祝红涛 刘海松 郝军启 编著

Ajax 从入门到精通

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Ajax从入门到精通

祝红涛　刘海松　郝军启　编著

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本书循序渐进地向读者展示了在开发中所需要掌握的Ajax知识，引导读者快速掌握Ajax技术。本书内容全面，涵盖了从事Ajax开发所要掌握的所有知识。在知识的介绍上，本书采用理论与实践相结合的方式，从程序运行的内部机制进行分析讲解，并通过大量的实例来验证及运用本书所讲知识。本书语言生动、通俗易懂、讲解细致，每个知识要点都有相应的实例。而且很多实例都是目前Ajax开发经常使用的功能，具有相当高的实用价值。

本书不仅可以作为Ajax的入门级教程，还可以作为从事Ajax开发的程序员的参考用书和必备手册。

图书在版编目（CIP）数据

Ajax从入门到精通/祝红涛，刘海松，郝军启编著.—北京：电子工业出版社，2008.6
ISBN 978-7-121-06353-4

Ⅰ.A… Ⅱ.①祝…②刘…③郝… Ⅲ.计算机网络—程序设计 Ⅳ.TP393.09

中国版本图书馆CIP数据核字（2008）第047369号

责任编辑：李红玉
印　　刷：北京天竺颖华印刷厂
装　　订：三河市金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
　　北京市海淀区翠微东里甲2号　邮编：100036
开　　本：787×1092　1/16　印张：27　字数：690千字
印　　次：2008年6月第1次印刷
定　　价：48.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

Ajax，异步JavaScript与XML，是使用客户端脚本与Web服务器交换数据的Web应用开发方法。这样，Web页面不用打断交互流程进行重新加载，就可以动态地更新。使用Ajax，可以创建接近本地桌面应用的、直接的、高可用的、更丰富的、更动态的Web用户接口界面。

Ajax技术是基于被各大浏览器和平台都支持的公开标准的技术。组成Ajax技术的大多数技术都经过很多年的实践考验，而不是那些热点的、最新的和未经考验的技术。现在，对于绝大多数的用户和企业来说，浏览器是一个可信任的应用平台，这在五年前就不是个问题了。对于Ajax来说，FireFox浏览器的基础Mozilla的发布是一个转折点。这种允许异步数据交换的技术好多年前就被IE浏览器支持了。这种支持和FireFox浏览器的大量被采用真正地使人们理解了跨浏览器的富Internet应用成为可能。

Ajax技术的广泛采用，已被业内领袖证明了市场的欢迎程度和该技术组的正确。每一个该技术的使用者都成为了胜利者：包括Google、Yahoo、Amazon和微软，等等。是Google地图吸引了Web开发人员的目光，人们由此才揭开了罩在Ajax头上的面纱。

本书首先从通过分析Ajax相关技术入手，深入Ajax的技术细节；然后对现在流行的Ajax实现和框架进行了介绍，掌握Ajax的思考方式；最后通过一个大型的Ajax开发实例，让读者全面了解怎样将Ajax在实际项目中加以应用。全书共分16章，其内容如下所述：

第1章，Ajax简介。首先从Web历史的发展入手，介绍了Ajax技术出现的必然性，在论述中将Ajax技术同传统的Web技术进行对比，并详细介绍了Ajax技术的概念、特点和优势。在章节的后面给出了一个Ajax实例。

第2章，CSS样式。主要介绍了Ajax技术中的一个相关技术CSS。本章首先从CSS的语法入手，介绍了使用CSS的三种方式，并在后面介绍了CSS的高级技术滤镜、区块和鼠标指针。

第3章，JavaScript。主要介绍了JavaScript脚本语言的基本语法。本章首先从JavaScript语言概述入手，详细介绍了JavaScript脚本语言的语法、函数和对象。对于不懂JavaScript语言的读者是一个很好的入门章节。

第4章，XML编程基础。主要介绍了XML置标语言的语法和相关技术，如DTD、Schema和XSLT。XML在Ajax技术中充当数据传递的载体，通过对本章的学习，读者可以把握Ajax技术中数据的结构。

