



附光盘
CD-ROM



JSP

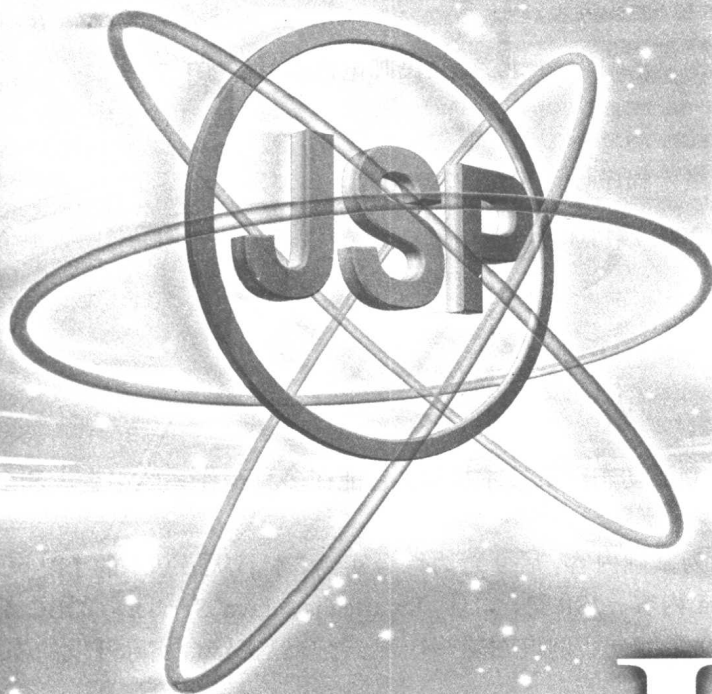
编程基础及 应用实例集锦

■ 柏亚军 等 编著

人民邮电出版社
www.pptph.com.cn

TP393.092

75D



JSP

编程基础及 应用实例集锦

■ 柏亚军 等 编著

北方工业大学图书馆



00493378

人民邮电出版社

J5815/10

图书在版编目(CIP)数据

JSP 编程基础及应用实例集锦/柏亚军编著. —北京:人民邮电出版社,2001.7

ISBN 7-115-09415-2

I .J... II .杨... III .计算机网络—程序设计 IV .TP393.092

中国版本图书馆 CIP 数据核字(2001)第 036946 号

内 容 提 要

JSP 是由 SUN 公司倡导建立的动态网页技术标准,它能在多种操作系统平台上和多种 Web 服务器上使用。

本书共分为 15 章,前 6 章为基础内容,后 9 章为综合实例。本书从最基本的 JSP 概述、JSP 运行模式和环境设置、Java 内容、JSP 语法,到 JavaBeans、Java Servlet、JDBC 都作了详尽的介绍,同时还结合了 9 个综合实例巩固这些知识。全书紧密结合实际开发的需要,使读者能够尽快上手。

本书适合于对开发 Web 有兴趣的读者。Web 开发的新手,可以从头开始学习本书;对 Web 编程有一定经验的读者,可以参阅书中给出的编程实例。

JSP 编程基础及应用实例集锦

◆ 编 著 柏亚军 等

责任编辑 张立科

◆ 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@pptph.com.cn

网址 <http://www.pptph.com.cn>

读者热线:010-67129212 010-67129211(传真)

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:21.75

字数:526 千字

2001 年 7 月第 1 版

印数:1-6 000 册

2001 年 7 月北京第 1 次印刷

ISBN 7-115-09415-2/TP·2302

定价:38.00 元(附光盘)

编者的话

JSP (Java Server Pages) 是由 SUN 公司倡导、许多公司参与建立的一种动态网页技术标准。它通过在传统的 HTML、XML 网页中加入 Java 程序片段和 JSP 标记构成。Web 服务器遇到访问 JSP 网页的请求时, 首先执行其中的程序片断, 然后将执行的结果以 HTML 格式返回给客户。程序片断具有可操作数据库、可重新定向网页、可发送电子邮件等等动态网站所需要的功能。

