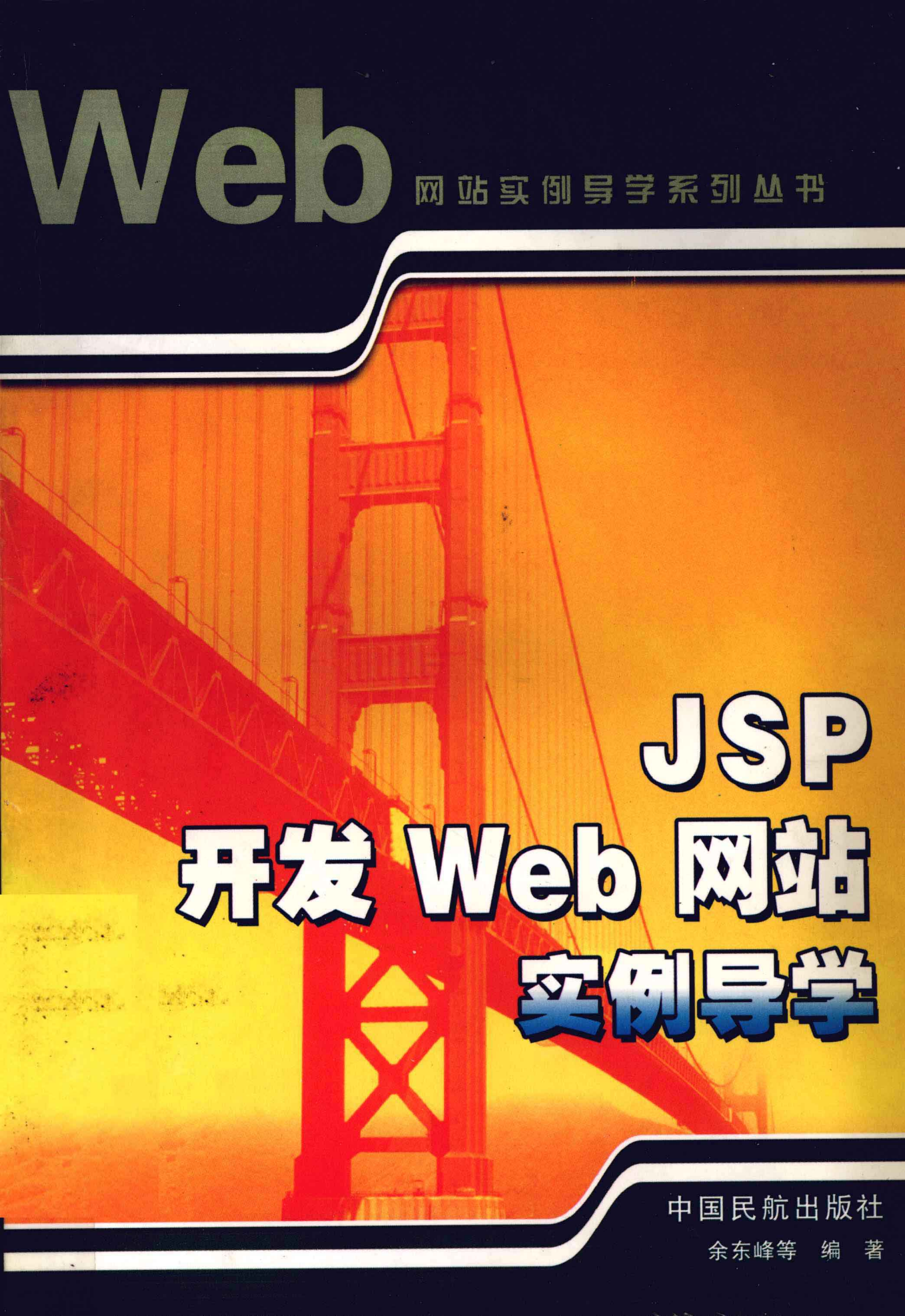


Web

网站实例导学系列丛书

The background of the book cover features a stylized, high-contrast image of the Golden Gate Bridge. The bridge's structure is rendered in a vibrant orange-red color, set against a gradient background that transitions from a deep blue at the top to a bright yellow at the bottom. The bridge's iconic towers and suspension cables are visible, creating a strong geometric pattern.

JSP 开发 Web 网站 实例导学

中国民航出版社

余东峰等 编 著

Web 网站实例导学系列丛书

JSP 开发 Web 网站实例导学

策划：余东峰 何学仪

主编：陆昌辉

编者：王 偲 方海燕

陆昌辉 蔡 勇

陈天霞

中国民航出版社

图书在版编目(CIP)数据

JSP 开发 Web 网站实例导学/陆昌辉等编著. —北京: 中国民航出版社, 2002.1

ISBN 7-80110-326-2

I. J…

II. 陆…

III. 计算机网络—基本知识

IV. TP3-49

中国版本图书馆 CIP 数据核字(2002)第 13777 号

JSP 开发 Web 网站实例导学

陆昌辉 等编著

出版	中国民航出版社
社址	北京市朝阳区光熙门北里甲 31 号楼 (100028)
发行	中国民航出版社
电话	64290477
印刷	中国电影出版社印刷厂
开本	787×1092 1/16
印张	14
字数	336 千字
版本	2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书号	ISBN 7-80110-326-2/G·136
定价	18.00 元

