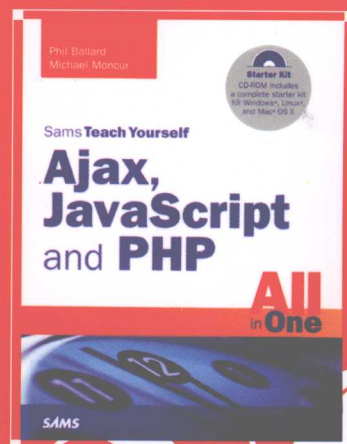


Ajax、JavaScript 和 PHP 编程导学

Sams Teach Yourself Ajax, JavaScript, And PHP All in One

(英) Phil Ballard Michael Moncur 著
刘红伟 等译

循序渐进，彻底掌握Web编程。
附赠光盘，迅速开始上手实践。



机械工业出版社
China Machine Press

Web开发系列丛书

Ajax、JavaScript 和 PHP 编程导学

Sams Teach Yourself Ajax, JavaScript, And PHP All in One

(英) Phil Ballard Michael Moncur 著
刘红伟 等译



机械工业出版社
China Machine Press

本书详尽地介绍了如何使用Ajax、JavaScript和PHP进行Web开发。全书分为6个部分,包括:Web基础知识回顾、使用JavaScript Web脚本编程基础、Ajax基础、使用PHP的服务器端脚本编程、较复杂的Ajax技术、Ajax工具和资源。另外,附录列出了JavaScript、PHP和Ajax Web站点,可方便开发人员使用。

本书非常适合Web应用开发人员及网页设计人员参考。

Authorized translation from the English language edition entitled *Sams Teach Yourself Ajax, JavaScript, and PHP All in One* by Phil Ballard, Michael Moncur, published by Pearson Education, Inc, publishing as Sams (ISBN 978-0-672-32965-4) Copyright © 2009 by Sams.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2009 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2008-3934

图书在版编目(CIP)数据

Ajax、JavaScript和PHP编程导学 / (英)巴拉德等著;刘红伟等译. —北京:机械工业出版社, 2008.10

(Web开发系列丛书)

书名原文: Sams Teach Yourself Ajax, JavaScript, and PHP All in One

ISBN 978-7-111-25311-2

I. A… II. ①巴… ②刘… III. ①计算网络—程序设计 ②JAVA语言—主页制作—程序设计 ③PHP语言—程序设计 IV. TP393.09 TP312

中国版本图书馆CIP数据核字(2008)第157225号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:盛东亮

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2009年1月第1版第1次印刷

186mm×240mm · 17.25印张

标准书号:ISBN 978-7-111-25311-2

ISBN 978-7-89482-842-2 (光盘)

定价:45.00元(附光盘)

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010) 68326294

译者序

Ajax提供与服务器进行异步通信的能力，从而使用户从请求、响应的循环中解脱出来。有了Ajax，在用户单击按钮时，我们可以使用JavaScript和DHTML立即更新UI，并向服务器发出异步请求，以执行更新或查询数据库。当请求返回时，就可以使用JavaScript和CSS来更新相应的UI，而不是刷新整个页面。最重要的是，用户甚至不知道浏览器正在与服务器通信：Web站点看起来是即时响应的。

PHP (PHP:Hypertext Preprocessor) 是一种在服务器端执行的嵌入HTML文档的脚本语言。PHP广泛地应用于Web开发中。在当前流行的Web开发框架LAMP中，PHP就是必不可少的元素。

那么，如何把Ajax和PHP这两种流行技术结合起来，开发出功能强大、响应迅速，具有Ajax功能的Web应用呢？本书对此给出了很好的解答。

本书划分为6个部分，每个部分都专注于一种特定的技术或讨论主题，分别是Web基础知识、JavaScript基础知识、Ajax基础知识、PHP服务器端脚本编程、高级Ajax技术以及Ajax工具和资源。在每个部分中，各章专门介绍每个给定的方面或子话题。在一些章的末尾，设有“自己动手”部分，启发读者使用该章所学的技术进一步扩展实例，起到举一反三、融会贯通的目的。

此外，本书的随书光盘包含了建立一个Web服务环境所需的一切内容。这个包叫做XAMPP。它包含了开发一个本书中描述的那样功能完整、具有交互性的Web站点所需的一切，并且如果你愿意的话，可以随时把它部署到一个基于Web的服务器上，不失为读者学习Ajax+PHP开发的有力工具。

本书英文版是美国最受欢迎的IT图书系列“Sams Teach Yourself”中的一本。

很高兴能够翻译本书，也非常希望读者朋友们能够通过本书掌握Ajax和PHP整合应用的Web开发技能。除了刘红伟，参加本书翻译工作的还有楚亚军、景文生、王志刚、安宁宁、党耀云、武文超、庞晋才、刘永昌和孙连升。

