.4 应知应会.....                           | 109 |
| 4.1.3 confirm() 方法.....          | 61  | 练习.....                                 | 110 |
| 4.2 应知应会.....                    | 62  | 第7章 函数.....                             | 112 |
| 练习.....                          | 63  | 7.1 什么是函数.....                          | 112 |
| 第5章 运算符.....                     | 64  | 7.1.1 函数声明和调用.....                      | 112 |
| 5.1 JavaScript 运算符和表达式.....      | 64  | 7.1.2 返回值.....                          | 119 |
| 5.1.1 赋值运算符.....                 | 64  | 7.1.3 匿名函数变量.....                       | 121 |
| 5.1.2 优先级和结合性.....               | 65  | 7.1.4 闭包.....                           | 124 |
| 5.2 运算符类型.....                   | 68  | 7.1.5 递归.....                           | 125 |
| 5.2.1 算术运算符.....                 | 68  | 7.1.6 函数是对象.....                        | 129 |
| 5.2.2 快捷赋值运算符.....               | 69  | 7.2 调试技巧.....                           | 129 |
| 5.2.3 递增运算符和递减运算符.....           | 70  | 7.2.1 函数语法.....                         | 129 |
| 5.2.4 拼接运算符.....                 | 72  | 7.2.2 使用 try/catch 和 throw<br>捕捉异常..... | 131 |
| 5.2.5 比较运算符.....                 | 73  | 7.3 应知应会.....                           | 134 |
| 5.2.6 逻辑运算符.....                 | 78  | 练习.....                                 | 134 |
| 5.2.7 条件运算符.....                 | 83  | 第8章 对象.....                             | 136 |
| 5.2.8 位运算符.....                  | 84  | 8.1 什么是对象.....                          | 136 |
| 5.3 数字、字符串还是布尔值? 数据类型<br>转换..... | 87  | 8.1.1 对象及点语法.....                       | 137 |
| 5.3.1 parseInt() 函数.....         | 88  | 8.1.2 使用构造函数创建对象.....                   | 137 |
| 5.3.2 parseFloat() 函数.....       | 90  | 8.1.3 对象的属性.....                        | 139 |
| 5.3.3 eval() 函数.....             | 91  | 8.1.4 对象的方法.....                        | 140 |
| 5.4 特殊运算符.....                   | 92  | 8.2 类和用户自定义函数.....                      | 142 |
| 5.5 应知应会.....                    | 93  | 8.2.1 什么是类.....                         | 142 |
| 练习.....                          | 93  | 8.2.2 什么是 this.....                     | 142 |
| 第6章 条件选择.....                    | 95  | 8.2.3 内联函数作为对象的方法.....                  | 145 |
| 6.1 控制结构、块及复合语句.....             | 95  | 8.3 对象字面量.....                          | 146 |
| 6.2 条件.....                      | 95  | 8.4 操作对象.....                           | 149 |
| 6.2.1 if/else.....               | 96  | 8.4.1 with 关键字.....                     | 149 |
| 6.2.2 if/else if.....            | 98  | 8.4.2 for/in 循环.....                    | 151 |
| 6.2.3 switch.....                | 99  | 8.5 通过原型扩展对象.....                       | 152 |
| 6.3 循环.....                      | 102 | 8.5.1 使用 prototype 属性为对象<br>添加属性.....   | 154 |
| 6.3.1 while 循环.....              | 102 | 8.5.2 原型查找链.....                        | 155 |
| 6.3.2 do/while 循环.....           | 103 | 8.5.3 使用原型为对象添加方法.....                  | 157 |
| 6.3.3 for 循环.....                | 104 |                                         |     |

