

**PACKT**  
PUBLISHING



From Technologies to solutions

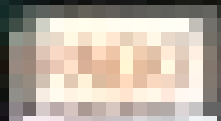
**AJAX and PHP** Building Responsive Web Applications

# Ajax 与 PHP

## Web 开发

[罗] Cristian Darie Filip Cherecheș-Toșa 著  
Bogdan Brinzarea Mihai Bucica  
王德民 王新颖 刘 昕 译

 **人民邮电出版社**  
POSTS & TELECOM PRESS



Learn to use Ajax and PHP to create dynamic web applications

Learn to use Ajax and PHP to create dynamic web applications

# Ajax与PHP

WILEY

Learn to use Ajax and PHP to create dynamic web applications

WILEY

TP393.09

120

2007

From Technologies to solutions

# Ajax 与 PHP

## Web 开发

[罗] Cristian Darie Filip Cherecheș-Toșa 著  
Bogdan Brinzarea Mihai Bucica

王德民 王新颖 刘 昕 译

人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

Ajax 与 PHP Web 开发 / (罗) 达里 (Darie, C.) 等著; 王德民, 王新颖, 刘昕译. —北京: 人民邮电出版社, 2007.4

ISBN 978-7-115-15785-0

I. A... II. ①达...②王...③王...④刘... III. ①计算机网络—程序设计  
②PHP 语言—程序设计 IV. TP393.09 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 009979 号

## 版 权 声 明

Copyright ©Packt Publishing 2006. First published in the English language under the title  
AJAX and PHP: Building Responsive Web Applications.  
All Rights Reserved.

本书由英国 Packt Publishing 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

## 内 容 提 要

本书指导读者使用 Ajax、PHP 及其他相关技术建立快速响应的网页。本书不仅从 Ajax 的客户端和服务端技术两个方面指导读者逐步掌握 Ajax 基础应用, 还通过实例详细演示了 Ajax 表单验证、Ajax 聊天室、Ajax Suggest、使用 SVG 实现的 Ajax 实时绘图程序、Ajax Grid、Ajax RSS 阅读器和 Ajax 拖放等的实现过程。同时还介绍了在 Windows 和 UNIX 系统下的应用程序环境配置和相关名词解释。

本书适合已经掌握 PHP、XML、JavaScript 和 MySQL 基础知识, 并想要了解 Ajax 核心和工作原理的读者阅读。

## Ajax 与 PHP Web 开发

- 
- ◆ 著 [罗] Cristian Darie Filip Cherecheș-Toșa  
Bogdan Brinzarea Mihai Bucica  
译 王德民 王新颖 刘 昕  
责任编辑 刘映欣
- ◆ 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>  
北京密云春雷印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16  
印张: 20  
字数: 451 千字 2007 年 4 月第 1 版  
印数: 1~5 000 册 2007 年 4 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-5371 号

ISBN 978-7-115-15785-0/TP

定价: 39.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

## 译者序

---

Ajax 全称为 Asynchronous JavaScript and XML（异步 JavaScript 和 XML），是一种开发交互式网页应用的新兴技术。Ajax 技术集成了目前在浏览器中通过 JavaScript 脚本可以使用的所有技术，并以一种崭新的方式来使用这些技术，使得古老的 B/S（浏览器/服务器）方式的 Web 开发焕发了新的活力。

应该说，使用 Ajax 开发应用程序在技术上已经不成问题，主要问题在于 Ajax 目前还不普及，熟练掌握 Ajax 还有一定的困难，用 Ajax 开发还远不如做一个 JSP/ASP 方便。然而，有需求出现就会有人提供相应的产品和相应的解决方案，目前已经有了一些基于 Ajax 技术的开发构件和开发平台出现。不论是在 .NET 下还是在 Java 环境下，都有这类产品可以帮助用户轻松地搭建基于 Ajax 技术的富客户端应用（或者富因特网应用 RIA），从而使用户可以轻松开发出基于 Ajax 技术的 Web 应用，就像过去用成熟控件开发 C/S（客户端/服务器）应用一样。因此，Ajax 技术的应用将会飞速地发展。