前言

在过去的十年里，万维网已经从相对简单的信息仓库变成了很多人寻找娱乐、教育、新闻和商务等资源的第一站。

Web站点本身不再局限于一些包含文本或简单图像的静态页面；目前有许多工具能够开发出高度交互和引人入胜的页面，其中包括动画、视觉效果、随环境变化的内容、嵌入式的工具，甚至更多。

产生这样的页面的技术有很多。然而，那些基于开源许可的技术已经变得流行起来并且持续流行，因为它们通常有低（往往是零）切入成本，并且有巨量的资源可以通过互联网或其他方式获取，包括用户贡献的脚本、教程、工具以及这些工具和应用程序的其他资源。

在本书中，我们详细地介绍如何使用服务器端和客户端代码技术和工具编写流畅、可交互的Web站点，以及如何使用Ajax技术来得到一种灵活的、类似桌面应用程序的用户体验。

本书中使用的编程语言包括随处可见的JavaScript（用于客户端编程）和非常流行的开源语言PHP（用于服务器端脚本编程，并且在大部分Web主机服务中可用）。对于Ajax编程的点点滴滴，以及包含了已经编写好的代码可供快速构建顶级的交互站点的高级开源框架，本书全都详细介绍。

什么是Ajax

Ajax表示Asynchronous JavaScript And XML（异步JavaScript和XML）。尽管严格来讲，Ajax本身并不是一种技术，它以不同寻常的方式混合了知名的编程技术，以允许开发者使用那些比我们已经习惯的界面更为吸引用户的界面来构建Internet应用程序。

在使用流行的桌面应用程序的时候，我们期望工作的结果可以立即可用，没有忙乱，并且不用等待程序重新绘制整个屏幕。例如，在使用Excel这样的电子表格的时候，我们期望在继续录入、滚动页面或使用鼠标的时候，在一个单元格所作的修改能够立即作用到相邻的单元格。

遗憾的是，对于基于Web的应用程序的用户来说，这种交互很少能够实现。更常见的体验是，从字段输入数据，点击一个按钮或超链接，然后静静地坐着等待页面缓慢重载显示请求的结果。此外，我们经常发现重载的页面包含的大多数元素和之前的页面是完全相同的，但还是被没有必要地重新载入了。背景图像、Logo和菜单都经常如此。

Ajax给我们提供了解决这一问题的方案。通过扮演用户的浏览器和Web服务器之间的一个额外的层，Ajax在后台处理服务器通信、提交服务器请求并且处理返回的数据。这样无需刷新页面，也不需要载入一个新页面，就可以将结果无缝地整合到查看的页面中。

在Ajax应用程序中，这样的服务器请求无需和点击按钮或链接这样的用户动作同步。一个编写良好的Ajax应用程序可能在用户意识到需要此信息之前已经向服务器发出请求，并接收到

用户所需的数据，这就是Ajax缩略语中的“异步”的含义。

Ajax应用程序的各个部分都工作在用户浏览器的底层，例如，向服务器发送查询和处理返回的数据，都是用JavaScript编写的。而XML是一种越来越流行的、编码和传输格式化信息的方法，被Ajax用来在服务器和客户机之间高效地传输数据。

随着各章的学习，我们将了解所有这些技术以及如何把它们综合到一起使用。

本书的目标读者

本书主要面向那些想要为Web应用程序用户构建更好的界面的开发者，以及要把自己的应用程序从桌面环境转变为Internet应用程序的程序员。

对于那些想要了解最新技术如何为自己的创造力提供新的思路的Web设计者来说，本书也是有益的。尽管PHP、JavaScript和Ajax应用程序的本质意味着需要一些编程工作，但本书中所有必需的技术都是从基本原理开始介绍的，因此，即便那些没有任何编程经验的人，也应该能够学习本书的内容而不会有太大的困难。

如何阅读本书

所有的技术（包括WWW基础知识的回顾）都是从基本原理开始介绍的，因此，即便非程序员或者那些不熟悉这些语言的人，也应该能够掌握这些概念而没有太大问题。

本书划分为几个部分，每部分介绍一个特定的技术或讨论主题。一个部分又分为几个章节专门介绍每个给定的方面或子话题。因此，通过快速浏览目录，可以很容易地知道本书的学习流程。

