

高职高专计算机系列规划教材



JSP 2.1

动态网站设计案例教程

陈战胜 · 沈大林 主编
王 辉 钮文良 王廷梅 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高职高专计算机系列规划教材

JSP 2.1 动态网站设计案例教程

陈战胜 沈大林 主 编
王 辉 钮文良 王廷梅 副主编
魏雪英 张 伦 郑淑晖 王浩轩 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书依据 JSP 2.1 和 JSTL 1.2 规范, 利用 JDK 6.0 和 Tomcat 6.0 作为 Web 应用开发与调试平台, 通过学院新闻版块、BBS 讨论区典型实际项目, 以及大量案例, 全面介绍 JSP 2.1 相关的 JSTL 标记的应用。全书共分为 11 章。第 1 章介绍搭建 JSP 开发环境; 第 2 章通过经典的“Hello World”程序, 介绍虚拟目录的设置, Web 站点的建立, 测试 JSTL 是否正常工作, 以及 JSP 页面运行机制。第 3~10 章, 分别介绍新闻版块的首页设计、概要设计、目录浏览和全文显示、快速检索、分页浏览、自定义标记、后台管理、后台管理(续)的设计与制作。第 11 章介绍 BBS 讨论区网站的设计与制作。附录 A、B、C 分别介绍了 EL 表达式语言、JSTL 标记库及 JSP 元素。

本书可作为高职高专计算机专业教材使用, 也可以作为非计算机专业本科教材, 另外还可作为中等职业学校计算机专业的教材和广大计算机爱好者自学的教材。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

JSP 2.1 动态网站设计案例教程/陈战胜, 沈大林主编. —北京: 电子工业出版社, 2011.5
高职高专计算机系列规划教材
ISBN 978-7-121-13268-1

I. ①J… II. ①陈… ②沈… III. ①JAVA 语言—程序设计—高等职业教育—教材 ②网站—设计—高等职业教育—教材 IV. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2011)第 060550 号

策划编辑: 吕 迈

责任编辑: 郝黎明 特约编辑: 田学清 罗树利

印 刷: 北京丰源印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 19 字数: 486 千字

印 次: 2011 年 5 月第 1 次印刷

印 数: 4 000 册 定价: 33.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言



JSP 2.1 规范推出的 EL 表达式、JSTL 和自定义标记等技术使得 Web 开发人员通过使用标记,来完成原来在 JSP 页面中嵌入 Java 语言脚本完成的功能,使得 Web 应用的开发更加快捷、易于维护。目前在实际的 Web 项目开发中,为了代码的可读性和 Web 应用的可维护性,不允许在 JSP 页面中出现任何的 Java 语言脚本,应该将用户交互界面和业务处理流程分离。

采用 JSP 2.1 中的 JSTL 标记可以替代 JSP 网页中大部分甚至全部程序代码,可以实现对数据库的查询、添加、删除和修改等操作,从而简化 Web 开发过程,降低开发难度,缩短开发周期,降低开发成本。

本书依据 JSP 2.1 和 JSTL 1.2 规范,利用 JDK 6.0 和 Tomcat 6.0 作为 Web 应用开发与调试平台,通过学院新闻版块、BBS 讨论区典型实际项目,以及大量案例,全面介绍 JSP 2.1 相关的 JSTL 标记的应用。

本书采用“项目导向,任务驱动”的教学方法,强调学以致用,将典型项目合理分解为 43 个案例进行教学,不仅覆盖大部分的核心知识点,而且体现了“重点项目带课程”的优点,特别适合学生自主学习。本教程中,每个案例均由“案例描述”、“设计过程”、“相关知识”和“思考练习”四部分组成。本书内容由浅入深、循序渐进,使读者在阅读学习时,不但知其然,还要知其所以然,不但能够快速入门,而且可以达到较高的水平。在本书编写过程中,作者努力遵从教学规律,注意知识结构与实用技巧相结合,注意学生的认知特点,注意提高学生的学习兴趣 and 创造能力的培养。

