

高等学校计算机应用人才培养系列教材

JSP应用开发技术

贾素玲 王强 主编
张成 朱磊 方敬 编著

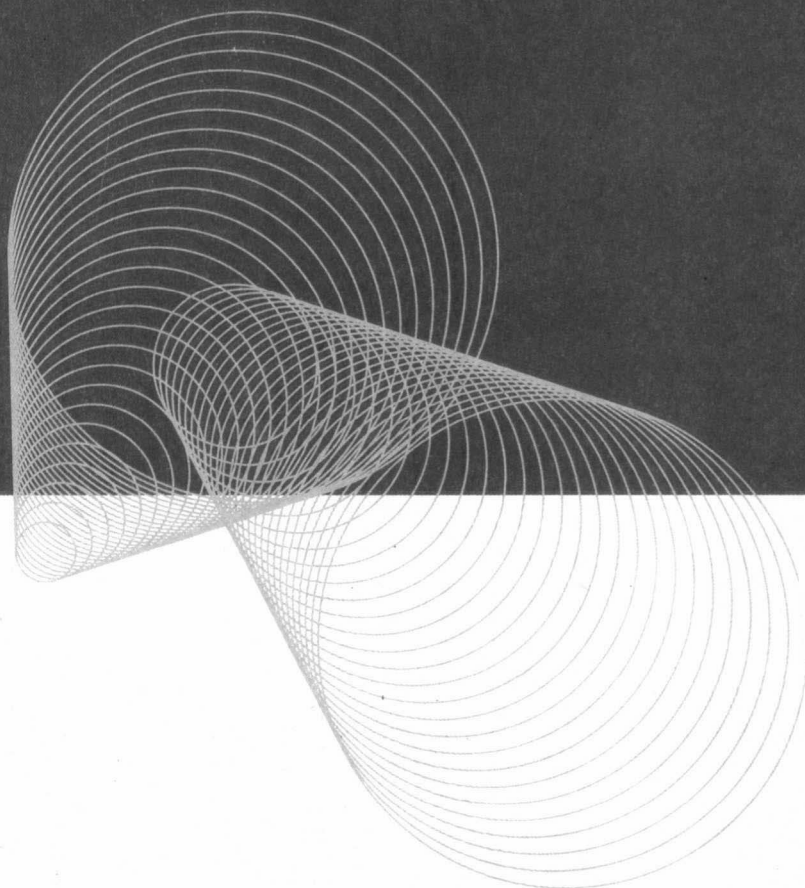


清华大学出版社

高等学校计算机应用人才培养系列教材

贾素玲 王强 主编
张成 朱磊 方敬 编著

JSP应用开发技术



清华大学出版社
北京

内 容 简 介

本书面向 JSP 实际应用开发,通过大量的实例,循序渐进地为读者介绍了有关 JSP 开发所涉及的各类知识,同时还包含 JDK、Tomcat 和 Struts 等 JSP 相关知识。全书共分 10 章,内容依次包含了 JSP 的基本常识、开发环境、语法、内置对象、JavaBean、Servlet、访问数据库、JSP 与 XML、Struts 基础知识和 JSP 实例应用。内容由浅入深,涵盖了 JSP 的各个主要知识点。本书语言通俗易懂,结构安排合理。

本书适合 JSP 初学者和 JSP 应用开发人员进行阅读和学习,同时也可作为计算机及相关专业的教材使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

JSP 应用开发技术/贾素玲,王强主编;张成等编著. —北京:清华大学出版社,2007.6

(高等学校计算机应用人才培养系列教材)

ISBN 978-7-302-15002-2

I. J… II. ①贾… ②王… ③张… III. JAVA 语言—主页制作—程序设计—高等学校—教材

IV. TP393.092

中国版本图书馆 CIP 数据核字(2007)第 048334 号

责任编辑:索梅 赵晓宁

责任校对:时翠兰

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社总机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印刷者:北京国马印刷厂