然而，如果你已经能胜任某一种或几种技术，如PHP或JavaScript，那么请放心加快阅读速度或跳过你不需要阅读的部分。

要尝试众多的示例，你需要访问一台支持PHP的Web服务器，并且需要一种方法来把文件上传到Web空间（可能是FTP）。大多数Web主机都在它们的服务选项中包含了PHP，只需较少的成本甚至不需要成本就可以做到这一点。

此外，本书的随书光盘包含了在你自己的计算机上建立一个Web服务环境所需的一切内容。这个包叫做XAMPP。它包含了开发一个本书描述的功能完整、具有交互性的Web站点所需的一切，并且如果你愿意的话，可以随时把它部署到一个基于Web的服务器上。注意书中标有“光盘说明”部分的内容。

编写惯例

本书包含了如下的特别元素：

你知道吗 这些框中的内容强调了可以使编程更高效和有效的信息。

提示 这些框中的内容是你刚刚阅读的材料的相关信息。

注意 这些框提醒你注意可能在特定情况下发生的问题或副作用。

光盘说明 这部分提醒你，随书光盘上有相关的信息或可用的工具。

自己动手

自己动手部分对创建自己的脚本、进一步实验或应用一章中所学到的知识提供了建议。这将帮助你根据已经学到的知识来创建实用的应用程序。

和编程相关的术语和语言使用特殊的半角字体。

建立自己的工作环境

尽管只要使用一个简单的文本编辑器就可以编写本书中的代码，但要运行示例，则需要一台运行Internet Explorer或Firefox这样的现代浏览器的计算机（装有Windows、Mac或Linux操作系统）。

你知道吗

可以从<http://www.microsoft.com/>下载Microsoft Windows Explorer，或者从<http://www.mozilla.com/>下载Firefox的最新版本。

你还需要在一台Web服务器上载入文件。如果你已经有了一台支持PHP的Web主机，可以使用那里的Web空间。如果没有，随书光盘中包含了在你自己的PC或网络上的其他PC上建立自己私用的Web服务器所需的一切。

光盘内容

随书光盘中包含了充分利用本书所需的一切内容。包括：

- XAMPP，一个完整的开源软件集，可以用来轻松地自己的计算机上安装Apache Web服务器、PHP语言和MySQL数据库管理器。它提供了针对Linux、Mac和Windows环境的不同版本。
- jEdit 一个基于Java的程序员编辑器，可以很好地创建或修改代码。CD包含了供Java、Mac和Windows使用的文件。
- 一个开发高级Web应用程序所需的开源框架的选集。本书的最后部分介绍了基于这些框架的一些编程示例。

目 录

译者序

前言

第一部分 Web基础知识回顾

第1章 Web的运作	1
1.1 Internet简史	1
1.2 万维网	2
1.2.1 什么是Web页面	2
1.2.2 Web服务器	2
1.2.3 服务器端编程	3
1.2.4 Web浏览器	4
1.2.5 客户端编程	5
1.2.6 DNS——域名服务器	5
1.3 HTTP简介	5
1.4 HTTP请求和响应	5
1.4.1 HTTP请求	6
1.4.2 HTTP响应	6
1.5 HTML表单	7
1.6 小结	10
第2章 用HTML和CSS编写样式化页面	11
2.1 HTML简介	11
2.1.1 什么是HTML	11
2.1.2 编写HTML需要什么工具	12
2.1.3 第一个HTML文档	12
2.2 HTML页面的元素	13
2.2.1 文档的标题<head>	13
2.2.2 文档正文<body>	14
2.2.3 为HTML元素添加属性	14
2.2.4 图像	15
2.2.5 表	15
2.2.6 超链接	16

2.3 HTML页面更高级知识	16
2.4 一些有用的HTML标签	18
2.5 添加自己的样式	18
2.6 定义规则	19
2.7 添加一些类	19
2.8 应用样式	21
2.8.1 嵌入样式	21
2.8.2 链接样式	21
2.8.3 内联样式	22
2.8.4 层叠优先级	22
2.9 使用样式格式化文本	23
2.9.1 链接样式	26
2.9.2 颜色样式	27
2.10 添加线条	27
2.11 小结	28
第3章 剖析Ajax应用程序	29
3.1 为何需要Ajax	29
3.1.1 传统客户端—服务器交互和Ajax	29
3.1.2 富用户体验	30
3.2 Ajax简介	31
3.3 Ajax的必备部分	32
3.3.1 XMLHttpRequest对象	32
3.3.2 与服务器通信	33
3.3.3 服务器上发生了什么	33
3.3.4 处理服务器响应	33
3.3.5 其他常见任务	33
3.4 整合	34
3.5 小结	35