通过本书的学习,读者可以快速掌握基于 JSP 2.1 技术的 Web 应用开发技术。书中提供的所有实例程序,全部通过了本书所采用开发平台上的调试和运行。

本书免费提供的电子教学课件,以及相关实例的程序源代码也可以从华信教育资源网(www.hxedu.com.cn)免费下载。

本书由陈战胜、沈大林主编,王辉、钮文良、王廷梅副主编。参加本书编写工作的人员有魏雪英、肖柠朴、曾昊、王爱赅、万忠、郑淑晖、张伦、王浩轩、陶宁、郑鹤。

本书可作为高职高专计算机专业教材使用,也可以作为非计算机专业本科教材,另外还可作为中等职业学校计算机专业的教材和广大计算机爱好者自学的教材。

由于时间仓促,编者水平有限,错误之处在所难免,敬请广大读者批评指正。

编 者

目 录

CONTENTS

第 1 章 搭建 JSP 开发环境 / 1

1.1 JSP 历史简介 / 1

1.1.1 JSP 简介 / 1

1.1.2 JSP 与 ASP、ASP+ 的比较 / 1

1.1.3 JSTL 简介 / 3

1.2 JSP 2.1 的特点 / 4

1.2.1 运行环境变化 / 4

1.2.2 引入表达式语言 / 6

1.2.3 SimpleTag / 6

1.2.4 使用 JSP fragment / 8

1.3 【案例 1】JSP 2.1 开发环境的配置 / 10

第 2 章 第一个 JSP 2.1 程序 / 21

2.1 【案例 2】经典“Hello World”程序 / 21

2.2 【案例 3】虚拟目录的设置 / 24

2.3 【案例 4】建立 Dreamweaver 站点 / 28

2.4 【案例 5】测试 JSTL 是否正常工作 / 32

2.5 【案例 6】JSP 页面运行机制 / 37

第 3 章 新闻版块——首页设计 / 45

3.1 【案例 7】学院新闻版块 / 45

3.2 【案例 8】管理员登录页面 / 57

3.3 【案例 9】新闻管理页面 / 61

第 4 章 新闻版块——概要设计 / 64

4.1 【案例 10】新闻版块的功能概述 / 64

4.2 【案例 11】新闻版块数据库表结构设计 / 66

4.3 【案例 12】SQL 语句基础 / 72

第 5 章 新闻版块——目录浏览和全文显示 / 78

5.1 【案例 13】自动显示学院新闻 / 78

5.2 【案例 14】限定显示新闻的数量 / 89

5.3 【案例 15】自动排序并显示新闻 / 92

5.4 【案例 16】实现新闻全文显示 / 96

第 6 章 新闻版块——快速检索 / 106

6.1 【案例 17】单个关键字模糊查找 / 106

6.2 【案例 18】多个关键字模糊查找 / 113

第 7 章 新闻版块——分页浏览 / 123

7.1 【案例 19】新闻目录的分页浏览 / 123

7.2 【案例 20】新闻搜索结果的分页浏览 / 135

7.3 【案例 21】新闻分类查找的分页显示 / 147

第 8 章 新闻版块——自定义标记 / 159

8.1 【案例 22】标记文件开发 JSP 页面页脚 / 159

8.2 【案例 23】用简单标记实现<greet>标记 / 167

8.3 【案例 24】新闻内容的分段落显示 / 173

第 9 章 新闻版块——后台管理 / 180

9.1 【案例 25】管理员登录的验证 / 180

9.2 【案例 26】新闻发表 / 191

9.3 【案例 27】新闻编辑 / 200

第 10 章 新闻版块——后台管理 (续) / 211

10.1 【案例 28】新闻删除 / 211

10.2 【案例 29】新闻的高级筛查 / 218

10.3 【案例 30】新闻多幅图片的显示 / 228

10.4 【案例 31】新闻图片的处理 / 237

第 11 章 BBS 讨论区网站 / 245

11.1 【案例 32】BBS 讨论区功能概述 / 245

- 11.2 【案例 33】BBS 数据库表结构设计 / 247
- 11.3 【案例 34】BBS 讨论区首页 / 251
- 11.4 【案例 35】BBS 讨论区切换 / 253
- 11.5 【案例 36】BBS 讨论区帖子搜索 / 255
- 11.6 【案例 37】BBS 讨论区的分页显示 / 257
- 11.7 【案例 38】BBS 讨论区的用户注册和登录 / 260
- 11.8 【案例 39】BBS 讨论区用户信息的管理 / 263
- 11.9 【案例 40】BBS 讨论区——发表新主题 / 266
- 11.10 【案例 41】BBS 讨论区——回复帖子 / 267
- 11.11 【案例 42】BBS 讨论区的删帖和删图 / 269
- 11.12 【案例 43】BBS 讨论区的后台管理 / 272

