TURING 图灵程序设计丛书 Web 开发系列

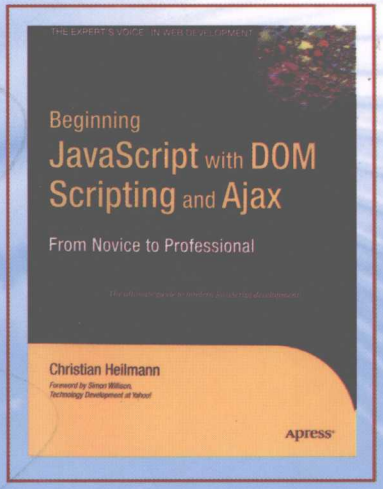
Apress®

Beginning JavaScript with DOM Scripting and Ajax

深入浅出 JavaScript

[德] Christian Heilmann 著
牛海彬 等译

- 世界级 JavaScript 程序员力作
- 全面、实用、丰富的经典示例
- 深入揭示现代 JavaScript 编程理念



人民邮电出版社
POSTS & TELECOM PRESS


深入浅出
JavaScript

TURING 图灵程序设计丛书 Web

深入浅出JavaScript

Beginning JavaScript with DOM Scripting and Ajax

[德] Christian Heilmann 著

牛海彬 等译

人民邮电出版社
北 京

图书在版编目（CIP）数据

深入浅出 JavaScript /（德）海尔曼（Heilmann, C.）著；
牛海彬等译. —北京：人民邮电出版社，2008.4
（图灵程序设计丛书）
ISBN 978-7-115-17168-9

Ⅰ. 深… Ⅱ. ①海…②牛… Ⅲ. Java 语言—程序设计
Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 174306 号

内 容 提 要

本书是一部优秀的、注重实践的 JavaScript 教程。作者首先概览了 JavaScript，包括它的语法、良好的编码习惯、DOM 编程原则等；然后构建了 JavaScript 工具包，包括动态操作标记、使用 CSS 和 DOM 修改页面风格、验证表单、处理图像等；接着通过一个完整的案例研究阐明了如何使用多种 JavaScript 技术协同工作；最后单独设计一章来讲述第三方示例，演示了 YUI 和 jQuery JavaScript 库的使用。

本书适合初级和中级水平的 JavaScript 开发人员阅读，可作为高等院校计算机专业的 JavaScript 课程教材。

图灵程序设计丛书

深入浅出 JavaScript

◆ 著 [德] Christian Heilmann
译 牛海彬 等
责任编辑 傅志红 张继发

◆ 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 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
三河市海波印务有限公司印刷
新华书店总店北京发行所经销

◆ 开本：800×1000 1/16
印张：25.25
字数：597 千字 2008 年 4 月第 1 版
印数：1–5 000 册 2008 年 4 月河北第 1 次印刷

著作权合同登记号 图字：01-2006-5095 号

ISBN 978-7-115-17168-9/TP

定价：55.00 元

读者服务热线：(010)88593802 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版 权 声 明

译 者 序

随着Ajax技术的兴起，JavaScript重新回到了大众的视野。Ajax是Asynchronous JavaScript and XML（异步JavaScript和XML）的缩写，使用它，不刷新整个页面也可以和Web服务器交换数据，这样提高了传输效率，增强了用户体验，搜狐和新浪的博客系统就是比较典型的成功案例。

JavaScript是一种解释性的脚本语言，它是网景公司于1996年发明的。当时把它嵌入到Netscape Navigator（NN）2.0浏览器中，需要解释器来读取和执行添加到.html页面的JavaScript代码。随着因特网的发展，现在几乎所有的浏览器都支持JavaScript，也就是说有上亿台电脑安装了JavaScript的运行环境，这说明它有着广泛的用户群。

JavaScript刚开始是作为一种脚本语言推出的，因此人们往往错误地认为这种技术很简单，其实它是一种功能强大而且比较复杂的语言，在企业级开发和Web开发中起着非常重要的作用。我做了多年的企业级开发和Web开发，发现真正会用JavaScript的人并不多，大部分人也只是复制和粘贴，很少有人能写出自己的JavaScript应用，我想这与大家对它的不重视可能有很大关系。

