

1
Mastering JSP

TP393.092

1-58

JSP从入门到精通

〔美〕 Todd Cook 著

谷 雨 阎 隽 高春蓉 等译

李双庆 审校

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 提 要

本书从JSP最基本的概念到Web站点的设计,系统讲述了基于JSP的应用开发和设计,并提供了完整而实用的例程。全书共分3部分,由16章组成,包括JSP基础、应用程序开发、高级数据库编程、Web应用模型和高级体系结构以及用JSP表示XML文件等内容。

本书通俗易懂、由浅入深、偏重应用,适用于所有希望掌握JSP技术的人们,包括初学者和有经验的Web设计人员。



Copyright©2002 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版,SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号: 01-2002-2269

图书在版编目(CIP)数据

JSP从入门到精通/(美)库克(Cook, T.)著;谷雨等译.一北京:电子工业出版社,2003.3

书名原文: Mastering JSP

ISBN 7-5053-8451-1

I. J… II. ①库… ②谷… III. Java语言—主页制作—程序设计 IV. TP393.092

中国版本图书馆CIP数据核字(2003)第002992号

责任编辑:徐云鹏

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

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

经 销:各地新华书店

开 本:787×1092 1/16 印张:38.25 字数:970千字

版 次:2003年3月第1版 2003年3月第1次印刷

定 价:62.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:(010) 68279077

简介

近年来，公司员工所承受的工作压力和所承担的责任都比以前大多了，在很多情况下，员工知识面与业务能力处于同等重要的地位。这种倾向也影响到业界的各个角落，甚至专职员工也要求具有多种业务能力。

这就带来了服务器端Web程序设计的需求。我不断接到对各个领域的员工进行Web编程培训和指导的请求，包括C++程序员、系统管理员、程序分析员、生物学家以及其他各类人员。所有的这些人员都能够学会Web程序设计和JSP（JavaServer Page），都能够增强自己的技能。俗话说“技不压身”，在一个公司里，员工的知识面越宽，竞争力就越强。

目前，Web应用越来越普遍，已经成为商务的一部分。许多人们都能够使用HTML（超文本标记语言）和JavaScript设计自己的页面。各个行业的人们都希望能够提高自己全方位的技能，包括Web程序设计技能。现在，随着Tomcat 4的使用，用户能够安装运行使用JavaServer的Web服务器（在解压缩之后），而且有了JBoss，企业开发人员能够在台式机上运行成熟的Java 2 Enterprise Server。

遗憾的是，许多关于服务器端编程方面的书都没有讲清楚如何从HTML编码轻松过渡到服务器端的程序设计上去。撰写本书的目的就是帮助熟悉HTML、JavaScript和Java的初级和中级程序员成长为优秀的服务器端程序员。刚参加工作的人一般没有时间回到学校系统地学习计算机知识，虽然说业余学习也是一条不错的途径，但是一般来说，成人学生都很难摒弃已有的知识背景。这一点恰恰是JSP的优势：它是表示层的一种应用技术，因此，见效很快，但是却允许开发人员创建和访问异常复杂的Java对象和结构。

为了面向为新工作而焦虑的用户，为了给有经验的Java程序员提供表示层程序设计技巧，本书集中讲述专业JSP Web应用：

- 真正实用的范例。在本书中没有“你好，世界”等无聊的范例。
- 尽可能使用标准Java类。所有的范例都编译过，而且大多数情况下，各章的范例互相独立，没有使用不常用的类，所有的源代码都位于本书选配的光碟上。
- 范例文档齐全、注释清晰。代码清单采用标准书写格式，读者可以检查使用。对于掌握得好的读者，可以在运行范例之前，或者使用调试器调试之前仔细阅读源代码，弄清楚究竟为什么这样做。
- 本书中有习题和解答，但是读者最好根据实际情况设计方案。现实生活中很多事情并不是只知道答案就行，成功经常来自深思熟虑和正确行动。

本书结构

第1章介绍JSP和Servlet的基本概念、软硬件环境，并将JSP与其他Web编程技术进行比较。

第2章讲述HTTP、Web页面与服务器之间的交互，同时介绍Servlet以及如何使用Servlet，通过响应HTTP（超文本传输协议）客户请求创建应用程序；并举例说明动态配置Servlet，以便简化其操作和维护。