附录 A EL 表达式语言 / 276

- A.1 EL 保留字 / 276
- A.2 EL 运算符 / 276
- A.3 EL 隐含对象 / 277

附录 B JSTL 标记库 / 280

- B.1 核心标记库 / 280
- B.2 格式标记库 / 284
- B.3 XML 标记库 / 286
- B.4 SQL 标记库 / 287
- B.5 函数标记库 / 289

附录 C JSP 元素 / 291

- C.1 指令元素 (Directive Elements) / 291
- C.2 脚本元素 (Scripting Elements) / 292
- C.3 行为元素 (Action Elements) / 292

搭建 JSP 开发环境

本章简要介绍 JSP 和 JSTL，以及 JSP 2.1 的新特点等理论知识，介绍了如何下载和安装 JDK（Java Development Kit）、Tomcat 应用服务器和 JSTL 标记库，以及如何配置标记库描述文件。另外，还介绍了 Microsoft SQL Server 2005 的安装和基本操作方法。

学习 Web 开发，首先要学习网站开发环境的搭建。在本教程中，我们将利用 Tomcat+JSP 2.1+Microsoft SQL Server 2005 完成学院“新闻版块”的开发过程。很好地掌握本章介绍的内容，可以为后续各章的学习打下坚实的基础。

1.1 JSP 历史简介

1.1.1 JSP 简介

JSP（Java Server Pages）是由 Sun 公司倡导、许多其他公司参与创建的一种新动态网页技术标准，类似 ASP、PHP、ColdFusion 等技术标准。JSP 技术是一个纯 Java 平台的技术，它主要用来产生动态网页内容，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。

在传统的网页 HTML 文件（扩展名为“*.htm”或“*.html”）中加入 Java 程序片段（Scriptlet）和 JSP 标记，构成了 JSP 网页（扩展名为“*.jsp”）。Servlet/JSP Container 收到客户端发出的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户端。其中程序片段可以是操作数据库、重新定向网页及发送 E-mail 等，这些都是建立动态网站所需要的功能。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，与客户端的浏览器无关，因此 JSP 称为 Server-Side Language。

1.1.2 JSP 与 ASP、ASP+的比较

早期的 Web 开发曾采用 Perl 语言编写公共网关接口（Common Gateway Interface, CGI）程序来访问数据库，并将 HTML 页面输出到客户端，这种方法开发周期长，成本高。

一般说来，Sun 公司的 Java Server Pages（JSP）和 Microsoft 的 Active Server Pages（ASP）在技术方面有许多相似之处。两者都为动态网页的技术，并且双方都能够替代

CGI 技术,使网站的开发过程能够大大缩短,在性能上也有较高的表现,更重要的一点是,两者都能够为程序员提供组件设计的功能,通过组件设计,将网页中逻辑处理部分交由组件负责处理(ASP 使用 COM 组件、JSP 则有 JavaBean 组件),而和网页上的排版、美工分离。

JSP 和 ASP 的本质区别是:两者来源于不同的技术规范组织。

1. 平台和服务器的弹性

Microsoft 公司推出的 ASP 和 ASP.NET 技术主要运行在 Microsoft 公司的 Windows 平台上,并且搭配其 Web 服务器 IIS (Internet Information Services)。另外,由于安装 Windows 平台要求服务器必须使用 x86 系列或 Intel Itanium 系列的 CPU,因此无法采用性能表现相当优异的 UNIX 平台。

Sun Microsystem 公司(简称 SUN)于 1995 年发布了 Java 语言,其最大特点就是平台无关性。1997 年, Sun 推出了 Servlet 技术,使 Java 语言被运用到 Web 网站的开发,利用 Java Servlet,再适当配合 JavaBeans,就可以开发网站,但工作量较大。1998 年, Sun 推出了 JSP 技术,其主要运行在操作系统上的一个 Java Virtual Machine (JVM) 虚拟机上,因此,它能够跨越所有的平台。例如: NT、Windows 2000、Solaris、Linux、OS/390、AIX、HP-UX 等,除了能在各式各样的操作系统上执行,并且能搭配现有的 Web 服务器: Apache、IIS、Netscape Enterprise Server 等,将静态的 HTML 网页交由执行速度较快的 Web Server 处理,而动态产生网页的部分,就交由 JSP Container 来执行。由上述可知, JSP (Java Server Pages) 技术在跨平台的表现比 ASP 来得更有弹性。

2. 语法结构

在语法结构上, ASP 和 JSP 均采用“<%”和“%>”作为标记符号,区别在于标记符号之间所使用的语言: ASP 采用 JavaScript 或 VBScript, 而 JSP 采用 Java。