译完本书，我感觉作者真的很棒，他没有介绍一些华而不实的例子，而是从基础开始教你如何编写高质量的JavaScript代码。除此以外，本书还包含了许多最新的JavaScript技术，包括在Ajax、RSS以及其他Web 2.0技术中的应用。它不只适合JavaScript的初学者，而且适合希望对JavaScript技术有深入研究的人。本书还可作为计算机专业开设的JavaScript课程的教材。

本书第3章和第9章由屈晓光翻译，第7章由易继开翻译，感谢他们与我合作完成了本书。

感谢人民邮电出版社图灵公司的傅志红编辑和谢工给予我的帮助和指导。

感谢我的父母和许多朋友给予我的帮助和支持。

由于时间比较仓促，加上译者学识有限，译文中错误疏漏在所难免，欢迎大家指正。我的电子邮件和博客地址分别是pleasechess@126.com和http://newhappy2008.blog.sohu.com，欢迎大家来信或留言，把问题反馈给我。

牛海彬

2007年年底

序

也许从来没有什么时候比现在学习JavaScript更令人兴奋的了。经历了多年的浏览器之争后，JavaScript终于成为Web开发人员必备的工具。不再只是广告和恶作剧的手段，现在它已是下一代Web应用程序非常有价值的组成部分。

是什么原因导致它突然又备受关注呢？第一个原因完全是实践的结果：浏览器的改进最终使得编写跨浏览器平台的JavaScript可行。第二个原因具有更多的革命性：Ajax作为一种使用新名称的老技术，使得客户端代码可以直接连接到服务器上而不用重新刷新整个页面。这种简单的功能使Web应用程序开发更开阔，它启用了创新的界面，并且戏剧性地改变着用户对网页界面行为表现的期望。

越来越多的人认识到，JavaScript不是一种玩具语言，这进一步促进了人们对它的接受。尽管它还有许多不足之处，但是在其非常简单的外表之下有着许多现代语言中都没有的强大功能：闭包、原型继承以及对函数式编程风格的广泛支持。这样一种灵活的语言现在被安装到了数以亿计的计算机上，这是值得庆祝的。

“只是因为你可以做”并不意味着就应该那样去做。并不是所有的浏览器都生而平等，可访问性（人和各种其他设备都可以访问）仍然是Web开发非常重要的一个方面。理解这些问题和有关渐进增强的技术都是JavaScript学习非常重要的一部分。

JavaScript开发所面临的挑战是非常巨大的。

浏览器经常会背离现有的标准规范，而且伪标准非常常见且经常无法避免。

不断开发出来的功能强大的新应用程序可能会暴露许多隐藏多年的浏览器bug。这些应用程序本身就很复杂，维护大量的代码又会引起新的问题。

幸运的是，应对这个挑战的全球JavaScript社区已经出现了。大量的代码和资源都在等待着无畏的开发人员使用，但是这些财富隐藏的价值只有通过对潜在平台的一致理解才能发掘出来。本书将会为你提供这方面的知识。

作为这个社区中一名长期的导师和领导者，Christian是通向这个复杂世界的理想引路人。本书还包含了许多只有通过多年的经验积累才能获得的智慧。

Christian会教你以一种高雅、可靠且优美的方式应用JavaScript，这会使你的用户非常满意并且会给你的同事留下深刻的印象。

Simon Willison
著名Web开发人员
Django框架开发者之一

前　言

如果你想从零开始学习JavaScript——它的含义、功能，以及如何综合使用它和其他技术（如CSS和HTML），那么你就选对书了。如果你已经对JavaScript有了一定的经验，但还想了解一些在最新的知识，那么你也选对了书——在最近几年里，JavaScript开发已经改变了许多。

在20世纪90年代中后期，JavaScript第一次开始应用到Web开发的时候（在1996年首次被Netscape 2支持），它很快就遭遇了许多批评和抱怨。这有很多原因——即使最好的时候，浏览器的支持也很一般，而最糟的时候，针对不同的浏览器（浏览器之争的主角是Netscape 4和IE 4）需要以不同的方式实现各种不同的功能。这就导致开发人员如果想获得跨浏览器平台的支持，就不得不编写完全不同的网站版本，或者纵容杂乱的代码分叉。