|                                           |     |                                        |     |
|-------------------------------------------|-----|----------------------------------------|-----|
| 8.5.4 所有对象都有的属性和方法 .....                  | 159 | 第 11 章 表单及输入设备的使用 .....                | 259 |
| 8.5.5 创建子类及继承 .....                       | 162 | 11.1 文档对象模型及传统 DOM 0 .....             | 259 |
| 8.6 应知应会 .....                            | 164 | 11.2 JavaScript 层次结构 .....             | 259 |
| 练习 .....                                  | 164 | 11.3 关于 HTML 表单 .....                  | 265 |
| 第 9 章 JavaScript 核心对象 .....               | 166 | 11.4 JavaScript 与 form 对象 .....        | 271 |
| 9.1 什么是核心对象 .....                         | 166 | 11.4.1 为表单及其输入类型 (控<br>件) 命名 .....     | 271 |
| 9.2 数组对象 .....                            | 166 | 11.4.2 表单中的传统 DOM .....                | 274 |
| 9.2.1 数组的声明及填充 .....                      | 167 | 11.4.3 为表单及按钮命名 .....                  | 277 |
| 9.2.2 Array 对象属性 .....                    | 171 | 11.4.4 提交填写完成的表单 .....                 | 281 |
| 9.2.3 关联数组 .....                          | 173 | 11.4.5 this 关键字 .....                  | 289 |
| 9.2.4 嵌套数组 .....                          | 175 | 11.4.6 submit() 及 reset()<br>方法 .....  | 291 |
| 9.3 数组方法 .....                            | 178 | 11.5 输入设备 (控件) 编程 .....                | 295 |
| 9.4 Date 对象 .....                         | 184 | 11.6 应知应会 .....                        | 324 |
| 9.4.1 Date 对象方法的使用 .....                  | 185 | 练习 .....                               | 324 |
| 9.4.2 操纵日期及时间 .....                       | 187 | 第 12 章 图片与链接 .....                     | 326 |
| 9.4.3 通过 prototype 属性自定义<br>Date 对象 ..... | 188 | 12.1 图片简介 .....                        | 326 |
| 9.5 Math 对象 .....                         | 190 | 12.1.1 重温 HTML 中的图片 .....              | 326 |
| 9.5.1 向上取整及向下取整 .....                     | 192 | 12.1.2 JavaScript 中的 image<br>对象 ..... | 328 |
| 9.5.2 生成随机数 .....                         | 193 | 12.2 回顾链接 .....                        | 329 |
| 9.5.3 wrapper 对象 .....                    | 194 | 12.3 图片地图 .....                        | 333 |
| 9.5.4 String 对象 .....                     | 195 | 12.3.1 使用 src 属性动态替换<br>图片 .....       | 336 |
| 9.5.5 Number 对象 .....                     | 205 | 12.3.2 预加载图片和 Image()<br>构造函数 .....    | 339 |
| 9.5.6 Boolean 对象 .....                    | 207 | 12.3.3 随机显示图片和 onClick<br>事件 .....     | 341 |
| 9.5.7 Function 对象 .....                   | 209 | 12.3.4 图片地图与 JavaScript .....          | 343 |
| 9.5.8 再论关键字 with .....                    | 210 | 12.4 缩放图片以适应窗口 .....                   | 345 |
| 9.6 应知应会 .....                            | 211 | 12.5 幻灯简介 .....                        | 347 |
| 练习 .....                                  | 212 | 12.5.1 带有控制按钮的简单幻灯<br>展示 .....         | 348 |
| 第 10 章 浏览器对象模型: BOM .....                 | 214 | 12.5.2 可单击的图片幻灯展示 .....                | 351 |
| 10.1 JavaScript 及浏览器对象模型 .....            | 214 | 12.6 动画和定时器 .....                      | 354 |
| 10.1.1 navigator 对象的使用 .....              | 215 | 12.6.1 变换图片位置 .....                    | 354 |
| 10.1.2 window 对象的使用 .....                 | 225 |                                        |     |
| 10.1.3 创建定时事件 .....                       | 231 |                                        |     |
| 10.1.4 Frame 的使用 .....                    | 240 |                                        |     |
| 10.1.5 location 对象 .....                  | 249 |                                        |     |
| 10.1.6 history 对象 .....                   | 253 |                                        |     |
| 10.1.7 screen 对象 .....                    | 256 |                                        |     |
| 10.2 应知应会 .....                           | 258 |                                        |     |
| 练习 .....                                  | 258 |                                        |     |

