

21 世纪高职高专计算机专业教材

JSP 程序设计教程

陈雪莲 主编

丁 旭 吴学毅 副主编

清华大学出版社

北京交通大学出版社

· 北京 ·

目 录

第 1 章 JSP 技术概述	1
1.1 动态网页技术介绍	1
1.1.1 动态网页编程技术——CGI	2
1.1.2 动态网页编程技术——ASP	2
1.1.3 动态网页编程技术——PHP	2
1.1.4 动态网页编程技术——JSP	3
1.2 JSP 及其特点	3
1.3 JSP 的运行原理	4
1.4 用 JSP 开发 Web 应用程序的几种主要方式	5
1.4.1 直接使用 JSP	5
1.4.2 JSP + JavaBeans	6
1.4.3 JSP + JavaBeans + Servlet	6
1.4.4 J2EE 开发模式	6
1.5 JSP 与 JavaScript 比较	7
习题	8
实验	8
第 2 章 JSP 运行环境配置	9
2.1 JSP 运行环境介绍	9
2.1.1 对硬件条件的要求	9
2.1.2 对操作系统的要求	9
2.1.3 对软件环境的要求	10
2.2 安装配置 JDK 6.0	10
2.2.1 在 Windows 上安装 JDK 6.0	10
2.2.2 在 Windows 上设置环境变量	14
2.2.3 在 Linux 上安装 JDK 6.0	15
2.2.4 在 Linux 中设定环境变量	17
2.3 安装配置 Tomcat 6.0	18
2.3.1 在 Windows 上安装 Tomcat 6.0	18
2.3.2 在 Windows 上测试 Tomcat 6.0	21
2.3.3 在 Linux 上安装 Tomcat 6.0	23
2.3.4 在 Linux 上测试 Tomcat 6.0	23
2.4 安装配置 WebSphere Application Server 6.0	24
2.4.1 在 Windows 上安装 WAS 6.0	24

2.4.2	WAS 6.0 的一些设定	26
2.4.3	在 Windows 上测试 WAS 6.0	27
2.5	JSP 简单实例	28
	习题	30
	实验	30
第 3 章	Java 的基本语法	31
3.1	变量、常量与数据类型	31
3.1.1	变量与常量	31
3.1.2	数据类型	33
3.2	运算符与表达式	35
3.2.1	赋值运算符	35
3.2.2	算术运算符	36
3.2.3	关系运算符	37
3.2.4	布尔逻辑运算符	37
3.2.5	位运算符	37
3.2.6	条件运算符	39
3.2.7	Java 运算符的优先级	39
3.3	流程控制	40
3.3.1	if 结构	40
3.3.2	switch 结构	42
3.3.3	for 循环	45
3.3.4	while 循环	47
3.3.5	do...while 循环	48
3.4	面向对象的概念	50
3.4.1	定义类与对象的建立	50
3.4.2	构造函数的概念	53
	习题	55
	实验	55
第 4 章	JSP 语法	56
4.1	JSP 页面的基本结构	56
4.1.1	JSP 页面的组成	58
4.1.2	变量和方法的声明	58
4.1.3	Java 程序片	66
4.1.4	表达式	66
4.1.5	JSP 注释	68
4.2	指令标签	71
4.2.1	page 指令	71
4.2.2	include 指令	75
4.2.3	taglib 指令	77

4.3	动作标签	78
4.3.1	<jsp:param>动作标签	78
4.3.2	<jsp:include>动作标签	80
4.3.3	<jsp:forward>动作标签	82
4.3.4	<jsp:useBean>动作标签	84
4.3.5	<jsp:getProperty>动作标签	85
4.3.6	<jsp:setProperty>动作标签	85
4.3.7	<jsp:plugin>动作标签	85
	习题	88
	实验	89
第 5 章	JSP 的内置对象	91
5.1	对象与内置对象	91
5.1.1	对象、类和消息	91
5.1.2	内置对象	93
5.2	request 对象	93
5.2.1	request 对象的常用方法	95
5.2.2	request 对象应用实例	96
5.3	response 对象	108
5.3.1	response 对象的常用方法	108
5.3.2	response 对象应用实例	108
5.4	session 对象	115
5.4.1	session 对象的常用方法	116
5.4.2	session 对象应用实例	117
5.5	application 对象	124
5.5.1	application 对象的常用方法	124
5.5.2	application 对象应用实例	125
5.6	out 对象	130
5.6.1	out 对象的常用方法	130
5.6.2	out 对象应用实例	132
5.7	其他对象	134
5.7.1	pageContext 对象	134
5.7.2	config 对象	136
	习题	138
	实验	139
第 6 章	JSP 的数据库操作	140
6.1	JDBC 介绍	140
6.1.1	什么是 JDBC	140
6.1.2	JDBC 的特点	140
6.1.3	JDBC 的结构	141