此外，还有很多人为因素。JavaScript的坏名声，开发人员和浏览器生产商的过错几乎要对半开。那时开发人员喜欢用JavaScript实现看上去很酷的特效，但是引起了许多可用性和可访问性的问题（这就是所谓DHTML，那时的一个流行词指的是使用JavaScript、CSS和HTML应用程序来产生动态效果）。如果由于某种原因JavaScript不可用或者用户要使用屏幕阅读器，那么网页就会完全失效。而且许多Web开发人员在不懂实际工作机理的情况下，就把一些脚本复制并粘贴到自己的网站中，导致了更多说不清楚的可用性和代码维护的问题。

正如我在前面所说，现在事情已经发生了改变。浏览器的支持性现在已处于可控级别，现代的浏览器大都使用DOM（文档对象模型）和其他构造的相同实现，而且现代技术也很大程度上都考虑了可访问性等。学完本书后，你会发现像DOM脚本这样的现代技术也都是围绕着下面的假设来建立的：既要将结构（在标记文件中）和表现（在CSS文件中）分离，也要分离JavaScript文件中的行为（而不是遍布在标记文档中）。JavaScript并不是只能做恶，它可以用来对网站编码，使用户体验更丰富，而在JavaScript不可用的时候也不会完全失效。这叫做分离式JavaScript（unobtrusive JavaScript）[①]——JavaScript增强应该被看作是那些能够使用它们的用户的一种额外好处，而不是运行一个网站所必需的特性。

如果以前使用过JavaScript，那么阅读本书时你要准备好接受一种新思想。如果你完全是刚刚学JavaScript，那么可以松一口气，很幸运你没有刚说过的JavaScript早期开发的经历！

---

① unobtrusive JavaScript 是现代 JavaScript 编程的核心思想，主要通过将 JavaScript 与表现层和结构层分离，从而实现平稳退化不影响禁用了某些功能的用户的目的。也译为“不唐突的（或非干扰的）JavaScript”。——编者注

本书内容

JavaScript可能是在Web开发中被低估且被滥用最严重的语言，但是如果正确使用，它会是一种非常有价值的工具。在接下来的章节中，我们会介绍JavaScript的基础和现代的JavaScript技术，包括用于动态行为和样式控制以及事件处理的DOM编程。接着，我们会学习一些JavaScript最基础的应用，包括数据验证、图片和窗口操作，以及表单和导航菜单的动态增强。

接下来，我们把注意力转移到与JavaScript相关的、可能是目前最热的一个词语——Ajax。Ajax代表异步JavaScript与XML（Asynchronous JavaScript and XML），这有点儿名不符实，因为这种技术不必包括XML，而且可能经常和HTML一起使用。但是不要只关注这一点，它主要指在网页上创建运行的动态功能，因为不用刷新整个页面，网页的各小部分就可以更新，例如，在线的邮件应用程序中的联系人信息（Gmail是我们最容易想到的例子）。现在实现Ajax最常用的方式就是使用XMLHttpRequest（XHR）对象。它非常流行，因为它允许创建拥有丰富功能且其外观和运行方式都和桌面应用程序类似的Web应用程序。但是Ajax确实也带来了它自己的一系列问题，这个也会在书中涉及。

接下来是一个综合的案例研究，它展示了一个成熟的含有现代JavaScript增强的Web应用程序。

最后，第11章重点讲解现代JavaScript开发的另一个重要的方面——使用第三方JavaScript解决方案。当你开发JavaScript应用程序的时候，不需要每次都对所有的东西从头编码。和创建自己可复用的对象和函数一样（这个会在本书前面几章中讲到），在Web上还有许多第三方资源，你可以下载并在自己的应用程序中使用。从函数库到成熟的API（应用编程接口），许多资源都可以供你使用。其中，我们重点介绍了jQuery、Google Maps API、Yahoo API等。

社区和支持

JavaScript是什么？你用它来做什么？询问拥有不同技术背景或面向设计的开发人员的时候，你很可能会得到完全不同的答案。本书会教你如何成为一个可以和所有这些开发人员一起工作的JavaScript开发人员，而且通过使用JavaScript来增强网站、构造Web应用程序，甚至不用强迫用户改变他的使用方式或者进行硬件设置就能扩展软件，使他们对你刮目相看。

