



JSP 程序设计

慕课版

明日科技·出品 ◎ 贾志城 王云 主编 ◎ 马俊 于晓强 副主编

课程简介 第一本程序设计互联网教材，买书送名师，面对面授课
体系完整的 60 学时慕课教学视频同步讲解，即学即练的在线测试
书中 187 个微课视频，扫一扫二维码，随时随地学习
300 余道习题和自测题，附赠 PPT 课件、案例源代码、自测试卷、
拓展案例等资源

适用对象 学习 JSP 的第一本书





JSP

程序设计

慕课版

明日科技·出品

◎ 贾志城 王云 主编 ◎ 马俊 于晓强 副主编



人民邮电出版社
北京

图书在版编目 (C I P) 数据

JSP程序设计 : 慕课版 / 贾志城, 王云主编. — 北京 : 人民邮电出版社, 2016. 4
ISBN 978-7-115-41763-3

I. ①J… II. ①贾… ②王… III. ①JAVA语言—网页制作工具—高等学校—教材 IV. ①TP312②TP393.092

中国版本图书馆CIP数据核字(2016)第028202号

内 容 提 要

本书系统地介绍了有关 JSP 开发所涉及的各类知识。全书共分 13 章, 内容包括 JSP 概述、JSP 开发基础、JSP 语法、JSP 内置对象、JavaBean 技术、Servlet 技术、JSP 实用组件、JSP 数据库应用开发、JSP 与 Ajax 及 JSP 高级技术, 并通过 JSP 综合开发实例——清爽夏日九宫格日记网, 介绍了 JSP 应用的开发流程和相关技术的综合应用。全书最后提供了两个课程设计方案, 在线投票系统和 Ajax 聊天室, 供学生综合实践使用。

本书为慕课版教材, 各章节主要内容配备了以二维码为载体的微课, 并在人邮学院 (www.rymooc.com) 平台上提供了慕课。此外, 本书还提供了课程资源包, 资源包中提供有本书所有实例、上机指导、综合案例和课程设计的源代码, 制作精良的电子课件 PPT, 自测试卷等内容。资源包也可在人邮学院上下载。其中, 源代码全部经过精心测试, 能够在 Windows 7、Windows 8、Windows 10 系统下编译和运行。

-
- ◆ 主 编 贾志城 王 云
副 主 编 马 俊 于晓强
责任编辑 刘 博
责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京中新伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 21.25 2016 年 4 月第 1 版
字数: 632 千字 2016 年 4 月北京第 1 次印刷
-

定价: 49.80 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广字第 8052 号

前言 Foreword

为了让读者能够快速且牢固地掌握JSP开发技术，人民邮电出版社充分发挥在线教育方面的技术优势、内容优势、人才优势，潜心研究，为读者提供一种“纸质图书+在线课程”相配套，全方位学习JSP开发的解决方案。读者可根据个人需求，利用图书和“人邮学院”平台上的在线课程进行系统化、移动化的学习，以便快速全面地掌握JSP开发技术。

一、如何学习慕课版课程

本课程依托人民邮电出版社自主开发的在线教育慕课平台——人邮学院（www.rymooc.com），该平台为学习者提供优质、海量的课程，课程结构严谨，用户可以根据自身的学习程度，自主安排学习进度，并且平台具有完备的在线“学习、笔记、讨论、测验”功能。人邮学院为每一位学习者，提供完善的一站式学习服务（见图1）。



图1 人邮学院首页

为了使读者更好地完成慕课的学习，现将本课程的使用方法介绍如下。

1. 用户购买本书后，找到粘贴在书封底上的刮刮卡，刮开，获得激活码（见图2）。
2. 登录人邮学院网站（www.rymooc.com），或扫描封面上的二维码，使用手机号码完成网站注册。



图2 激活码



图3 注册人邮学院网站

3. 注册完成后，返回网站首页，单击页面右上角的“学习卡”选项（见图4），进入“学习卡”页面（见图5），输入激活码，即可获得该慕课课程的学习权限。



图4 单击“学习卡”选项



图5 在“学习卡”页面输入激活码

4. 输入激活码后，即可获得该课程的学习权限。可随时随地使用计算机、平板电脑、手机学习本课程的任意章节，根据自身情况自主安排学习进度（见图6）。
5. 在学习慕课课程的同时，阅读本书中相关章节的内容，巩固所学知识。本书既可与慕课课程配合使用，也可单独使用，书中主要章节均放置了二维码，用户扫描二维码即可在手机上观看相应章节的视频讲解。
6. 学完一章内容后，可通过精心设计的在线测试题，查看知识掌握程度（见图7）。