|                                     |     |                             |     |
|-------------------------------------|-----|-----------------------------|-----|
| 12.6.2 改变图片的高度和宽度                   |     | 13.7.1 捕获和冒泡 (滴漏和冒泡)        | 394 |
| 属性                                  | 355 | 13.7.2 事件对象属性               | 395 |
| 12.7 应知应会                           | 356 | 13.7.3 使用事件对象属性             | 396 |
| 练习                                  | 357 | 13.7.4 将事件传递给 JavaScript 函数 | 398 |
| 第 13 章 事件处理                         | 358 | 13.7.5 鼠标坐标                 | 400 |
| 13.1 事件处理函数介绍                       | 358 | 13.7.6 按键事件                 | 404 |
| 13.2 事件处理的内联模型                      | 358 | 13.8 处理事件的脚本模型              | 407 |
| 13.2.1 HTML 和事件处理函数                 | 359 | 13.9 应知应会                   | 412 |
| 13.2.2 设置事件处理函数                     | 361 | 练习                          | 412 |
| 13.2.3 返回值                          | 363 | 第 14 章 CSS 与 JavaScript     | 414 |
| 13.2.4 JavaScript 对象方法和事件           | 364 | 14.1 什么是 CSS                | 414 |
| 13.3 处理窗口事件或框架事件                    | 366 | 14.2 什么是样式表                 | 414 |
| 13.3.1 onLoad 和 onUnload 事件         | 366 | 14.2.1 什么是兼容 CSS 的浏览器       | 415 |
| 13.3.2 onFocus 和 onBlur 事件处理函数      | 368 | 14.2.2 样式表的工作原理             | 415 |
| 13.3.3 onResize 事件处理函数              | 371 | 14.3 CSS 程序结构               | 417 |
| 13.4 鼠标事件处理                         | 373 | 14.3.1 注释                   | 417 |
| 13.4.1 怎样使用鼠标事件                     | 374 | 14.3.2 组合                   | 417 |
| 13.4.2 鼠标事件和图像——翻转                  | 376 | 14.4 常用样式表属性                | 418 |
| 13.4.3 通过鼠标事件创建幻灯片                  | 376 | 14.4.1 计量单位                 | 420 |
| 13.5 处理链接事件                         | 379 | 14.4.2 使用颜色                 | 421 |
| 13.6 处理表单事件                         | 380 | 14.4.3 使用字体                 | 423 |
| 13.6.1 按钮                           | 381 | 14.4.4 使用文本                 | 425 |
| 13.6.2 表单和按钮中的 this 关键字             | 382 | 14.4.5 使用背景和图片              | 427 |
| 13.6.3 表单和 onClick 事件处理函数           | 383 | 14.4.6 使用边距和边框              | 429 |
| 13.6.4 表单和 onFocus、onBlur 事件处理函数    | 384 | 14.5 样式表类型                  | 432 |
| 13.6.5 表单和 onChange 事件处理函数          | 386 | 14.5.1 嵌入式样式表和<style> 标签    | 432 |
| 13.6.6 表单和 onSubmit 事件处理函数          | 387 | 14.5.2 内联样式和<style> 属性      | 435 |
| 13.6.7 HTML 事件处理函数和 JavaScript 事件方法 | 391 | 14.6 链接的样式表                 | 436 |
| 13.6.8 onError 事件                   | 393 | 14.6.1 <link> 标签            | 436 |
| 13.7 event 对象                       | 394 | 14.6.2 使用@import 导入         | 438 |
|                                     |     | 14.7 创建样式类                  | 438 |
|                                     |     | 14.7.1 用类设定简单表格样式           | 439 |
|                                     |     | 14.7.2 使用特定类选择器             | 441 |
|                                     |     | 14.8 ID 选择器和 ID 属性          | 443 |
|                                     |     | 14.9 使用<span> 标签重写或添加样式     | 444 |

|         |                                    |     |         |                                     |     |
|---------|------------------------------------|-----|---------|-------------------------------------|-----|
| 14.9.1  | <span>标签和 style 属性                 | 445 | 15.7    | DOM 的修改 (追加、复制和删除节点)                | 493 |
| 14.9.2  | <span>标签和 class 属性                 | 446 | 15.7.1  | innerHTML 属性和元素的内容                  | 493 |
| 14.9.3  | 继承和上下文选择器                          | 447 | 15.7.2  | 更改元素的内容                             | 495 |
| 14.10   | 定位元素和层                             | 449 | 15.7.3  | 使用 DOM 创建新元素                        | 496 |
| 14.10.1 | 绝对定位                               | 450 | 15.7.4  | 在节点前插入                              | 498 |
| 14.10.2 | <div>容器                            | 455 | 15.7.5  | 为节点创建属性                             | 499 |
| 14.10.3 | 绝对定位                               | 455 | 15.7.6  | DOM 回顾: 创建博客                        | 501 |
| 14.10.4 | 相对定位                               | 457 | 15.7.7  | 使用 DOM 创建表格                         | 504 |
| 14.10.5 | z 索引和三维                            | 458 | 15.7.8  | 复制节点                                | 507 |
| 14.11   | 如何与 JavaScript 融合                  | 459 | 15.7.9  | 删除节点                                | 512 |
| 14.11.1 | 什么是 DHTML                          | 459 | 15.7.10 | 利用节点实现文本滚动                          | 515 |
| 14.11.2 | JavaScript 如何操作样式表                 | 459 | 15.8    | 事件处理和 DOM                           | 517 |
| 14.11.3 | style 对象                           | 462 | 15.8.1  | HTML 内联方式                           | 517 |
| 14.11.4 | className 属性                       | 470 | 15.8.2  | 脚本方法                                | 517 |
| 14.11.5 | 下拉菜单和提示框                           | 472 | 15.8.3  | DOM 方法                              | 518 |
| 14.12   | 应知应会                               | 477 | 15.8.4  | 冒泡和捕获                               | 518 |
| 练习      |                                    | 478 | 15.9    | W3C 模型中的事件监听器                       | 522 |
| 第 15 章  | W3C DOM 与 JavaScript               | 479 | 15.9.1  | 添加事件                                | 522 |
| 15.1    | W3C DOM                            | 479 | 15.9.2  | 注册多个事件                              | 524 |
| 15.2    | 理解 DOM 节点                          | 480 | 15.9.3  | 删除事件                                | 526 |
| 15.3    | 节点                                 | 481 | 15.9.4  | Microsoft Internet Explorer 中的事件监听器 | 528 |
| 15.3.1  | 父节点和子节点                            | 482 | 15.9.5  | 回顾事件属性                              | 530 |
| 15.3.2  | 兄弟节点                               | 482 | 15.10   | 不唐突的 JavaScript                     | 533 |
| 15.3.3  | nodeName 和 nodeType 属性             | 483 | 15.11   | 应知应会                                | 539 |
| 15.3.4  | 空白 bug                             | 483 | 练习      |                                     | 540 |
| 15.4    | 遍历 DOM                             | 484 | 第 16 章  | cookie                              | 543 |
| 15.5    | DOM 查看器                            | 486 | 16.1    | 什么是 cookie                          | 543 |
| 15.6    | 查询 DOM 的快捷方式                       | 487 | 16.1.1  | cookie 的组成                          | 545 |
| 15.6.1  | document.getElementById() 方法       | 487 | 16.1.2  | cookie 的属性                          | 546 |
| 15.6.2  | document.getElementsByTagName() 方法 | 490 | 16.2    | 使用 JavaScript 创建 cookie             | 547 |
| 15.6.3  | 对应于 HTML 属性的 JavaScript 属性         | 491 | 16.2.1  | cookie 对象                           | 548 |
|         |                                    |     | 16.2.2  | 设置 cookie 属性                        | 548 |
|         |                                    |     | 16.2.3  | 创建 cookie                           | 550 |
|         |                                    |     | 16.2.4  | 从服务器获取 cookie 信息                    | 552 |
|         |                                    |     | 16.2.5  | 删除 cookie                           | 554 |