本书共 15 章, 1~6 章主要介绍了 JSP 编程基础, 包括 JSP 简介、JSP 的基本概念、JSP 语法、JavaBeans、Java Servlet、JDBC 等内容; 7~15 章介绍了一些流行的 JSP 应用实例, 包括网上投票、网上测试、购物车的实现、查看和接收 E-mail、用户注册和登录、网上书店、聊天室、BBS 以及服务器端生成图片等。

参与本书编写的有柏亚军、马磊、宋现锋、张宏林、刘翼、苏玉梅、刘文平、任国利、黄迎春、胡建飞、于增强、龚斌、刘继华、张勇、夏勇、许美林等。

由于时间仓促, 书中难免会有疏漏和错误, 欢迎读者提出宝贵的意见和建议。

配套光盘内容简介

本书配套光盘中包括以下内容。

JSP 文件

主要提供书中讲述的各个实例的源文件。可以通过查看 index.txt 文件, 获取每个文件的简单说明和相应源代码文件名。

JSP 辅助软件

主要提供几个 JSP 编辑软件。可以通过查看 index.txt 文件, 获取每个软件的简单说明和相应源代码文件名。

JSP 源码

主要提供几个利用 JSP 开发的 Web 应用系统。可以通过查看 index.txt 文件, 获取每个文件的简单说明和相应源代码文件名。

编者

2001 年 5 月

目 录

第 1 章 JSP 简介	1
1.1 JSP 技术介绍	1
1.1.1 Web 编程技术	1
1.1.2 JSP 概述	1
1.1.3 JSP 的形式	2
1.1.4 JSP 的特点	3
1.2 JSP 与其他动态网页技术比较	4
1.2.1 JSP 与 CGI 比较	4
1.2.2 JSP 与 ASP 的比较	4
1.2.3 JSP 与 Servlet 的比较	6
1.3 JSP 的运行环境	6
1.3.1 JSP 运行环境需要的组件	6
1.3.2 JSWDK 的安装和配置	7
1.3.3 Tomcat 的安装和配置	8
1.4 HelloWorld	8
1.5 小结	9
第 2 章 JSP 的基本概念	10
2.1 对象	10
2.1.1 对象域	10
2.1.2 对象和变量	11
2.1.3 隐含对象	11
2.1.4 PageContext 对象	12
2.1.5 对象的应用	12
2.2 Java 简介	14
2.2.1 类型	14
2.2.2 Java 的运算符	16
2.2.3 Java 控制流	20
2.2.4 小结	22
第 3 章 JSP 语法	25
3.1 JSP 的语法规则	25
3.1.1 开始和结束的标记	25
3.1.2 HTML 注释	25
3.1.3 隐藏注释	26

3.1.4	声明	27
3.1.5	表达式	28
3.1.6	脚本 scriptlet	28
3.2	JSP 语法	29
3.2.1	指令类语法	29
3.2.2	动作语法	34
3.2.3	脚本语法	43
3.3	小结	44
第 4 章	JavaBeans	45
4.1	JavaBeans 的属性	45
4.1.1	Simple 属性	45
4.1.2	Indexed 属性	46
4.1.3	Bound 属性	47
4.1.4	Constrained 属性	48
4.2	JavaBeans 的事件	50
4.2.1	概述	51
4.2.2	事件状态对象 (Event State Object)	52
4.2.3	事件监听接口 (EventListener Interface) 与事件监听者 (EventListener)	52
4.2.4	事件监听者的注册与注销	53
4.2.5	适配类	54
4.3	用户化	55
4.3.1	用户化器接口 (Customizer Interface)	56
4.3.2	属性编辑器接口 (PropertyEditor Interface)	56
4.3.3	BeanInfo 接口	57
4.4	持久化	57
4.5	JavaBeans 的存储格式	58
4.6	小结	58
第 5 章	Java Servlet	61
5.1	Java Servlet 概述	61
5.1.1	关于 Servlets	61
5.1.2	Servlet 应用范围	61
5.1.3	Servlet 结构总视	62
5.1.4	Servlet Lifecycle	62
5.2	编写 Servlet	63
5.2.1	与客户端的交互性	63
5.2.2	一个 HTTP Servlet 处理 GET 和 HEAD 方法的例子	64
5.2.3	一个 HTTP Servlet 处理 POST 方式的例子	64
5.3	Lifecycle 方法	67