图6 课时列表



图7 在线测试题

7. 如果对所学内容有疑问，还可到讨论区提问，除了有大牛导师答疑解惑以外，同学之间也可互相交流学习心得（见图8）。
8. 书中配套的PPT、源代码等教学资源，用户也可在该课程的首页找到相应的下载链接（见图9）。

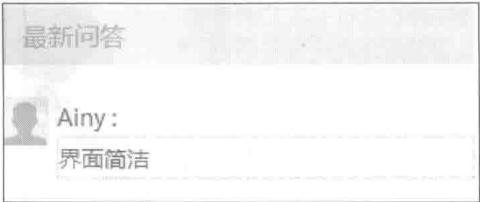


图8 讨论区



图9 配套资源

关于人邮学院平台使用的任何疑问，可登录人邮学院咨询在线客服，或致电：010-81055236。

二、本书特点

JSP (Java Server Page) 是由Sun公司在Java语言上开发出来的一种动态网页制作技术，它是Java开

发阵营中最具代表性的解决方案。JSP不仅拥有与Java一样的面向对象、便利、跨平台等优点和特性，还拥有Java Servlet的稳定性，并且可以使用Servlet提供的API、Java Bean及Web开发框架技术，使页面代码与后台处理代码分离，从而提高工作效率。在目前流行的Web程序开发技术中，JSP是比较热门的一种，它依靠Java语言的稳定、安全、可移植性好的优点，成为大、中型网站开发的首选。

本书通过通俗易懂的语言和实用生动的例子，系统地介绍了JSP的基本常识、开发环境与开发工具、Java和JavaScript语言基础、JSP基本语法、内置对象、JavaBean技术、Servlet技术、实用组件、数据库应用开发和高级程序设计等技术，并且在主要章节的后面还提供了习题及上机指导，方便读者及时验证自己的学习效果。最后，通过一个综合实例及两个课程设计来使读者快速掌握JSP程序的开发过程。

本书作为教材使用时，课堂教学建议32~42学时，实验教学16~32学时。各章主要内容和学时建议分配如下，老师可以根据实际教学情况进行调整。

章	主 要 内 容	课堂学时	实验学时
第1章	JSP概述，包括JSP技术概述、JSP技术特征、JSP的处理过程、JSP开发环境的搭建、JSP开发工具介绍和JSP程序开发模式	2	1
第2章	Java语言基础和JavaScript脚本语言（选讲）	6	4
第3章	JSP技术的基本语法，主要包括JSP中的指令标识、脚本标识、JSP注释和动作标识	3	3
第4章	JSP的内置对象，包括request对象、response对象、session对象、application对象、out对象及其他内置对象的基本应用	4	2
第5章	JSP程序开发中的JavaBean技术，主要包括JavaBean的相关概念、JavaBean中的属性、JavaBean的创建以及JavaBean的应用	4	2
第6章	Servlet的基础知识及Servlet的应用，主要包括Servlet基础、Servlet API编程常用接口和类、Servlet的开发以及Server过滤器	4	2
第7章	JSP的实用组件，主要包括对文件进行操作的Commons-FileUpload组件、发送E-mail的Java Mail组件、生成动态图表的JFreeChart组件和生成JSP报表的iText组件	3	2
第8章	在JSP中应用数据库开发技术，包括常用数据库管理系统、JDBC技术概述及常用接口、连接数据库、数据库操作技术及连接池技术等	4	2
第9章	JSP与Ajax，主要包括了解Ajax、使用XMLHttpRequest对象、传统Ajax的工作流程、jQuery实现Ajax、需要注意的几个问题	2	
第10章	JSP高级技术，主要包括EL表达式，JSTL标准标签库的应用，自定义标签库的开发以及JSP框架技术	2	
第11章	JSP综合开发实例——清爽夏日九宫格日记网	4	2
第12章	课程设计——在线投票系统	2	可选
第13章	课程设计——Ajax聊天室	2	可选