装订者:三河市深源装订厂

经 销:全国新华书店

开 本:185×260 印 张:19.5 字 数:469 千字

版 次:2007 年 6 月第 1 版 印 次:2007 年 6 月第 1 次印刷

印 数:1~4000

定 价:27.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:022481-01

打开本套丛书的朋友,如果你抱有获得 IT 应用成功的愿望或想尽快加入令人羡慕的 IT 行业发展的期盼,相信这套丛书将使你豁然开朗。

信息技术是当代人类社会中发展最快、渗透性最强、应用面最广的先导技术,这些日新月异的高新技术不仅改变了世界,也改变了人们的生活。掌握计算机基础知识,提高计算机应用能力,不再是计算机专业人士的特长,并且已经成为许多行业必备的基本技能。计算机技术不仅仅是一门独立的学科,而且日益成为其他学科飞速发展的助推器。对于当代的大学生,或者想进入五彩缤纷的 IT 应用领域的朋友,要跟上信息时代的步伐,就必须掌握现代科学技术,调整自己的知识结构,使自己具备开拓创新的意识和能力,以适应当前社会发展。

信息时代离不开信息化人才,掌握计算机基础知识和提高应用能力,是信息化人才培养的一个重要环节。我们知道,印度是当今的计算机软件出口王国,软件产业的发展成就令人赞叹。之所以如此,除了政府长期实施的一系列扶持政策外,主要还是得益于持之以恒、行之有效的系列化的优秀教材和教育培训。

本套系列教材的形成,一是根据教育部人才培养的指导方针,以培养 IT 应用人才为目标,在引进推广印度 IT 培训教材的同时,借鉴国内外的计算机专业、信息管理专业人才培养的经验,力求课程的设置重点突出、循序渐进,将知识学习与能力培养相结合,使理论与实践完美融合;二是以企业对信息化人才的需求为依据,把面向对象、数据库、软件体系结构、软件工程的思想融入教材体系中,将基本技能的培养与主流应用技术相结合,培养具有扎实基础的实用型人才;三是在多年从事信息化人才培养和信息系统项目开发的经验基础上,充分理解企业人才需求层次和大学传统人才培养模式的错位,把课程体系的理论知识学习成功转变为应用能力的掌握,使大学真正成为企业的人才资源库。这就是我们开发这套系列教材的最终目标。

计算机基础和语言类的教材,可以说是多如牛毛,那么本套系列教材又凭什么在市场竞争中获得优势呢?

一是知识系列化。本套系列教材以“面向对象、数据库、软件工程、信息系统开发、……、项目管理”的思想为主线,以 Java、XML 为主流技术,形成系列化的能力培养阶梯,使得学生能从一个初学入门者,逐渐成长为合格的 IT 应用技术开发人才。

二是理论实践化。本套系列教材从应用实践的需要入手,合理组织每门课程的结构和内容,在总体框架下,通过大量的案例训练使学生掌握程序设计语言的核心技术和应用技巧,使理论知识在实践中得到升华,在不知不觉中能力得到积累和提高。

三是案例实战化。本套系列教材的编著者既是教师又是软件工程师,具有丰富的教学经验和软件项目开发经验,善于把握计算机技术与学生能力需求之间的尺度,按照循序渐进、突出重点的原则,从多个承担的科研项目中精心抽取和设计教材中引入的案例,使案例

与技术更接近实战的要求。

四是重点内容课件化。除学生用书外,本套系列教材还配有完善的课件,既可以直接为教师上课服务,也能为学生提供本课程的学习要点,引导学生深刻理解每一章节的主体内容,轻松完成相关知识的学习和案例设计。

总之,本套系列教材的指导思想是力求内容新颖、概念清晰、结合实践需要,突出应用能力的培养,使学生在循序渐进的学习中,达到软件项目开发的技术能力要求,成为满足企业信息化需要的人才。

本套系列教材在编写过程中得到了多方专业人士的指导、支持和帮助,在此表示由衷的感谢。尽管我们在教材编著时力求准确,但难免存在不当之处,恳请各位同仁和读者批评指正。

编 者

2007年3月10日

前言

随着 Internet 的日益流行和 Java 技术的广泛应用,越来越多的人开始重视怎样使用 Java 这一强大语言。

过去,Java 应用主要集中在客户端,Web 服务器传统上使用 C、Perl 以及 CGI 等开发技术,这些开发技术晦涩难懂,开发效率和安全性都比较低。后来,代替 Perl 及 CGI 的 ASP(Active Server Page)、PHP(Personal HomePage Tools)和 JSP(Java Server Page)等一系列动态网页技术的出现使得这些上手困难的技术逐步淡出市场,而 JSP 又是这些新生技术中最优秀的代表。