5.3.1	重编 Init 初始化方法	67
5.3.2	重编 Destroy 方法	68
5.3.3	关于一个 Servlet 中断涉及的多线程	68
5.3.4	提供关于 Servlet 的信息	70
5.4	运行 Servlet	71
5.4.1	属性	71
5.4.2	使用 Servlet Runner	72
5.5	小结	74
第 6 章	JDBC	85
6.1	JDBC 简介	85
6.1.1	什么是 JDBC	85
6.1.2	JDBC 产品	89
6.2	连接	90
6.2.1	概述	90
6.2.2	打开连接	90
6.2.3	JDBC URL	91
6.2.4	“odbc” 子协议	92
6.2.5	注册子协议	92
6.2.6	发送 SQL 语句	92
6.2.7	事务	93
6.2.8	事务隔离级别	93
6.3	DriverManager	94
6.3.1	概述	94
6.3.2	跟踪可用驱动程序	94
6.3.3	建立连接	95
6.4	小结	96
第 7 章	应用实例一——网上投票	100
7.1	概述	100
7.2	投票表单的设计	100
7.3	投票处理过程	101
7.4	源程序展示	103
7.5	小结	108
第 8 章	应用实例二——网上测试	109
8.1	设计概述	109
8.2	具体实现	109
8.2.1	试题	109
8.2.2	调用 JavaBeans	110
8.2.3	答题页面	110

8.2.4 答题结果处理	112
8.3 实例程序展示	113
8.3.1 exam.jsp	113
8.3.2 exam.java	118
第 9 章 应用实例三——购物车	124
9.1 概述	124
9.2 设计目标	125
9.2.1 购物车的实现	125
9.2.2 商品列表主界面	128
9.2.3 清除购物车	131
9.3 所有源程序	135
9.3.1 shop.jsp	135
9.3.2 shopping.java	145
9.4 小结	149
第 10 章 应用实例四——查看和接收 E-mail	150
10.1 设计目标	150
10.2 Pop3 协议	150
10.2.1 Pop3 服务器的认证	152
10.2.2 获取总邮件数	152
10.2.3 获取某个邮件的内容	152
10.2.4 Pop3 服务器返回的内容	153
10.3 利用 Pop3 协议实现收取信件的底层方法	153
10.3.1 连接服务器	154
10.3.2 向服务器发送命令	154
10.3.3 取得服务器发来的回应	155
10.3.4 Pop3 认证	155
10.3.5 取得邮箱的状态	156
10.3.6 取得邮件的内容	157
10.3.7 往 Pop3 邮件服务器发送空指令	159
10.4 完整的代码	160
10.4.1 利用 Pop3 实现收取信件的底层模块	160
10.4.2 编写一个线程, 用来保持与 Pop3 服务器之间的连接	168
10.4.3 编写一个 bean, 用来封装收发 E-mail 的一些方法	169
10.4.4 登录页面	174
10.4.5 编写显示 E-mail 列表的 JSP 页面	176
10.4.6 编写显示 E-mail 内容的 JSP 页面	180
10.5 小结	182