本书由明日科技出品，贾志城、王云任主编，马俊、于晓强任副主编。其中贾志城编写了第1~5章，王云编写了第6~8章，马俊编写了第9~11章，于晓强编写了第12~13章。

编 者
2016年1月

目录

Contents

第1章 JSP概述	1
1.1 JSP技术概述	2
1.2 JSP技术特征	3
1.3 JSP的处理过程	4
1.4 JSP与其他服务器端脚本语言的比较	5
1.5 JSP开发环境搭建	6
1.5.1 JSP的运行环境	6
1.5.2 JDK的安装与配置	6
1.5.3 Tomcat的安装与启动	9
1.6 JSP开发工具	10
1.6.1 Eclipse的安装与启动	11
1.6.2 Eclipse的使用	11
1.7 JSP程序开发模式	16
1.8 第一个JSP应用	17
1.9 小结	19
习题	19
上机指导	19
第2章 JSP开发基础	20
2.1 Java语言基础	21
2.1.1 基本数据类型及基本数据类型间的转换	21
2.1.2 变量与常量	22
2.1.3 运算符的应用	23
 实例：应用条件运算符输出库存信息	
2.1.4 流程控制语句	26
 实例：if...else语句示例	
 实例：应用switch语句，根据输入的星期数，输出相应的提示信息	

 实例：分别利用for、while和do...while循环语句计算1到100之间所有整数和	
2.1.5 字符串处理	31
 实例：字符串应用实例	
2.1.6 数组的创建与应用	33
2.1.7 面向对象程序设计	34
 实例：在类中声明两个成员方法	
 实例：在类中声明3个成员变量，并且在其成员方法中声明两个局部变量	
 实例：对象的使用方法	
2.1.8 集合类的应用	39
 实例：使用ArrayList集合存储数据	
 实例：实现创建空的Vector对象，并向其添加元素，然后输出所有元素	
2.1.9 异常处理语句	40
2.2 JavaScript脚本语言	41
2.2.1 JavaScript脚本语言概述	42
2.2.2 在JSP中引入JavaScript	42
2.2.3 JavaScript的数据类型与运算符	42
2.2.4 JavaScript的流程控制语句	44
 实例：利用while循环语句将数字7格式化为00007	
 实例：利用do...while循环语句将数字7格式化为00007	
 实例：利用for循环语句将数字7格式化为00007	
2.2.5 函数的定义和调用	46
2.2.6 事件	47
2.2.7 JavaScript常用对象的应用	48
 实例：在新窗口的状态栏中显示当前年份	
2.3 小结	50

习题	51
上机指导	51
第3章 JSP语法	52
3.1 了解JSP的基本构成	53
3.2 JSP的指令标识	54
3.2.1 使用page指令	54
实例：设置错误提示页面内容	
3.2.2 使用include指令	56
3.2.3 使用taglib指令	57
3.3 JSP的脚本标识	57
3.3.1 JSP表达式 (Expression)	57
3.3.2 声明标识 (Declaration)	58
实例：简单的网站计数器	
3.3.3 脚本程序 (Scriptlet)	59
实例：在JSP中实现选择输出脚本程序	
3.4 JSP的注释	60
3.4.1 HTML中的注释	60
实例：HTML注释的应用	
3.4.2 带有JSP表达式的注释	60
实例：带有JSP表达式注释的应用	
3.4.3 隐藏注释	61
实例：隐藏注释的应用	
3.4.4 脚本程序 (Scriptlet) 中的注释	61
实例：单行注释的应用	
实例：多行注释的应用	
3.5 动作标识	63
3.5.1 <jsp:include>	63
实例：通过include指令和动作标识包含文件	
3.5.2 <jsp:forward>	65
3.5.3 <jsp:useBean>	66
3.5.4 <jsp:setProperty>	68
实例：<jsp:setProperty>标识的使用	
3.5.5 <jsp:getProperty>	71
实例：利用<jsp:getProperty>标签输出JavaBean中的属性	