JSP 是当今一项非常重要的动态 Web 开发技术,它充分继承了 Java 的众多优势,包括一次编写到处运行的承诺、高效的性能以及强大的可扩展能力。特别是结合了 Servlet 和 JavaBean 技术,使得 JSP 技术较其他 Web 开发技术有着得天独厚的优势。几乎所有的 Java Web 应用都是使用 JSP。

从 1998 年 JSP 0.9 公开问世到最新的 JSP 2.1 规范,JSP 技术的发展可谓日新月异。即使读者从来没有接触过 JSP,笔者也相信读者一定听说过 JSP、Servlet、JavaBean 等名词。JSP 在网络中的地位正随着 Java 技术的不断完善而变得越来越重要。

本书面对的主要是那些 JSP 初学者,利用通俗的语言和简单的例子,系统地为读者介绍了 JSP 基本常识、JSP 开发环境、JSP 基本语法、JSP 内置对象、JavaBean、Servlet、JDBC 和 XML 等方方面面的知识,大量的实例、配套的思考题,必将使读者对 JSP 的认识有大幅度的提高。

第 1 章以一个典型的 JSP 实例开篇,通过该实例读者对 JSP 程序会有一个感性认识。

第 2 章主要为读者介绍了如何构建 JSP 开发环境,包括 JDK 和 Tomcat 的配置、JSP 开发工具的选择等。

第 3 章详细介绍了 JSP 基本语法,包括 JSP 注释、声明、表达式、程序段、JSP 指令、JSP 动作等,这些都是学习 JSP 所必备的知识。

第 4 章主要介绍了 JSP 内置对象,包括 application 对象,session 对象,request 对象和 response 对象等 JSP 常用的核心对象。笔者在介绍每一个对象时都列举了相应的实例,这样有助于读者更加快速掌握各 JSP 对象的实际使用方法。

第 5 章介绍了 JavaBean 的基本知识及其使用方法。

第 6 章介绍了 Servlet 相关知识,包括 Servlet 类结构、Servlet 运行条件、Servlet 生命周期等。

第 7 章介绍了如何使用 JSP 访问数据库,通过实例,读者可以学习 JDBC、数据库连接池等非常实用的知识。

第 8 章通过典型的实例为读者介绍了 XML 的基本知识以及 JSP 访问和控制 XML 的

基本方法。

第9章为读者介绍了有关 Struts 应用开发的基本知识,包括 Struts 基本思想、开发流程、Struts 组件和 Struts HTML 标记等,作为学习 JSP 的扩展。

第10章为读者介绍了一个基于 MVC 多层构架的 JSP 应用实例,通过该实例读者不但可以掌握 JSP 各类元素的综合用法,还可以加深对多层构架 Web 应用的理解。

最后的附录提供了 JSP 语法参考和 Tomcat 服务器配置参考,这对读者日后的实际开发会有很大的帮助。

正如本书第1章所指出,JSP 本身是一项发展很快、实践性很强的技术,读者在学习 JSP 时应当注重多实践、多利用网络资源、多交流,只要读者能够把对本书的学习和自身的实践结合起来,相信学好 JSP 应该不是什么难事。

当然,由于笔者水平有限,书中难免有错误和牵强之处,欢迎读者批评和指正!

最后,我们期待着本书成为读者最好的朋友!

编 者

2007年3月



目 录

第 1 章 JSP 简介	1
1.1 初识 JSP	1
1.2 第一个 JSP 实例	2
1.3 JSP 运行原理	5
1.4 如何学习 JSP	5
本章小结	6
习题 1	6
第 2 章 构建 JSP 开发环境	7
2.1 JSP 运行环境	7
2.1.1 JDK	7
2.1.2 Tomcat	8
2.2 JDK 的安装与配置	9
2.3 Tomcat 的安装和配置	12
2.4 JSP 开发工具	15
2.5 测试第一个 JSP 应用程序	17
本章小结	18
习题 2	18
第 3 章 JSP 基本语法	19
3.1 注释	19
3.2 声明	20
3.3 表达式	20
3.4 程序段	21
3.5 JSP 指令	21
3.5.1 include 指令	21
3.5.2 page 指令	22
3.5.3 taglib 指令	24
3.6 JSP 动作	26
3.6.1 <jsp:include>动作	26
3.6.2 <jsp:forward>动作	28