第5章，XMLHttpRequest异步传输对象。介绍了Ajax的核心技术XMLHttpRequest对象。本章首先从XMLHttpRequest对象定义和创建方式入手，介绍了该对象所具有的属性、方法和运行周期，并讨论与服务器进行通信的方法。最后以XMLHttpRequest对象的一个实例结束。

第6章，DOM。介绍了文档对象模型DOM对象。本章首先从DOM对象的历史入手，介绍了DOM的发展历史、概念、结构模型、DOM文档对象与DOM对象实例。最后以DOM实例结束。

第7章，Ajax客户端应用。介绍了Ajax技术在客户端可以完成的操作。本章首先从Ajax运行原理入手，介绍了Ajax的运行流程、客户端发送参数形式和客户端处理服务器端响应形式，并在章节的后面以导航树为例，介绍了Ajax客户端的应用。

第8章，Ajax基本技术实现。介绍了Ajax技术在Web网站上常用的经典案例。在介绍案例时，首先和传统的Web实现技术进行对比，显示出Ajax的优点。然后将Ajax案例划分为客户端代码、服务器端代码和运行三个板块，清晰地列出了案例的基本结构。

第9章，Ajax服务器端编程。介绍了Ajax服务器端编程技术JSP。本章首先从JSP技术的概念入手，介绍了JSP技术的页面元素、内置对象和JSP运行环境。在其后通过实例介绍了JavaBean和Servlet技术。最后以JSP作为服务器端技术，介绍了Ajax获取和写入客户端数据的实例。

第10章，Ajax设计模式。主要介绍了在Ajax中利用模式开发Web程序。本章首先从设计模式的原则入手，介绍了基本设计模式、常用设计模式、MVC模式和模式在Ajax中的应用。本章后面的几小节，详细介绍了Ajax视图、Ajax控制器和Ajax模型的创建。

第11章，JavaScript高级技术。主要介绍了JavaScript脚本语言的高级应用。本章首先从JavaScript的对象机制入手，详细介绍了JavaScript的继承、框架编程、正则表达式、测试框架和调试工具。对于想掌握JavaScript技术的读者，本章是一个很好的提高运用JavaScript技术水平的章节。

第12章，Ajax安全性和性能。主要介绍了Ajax在运行时，性能优化和安全性问题。本章首先从Web技术的安全问题讲起，介绍了Ajax访问远程服务、数据保护、Ajax性能和内存问题。

第13章，Ajax优化技术。介绍提高Ajax应用开发中常见的优化技术。本章首先从用Ajax开发高质量的应用开始，详细介绍了响应客户端优化、设计通知系统、实现通知框架和数据的时效性。

第14章，常用的Ajax框架。介绍了现在比较流行且常用的Ajax框架。本章首先详细介绍了基于客户端的框架，如Dojo、Open Rico和jQuery框架，然后介绍了Microsoft公司提供的ASP.NET框架，并在本章的最后一节介绍了其他常用的框架。

第15章，Ajax综合实例。介绍了使用Ajax技术实现的综合实例。本章选取了在Web技术上常见的、比较复杂的案例进行介绍。在介绍这些案例时，从客户端代码、服务器端代码两个方面入手，详细阐述了案例的实现细节。

第16章，在线OA办公系统。介绍了使用Ajax技术创建软件系统的流程、编码和测试。该程序实现过程中，包含了对数据库各种不同的操作，如分页、更新、删除和采用数据库函数等，体现了Ajax的异步传输数据和动态更新的功能。

通过阅读本书，读者将全面了解近年来最热门的Web编程技术Ajax。本书适合有一定Java基础的读者阅读，可作为在校学生，中高级技术开发工程师和其他IT技术人员的参考书，也可作为大专院校和培训机构的教学用书。

参加本书编写与制作的人员除封面署名者以外，还有陈红旭、董志鹏、赵俊昌、秦雨、朱珀煜、李振、王俊伟、唐有明、王咏梅、郑千忠等。在此，对他们表示衷心的感谢。由于编写时间仓促，加之作者水平有限，书中难免会有错误和疏漏之处，恳请广大读者给予批评和指正。