第二部分 JavaScript Web脚本编程基础

第4章 用JavaScript创建简单脚本	37
4.1 脚本编程工具	37

4.1.1 文本编辑器	37	5.4 访问浏览器历史	55
4.1.2 浏览器	38	5.5 使用location对象	56
4.2 使用JavaScript显示时间	38	5.6 小结	58
4.3 开始脚本	38	第6章 使用变量、字符串和数组	59
4.4 添加JavaScript语句	39	6.1 使用变量	59
4.4.1 把日期存储到变量中	39	6.1.1 选择变量名称	59
4.4.2 计算结果	39	6.1.2 使用局部变量和全局变量	60
4.5 创建输出	39	6.1.3 给变量赋值	61
4.6 把脚本添加到Web页面	40	6.2 理解表达式和操作符	61
4.7 测试脚本	41	6.2.1 使用JavaScript操作符	62
4.7.1 修改脚本	41	6.2.2 操作符优先级	62
4.7.2 处理JavaScript错误	43	6.3 JavaScript中的数据类型	63
4.7.3 语句	44	6.4 数据类型之间的转换	63
4.7.4 使用函数组合任务	44	6.5 使用String对象	64
4.7.5 变量	45	6.5.1 创建一个String对象	64
4.7.6 条件	45	6.5.2 赋值	65
4.7.7 循环	45	6.5.3 计算字符串的长度	66
4.7.8 事件处理程序	45	6.5.4 转换字符串的大小写	66
4.7.9 哪个脚本先运行	46	6.6 操作子字符串	67
4.8 JavaScript语法规则	46	6.6.1 使用字符串的一部分	67
4.8.1 区分大小写	47	6.6.2 获取单个字符	67
4.8.2 变量名、对象名和函数名	47	6.6.3 查找一个子字符串	67
4.8.3 保留字	47	6.7 使用数值数组	68
4.8.4 空白	47	6.7.1 创建一个数值数组	68
4.9 使用注释	47	6.7.2 理解数组长度	69
4.10 JavaScript最佳实践	48	6.7.3 访问数组元素	69
4.11 小结	49	6.8 使用字符串数组	70
第5章 使用文档对象模型 (DOM)	51	6.8.1 创建字符串数组	70
5.1 理解对象	51	6.8.2 分隔字符串	70
5.2 理解DOM	52	6.8.3 排序字符串数组	71
5.2.1 DOM的历史	52	6.9 数值数组排序	71
5.2.2 DOM层级	53	6.10 小结	73
5.3 操作Web文档	53	第7章 使用函数和对象	74
5.3.1 获取有关文档的信息	53	7.1 使用函数	74
5.3.2 把文本写入一个文档	54	7.1.1 定义函数	74
5.3.3 使用链接和锚点	55	7.1.2 调用函数	75

7.1.3 返回值	77
7.2 对象简介	78
7.2.1 创建对象	78
7.2.2 对象属性和值	78
7.2.3 理解方法	79
7.3 使用对象来简化脚本编程	79
7.3.1 定义对象	79
7.3.2 定义对象方法	80
7.3.3 创建对象实例	81
7.4 扩展内建的对象	81
7.5 小结	84
第8章 使用条件和循环来控制流程	85
8.1 if语句	85
8.1.1 条件操作符	86
8.1.2 使用逻辑操作符组合条件	86
8.1.3 else关键字	87
8.2 使用简捷条件表达式	88
8.3 使用if和else测试多个条件	88
8.3.1 HTML文件	89
8.3.2 JavaScript文件	89
8.4 使用带有switch的多个条件	90
8.5 使用for循环	91
8.6 使用while循环	93
8.7 使用do...while循环	94
8.8 使用循环	94
8.8.1 创建一个无限循环	94
8.8.2 退出循环	95
8.8.3 继续循环	95
8.9 遍历对象属性	96
8.10 小结	98
第9章 使用内建函数和库	99
9.1 使用Math对象	99
9.1.1 舍入和截断	99
9.1.2 生成随机数	99
9.1.3 其他Math函数	100
9.2 使用Math函数	100

9.3 使用with关键字	102
9.4 使用Date	102
9.4.1 创建一个Date对象	103
9.4.2 设置Date值	103
9.4.3 读取日期值	103
9.4.4 使用时区	104
9.4.5 在日期格式之间转换	104
9.5 使用第三方库	104
9.5.1 Prototype	105
9.5.2 Script.aculo.us	105
9.6 其他的库	105
9.7 小结	106

第三部分 Ajax基础

第10章 Ajax的核心——XMLHttpRequest

对象	107
10.1 XMLHttpRequest简介	107
10.2 创建XMLHttpRequest对象	108
10.2.1 不同浏览器的不同结果	108
10.2.2 一个提供给所有浏览器的解决方案	108
10.2.3 方法和属性	111
10.2.4 open()方法	112
10.2.5 send()方法	112
10.3 小结	112