3.5.6 <jsp:fallback>	73
3.5.7 <jsp:plugin>	73
实例：codebase属性的使用	
3.5.8 <jsp:param>子标识	76
实例：<jsp:param>子标签的使用	
3.6 小结	76
习题	77
上机指导	77
第4章 JSP内置对象	78
4.1 JSP内置对象概述	79
4.2 request对象	80
4.2.1 访问请求参数	80
实例：在login.jsp页面中通过表单向login_deal.jsp页面提交数据，在login_deal.jsp获取提交数据并输出	
4.2.2 在作用域中管理属性	81
实例：使用request对象的setAttribute()方法设置数据，然后在请求转发后取得设置的数据	
4.2.3 获取Cookie	81
实例：使用request对象的addCookie()方法实现记录本次及上一次访问网页的时间	
4.2.4 获取客户信息	82
实例：获取客户信息示例	
4.2.5 访问安全信息	83
4.2.6 访问国际化信息	83
4.3 response对象	84
4.3.1 重定向网页	84
实例：重定向网页示例	
4.3.2 设置HTTP响应报头	85
实例：将JSP页面保存为word文档	
4.3.3 缓冲区配置	86
实例：输出缓冲区的大小并测试强制将缓冲区的内容发送给客户	
4.4 session对象	86
4.4.1 创建及获取客户的会话	87
实例：创建并获取客户会话	
4.4.2 从会话中移除指定的对象	87