|                       |     |                                   |     |
|-----------------------|-----|-----------------------------------|-----|
| 16.2.6 使用浏览器删除 cookie | 556 | 17.5.8 验证信用卡号                     | 610 |
| 16.3 应知应会             | 557 | 17.5.9 汇总                         | 616 |
| 练习                    | 557 | 17.6 应知应会                         | 619 |
| 第 17 章 正则表达式和模式匹配     | 558 | 练习                                | 619 |
| 17.1 什么是正则表达式         | 558 | 第 18 章 Ajax(及 JSON)简介             | 621 |
| 17.2 创建正则表达式          | 560 | 18.1 为什么要用 Ajax                   | 621 |
| 17.2.1 字面量方式          | 560 | 18.2 为什么最后才提到 Ajax                | 623 |
| 17.2.2 构造函数           | 560 | 18.3 建立 Ajax 通信的步骤                | 623 |
| 17.2.3 测试正则表达式        | 561 | 18.3.1 步骤 1: 创建 XMLHttpRequest 对象 | 623 |
| 17.2.4 RegExp 对象的属性   | 564 | 18.3.2 步骤 2: 初始化对象                | 625 |
| 17.3 使用字符串的正则表达式方法    | 566 | 18.3.3 向服务器发送请求                   | 628 |
| 17.3.1 match() 方法     | 566 | 18.3.4 步骤 3: 监控服务器响应状态            | 628 |
| 17.3.2 search() 方法    | 567 | 18.3.5 使用回调函数处理响应                 | 630 |
| 17.3.3 replace() 方法   | 568 | 18.3.6 浏览器缓存问题                    | 632 |
| 17.3.4 split() 方法     | 569 | 18.4 结合起来                         | 632 |
| 17.4 获取控制——元字符        | 571 | 18.4.1 使用 Ajax 从文件中获取文本           | 638 |
| 17.4.1 点元字符           | 574 | 18.4.2 使用 Ajax 从文件中获取 XML         | 641 |
| 17.4.2 字符类            | 575 | 18.4.3 Ajax 和表单                   | 644 |
| 17.4.3 元符号            | 578 | 18.5 Ajax 和 JSON                  | 651 |
| 17.4.4 重复模式匹配元字符      | 580 | 18.5.1 JSON 数据结构                  | 652 |
| 17.4.5 锚元字符           | 587 | 18.5.2 使用 JSON 的步骤                | 652 |
| 17.4.6 或模式            | 592 | 18.5.3 JSON 综合示例                  | 655 |
| 17.5 用正则表达式验证表单       | 596 | 18.5.4 解决 eval() 安全问题             | 658 |
| 17.5.1 检查空字段          | 596 | 18.6 使用 Firebug 调试 Ajax           | 662 |
| 17.5.2 检查邮政编码         | 598 | 18.7 应知应会                         | 664 |
| 17.5.3 检查字母数据         | 600 | 练习                                | 665 |
| 17.5.4 删除多余字符         | 601 |                                   |     |
| 17.5.5 有效社会安全号码验证     | 604 |                                   |     |
| 17.5.6 验证电话号码         | 606 |                                   |     |
| 17.5.7 验证电子邮件地址       | 608 |                                   |     |