本书深入浅出地剖析了 Ajax 的各个方面，包括 Ajax 的基础技术，如 JavaScript、DOM、CSS 以及 Ajax 的核心技术 XMLHttpRequest。书中以服务器端 PHP 及数据库 MySQL 应用程序为依托，详细讲解了 Ajax 技术应用上的各种实例，包括 Ajax 表单验证、Ajax 聊天室、Ajax Suggest、使用 SVG 实现 Ajax 实时绘图程序、Ajax Grid、Ajax RSS 阅读器和 Ajax 拖放等。同时，本书还介绍了如何在 Windows 和 UNIX 系统下进行应用程序环境配置及相关的名词解释。相信无论是首次接触 Ajax 技术的初学者，还是已经有一定开发经验的技术人员，都可以从本书中获得丰富信息。

感谢向林艳、曹艳、孟丽霞、吕贵林、张大鹏、裴立秋、毕永东等积极参与本书的翻译工作。虽然在翻译过程中我们尽量做到尊重原意、翻译准确，并尽力修正了一些原文的小错误，但由于水平有限，不当和疏漏之处在所难免，敬请广大读者不吝指正。

译者

2007 年 1 月

## 作者简介

---

**Cristian Dari** 既是一位在众多技术领域有着很高造诣的软件工程师，也是很多技术书籍的作者，其中包括非常流行的 *Beginning E-Commerce*。他小时候刚能摸到键盘时就开始使用电脑，首次尝到编程的甜头是在他 12 岁时，第一次参加编程比赛便获得第一名。从那时起，Cristian 陆续在其他很多相关领域都取得了成功，目前他正在攻读博士学位，研究方向是分布式应用体系结构。他总是很喜欢听取关于自己所著书籍的反馈意见，所以希望读者有空时留下宝贵意见。通过 Cristian 的个人网站 <http://www.cristiandarie.ro> 可以与他取得联系。

在此，Cristian 非常感谢这本书的其他作者 Bogdan、Filip 以及 Mihai，感谢本书的技术编辑 Jimmy，正是由于他们的辛苦工作才会有这本精彩书籍的问世。

**Bogdan Brinzarea** 在计算机科学方面有着很深厚的背景，他在罗马尼亚布加勒斯特的 Politehnica 大学自动控制与计算机学院获得计算机科学学士和硕士学位，还持有法国巴黎理工学院计算机科学专业的审计员证书。

他的兴趣很广泛，主要包括嵌入式编程、分布式和移动计算，以及新兴的 Web 技术。目前，他是罗马尼亚 Banca 公司的一名分销渠道专家，还是希腊国家银行的一名成员，负责国际银行业的项目和其他一些在银行业领域的安全应用以及与新兴技术相关的项目。

**Filip Cherecheș-Toșa** 是一名对以 Web 为基础的软件应用的未来充满信心的 Web 开发师。他 9 岁时第一次拥有了一个带磁带驱动器的 Commodore 64 机器，并开始了他的职业生涯。在罗马尼亚，Filip 拥有一个 Web 开发公司 eXigo (<http://www.exigo.ro>)，这家公司主要活跃在以 Web 为基础的应用程序开发和 Web 设计两个领域。目前，他是 Oradea 大学的一名学生，主修计算机专业，同时他还是罗马尼亚 PHP 学会 (<http://www.phpromania.net>) 的主要成员。

**Mihai Bucica** 12 岁时开始编程并参加编程竞赛（获得了许多荣誉），那时他已经获得了罗马尼亚布加勒斯特的 Politehnica 大学自动控制与计算机学院计算机专业学士学位，Bucica 致力于研究不同电子市场上的通信软件。在使用过多种编程语言后，Bucica 最喜欢的编程语言依然是 C++，而且他非常喜欢 LGPL 命令。Mihai 还是 *Beginning PHP 5 and MySQL E-Commerce* 的合著者，读者可以通过他的私人网站 <http://www.valentinbucica.ro> 与他联系。