目　录

第1章　Ajax简介

内容摘要 Abstract

当今时代的因特网已发生了翻天覆地的改变，最早它只是提供对文本的浏览等有限的功能，以便科学家们进行学术交流。而如今，它已经成为商务和信息的中心。这期间，许多新方法和新技术相继亮相，因特网已经是大量应用程序部署的首选平台。然而，尽管因特网提供了很大的便利，但是Web应用与桌面应用程序还是存在明显的不同。本章将先对Web应用的发展里程做简要的回顾，通过对传统Web应用解决方案的分析，介绍一种解决Web应用的新技术——Ajax。

在本章将对Ajax进行全面的概要介绍，使读者对Ajax有一个概要的了解，为以后章节的学习做铺垫。

学习目标 Objective

- 了解Web发展史
- 了解Ajax的背景
- 了解Ajax的特性
- 了解Ajax的优势以及特征
- 了解Ajax的知识体系结构
- 理解Ajax的基本思想
- 理解Ajax的工作原理及基本原则

1.1　Web应用简史

起初为了连接美国几个顶尖的科研机构，人们设计了最早的Internet网络，以便共同开展科学研究，并交流信息。为了使用Internet，使用者必须学习一个相当复杂的系统。就是在1962年，麻省理工学院最早提出的“Galactic Network”（超大网络）思想时，IE和Firefox之类的便捷工具连最基本的概念都未形成。

Web这个Internet上最热门的应用架构是由Tim Berners-Lee发明的。Web的前身是1980年Tim Berners-Lee负责的Enquire（Enquire Within Upon Everything的简称）项目。1990年11月，第一个Web服务器nxoc01.cern.ch开始运行，Tim Berners-Lee在自己编写的图形化Web浏览器“World Wide Web”上看到了最早的Web页面。1991年，CERN（European Particle Physics Laboratory）正式发布了Web技术标准。目前，与Web相关的各种技术标准都由著名的W3C组织（World Wide Web Consortium）管理和维护。

从技术层面看，Web架构的精华有三处：用超文本技术（HTML）实现信息与信息的连接；用统一资源定位技术（URL）实现全球信息的精确定位；使用新应用层协议（HTTP）实

现分布式的信息共享。这三个特点无一不与信息的分发、获取和利用有关。其实，Tim Berners-Lee早就明确无误地告诉我们：“Web是一个抽象的（假想的）信息空间。”也就是说，作为Internet上的一种应用架构，Web的首要任务就是向人们提供信息和信息服务。

1.1.1 Web技术

Web是一种典型的分布式应用架构。Web应用中的每一次信息交换都要涉及到客户端和服务端两个层面。因此，Web开发技术大体上也可以被分为客户端技术和服务器端技术两大类。

1. 客户端技术发展

Web客户端的主要任务是展现信息内容，而HTML语言则是信息展现的最有效载体之一。最初的HTML语言只能在浏览器中展现静态的文本或图像信息，这满足不了人们对信息丰富性和多样性的强烈需求，这件事情最终的结果是，由静态技术向动态技术的转变成为Web客户端技术演进的永恒定律。

能存储、展现二维动画的GIF图像格式早在1989年就已发展成熟。Web出现后，GIF第一次为HTML页面引入了动感元素。但更大的变革来源于1995年Java语言的问世。Java语言天生就具备的平台无关的特点，让人们一下子找到了在浏览器中开发动态应用的捷径。1996年，著名的Netscape浏览器在其2.0版中增加了对Java Applets和JavaScript（它与Java并没有必然联系）的支持。

最初，创建JavaScript是为了帮助开发人员动态地修改页面上的标记，以便为客户提供更丰富的体验。随着对事物认识的加深，人们认识到：页面也可以当作对象。因此文档对象模型（Document Object Model，DOM）应运而生。起先，DOM和JavaScript紧密地交织在一起，但最后它们各自发展。DOM是页面的一个完全面向对象的表示，该页面可以使用其他脚本语言（如JavaScript或VBScript）进行修改。由于许多因素的存在，导致开发JavaScript有很大的困难，使许多开发人员对JavaScript退避三舍，有些开发人员干脆不考虑JavaScript，认为这是图形设计人员使用的一种“玩具”语言。另外，调试复杂的JavaScript使很多使用它的开发人员身心疲惫，所以大多数人在经历了这种痛苦后，只把JavaScript用作基于表单的验证工具。