第3章重点讲述JSP处理以及从预先定义好的标签中（像HTML）中创建JSP。本章讲述并举例说明Scriptlet标签，该标签允许Java代码直接插入到JSP中。同时本章还将介绍JSP中的两个重要的概念：反射和类，这两个概念对于理解Java类在服务器端的工作机理非常重要。

第4章介绍JSP和Java类之间的基本交互机制，并简单提一下JavaBean的概念，为下一章内容做铺垫。

第5章详细介绍如何创建JavaBean以及JavaBean如何与JSP相结合进行代码重用并减少页面之间的混乱。本章举例说明JavaBean的使用，在所举的例子中用户能够进行参数自定义和设置。

任何应用程序都应该考虑防范非法入侵的安全问题。在第6章里，将列举页面之间共享数据时应注意的一些安全问题；讨论身份认证问题并提出有关安全方面的一些通用观点。

第7章讲述如何使用JDBC（Java数据库连接）访问数据库、数据表示技术和基本连接池技术，并举例说明基本SQL数据库的创建。

第8章讲述异常处理和调试。本章重点强调实际操作、测试以及如何恰当地与用户就异常问题进行沟通。本章详细列出了JSP开发过程中所遇到的所有的异常类型、排除方法和建议，介绍Java 1.4的一些新功能，举例说明通过扫描大型日志文件排除异常的过程。

第9章讲述性能评估以及如何与用户就性能问题进行沟通，同时对性能、优化、自我测试以及压力测试等设计问题进行讨论。

第10章介绍使用JSP来表示XML（扩展标记语言）文档。XML非常复杂，因此在本章的论述中，我们将分成几个部分，通过范例进行说明：XML入门、使用CSS（层叠样式表）显示XML、编写XSL（扩展样式表语言）终端、使用XSL参数显示XML、从XML和XSL来生成HTML、使用XML/XSL进行数据搬移。

第11章讲述JSP与已有应用系统的结合。内容包括编写Java封装器和系统调用封装器，以及用于封装器调用的一些常用语法。通过一个简单的封装器的例子来说明构建大型封装器的方法。本章还将介绍有关接口方面的知识，如JNI（Java Native Interface）和RMI（远程方法调用）。RMI是EJB（企业JavaBean）的基础，有关EJB的内容在本书最后一章中讲述。

第12章讲述Java扩展和JSP的结合。内容包括使用JAI（Java高级图像技术）扩展更改图形显示以及使用Java 2D API（应用程序接口）生成自定义的图形。本章还将介绍Java语言的国际化和本地化的有关特征。

第13章介绍模型显示控制器（MVC）Web应用的框架结构以及使用JSP或者基于servlet的控制器进行实现的过程。同时介绍在现实生活中部署应用程序时一些有用的实体类，并通过一个数据库应用的例子，说明实现动态身份认证和处理角色、许可和访问的方法。

第14章描述使用数据库时的一些常见问题与解决方案，解释JSP开发者在使用ResultSet类时可能会遇到的一些潜在的问题，并提供将数据访问从表示层移开的策略（通过一个范例进行说明）。

在构建JSP时，主要使用预先定义的JSP标签。幸运的是，JSP体系结构非常灵活，以至于允许用户自行创建标签。第15章介绍如何创建自定义标签、标签接口和工作机理，并对JSTL（Java标准模板库）进行综述。

第16章讲述JSP与EJB（企业JavaBean）之间的交互，以及移动数据交互代码的策略，并举例说明一个数据库驱动的框架，该框架用于抽象SQL查询业务，数据抽象是对EJB Entity Bean的进一步使用。开放资源J2EE服务器JBoss的使用促进了EJB的测试和开发。

附录A含有进行Tomcat JavaServer、MySQL数据库和JBoss J2EE服务器等安装所需要的有关信息；附录B介绍JSP和Servlet API；附录C是Java语言的初级读物，为第一次进行Java程序设计的初学者介绍Java语言以及对Java程序进行编译和打包，这对于任何对JavaServer程序设计感兴趣的人员来说都是非常重要、经常使用的技巧，遗憾的是大多数Java方面的书籍都没有这方面的内容。在这个初级读物中也包括一些常用的语言功能；附录D提供在Web编程中建议使用的JSTL API；附录E列出一些Web上经常用的实用软件。

本书选配光碟包括两部分内容：软件部分及其安装。软件部分包括下列应用的安装程序：Tomcat、Jboss、MySQL、JODE、Jbuilder、Struts和mm.mysql Driver。SRC部分包括在Tomcat Webapps路径下需要安装的模块包。