第11章 与服务器通信

11.1 发送服务器请求	113
11.2 检测服务器状态	117
11.2.1 readyState属性	117
11.2.2 服务器响应状态码	118
11.3 回调函数	118
11.4 小结	120

第12章 使用返回的数据

12.1 responseText和responseXML属性	121
12.1.1 responseText属性	121
12.1.2 responseXML属性	123
12.1.3 getElementsByTagName()方法	124

12.2 解析responseXML	124	15.5.1 舍入数值	147
12.3 提供用户反馈	125	15.5.2 随机数	147
12.4 小结	127	15.5.3 数学函数	147
第13章 第一个Ajax应用程序	128	15.6 使用字符串	148
13.1 构建Ajax应用程序	128	15.6.1 使用反斜杠转义字符	148
13.2 HTML文档	128	15.6.2 连接	148
13.3 添加JavaScript	130	15.6.3 比较字符串	149
13.3.1 XMLHttpRequest对象	130	15.7 格式化字符串	149
13.3.2 服务器请求	130	15.7.1 使用printf	149
13.3.3 PHP服务器端脚本	131	15.7.2 使用sprintf	150
13.3.4 回调函数	131	15.8 字符串函数	150
13.3.5 使用事件处理程序	132	15.8.1 大写	150
13.4 综合应用	132	15.8.2 分解字符串	151
13.5 小结	136	15.9 使用数组	151
第四部分 使用PHP的服务器端脚本编程		15.9.1 创建和访问数组	152
第14章 认识PHP	137	15.9.2 遍历数组	152
14.1 PHP基础	137	15.9.3 关联数组	153
14.1.1 服务器端脚本编程	137	15.10 数组函数	153
14.1.2 PHP标签	138	15.11 日期格式	154
14.2 第一个PHP脚本	139	15.11.1 DIY日期格式	154
14.2.1 echo命令	140	15.11.2 UNIX时间戳格式	154
14.2.2 注释	141	15.12 使用时间戳	155
14.3 小结	142	15.12.1 格式化日期	155
第15章 变量	143	15.12.2 创建时间戳	156
15.1 理解变量	143	15.12.3 把其他的日期格式转换为时间戳	157
15.1.1 命名变量	144	15.13 小结	157
15.1.2 表达式	144	第16章 流程控制	158
15.1.3 字符串中的变量	144	16.1 条件语句	158
15.2 数据类型	145	16.1.1 条件操作符	159
15.3 使用数值	145	16.1.2 逻辑操作符	159
15.3.1 算术操作符	145	16.1.3 多条件分支	160
15.3.2 自增和自减	146	16.1.4 switch语句	161
15.3.3 复合操作符	146	16.2 循环	162
15.4 数值数据类型	147	16.2.1 while循环	162
15.5 数值函数	147	16.2.2 do循环	162
		16.2.3 for循环	163

16.2.4 嵌套条件和循环	163
16.2.5 跳出循环	163
16.3 小结	163
第17章 函数	164
17.1 使用函数	164
17.2 参数和返回值	165
17.2.1 返回成功或失败	166
17.2.2 默认参数值	166
17.2.3 变量作用域	167
17.3 使用库文件	168
17.4 小结	169
第18章 使用类	170
18.1 面向对象的PHP编程	170
18.2 什么是类	170
18.2.1 何时使用类	171
18.2.2 类是什么样的	171
18.3 创建和使用对象	171
18.3.1 方法和属性	171
18.3.2 使用第三方类	172
18.4 小结	174
第五部分 较复杂的Ajax技术	
第19章 返回文本数据	175
19.1 从responseText属性获取更多	175
19.1.1 返回文本	175
19.1.2 直接在页面元素中使用返回的文本	175
19.1.3 在responseText中包含HTML	178
19.1.4 更复杂的格式化数据	179
19.2 小结	180
第20章 AHAH——异步HTML和HTTP	182
20.1 AHAH简介	182
20.2 为AHAH创建一个较小的库	183
20.3 使用myAHAHlib.js	184
20.4 小结	189
第21章 返回XML数据	190
21.1 添加Ajax中的“x”	190