## 审校者简介

---

Emilian Balanescu 是一位在众多技术方面都有着丰富经验的编程专家，包括 PHP、Java、.NET、PostgreSQL、MS SQL Server 及 MySQL 等。目前，他是一名无线网络管理员，在罗马尼亚的 accessNET International S.A. 工作，这是一家覆盖全国范围的稳定的无线访问服务商，提供单点对多点数字无线通信网络操作。他在这个职位上最近的一个项目是开发嵌入式 Ajax 实时网络管理系统（使用 SNMP、Perl、PHP 及 PostgreSQL），用于远程调试、监测系统状态、隔离并解决系统故障问题。读者可以通过 <http://www.emilianlanescu.ro> 与 Emilian 联系。

Paula Badascu 是罗马尼亚最著名的技术型大学之一——布加勒斯特的 Politehnica 大学的一名大三学生，主修电子学、电子通信和信息技术。Paula 目前是罗马尼亚 NCH Advisors 的一名分析员/程序员，使用 UML、OOP、PHP、SQL、JavaScript 以及 CSS 建立 Web 应用程序。她在分析和开发用于跟踪和监视罗马尼亚资本市场的框架中做出了重要的贡献。

Ajax 是一种综合技术,对不同的人来说意味着不同的东西。计算机使用者们高兴地发现他们喜爱的网站界面越来越友好,并且响应信息越来越多。Web 开发人员则学到了一门使他们只需要付出很少的努力就能够创建健壮的 Web 应用程序的新技术。的确, Ajax 给很多人都带来了福音。

从根本上说, Ajax 是一种综合技术,它能够避免烦琐的网页重载,这意味着从一个页面跳转到另一个页面无须等待。消除页面重载离在网站中支持更复杂特性只有一步之遥,这些特性包括实时数据验证、拖动功能以及其他一些和传统的 Web 应用程序不相关的功能。虽然 Ajax 的组成部分已经成熟(Ajax 的核心 XMLHttpRequest 对象是微软公司在 1999 年提出的),但它们在 Web 发展浪潮中扮演的新角色却非常年轻,在这些技术带给终端用户最好体验之前,我们会见证众多的变化。到写作这本书的时候为止, Ajax 这一名词仅仅提出了一年的时间。

当然, Ajax 并不像当前大肆宣传的那样能解决所有的 Web 问题。就像其他技术一样, Ajax 可能被滥用或错误地使用。 Ajax 同样有它自身的不足:需要克服与浏览器的矛盾;如果浏览器禁用 JavaScript,使用 Ajax 技术的网页将无法显示;用户将 Ajax 页面添加到书签比较困难;搜索引擎有时不知道如何解析 Ajax 页面。同样,并不是每个人都喜欢 Ajax,就像有的人喜欢在开发架构中使用 JavaScript,而有的人根本就不喜欢 JavaScript 一样。当过度的炒作过后,大多数人会发现,对多数这种现象保持一种中立的态度才是明智之举。

在本书中采用了一种既实用又安全的方法:讲解相关模式和最佳实例,这对所有 Web 开发人员来说迟早都会有用。本书将介绍如何避免陷入常见的陷阱,如何编写高效的 Ajax 代码,以及如何将 Ajax 与现有的和将来的 Web 应用程序简单地融为一体并完成它的功能,而不是重新建立围绕 Ajax 的整个解决方案。读者可以将从本书学到的知识运用到开发 PHP 的 Web 应用程序中。