6.1.4	JDBC 驱动的类型	142
6.2	通过 JDBC-ODBC 桥访问数据库的方法	143
6.2.1	利用 JDBC-ODBC 桥访问 Access 数据库	144
6.2.2	以 JDBC-ODBC 桥方式连接数据源步骤分析	148
6.2.3	利用 JDBC-ODBC 桥访问 SQL Server 2000 数据库	150
6.3	通过 JDBC 驱动程序访问数据库的方法	156
6.3.1	连接 SQL Server 2000 数据库	156
6.3.2	以 JDBC 驱动程序方式连接数据源步骤分析	157
6.3.3	以 JDBC 驱动程序方式连接其他数据库的方法	159
6.4	查询数据库	164
6.4.1	SQL 查询语句	164
6.4.2	执行查询语句	164
6.4.3	ResultSet 的方法	167
6.5	更新数据库	170
6.5.1	添加操作	170
6.5.2	修改操作	173
6.5.3	删除操作	176
6.6	JSP 在数据库应用中的相关问题	178
6.6.1	JSP 的分页浏览	178
6.6.2	汉字处理问题	181
6.6.3	JSP 访问 Excel 表	182
	习题	186
	实验	186
第 7 章	JSP 的文件操作	187
7.1	File 对象的使用	187
7.1.1	File 对象简介	187
7.1.2	目录和文件操作	189
7.2	JSP 中利用数据流来读写文件	194
7.2.1	使用字节流读写文件	194
7.2.2	使用字符流读写文件	199
7.3	文件操作应用实例	205
7.3.1	追加文件	205
7.3.2	复制文件	208
7.3.3	文件上传	210
7.3.4	文件下载	216
7.3.5	保存网页	218
	习题	219
	实验	220
第 8 章	JSP 与 JavaBeans	221

8.1	JavaBeans 概述	221
8.1.1	JavaBeans 的产生	221
8.1.2	JavaBeans 的技术介绍	222
8.1.3	JavaBeans 的种类	222
8.1.4	实现一个简单的 JavaBeans	222
8.2	JavaBeans 的创建	223
8.2.1	在 JSP 页面中声明 JavaBeans	223
8.2.2	读取 JavaBeans 的属性	227
8.2.3	设置 JavaBeans 的性质	227
8.2.4	JavaBeans 的存放目录	233
8.3	JavaBeans 的应用	234
8.3.1	在页面中使用 JavaBeans 组件	234
8.3.2	使用通用数据库 Bean 访问数据库	236
8.4	JavaBeans 应用实例	237
	习题	242
	实验	243
第 9 章	JSP 与 Servlet	244
9.1	Servlet 介绍	244
9.1.1	什么是 Servlet	244
9.1.2	Servlet 的技术特点	244
9.1.3	Servlet 的生命周期	245
9.1.4	开发一个简单的 Servlet	246
9.2	Servlet 编程接口	247
9.2.1	与 Servlet 实现相关	247
9.2.2	与 Servlet 配置相关	249
9.2.3	与 Servlet 异常相关	249
9.2.4	与请求和响应相关	250
9.2.5	会话管理	251
9.2.6	与 Servlet 上下文相关	251
9.2.7	Servlet 协作相关	252
9.2.8	过滤器	252
9.3	Servlet 的部署	253
9.3.1	部署描述符介绍	254
9.3.2	Servlet 的名字、类和初始化参数	255
9.3.3	启动装入优先级	255
9.3.4	Servlet 的映射	256
9.4	Servlet 与 JSP 结合编程	257
	习题	260
	实验	260