(如有印装错误, 本社负责调换)

前 言

在上个世纪，Internet 得到了迅速的发展，在此之中，HTML 起到了至关重要的作用。正是由于 HTML 的出现，网络才会变得如此绚丽多彩，然而 HTML 并不能完全满足人们的要求，原来的 HTML 只能产生一个静态的页面，而人们希望制作的页面能够产生丰富的动态变化，随着社会的网络化、信息化进程不断加速，Internet 正在飞速发展，动态页面技术已经成为了一种必然的趋势，在社会的需求之下，出现了各种动态页面制作技术，如 ASP、JSP、PHP 等。

现在的网站建设中，ASP、PHP 和 JSP 使用都很广泛。JSP 是架构在 Java Servlet 以及 SUN 公司灵活而强大的 Java 体系之上的，Java 语言“一次编写，到处使用！”的开发特点让程序员欢心鼓舞，JSP 技术也实现了跨平台和兼容各种 WEB 服务器产品，JSP 技术模型的定义借鉴了 ASP 的许多优点。使用 JSP 技术可以轻松地建立集安全与高效于一体的跨平台动态网站，所以业界都认为只有 JSP 才是未来发展的必然趋势。

本书虽然重点介绍的是 JSP 技术，但是同样介绍了很多的基础知识，如 HTML、数据库基础等等，只有打稳了根基才能做到进一步的升华。本书详细地介绍了如何使用 JSP 技术制作动态网页以及相关的 JavaBeans 技术，并且对流行的数据库技术也作了介绍。本书通过一系列的实例展示，教会你利用 JSP 技术开发出一个功能丰富的动态 WEB 站点。

本书内容深入浅出，可以作为普通 JSP 爱好者的入门级教材，对从事专门的网站建设的专业开发人员也有一定参考价值。

本书由余东峰、何学仪策划，陆昌辉主编，王偲、方海燕、陆昌辉、蔡勇、陈天霞等编写。由于时间紧促，作者水平有限，书中存在错误在所难免，恳请广大读者指正批评。

目 录

第一章 JSP 简介	
1.1 JSP 的起源	1
1.2 JSP 的优点	2
1.3 JSP 的运行	4
1.4 简单例子: HELLO WORLD!	7
1.5 思考与练习	8
第二章 Java 入门	
2.1 Java 语言简介	9
2.2 Java 语法	11
2.3 思考与练习	35
第三章 JSP 的语法	
3.1 JSP 的注释	36
3.2 HTML 中嵌入 JSP 脚本	38
3.3 脚本元素	39
3.4 指令	41
3.5 HTML 基础	50
3.6 思考与练习	63
第四章 JSP 的数据库	
4.1 数据库简介	4
4.2 SQL 语言	66
4.3 SQL Server 2000 简介	72
4.4 JDBC 简介	75
4.5 JDBC 编程	79
4.6 思考与练习	88
第五章 JavaServlet	
5.1 Java Servlet 技术简述	89
5.2 Java Servlet API	91
5.3 创建一个 Servlet 程序	94
5.4 调用 Servlet	97
5.5 Cookies	98
5.6 Session 基础	104
5.7 思考与练习	105
第六章 HTML 表单	
6.1 HTML 表单的基本结构	106
6.2 Servlet 与 HTTP 请求头	113
6.3 处理复杂的 Form 元件	118
6.4 思考与练习	120
第七章 JavaBeans 技术	
7.1 JavaBeans 技术简介	121
7.2 JavaBeans 在 JSP 中的应用	129

7.3 JavaBeans 的数据库应用实例.....	133
7.4 思考与练习.....	136
第八章 聊天室与 BBS	
8.1 功能与原理介绍.....	137
8.2 源代码分析.....	137
8.3 思考与练习.....	172
第九章 计数器	
9.1 源代码分析.....	173
9.2 思考与练习.....	192
第十章 留言本	
10.1 源代码分析.....	194
10.2 思考与练习.....	209
第十一章 发送电子邮件	
11.1 电子邮件和 SMTP 协议简介.....	210
11.2 主界面介绍.....	211
11.3 源代码分析.....	213
11.4 思考与练习.....	217
附录	
参考书目和 Internet 资源	218

第一章 JSP 简介

本章学习目标:

- ◆ 了解 JSP 的概念、历史、特点
- ◆ 了解 JSP 编译环境的基本概念以及安装

Java Server Pages(Java 服务器系统页面), 是由 Sun MicroSystem 公司倡导, 并在许多公司共同参与下建立的一种动态网页技术标准, 它在动态网页的创建中有强大的功能。Java Server Pages 为创建动态的 Web 应用提供了一个优良的开发环境。按照 Sun 的说法, Java Server Pages 能够适应市场上包括 Apache WebServer、IIS 在内的 85%以上的服务器产品。

Java Server Pages(以下简称为 JSP)是 Java 平台上用于编写包含诸如 HTML、XHTML 和 XML 等含有动态生成内容的 Web 页面的应用程序的技术。JSP 技术功能强大, 使用灵活, 为创建显示动态 Web 内容的页面提供了一个简捷而快速的方法。JSP 技术的设计目的是使构造基于 Web 的应用程序更加容易和快捷, 而这些应用程序能够与各种 Web 服务器、Web 应用服务器、浏览器和开发工具共同工作。JSP 技术就是在传统的网页 HTML 文件 (*.html, *.htm) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标记 (tag), 从而构成了 JSP 网页 (*.JSP)。服务器在接到访问 JSP 网页的请求后, 首先执行其中的程序段然后把结果以 HTML 格式返回给用户程序片段, 可以是操作数据库、重新定向网页、发送 E-mail 等等, 这些都是建立动态网站的基本要求。