希望本书对读者的实际项目开发有所帮助。访问 <http://ajaxphp.packtpub.com> 可以获得本书的最新消息和更新内容。

在这本书的网站上还包括免费的附加内容和资源,建议读者下载阅读。

## 本书概述

第 1 章 Ajax 与未来的 Web 应用程序 将首次带你进入 Ajax 世界,它将向 Web 开发人

员和开发团队展示它的巨大潜能，它将为其他用户提供更加美好的体验。在这一章中，读者还将建立第一个嵌入 Ajax 的 Web 页面，第一次体验这种组件技术。

**第 2 章 JavaScript 下灵活的客户端技术** 主要介绍了如何使用 JavaScript、DOM、XMLHttpRequest 对象和 XML 技术实现 Ajax 客户端。虽然没有对所有这些技术作完整详细的说明，但足以使读者能够正确地使用这些技术，为将来的应用打下坚实的基础。

**第 3 章 使用 PHP 和 MySQL 实现服务器端技术** 通过介绍如何创建与 Ajax 客户端用户进行交互的更灵活的服务器来阐述理论基础，在本章中将介绍完成一般任务的各项技术，包括处理基本的 JavaScript 安全以及出错处理问题。

**第 4 章 Ajax 表单验证** 指导读者创建一个时髦的、快速响应的和安全的表单验证系统，它不仅在表单提交时能够实现实时的 Ajax 验证，还能实现服务端验证。

**第 5 章 Ajax 聊天** 展现了一个专门使用 Ajax 代码来工作的简单在线聊天程序，它没有使用 Java applets、Flash 代码，也没有使用目前其他大多数聊天应用程序使用的专用库。

**第 6 章 Ajax 建议和自动完成** 建立了一个类似 Google Suggest 的功能，帮助读者快速查找 PHP 函数，并转到所选择函数的官方帮助网页。

**第 7 章 使用 SVG 实现 Ajax 实时绘制图表** 教读者使用 Ajax 和 SVG 来实现实时绘图。SVG 是一种可以用来绘制图形和文本的制图语言。

**第 8 章 Ajax 数据表格** 指导读者建立一个功能强大的带有 Ajax 功能的数据表格，读者将学会如何利用 XSLT 解析 XML 文档来生成表格。

**第 9 章 Ajax RSS 阅读器** 使用 SimpleXML PHP 库、XML 和 XSLT 来建立一个简单的 RSS 阅读器。

**第 10 章 Ajax 的拖动功能** 使用 script.aculo.us 框架构建一个有拖动功能的简单元素列表的示例。

**附录 A 环境配置** 讲述怎样安装和配置所需要的软件，如 Apache、PHP、MySQL 及 phpMyAdmin 等。本书的例子要求读者搭建好运行环境以及与本书相关的示例数据库。

在本书的网站 <http://ajaxphp.packtpub.com> 上，可以找到本书所有 Ajax 学习实例的在线演示。

## 阅读本书的前提

阅读本书的例子需要了解 PHP 5，它既是一种 Web 服务器，也是一个数据库服务器。本书的代码已在多个环境下测试过，大多数情况采用的 Web 服务器是 Apache 2，数据库采用 MySQL 4.1 和 MySQL 5。

也可以选择其他 Web 服务器，或者其他数据库服务器。在这种情况下，各章中介绍的过程也许不是百分之百正确。最重要的是采用 PHP 5 或更新的版本，因为我们使用了它的一些特性，如支持面向对象编程，它的低版本并不支持这些特性。

关于计算机配置的更多细节请阅读附录 A。如果您的计算机已经安装了所需要的软件，同样需要阅读最后一章和附录 A，这两章讲述了如何为本书使用的例子创建数据库。

## 本书约定

在本书中有很多不同格式的文字，它们表示了不同类型的含义。下面是一些例子的格式和它们代表的意义。

代码的格式有 3 种。代码之前的说明性文字和代码中的说明性文字一样，都用引用符号“//”引出。例如以下代码：

```
// 使用 XMLHttpRequest 对象调用服务器
function process()
{
    // 获取表单中用户输入的用户名
    name = encodeURIComponent(document.getElementById("myName").value);
    // 从服务器端执行 quickstart.php
    xmlhttp.open("GET", "quickstart.php?name=" + name, true);
    // 向服务器端发送异步请求
    xmlhttp.send(null);
    // 读取响应
    handleServerResponse();
}
```