第 11 章 应用实例五——用户注册登录	183
11.1 用 SQL SERVER 建立数据库	183
11.1.1 设计表的结构	183
11.1.2 建立数据源	184
11.1.3 建立连接数据库的 JavaBeans	186
11.2 设计具体的 JSP 页面	188
11.3 小结	206
第 12 章 应用实例六——我的书店	207
12.1 用 SQL SERVER 建立数据库	207
12.2 设计具体的 JSP 页面	209
12.3 小结	233
第 13 章 应用实例七——聊天室	234
13.1 功能实现和技术要点	234
13.1.1 使用 Java 的集合类	234
13.1.2 在线用户列表	234
13.1.3 窃窃私语	236
13.2 代码和分析	237
13.3 小结	251
第 14 章 应用实例八——BBS	252
14.1 用 SQL SERVER 建立数据库	252
14.1.1 设计表的结构	252
14.1.2 构造表	253
14.2 用 JSP 具体实现以及代码分析	254
第 15 章 应用实例九——服务器端生成图片	306
15.1 设计目标	306
15.2 PngEncoder 类的实现	306
15.2.1 构造方法	307
15.2.2 转换生成 PNG 文件格式的字节流	307
15.2.3 写 PNG 格式的文件头	308
15.2.4 写 PNG 格式的文件主体	309
15.2.5 写 PNG 格式的文件结尾	312
15.2.6 几个往 Byte 数组里写入 int 或者 String 的方法	313
15.3 createImg 类的实现	314
15.3.1 Init 方法	314
15.3.2 drawBall 方法	315
15.3.3 encode()方法	316
15.3.4 getImage 方法	317

15.4	页面的自动刷新	318
15.5	完整的代码	318
15.5.1	由 Image 对象生成 Png 格式的字节数组	318
15.5.2	画图并且生成图片的程序	330
15.5.3	一个用作图片的 JSP	335
15.5.4	主页面	335

第 1 章 JSP 简介

本章将对 JSP 进行简单的介绍,通过与其他工具的比较,使读者认识什么是 JSP、JSP 运行的优点、JSP 具备的功能以及它的运行方式。同时本章还将介绍 JSP 运行环境的配置和需要的组件。最后给出一个 HelloWorld 的程序实例。

1.1 JSP 技术介绍

1.1.1 Web 编程技术

最初 Web 应用程序主要是用文件存储为手段以文件导航的方式提供信息资源,后来随着 NCSA HTTP Web 服务器中的 Common Gateway Interface (CGI, 通用网关接口) 的出现,这一切发生了巨大的变化。

CGI 引入了“动态内容”的思想,能够根据需要随时生成 Web 页的等价物。CGI 脚本可以搜索数据,可以通过简单的鼠标点击来实现。毫无疑问,能够发布动态内容已经成为导致 Web 增长的一个主要因素。另外,值得注意的是,实现 CGI 并不需要修改客户机/服务器协议——Web 浏览器请求 CGI 脚本,脚本所使用的语言与浏览器请求文档使用的是“相同”的语言,浏览器甚至不关心所返回的文档是来自一个文件还是针对一个请求而专门创建的。

为了占领 Internet 这个诱人的市场,IBM 公司的 Visual Age for Java 和 Net.data, Sybase 在 PowerBuilder 6.0 的版本中加入了 Web 特性, Netscape 公司也推出了自己的 Web Server 及完整的 Web 开发工具等。在这种情况下, Sun 公司倡导、许多公司共同参与建立了新的动态网页技术标准——Java Server Pages。

1.1.2 JSP 概述

JSP 是 Java Server Pages 的缩写,它是一种动态网页技术标准。在传统的 HTML 网页文件 (*.html, *.htm) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标记,就构成了 JSP 网页 (*.jsp)。Web 服务器在遇到访问 JSP 网页的请求时,首先执行其中的程序片段,然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、可以重新定向网页、可以发送和接受电子邮件等动态网页所需要的功能。所有的程序都在服务器端执行,在通过网络传送到客户端的仅是程序执行的结果,所以对客户端浏览器的要求较低。