3.6.3	<jsp:plugin>动作	30
3.6.4	<jsp:param>动作	32
3.7	JSP 异常	33
3.8	JSP 字符使用惯例	33
	本章小结	34
	习题 3	34

第 4 章 JSP 内置对象 35

4.1	JSP 内置对象概述	35
4.1.1	Application Scope	36
4.1.2	Session Scope	36
4.1.3	Request Scope	36
4.1.4	Page Scope	38
4.2	HTML 中的<form>和<input>	39
4.2.1	<form>标记	39
4.2.2	<input>标记	40
4.3	application 对象	40
4.4	config 对象	42
4.5	exception 对象	42
4.6	out 对象	44
4.7	page 对象	46
4.8	pageContext 对象	47
4.9	request 对象	49
4.10	response 对象	52
4.11	session 对象	54
4.12	session 跟踪	56
	本章小结	61
	习题 4	61

第 5 章 使用 JavaBean 63

5.1	什么是 JavaBean	63
5.1.1	JavaBean 中的属性	65
5.1.2	JavaBean 中的事件	68
5.2	如何在 JSP 中使用 JavaBean	69
5.2.1	<jsp:useBean>	71
5.2.2	<jsp:setProperty>	72
5.2.3	<jsp:getProperty>	73
5.2.4	使用 JavaBean 的其他问题	74
5.3	JavaBean Scope	75

5.3.1	page 范围的 JavaBean	76
5.3.2	request 范围的 JavaBean	77
5.3.3	session 范围的 JavaBean	81
5.3.4	application 范围的 JavaBean	83
5.4	JavaBean 使用实例	83
5.4.1	使用 JavaBean 处理购物车	83
5.4.2	使用 JavaBean 处理数据库连接	87
	本章小结	90
	习题 5	90
第 6 章	JSP 与 Servlet	91
6.1	Servlet 概述	91
6.2	Servlet 的类结构	92
6.2.1	HttpServlet 类	92
6.2.2	HttpSession	94
6.2.3	ServletConfig	95
6.2.4	ServletContext	95
6.3	Servlet 的运行条件	97
6.4	用 Servlet 实现 HelloBUAASem108	97
6.5	用 Servlet 访问 request 对象	100
6.6	Servlet 生命周期	102
6.7	JSP、Servlet 与 JavaBean	103
6.8	用 Servlet 实现购物车	104
	本章小结	112
	习题 6	112
第 7 章	JSP 访问数据库	113
7.1	JDBC 简述	113
7.2	JDBC 中的类结构	114
7.3	JDBC 驱动	119
7.4	JSP 数据库操作实例	121
7.4.1	在 JSP 页面中直接访问数据库	121
7.4.2	在 Servlet 中使用 JDBC 技术	124
7.4.3	在 JavaBean 中封装数据库操作	129
7.5	数据库连接池	133
7.5.1	连接池的基本原理	134
7.5.2	在 Tomcat 中配置连接池	134
7.5.3	使用连接池访问数据库的实例	137
	本章小结	139

习题 7	139
第 8 章 JSP 与 XML	140
8.1 XML 简介	140
8.2 DOM 和 SAX	141
8.3 使用 DOM 解析 XML	142
8.4 使用 SAX 解析 XML	148
8.5 JSP 中的 XML 语法	154
本章小结	156
习题 8	156
第 9 章 Struts 应用开发基础	158
9.1 MVC 设计模式	158
9.2 Struts 实现 MVC 的机制	159
9.3 Struts 的第一个实例	159
9.3.1 实例执行流程	160
9.3.2 实例开发	160
9.4 Struts 配置文件	176
9.4.1 <struts-config>元素	176
9.4.2 <data-sources>元素	177
9.4.3 <form-beans>元素	179
9.4.4 <global-exceptions>元素	179
9.4.5 <global-forwards>元素	180
9.4.6 <action-mappings>元素	181
9.4.7 <controller>元素	182
9.4.8 <message-resources>元素	183
9.4.9 <plug-in>元素	183
9.5 Struts 组件	184
9.5.1 Struts 控制器组件	185
9.5.2 Struts 模型组件	188
9.5.3 Struts 视图组件	192
9.6 Struts HTML 标记	193
本章小结	204
习题 9	205
第 10 章 JSP 实例应用	206
10.1 使用 JSP 实现会员服务系统	206
10.1.1 系统需求概述	206
10.1.2 业务实体说明	207