希望你能够喜欢本书。在使用本书的过程中如果有什么问题或建议，可以通过电子邮件与我联系，我的信箱是：masteringjsp@cookconsulting.com。

仅以此书献给Mackenzie和所有渴望掌握Java语言的学生。

——Todd Cook

致 谢

在本书的编写过程中，许多人给予了我极大的支持，首先我要感谢我的朋友和家人；尤其感谢Charles Mohnike对我的鼓励和忠告；感谢我的代理Jimmy Vines，他为本书的出版费尽了心思。Bettina Roth、Parag Agarwal、Matt Madsen为某些章节做了校对工作，一并表示感谢。

Sybex是一个享有盛誉的出版公司，在此我要感谢出版公司为本书的出版所做的一切。感谢Tom Cirtin，他的敬业为本书的质量提供了保证；感谢Pat Coleman，我为他娴熟的编辑技能而打动；感谢Liz Burke，是他负责了本书整个出版工作。

在本书的编写过程中，与我一同共事的读者提供了非常有用的资料，帮助我完善范例程序，使这些程序适用于初学者和各类编程高手。

再一次对曾经帮助过我的朋友和家人表示衷心的感谢！

译者序

由Sybex出版的《JSP从入门到精通》是一本关于服务器端程序设计的非常实用的参考书。

本书的作者Todd Cook先生是一位资深教育家，其主要研究领域是计算机语言和软件开发技术，曾经是Sun Microsystem等业内大公司的高级技术顾问。在做技术顾问的过程中，Todd Cook先生了解并理解了广大员工的真正需求，撰写本书的目的就是帮助熟悉HTML、JavaScript和Java的初级和中级程序员成长为优秀的服务器端程序员。

本书内容通俗易懂、由浅入深、语言简练、概念明确，从JSP最基本的概念到Web站点的设计，系统讲述了基于JSP应用的开发和设计。全书共分3部分，由16章组成，内容包括：JSP基本概念，使用JSP进行服务器端应用程序设计，专业JSP设计、开发和实现，每一章都有实用性很强的范例程序（范例程序源代码在本书选配的光碟上）。

通过本书的学习，能够迅速掌握使用JSP技术进行服务器端应用开发最简便、最快捷的方法。

本书适用的读者对象非常广泛，既可作为初学者的教材，又能作为Web开发人员、数据库人员、C++程序员等各类专业人员的参考书。

本书由谷宇、阎隽、高春蓉翻译，全书由李双庆审校，先后参加翻译工作的还有岳阳、单云、刘千里、郝茜、高春波、李苏云、许柏庆、陈延生、丘英、王向荣、张春学等同志。由于本书涉及的内容比较新，翻译过程中难免有疏漏或错误，欢迎广大读者批评指正。

光 碟 内 容

本书选配的光碟中有一个简便的Sybex界面，只要将盘片插入光驱并启动，就可以存取以下软件和范例源程序。

软件

Tomcat Server 4.0.3 Java Servlet和JavaServer Page技术在官方参考实现中的Servlet容器。Java Servlet和JavaServer Page技术是在Java Community Process中由Sun公司开发的。

JBoss 3.0 JBoss是用纯Java语言设计的应用服务器，是一种开放资源，与标准兼容。

MySQL 3.23.51 (Solaris、Windows、Macintosh和Linux版) **MySQL 3.23.51**服务器是现今世界上应用最广泛的开放资源数据库服务器。由于该软件所具有的创造性的软件体系结构，用户能够快速、简便地按照需求对其进行定制。

JODE 1.1 JODE 1.1是一个Java软件包，含有一个反编译器和Java语言的优化版。

Jbuilder 7 Jbuilder是最全面、并得到一致认可的可视化开发环境，用于为Java 2 Platform设计开发应用程序、Java小程序、JSP/Servlet、JavaBean、Enterprise JavaBean以及分布式J2EE应用。

Struts 1.0.2 该项目的目的就是为Web应用提供一个开放资源框架，其核心是基于标准技术（如JSP/Servlet、JavaBean、Resource Bundle和XML）的非常灵活的控制层。

范例源程序

本书中所用到的所有范例及源代码都打包成WAR（Web应用归档）文件，因此读者能够快速查看这些范例代码并安装运行。在附录A中有一个Java语言的入门教程，并介绍了如何准备Web应用归档文件。在附录C中介绍了控制文件、创建Web归档的过程。