第 10 章 新闻发布系统	261
10.1 系统目的及特色	261
10.2 系统功能及描述	261
10.3 系统结构图	261
10.4 数据库设计	262
10.5 系统界面及说明	262
10.5.1 用户登录	262
10.5.2 新闻管理	266
10.5.3 新闻搜索	283
第 11 章 网上书店	287
11.1 系统目的及特色	287
11.2 系统功能及描述	287
11.3 系统结构图	288
11.4 数据库设计	288
11.5 系统界面及说明	289
11.5.1 系统通用模块	289
11.5.2 用户注册	291
11.5.3 用户登录	300
11.5.4 图书浏览	304
11.5.5 在线购书	311
11.5.6 购物车	319
11.5.7 订单处理	326
附录 A JavaScript 语法	341
A1 JavaScript 语言的简介	341
A2 JavaScript 语言与 Java 语言的区别	342
A3 JavaScript 的语法结构	343
A3.1 数据类型	343
A3.2 变量的定义与使用	343
A3.3 函数的定义及使用	343
A3.4 程序结构	344
A3.5 对象	346
A3.6 事件	346
附录 B WebSphere 5.1 的安装及使用说明	347
B1 WebSphere 的安装过程	347
B2 Workbench 窗口	350
B2.1 透视图	351
B2.2 视图和编辑器	351
B3 项目	353
B3.1 构建 Web 项目	353

B3.2 构建服务器项目	356
B4 服务器配置	357
附录 C Oracle 安装及使用说明	361
C1 Oracle 的安装	361
C2 Oracle 的使用	367
C2.1 SQL*PLUS 窗口	367
C2.2 Oracle 的企业管理器	367
参考文献	369

第 1 章 JSP 技术概述

JSP (Java Server Pages) 是由 Sun Microsystems 公司倡导、许多公司参与而建立的一种动态网页技术标准, 它以 Java 技术为主, 具有跨平台与跨网页服务器支持的特性。本章主要介绍动态网页基础知识、JSP 的优点、JSP 的运行原理、JSP 开发 Web 应用程序的主要方式及 JSP 与 JavaScript 的区别。

1.1 动态网页技术介绍

Internet 起源于 20 世纪 60 年代末的美国, 它在近几年迅速风靡全球, 其根本原因不仅在于它拥有卓越的国际通信功能, 更在于它拥有巨大的信息资源。所谓的 Internet 是指由分布在全世界成千上万的计算机网络遵循一定的通信协议, 并相互联系在一起而形成的国际互连网络。也就是说, Internet 是建立和使用这些网络的人群、团体、公司及各种网络资源的集合体。

Internet 无疑是一种重要的信息传播媒体。随着其迅猛发展, 将会有越来越多的企业、商团、政府机关、学校、科研机构需要在 Internet 上建立自己的网站, 发布信息。目前, 在 Internet 上广泛应用的是 WWW 系统。这种系统采用 HTML 文件格式(即通常所说的网页)传播信息, 用统一资源定位符连接世界各地计算机上的信息资源, 按照 HTTP 协议在浏览器和 Web 服务器之间进行通信。

网页主要分为静态网页和动态网页两种类型。静态网页是指网站的网页内容“固定不变”, 当用户浏览器通过互联网的 HTTP 协议向 Web 服务器请求提供网页内容时, 服务器仅仅是将原来已经设计好的静态 HTML 文档传送给用户浏览器。页面的内容使用的仅仅是标准的 HTML 代码, 最多再加上流行的 GIF98A 格式的动态图片。例如, 产生文字变化的动画效果, 而不是根据用户请求的不同产生不同的页面。若网站维护者要更新网页的内容, 就必须手动更新其所有的 HTML 文档。

那么, 什么是动态网页呢? 动态网页要具有以下几项特征。

- ☞ 交互性。网页会根据用户的要求和选择而动态改变和响应, 将浏览器作为客户界面, 这将是今后 Web 发展的主要趋势。
- ☞ 自动更新。无须手动更新 HTML 文档, 便会自动生成新的页面, 可以大大节省工作量。
- ☞ 因时因人而变。当不同的时间、不同的人访问同一网址时会产生不同的页面。