JSP 技术功能强大、使用灵活,为创建显示动态内容的 Web 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编写基于 Web 的应用程序更加容易和快捷的同时，保证这些应用程序能够与各种 Web 服务器、浏览器和开发工具共同工作。

1.1.3 JSP 的形式

一个 JSP 页面从外表上看与 HTML 和 XML 并没有根本的区别，只是多了一些使用 Java 语言编写的、JSP 引擎能够处理和抽取的额外的组件而已，JSP 中的 Java 代码是通过“<%”和“%>”符号加入到 HTML 代码中的。如程序 1-1 所示的是一个简单的 JSP 页面的例子。虽然程序 1-1 比较简单，但是它仍包括了：HTML 组件、JSP 组件以及 JavaBeans 的调用。这个例子的功能是可打印出带有年、月的日期信息，并且还会根据浏览的时间显示“Good Morning”和“Good Afternoon”来表示对客户的欢迎。

程序 1-1 一个简单的 JSP 页面

```
<html>

<%@ page language="java" imports="com.wombat.jsp.*" %>

<H1>Welcome</H1>

<P>Today is </P>

<jsp:useBean id="timer" class="calendr.timer" />

<UL>

<LI>DAY:<%timer.getDayOfMonth()%>

<LI>Year:<%timer.getYear()%>

</UL>

<%if(Calendar.getInstance().get(Calendar.AM_PM)==Calendar.AM){%>

    Good Morning

<%}else{%>

    Good Afternoon

<%}%>

<%@include file="HappyDay.html"%>

</html>
```

在这段 JSP 源文件中包含了下面的一些内容。

- 一个 JSP 指令可以将有关的信息传送到 JSP 引擎。在这个例子中，如下的语句：

```
<%@ page language="java" imports="com.wombat.jsp.*" %>
```

指出该页面即将访问的一些 Java 编程语言的扩展的位置。指令必须被放置在相应的标记中。

- 固定模板数据。这些 JSP 引擎不能识别的标记将随着结果页面发送，通常这些标记是 HTML 或者 XML 标记。

- JSP 动作或者标记。这些标记通常作为标准的或定制的标记被实现，并且使用 XML 标记语法。

- 表达式。这是在标记之间要求 JSP 引擎计算的内容，在上面的例子中是 Day 和 Year。

- Scriptlet。是执行不为标记所支持的功能或者将所有的东西捆绑在一起的短小的脚本语言程序片段。

这个例子很小，但是包括的技术很全面。在实际应用中，可以将关键的处理封装在服务器端的组件中，这样开发人员就可以很容易地访问这些信息。基于 Java 的 Scriptlet 为执行不同的功能提供了一种灵活的方式，而且不要求扩展脚本语言。JSP 页面作为一个整体是可读和可理解的，这就使得查找或者预防问题以及共享工作内容变得更加容易。

1.1.4 JSP 的特点

1. 适用于任何平台

JSP 在设计时，充分考虑到应用平台的无关性。因此 JSP 是完全与平台无关的，可以在任何平台上运行。

2. 程序执行的效率高

在 JSP 下，代码被编译成 Servlet 并由 Java 虚拟机执行，编译过程只在程序第一次运行时进行，不需要在每次执行程序时都编译，因此，代码的执行效率很高。

3. 拥有 Java 的优势

JSP 技术用 Java 语言作为脚本语言，而 ASP 网页使用的是 VBScript 或者 JavaScript。Java 是成熟的、易扩充的、强壮的编程语言，优于基于 BASIC 的脚本语言。Java 使开发人员的工作在其他方面也变得一样容易。Java 语言提供防止内存泄漏的方法，在内存管理方面也大显身手。而且 JSP 为应用提供了强大的意外事件处理机制。