目 录

简介	v
致谢	viii
译者序	ix
光碟内容	x
第一部分 JSP基础	1
第1章 JavaServer页面和JSP入门	1
服务器端Java应用的不断增长和JSP所处的地位	1
Web编程语言比较	4
不同服务器环境下的JSP	6
怎样起步	14
小结	14
第2章 HTTP与Servlet基础	15
HTTP的功能和局限性	15
HTTP请求	16
HTTP和Servlet	21
可配置Servlet编程	31
小结	42
第3章 JSP处理	43
JSP页面标签	43
JSP页面结构	53
处理JSP页面的服务器	54
小结	66
第二部分 JSP应用开发	67
第4章 使用脚本元素	67
JSP中的变量	67
理解隐含对象	71
使用流控制语句	72
显示值	75
共同放置	79

JavaBean代码支持本章中的例子	93
小结	104
第5章 JavaBean介绍	105
JavaBean概述	105
JavaBean设计注意事项	109
JavaBean范围	114
命名规则	115
JavaBean设计指导方针	118
反思：JavaServer引擎或者IDE如何学习JavaBean的风格	119
JavaBean中的错误	119
共同放置	123
小结	138
第6章 共享数据：JSP安全、身份认证和完整性	139
共享数据	139
进行身份验证	151
附加的安全关系问题	159
小结	164
第7章 数据库访问	165
数据库连接性和JDBC	165
连接池	177
小结	201
第8章 错误处理与调试	202
调试用户的经验	202
错误类型	205
处理问题	209
小结	231
第9章 性能	233
性能：可用性检查	233
性能的体系结构	234
实际JSP的性能：问题与解答	236
性能测试	242
小结	258
第10章 联合使用XML和JSP	260
必须学习用XML编写JSP吗	260
XML是什么	261

在客户端独立使用XML	268
编写XSL模板	271
小结	294
第11章 集成传统应用程序：包装和接口	295
Java包装	295
使用接口程序	317
远程方法调用（RMI）	321
小结	330
第12章 更多的扩充功能与国际化	331
改变图形的显示效果	331
应用程序国际化	352
小结	356
第三部分 专业级JSP设计、编程和实现	357
第13章 Web应用程序模型及高级结构	357
几个有用的组件	357
鉴别表	376
用JSP实现MVC	378
用Servlet控制器实现MVC	391
小结	404
第14章 高级数据库编程技术	405
java.sql.Connection和java.sql.ResultSet的缺陷	405
RowSet: JDBC 2.x类对ResultSet缺陷的解决方案	414
小结	442
第15章 自定义标记：Java标准标记库的组成	443
为什么要使用自定义标记	443
标记接口及其实现	444
用BodyTag标记转换文本	451
用确认标记预防错误	460
小结	469
第16章 企业JavaBean与JSP	470
EJB概述	470
Session Bean基础	471
Entity Bean基础	471

使用Entity Bean描述数据访问代码	471
用户概述查询的数据库驱动框架	480
使用AbstractedQuery类删除数据访问并处理代码	482
一个用于查询的简单Entity Bean, 可以读取、编写和创建功能	513
使用JBoss	541
小结	545
 附录	 547
附录A 安装	547
附录B JSP及Servlet API应用参考	557
附录C Java入门手册	577
附录D JSTL API使用参考	587
附录E Web资源	599



第一部分 JSP基础

第1章 JavaServer页面和JSP入门

在Web发展的早期（大约1995年前后），多数Web页面是在浅灰色的背景下布满黑字。后来出现了Java小程序，由于丰富多彩的设计，这些小程序很快便在世界范围内流行。对于人们来说，Web所提供的娱乐价值远大于其本身所能提供的信息量。很快，一些新技术（Flash 和ActiveX等）和新型脚本描述语言（VBScript和JavaScript等）开始使用。然而，所有的这些语言和工具带给人们感官上的满足都无法取代Java的优势：平台的独立性。后来，在Java作为服务器端首选的设计语言时这个优势起到了关键作用，而且，最终JavaServer页面也成为在Web上Java企业版的官方格式。