符合上述几项特征的网页则称为动态网页。

一般来说, 在 Web 服务器上可以通过多种方法和途径实现动态网页的发布, 常用的有 CGI、ASP、PHP 和 JSP 等。

1.1.1 动态网页编程技术——CGI

在早期,动态网页主要采用 CGI 技术,CGI 即 Common Gateway Interface (公用网关接口)。程序员可以使用不同的计算机编程语言编写适合的 CGI 程序,如 Visual Basic、Delphi 或 C/C++ 等。虽然 CGI 技术发展成熟而且功能强大,但由于具有编程困难、效率低下、修改复杂等缺陷,所以有逐渐被新技术取代的趋势。

1.1.2 动态网页编程技术——ASP

ASP (Active Server Pages) 是一套微软公司开发的服务器端脚本环境,它内含在 IIS 中。它没有提供自己专门的编程语言,而是使用许多已有的脚本语言进行 ASP 编程,结合 HTML 网页和 ActiveX 控件,建立动态、交互且高效的 Web 服务器的应用程序。

ASP 是在 Web 服务器端运行,运行后再将运行结果以 HTML 格式传送至客户端的浏览器,减轻了客户端浏览器的负担,大大提高了交互的速度,也比一般的脚本语言要安全得多。ASP 具有如下特点。

- ✧ 使用 VBScript、JavaScript 等简单易懂的脚本语言,结合 HTML 代码,即可快速地完成网站的应用程序。
- ✧ 无须编译,容易编写,可在服务器端直接运行。
- ✧ 使用普通的文本编辑器,例如,Windows 的记事本即可进行编辑设计。
- ✧ 与浏览器无关,客户端只要使用可执行 HTML 编码的浏览器,即可浏览 ASP 所设计的网页内容。ASP 所使用的脚本语言 VBScript 和 JavaScript 均在 Web 服务器端执行,客户端不需要执行这些脚本语言。
- ✧ ASP 能与任何 ActiveX Scripting 语言相兼容。除了 VBScript 和 JavaScript 语言设计外,通过 Plug-in 的方式,还可以使用由第三方所提供的其他脚本语言,例如,Perl、TCL 等。
- ✧ ASP 的源程序不会被传送到客户浏览器,因而可以避免所编写的源程序被他人剽窃,从而保证了程序的安全性。
- ✧ 可使用服务器的脚本产生客户端的脚本。
- ✧ ActiveX 服务器组件具有无限可扩充性。可以使用 Visual Basic、Java、Visual C++、COBOL 等编程语言编写所需要的组件。

但 ASP 技术也并非完美无缺,由于它基本上是局限于微软的操作系统平台,主要的工作环境是微软的 IIS 应用程序结构,又因 ActiveX 对象具有平台特性,所以 ASP 技术要实现在跨平台 Web 服务器上工作不是很容易。

1.1.3 动态网页编程技术——PHP

PHP (Professional Hypertext Preprocessor) 是基于 GNU 协议的自由软件,是内嵌于 HTML 中的、当今 Internet 上最为火热的服务器脚本编程语言,它的语法借鉴了 C、Java 及 Perl 语言,只需要很少的编程知识就能使用标准的 PHP 建立一个真正交互的 Web 站点。PHP 具有如下特

点。

- ☞ 基于服务器端。PHP 是在 Web 服务器运行的，所以 PHP 程序可以很大、很复杂而不会降低客户端的运行速度。
- ☞ 嵌入 HTML 中。PHP 语言嵌入到 HTML 内部，很容易学习。
- ☞ 简单的语言。PHP 语言坚持以基本语言为基础，但其功能也强大到足以支持任何类型的 Web 站点。
- ☞ 效率高。与其他的解释性语言相比，PHP 语言消耗较少的系统资源。当 PHP 作为 Apache Web 服务器一部分时，运行代码不需要调用外部二进制程序，服务器解释脚本不需要承担任何额外的负担。
- ☞ 分析 XML。用户可以组建一个可以读取 XML 信息的 PHP 版本。
- ☞ 数据库模块。用户可以使用 PHP 存取 Oracle、Sybase、MSSQL、MySQL、PostgreSQL、dBase、FoxPro、Solid 等类型的数据库，以及任何支持 ODBC 标准的数据库。
- ☞ 文件存取。PHP 有许多支持文件存取的函数。
- ☞ 文件处理。PHP 有许多函数处理字符串，其中包括模式匹配的能力。
- ☞ 复杂的变量。PHP 支持标量、数组、关联数组等变量，这给用户提供了支持其他高级数据结构的坚实基础。
- ☞ 图像处理。用户可以使用 PHP 动态地创建图像。