本书所提供的所有代码例子都可以在网站http://www.beginningjavascript.com下载和测试，在那里你还会发现更多的信息、勘误以及其他例子（Apress出版社在http://www.apress.com上也提供了勘误表和可下载的代码）[①]。

但是遇到问题时该怎么办呢？你有3种选择。第一，可以通过网站（http://wait-till-i.com）通知我或者把问题通过邮件发送到Apress（详细的联系地址在http://www.apress.com上）。

第二，在因特网的一些JavaScript论坛里求助，其中比较好的有下列几个。

- evolt的thelist论坛：http://lists.evolt.org/mailman/listinfo/thelist。
- Mozilla JavaScript论坛：http://developer.mozilla.org/en/docs/JavaScript。

① 图灵网站为本书中文版提供了配套网页，有代码下载和勘误表以及其他资源。——编者注

- Webdeveloper.com JavaScript论坛：http://www.webdeveloper.com/forum/forumdisplay.php?f=3。
- comp.lang.javascript FAQ：http://jibbering.com/faq/。

这些论坛经常有许多像你这样寻找问题答案的人光顾，还有许多非常有经验的JavaScript高手，他们乐于帮助社区里的人解决问题。要确保你的问题经过了思考，不要只粘贴你的代码，然后就问“这里什么地方有错误？”。也可以看一下论坛里过去的帖子，或许你的问题其他的人已经问过并得到解答了。

最后一个，阅读博客！许多天才的JavaScript高手喜欢通过博客和大家分享他们的思想、创新以及经验，其中包括我。这是获取新思想非常好的一种方式。我推荐你阅读下面的博客。

- Jeremy Keith[①]：http://www.adactio.com。
- Simon Willison：http://simon.incutio.com/。
- WaSP DOM脚本编程任务组：http://www.webstandards.org/action/dstf/。
- Stuart Langridge：http://kryogenix.org/days/。
- Robert Nyman：http://robertnyman.com/。
- Jon Snook：http://www.snook.ca/jonathan/。

你现在是一个非常活跃的社区的一份子了。除了可以学习许多有用的东西，你还会遇到各种各样有趣的人，而且这种学习方式会很快乐！让我们继续快乐地学习吧——不断地阅读……

致谢

感谢在完成本书的过程中所有帮助过我的人——Chris、Beth、Ami、Katie以及Apress出版社的Charles和Jon Stephens。我学到了许多，尤其是认识到了写书要做的工作比我想象的要多得多。

还要感谢那些通过解决问题并询问更多功能而由此帮助我的人——WaSP DOM脚本编程任务组的成员Stuart Colville、Matt Warden、Jens Grochtdreis、Ingo Chao、Volkan Ozcelik以及许多在evolt列表、CSS讨论区和我的博客上留言的朋友。

感谢以前在Agilisys的同事和现在在Yahoo的同事所做的测试和支持，还要感谢Good for Food、Spence、Pizzadelique以及Belle Epoque饭店为我提供了食物，让我可以保持健康的身体（还要感谢这些地方的邻居所提供的无线接入点）。

最后，我还要感谢你，因为购买本书说明有人真地想学习JavaScript，而不只是复制和粘贴脚本。

① 畅销书《JavaScript DOM 编程艺术》、《Bulletproof Ajax 中文版》（人民邮电出版社）的作者。——编者注

目　录

第 1 章　JavaScript 入门 …… 1

1.1　JavaScript 产生的原因 …… 3
1.2　JavaScript 是什么 …… 3
1.3　JavaScript 的问题和价值 …… 4
1.4　JavaScript 不可靠为什么还要用 …… 5
1.5　网页中的 JavaScript 和基本语法 …… 6
1.5.1　JavaScript 语法 …… 7
1.5.2　执行代码 …… 8
1.5.3　函数 …… 9
1.6　对象 …… 10
1.7　简单的 JavaScript 示例 …… 11
1.8　小结 …… 13