1996年夏天，Future Wave发布了一个名字叫Future Splash Animator的产品。这个产品起源于一个基于Java的动画播放器。Future Wave很快被Macromedia兼并，并将这个产品改名为Flash。利用Flash，设计人员可以创建令人惊叹的动态应用。公司可以在Web上发布动态交互性的应用，几乎与富客户应用相差无几。Flash不需要编程技巧，不过这种易用性也是有代价的。Flash需要客户端软件并且Flash应用可能还需要大量网络带宽才能正常地工作；使用完整的Flash工具包需要按用户数付费，这虽然不是难以逾越的障碍，但它们确实减慢了Flash在动态Web应用道路上的前进步伐。

2. 服务器端技术发展

与客户端技术从静态向动态的演进过程类似，Web服务端的开发技术也是由静态向动态逐渐发展、完善起来的。

最早的Web服务器简单地响应浏览器发送来的HTTP请求，并将存储在服务器上的HTML

文件返回给浏览器。一种名为SSI（Server Side Includes）的技术可以让Web服务器在返回HTML文件前，更新HTML文件的某些内容，但其功能非常有限。第一种真正使服务器能根据运行时的具体情况，动态生成HTML页面的技术是CGI（Common Gateway Interface）技术。CGI技术允许服务端应用程序根据客户端请求，动态生成HTML页面，这使客户端和服务端动态信息交换成为可能。随着CGI技术的普及，聊天室、论坛、电子商务、信息查询、全文检索等各式各样的Web应用蓬勃兴起，人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。

早期的CGI程序大多是编译后的可执行程序，其编程语言可以是C、C++、Pascal等任何通用程序设计语言。为了简化CGI程序的修改、编译和发布过程，人们开始探寻用脚本语言实现CGI应用的可行方式。在此方面，不能不提的是Larry Wall于1987年发明的Perl语言。Perl结合了C语言的高效以及sh、awk等脚本语言的便捷，似乎天生就适用于CGI程序的编写。1995年，第一个用Perl编写的CGI程序问世。很快，Perl在CGI编程领域的风头就盖过了它的前辈C语言。随后，Python等著名的脚本语言也陆续加入了CGI编程语言的行列。

1994年，Rasmus Lerdorf发明了专用于Web服务端编程的PHP（Personal Home Page Tools）语言。与以往的CGI程序不同，PHP语言将HTML代码和PHP指令合成为完整的服务端动态页面，Web应用的开发者可以用一种更加简便、快捷的方式实现动态Web功能。1996年，Microsoft借鉴PHP的思想，在其Web服务器IIS 3.0中引入了ASP技术。ASP使用的脚本语言是我们熟悉的VBScript和JavaScript。借助Microsoft Visual Studio等开发工具在市场上获得成功，ASP迅速成为Windows系统下Web服务器端的主流开发技术。当然，以Sun公司为首的Java阵营也不会示弱。1997年，Servlet技术问世；1998年，JSP技术诞生。Servlet和JSP的组合（还可以加上JavaBean技术）让Java开发者同时拥有了类似CGI程序的集中处理功能和类似PHP的HTML嵌入功能。此外，Java的运行时编译技术也大大提高了Servlet和JSP的执行效率——这也正是Servlet和JSP被后来的J2EE平台吸纳为核心技术的原因之一。

3. 两种重要的企业开发平台

Web服务器端开发技术的完善，使开发复杂的Web应用成为可能。在此起彼伏的电子商务大潮中，为适应企业级应用开发的各种复杂需求，为给最终用户提供更加可靠、更加完善的信息服务，两个最重要的企业级开发平台，即J2EE和.NET，在2000年前后分别诞生于Java和Windows阵营。它们随即就在企业级Web开发领域展开激烈的竞争。从某种意义上说，也正是这种针锋相对的竞争促使了Web开发技术以前所未有的速度提高和跃进。