# JavaScript简介



## 1.1 JavaScript 是什么

JavaScript是一种流行的通用脚本语言。使用JavaScript可以为呆板的Web页面注入活力，因为基于JavaScript可以实现用户与页面的交互，并响应页面上发生的事件。JavaScript被形容为Web页面之间的粘合剂<sup>①</sup>。不仅是商业站点，几乎所有的Web页面中都有JavaScript代码（见图1-1）。

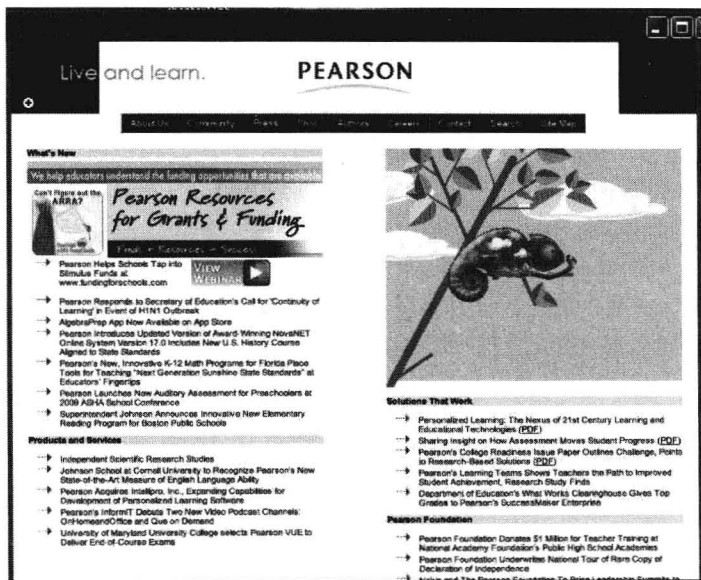


图1-1 通过JavaScript赋予了生命力的一个动态Web页面（例如，将鼠标移动到箭头后面的文字上，相应的文字就会变成带下划线的超链接形式）

① 不过JavaScript之父Brendan Eich认为不但如此。他在名为Innovators of the Net: Brendan Eich and JavaScript的文章中写道：“‘Web页面之间的粘合剂’这种说法的确琅琅上口，不过对于JavaScript真正发挥的作用来讲却描述得不够公正。粘合剂会凝结变硬，但JavaScript要比粘合剂更具动态性。JavaScript更像催化剂，它可以触发反应，并使事件继续执行下去。”

LiveScript是JavaScript的前身，由Netscape的Brendan Eich于1995年开发完成，并同Netscape Navigator 2.0 beta版一同发布。从LiveScript开始，JavaScript就成为了一种可以赋予静态HTML页面生命力的脚本语言。JavaScript能运行在绝大多数平台上，并且与硬件无关。它是一种客户端语言，运行于用户的浏览器中而不是服务器上<sup>①</sup>。微软的Internet Explorer和Mozilla Firefox是我们比较常见的浏览器<sup>②</sup>，它们都直接内置了JavaScript（但JavaScript的应用不仅限于浏览器）。在语法上，JavaScript同C、Perl和Java类似。例如，if语句同while和for循环几乎完全相同。同Perl一样，JavaScript是一种解释语言，不是编译语言。

由于JavaScript是同浏览器关联的，所以它跟HTML紧集成。HTML在浏览器中由专门的网络库和图形渲染器负责处理，而JavaScript代码则由浏览器内嵌的JavaScript解释器执行。当浏览器请求页面的时候，服务器会通过网络将包括HTML和JavaScript语句在内的整个文档的全部内容发送到客户端。在页面加载的时候，HTML内容会被按行读取并显示，遇到JavaScript起始标签后，由JavaScript解释器进行解释。到达JavaScript结束标签后，继续进行HTML处理。

由浏览器处理的JavaScript被称作客户端JavaScript。虽然JavaScript大都被当做客户端脚本语言来使用，不过也可以在Web浏览器以外的其他环境中使用。Netscape创建了服务器端的JavaScript，以CGI语言（如Python或者Perl）的形式进行编程。不过本书的重点会放在JavaScript最常见的使用方式上——运行于客户端，也就是浏览器上。

## 1.2 JavaScript 不是什么

JavaScript不是Java。“Java同JavaScript就好比是汽车（Car）和地毯（Carpet）”<sup>③</sup>，这种说法尽管有些极端，但它告诉我们Java和JavaScript是非常不同的两种语言。