 实例：通过setAttribute()方法将数据保存在session中，然后通过removeAttribute()方法移除指定对象

4.4.3 销毁session 88

4.4.4 会话超时的管理 88

4.5 application对象 88

4.5.1 访问应用程序初始化参数 88

 实例：访问程序初始化参数

4.5.2 管理应用程序环境属性 89

 实例：通过application对象中的setAttribute()和getAttribute()方法实现网页计数器

4.6 out对象 90

4.6.1 管理响应缓冲 90

4.6.2 向客户端输出数据 91

4.7 其他内置对象 91

4.7.1 获取会话范围的pageContext对象 91

4.7.2 读取web.xml配置信息的config对象 91

4.7.3 应答或请求的page对象 92

4.7.4 获取异常信息的exception对象 92

4.8 小结 93

习题 93

上机指导 93

第5章 JavaBean技术 94

5.1 JavaBean概述 95

5.1.1 JavaBean技术介绍 95

5.1.2 JavaBean的种类 95

 实例：值JavaBean示例

 实例：工具JavaBean示例

5.1.3 JavaBean规范 96

 实例：JavaBean规范示例

5.2 JavaBean中的属性 97

5.2.1 简单属性 (Simple) 97

 实例：定义简单属性，并定义相应的set×××()与get×××()方法进行访问

5.2.2 索引属性 (Indexed) 98

 实例：定义索引属性，并定义相应的set×××()与get×××()方法进行访问

5.3 JavaBean的应用 98

5.3.1 创建JavaBean 99

 实例：在Eclipse下创建JavaBean

5.3.2 在JSP页面中应用JavaBean 100

 实例：获取用户留言信息

5.4 JavaBean的应用实例 104

5.4.1 应用JavaBean解决中文乱码 104

 实例：应用JavaBean解决中文乱码

5.4.2 应用JavaBean实现购物车 107

 实例：应用JavaBean实现购物车

5.5 小结 114

习题 115

上机指导 115

第6章 Servlet技术 116

6.1 Servlet基础 117

6.1.1 Servlet技术简介 117

6.1.2 Servlet技术功能 117

6.1.3 Servlet技术特点 117

6.1.4 Servlet的生命周期 118

6.1.5 Servlet与JSP的区别 119

6.1.6 Servlet的代码结构 119

6.2 Servlet API编程常用接口和类 120

6.2.1 Servlet接口 120

6.2.2 HttpServlet类 120

6.2.3 ServletConfig接口 121

6.2.4 HttpServletRequest接口 121

6.2.5 HttpServletResponse接口 122

6.2.6 GenericServlet类 123

6.3 Servlet开发 123

6.3.1 Servlet的创建 123

6.3.2	Servlet的配置	125
	 实例：通过Servlet向浏览器中输出文本信息	
6.4	Servlet过滤器	128
6.4.1	什么是过滤器	128
6.4.2	过滤器核心对象	129
6.4.3	过滤器创建与配置	130
	 实例：创建名称为MyFilter的过滤器对象	
	 实例：通过过滤器实现网站访问计数器	
6.4.4	字符编码过滤器	133
	 实例：添加并显示图书信息	
6.5	Servlet监听器	137
6.5.1	Servlet监听器简介	137
6.5.2	Servlet监听器的工作原理	137
6.5.3	监听Servlet上下文	137
6.5.4	监听HTTP会话	138
6.5.5	监听Servlet请求	139
6.5.6	使用监听器查看在线用户	139
	 实例：通过监听器查看用户在线情况	
6.6	Servlet的应用实例	141
6.6.1	应用Servlet实现留言板	141
	 实例：应用Servlet实现留言板	
6.6.2	应用Servlet实现购物车	145
	 实例：应用Servlet实现购物车	
6.7	小结	153
	习题	154
	上机指导	154

第7章 JSP实用组件 155

7.1	JSP文件操作	156
7.1.1	添加表单及表单元素	156
7.1.2	创建上传对象	156
7.1.3	解析上传请求	156
	 实例：应用Commons-FileUpload组件将文件上传到服务器	
7.2	发送E-mail	159
7.2.1	Java Mail组件简介	159
7.2.2	Java Mail核心类简介	159

7.2.3	搭建Java Mail的开发环境	163
7.2.4	在JSP中应用Java Mail组件发送E-mail	164
	 实例：发送普通文本格式的E-mail	
7.3	JSP动态图表	166
7.3.1	JFreeChart的下载与使用	166
7.3.2	JFreeChart的核心类	167
7.3.3	利用JFreeChart生成动态图表	167
	 实例：生成论坛版块人气指数排行的柱形图	
	 实例：生成论坛版块人气指数排行的饼形图	
7.4	JSP报表	169
7.4.1	iText组件简介	169
7.4.2	iText组件的下载与配置	169
7.4.3	应用iText组件生成JSP报表	170
	 实例：在JSP页面中输出PDF文档	
	 实例：创建表格	
	 实例：图像处理	
7.5	小结	177
	习题	177
	上机指导	177

第8章 JSP数据库应用开发 178

8.1	数据库管理系统	179
8.1.1	SQL Server 2008数据库	179
8.1.2	MySQL数据库	181
8.1.3	Oracle数据库	181
8.1.4	Access数据库	181
8.2	JDBC概述	181
8.2.1	JDBC技术介绍	181
8.2.2	JDBC驱动程序	182
8.3	JDBC中的常用接口	183
8.3.1	驱动程序接口Driver	183
8.3.2	驱动程序管理器	

DriverManager	183	9.1.1 什么是Ajax	199
8.3.3 数据库连接接口		9.1.2 Ajax开发模式与传统开发模式的比较	199
Connection	183	9.2 使用XMLHttpRequest对象	200
8.3.4 执行SQL语句接口		9.2.1 初始化XMLHttpRequest对象	200
Statement	184	9.2.2 XMLHttpRequest对象的常用方法	201
8.3.5 执行动态SQL语句接口		9.2.3 XMLHttpRequest对象的常用属性	202
PreparedStatement	185	9.3 传统Ajax的工作流程	203
8.3.6 执行存储过程接口		9.3.1 发送请求	203
CallableStatement	185	9.3.2 处理服务器响应	205
8.3.7 访问结果集接口ResultSet	186	9.3.3 一个完整的实例——检测用户名是否唯一	206
8.4 JDBC访问数据库过程	187	 实例：编写一个会员注册页面，并应用Ajax实现检测用户名是否唯一的功能	
8.5 典型JSP数据库连接	188	9.4 jQuery实现Ajax	208
8.5.1 SQL Server 2008数据库的连接	188	9.4.1 jQuery简介	208
 实例：在JSP中连接SQL Server 2008数据库		9.4.2 我的第一个jQuery脚本	209
8.5.2 Access数据库的连接	189	 实例：应用jQuery弹出一个提示对话框	
 实例：在JSP中连接Access数据库		9.4.3 应用load()方法发送请求	210
8.5.3 MySQL数据库的连接	190	 实例：显示实时走动的时间	
 实例：在JSP中连接MySQL数据库		9.4.4 发送GET和POST请求	211
8.6 数据库操作技术	190	 实例：采用jQuery的get()方法实现例9.1	
8.6.1 查询操作	191	 实例：聊天室中实时显示聊天内容	
 实例：按照name查询用户信息		9.4.5 服务器返回的数据格式	215
8.6.2 添加操作	192	 实例：使用JSON格式返回聊天内容	
8.6.3 修改操作	192	9.4.6 使用\$.ajax()方法	219
8.6.4 删除操作	193	9.5 Ajax开发需要注意的几个问题	220
8.7 连接池技术	194	9.5.1 安全问题	220
8.7.1 连接池简介	194	9.5.2 性能问题	221
8.7.2 在Tomcat中配置连接池	195	9.5.3 浏览器兼容性问题	221
8.7.3 使用连接池技术访问数据库	195	9.5.4 中文编码问题	221
 实例：获取用户信息表中的所有数据		9.6 小结	222
8.8 小结	197		
习题	197		
上机指导	197		
第9章 JSP与Ajax	198		
9.1 了解Ajax	199		