第 2 章　数据和判定 …… 15

2.1　数据、数据类型和数据运算符 …… 15
2.1.1　字符串数据类型 …… 16
2.1.2　运算符 …… 18
2.1.3　JavaScript 变量 …… 20
2.1.4　不同数据类型的转换 …… 22
2.2　复合数据类型：数组和对象 …… 25
2.2.1　JavaScript 提供的对象：String、Date 和 Math …… 26
2.2.2　数组 …… 33
2.3　在 JavaScript 中进行判定 …… 39
2.3.1　逻辑运算符和比较运算符 …… 39
2.3.2　条件语句 …… 41
2.3.3　测试多个值：switch 语句 …… 44
2.3.4　重复事件：循环 …… 45
2.4　小结 …… 50

第 3 章　从 DHTML 到 DOM 编程 …… 51

3.1　作为“行为层”的 JavaScript …… 53
3.1.1　对象检测与浏览器依赖性的比较 …… 55
3.1.2　渐进增强 …… 57
3.2　JavaScript 和可访问性 …… 58
3.3　良好的编码实践 …… 59
3.3.1　命名习惯 …… 59
3.3.2　代码布局 …… 60
3.3.3　注释 …… 62
3.3.4　函数 …… 64
3.3.5　使用三元运算符简化代码 …… 66
3.3.6　函数的分类和复用 …… 67
3.3.7　变量和函数作用域 …… 67
3.3.8　使用对象字面量保证脚本安全 …… 68
3.4　小结 …… 70

第 4 章　HTML 与 JavaScript …… 71

4.1　HTML 文档剖析 …… 71
4.2　在网页中使用 JavaScript 提供反馈信息：老的方式 …… 75
4.3　通过 DOM 访问文档 …… 80
4.4　元素的子节点、父节点、兄弟节点和值 …… 83
4.4.1　从父节点到子节点 …… 84
4.4.2　从子节点到父节点 …… 85
4.4.3　兄弟节点之间 …… 86
4.5　修改元素属性 …… 90
4.6　创建、移除和替换元素 …… 91
4.6.1　避免 NOSCRIPT …… 94
4.6.2　通过 innerHTML 简化脚本 …… 96
4.6.3　DOM 小结：你的备忘单 …… 97
4.6.4　DOMhelp：我们自己的辅助函数库 …… 98
4.7　小结 …… 102

第5章 表现与行为（CSS与事件处理）……103
5.1 通过JavaScript改变表现层……103
5.2 通过事件处理改变文档的行为……129
5.2.1 W3C标准兼容的事件……131
5.2.2 修正事件以适应W3C不兼容的浏览器……139
5.2.3 永不停止优化……144
5.2.4 页面加载问题及其解决方案……145
5.2.5 读取和过滤键盘输入……146
5.2.6 事件处理的危险……150
5.3 小结……151
第6章 JavaScript的常用对象：图片和窗口……152
6.1 图片与JavaScript……152
6.1.1 图片编程基础……153
6.1.2 预载图片……154
6.1.3 翻转效果……155
6.1.4 幻灯片显示……163
6.1.5 图片与JavaScript小结……176
6.2 窗口与JavaScript……177
6.2.1 窗口属性……178
6.2.2 窗口方法……179
6.2.3 窗口与JavaScript小结……198
6.3 小结……199
第7章 JavaScript与用户的交互：导航与表单……200
7.1 导航与JavaScript……200
7.1.1 重新加载网页的恐惧……200
7.1.2 JavaScript导航基础……201
7.1.3 浏览器导航……203
7.1.4 页内导航……204
7.1.5 网站导航……212
7.1.6 分页……219
7.1.7 使用JavaScript进行导航小结……226
7.2 表单与JavaScript……226
7.2.1 JavaScript表单基础……227
7.2.2 表单元素……228
7.2.3 交互式表单：隐藏或显示独立元素……241
7.2.4 定制表单元素……245
7.2.5 表单与JavaScript小结……246
7.3 小结……246
第8章 与Ajax后端交互……247
8.1 Ajax到底是什么……248
8.2 高速缓存竟带来了麻烦……254
8.3 把X放回到Ajax里面……255
8.3.1 使用JSON代替XML……259
8.3.2 使用服务器端脚本来访问第三方内容……261
8.3.3 关于缓慢链接的XHR问题……264
8.3.4 一个更大的Ajax示例：关联选择框……266
8.3.5 可选的动态Ajax菜单……273
8.4 小结……280
第9章 数据验证技术……282
9.1 客户端JavaScript验证的优点和缺点……282
9.2 使用JavaScript保护文件内容……283
9.3 全能验证的神话……284
9.4 使用字符串和数字方法的基本JavaScript验证……284
9.4.1 字符串验证方法……284
9.4.2 数字验证方法……290
9.5 正则表达式……293
9.5.1 语法和属性……294
9.5.2 通配符搜索、约束范围以及其替换……295
9.5.3 使用量词约束字符的数量……295
9.5.4 词界、空白字符以及其他快捷符号……296
9.5.5 使用正则表达式的方法……297
9.5.6 圆括号分组的功能……297
9.5.7 正则表达式资源……298
9.6 验证方法小结……299
9.7 表单验证技术……299
9.7.1 指定强制字段……299
9.7.2 隐藏字段方法……300
9.7.3 指示元素方法……301