21.2 responseXML属性	190
21.2.1 更多的JavaScript的DOM方法	191
21.2.2 DOM方法概览	193
21.3 项目——一个RSS新闻阅读器	193
21.3.1 应用程序的HTML页面	194
21.3.2 完整代码	195
21.3.3 回调函数	198
21.3.4 服务器端代码	199
21.4 小结	201
第22章 Web服务器和REST及SOAP 协议	202
22.1 Web服务简介	202
22.2 REST——Representational State Transfer	203
22.2.1 一个假设的REST示例	203
22.2.2 使用GET查询信息	204
22.2.3 无状态操作	204
22.3 使用REST	204
22.3.1 读取可用文章列表	204
22.3.2 获取特定的文章	205
22.3.3 上传一篇新文章	205
22.3.4 REST和Ajax	207
22.4 使用SOAP的Web服务	208
22.5 SOAP协议	208
22.5.1 一个SOAP请求的代码示例	210
22.5.2 通过HTTP发送SOAP请求	210
22.6 使用Ajax和SOAP	211
22.7 再谈SOAP和REST	212
22.8 小结	212
第23章 用于Ajax的JavaScript库	213
23.1 一个Ajax库	213
23.2 回顾myAHAHlib.js	213
23.3 实现库	214
23.3.1 创建XMLHttpRequest实例	214
23.3.2 HTTP GET和POST请求	215
23.3.3 回调函数	216

23.4 使用库	218
23.5 小结	221
第24章 Ajax陷阱	222
24.1 常见Ajax错误	222
24.2 Back按钮	222
24.3 书签和链接	223
24.4 告诉用户正在发生什么	223
24.5 让Ajax优雅降级	224
24.6 处理搜索引擎蜘蛛	224
24.7 指出活动页面元素	224
24.8 不要在不合适的时候使用Ajax	225
24.9 安全性	225
24.10 跨多平台测试代码	226
24.11 Ajax不能弥补糟糕的设计	226
24.12 一些编程陷阱	226
24.12.1 浏览器捕获GET请求	226
24.12.2 许可拒绝错误	227
24.12.3 转义内容	227
24.13 小结	227

第六部分 Ajax工具和资源

第25章 prototype.js工具箱	229
25.1 prototype.js简介	229
25.1.1 \$()函数	230
25.1.2 \$F()函数	230
25.1.3 Form对象	230
25.1.4 Try.these()函数	231
25.2 包装XMLHttpRequest——Ajax对象	231
25.2.1 Ajax.Request	231
25.2.2 Ajax.Updater	232
25.2.3 Ajax.PeriodicalUpdater	233
25.3 示例项目——股票价格阅读器	233

25.4 小结	235
第26章 使用Rico	236
26.1 Rico简介	236
26.1.1 在应用程序中使用Rico	236
26.1.2 Rico的AjaxEngine	236
26.1.3 一个简单的示例	238
26.2 Rico的其他界面工具	240
26.2.1 拖拽	240
26.2.2 拖拽界面示例	240
26.2.3 电影效果	243
26.3 小结	244
第27章 使用Script.aculo.us	245
27.1 下载库	245
27.2 包含文件	245
27.3 使用效果	246
27.4 构建脚本	246
27.5 小结	248
第28章 使用XOAD	249
28.1 XOAD简介	249
28.1.1 下载和安装XOAD	249
28.1.2 一个简单的XOAD页面	250
28.2 XOAD HTML	252
28.2.1 XOAD_HTML::getElementById()	252
28.2.2 XOAD_HTML::getElementsByName()	253
28.3 XOAD_HTML高级编程	255
28.3.1 XOAD事件	255
28.3.2 使用XOAD缓存处理	255
28.3.3 XOAD控件	255
28.4 小结	255
附录 JavaScript、PHP和Ajax Web站点	256
术语表	258

第一部分 Web基础知识回顾

第1章 Web的运作

在本章中，你将学到：

- Internet简史。
- 万维网。
- HTTP简介。
- HTTP请求和响应。
- HTML表单。

在本书中我们有很多基本知识需要介绍。我们将在本章中介绍万维网是什么以及它的由来。随后，我们将看看让万维网能够运转的一些重要内容，特别是用来请求和发送Web页面的HTTP协议。

1.1 Internet简史

在20世纪50年代后期，美国政府成立了美国国防部高级计划署（Advanced Research Projects Agency, ARPA）。这主要是针对俄国成功发送Sputnik卫星的响应，在美国军事应用相关研究工作上部署一些国家级高端科学力量。

20世纪60年代，计划署创建了一个名为ARPAnet的分布式计算机网络。这个网络雏形最初只是连接了位于加利福尼亚大学Los Angeles分校、斯坦福研究院、加利福尼亚大学Santa Barbara分校以及犹他州立大学的4台计算机。更多的节点是在20世纪70年代添加进来的。