由于 JSP 是在 JVM 上执行程序,且提供强大的异常事件处理机制,因此不会因为程序撰写的疏忽,而导致服务器操作系统的损毁。并且, Java 语言提供防止直接存取内存的功能,避免由于存取内存产生的错误造成服务器的损毁。

3. 开放的开发环境

自从 1995 年, Sun 公司已经开放技术与国际 Java 组织合作开发和修改 Java 技术与规范。针对 JSP 的新技术, Sun 公司授权工具供货商(如 Macromedia)、同盟公司(如 Apache、Netscape)、协力厂商及其他公司。最近, Sun 公司将最新版本的 Servlet 2.4 和 JSP 2.1 的源代码发放给 Apache,以求 JSP 与 Apache 紧密地相互发展。Apache、Sun 和许多其他的公司及个人公开成立一个咨询机构,以便任何公司和个人都能免费取得信息,详见网址“<http://jakarta.apache.org>”。

JSP 应用程序接口(API)毫无疑问已经取得成功,并随着 Java 组织不断扩大其应用的范围,目前全力发展 Java 技术的厂商不胜枚举,例如:最近 IBM 公司大力推广的 WebSphere 家族,正是完全支持 J2EE 标准而开发的。数据库厂商 Oracle 也发展自己的 Application Server 来和自己公司本身数据库产品 Oracle 9i 进行紧密的结合。Amazon 系

统的供货商 BEA 公司，它的产品 WebLogic 也是完全支持 JSP 技术和 J2EE 规范的。

相反，ASP 技术仅依靠微软本身进行推动，其发展建立在独占、封闭的基础之上，并且微软本身的技术又只允许在微软相关平台的服务器上执行，因此，在标准方面显得有点力不从心。

► 4. 语法的延展性

ASP 和 JSP 都使用标记与 Scripting Language 来制作动态 Web 网页，JSP 2.1 新规范中，能够让程序员自由扩展 JSP 标记来应用。JSP 开发者能自定义标记库 (Tag Library)，所以网页制作者能充分利用与 XML 兼容的标记技术强大的功能，大大降低对 Java 语法的依赖，并且也可以利用 XML 强大的功能，做到数据、文件格式的标准化。

1.1.3 JSTL 简介

JSP 标准标记库 (JSP Standard Tag Library, JSTL) 是由 JCP (Java Community Process) 所制定的标准规范，它主要提供给 Java Web 开发人员一个标准通用的标记函数库。

在 JSP 2.1 出现以前，对于大型规模网站的开发，JSP 被用在表示层，配合 Servlet、Java Beans 和 EJB 等进行网站开发。对于较简单的小型网站，则可全部使用 JSP 进行开发，这使 Java 网站的开发工作得到很大的简化，但仍然需要编写程序。

JSP 2.1 的出现，使得上述情况得到极大的改观。Web 程序员能够利用 JSTL 标记和 EL 语言来替代 JSP 网页中的部分甚至全部 Java 程序 (Scriptlet)，不仅缩短开发周期、降低了开发难度和开发成本，而且能够进一步提高程序的阅读性、维护性和方便性。

JSP 标准标记库按照功能分为 5 大部分，分别如下。

(1) 核心标记库 (Core tag library): 包括设置变量、输出变量、移除变量、捕获异常、条件处理、循环和迭代、URL 操作等。

(2) I18N 格式标记库 (I18N-capable formatting tags library): 包括处理国际化问题，以及格式化和解析日期、时间、数字、百分比和货币格式等。

(3) 函数标记库 (Functions tag library): 可调用一批函数，完成如字符串处理等功能，当函数库不够用时，可以使用自定义函数。

(4) SQL 标记库 (SQL tag library): 提供了对数据库的处理功能，可以设置数据源，查询数据库、更新数据库。

(5) XML 标记库 (XML tag library): 用于开发基于 XML 的 Web 应用。

若出现上述 5 类 JSTL 标记不能够满足需要的情况，可利用 Java 语言开发自定义标记来满足特定的需求。

使用 JSTL，除上述的 5 个标记库外，还必须使用 EL 语言 (Expression Language, 表达式语言)，在后续章节中，将通过实例介绍 EL 的应用。

► 思考练习 1-1

1. JSP 的全称是_____。
2. JSTL 的全称是_____，含义为 JSP 标准标记库。
3. JSP 标准标记库按照功能划分，包含哪 5 部分？

1.2 JSP 2.1 的特点

在 JSP 2.1 规范中, Sun 为了统一 JSP 与 JSF 的 EL 表达式语法, 将#{ } 作为保留了的表达式, 使得在 JSP 中能够直接引用 JSF 的 managedBean。也就是说, 采用 JSP 2.1 规范的服务器会自动编译#{ } 表达式;