4. XML技术

HTML语言具有较强的表现力，但也存在结构过于灵活、语法不规范的弱点。为了克服HTML的不足，1996年，W3C在SGML语言的基础上，提出了XML（Extensible Markup Language）语言草案。1998年，W3C正式发布了XML 1.0标准。XML语言对信息的格式和表达方法做了最大程度的规范，应用软件可以按照统一的方式处理所有XML信息。这样一来，信息在整个Web世界里的共享和交换就有了技术上的保障。

1.1.2 Web开发框架和应用模型

2000年以后，随着Web应用的日益复杂，人们逐渐意识到，单纯依靠某种技术多半无法达到快速开发、快速验证和快速部署的最佳境界。研究者开始尝试着将已有的Web开发技术综合起来，形成完整的开发框架或应用模型，并以此来满足各种复杂应用的需求。

Microsoft在客户端的技术集成方面走在了最前面。1998年，Microsoft推出的Windows 98就可以在桌面上集成Web页面，这实际上是将资源管理器和Web浏览器的功能有效地结合了起来。2000年后，Microsoft陆续推出了MSN Explorer和与之相关的MSN在线服务。这一应用模型将Web浏览、视频点播、邮件处理、网上游戏、在线聊天等许多种用户常用的Web功能集成在一个统一的界面中。从信息利用的角度看，MSN试图让用户在一个最舒适的环境中获取足够的信息，这种努力的确值得人们称道。

另一个与客户端技术集成相关的例子是搜索引擎。Google在2003年展示Google搜索工具栏功能。安装Google工具栏之后的IE浏览器将信息浏览和信息检索有机地结合起来，这种小小的功能改进确实是对用户的体贴和帮助。

在Web服务端，2000年以后出现了几种主要的技术融合方式。首先，越来越多的Web开发环境开始支持MVC（Model-View-Controller）的设计模型，为开发者提供了全套的开发框架。实际上，J2EE和.NET平台本身就是这种开发框架的典型代表。其次，门户服务（Portal Server）和Web内容管理（Web Content Management）在最近几年里成为应用集成的重点模型。这两种应用模型可以直接为开发者或最终用户提供构建Web应用的高级平台，可以让Web开发和信息发布工作大为简化。在商业软件领域，这一类应用的例子包括Microsoft的SharePoint、IBM的WebSphere Portal、FileNet的Web Content Manager等。开源项目在Web开发框架和应用模型方面表现得非常积极，Struts、Jetspeed、jPortlet、Cocoon、Lenya、XOOPS等都是开源世界里与MVC开发框架、门户服务和Web内容管理相关的优秀解决方案。

当然，技术集成绝不等于技术堆砌。一些Web站点和Web应用的开发者把XML语言、MVC框架等时髦技术拼凑起来，却不管它们是否能适应具体的应用环境。结果，他们的系统要么运行效率低下，要么功能残缺不全。反之，一个值得注意的事实是，像新浪、搜狐或网易这样的门户网站，在他们的信息发布页面（如新闻页面）里，尽管信息内容时刻都在刷新，但Web服务器上存放的始终都是静态的HTML页面。这种“落后技术”的优点是，在大量并发访问的情况下，门户网站的响应速度仍然很快。深入到技术层面，我们通常会惊讶地发现，这些网站使用的大多是自行研发的Web内容管理系统。当网站的内容编辑提交新的信息时，系统会自动将信息转换为HTML格式，发布到Web服务器集群的每一个结点上。

这一份有趣的历史记录再一次印证了关于Web开发技术的基本观点：一种技术只要能为用户提供高水平的信息服务，它就是最好、最先进的技术。

1.1.3 Web 2.0是什么

Web 2.0是2003年之后互联网的热门概念之一，不过目前对什么是Web 2.0并没有很严格的定义。一般来说Web 2.0（也有人称之为互联网2.0）是相对Web 1.0的新一类互联网应用的统称。Web 1.0的主要特点在于用户通过浏览器获取信息，Web 2.0则更注重用户的交