9.7.4 CSS 类方法……301
9.7.5 自定义属性方法……302
9.7.6 这些方法的缺点……302
9.7.7 共用验证规则……302
9.8 为用户反馈验证信息……304
9.8.1 显示错误字段的列表……304
9.8.2 使用可单击的错误消息代替主表单……308
9.8.3 单独地突出显示错误的字段……310
9.8.4 即时验证反馈……313
9.9 其他的动态验证方法……314
9.10 小结……317
第 10 章 现代的 JavaScript 案例研究：动态图库……319
10.1 缩略图图库基础……319
10.2 缩略图图库是什么以及它应该做什么……319
10.3 静态缩略图图库……320
10.4 使用 JavaScript 模拟动态图库……320
10.5 显示标题……326
10.6 动态的缩略图库……330
10.7 从文件夹中创建图片徽章……333
10.8 小结……340
第 11 章 使用第三方 JavaScript……341
11.1 网络为你提供了什么……341
11.2 代码片段、RSS 提要、各种 API 以及函数库……342
11.2.1 RSS 提要和 REST API……342
11.2.2 REST API 示例……344
11.3 使用简短精练的函数库：jQuery……344
11.4 使用 API：用 Google Maps 为你的网站添加地图……351
11.5 完整的服务：雅虎开发人员网络以及 YUI……360
11.5.1 使用 YUI 的弹性标题……361
11.5.2 使用 YUI 的连接管理器和容器组件代替弹出窗口……366
11.5.3 YUI 小结……370
11.6 小结……371
附录 A 调试 JavaScript……372

第1章 JavaScript入门

本书主要讲述一种名为JavaScript的脚本语言以及如何在实际开发中使用它。在读完本书之后，你就能够：

- 理解JavaScript的语法和结构。
- 创建容易理解和维护的脚本。
- 编写不与其他JavaScript冲突的脚本。
- 编写脚本，使网站更加容易使用，且不排斥未启用JavaScript的用户。
- 编写与浏览器或用户代理无关的脚本——也就是在未来许多年中它们仍然是有用的，不会依赖于过时的技术。
- 使用JavaScript增强网站，允许没有任何编程知识的开发人员改变网站的界面外观。
- 使用JavaScript增强网站文档，允许HTML开发人员通过给某个元素简单地添加一个CSS类来使用你开发的功能。
- 在用户代理允许的时候，可以使用渐进增强来使Web文档变得更好。
- 使用Ajax架设客户端和服务器端之间的桥梁，创建更易维护，而且用户体验更加平滑流畅的网站。
- 将JavaScript作为Web方法的一部分，使你能够独立地维护它而不与其他的开发流程冲突。

在这里你不会发现：

- 如何创建华丽的但对访问者没有任何价值的特效的指导。
- 特定于某种浏览器的JavaScript应用。
- 只是为了证明它可以被使用但不能增强访问者体验的JavaScript代码。
- 强制“推销”垃圾内容的JavaScript脚本，如弹出窗口或其他展示动画效果的华而不实的技术。

我坚信JavaScript在现代的Web开发中占有重要的地位，但我们无法保证访问者能够使用或体验用JavaScript所能达到的所有特效和功能。JavaScript允许通过添加、移除、显示或隐藏元素来完全地改变网页效果。我们可以提供丰富的界面，如拖放式的应用程序或是多层下拉菜单。但是，一些访问者并不能使用拖放式的界面，因为他们只能使用键盘或是依赖语音识别来访问我们的网站，还有一些用户可能是通过听而不是看（通过屏幕阅读器）来访问的，所以不能够察觉由