10.2	系统体系结构设计	208
10.3	数据库设计	210
10.4	各功能模块的实现	211
10.4.1	系统运行环境配置	211
10.4.2	编写实用类	212
10.4.3	编写抽象基类	217
10.4.4	模型层的实现	222
10.4.5	业务层的实现	240
10.4.6	视图层的实现	245
	本章小结	276
	习题 10	277
附录 A JSP 语法总结		278
附录 B Tomcat 服务器配置参考		282
参考文献		296

第 1 章 JSP 简介

1.1 初识 JSP

网络和 Web 技术的不断发展,使得传统的静态 HTML 网页已经无法满足实际应用的需求,更多的时候,需要有数据库和动态网站开发技术的支持。

早期的动态网站开发技术主要使用 CGI(Common Gateway Interface)技术。CGI 的基本原理是,将浏览器提交至 Web 服务器的数据通过环境变量传递给其他外部程序,经外部程序处理后,再由 CGI 把处理结果传送给 Web 服务器,最后由 Web 服务器把处理结果返回浏览器。

CGI 本质上只是定义了这种基于环境变量的规范。因此,任何符合 CGI 规范的语言都可以用来编写 CGI,这也是 CGI 的优点之一。但 CGI 有着致命的缺点,即 CGI 本身功能过于弱小,难以满足日益复杂的 Web 应用。此外,CGI 编写困难,可扩展性和安全性都比较差,运行效率也严重低下。例如,每新增一个 CGI 程序,都要求在服务器上新增一个进程,如果多个用户并发访问该程序,很有可能在短时间内耗尽服务器的所有可用资源,导致系统崩溃。

CGI 的这些固有缺点,导致新的技术纷纷面世,微软的 ASP(Active Server Pages),Tcx 的 PHP(Hypertext PreProcessor)和 Sun 公司的 JSP(Java Server Pages),是继 CGI 之后最为流行的动态网站开发技术。

JSP 是 1998 年由 Sun 公司推出的基于 Java 语言的动态网页技术标准,被认为是最有前途的 Web 技术之一。

JSP 具有以下一些特点:

- 一次编写,随处运行。JSP 作为 Java 技术家族的一部分,继承了 Java“一次编写,随处运行”(Write once,Run anywhere)的特点。使用 JSP,意味着即使更换服务器平台,也不会影响现有程序的正常运行。
- 可重用组件技术。JSP 可以通过 JavaBean、EnterpriseBean 等组件技术来封装较为复杂的应用,开发人员可以共享已经开发完成的组件,从而大大提高 JSP 应用的开发效率和可扩展性。
- 标记化页面开发。JSP 将许多常用功能封装起来,以 XML 标记的形式展现给 JSP 开发人员,这样,即使 JSP 开发人员不熟悉 Java 语言,也可以轻松编写 JSP 程序,这就

降低了 JSP 的开发难度。此外,标记化的 JSP 应用,也有助实现“形式和内容相分离”这一重要原则。形式和内容相分离,使得 JSP 页面结构更加清晰,有助于日后的维护。

- ▶ 对大型复杂 Web 应用支持良好。未来的 Web 应用将日趋复杂,基于数据库的多层企业应用构架将日渐成为主流。JSP 提供的 Servlet、JavaBean、EnterpriseBean 和 JDBC 等技术,以及 JSP 本身所具有的健壮的存储管理和安全性,使得 JSP 很容易整合到多层应用结构中。JSP 完全有能力支持高度复杂的基于 Web 的应用。

1997 年,Sun 公司推出了 Servlet 1.0 规范。Servlet 可以简单理解为可供 Web 服务器端调用和执行的 Java 类。基于 Servlet 1.0 规范,Sun 公司于 1998 年 4 月又推出 JSP 0.90 规范,这是因为 JSP 在本质上就是一种 Servlet,它在第一次被访问时会自动编译成 Servlet,并且存储在服务器内存中,因而接下来在对 JSP 页面的访问中,服务器将会有非常快的响应(避免了 CGI 为每个 HTTP 请求生成一个新进程的问题)。而对 Web 开发人员而言,JSP 显然比 Servlet 要简单,更加适合于 Web 开发。