实际上, JSP 也是 Java 语言, 只不过它是一个特别的 Java 语言, 是在 Java 语言中加入了一个特殊的引擎, 这个引擎将 HTTPServlet 这个类的一些对象自动进行初始化好让用户使用, 而用户不用再去操心前面的工作, 从而极大的节省编程人员写编码的时间。同时这个引擎又引入了一系列的特别语法, 用这些语法来进行一些特定的操作, 如引用文件、URL 转向、Java Bean 引用等这样的操作, 这些都是由引擎自动帮你完成初始化的工作。可以说 JSP 就是一个 Java Servlet 生成器。

1.1 JSP 的起源

要说 JSP 的起源, 就不得不从 Java 说起。Java 来自 Sun 公司在 1991 年启动的一个叫 Green Project(绿色计划)的项目, 这个项目最初的目的是为家用消费电子产品开发一个分布式操作软件, 这样我们可以把 E-mail 发给电冰箱、电视机或洗衣机等家用电器, 或对它们进行远程控制, 和它们进行信息交流。开始, 准备采用 C++, 但 C++太复杂, 安全性差, 最后基于 C++开发一种新的语言 Oak(Java 的前身), Oak 保留了 C++的语法, 但抛弃了明确的资源引用、指针算法和操作符过载等潜在的影响系统安全性的一些特性。Oak 是一种用于网络的精巧而安全的语言, 同时, 由于该项目要求具有可移植性, 即要求它具有与运行平台和环境无关性, 所以 Oak 语言具有平台无关性。Sun 公司曾用此优势去投标一个交互式电视项目, 但

结果被 SGI 打败。这时的 Oak 几乎就要淹没在技术的洪流之中，恰巧这时 Mark Ardreessen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员，他们用 Oak 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 Scott McNealy 的大力支持，从此 Java 就大步走向 Internet 领域。

而从 Oak 这个名字改为 Java，有一个有趣的故事，据说是因为开发小组的成员都喜欢喝咖啡，在研发期间饮用了大量的又香又浓的咖啡，并且一致认为 Java 所产的咖啡最好，于是，就选择了 Java 这个词作为名字，Java 就这样传开了。

Java 语言是一种软件编程语言，旨在为 Internet 发展而使网页从静态变为动态，并能用它写小应用程序。它具有能有效地阻碍网络病毒传播的功能，使用者无需花太多的时间去学习它；其最大的特点是与平台无关性，它突破了 CPU 构架和机器环境的限制，可做到“一次编译，到处运行(Writing Once , Running Everywhere)”。

当时，要使 Java 语言让广大的程序设计开发人员接受，还是不够成熟的。首先，Java 缺少主流系统的广泛支持，而且，在所有的平台和技术上都缺少可以直接参考的应用方式。这时的 Java 就在各个应用层面上积极进取，Java Servlet Pages 就是一大技术成就。Java 利用它的跨平台技术，使 Java 能应用于各个不同的平台之上，在各个平台之间能自由交流，真正做到了一次编写，到处运行。

Java 语言远比 Perl 和 PHP 语言更容易维护，但在 JSP 技术出现以前，它在 Web 页面设计能力方面比 Perl 和 PHP 差。但 JSP 技术出来以后，Java 语言的页面设计能力大大增强，这就极大的弥补了 Java 语言在网页设计方面的不足。

Sun 公司采用社团的方式来开发 JSP 技术，同其他公司合作开发和修订了 Java 的技术标准。并把 JSP 产品授权给和自己合作的公司，如：Macromedia 、Apache、Netscape。Sun 和许多其它公司和个人成立了一个强大的咨询和服务机构，以便为任何公司和个人提供信息服务。网址为 [http:// www.jalarta.apache.org](http://www.jalarta.apache.org)。

JSP 与 ASP 非常相似。ASP 由于是微软的产品，所以也就继承了微软的传统，只能用于 Windows NT/2000 平台，它是一个封闭的系统，其完善和发展只能靠微软一家公司的推动。而 JSP 是一个开放的体系，可以不加修改地运行在大部分 Web 服务器上，其中也包括 Windows NT 系统。

对开发人员来说，ASP 和 JSP 都能使网页设计者制作交互的、动态的 Web 网站。ASP 仅支持组件对象模型 COM，而 JSP 技术提供组件都基于 Java Beans 技术或 JSP 标签库。

当然，现在的科技日新月异，任何一种技术都无法一成不变，JSP 技术也在不断的更新优化，它一定会变得更加强大，更加优秀。要想了解有关 Java 与 JSP 的最新动态，请访问以下网址 <http://www.javasoft.com/>。

1.2 JSP 的优点

1. 与平台和服务器无关性

JSP最大的特色就是可以“Writing Once, Running Everywhere”(一次编写，到处运行)，这是因为JSP页面的内置脚本语言是基于Java编程语言的，而且所有的JSP页面都要被编译成Servlet，JSP页面就具有Java技术的所有优点，包括健壮的存储管理和安全性等。