JavaScript带来的变化。不仅如此，有的用户不能启用JavaScript功能，如在银行等安全级别高的环境中。因此，必须通过一些服务器端解决方案为我们用JavaScript实现的许多功能提供备用支持。

遗憾的是，JavaScript也有一段被用于弹出强制信息的历史，这些信息往往并不是用户所请求的（例如弹出窗口）。对于这种行为，我和许多专业Web设计人员感觉都很头疼。希望你不要使用从本书中获得的知识去做这样的事情。

注解　网页设计经过这些年的发展已经成熟——我们已经不再使用FONT标签，而且一些可视化属性例如bgcolor也已被废弃了。我们推荐把所有的格式化和界面表现属性放到CSS文件中。发生在JavaScript上的演化过程同样是Web开发的一部分。我们已把内容、结构和表现分离开了，现在是把网站的行为从其他层分离出来的时候了。Web开发现在主要考虑的是业务需求和用户体验，而不是把一些程序放到那里，然后痴心妄想它们能在大多数平台环境下都可以使用。

现在是把JavaScript看作整个开发技术一部分的时候了，这意味着我们开发JavaScript程序与其他的像HTML、CSS这样的技术不仅不冲突，而且要和它们配合使用或弥补它们的不足。现在，我们看到出现了一种新的技术（或者至少是现有技术的一种新的使用方式），它叫Ajax。第8章将会讨论它。

在20世纪90年代，Web开发得到了飞速发展，现在创建静态或者大小固定的网站已经意义不大了。任何现代的网页设计都必须考虑未来的扩展需求。同样，它对每个人来说都应该是可访问的（这并不意味每个人都会得到相同的显示效果：例如，一个好的多列布局。可能在高分辨率的显示器上显示得很好却很难在手机或者PDA上使用），并且是可以国际化的。如今，我们已经不可能一劳永逸，做一个产品，永远都好用。因为Web是与内容相关的，且需要不断地变化，所以如果不经常升级Web产品并允许其他数据源为其添加新数据或从中获取数据，那么它很快就会过时。

简介已经足够了——你拿这本书是来学习JavaScript的。在深入讲解它之前，让我们先来快速介绍一下JavaScript的历史和渊源。

本章主要介绍：

- JavaScript是什么以及它的功用。
- JavaScript的优缺点。
- 如何把JavaScript添加到一个Web文档中，JavaScript的基本语法。
- 面向对象编程（OOP）与JavaScript的关系。
- 如何编写并运行一个简单的JavaScript程序。

很有可能你已接触过JavaScript，并且对JavaScript的含义及功用有了自己的想法，所以我们先快速浏览一下这种语言的一些基础和用途。如果你对JavaScript已经有一定的了解，想简单了解一下更多新的特性和概念，可以直接跳到第3章。不用在这里占用太多的时间——不过如果你已经遗忘了一些知识点，复习一下也不会有坏处。

1.1 JavaScript 产生的原因

在Web发展的初期，主要有HTML和公共网关接口（CGI）。HTML定义了大部分的文本文档并指示用户代理（通常是网页浏览器）如何来显示。举个例子，标签 之间的文字就变成一个段落，在这个段落中可以使用标签<h1></h1>来定义最主要的页面标题。注意大多数开始标签，都会有对应的以</开头的结束标签。

HTML有个缺点，即它的状态是固定不变的。如果想改变一些东西或者使用用户输入的数据，就需要向服务器做一个往返的请求。使用动态技术（如ColdFusion、ASP、ASP.NET、PHP或JSP）可以从表单或者参数中把信息发送到服务器，服务器然后完成计算、测试、数据库查找等。和这些技术相关联的应用程序服务器会写一个HTML文档来显示结果，然后把处理的结果以HTML文档的形式返回到浏览器来供用户查看。

这样做的问题在于任何时候只要有变化，以上整个过程都需要再重复一遍（并且重新加载网页）。这样显得比较笨重缓慢，没有网络这个新媒介向我们承诺的那么美好。现在，人们已经普遍拥有了快速的因特网连接。但是显示一个页面仍然意味着重新加载，这是一个经常失败的缓慢过程（遇到过Error 404没有？）。