Java是由Sun公司开发的，而JavaScript由Netscape开发。Java应用可以脱离Web页面存在，而JavaScript程序要嵌于Web页面中，并且在浏览器窗口中执行<sup>④</sup>。Java是一种强类型语言，有严格的语法规则；而JavaScript是松散型语言并且更灵活。Java数据类型必须先声明再使用；而JavaScript中诸如变量、参数和函数返回值类型等都不需要声明。Java程序是需要编译的；而JavaScript是被浏览器内置的JavaScript引擎解释执行的。

JavaScript不是HTML，但JavaScript代码可以嵌入HTML文档，并用HTML的标签括起来。JavaScript有自己的一套语法规则，需要按照特定的方式编写。JavaScript与HTML是两码事，但却能在自身的语句中包含HTML内容。我们会在以后的讨论中进一步解释这些内容。

JavaScript不能用来读写客户端机器上的文件，写入cookie除外（详见第16章）。不能用JavaScript来向服务器写入或者存储文件。JavaScript也不能用来打开或者关闭其他应用程序的窗

---

① 实际上。也存在服务器端实现的JavaScript版本，如Rhino。本书后面也简单提及。——编者注

② 目前主流浏览器Internet Explorer、Firefox、Chrome、Opera、Safari中，都内置了JavaScript引擎。——译者注

③ 出自Usenet讨论组，以及*Beginning JavaScript with DOM Scripting and Ajax*第4页。Christian Heilmann, APRESS, 2006。

④ JavaScript解释器通常嵌入在Web浏览器中，但不限于浏览器。服务器和其他应用程序也都可以使用JavaScript解释器。

口。此外JavaScript还不能读取其他服务器发来的Web页面。

JavaScript是基于对象的,但并不严格地面向对象。这是由于它不像Java和C++一样,支持继承和类这样传统的面向对象机制。同时Java和C++中的private、protected、public也不能应用于JavaScript的方法上。

JavaScript不是唯一一种可以嵌入到应用程序中的语言。拿微软开发的VBScript来说,它同JavaScript很类似,但是也嵌入到了微软的Internet Explorer中。

## 1.3 JavaScript 的用处

JavaScript程序被用来侦测和响应来自用户的事件,例如鼠标移到超链接或者图片之上等。这些用户事件可以通过辅助导航、滚动消息和翻转、对话框、动态图像等方式来优化Web站点。有了JavaScript,还可以在页面解析的同时动态控制页面的显示效果。在没有任何网络数据传输的情况下,可以在表单提交至服务器之前就对用户所填内容进行验证。JavaScript还可用来检测用户的浏览器是否装有插件,并在必要的情况下将用户引导至新的页面来获取插件。JavaScript拥有字符串函数,支持正则表达式,可以此来检测电子邮件地址、社会保障号码、信用卡数据等的有效性。JavaScript是一种编程语言,其核心语言描述了变量、数据类型、循环控制、if/else语句、switch语句、函数和对象等的基本结构<sup>①</sup>。JavaScript可用来进行算术计算,操作日期和时间,并与数组、字符串和对象协同操作等。JavaScript处理用户发起的事件、设置计时器、动态更改内容和样式。JavaScript还能读写cookie值,并基于cookie值动态生成HTML。

## 1.4 JavaScript 及其在 Web 页面中的位置

### 分析图

#### 1. 角色

图1-2中的角色是Web页面整个生命周期中所涉及的如下应用。

- (1) 浏览器 (Firefox、Internet Explorer、Safari、Opera)。这是执行JavaScript的地方。
- (2) 网络 (HTTP)。
- (3) 服务器 (Apache、Windows IIS、Zeus)。
- (4) 服务器模块 (PHP、ASP.NET、ColdFusion、Java servlet)。
- (5) 外部文件以及/或者数据库 (MySQL、Oracle、Sybase)。