JSP技术是与设计平台完全无关的，包括它的动态Web页面，它的Web服务器和底层的服务器组件。你可以在任何平台上编写JSP页面，在任何Web服务器或者Web应用服务器上运行，或者通过任何Web浏览器访问。你还可以在任何平台上建立服务器组件并且在任何服务器上运行它们，目

前主要是Java Beans和Servlet。有了这个优点，随着越来越多的供应商将JSP支持添加到他们的产品中，你就可以使用自己所选择的服务器和工具，更改工具或服务器并不会影响到当前的应用。

JSP技术依靠第一次写入，从此之后，就可以运行在任何具有符合JavaTM语法结构的环境中。而以前，程序依赖于单一平台或开发商，JSP技术能够运行在任何Web服务器上，并且支持来自多家开发商提供的各种各样的工具包。

2. 开放的开发过程，开放的源代码

SUN应用JAVA社团性过程开发JSP技术。自从1995年，SUN已经用这种开放过程方法同国际JAVA组织合作开发和修改了JAVA技术与规范。针对JSP的产品，SUN授权了工具提供商（如Macromedia）、结盟公司（如Apache, Netscape）、最终用户、协作商及其他。最近，SUN将最新版本的JSP和JavaTM Servlet（JSP 1.1, JAVA SERVLET 2.2）的源代码发放给Apache，以求JSP与Apache紧密的相互发展。

而由于Apache对源代码采用了OPEN TO PUBLIC的技术战略，所以，JSP平台的系统源代码是开放的，你甚至可以在Internet上免费得到它们，如果有必要，你还可以修改源代码，加入自己的特色或者是特有的功能。JSP将随着JAVA组织不断开放扩大继续完善。

3. 良好的可扩展性

尽管JSP和ASP一样，都是使用标签和脚本技术来制作动态Web网页，但JSP技术能使开发者扩展JSP标签。JSP开发者能定制标签库，所以，网页制作者能充分利用与XML兼容的标签技术，大大减少对脚本语言的依赖。由于定制标签技术，使网页制作者降低了制作网页和向多个网页扩充关键功能的复杂程度。

4. 出色的可重用性

JSP的开发人员在最初的开发过程开始就一直关注它的可重用性。JSP组件（企业Java BeansTM, Java Beans, 或定制的JSP标签）都是跨平台可重用的。企业Java Beans组件可以访问传统的数据库，并能以分布式系统模式工作于UNIX和WINDOWS平台。JSP技术的标签可扩充功能为开发人员提供简便的与XML兼容的接口。

绝大多数JSP页面都要利用可重用的、跨平台的Java Beans组件来执行应用程序中所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者客户团体所使用。这些组件有助于将网页的设计与逻辑程序的编写分开，节约了开发时间，同时充分发挥了Java和其他脚本语言的跨平台的能力和灵活性。基于组件的方法加速了总体开发过程，并且使得各种组织在他们现有的技能和优化结果的开发中得到平衡。

这种基于组件的模式提高了应用程序的开发效率，因为这种模式能够使开发人员利用快捷的子组件快速创建模板应用程序，然后再整合一些附加功能以后便可使用。

5. 应用程序内容与页面显示分离

使用JSP技术，Web页面开发人员可以使用HTML或者XML标记来设计和格式化最终页面，使用JSP标记或者Script来生成页面上的根据请求页变化的动态内容（例如请示用户信息或者特定的一种商品的价格）。生成内容的逻辑被封装在标记和Java Beans组件中，并且捆绑在Scriptlet中，所有的脚本程序在服务器端运行。这样Web管理人员和页面设计者，能够编辑和使用JSP页面，而不影响内容的生成。

6. 采用标记简化页面的开发

Web页面开发人员不一定是熟悉脚本语言的编程人员。JSP技术封装了许多功能，这些功能是在易用的与JSP相关的XML标记进行动态内容生成时所必需的。标准的JSP标记能够访问和实例化Java Bean组件，设置或者检索组件属性，下载Applet，以及执行用其他方法难于编码和耗时的功能。

通过开发定制化的标记库，JSP技术是可以扩展的。今后，第三方开发人员和其他人员可以为常用功能创建自己的标记库。这使得Web页面开发人员能够使用熟悉的工具和如同标记一样的执行特定功能的组件来工作。

7. 强大的可伸缩性

从只有一个小的jar文件就可以运行Servlet/JSP到由多台服务器进行集群和负载均衡，到多台Application进行事务处理、消息处理；从一台服务器到无数台服务器，Java都能找到他自己的位置，显示了它的广泛的应用基础和强大的生存能力。

1.3 JSP 的运行

1.3.1 JSP 的应用平台

JSP是基于Java Servlet的技术，是Servlet 2.1 API的扩展。因此支持Servlet的新版本平台都应该支持JSP。这样的平台有不少，而且各有特色。

1. SUN公司的JSWDK平台

JSWDK是由Sun公司自己提供的平台，比较简捷，它只包含一些基本的内容，初学JSP的朋友可用此平台，连接本机的数据库（比如使用Microsoft Access数据库）来与JSP进行第一次亲密的接触；但JSWDK毕竟只是一个简化的服务器平台，性能和稳定程度都比较有限。