我们需要更加灵活的东西——要允许Web开发人员快速对用户给予反馈并且不用从服务器重新加载页面来改变HTML。可以想象一个表单，只要有一个字段中产生了错误，它都需要重新加载，如果能够不用重新从服务器加载页面，就能快速地获得错误提示，岂不是更方便实用？这正是JavaScript的用武之地。

一些信息（比如表单上的一些计算和验证信息）并不需要依靠服务器。JavaScript可以由访问者电脑上的用户代理（通常是一个浏览器）执行。我们把这叫作*客户端代码*（client-side code）。这样可以减少与服务器的交互成本并且使网站运行更快。

1.2 JavaScript 是什么

JavaScript的前身是LiveScript，但是网景公司后来把名字改成了JavaScript，很可能是由于Java的火爆。这个名字经常会令人感到迷惑，因为尽管Java与JavaScript有些语法比较相近，但它们之间并没有必然的联系。

> Java之于JavaScript就好比Car（汽车）之于Carpet（地毯）。
>
> ——来自Usenet[①]上的JavaScript讨论组

网景公司在1996年创造了JavaScript语言，它包含在Netscape Navigator（NN）2.0浏览器中，用解释器来读取和执行添加到.html页面的JavaScript代码。从此，这种语言稳步发展壮大并越来越普及，现在大多数的浏览器都支持它。

① 一个世界性的新闻组网络系统。——译者注

这意味着JavaScript可以用于网页中，被所有现代的浏览器所理解。但是，不同的浏览器在实现JavaScript的方式上是不同的，尽管核心的JavaScript语言是一样的。不过，JavaScript可以被用户关闭掉，并且一些公司和机构从安全角度考虑要求他们的用户这样做。这个我们稍后（贯穿本书）会进一步讨论。

关于JavaScript最大的特点就是，一旦学会了如何在浏览器编程中使用它，你就可以把它应用到其他领域中。微软的服务器（IIS）使用JavaScript去做服务器端网页编程（ASP），PDF文件现在也使用JavaScript，甚至Windows的任务管理也可以使用JavaScript代码来自动运行。许多应用程序，如Dreamweaver和Photoshop，都可以使用JavaScript来编写脚本。操作系统上的许多插件，如苹果公司的Dashboard或者Linux和Windows平台上的Konfabulator[①]，甚至允许使用JavaScript编写小的帮助程序。

最近许多大公司也提供了可用在网页中的JavaScript对象和方法组成的API（应用编程接口），Google Maps就是其中的一种。只需要使用几行代码就可以在你的网站中提供可缩放和可滚动的地图。

另一个更好的特点就是，JavaScript比高级编程语言和服务器端编程语言更容易开发。它不需要像Java和C++那样需要编译，也不需要像Perl、PHP或Ruby语言那样运行在服务器上或需要在命令行执行。编写、执行、调试和应用JavaScript脚本所需的就是文本编辑器和浏览器，而这两者在所有的操作系统中都提供。当然，也有工具可以使你更加方便，例如Mozilla Venkman、Microsoft Script Debugger和kjscmd这样的JavaScript调试器。

1.3　JavaScript的问题和价值

正如我在本章的开始提到的，JavaScript在过去的几年里已经成为Web开发的一个完整部分，但它也经常被错误地使用。结果，它就落了一个不好的名声。导致这个结果的原因是某些严重影响用户的JavaScript特效，如移动的页面元素和弹出窗口。这种情况你第一次看到印象会很深刻，但很快就变成是“有了也不错”，在有些情况下，甚至变成“没有更好”。许多类似的效果都来自DHTML时代（详见第3章）。

术语用户代理（user agent）和对其含义的缺乏理解同样也是个问题。通常，用户代理是指一个浏览器，如微软的IE、Netscape、Mozilla（Moz）、Firefox（Fx）、Opera或Safari。但是浏览器不是Web上唯一的用户代理，其他用户代理还包括：

- 辅助技术，用来帮助用户克服它们的缺陷——如语音合成（text-to-speech）软件或者盲文显示器。
- 纯文本代理，如Lynx。
- 支持Web的应用程序。
- 游戏控制台。
- 手机。

① Konfabulator已被Yahoo!收购，并被更名为Yahoo! Widget Engine。——编者注