#### 2. 步骤

图1-2展示了一个Web页面从客户端发起请求到获得响应的整个生命周期。

<sup>①</sup> JavaScript语言的最新版是1.8.1。Mozilla公司和微软公司均提供支持。

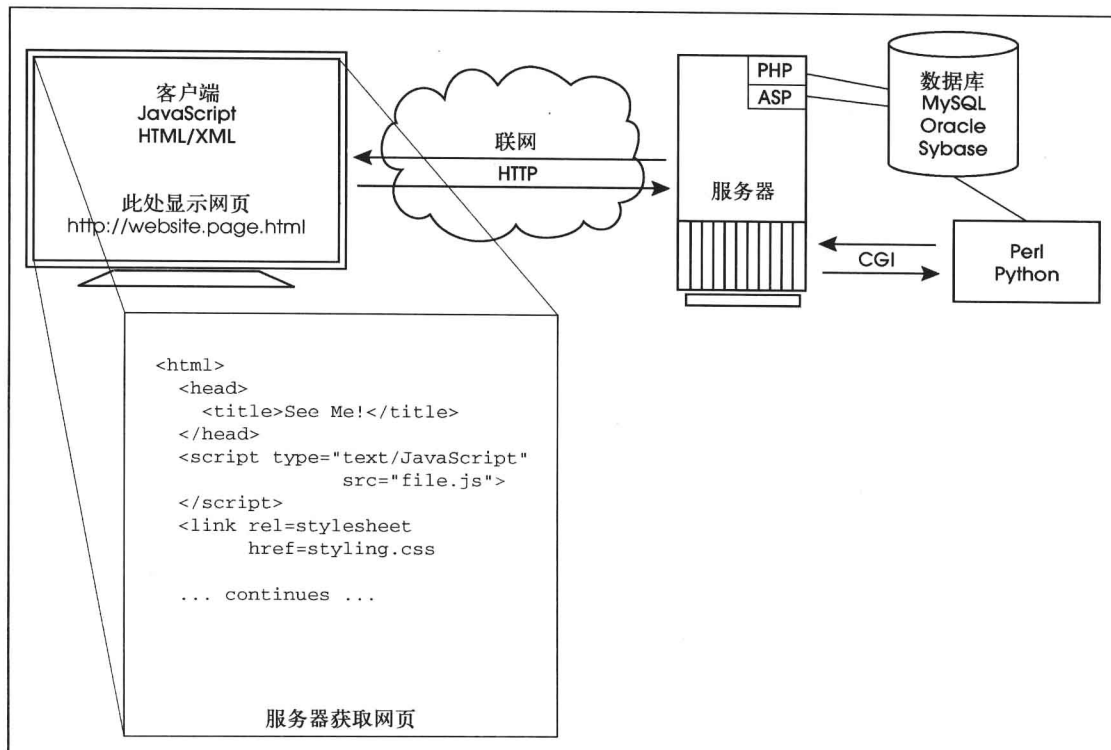


图1-2 一个典型Web页面的生命周期

(1) 在图的左侧,我们看到请求是从客户端或者说浏览器发起的。用户通过在浏览器的URL地址栏中输入Web站点的地址来发起请求。“请求”是通过HTTP (Hypertext Transfer Protocol, 超文本传输协议) 提交至服务器的。另一侧的Web服务器作出响应,简单地将文件返回给客户端浏览器。接着浏览器就会提取HTML标签,格式化页面并显示,然后等待另一个请求。如果页面包含JavaScript标签,那么JavaScript解释器就会基于用户触发的事件来执行这些代码,这些事件包括单击按钮、将鼠标移动到超链接或者图片上、提交表单等。正是有了JavaScript,页面才具备了交互性。JavaScript检测所有在页面上发生的事件,并进行响应。它可以对表单填写、反馈、动画、幻灯片放映和多媒体等进行处理。能响应按键、图片鼠标悬停、用户提交表单等事件。JavaScript可以读取cookie并验证数据。可以动态修改HTML表格的某单元格,修改某段落中的文字,或者为某个列表增加一行可选项。不过不是所有的事情JavaScript都能胜任。它不能关闭其他程序打开的窗口,不能查询数据库,不能更新上传文件控件的值,也不能对服务器端的文件进行写操作。在JavaScript解释器完成了自己的任务后,页面就完全显示出来了,此时便可再一次发起请求。在浏览器和服务器的往返被称作请求/响应循环,是Web运转的基础。

(2) 在客户端和服务端之间的云代表网络。这个网络可以像包含数万亿台电脑的Internet一样非常巨大,也可以像一个公司的局域网一样小,还可以是一台个人台式电脑或者手持设备的无线网络。用户不必关心网络的大小——这对用户是透明的。客户端同服务器传送文件的协议被称作HTTP协议。

(3) 服务器端包括支持HTTP协议的Web服务器,例如Apache,微软公司的IIS和lighttpd<sup>①</sup>等。Web服务器通常既能接收基于Web的请求,也能进行相应的响应,是一种泛型程序。在大多数情况下,所谓的响应仅仅是从服务器本地文件系统中获取某个文件。对于动态Web页面而言,需要处理很多JavaScript所不能胜任的工作,例如处理网页表单中的内容,发送电子邮件,启动一个会话(session)或者是连接数据库等。这些事件都要在服务器端完成,Web服务器把查询请求转发给对应的助手程序来处理。像Apache、IIS等Web服务器都有一组助手应用程序来处理各种语言。助手程序有CGI/Perl脚本编译器之类的外部程序,也有ColdFusion, ASP.NET和PHP脚本处理器等这种内嵌在服务器中的应用。举个例子,如果Web服务器收到一个PHP的请求,会先查找哪个助手程序是用来处理PHP请求的,然后把请求转发给PHP模块,最后等待这个模块返回结果。