2. Apache公司的Tomcat平台

Tomcat平台是Sun公司借助Apache的影响来开发服务器端的Java技术的战略平台，很可能成为下一代Java Web Server的主流。因此，可以相信Tomcat是一个较理想的JSP Servlet开发和支撑平台。

Tomcat 作为Servlet 2.2和JSP 1.1的官方参考平台，既可以单独作为小型的Servlet、JSP测试服务器，也可以集成到Apache Web服务器中。

Tomcat也是免费的，对用户开放源代码，但它配置起来比较麻烦。

3. Caucho公司的Resin平台

Resin平台是由Caucho公司公布的JSP平台，据<http://www.caucho.com/>自己提供的测试结果，Resin平台是迄今为止最快的商业JSP平台，Resin平台是Tomcat平台运行速度的3倍。它采用一种行径的语言——XSL(XML Stkylesheet Language)，可以使形式和内容相分离。Resin的速度和稳定性都不错，能满足商业上的应用要求，而且还是免费的，只要你能上网，你就可以自由下载它。

Resin1.3支持下列特性：

- (1)自动编译Java Servlet和Java Bean。
- (2)支持IIS、Apache、Netscape和其他内置HTTP1.1的Webfwq。
- (3)支持JSP1.1和在服务器端编译JavaScript。
- (4)使用Servlet2.2引擎。
- (5)比mod-Perl和mod-PHP要快，比Tomcat也要快得多。

由于技术一直在不断的更新和发展，许多的新技术也许会在“一夜之间”冒出来，因此，要想全面了解JSP的运行平台，你可以时不时去以下站点看一看，也许总能看到一些新的版本或平台。商业软件支持JSP的平台可在以下站点中查看到：<http://www.interpasnet.com/JSS/textes/JSP2.htm>。

1.3.2 JSWDK 平台

目前发布的最新的 JSWDK 可下载的版本是以可安装的形式压缩的。下载来的文件大约是 20MB，可提供完整的 Java 开发环境，让你建立以标准 API 为核心的 Java 解决方案。你的网络服务器需要应用到 JSP 的惟一一件事是 Java 编译器。它必须要让网络服务器知道编译器的位置，将环境变量 JAVA.HOME 设到 JSWDK 的安装目录下。如果你是在 Windows 上安装并且接受预设目录，将这行程序代码 `set JAVA.HOME=C:\1.2.2` 加到你的 `autoexec.bat` 文件并且要重新开机。

1. 安装JDK

首先要安装JDK，并且还需要修改系统环境参数。

JDK (Java Developers Kit) 是Java语言编写的集成开发工具，它包含Java语言的编译器和Bytecode的解释器。它用于编译和执行用Java编写的程序，因为JSP是以Java作为开发语言的，所以，在安装JSWDK之前必须先安装JDK。

而JSWDK (JavaServer Web Development Kit) 则是用来开发JSP和Java Servlet API应用程序的软件开发包。

JDK 可以是 Sun JDK1.2.2 或 1.3。它们都可以从网址 <http://java.sun.com/products/jdk/1.1/> 下载得到。

如果你是在Windows 98上安装，将这行`set JAVA_HOME=[x:\jdk1.2.2]`加到`autoexec.bat`文件中；如果是在Windows 2000，可先打开控制面板，然后打开“系统”窗口，选“高级”选项，再选环境设置。

选择并编辑path参数，加入`[x:\jdk1.2.2\bin]`，如图1-1所示。

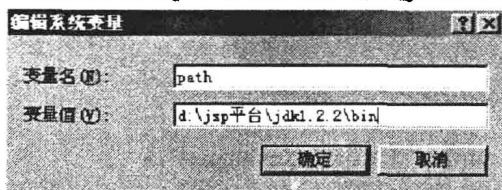


图1-1 设置JDK的路径图

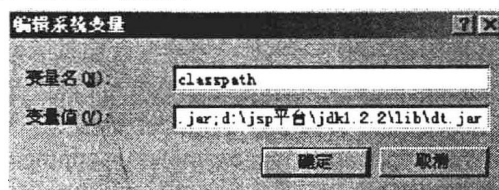


图1-2 配置JDK环境变量

然后单击“新建”按钮增加classpath变量，在变量名后输入classpath，在变量值后输入`[x:\jdk1.2.2\lib\tools.jar;x:\jdk1.2.2\lib\bt.jar]`即可，如图1-2所示。

2. 安装JSWDK

JSWDK可以从<http://javasoft.com/products/jsp>中下载。下载完后，你只要将`jswdk1_0_1-win.zip`带目录解压到硬盘根目录下就可以了，然后就在硬盘上找`jswdk-1.0.1`目录。

安装JSWDK之后，只需执行`startserver.bat`命令即可启动服务器。其界面如图1-3所示。

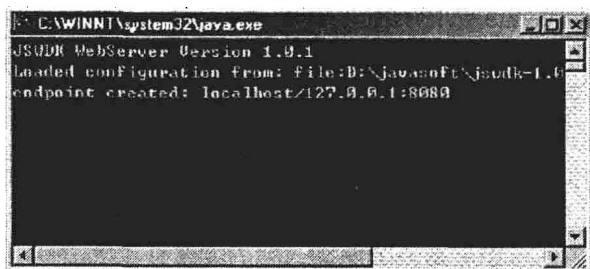


图1-3 JSWDK的启动画面

如果要退出，可执行`stopserver.bat`。顺便说一下，如果是UNIX系统下，则要分别执行`%startserver`和`%stopserver`。

这时，只要在浏览器地址栏中键入`http://localhost:8080`或者是`http://127.0.0.1:8080`，就能看到JSWDK的欢迎界面图，如图1-4所示。