该网络最初设计时，使用当时最新的包交换技术，并且设计成即便一些节点遭受核攻击而被摧毁也能够继续发挥作用的通信系统。

Email在1972年实现，紧随其后，登录远程计算机的telnet协议和实现计算机之间的文件互传的FTP协议也出现了。

在随后的几年里，这个网络不断发展，继续增强，在协议和工具等众多方面都有所改善。然而，直到1989年，Tim Berners-Lee及其欧洲粒子物理实验室CERN的同事提出了使用超文本链接文档的概念，才使我们现在所熟悉的万维网逐渐成形。1993年，Mosaic浏览器出现了，这是第一款图形化Web浏览器，是著名的Netscape Navigator浏览器的前身。

超文本页面和超链接的应用，有助于定义基于页面的界面模型，而这一模型至今仍作为Web应用程序的标准。

1.2 万维网

万维网使用了一种客户/服务器的网络原理。当你在浏览器中输入一个Web页面的URL (Web地址), 并且点击“Go”, 就要求浏览器对拥有该地址的一台特定计算机发出一个HTTP请求。在接到这个请求之后, 计算机就会以浏览器可以解释和显示的格式返回 (服务) 请求的页面。图1-1说明了这个关系。当然, 在互联网中, 服务器和客户机可能位于世界上的任何地方。

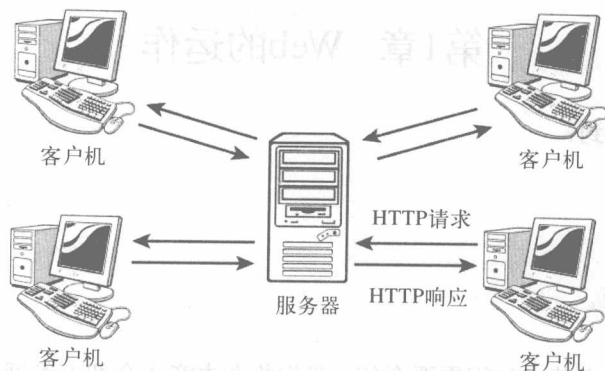


图1-1 Web服务器和客户机 (浏览器) 如何交互

稍后, 我们将更详细地讨论HTTP请求的实质。现在, 只需知道HTTP请求包含了一些信息片段, 以使页面可以正确地识别和提供。这些信息片段包括以下几点:

- 存储页面的域 (例如, mydomain.com)。
- 页面的名字 (这是Web服务器的文件系统中的文件的名字, 例如mypage.html)。
- 想要同请求一起发送的任何参数的名字和值。

1.2.1 什么是Web页面

使用过万维网的人都会熟悉Web页面这一术语。传统的Web站点用户界面主要包含访问者在一系列互相关联的页面之间的导航, 其中每个页面包含文本、图像等等, 就像是杂志的页面。

一般来说, 每个Web页面实际上都是服务器上的单独的文件。单个页面的集合组成了一个站点。站点由一个叫做Web服务器的程序来管理。

1.2.2 Web服务器

Web服务器是一个解释HTTP请求并以浏览器可以理解的形式来发送相应的Web页面的程序。Web服务器的例子有很多, 大多数运行在UNIX/Linux操作系统下, 或者运行在Microsoft Windows操作系统的某个版本下。

注意

一般来讲, 术语Web服务器 (web server) 既用来表示Web服务器程序 (如Apache), 也用来表示运行Web服务器程序的计算机。

或许，最知名的服务器应用程序就是来自Apache软件基金会（<http://www.apache.org>）的Apache Web服务器。这是用于全世界数以百万计的Web站点的一个开源项目（如图1-2所示）。

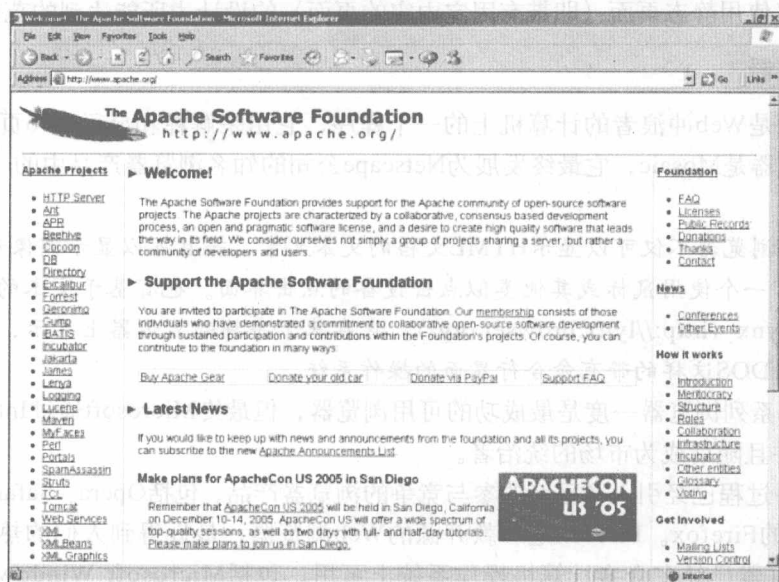


图1-2 在IE中显示的位于<http://www.apache.org/>的Apache软件基金会的主页

另一个例子就是Microsoft的IIS（Internet Information Services），常用于运行Microsoft Windows操作系统的计算机主机上。