我们将在本章讨论JSP的作用和它的优势，并将其与同类技术进行对比。同时，本章将采用一个具体的Web程序设计来辅助学习JSP与几种常见的可用Web服务器环境。最后，将介绍开发JSP页面的客户端的集成开发环境（IDE）。本书中的范例程序均针对标准的Tomcat Java服务器进行设计（有关Tomcat的内容，请参考附录A内容）。

本章内容：

- Web编程概述
- 学习JSP所需的背景知识
- Web服务器环境
- 客户端IDE

服务器端Java应用的不断增长和JSP所处的地位

从最初开始，Java语言的营销口号就是“一次开发，随处可用”，而且，如果编写适当的话，Java程序设计能够做到100%纯Java，这就意味着，能够在任何带有Java虚拟机的平台下运行。当然，如果程序代码编写不够严谨的话，程序可能仅运行其中某一部分。

平台无关性是Java语言的一个重要特征，正是这个特征，使得Java语言能够作为一种标准语言将所有操作系统下的服务器页面统一到万维网（World Wide Web）上。虽然支持决大多数操作系统和硬件平台，但是，统计结果却表明，目前，Web服务器多数为运行UNIX或*nix的Apache Web服务器。

说明：编写*nix通常是指UNIX的不同版本，例如HP UNIX、SGI Irix、Linux、Sun Solaris UNIX、FreeBSD等等。请访问www.ugu.com。

说明：Apache由Apache Software Foundation开发，是目前世界上最流行的开放资源Web服务器，1995年前后开始面市。Apache Software Foundation也为Jakarta项目提供支持，均在Tomcat下用Java编写。

由于Web服务器上使用的操作系统种类繁多，因此“一次开发，随处可用”更像是一种夸张的营销手段，仅仅是一种需求。而且，Java的创始人和拥有者Sun微系统公司在UNIX平台上实现Java环境（可用Solaris源代码），而微软公司的视窗Java环境则位于视窗代码之上。幸运的是，事实上所有的Java版本都采取各种措施来平滑彼此之间的差异，因此，目前Java是一种稳定的企业环境。接下来，我们会进一步了解为什么决大多数Web服务器都选用Java环境。

什么是JSP页面

JSP页面是通过特殊的标记和Java代码进行扩展的HTML形式的Web页面，Web服务器和JavaServer引擎都采用这两种方式来创建客户机端的Web浏览器。

说明：JavaServer Pages (JSP) 是一种服务器端的编程语言，客户机端不使用原始的Java/JSP代码。Java能够以小程序的形式或者以安装在台式机上的应用程序的形式在客户机端使用。

警告：最靠近客户机端的脚本语言大概就是JavaScript，但是JavaScript不是Java。

JavaServer引擎抛去特殊的标记和Java源代码，并使用这些代码解析并创建Web服务器要发送的页面，实际上Java代码并不发送到客户机端，因此任意安装了Java环境或没有安装Java环境的Web浏览器都可以使用JSP应用。JSP页面实际上在具有Java环境的Web服务器上被编译成servlet，因此，为了使用JSP页面，Web服务器必须安装JavaServer引擎或者提供这种功能。本章后面的内容，将介绍几种不同的JavaServer引擎和JSP开发环境。

服务器端的Java出现得比较晚，Servlet是在Java 1.2版本时的应用。Servlet是只能在具有Java的Web服务器Servlet容器上运行的一种Java程序。Servlet container是一种能够安装Servlet和JSP页面的JavaServer引擎。Servlet只能在Web服务器上的Servlet容器内运行，不能在单独的应用或在浏览器窗口内运行，但是浏览器能从Servlet发送请求和接收响应。

Servlet接收一个请求，进行有关处理，然后发送结果。听起来似乎有些含糊，但是事实就是这样，因为Servlet具有相当含糊的能力。例如，设计一个应用，客户机端从这个应用中接收赛马的有关信息。一个Servlet能够返回统计信息，而另一个Servlet返回所选马匹的照片，另外一个Servlet可能调用数据库来记录所选马匹的声誉，并检查客户的连接速率。如果客户机正巧屏幕很小而且连接速率很慢，另一个Servlet将传递这些信息给一个图像Servlet，这个图像Servlet将返回最终的页面。Servlet可以联结在一起，这样一来，它们的功能就远远超过典型的Web页面的功能。

HTTP Servlet的处理过程和CGI程序类似，从客户机端的浏览器中接收原始输入，进行处理，然后返回结果。这就是说，在一个关键步骤上，它们具有另外的功能。例如，使用普