互作用，用户既是网站内容的消费者（浏览者），也是网站内容的制造者。

Blogger Don在他的“Web 2.0概念诠释”一文中提到“Web 2.0是以Flickr、Craigslist、Linkedin、Tribes、Ryze、Friendster、Del.icio.us、43Things.com等网站为代表，以Blog（博客）、TAG（标签，它体现了群体的力量，使得内容之间的相关性和用户之间的交互性大大增强）、SNS（社会网络）、RSS（一种描述和同步网站内容的格式）、Wiki（百科全书）等社会软件的应用为核心，依据六度分隔、XML、Ajax等新理论和技术实现的互联网新一代模式。”

所以，到目前为止，对于Web 2.0概念的说明，通常采用Web 2.0典型应用实例介绍，加上对部分Web 2.0相关技术的解释，这些Web 2.0技术主要包括：博客（BLOG）、RSS、百科全书（Wiki）、网摘、社会网络（SNS）、P2P、即时信息（IM）等。

国内典型的Web 2.0网站主要包括一些以博客和社会网络应用为主的网站，尤其以博客网站发展最为迅速，影响力也更大，例如博客网（www.bokee.com）、DoNews IT社区（www.donews.com）、百度贴吧（post.baidu.com）和新浪博客（blog.sina.com.cn）等。

1.2 传统Web应用解决方案

提到Web标准的历史就不得不谈到一个经典的技术——HTML。事实上HTML技术是目前最优秀也是核心的Web技术之一。目前计算机上包含互联网在内的大部分应用程序在交互操作上的核心原理都来自HTML的链接设计思想。

早在1945年，Vannevar Bush就表达了一种利用文字之间相互链接来进行阅读与关联的创新思想，而在1946年IBM利用这种语言思想发明了GML语言。用于描述这种可以相互链接的文本信息，随着互联网的发展，一种专门用于超文本的描述语言HTML便诞生了。

超文本式的浏览从根本上改变了人们的阅读习惯，这种非线性的阅读方式可以灵活地组织信息的内容，用户也不再被从上到下的段落阅读方式所束缚。当用户在查阅文章的时候，可通过附加在某个关键字上的链接来查看相关的注释或者其他信息。更重要的是，这种链接文本的出现使得传统的信息构架可以进行合理的分类与检索，从而改变了信息的展示方式。

目前互联网上的优秀网站无不通过对信息进行合理的分析、分类与处理来创造商业财富价值，Google、Amazon、eBay等，通过信息的超文本式整理与业务模式进行整合。全新的商业模式带给用户与企业可观的价值。目前HTML也是最为普及的网页设计技术，无论是不同的操作系统或者浏览器都通过HTML进行信息的设计整合。最新的HTML 4.0版本也已经是一种非常成熟的页面描述语言，支持包括段落、列表、表格等众多元素对信息进行组织，还有一定的设计功能，能对版式、字体颜色大小及图片等信息做出控制，是目前最普及的网页设计技术。

然而仅仅依靠一种文本技术来进行页面表现是远远不够的，通过长期的技术制定，一种新的用于文本设计的标准也诞生了，就是CSS。

在CSS设计的初期，由于它的出现晚于浏览器的推出，没能被当时的浏览器所支持，所以一直未能普及，CSS直到Web 2.0版本才被广大Web设计师所接受。如果国内读者在1999年开始了解网页制作技术的话，应该能够体会到，当时通过CSS来定义全站文字的字体颜色和

链接样式的方式已经让相当多的网页网站设计工作变得高效灵活。

随着技术的发展与用户的需求，单纯的信息展示已经不能够满足大家对获取信息的需求。交互性是Web发展的重要标志之一，JavaScript的诞生恰好满足了日益增长的对页面交互的需求。它使得用户通过鼠标或者键盘的操作来对页面上的信息进行交互行为，实现增加、改变或者删除信息以及更丰富的交互方式。