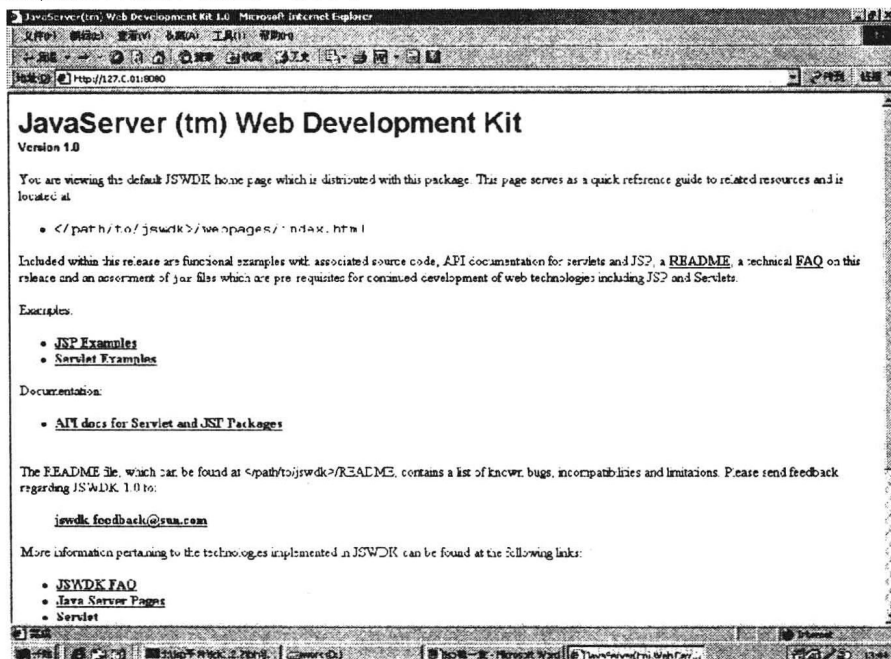


图1-4 JSWDK的欢迎界面

如果发生问题，很有可能是JDK的环境参数没有正确设置，你可以再次检查。如果是8080端口已经被征用，可以打开JSWDK目录下的`webserver.xml`对此端口重新进行设置。

具体设置如下：

```
<WebServer id="webServer">
<Service id="service0" port="8080">
<WebApplication id="examples" mapping="/examples" docBase="examples"/>
</Service>
</WebServer>
```

将port口设为其它端口即可。

完全安装好以后，打开 JSP 页面的方式为：`http://localhost:8080/examples/xxx.Jsp`，当然，你可以建立自己的执行目录，只要在 `webserver.xml` 文件中增加这样的设置：

```
<WebApplication id="myweb" mapping="/myweb" docBase="myweb"/>
```

就可以了。

3. 安装JRE

JRE (Java Run Enviromment) 是Java的运行环境，它可以从`http://java.sun.com/products/jdk/1.1/jre/`中下载。

下载完并解压缩之后，按提示一步步安装就可以了。

在运行 JSP 示例页面之前，可以注意一下安装 JSWDK 的目录，特别是“work”子目录下的东西。执行示例页面时，可以在这里看到 JSP 页面如何被转换成 Java 源文件，然后又被编译成 class 文件（即 Servlet）。

JSWDK 软件包中的示例页面分为两类，它们或者是 JSP 文件，或者是包含一个表单的 HTML 文件，这些表单均由 JSP 代码处理。与 ASP 一样，JSP 中的 Java 代码均在服务器端执行。因此，在浏览器中使用“查看源文件”菜单是无法看到 JSP 源代码的，只能看到结果 HTML 代码。所有示例的源代码均通过一个单独的“examples”页面提供。

1.3.3 Tomcat 平台

Tomcat 可以从网上免费得到，其下载地址为 <http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.2.3/bin/>。国内也有很多网站提供免费下载服务。下载完，解压到一个目录之下即可。

下面以在 Windows 平台下的安装为例说明安装过程：

在桌面上用右键单击“我的电脑”，在快捷菜单中单击“属性”，在打开的“系统特性”对话框中选“高级”选项卡。单击“环境变量”按钮，弹出“环境变量”对话框。在“系统变量”中单击“新建”按钮，在“新建系统变量”对话框的变量名中输入“JAVA_HOME”，在变量值中输入“[x:\]JDK1.2.2”，然后确定；再新建一个系统环境变量、变量名为“TOMCAT_HOME”，变量值为“[x:\]tomcat”。

至此，配置就全部完成了，剩下的事就是要重新启动计算机了，这样，环境变量才会有效。

下面测试一下安装配置是否正确：

启动 tomcat：

在 d:\tomcat\bin 下有一个 startup.bat，运行它，如果出现一个 DOS 窗口。在浏览器中输入：<http://localhost:8080/>，应该可以看到一只可爱的小猫。在这个页面有 JSP Examples 和 Servlet Examples，可以自己试着去运行它们。