经过 8 年的发展,JSP 已经有了最新的 2.1 规范,并且还在不断完善中。JSP 已经能够胜任越来越多类型的应用了,不论采用什么开发工具,开发人员都可以使用 JSP 页面来创建可移植的 Web 应用,在不同的 Web 应用服务器上运行。

1.2 第一个 JSP 实例

例 1.1 是一个典型的 JSP 实例。

例 1.1

first-example.jsp

```
<% @page language = "java" import = "java.util.*" %>
<% @page contentType = "text/html; charset = GB2312" %>
<html>
  <head>
    <title>
      第一个 JSP 实例
    </title>
  </head>
  <body>
    <h1>第一个 JSP 实例</h1>
    <jsp:useBean id = "clock" class = "mycalendar.BaseCalendar" />
    <ul>
      <li>日期: <% = clock.getDate () %>
      <li>年份: <% = clock.getYear () %>
    </ul>
    <% -- 计算当前时间为 AM 或 PM -- %>
    <% !int time = Calendar.getInstance().get(Calendar.AM_PM); %>
    <%
      if (time == Calendar.AM){
```

```

%>
早上好
<%
}
else {
%>
下午好
<%
}
%>
<% @ include file = "copyright.html" %>
</body>
</html>

```

图 1-1 是例 1.1 的运行结果。

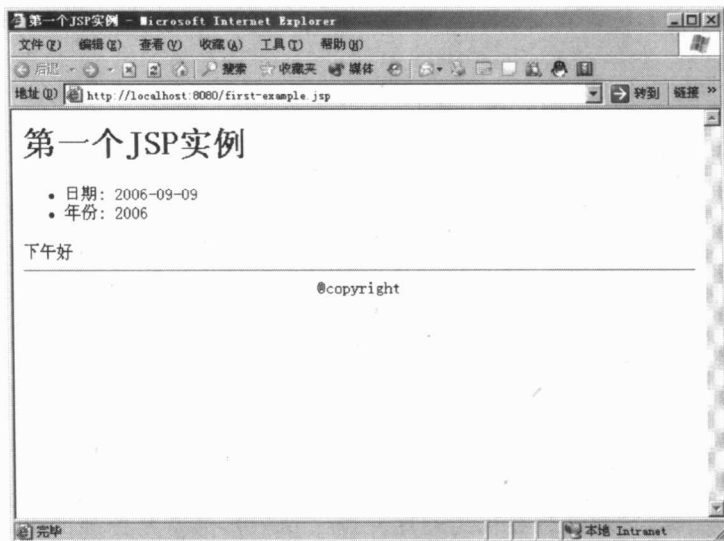


图 1-1 例 1.1 的运行结果

例 1.1 中的代码由两部分组成：HTML 代码(正常字体部分)和 JSP 代码(黑体部分)。在传统的 HTML 文件中加入 JSP 代码，就构成了 JSP 网页。

作为 JSP 初学者，如果不明白黑体部分的 JSP 代码，这是很正常的，如果完全不明白其中的 HTML 代码，那么最好先学习一下有关 HTML 的知识，可以参考本系列教程中的《HTML 网页设计》一书。此外，由于 JSP 是基于 Java 技术的，因此想要熟练掌握 JSP 的知识，必须对 Java 基础知识有所了解，读者可以参考本系列教程中的《Java 程序设计》一书。

JSP 代码和 HTML 代码混合在一起，以 .jsp 文件形式存在，下面对例 1.1 进行解释说明。

- ▶ 形如“<%@ %>”的标记为 JSP 指令，其作用是设定页面级的指令信息、引用其他外部文件或者定制标记库，它会随 JSP 页面编译成 Servlet，然后由 JSP 引擎负责处理。如下代码：

```
<% @page language = "java" import = "java.util.*" %>
```

定义 JSP 语言的类型为 Java, 并且引入了该页面操作所需要的 java. util. Calendar 类。

```
<% @page contentType = "text/html; charset = GB2312" %>
```

定义当前 JSP 页面采用 GB2312 编码。

```
<% @include file = "copyright.html" %>
```

在 JSP 文件中包含一个 HTML 文件, 该 HTML 文件中的所有内容都会在 JSP 页面中显示。

- ▶ 形如 XML 标记的部分称为 JSP 动作(也称 JSP 行为), 它们通常封装了一些常用的功能。JSP 动作利用类似 XML 的语法来管理各种 JavaBean 组件。如下代码:

```
<jsp:useBean id = "clock" class = "mycalendar.BaseCalendar" />
```