1.1.4 动态网页编程技术——JSP

JSP 是基于 Java Servlet 及整个 Java 体系的 Web 开发技术。可以在多平台上运行，而且基于 JSP 技术的应用程序比基于 ASP 的应用程序易于维护和管理，被许多人认为是未来最有发展前途的动态网页技术。本书将着重讲述 JSP 的基本语法及其基本的应用。

1.2 JSP 及其特点

JSP 的发展历程主要如下所述。

1998 年 4 月，Sun 公司发布 JSP 0.90 规范。

1999 年 1 月，Sun 公司在发布了 JSP 0.92 规范的同时，推出支持 JSP 的 Web 服务器——Java Web Server 2.0。

1999 年 11 月，Sun 公司发布了 JSP 1.1 规范，同时推出了 JSWDK 1.0.1 和 Server 2.2 版。

2000 年 9 月，Sun 公司发布 JSP 1.2 规范，同时推出 Java Server API 2.3 版。

JSP 规范是 Web 服务器、应用服务器、交易系统及开发工具供应商之间广泛合作的结果，而且该规范还在不断完善与发展。Sun 公司开发出这个规范来整合和平衡已经存在的规范，对 Java 编程环境进行支持的技术和工具。其结果是产生了一种新的、开发基于 Web 应用程序的方法，给使用基于组件应用逻辑的页面设计者以强大的功能支持。

JSP 是一个纯 Java 平台的技术，它主要用来产生动态网页内容。其中包括：HTML、DHTML、XHTML 和 XML。JSP 技术能够让网页人员轻易建立起功能强大、有弹性的动态内容。

JSP 技术有如下特点。

(1) 一次编写，随处运行

作为 Java 平台的一部分，JSP 技术拥有 Java 语言“一次编写，随处运行”的特点。随着越来越多的供应商将 JSP 技术添加到他们的产品中，用户可以使用自己所选择的服务器和工具，但并不影响当前的应用。

(2) 搭配可重复使用的组件

JSP 技术可依赖于重复使用跨平台的组件来执行更复杂的运算、数据处理。开发人员能够共享开发完成的组件，或者能够加强这些组件的功能，让更多用户或是客户团体使用。基于组件的开发方法，可以加快整体开发过程，也大大降低公司的开发成本和人力。

(3) 采用标签化页面开发

Web 网页开发人员不一定是熟悉 Java 语言的程序员。因此，JSP 技术能够将许多功能封装起来，成为一个自定义的标签，这些功能是完全根据 XML 的标准来制定的，即 JSP 中的标签库。因此，Web 页面开发人员可以运用自定义好的标签来达成工作需求，而无须再写复杂的 Java 语法，让 Web 页面开发人员亦能快速开发出动态内容网页。

今后，第三方开发人员和其他人员可以为常用功能建立自己的标签库，让 Web 网页开发人员能够使用熟悉的开发工具，来执行特定功能的工作。

(4) 内容的生成与显示进行分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面。使用 JSP 标识或小脚本来生成页面上动态内容。生成内容的逻辑被封装在标识和 Java Beans 组件中，并且捆绑在小脚本中，所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 Bean 中，那么其他人，如 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。

在服务器端，JSP 引擎解释 JSP 标识和小脚本，生成所请求的内容，并且将结果以 HTML 或 XML 页面的形式发送回浏览器。这有助于作者保护自己的代码，而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