通HTML，Web站点能够指导用户选择产品，隐藏所做的选择（虽然方法有点笨），将用户引到一个提交页面，通过检查参数来处理最终的定单，调用数据库从客户账号上划账，从库中减去相应的产品。

CGI程序是用C语言编写的，Servlet充分利用了Java语言的结构化异常处理。下面我们将看到，JSP环境提高了故障处理能力。Java Servlet增加会话和Servlet上下文对象功能，在这一点上远远超过了CGI，第2章将讲述有关内容。

为什么要使用JSP

下面所列出的就是JSP的优点：

- JSP鼓励平台无关性，这样一来程序员精心编写的程序就能够做到“仅写一次，随处可用”。
- 对于企业服务来说，JSP是首选的表示层服务。
- 通过允许开发者使用和集成完全测试过的Java代码，JSP提供服务的稳定性。
- JSP鼓励应用的表示层和数据层之间明显的分界线。
- JSP与J2EE服务的无缝集成，增强了扩展体系结构的开发，而不仅仅限于相互联结的页面。

除了为JSP页面服务外，JavaServer引擎提供各种功能和存储器用于创建持续用户会话状态。因此，用户能够访问一个页面，开始一个定单业务，查看一些无关的联结，也许通过相关产品的联结进行访问，而这时候，JavaServer引擎则拿着用户的购物车耐心地等待，直到返回整个定单。在用户进行选择时，所有的幕后工作都不需要另外的编码来支持。巨大的省机制和代码的可重用使JSP在开发者之间迅速流行，尤其是与同类产品进行对比之后。

JSP很难学吗

从前面一些初学者的情况来看，只要有足够的热心和努力，任何人都能够精通JSP和Web编程。然而，基础不同，初学者所面临的困难也有所不同。

掌握HTML后开始学习JSP

对于掌握了HTML知识的开发者来说，由于HTML编码器通常知道为用户做什么，而且通常能给用户一个时间上的合理估计，他们所面临的困难是很容易被漂亮的图形所迷惑，而没有真正与有关数据库相互联系，进行实际的处理。

对于熟悉了HTML的用户来说，所面临的最大障碍就是Java和JavaScript不是一回事，虽然很多特征非常相似，但是这是两种不同的语言。困难的大小取决于用户希望掌握Java的程度。如果仅仅创建漂亮的JSP页面，很少的Java知识就足够了；如果希望进一步掌握甚至精通用户图形界面，例如GUI等来增强JSP功能，则需要花费较长的时间和精力。根据用户的需求和自身的背景知识，学习JSP其实并不难，只要认真学习JavaBeans、Java类和JSP标记库，很容易掌握复杂的Java和JSP知识，这些内容在后面的章节中我们将要讲述。

有C/C++或Java编程经验的人学习JSP

迟早，所有的应用程序员都会希望能够设计Web页面或者使用Web页面运行自己的程序。这种愿望正随着Web的迅速扩展以及源代码的HTML文档已经成为很多工具的标准特征而迅速增长。例如，捆绑在Java开发包（JDK）里的javadoc，能够从Java源代码中建立超级链接的HTML文件。其他的许多开发商也效仿JDK，纷纷在C、C++以及其他的编程语言中增加这项功能来创建HTML路径映射。

对应用程序设计者来说，最大的困难是时时想起和理解终端用户。这个观点看起来简单，事实上很具有迷惑性，而且许多工程师经常很快就忘记Web应用的生死存亡都是用户说了算。某种意义上来说，因为直到最近，全球计算机科学界才把为终端用户设计系统提到应有的重视程度上来，因此，毫不奇怪，这些工程师没有把用户放在首要的位置上来考虑。

在处理这个技巧的问题上，许多公司采取在项目组内加盟艺术总监（或导演）的方式，在补救这种不平衡上，无疑是一种很好的策略，然而，这与真正解决该类问题，并将该类问题解决在代码编写阶段还很远。对于这个问题，理解得越透彻，作为程序员来说，价值就越大。虽然不同的人对问题的理解上存在一定的差距，比如用户希望的Web页面形式和他们希望能够得到的帮助因人而异，相比较而言，理解用户的需求要比程序员拥有的专业知识更重要。