## 1.5 Ajax 是什么

Ajax是Asny chronous JavaScript and XML(异步JavaScript和XML)的缩写形式,这种说法是在2005年由Jesse James Garrett首创的。Ajax并不新鲜,早在1996年就有了。Ajax用来在无需等待服务器端响应的情况下创建快速的交互。在图1-2的Web生命周期中可以看到,浏览器向服务器发出了一项请求,然后等待响应。通常会在地址栏上显示一个转动着的轮子形状的载入图标,提醒用户页面正在加载。在用户等待的同时,浏览器也在一起等待,然后等到每个请求序列都完成后,还要继续等待整个页面全部重新加载一次才能看到新页面的内容。Ajax可以在无需刷新整个页面的情况下实现数据在浏览器和服务端间的传送,页面中只有一部分内容被替换掉。即使用户在进行滚动、缩放、填写表单等操作的同时,请求照样可以发出。由于交互是异步的,所以请求会无需任何等待时间而得到最快速的响应。Ajax的应用实例有很多,如Ajax Stock Qutos Ticker(SentoSoft LTD)、Flickr的图片存储和显示、Gmail、Google Suggest等,而最棒的一个例子莫过于maps.google.com上的Google Maps了(见图1-3)。

在浏览Google Maps网站的时候,用户会感受到完整而快速的交互性。用户可以对地图进行放大、缩小、随意拖动,在地图上获取两点间的位置信息、观察地形、查看交通状况、浏览卫星图片等。在第18章我们会详细讨论这种技术的实现细节,不过暂时我们可以将它视为增强型的JavaScript。

① Lighttpd是众多OpenSource轻量级的Web服务器中较为优秀的一个。——译者注

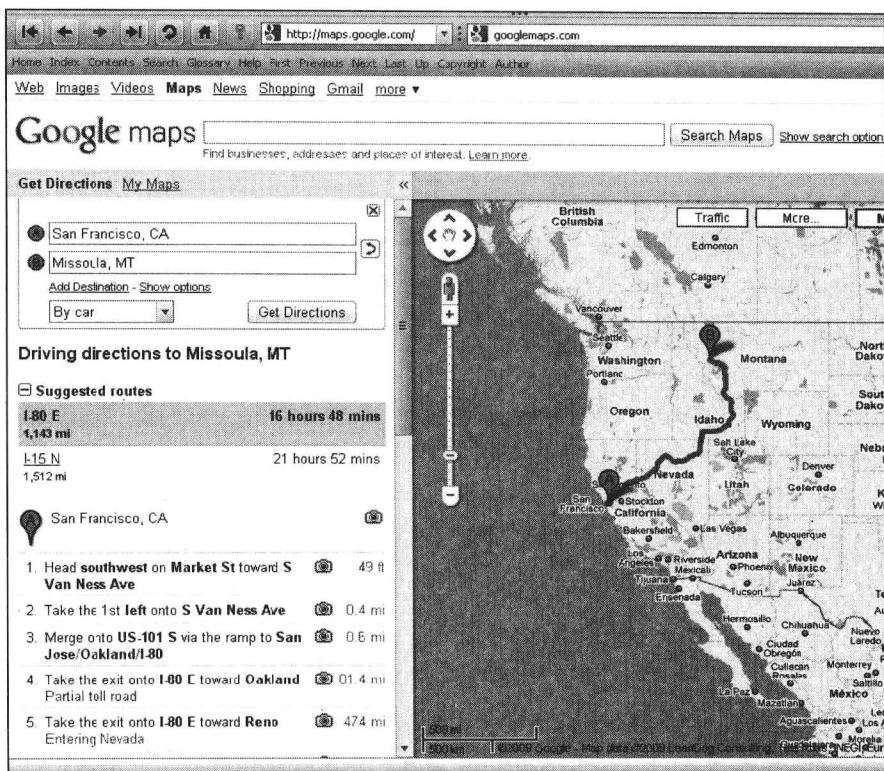


图1-3 Google使用Ajax进行交互 (©2010 Google)

## 1.6 JavaScript 是什么样子的

示例1-1演示了一小段JavaScript代码。在Web页面上显示简单的HTML表格单元格，其中有一条滚动消息（见图1-4）。没有JavaScript的话，这段消息就是静态的。而有了JavaScript以后，消息会持续不断地沿着屏幕滚动，为死气沉沉的页面带来了生机。以后我们会详细解释这段代码，在这里，只是用来告诉大家JavaScript代码是什么样子。注意<script></script>这两个加粗了的标签，在这两个标签中间就是能产生滚动效果的JavaScript代码。不用多久，我们就能看懂并编写这类脚本了。

### 示例1-1

```
<html>
<head><title>Dynamic Page</title>
<script type="text/javascript">
    // 这就是JavaScript。耐心点，不用多久你就能写出比这更好的代码了
    var message="Learning JavaScript will give your Web
    page life!";
```