如果希望读者注意代码中的某个部分，则相关的行或内容会使用黑体字：

```
// 使用 XMLHttpRequest 对象调用服务器
function process()
{
    // 获取表单中用户输入的用户名
    name = encodeURIComponent(document.getElementById("myName").value);
    // 从服务器端执行 quickstart.php
    xmlhttp.open("GET", "quickstart.php?name=" + name, true);
    // 向服务器端发送异步请求
    xmlhttp.send(null);
    // 读取响应
    handleServerResponse();
}
```

所有的命令行输入、输出使用下面的格式：

```
./configure --prefix=/usr/local/apache2 --enable-so --enable-ssl --with-ssl
```

`--enable-auth-digest`

引入新词和重要的词汇时使用黑体字。在计算机屏幕上的词汇，如菜单、对话框等，在本书中的格式类似于“单击 **Next** 进入下一屏”。



**注意**

警告或重要内容使用这样的方形框。



**提示**

提示和窍门使用这样的格式。

## 客户支持

既然您是尊敬的 Packt 图书的拥有者，我们将从您购买时起竭诚为您提供服务。

## 下载本书的实例代码

请访问 <http://www.packtpub.com/support>，从列表中选择书名就可以下载到本书实例代码或其他相关的资源。在网页上会显示可下载的文件。



**注意**

可下载文件包中含有使用指南。

## 勘误

虽然我们尽力确保书中的内容准确无误，但还是无法避免发生错误。如果发现本书的任何错误——可能是文章或代码的错误——请将错误反馈给我们，我们将非常感谢您。因为这将使其他读者免受困扰，并帮助我们在后续版本中有所提高。如果发现错误，请登录 <http://www.packtpub.com/support>，选择书名，单击 Submit Errata 链接，并输入详细的错误信息。一旦提交的错误得到了证实，提交的内容将显示在已有错误列表中，可以通过 <http://www.packtpub.com/support> 查看。

## 问题

如果在阅读本书时遇到问题，请发邮件到 [questions@packtpub.com](mailto:questions@packtpub.com)，我们将尽力解答。

# 目 录

|                                              |    |
|----------------------------------------------|----|
| 第 1 章 Ajax 与未来的 Web 应用程序 .....               | 1  |
| 1.1 通过 Web 发布的功能 .....                       | 2  |
| 1.2 20 世纪 90 年代开始出现网站 .....                  | 3  |
| 1.2.1 HTTP 与 HTML .....                      | 3  |
| 1.2.2 PHP 及其他服务器端技术 .....                    | 5  |
| 1.2.3 JavaScript 及其他客户端技术 .....              | 5  |
| 1.2.4 缺少的是什么 .....                           | 6  |
| 1.3 理解 Ajax .....                            | 7  |
| 1.4 使用 Ajax 和 PHP 建立一个简单应用程序 .....           | 10 |
| 1.5 小结 .....                                 | 22 |
| 第 2 章 JavaScript 下灵活的客户端技术 .....             | 23 |
| 2.1 JavaScript 和 Document Object Model ..... | 23 |
| 2.2 JavaScript 事件和 DOM .....                 | 27 |
| 2.3 关于 DOM 的进一步介绍 .....                      | 31 |
| 2.4 JavaScript、DOM 和 CSS .....               | 34 |
| 2.5 使用 XMLHttpRequest 对象 .....               | 38 |
| 2.5.1 创建 XMLHttpRequest 对象 .....             | 39 |
| 2.5.2 使用 XMLHttpRequest 初始化服务器请求 .....       | 43 |
| 2.5.3 服务器响应处理 .....                          | 46 |
| 2.6 使用 XML 结构 .....                          | 54 |
| 2.6.1 处理更多的错误和抛出异常 .....                     | 60 |
| 2.6.2 建立 XML 结构 .....                        | 63 |
| 2.7 小结 .....                                 | 64 |
| 第 3 章 使用 PHP 和 MySQL 实现服务器端技术 .....          | 65 |
| 3.1 PHP 与 DOM .....                          | 65 |
| 3.2 参数传递与 PHP 错误处理 .....                     | 72 |