1.3.4 Resin 平台

Resin 可以从 <http://www.caucho.com/download/index.xtp> 下载。Resin 的压缩文件大约 4M，下载完后，解压到一个目录下。接着，打开 Apache 的 conf 子目录下的 httpd.conf 文件，用写字板或其它文本编辑工具编辑它，把如下几行加入进去：

```
LoadModule caucho_module "E:\resin114\sruntime\win32-1.3.6\mod_caucho.dll"
AddModule mod_caucho.c
<IfModule mod_caucho.c>
CauchoConfigFile "E:\resin114\conf\resin.conf"
<Location/caucho-status>
SetHandler caucho-status
</Location>
</IfModule>
```

然后，在 Resin 的 conf 子目录下找到 resin.conf 这个文件，将该文件做一个备份。现建立一个 resin.conf 文件，包括如下内容：

```
<caucho.com>
<http-server app-dir='e:/apache/htdocs'>
<error-log id='log/error.log'/>
<classpath id='WEB-INF/classes' source='WEB-INF/classes' compile='true'/>
<servlet-mapping url-pattern='/servlets/*' servlet-name='invoker'/>
<servlet-mapping url-pattern='*.xtp' servlet-name='com.caucho.jsp.XtpServlet'/>
<servlet-mapping url-pattern='*.jsp' servlet-name='com.caucho.jsp.JspServlet'/>
</http-server>
</caucho.com>
```

至此，你的 Resin 就算是安装好了。

1.4 简单例子：HELLO WORLD!

JSP 网页能结合 JavaBeans 技术来扩充网页程序的功能，所谓 JavaBeans 技术其实就是一种

Java类，它通过封装属性和方法成为一个具有某种功能的对象。JavaBeans被组织成Package来进行管理，把一组JavaBeans放在一个目录中，每个类的定义之前加上Package名。目录放在系统环境Classpath所包含的目录下。如在JSDK中，默认的目录为：\jswdk\1.0.1\webpage\Web-INF\jsp\beans\。

下面是一个简单的JavaBeans构架。用文本编辑器写一个简单的也是一个著名的java程序：“Hello World!”。

打开记事本，编辑一个包含以下内容的文件，以helloWorl.java为名字保存在\jswdk-1.0.1\Web-INF\jsp\beans\test目录下。

内容如下：

```
<html>
<head>
<title>jsp简单实例一</title>
<meta http-equiv="content-Type" content="text/html; charset=gb312">
</head>
<body bgcolor="#FFFFFF">
<%@ page language="java"%>
<%!String mstr;%>
我的第一个JSP程序
<br>
<%
mstr="HELLO WORLD!";
for(int i=1;i<10;i++){
out.print(mstr);
out.print("<br>");
}
%>
</body>
</html>
```

然后，打开一个浏览器，在地址栏键入 `http://localhost:8080/HelloWorld.jsp`。好了，接下来的事就是好好欣赏你的第一个JSP程序的结果了（如图 1-5 所示）。



图 1-5 第一个例子的运行结果

1.5 思考与练习

1. 简述 JSP 与 Java 的关系。
2. JSP 技术有什么优点？

第二章 Java 入门

本章学习目标：

- ◆ 掌握 Java 语言的语法
- ◆ 能用 Java 进行简单程序设计
- ◆ 理解面向对象程序设计思想

2.1 Java 语言简介

Java是一种面向对象的，分布式的，解释型的，具有良好健壮性和安全性的，可移植性强的，多线程动态语言。

Java是新一代完全按面向对象思想设计的程序设计语言，由于它的出色的与平台无关性，所以特别适合于Internet应用程序开发，因为连在Internet上的计算机有千千万万，而且有的是大型机，有的是工作站，更大的则是个人计算机，它们的CPU构架有很大的差别，而且，它们有的基于Windows平台，有的基于UNIX平台。这个客观事实要求有一种与平台无关的编程语言，而Java几乎就是为此定制的。因此，Java能广受人们的欢迎，也就不足为奇了。

那么，神奇的Java语言有何过人的本领呢？

1. 简单

Java最初是为对家用电器进行集成控制而设计的一种语言，而家用电器不可能有太复杂的结构，因此用于控制它的系统也必须简单明了。Java语言的简单性主要体现在以下方面：首先，Java的风格类似于C++，它保留了C++的语法，但抛弃了明确的资源引用、指针算法和操作符过载等潜在的影响系统安全性的一些特性。因而C++程序员应该是非常熟悉的。也许，可以这样说：Java语言是C及C++语言的一个变种，因此，学过C/C++的人可以很快就转入到Java中。其次，Java提供了丰富的类库，方便了广大开发人员。

2. 分布式

Java包括一个支持HTTP和FTP的基于TCP/IP协议的子库。因此，Java应用程序可凭借URL打开并访问网络上的对象，其访问方式与访问本地文件系统几乎毫无差别。为分布环境尤其是Internet提供动态内容无疑是一项比较艰巨的任务，但Java的语法特性恰好适合实现这个目标的条件。

3. 面向对象化

面向对象可以说是Java最重要的特性。Java语言的设计完全是面向对象的，它不支持类似C语言那样的面向过程的程序设计技术。Java支持静态和动态风格的代码继承及重用。单从面向对象的特性来看，Java类似于SmallTalk，但其它特性、尤其是适用于分布式计算环境的特性远远超越了SmallTalk。