(5) 支持多端企业应用架构

随着网际网络的发展，未来服务越来越繁杂，而且不受地域的限制，因此必须放弃客户-服务器的架构，转向更具有威力、弹性的分散性对象系统。基于 Java Servlet 及整个 Java 体系的 JSP 技术支持这一应用架构。

(6) 广泛的应用平台

支持 JSP 的 Web 服务器——Apache 可以在 UNIX、Linux 和 Windows 操作系统环境中运行，而且 Java、JSP+JavaBeans 可以在任何平台上通行无阻，因此 JSP 有更广泛的运行平台。

1.3 JSP 的运行原理

JSP 的运行原理如图 1-1 所示，从形式上看与客户-服务器模式相似，但实际是多端企业应用模式。

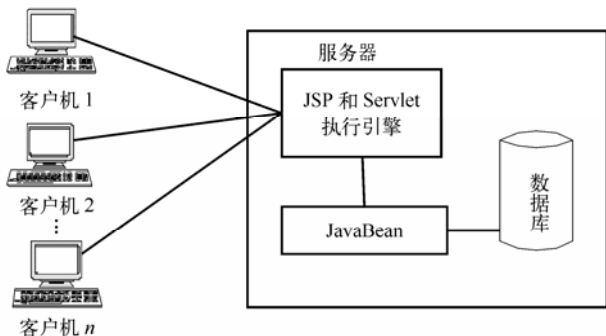


图 1-1 JSP 运行原理

当 Web 服务器收到一个扩展名为.jsp 页面的请求时，触发 JSP 引擎。JSP 引擎是负责管理并执行 JSP 程序的实体。如果请求 URL 是以.jsp 扩展名为结束，Web 服务器和 Servlet 引擎调用它。JSP 引擎首先检查 JSP 文件是新的还是未修改过的文件，对这两种情况会有不同的处理。

对于新的 JSP 文件，JSP 引擎把 JSP 文件转换成 Java Servlet，然后使用标准的 Java 编译器编译 Servlet，使用标准的 API 执行 Java Servlet。

对于旧文件则直接进行编译，省略了前面翻译的工作，接下来的步骤是一样的。JSP 是将网页的表现形式和服务端端的代码逻辑分开。作为服务器进程的 JSP 页面，首先被转换成 Servlet。Servlet 支持 HTTP 协议的请求和响应。当多个用户同时请求一个 JSP 页面时，应用实例化线程来响应请求，这些线程由 Web 服务器进程来管理。

JSP 在执行前先被编写成字节码，然后字节码由 Java 虚拟机解释执行，比源码解释的效率高——服务器上还有字节码的 Cache 机制，能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢，因为它被编译成 Cache，以后就快得多了。另外，JSP 源程序不大可能被下载，特别是 JavaBeans 程序完全可以放到不对外的目录中。JSP 的执行过程如图 1-2 所示。

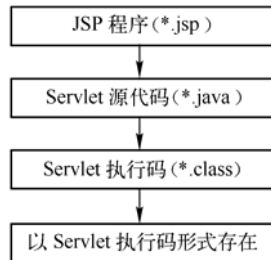


图 1-2 JSP 的执行过程

1.4 用 JSP 开发 Web 应用程序的几种主要方式

对于不同规模的 Web 系统，使用不同方式的 JSP 进行开发。JSP 开发方式有直接使用 JSP、JSP+ JavaBeans、JSP + JavaBeans + Servlet 和 J2EE 四种开发模式。

1.4.1 直接使用 JSP

对于小型的 Web 站点，可以直接使用 JSP 来构建动态网站。这种站点最为简单，所需要的仅仅是简单的留言板、动态日期等基本功能。对于这种开发模式，一般可以将所有的动态处理部分都放置在 JSP 的程序块中，就像一般使用 PHP 或 ASP 开发动态网页一样。

1.4.2 JSP + JavaBeans

中型站点面对的是数据库查询、用户管理和小量的商业业务逻辑。对于这种站点，不能将所有的操作全部交给 JSP 页面来处理。在单纯的 JSP 中加入 JavaBeans 技术将有助于这种中型网站的开发。利用 JavaBeans 将很容易完成如数据库连接、用户登录与注销、商业业务逻辑封装的任务。如将常用的数据库连接写为一个 JavaBeans，既方便了使用，又可以使 JSP 文件简单而清晰。通过封装，还可以防止一般的开发人员直接获得数据库的控制权。