在 JSP 2.1 规范中, EL 表达式存在两种语法格式: \${ } 和#{ }, 对这两种表达式采用相同的解析方法和评价方法。在 JSP 2.1 规范中, 对于 EL 表达式\${ }, 采用及时评价机制, 而对于 EL 表达式#{ }, 则采用延迟评价机制。

鉴于本教程不会涉及 JSF 1.1 内容, 在本教程中统一采用 JSP 中的\${ } EL 表达式。

JSP 2.1 在 JSP 2.0 基础之上增加了新的 EL 表达式#{ }、延迟方法、延迟评价机制等新功能, 两者都属于 J2EE 1.4 平台, 它保证了向下兼容, 原先使用的 JSP 技术在 JSP 2.1 中都可以支持。

JSP 2.1 具有两个最重要的新特征, 其中一个是 JSTL, 另一个是 EL。鉴于 JSP 2.1 的 EL 表达式#{expr}、延迟方法、延迟评价机制在本教程中未使用, 下面集中介绍 JSP 2.1 相对于 JSP 1.2 而言增加的新功能, 主要包括下面几部分内容。

4

1.2.1 运行环境变化

1. web.xml 格式变化

JSP 1.2 可以在 Java 2 标准版 1.3 版本运行, 而 JSP 2.1 要求使用 Java 2 标准版 1.4 或更新版本, JSP 2.1 使用由 Servlet 2.4 规定的 Web 程序部署描述格式。

在 Web 程序描述文件 web.xml 中需要遵循 xml schema 格式, 在 web.xml 中将所有关于 JSP 的设置信息要放在<jsp-config>标记中。

【例 1-1】显示了 web.xml 文件的内容, 其内容格式如下。

```
1. <?xml version="1.0" encoding="ISO-8859-1"?>
2. <web-app xmlns=http://java.sun.com/xml/ns/j2ee
3.   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
4.   xsi:schemaLocation=http://java.sun.com/xml/ns/j2ee web-app
   _2_4.xsd
5.   version="2.4">
6.   .....
7.   <jsp-config>
8.     <taglib>
9.       <taglib-uri>http://www.icconcept.com/ics/sample-taglib
10.     </taglib-uri>
11.     <taglib-location>/WEB-INF/jsp/sample-taglib.tld
12.     </taglib-location>
13.   </taglib>
14.   .....
15. </jsp-property-group>
```

```
16.     <description>For config the ICW sample application
17.     </description>
18.     <display-name>JSPConfiguration</display-name>
19.     <url-pattern>/jsp/datareset.jsp</url-pattern>
20.     <el-ignored>>true</el-ignored>
21.     <page-encoding>ISO-8859-1</page-encoding>
22.     <scripting-invalid>>true</scripting-invalid>
23.     <include-prelude>/jsp/prelude.jspf</include-prelude>
24.     <include-coda>/jsp/coda.jspf</include-coda>
25. </jsp-property-group>
26. </jsp-config>
27. </web-app>
```

2. JSP 设置

如【例 1-1】中 web.xml 格式所示，在<jsp-config>标记提供了 Web 程序中 JSP 程序的相关配置信息。例如，第 7 行的<jsp-config>标记包括<taglib>和<jsp-property-group>两类子元素。其中，<taglib>定义了 Web 程序使用的 custom tag，其用法和 JSP 1.2 中的用法相同，<jsp-property-group>定义了 JSP 的一组特性，这些特性对应 JSP 的 page directive 定义的特性。

通过<jsp-property-group>可以方便地对多个具有相同属性的 JSP 统一定义。<jsp-property-group>定义一个或多个 URL 样式，在<jsp-property-group>中定义的属性会适用于匹配这些 URL 样式的所有 JSP 文件。

在<jsp-property-group>中的属性可以进行如下配置：

(1) 允许或禁止使用表达式语言 (EL)：对于对应<url-pattern>的 JSP，<jsp-property-group>可以设定是否允许使用 JSTL EL。如果<el-ignored>属性标记若被设定为 true，Web 容器会在转换 JSP 时忽略 EL 表达式，若设定为 false，JSP 中的 EL 表达式则会被处理。