4. 与平台无关性

为了更好地为网络应用服务, Java将它的程序编译成一种固定格式的中间文件。以便所有Java运行系统的机器都能执行这种中间代码。现在, Java运行系统有Solaris2.4(SPARC)、Win32系统(Windows95和WindowsNT)等。Java源程序首先被编译成一种与机器无关的byte-code格式语言, 然后, 在虚拟机上运行这种语言, 最后, 由与机器相关的运行调试器实现执行, 得出结果。

为了适应不同的操作系统、CPU结构, Java中的数据类型有byte8-bit二进制补码、short16-bit二进制补码、int32-bit二进制补码、long64-bit二进制补码、float32-bitIEEE754定点数、double32-bitIEEE754定点数、char16-bitUnicode字符。

在任何Java解释器中, 数据类型都是依据以上标准具体实现。因为目前几乎所有的CPU都能支持以上数据类型、8~64位整数格式的补码运算和单/双精度浮点运算。Java编译器本身就是用Java语言编写的。Java运算系统的编制用C语言写成。

5. 自主检查错误

Java自主地检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。程序不能自主控制内存, 内存控制大权由Java自己掌握, 这样就大大减少了内存出错的可能性。这些功能特征大大缩短了开发Java应用程序的周期, 增加了安全性。

6. 解释的语言

Java的运行系统能直接运行目标代码指令。因为链接程序通常比编译程序所需资源少, 所以程序员可以在创建源程序上花上更多的时间。

7. 类装载模式

ClassLoader通过将本机类与网络资源类的名称分开, 来保持安全性。而且总是检查调入的, 这样可以有效地避免所谓特洛伊木马现象的出现。

8. 动态性

Java的动态特性是其面向对象设计方法的体现, 它允许程序动态地装入运行过程中所需要的类, 这是C++语言进行面向对象程序设计所无法实现的。在C++程序设计过程中, 每当在类中增加一个实例变量或一种成员函数后, 引用该类的所有子类都必须重新编译, 否则将导致程序崩溃。Java则不同, 它的编译器不是对实例变量和成员函数的引用编译为数值引用, 而是将符号引用信息在字节码中保存下来传递给解释器, 再由解释器在完成动态连接类后, 将符号引用信息转换为数值偏移量。这样, 一个在存储器生成的对象不在编译过程中确定, 而是要等到运行时才由解释器确定的。这样, 对类中的变量和方法进行更新时就对现存的代码没有影响。解释执行字节码时, 这种符号信息的查找和转换过程仅在一个新的名字出现第一次时才进行, 以后代码便全速执行。如果程序连接了网络中另一系统中的某一类, 该类的所有者也可以自由地对该类进行更新, 而不会影响任何引用该类的程序。Java还简化了使用一个升级的或全新的协议的方法。如果你的系统运行Java程序时遇到了不知怎样处理的程序, Java能自动下载你所需要的功能程序。

9. 多线程

Java提供的多线程功能使得在一个程序里可同时执行多个分任务。线程——有时也叫小进程。因为Java采用多线程技术, 所以比C和C++更健壮。多线程带来的好处还不能实现更好的交互性能和实时控制性能。打个浅显的比喻吧, 比如你要烧开水泡茶, 你不需要一直等到水烧开后才去取茶杯, 而是一边烧开水, 一边把茶叶之类的东西都预备好。这正如在Java里, 你可用一个单线程来调一幅图片, 同时也可以访问HTML里的其它信息。

2.2 Java 语法

2.2.1 数据类型

Java 数据类型分为简单类型和复合类型。

简单类型包括逻辑型 (logical)、字符类型 (textual)、整型 (integer) 和浮点型 (floating)。

复合类型包括类 (class)、接口 (interface) 和数组 (array)。

1. 逻辑类型

逻辑类型数据的值只有两个：true 和 false。定义的方法如下：

```
Boolean a = true
```

```
Boolean b = false
```

2. 字符类型

字符型变量：是用单引号引起来的一个字符，如：

```
'a' -
```

```
'n'
```

3. 整型类型

整型变量包括四种类型：

byte：用 8 位二进制表示；

short：用 16 位二进制表示；

int：用 32 位二进制表示；

long：用 64 位二进制表示；

整型常量包括以下三种：

十进制整数：

如 34, -36, 0。

八进制整数：

它是以 0 开头的数。如：035，它表示十进制数 29。-021，它表示十进制数-17。

十六进制整数：

它以 0x 或 0X 开头。如：0x11 表示十进制数 17，-0x24 表示十进制数-37。

4. 浮点类型

浮点常量分为两种：

十进制表示法。由整数和小数点组成，如：0.556, -14.0, 24.0。

科学计数法。如：12e2, -55E3。E 或 e 必须是整数。

浮点变量包括 Float 型和 Double 型。Float 类型占 32 位，double 类型占 64 位。

2.2.2 数组

数组是一有序的数据集合，chars[]是字符型数组；int[]是整型数组；数组中的元素可以是数组，这样就可以定义二维甚至多维数组，如：intblock[][]=newint[2][3]。

在 Java 里，数组实际上是一个对象，数组有一个成员变量：length。这个成员函数用来查看任意数组的长度。如：inta[][]=newint[20][13]a.length/*20*/a[0].length/*13*/。

1. 一维数组

(1)定义格式

```
type arrayname[];
```