|              |                              |            |
|--------------|------------------------------|------------|
| 3.3          | 连接远程服务器与 JavaScript 的安全性     | 83         |
| 3.4          | 使用代理服务器脚本                    | 89         |
| 3.5          | 重复异步请求框架                     | 96         |
| 3.6          | 使用 MySQL                     | 108        |
| 3.6.1        | 创建数据库表                       | 109        |
| 3.6.2        | 数据操作                         | 111        |
| 3.6.3        | 连接数据库并执行查询操作                 | 112        |
| 3.7          | 程序封装与程序结构                    | 117        |
| 3.8          | 小结                           | 130        |
| <b>第 4 章</b> | <b>Ajax 表单验证</b>             | <b>131</b> |
| 4.1          | 实现 Ajax 表单验证                 | 132        |
| 4.2          | 小结                           | 161        |
| <b>第 5 章</b> | <b>Ajax 聊天</b>               | <b>162</b> |
| 5.1          | Ajax 聊天简介                    | 162        |
| 5.2          | 实现 Ajax 聊天                   | 163        |
| 5.3          | 小结                           | 186        |
| <b>第 6 章</b> | <b>Ajax 建议和自动完成</b>          | <b>187</b> |
| 6.1          | Ajax 建议和自动完成简介               | 187        |
| 6.2          | 实现 Ajax 建议和自动完成              | 188        |
| 6.3          | 小结                           | 215        |
| <b>第 7 章</b> | <b>使用 SVG 实现 Ajax 实时绘制图表</b> | <b>216</b> |
| 7.1          | 使用 Ajax 和 SVG 实现实时图表         | 216        |
| 7.2          | 小结                           | 231        |
| <b>第 8 章</b> | <b>Ajax 数据表格</b>             | <b>232</b> |
| 8.1          | 使用客户端 XSLT 实现 Ajax 数据表格      | 233        |
| 8.2          | 小结                           | 257        |
| <b>第 9 章</b> | <b>Ajax RSS 阅读器</b>          | <b>258</b> |
| 9.1          | 使用 RSS                       | 258        |
| 9.1.1        | RSS 文档结构                     | 258        |
| 9.1.2        | Google Reader                | 259        |

---

|                               |     |
|-------------------------------|-----|
| 9.2 实现基于 Ajax 的 RSS 阅读器 ..... | 260 |
| 9.3 小结 .....                  | 275 |
| 第 10 章 Ajax 的拖放功能 .....       | 276 |
| 10.1 在 Web 上使用拖放功能 .....      | 276 |
| 10.1.1 购物车 .....              | 276 |
| 10.1.2 分类列表 .....             | 277 |
| 10.2 创建 Ajax 拖放分类列表应用 .....   | 277 |
| 10.3 小结 .....                 | 296 |
| 附录 A 环境配置 .....               | 297 |
| A.1 Windows 下的环境配置 .....      | 298 |
| A.1.1 安装 Apache .....         | 298 |
| A.1.2 安装 MySQL .....          | 300 |
| A.1.3 安装 PHP .....            | 300 |
| A.2 *nix 下的环境配置 .....         | 302 |
| A.2.1 安装 Apache .....         | 302 |
| A.2.2 安装 MySQL .....          | 302 |
| A.2.3 安装 PHP .....            | 303 |
| A.3 安装 phpMyAdmin .....       | 304 |
| A.4 配置 Ajax 数据库 .....         | 305 |

# 第 1 章

## Ajax 与未来的 Web 应用程序

我的小堂弟第一次见到计算机时说“计算机，给我画一个机器人！”（我解释说是因为我教导计算机不要听陌生人的话，所以它才不接受这个命令）。如果您是我，您的第一反映也许是“多愚蠢”或“多好笑”，但这样的想法是错误的。我们所接受的教育和大脑的思维模式已经使我们在很大程度上学会了如何使用计算机。人们要接受教育来适应计算机，弥补计算机无法像人一样思考的缺陷（另一方面，人类并不能很好地了解自身，但这是另一码事）。