4. 强大的服务器端组件

JSP 可以使用 JavaBeans 来增强功能，JavaBeans 已经是很成熟的技术，可以避免开发人员许多重复性的工作，从而节省很多时间。

5. 强大的数据库支持

JSP 可以和任何与 JDBC 兼容的数据库建立连接，操纵数据库数据。Sun 公司还开发了 JDBC-ODBC bridge，用此技术，Java 程序可以访问带有 ODBC 驱动程序数据库。目前大多数数据库系统都带有 ODBC 驱动程序，所以 Java 程序能访问诸如 Oracle、Sybase、MS SQL Server 和 MS Access 等主流数据库。

1.2 JSP 与其他动态网页技术比较

1.2.1 JSP 与 CGI 比较

CGI (Common Gateway Interface, 通用网关接口) 是外部程序和 Web 服务器之间的标准编程接口, CGI 标准不是独立存在的, 它依赖于 HTML 和 HTTP 标准。简单地说, HTML 标准是一种让 Web 浏览器理解文本内容的标准, HTTP 标准是使 Web 浏览器和 Web 服务器能够相互对话的通信协议。许多 Internet 赖以生存的协议都是由 RFC 文档进行详细描述

的。

CGI 的基本概念很简单, 但是要编写出 CGI 程序却不是这样, 它需要实实在在的编程技巧, 必须具有较强的调试程序的能力。JSP 摆脱了传统 CGI 编程中的大部分复杂性, 使得服务器端的编程大大简化。同时, JSP 程序由 JSP 引擎执行, 而 JSP 执行引擎与 HTTP 服务器一起常驻内存, 并且每一时刻都只有一个 JSP 执行引擎在内存, 执行多个 JSP 程序。但是执行一个 CGI 程序时, 服务器都要建立一个进程来执行程序。当使用 Perl 之类的解释性执行语言进行 CGI 编程时, 对每一个 CGI 程序, 系统都要加载和清除 Perl 解释器一次, 这样系统同时运行多个 CGI 程序时, 在内存中就有多个 Perl 解释器, 非常消耗系统资源。所以 JSP 的执行效率比普通的 CGI 程序高。

1.2.2 JSP 与 ASP 的比较

总地说来, JSP 和 ASP (Active Server Pages) 在技术方面有很多相似之处。两者都是为基于 Web 应用实现动态交互网页制作所提供的技术环境。两者都能够为开发人员提供实现应用程序的编制与自带组件设计网页的从逻辑上分离的技术。并且两者都能够替代 CGI, 使网站建设与发展变得较为简单与快捷。

但是 JSP 和 ASP 仍然存在很多不同之处。两者本质的区别在于: 两者是来源于不同的技术规范组织, 其实现的基础——Web 服务器平台要求——不同。

JSP 和 ASP 技术最明显的不同点在于开发人员对两者各自软件体系设计的深入了解的方式不同。JSP 技术基于平台和服务器的相互独立, 技术支持由广泛的、专门的各种工具包、服务器的组件和数据库产品开发商提供。相比之下, ASP 技术则主要是依靠 Microsoft 的支持。

1. 平台和服务器的独立性

JSP 技术依附于一次写入后便可以运行在任何具有符合 Java 语法结构的环境。取代了过去的依附于单一平台或开发商的情况, JSP 技术能够运行在任何 Web 服务器上并且支持多家开发商提供的各种各样的工具包。

由于 ASP 是基于 ActiveX 控件技术的客户端和服务器的开发组件, 因此 ASP 技术基本上是局限于微软的操作系统平台之上的。ASP 的主要工作环境是微软的 IIS 应用程序结构, 又 ActiveX 对象又平台局限性, 所以 ASP 技术不能很容易实现在跨平台的 Web 服

务器上的工作。尽管 ASP 技术通过第三方所提供的组件和服务实现跨平台的应用程序，但是 ActiveX 对象必须事先放置于所选择的平台中。

2. 开发的开放过程及开放的源代码

Sun 公司自 1995 年以来已经用开放过程方法和国际 Java 组织合作开发和修改了 Java 技术与规范。针对 JSP 产品，Sun 授权了工具提供商、结盟公司、最终用户、协作商等。而且最近 Apache、Sun 和许多其他的公司及个人公开成立了一个强大的咨询机构以便任何公司和个人都能免费取得信息。