(2) 允许或禁止使用 scripting：<scripting-invalid>属性用来允许或禁止使用 JSP 的脚本语言 (scripting)。若设定其值为 true，则禁止使用 scripting 元素，即在 JSP 中不能使用 scriptlet、scripting 表达式和 declaration。若存在 scriptlet、scripting 表达式和 declaration，则会发生转换错误。当设定其值为 false 时，JSP 可以像在 1.2 版本之前那样使用脚本语言。

(3) 声明 JSP 编码：通过<page-encoding>标记可设置对应<url-pattern>的 JSP 网页的编码，该属性和每个 JSP 中的 pageEncoding 属性相对应，Web 容器将根据这个属性对 JSP 内容进行编码。

(4) 对应隐含包括 (Implicit Includes)：在<jsp-property-group>中可以在对应 JSP 中加入抬头 (preludes) 和结尾 (coda)，使用<include-prelude>和<include-coda>属性可以设定 JSP 网页中包括的 preludes 和 coda 的 jspf 文件。这些文件的位置相对于当前 Web 程序的 context，当有超过一个 preludes 或 coda 元素在<jsp-property-group>中时，JSP 会按照其顺序加入到内容中。

1.2.2 引入表达式语言

JSP 2.1 的一个主要特点是它支持表达式语言 (Expression Language, EL)。JSTL 表达式语言可以使用标记格式方便地访问 JSP 的隐含对象和 JavaBeans 组件, JSTL 的核心标记提供了流程和循环控制功能。自定义标记也有自定义函数的功能, 因此基本上所有 scriptlet 能实现的功能都可以由 JSP 替代。在 JSP 2.1 中建议尽量使用 EL, 使得 JSP 的格式一致。

在 web.xml 的 <jsp-property-group>中可以控制一组 JSP 是否使用 EL, 在每个 JSP 中也可以指定是否该 JSP 使用 EL。在 page directive 中的 isELIgnored 属性用来指定是否忽略, 其格式如下。

```
<%@ page isELIgnored="true|false"%>
```

如果设定为 true, 那么 JSP 中的表达式被当成字符串处理。比如下面这个表达式 <p>\${2000 % 20}</p>在 isELIgnored="true"时输出为 \${2000 % 20}, 而 isELIgnored="false"时输出为 100。Web 容器默认 isELIgnored="false"。

虽然 JSP 2.1 可以使 JSP 中完全使用表达式语言而避免 scriptlet, 在实际编程中, 应该根据程序的功能要求和编程人员的自身条件选择合适的方式。使用表达式语言的 JSP 比较方便规整, 但是由于需要将标记进行转换, 在第一次被调用时会比较慢; 有些编程人员由于对 Java 比较了解, 因而更习惯 JSP 1.2 之前的编程方式, 因此, 在使用中应因地制宜地选择适用的编程方法。

1.2.3 SimpleTag

JSP 2.1 中加入了新的创建自定义标记的 API, javax.servlet.jsp.tagext.SimpleTag 定义了用来实现简单标记的接口。和 JSP 1.2 中的已有接口不同的是, SimpleTag 接口不使用 doStartTag()和 doEndTag()方法, 而提供了一个简单的 doTag()方法。这个方法在调用该标记时只被使用一次。而需要在一个自定义标记中实现的所有逻辑过程、循环和对标记体的评估等都在这个方法中实现。从这个方面来讲, SimpleTag 和 IterationTag 可以达到同等的作用。但 SimpleTag 的方法和处理周期要简单得多。在 SimpleTag 中还有用来设置 JSP 内容的 setJspBody()和 getJspBody()方法。Web 容器会使用 setJspBody()方法定义一个代表 JSP 内容的 JspFragment 对象。实现 SimpleTag 标记的程序可以在 doTag 方法中根据需要多次调用 getJspBody().invoke()方法以处理 JSP 内容。

【例 1-2】新创建的自定义标记 IterationSimpleTag, 其功能是根据指定的次数 (times) 进行循环并输出当前序号 (sequence)。程序的逻辑在 doTag 方法中实现, 结构简单, 具体内容如下。

```
1. package ICW.taglib;
2. import javax.servlet.jsp.JspException;
3. import javax.servlet.jsp.tagext.SimpleTagSupport;
4. import java.util.HashMap;
5. import java.io.IOException;
6. public class IterationSimpleTag extends SimpleTagSupport {
```

```
7. private int times;
8. public void setTimes(int _times){
9.     this.times=_times;
10. }
11. public void doTag() throws JspException,IOException{
12.     HashMap params=new HashMap();
13.     for(int i=0; i<times;i++){
14.         params.put("sequence",String.valueOf(i+1));
15.         getJspBody().invoke(null,params);
16.     }
17. }
18. }
```