如何理解用户和经理的难以预测的需求，可以买几本有关方面的书来学习一下。我们推荐Jakob Nielsen编写的《Designing Web Usability》一书（该书1999年由New Riders Publishing出版），该书主要针对提高Web页面的可视性；以及Alan Cooper编写的《The Inmates Are Running the Asylum》（该书1999年由Sams出版），该书主要针对如何均衡软件设计和用户需求之间的矛盾。

Web编程语言比较

随着Web应用的成长，也带来了设计的复杂性。Web编程语言简化了原本非常困难的设计，并且将很多不可能的设想成为现实。本节内容，将重点讲述几种常用的Web编程语言，对比他们各自的优缺点，并适当地介绍使用JSP而不是其他语言的优点。

Perl

Perl（Practical Extraction and Report Language，实用萃取报告语言）对文本进行相当好的处理和格式化。可以通过使用代码模块进行扩展，许多操作系统都支持这种语言。

CGI（Common Gateway Interface，通用网关接口）是Perl在Web服务器环境中的具体应用。由于没有内置的共享应用资源（如会话跟踪、数据库连接池等等），因此Perl的资源非常昂贵。CGI/Perl的故障处理能力非常有限，必须先用CGI脚本进行Web编程，Perl在处理小型任务时功能强大，但是在处理代码重用等问题上明显不足。一般来说，由于其语言结构的特殊性，Perl学习起来很困难。

相比较而言，JSP具有专门的、能够自行定制的故障处理能力，在本书第4章和第8章中将要讲述有关内容。

ASP（活动服务器页面）

ASP是一种脚本语言，与JSP类似，能够为Microsoft的产品所创建的文档（如Excel电子表格等）提供基本支持。然而ASP只能在视窗Web服务器上运行。

说明：有一些公司已经开发了或者正在开发非Windows平台上的ASP引擎/解释器，然而，开发者必须精通所有的Microsoft语言。VBScript、C#和Microsoft的ASP框架都有各自的特点，ActiveX数据对象、COM（部件对象模型）DLL和COM服务器必须从Windows注册表和SCM（服务控制管理员）中提供。

ASP和IIS（因特网信息服务）、Microsoft的Web服务，都依赖于位于Microsoft服务器上的COM/COM+企业应用框架结构。这就限制了应用的灵活性，因为应用是将自己构建在企业的软硬件环境之上。

使用JSP而不用ASP的优点是JSP本身是跨平台的，没有浪费各种功能。JSP的骨干Java体系结构支持以源代码的形式或者近似替代物分布的部件，允许用户对产品具有更高级的拥有权和潜在的全程调试权利（源代码可以被重新编译），而ActiveX和COM/COM+旨在建立到Windows平台上的坚固的“黑盒”服务。

ColdFusion

ColdFusion在Web应用工具中相当独特，因为它使用了一种基于标记的脚本描述语言。当用户浏览ColdFusion创建的源文件时，可以看到它的标记集和HTML标记的机制几乎一致。因此ColdFusion能够用于程序员和图形设计者合作的工程中。程序员设计相当成熟的Web应用，用户通过基于标记的源代码寻找自己的路径，即便是不熟悉编程的用户也能够做到这一点。ColdFusion具有可扩展性，也就是说，如果用户遇到不能处理的任务时，可以采用其他的编程语言（Perl或C）编写相应的处理模块，然后将其无缝集成到ColdFusion中。

基于标记的这种特性也是ColdFusion最大的障碍。虽然标记为常用的程序设计工作提供了捷径，同时也造成了代码的不实用，除了原始的程序设计人员，其他的人很难进行故障检测。而且，与这里所提到的几种语言不同的是，ColdFusion服务器是一种应用，必须与已有的Web服务器一起购置和安装，许多Web主机服务支持ColdFusion，然而麻烦很多。

PHP: 超文本预处理器

PHP是一种开放资源脚本描述语言，而且发展变化非常快，每个人都能够为该语言的发展和成熟尽自己的力量。

在Web服务器的编程语言领域，PHP进入的比较晚。它的脚本语法非同一般，是Perl、C++和Java的混合体。虽然PHP最初是为小型站点而设计的，现在该语言的处理能力和速度都能够满足大型站点的需求。许多Web程序设计人员选择PHP，是因为有大量免费、开放源代码的入口和公告牌软件可用。

与PHP相比，JSP的优点是低层采用Java语言，Java语言是当今世界上最流行的程序设计语言。PHP的语言功能非常强大，但是由于很难找到开发人员和有关方面的技术支持，因此许多企业业主和公司一般不采用PHP。