1.4.3 JSP + JavaBeans + Servlet

无论用 ASP 还是 PHP 开发动态网站，长期以来都有一个比较重要的问题，就是网站的逻辑关系和网站的显示页面不容易分开。常常可以看见一些夹杂着 `if...then...`，`case`，`select` 或是 `if{...}` 和大量显示用的 HTML 代码的 ASP、PHP 页面，即使是有着良好的程序写作习惯的程序员，其作品也几乎无法阅读。另一方面动态 Web 的开发人员也在抱怨，将网站美工设计的静态页面和动态程序合并的过程是一个异常痛苦的过程。

如何解决这个问题呢？解决方法就是将 JSP、JavaBeans 和 Servlet 组合使用。因为 Servlet 在不再担负动态页面生成的任务以后，开始担负起决定整个网站逻辑流程的任务。在逻辑关系异常复杂的网站中，借助于 Servlet 和 JSP 良好的交互关系和 JavaBeans 的协助，完全可以把网站的整个逻辑结构放在一个或几个核心的 Servlet 上来处理网站的逻辑。通过调用 JSP 页面来完成客户端的请求。

1.4.4 J2EE 开发模式

在 J2EE 开发模式中，整个系统可以分为三个主要的部分。

(1) 视图

视图就是用户界面部分，在 Web 应用程序中也就是 HTML、XML、JSP 页面。这个部分主要处理用户看到的东西，动态的 JSP 部分处理了用户可以看见的动态网页，而静态的网页则由 HTML、XML 输出。

(2) 控制器

控制器负责网站的整个逻辑，用于管理用户与视图发生的交互。可以将控制器看作处在视图和数据之间，对视图如何与模型交互进行管理。通过使视图完全独立于控制器和模型，可以轻松替换前端客户程序，也就是说，网页制作人员将可以独立自由地改变 Web 页面，而不用担心影响这个基于 Web 的应用程序的功能。

在 J2EE 中，控制器的功能一般是由 Servlet、JavaBeans、Enterprise JavaBeans 中的 Session Beans 来担当的。

(3) 模型

模型就是应用业务逻辑部分。这一部分的主要角色是 Enterprise JavaBeans，借助于 EJB 强大的组件技术和企业级的管理控制，开发人员可以轻松地创建出可重用的业务逻辑模块。

1.5 JSP 与 JavaScript 比较

Java Script 是 Netscape 公司的产品，其目的是为了扩展 Netscape Navigator 功能而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言，它的前身是 LiveScript。JSP 则是 Sun 公司推出的一种用来产生动态网页内容的规范，以 Java 技术为主，并且具有跨平台和跨网页服务器支持的特性。

由于 JSP 主要采用 Java 技术，因此下面对 JavaScript 与 Java 两种语言的不同点做一下比较。

(1) JavaScript 与 Java 是两个公司的不同的两种产品

Java 是 Sun 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序开发。而 JavaScript 是 Netscape 公司的产品，是为扩展 Netscape Navigator 功能而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言。

(2) 基于对象和面向对象

JavaScript 是基于对象的，而 Java 是面向对象的，即 Java 是一种真正的面向对象的语言，即使是开发简单的程序，也必须设计对象。JavaScript 是一种脚本语言，它可以用来制作与网页无关的、与用户交互作用的复杂软件，它是一种基于对象和事件驱动的编程语言，因而它本身提供了非常丰富的内部对象供设计人员使用。

(3) 解释与编译

两种语言在其浏览器中所执行的方式不一样。Java 的源代码在传递到客户端执行之前，必须经过编译，因而客户端上必须具有相应平台上的仿真器或解释器，它可以通过编译器或解释器实现独立于某个特定的平台编译代码的束缚。JavaScript 是一种解释性编程语言，其源代码在发往客户端执行之前不需要经过编译，而是将文本格式的字符代码发送给客户，由浏览器解释执行。