这个小故事与人类本能的使用计算机的方式有关。在一个理想世界里，计算机应该能够按照我堂弟的口头命令完成任务。在过去的几年里，实现用户友好的技术水平已经得到了快速的发展，但是，距离真正意义上的智能计算机的出现还有相当长的路要走。在此之前，人们需要学习的是如何让计算机来为自己工作，在某种程度上结束向 DOS 窗口输入命令的习惯。

许多计算机的工作方式是通过软件来完成的，它通过用户界面，允许人们凭直觉（或喜好）来工作，这种现象并不是偶然的。这就像是单击鼠标右键的功能，如拖放功能，或通过简单的文本框就可以在 0.1 秒的时间里在整个 Internet 上搜索相关的内容等奇妙功能。软件工业就是这样（或者说其中有利可图的部分），它是一个发现、分析和学习的过程。现今市场上到处都是这样昂贵的程序：有着鲜明的按钮、图标、窗口以及向导。

通过软件工业我们发现，就像是一辆红色运动跑车需要有一个强大的引擎一样，软件需要具有可用性和易用性。当以商业的眼光和人性的眼光来看都觉得很好的时候，那将是一件非常美好的事情，因为商业利润是或多或少地与客户的满意度成比例的。

我们计划在本书中采用非常简单实用的方式进行讲解，但在进行您最喜欢的事情（编写代码）之前有必要退回小小的一步，这就是记住我们在做什么，以及我们为什么要这么做。我们总是喜欢按部就班地使用一项技术，却容易忘记技术存在的真正目的是为了人们的生活更加愉悦，工作更加高效。

理解人类大脑的工作方式是建立最优应用软件的关键。我们要达到这一目标还相距甚远，而我们应做的是为终端用户提供直观的用户界面。终端用户并不真正关心他们所使用的操作系统，只要系统按要求正常工作。这对程序员来说，是非常重要的而且必须牢记的一点，许多程序员即使在与终端用户打交道的时候也习惯于使用技术的观点来思考和交谈（尽管在

一个典型的开发团队中程序员不直接和终端用户打交道)。如果不同意,请回想一下在和非技术人员交谈的过程中您使用过多少次数据库这个词。

在使用计算机系统的过程中通过观察人们的需求和习惯,产生了软件可用性 (software usability) 这个术语——它是一种艺术,要迎合用户对界面的期望,要理解用户工作的规律,从而开发软件应用。

以前,可用性技术主要用于桌面应用程序 (desktop applications) 中,这只是因为当时没有 Web 应用程序 (Web applications) 工具。然而,随着 Internet 技术的逐渐成熟,这项技术得到了越来越有力的支持。

现代因特网技术不仅支持搭建更好的在线系统,同时还支持搭建更好的内部或专项应用。拥有界面友好的网站对于在线业务来说至关重要,因为因特网从不停息,用户经常会迅速地迁移到另一个更大,看起来或感觉更好的网站上。同时,建立友好的 Web 用户界面为企业软件提供了另一种选择,因为以前企业主要使用的是桌面应用。

通常在建立友好的用户界面方面,桌面应用要比 Web 应用更简单,因为在 Web 最初的设计上,它主要是用于传送文字和图片,而没有其他复杂的功能。在过去的几年里这个问题明显变得更加重要,因为越来越多的软件服务和功能通过 Web 来提供。

同时,为了增强页面的动态性,开发了许多其他技术(还有一些技术正在开发中),这使得 Web 应用的功能得到了增强。比较著名的例子有 Java applets 和 Macromedia Flash,它们都需要用户为浏览器安装它们各自的组件。