JSP 应用程序接口（API）毫无疑问已经取得成功，并将随 Java 组织的不断开放扩大而继续完善。相反，ASP 技术仅依靠本身的推动，其发展是建立在独占的、封闭性的开发过程基础上的。

表 1-1 JSP 与 ASP 的比较

	JSP 技术	ASP 技术
Web 服务器	任何 Web 服务器，包括 Apache、Netscape 和 IIS	微软的 IIS 或 Personal Web Server
操作系统平台	绝大多数的流行平台，包括 Solaris 操作系统、Windows 系统、MAC OS、Linux 及其他 UNIX 系列平台	微软的 Windows 系统
跨平台访问	支持 Web 系统中不同系列的计算机群，JSP 技术提供灵活、开放的选择：可以使用各种各样的工具提供商提供的工具，高度体现工业化标准输入与配置	需要第三方 ASP 的引入产品
兼容传统数据库	可以（用 JDBC API）	可以（用 COM）
集成数据源能力	能工作于任何符合 ODBC 规范的数据库，而且能访问符合 JDBC 技术规范数据库	能工作于任何符合 ODBC 规范的数据库
组件	JavaBeans，企业级 JavaBeans 或扩展的 JSP 标记	COM 组件
扩展工具支持	有	有
可重用、跨平台组件	JavaBeans，企业级 JavaBeans，定制 JSP 标记	没有
安全防范系统	有	没有
内存泄漏保护	有	没有
脚本语言	Java	VBScript，JavaScript
定制标记	有	没有

从开发人员的角度来看，ASP 和 JSP 技术都能开发具有交互式的、动态的内容和应用程序的 Web 站点。ASP 仅支持组件对象模型 COM，而 JSP 技术提供的组件都是基于 JavaBeans 技术或者 JSP 标签库的。由此可以看出两者虽有相同之处，但其区别是很明显的，

JSP 具有标签可扩充性，并且具有跨平台的可重用性。

这种基于组件的模式很有效地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便可用。像这样有效的方法在 JSP 中无处不在，并可将其打包成一个 JavaBeans 或一个工业标准化的 JavaBeans 组件。

1.2.3 JSP 与 Servlet 的比较

一个 Java Servlet 就是一个基于 Java 技术的运行在服务器端的程序，开发人员可以通过开发出来的 Servlet 接收来自 Web 浏览器的 HTTP 请求，动态地生成响应，然后发送包含 HTML 或者 XML 文档的响应到浏览器上。采用这种方法时，整个网页必须都在 Java Servlet 中制作。如果开发人员或者 Web 管理员想要调整页面显示，就不得不编辑并重新编译该 Java Servlet。

而 JSP 主要是将 HTML（或 XML）的标记进行扩展以加入 Java 代码，当 Web 服务器（或者是 Servlet 引擎、应用服务器等）支持 JSP 的时候，就会按照它的语法将其转化为 Servlet。此时 JSP 就变成了 Servlet，接着它就会被编译成 Java 执行码，按照一般的 Servlet 的运行方式载入、执行。

JSP 与 Servlet 的主要区别在于，JSP 为不了解 Servlet 的人员提供了一种可以简单的利用可扩展的标记制作一些动态网页的方法。一个 JSP 就是一个 HTML 再加上一些额外的程序代码，利用“<%”和“%>”与一般的 HTML 的语法相区别。

JSP 实际上还是被翻译成 Servlet，那么为什么不直接用 Servlet 呢？因为 JSP 有下面两个优点：

第一是 JSP 比 Servlet 开发方便。对 JSP 进行修改以后，马上就可以看到执行效果，一切都由 JSP 引擎来完成；而 Servlet 却要进行编译、放入执行代码等复杂的动作。