(4) 变量

两种语言所采取的变量是不一样的。Java 采用强类型变量检查，即所有变量在编译之前必须作声明。而 JavaScript 中，变量在使用前不需要声明，而是解释器在运行时检查其数据类型。

(5) 代码格式不一样

Java 是一种与 HTML 无关的格式，必须通过像 HTML 中引用外媒那样进行装载，其代码以字节码的形式保存在独立的文档中。Java Script 的代码是一种文本字符格式，可以直接嵌入 HTML 文档中，并且可动态装载。编辑 HTML 文档就像编辑文本文件一样方便。

(6) 嵌入方式不一样

在 HTML 文档中，两种编程语言的标识不同。Java 使用<APPLET>...</APPLET>来标识，而 JavaScript 使用<SCRIPT>...</SCRIPT>来标识。

(7) 静态绑定和动态绑定

Java 采用静态联编，即 Java 的对象引用必须在编译时进行，以便使编译器能够实现强类型检查。而 JavaScript 采用动态联编，即 JavaScript 的对象引用在运行时进行检查，如不经编

译则无法实现对象引用的检查。

(8) 图形和 GUI 能力

Java 提供了 JavaScript 所不具备的产生 GUI 要素的图形原始处理能力。

(9) 应用不同

Java 主要应用在 Web 服务器端，用于服务器端应用程序的开发；而 JavaScript 既可应用在服务器端，也可应用在客户端，只是随着服务器端动态网页开发技术不断发展，大多数用户使用 JavaScript 辅助客户端开发，而服务器端则使用其他动态技术开发，因此 JavaScript 在服务器端应用开发已经被人们遗忘。

习题

1. 什么是动态网页？开发动态网页的技术有几种？
2. JSP 有什么特点？
3. JSP 如何运行？
4. 使用 JSP 开发 Web 应用程序主要有几种模式？
5. 描述 Java 与 JavaScript 的区别。

实验

1. 上网浏览网站，比较静态网页与动态网页的区别。
2. 在互联网上比较 CGI、ASP、PHP 和 JSP 的速度。

第 2 章 JSP 运行环境配置

编写 JSP 网页除了必须具备 HTML 与 Java 程序的编写能力外，还必须要能运行 JSP 程序的环境，最基本的安装程序为 JDK 工具与应用服务器。JDK 是运行 Java 的工具程序，是 Sun 公司开发的一套公开的编程语言工具；应用服务器是运行网页的服务器，例如 Tomcat、Resin、WebSphere 等。本章介绍 JSP 的运行环境、JDK 的安装与配置、Tomcat 服务器的安装与配置、IBM WebSphere Studio 的安装。

2.1 JSP 运行环境介绍

JSP 是 Sun 公司开发的、基于 Java 语言的技术，实现了一次编写、随处运行、与硬件设备无关的特点。因此 JSP 可以在任何硬件平台运行，如 Intel、Alpha 机等，但对计算机的配置有一个基本要求。在软件方面，可以跨平台使用，如在 Linux、UNIX、Windows 等系统下使用，对网页服务器要求也十分宽松。下面将从硬件条件、操作系统和软件环境三个方面加以阐述。

2.1.1 对硬件条件的要求

对硬件条件的要求主要是对 CPU、内存、硬盘空间、显示器等的要求，其他硬件对 JSP 无影响。表 2-1 为对硬件要求的最低配置，建议使用当前中档以上计算机配置为佳。

表 2-1 对硬件条件的要求

CPU	使用 Pentium III 或更高级别的处理器
内存	128 MB 以上
硬盘	1 GB 以上
光驱	8 倍速以上 CD-ROM
显示器	VGA

2.1.2 对操作系统的要求

JSP 能够运行在目前绝大多数的操作系统上，目前在普通用户中用到的系统主要为 Windows 系列和 Linux 系列。表 2-2 列出了当前常用的一些操作系统。

表 2-2 对操作系统的要求

操作系统	
Windows 系列	Linux 系列
Microsoft Windows 98, Windows NT 4.0, Windows 2000 Professional, Windows 2000 Server, Windows XP, Windows 2003 Server 等	Red Hat Linux, Turbo Linux, Red Flag Linux