时至今天，Web标准已经由一些构架分明的技术组成，这些技术都已经成为目前Web表现层技术的头号应用。

基于Web的应用很容易部署，这种低门槛正是Web应用最耀眼的地方。由于浏览器无处不在，而且不需要下载和安装新的软件，用户利用基于浏览器的客户端就能很容易地尝试新的应用。用户只需要单击一个链接就能够运行新的应用程序，而不用下载几兆位的安装程序。基于浏览器的应用不与特定的操作系统绑定，这就大大节约了开发和维护成本。

最初的因特网实际上是为科学家和学术机构之间交换静态信息的，这是一种简单的请求/响应模式。在那个时候，不需要会话状态，只需要进行简单的文档交换。然而现在对Web功能的需求有巨大的改变。因特网以请求/响应模式作为基础，由此带来的同步性给用户愉快的网络体验带来了阻碍。因为Web浏览器在与服务器交互的过程中，用户唯一能做的事情是等待，直到浏览器接收到服务器返回的信息后，用户才可以进行接下来的操作。这种传统的交互模式如图1-1所示。

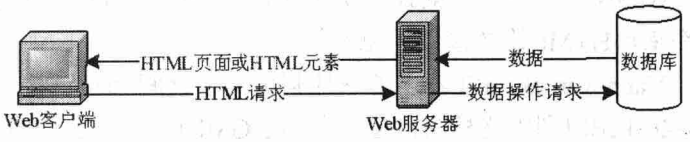

图1-1　传统Web应用模式

与Microsoft Word或WPS之类的富客户端应用相比，Web模型只能根据一般用户需要做折中考虑。不过，由于Web应用很容易部署，而且浏览器的发展相当迅速，大多数用户已经学会了适应。因为因特网是一个同步的请求/响应系统，所以浏览器中的整个页面会进行刷新。一般情况下这种请求模式并没有什么问题，但是如果用户只需要修改一两处，也不得不向服务器发送回整个文档，并且要重新绘制整个页面。尽管这样可行，但是这种完全刷新的局限，意味着应用还不够人性化，还是很粗糙。

因特网的宏伟蓝图是将这个世界上所有的计算机都连接起来，形成一个无比巨大的计算资源。如果能把本地调用和远程调用等同起来，那将是一件激动人心的事情。然而本地调用和远程调用是完全不同的东西。在现有的技术水平之下，网络通信仍然是一件代价高昂的事情（也就是说，通常很慢，而且并不可靠）。在没有网络调用的情况中，不同的方法和函数以及它们所操作的数据都位于相同的本地内存中，向方法内部传递数据并且获得方法的返回结果是非常直接的，因为程序逻辑与数据模型都保存在本地内存中，并且彼此可以直接访问。

远程调用是一个很复杂的交互过程。幸运的是，现代的编程环境如Java和Microsoft的.NET框架都内置了这个能力。尽管如此，执行远程调用时，上述所有这些操作仍然会在内部执行。如果我们到处使用远程调用，性能势必会大受影响。远程调用是不可能和本地调用一样有效

率的。更令人遗憾的是，网络的不稳定更让这种效率损失捉摸不定，难以预计。相比之下，运行在本地内存之中的本地调用，在这一点上具有得天独厚的优势。

然而幸运的是，现在我们又有一个新的选择，一个新的工具可以创建丰富的基于浏览器的应用。这就是Ajax（在下一节中将概要介绍Ajax）。Google是最早采用Ajax的公司之一，而且已经在很多技术中使用它，如Google Maps、Google Suggest和Gmail。Google Suggest会在用户输入搜索关键字时，动态给出相关的关键字，部分截图如图1-2所示。

图1-2 Google Suggest

1.3 Ajax技术介绍

2005年2月，Adaptive Path的Jesse James Garrett最早创造了“Ajax”这个词，Ajax不是一种技术。实际上，它由几种蓬勃发展的技术以新的强大方式组合而成。