第二是 JSP 便于与 HTML 相混合。从 HTML 的角度来说，Java 语法以 JSP 包装并加以扩展，使 HTML 更容易加入动态部分。相反地，在 Servlet 中却要对 HTML 的输出作出额外的处理，尤其是在复杂的 HTML 网页中，加入的动态部分如果用 Servlet 来处理的话，那将是非常麻烦的事情。

但是事物总是有它的两面性的，HTML 与程序语言相混合势必使程序显得很混乱，纠错和维护都不容易。而传统程序开发中的语法和执行时的纠错都处理得很好。因此 JSP 比 Servlet 方便是指对于 Java 不熟悉的人来说的。

1.3 JSP 的运行环境

1.3.1 JSP 运行环境需要的组件

要运行 JSP，需要建立 JSP 运行的环境。

首先, 需要 Java 2 的软件开发工具箱。目前最新版的是 Sun 公司的 JDK 1.3, 可以在 Sun 公司主页上下载 (<http://Java.sun.com/jdk/>), 国内也有很多 ftp 站点提供下载服务。下面只介绍 JDK 在 Windows 下的安装, 下载 j2sdk1_3_0rc2-win.exe 后, 直接运行这个程序, 然后按照提示一步步操作就行了。

第二步是安装运行 JSP 的引擎。JSP 引擎是安装在 Web 服务器端或者是 Web 应用服务器上的组件。现在很多服务器都提供了对 JSP 的支持, 例如 JSWDK (Java Servlet Web Development Kit)、Tomcat、Jrun 或者其他支持 JSP 的 Web 服务器, 用户可以根据自己的需要选用不同的软件。下面介绍 JSWDK 和 Tomcat 在 Windows 下的安装和配置。本书的程序主要是在 Tomcat 环境下运行的。

1.3.2 JSWDK 的安装和配置

JSWDK 是一种常见的 JSP 的 Web 服务器。目前最新的版本是 jswdk1-0-1, 用户可以从 <http://Java.sun.com/products/jsp/> 处下载 jswdk1_0_1-win.zip。JSWDK 的安装只需要将 jswdk1_0_1-win.zip 解压到硬盘的一个目录里就可以了, 比如解压到 D:\Jswdk101\。如果将来不想保留 JSWDK, 直接删除就可以了, 不会有注册表和系统文件的问题。

JSWDK 默认的文档目录是 \Jswdk101\Webpages, 在这个目录下可以建立 JSP 文件和子目录, 就能在浏览器中用 <http://localhost:8080> 来访问这个目录了。

在 JSWDK 安装路径下有个 webserver.xml 文件, 在此文件中可以调整服务器和应用程序的设置, 包括默认端口和 Web 应用程序的属性。常见的端口是 80, 但是 JSWDK 默认的端口是 8080, 在 webserver.xml 中可以修改默认端口。方法如下:

该文件的倒数第 4 行为:

```
<Service id="service0" >
```

只需改为:

```
<Service id="service0" port="80">
```

即可。

打开编辑 JSWDK 安装目录下的 startserver.bat 文件, 在里面加上 JDK 目录的设置:

```
set JAVA_HOME=你安装 JDK 的路径
```

现在一切都已经准备完毕, 打开浏览器 (Internet Explorer 或者 Netscape 等), 在地址栏中输入 <http://localhost>, 即可看到 Webpages 下的 index.html 文件。如果想要看自己的源文件, 输入 <http://localhost/源文件名.jsp> 即可。

注意: 运行 JSP 的时候经常会出这样一个错误 “Out of environment space”, 这是 startserver.bat 文件运行的初始内存不够。需要将 startserver.bat 的属性中的初始内存改为 4096 或者以上。

运行 JSWDK 安装目录下的 stopserer.bat 即可停止 JSWDK 的 Web 服务器。