## 1.1 通过 Web 发布的功能

Web 应用程序是指通过 Web 服务器来完成应用程序的功能,并将运行的结果通过网络(如因特网或企业内部网)传递给终端用户。终端用户使用瘦客户 (thin client, Web 浏览器) 来运行 Web 应用程序,客户端知道如何显示和执行从服务器端接收到的数据。而桌面应用程序的大多数的数据处理功能是以厚客户 (thick client, 也称作富客户或胖客户) 为基础来实现的。

Web 应用程序的梦想就是希望有一天它能在用户界面和运行效率方面都像它成熟(而且功能强大)的亲戚——桌面应用程序一样强大。任何一个人们使用的计算机软件都要比它以前更重要,因为现今的计算机用户比过去的更加多样化,过去的用户主要是技术用户。现在却需要将一份表示清楚的报告交给销售部经理 (Cindy), 而且还需要为销售员 (Dave) 提供易于使用的数据输入表单。

编写的应用软件要让每一个使用它的用户都感到满意,因为终端用户满意至关重要。就出现 Web 应用程序而言,当应用的界面和操作不暴露出它的功能是通过本地桌面还是光纤或远程实现时,就完成了它进化到成熟的过程。通过 Web 传递可用的界面曾遭到过质疑,仅仅因为过去人们无法在桌面应用中完成拖放和在一个窗口下同时执行多任务这样的功能。

建立 Web 应用程序的另一个问题就是标准化。现在，每一个 Web 访问都必须通过至少两种或三种浏览器的测试，以确保所有的访问者都能充分利用网站。

## Web 应用程序的优势

诚然，试图利用 Web 来传递功能的时候会遇到很多头疼的事情。那为何还要不厌其烦地选择 Web 而不直接使用桌面应用程序呢？尽管 Web 应用程序目前还有一些问题，但它具有用户友好性，它们得到了相当大程度的普及，因为它们与桌面应用程序相比，有着大量的重要技术方面的优势。

- **Web 应用程序易于传递而且发布便宜。**使用 Web 应用程序，公司可以减少 IT 部门客户端软件安装成本。对于 Web 应用程序来说，所有用户所需要的只是一台接入因特网或企业内部网的计算机和一个能正常工作的 Web 浏览器。

- **Web 应用程序易于更新而且便宜。**软件维护成本通常是很惊人的。因为更新现有的软件和安装新软件的成本很接近，前面提到的 Web 应用程序的优点在这里同样适用。只要服务器端程序得到了更新，客户就能使用新版本了。

- **Web 应用程序针对终端用户的需求变化灵活。**只需要将 Web 应用程序安装在服务器上——任何现代操作系统均可——就可以立即使用 Mac、Windows 或 Linux 系统通过因特网/企业内部网进行访问。只要应用程序安装恰当，任何一个现代的 Web 浏览器，如 Internet Explorer、Mozilla FireFox、Opra 或 Safari 上看到的效果都是一样的。

- **Web 应用程序易于实现数据集中存储。**当多个地方需要访问相同的数据时，将所有数据存放在一个地方要比数据分散存储容易得多。这样可以避免潜在的数据异步操作，降低安全风险。

本书将进一步研究如何使用现代 Web 技术来建立更好的 Web 应用程序，并尽可能多地使用 Web 提供的功能。在介绍详细内容之前，让我们简单地回顾一下历史。

## 1.2 20 世纪 90 年代开始出现网站

虽然 Internet 出现的历史要比这个时间稍微长一些，但超文本传输协议（HTTP）出现在 1991 年，当时只用于在 Internet 上传送数据。在它早期的几个版本里，其功能只是打开和关闭链接。HTTP 后期的版本（1996 年出现的 1.0 和 1999 年出现的 1.1）成了现在读者所熟悉和使用的协议。

### 1.2.1 HTTP 与 HTML

所有的 Web 浏览器都支持 HTTP，HTTP 出色地完成了当初的构想——解析简单的 Web 内容。无论何时通过读者喜欢的 Web 浏览器发送网页请求，都得使用 HTTP。比如，在 FireFox