提示

并不是所有的基于Windows的Web主机都使用IIS。还有其他的可用于Windows的Web服务器，包括流行的Apache Web服务器的一个版本。

1.2.3 服务器端编程

服务器端程序、服务器端脚本或服务器端语言，指的是运行在服务器上的程序。很多语言和工具可以用来进行服务器端编程，这包括PHP、Java和ASP（后者只能够在运行Microsoft Windows操作系统的服务器上使用）。高级的服务器设置通常也包括可以通过服务器端脚本来处理的信息数据库。

提示

本书中的服务器端编程使用流行的PHP脚本编程语言来执行，这种语言很灵活，也很容易使用，并且几乎可以在所有的服务器上运行。Ajax则可以起到与任何服务器端脚本编程语言一样好的功能。

这样的脚本用途很多，而且各不相同。通常，它们都设计用来预处理一个Web页面，然后再将这个页面返回给你。我这么说的意思是，页面内容的一些或全部将要修改以适合你的请求。这可能是显示某天到某个特定目的地的火车时刻表，或者只是显示和你所描述的爱好和兴趣相

匹配的目录中的产品。

按照这种方式,服务器端脚本编程允许Web页面提供丰富而多样的内容,而这些可能远远超越了任何仅仅使用静态页面(即带有固定内容的页面)的设计力所能达到的范围。

1.2.4 Web浏览器

Web浏览器是Web冲浪者的计算机上的一个程序,它用来解释和显示Web页面。第一款图形化的Web浏览器是Mosaic,它最终发展为Netscape公司的知名浏览器产品中的一款。

提示

图形化Web浏览器不仅可以显示HTML文档的文本元素,而且可以显示图像和颜色。通常,这样的浏览器有一个使用鼠标或其他类似点击设备的点击界面。也有基于文本的浏览器,其中最知名的就是Lynx (<http://lynx.browser.org/>)。它在基于字符的显示器上显示,例如终端、终端模拟器以及像DOS这样的带有命令行界面的操作系统。

Netscape的系列浏览器一度是最成功的可用浏览器,但最终Microsoft通过Internet Explorer加入市场竞争并且随后成为市场的统治者。

最近的竞争过程已经引入了更多的参与竞争的浏览器产品,包括Opera、Safari、Konqueror,尤其是Mozilla的Firefox。Firefox是一款开源的Web浏览器,最近得到人们的热情追捧(如图1-3所示)。浏览器已经在众多计算机操作系统上可用,包括Microsoft Windows的各个版本、UNIX/Linux、Macintosh,并且也有针对各种计算设备的浏览器,从手机到PDA以及便携式计算机。

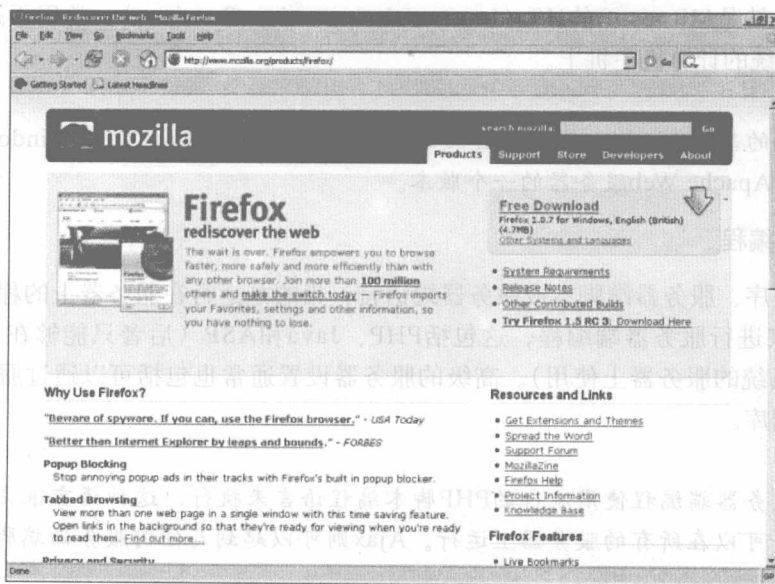


图1-3 用来自Mozilla.org的Firefox浏览器浏览Firefox项目主页