【例 1-3】展示了【例 1-2】中的简单标记 `IterationSimpleTag` 的 TLD 文件内容，文件内容遵循 XML schema 格式，具体内容如下。

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <taglib xmlns=http://java.sun.com/xml/ns/j2ee
3.     xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
4.     xsi:schemaLocation=
5.         http://java.sun.com/xml/ns/j2ee web-jsptaglibrary_2_0.xsd
6.     version="2.1">
7. <taglib>
8.     <tlib-version>1.0</tlib-version>
9.     <short-name>Jwad book simple tag</short-name>
10.    <uri>/JwadSimpleTag</uri>
11.    <description>Simple Tag Handler</description>
12.    <tag>
13.        <name>iteration</name>
14.        <tag-class>ICW.taglib.IterationSimpleTag</tag-class>
15.        <body-content>scriptless</body-content>
16.        <description>Iteration Tag</description>
17.        <variable>
18.            <description>Current iteration number</description>
19.            <name-given>sequence</name-given>
20.        </variable>
21.        <attribute>
22.            <name>times</name>
23.            <required>true</required>
24.            <rtexprvalue>true</rtexprvalue>
25.        </attribute>
26.    </tag>
27. </taglib>
```

【例 1-4】介绍【例 1-3】中自定义标记 `IterationSimpleTag` 的具体应用，在应用中要求根据给定的 Web 请求参数“times”的值进行指定次数的循环，并且在每次循环中输出“sequence”的值。具体程序代码如下。

```
1. <%@ taglib prefix="ictag" uri="/WEB-INF/ics-jsp2.tld" %>
2. <HTML>
3. <HEAD>
4. <TITLE>Simple Tag Sample</TITLE>
5. </HEAD>
6. <BODY>
7. <CENTER>
8. <FONT COLOR='#009999' SIZE='4' face='Arial'>
9. <STRONG>Iteration Simple Tag</STRONG>
10. </FONT>
11.</CENTER>
12.<HR>
13.<c:setvar="time" value="${param.times}"/>
14.<p><B>Reminder:</B></p><br>
15.<ictag:iteration times="${times}">
16. This is the ${sequence} Of ${times} times of reminder<br>
17.</ictag:iteration>
18.</body>
19.</html>
```

1.2.4 使用 JSP fragment

JSP 2.1 中的一个主要功能是 JSP fragment，其基本特点是提供动态属性，使得 JSP 容器延迟对 JSP 标记属性的评估处理。通常，JSP 容器首先评估 JSP 标记的属性，然后在处理 JSP 标记时使用这些属性。也就是说，JSP fragment 提供的动态属性在 JSP 处理其标记体时是可以被改变的。

JSP 将提供的动态属性定义为 `javax.servlet.jsp.tagext.JspFragment` 类型，在实现标记的程序中，其处理方法类似于标记体，标记属性可以被反复评估多次，这种用法被称为 JSP fragment。JSP fragment 允许在一个 SimpleTag 处理程序中使用的自定义标记动作，如【例 1-4】所示，`getJspBody` 返回一个 JspFragment 对象并可以在 `doTag` 方法中多次使用。注意，使用 JSP fragment 的 JSP 只能有一般文本和 JSP action，不能有 scriptlet 和 scriptlet 表达式。

JSP fragment 的本质是一段可以重复使用的 JSP，一段 JSP fragment 可以被传递到另一个 JSP 中并被使用，与 JSP include 概念不同的是，JSP fragment 一般功能比较短小简单而且重复使用率高。JSP fragment 通常在 `<jsp:attribute>` 标记体内或 `<jspcbody>` 标记体内定义，每次当含有 JSP fragment 的标记被使用时，Web 容器生成一个 JSP fragment 对象，使其和当前 JSP 的 page scope 联系起来，同时和包含它的父标记建立联系。

JSP fragment 对象可以有两种调用方式，使用 Java 编写的标记处理程序，或使用标记文件 (tag file)。标记文件可使用 `<jsp:invoke>` 或 `<jsp:doBody>` 动作使用 JSP fragment。

JSP 标记文件基于 JSP 语法遵循文本文件格式实现自定义标记 (Custom Tag)，采用编程方式实现，它可以和 JSP fragment 共同工作，这也是 JSP 2.1 的一个主要新功能。

标记文件一般使用 .tag 为后缀并放在 Web 程序的 /WEB-INF 目录下。其中，一个标

记文件可以对应一个标记，可以不需 TLD 文件定义该标记的方法，这样的优点是：即使编程人员不熟悉 Java，也可以使用 JSP 语法定义自己的标记。

【例 1-5】该案例中 taskstatus.jsp 使用了两个 JSP fragment。这个 JSP 的功能是显示一组 Task 的名称和完成日期，它通过<jsp:attribute name="...">定义了两段 JSP fragment，名称分别为 onSchedule 和 delayed。在<jsp:attribute>标记内的 JSP 就是 JSP fragment，而<jsp:attribute>标记被一个<ietag: listTasks>包围。其中，<ietag: listTasks>标记是一个通过标记文件定义的自定义标记，它使用到前面定义的两个 JSP fragment，其定义文件在/WEB-INF/tags 目录下，文件名称和标记名称一致为"listTasks.tag"。