Ajax不是单一的技术，而是四种技术的集合。表1-1简要介绍了这些技术，以及它们所扮演的角色。

表1-1 Ajax的关键元素

技术	功能简介
JavaScript	JavaScript是通用的脚本语言，用来嵌入在某种应用之中。Web浏览器中嵌入的JavaScript解释器允许通过程序与浏览器的很多内建功能进行交互。Ajax应用程序是使用JavaScript编写的
CSS（层叠样式表）	CSS为Web页面元素提供了一种可重用的可视化样式的定义方法。它提供了简单而又强大的方法，以一致的方式定义和使用可视化样式。在Ajax应用中，用户界面的样式可以通过CSS独立修改
DOM（文档对象模型）	DOM以一组可以使用JavaScript操作的可编程对象展现出Web页面的结构。通过使用脚本修改DOM，Ajax应用程序可以在运行时改变用户界面，或者高效地重绘页面中的某个部分
XMLHttpRequest对象	XMLHttpRequest对象允许Web程序员从Web服务器以后台活动的方式获取数据

Ajax为用户提供了复杂的、运转良好的应用，改善了用户的交互体验。Ajax中主要技术之间的关系如下所述：

JavaScript就像胶水将各个部分粘合在一起，定义应用的工作流程和业务逻辑。通过使用JavaScript操作DOM来改变和刷新用户界面，不断地重绘和重新组织显示给用户的数据，并且处理用户基于鼠标和键盘的交互。CSS为应用提供了一致的外观，并且为以编程方式操作DOM提供了强大的捷径。XMLHttpRequest对象（或者类似的机制）则用来与服务器进行异步通信，在用户工作时提交用户的请求并获取最新的数据。图1-3所示显示了这些技术在Ajax中是如何配合的。

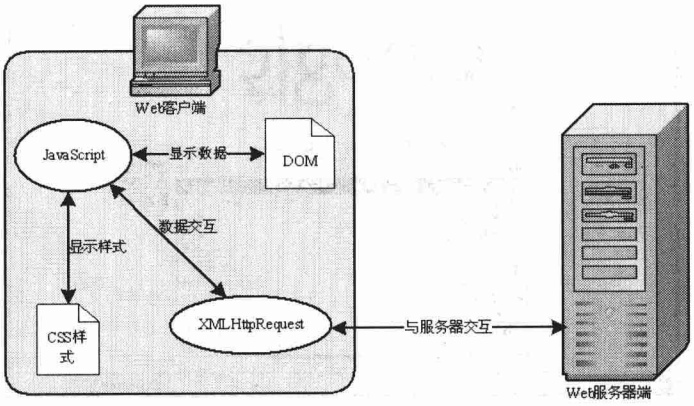

图1-3　Ajax的四个主要组件

JavaScript定义了业务规则和程序流程。应用程序使用XMLHttpRequest对象（或类似的机制）以后台方式从服务器获得数据，通过DOM和CSS来改变界面的外观。

Ajax的四种技术之中，CSS、DOM和JavaScript这三个都不是新面孔，它们以前合在一起称作动态HTML，或者简称DHTML。DHTML可以为Web页面创造新奇古怪的、交互性很强的界面，但是它永远也无法克服需要完全刷新整个页面的问题。问题在于，如果没有和服务器通信的能力，空有漂亮的界面，还是无法实现一些真正有意义的功能。Ajax除了大量使用DHTML，还可以发送异步请求，这大大延长了Web页面的寿命。通过与服务器进行异步通信，无须打断用户正在界面上执行的操作，Ajax与其前任DHTML相比，为用户带来了真正的价值。

更加方便的是，所有这些技术都已经预先安装在绝大多数的现代Web浏览器之中，包括微软公司的IE、Mozilla/Gecko系列的浏览器（例如Firefox、Mozilla Suite、Netscape Navigator和Camino）、Opera、苹果公司的Safari，以及它的近亲UNIX　KDE桌面系统里的Konqueror。遗憾的是，这些技术的实现细节在不同的浏览器之间，甚至在同一浏览器的不同版本之间存在着很多差异，这就是所谓的跨浏览器不兼容（cross-browser incompatibilities）问题。不过近几年来，这一状况得到了持续的改善。

Ajax的出现给JavaScript带来了生机，应用和测试框架再加上更优秀的工具支持，减轻了开发人员的重担。Ajax在大多数浏览器中都能够使用，并且不需要任何专门的软件或硬件。这种方式的一大优势就是开发人员不需要学习一种新的语言，也不必丢弃原先掌握的服务器