用于初始化该 JSP 页面中需要用到 JavaBean。有关 JavaBean 的知识, 将在第 5 章详细讲述, 现在读者只需要了解, JavaBean 是一种可供 JSP 调用的 Java 类。

- ▶ 形如“<!% %>”的部分是 JSP 声明, 它与 Java 中的声明类似, 都是用于定义变量以便在随后的程序中使用。如下代码:

```
<!% int time = Calendar.getInstance().get(Calendar.AM_PM); %>
```

声明一个整型变量并且赋值为当天时间(AM 或 PM 对应的整型变量)。

- ▶ 形如“<% = %>”的内容称为 JSP 表达式, 由变量和常量组成。它通常包含来自 Web 服务器端的数据。如下代码:

```
<% = clock.getDate() %>
```

```
<% = clock.getYear() %>
```

两个表达式分别调用了 JavaBean 组件的 getDate() 方法和 getYear() 方法, 用来获取当前日期和年份。

- ▶ 形如“<% %>”的部分被称为程序段(Scriptlet), 通常是标准的 Java 代码, 在运行后可以自动转化成字符串, 然后插入到所在 JSP 页面中的位置显示。实例 1.1 中的程序段根据当前时间显示合适的问候语:

```
<%  
if (time == Calendar.AM){  
%>  
早上好  
<%  
}  
else {  
%>  
下午好  
<%  
}  
%>
```


- ▶ 形如“<%---%>”的部分为注释。JSP 中的注释与 HTML 类似,在被 JSP 引擎执行时,它会被剥去,也就是说,JSP 注释中的内容并不会在客户端浏览器中显示。与 HTML 注释不同的是,HTML 注释位于“<!-- -->”之间,而 JSP 注释位于“<%---%>”之间。例如:

```
<%-- 计算当前时间为 AM 或 PM --%>
```

有的时候,JSP 页面仅由静态 HTML 代码组成,这样的 JSP 看似与普通的 HTML 没有区别,语法也完全一致,只是它同样需要转换成 Servlet。

通过对例 1.1 中各段 JSP 代码的说明,读者应该对 JSP 代码的基本组成有了感性的认识,同时也会带来一些疑问。本书第 3 章将详细讲述 JSP 语法,读者带着这些疑问去学习第 3 章,收获将会更大。

1.3 JSP 运行原理

图 1-2 显示了 JSP 的基本运行原理。首先由浏览器向 Web 服务器提出访问 JSP 页面的请求(Request),然后由 JSP 容器负责将 JSP 转换成 Servlet,产生的 Servlet 经过编译后生成类文件,然后再把类文件加载到内存进行执行。最后由 Web 服务器将执行结果响应(Response)给客户端浏览器。

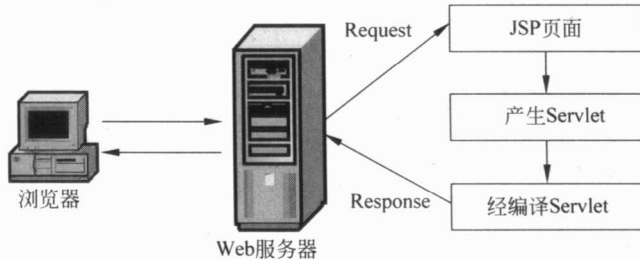


图 1-2 JSP 运行原理

JSP 需要转换成 Servlet 是因为 JSP 的执行效率要低于 Servlet,但这仅限于第一次执行。由于 JSP 在第一次执行后即被编译成类文件,当再次重复调用时,如果 JSP 容器没有发现该 JSP 页面修改的痕迹,就会直接执行编译后的类文件,而不是重新编译 Servlet。因此,除了第一次的编译会花比较久的时间外,之后的 JSP 和 Servlet 的执行速度就几乎相等。当然,如果 JSP 容器检查到 JSP 页面被修改过,则需要重新进行编译。

1.4 如何学习 JSP

在因特网时代,各类技术发展迅速,各种专用名词层出不穷,JSP 作为一种重要的 Web 技术也不例外。从 1998 年 JSP 技术诞生至今,JSP 所包含的内容已经越来越丰富了,即使一直在关注 JSP,也可能有很多从未听说过的 JSP 相关名词。作为 JSP 的初学者,实在没有