```

1. <%@ taglib prefix="ictag" tagdir="/WEB-INF/tags" %>
2. <HTML>
3. <HEAD>
4.   <TITLE>JSP Fragment Sample</TITLE>
5. </HEAD>
6. <BODY>
7. <CENTER>
8. <FONT COLOR='#009999' SIZE='4' face='Arial'>
9.   <STRONG>JSP Fragment Sample Using Tag Files</STRONG>
10.</FONT>
11.</CENTER>
12.<HR>
13.<h2>Tasks</h2>
14.<ietag:listTasks>
15.<jsp:attribute name="onSchedule">
16.  <td>Name:${name}<br/></td>
17.  <td>Date:${date}</td>
18.</jsp:attribute>
19.<jsp:attribute name="delayed">
20.<td>Name:${name}<br/></td>
21.<td><font color="red">Plan: ${pianDate}</font><br/>
22.   <b>Actural:${actDate}</b>
23.</td>
24.</jsp:attribute>
25.</ictag:listTasks>
26.</BODY>
27.</HTML>

```

JSP 2.1 还有一些其他特性变化，比如严格修正了 I18N 的语法规则，改进 JSP 对应 XML 语法从而允许使用 namespaces 等，本教程不再赘述。

思考练习 1-2

1. JSP 2.1 的一个显著特点是_____。
2. JSP 2.1 相对于 JSP 1.2 的两个显著特点是_____和_____。
3. JSP 2.1 中标记文件的后缀名为_____。
4. EL 的全称是_____，含义为_____。

1.3 【案例 1】JSP 2.1 开发环境的配置

案例描述

要使用 JSP 技术开发 Web 应用，需要掌握相应的开发工具，需要搭建好开发环境，这要实现如下功能。

- ☆ 下载和安装 JDK。
 - ☆ 下载和安装 Tomcat 应用服务器。
 - ☆ 下载 JSTL，进行安装和配置。
- 通过【案例 1】的学习，培养实践技能如下。
- ★ 熟悉 JDK 的下载、安装过程、环境变量的设置。
 - ★ 熟悉 Tomcat 的下载，安装及配置过程。
 - ★ 熟悉 JSTL 的下载，安装及配置过程。
 - ★ 了解数据库知识、Microsoft SQL Server 数据库。

设计过程

10

1. 下载和安装 JDK

JDK (Java SE Development Kit) 是 Sun 提供的 Java 开发工具和运行环境。目前的版本是 JDK 6 Update 21，可以从 Sun 公司网站免费下载。下载网址为：<http://java.sun.com/javase/downloads/index.jsp>。

(1) 在浏览器中输入该网址，打开如图 1-3-1 所示页面。

(2) 在图 1-3-1 中定位至如图 1-3-2 所示的位置，单击“Download JDK”按钮，显示页面如图 1-3-3 所示。



图 1-3-1 JDK 官方下载页面

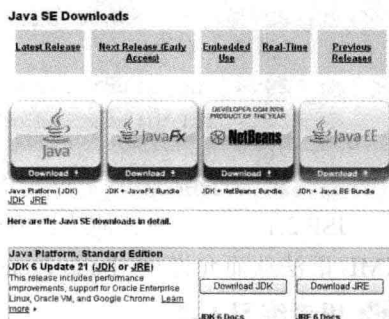


图 1-3-2 选择“Download JDK”下载

(3) 在图 1-3-3 中，单击“Download”按钮，显示如图 1-3-4 所示页面。

(4) 在图 1-3-4 中，选择 Platform (运行平台) 为 Windows，单击“Continue”按钮，显示下载页面，如图 1-3-5 所示。