习题	222
上机指导	222
第10章 JSP高级技术	223
10.1 EL表达式	224
10.1.1 表达式语言	224
10.1.2 EL表达式的简单使用	224
10.1.3 EL表达式的语法	224
10.1.4 EL表达式的运算符	225
10.1.5 EL表达式中的隐含对象	226
10.1.6 EL表达式中的保留字	226
10.2 JSTL标准标签库	227
10.2.1 表达式标签	230
 实例：测试<c:out>标签的escapeXml属性及通过两种语法格式设置default属性时的显示结果	
 实例：应用<c:set>标签定义不同范围内的变量，并通过EL进行输出	
 实例：应用<c:set>标签定义一个page范围内的变量，然后应用通过EL输出该变量，再应用<c:remove>标签移除该变量，最后再应用EL输出该变量	
10.2.2 条件标签	232
 实例：应用<c:if>标签判断用户名是否为空，如果为空则显示一个用于输入用户名的文本框及“提交”按钮	
 实例：应用<c:choose>标签、<c:when>标签和<c:otherwise>标签根据当前时间显示不同的问候	
10.2.3 循环标签	235
 实例：应用<c:forEach>标签循环输出List集合中的内容，并通过<c:forEach>标签循环输出字符串“编程词典”6次	
 实例：应用<c:forEachTokens>标签分割字符串并显示	
10.2.4 URL操作标签	236
10.3 自定义标签库的开发	238
10.3.1 自定义标签的定义格式	238
10.3.2 自定义标签的构成	238
10.3.3 在JSP文件中引用自定义标签	240

 实例：创建用于显示当前系统日期的自定义标签	
10.4 JSP框架技术	242
10.4.1 Struts 2框架	242
10.4.2 Spring框架	243
10.4.3 Hibernate技术	244
10.5 小结	244
习题	245
上机指导	245
第11章 JSP综合开发实例——清爽夏日九宫格日记网	246
11.1 项目设计思路	247
11.1.1 功能阐述	247
11.1.2 系统预览	247
11.1.3 功能结构	248
11.1.4 文件夹组织结构	249
11.2 数据库设计	249
11.2.1 数据库设计	249
11.2.2 数据表设计	250
11.3 公共模块设计	250
11.3.1 编写数据库连接及操作的类	250
11.3.2 编写保存分页代码的JavaBean	253
11.3.3 配置解决中文乱码的过滤器	255
11.3.4 编写实体类	256
11.4 主界面设计	257
11.4.1 主界面概述	257
11.4.2 让采用DIV+CSS布局的页面内容居中	257
11.4.3 主界面的实现过程	258
11.5 用户模块设计	259
11.5.1 用户模块概述	259

11.5.2	实现Ajax重构	259
11.5.3	用户注册的实现过程	261
11.5.4	用户登录的实现过程	270
11.5.5	退出登录的实现过程	274
11.5.6	忘记密码的实现过程	274
11.6	显示九宫格日记列表模块设计	276
11.6.1	显示九宫格日记列表	
	概述	276
11.6.2	展开和收缩图片	277
11.6.3	查看日记原图	279
11.6.4	对日记图片进行左转和	
	右转	279
11.6.5	显示全部九宫格日记的	
	实现过程	282
11.6.6	我的日记的实现过程	285
11.6.7	删除我的日记的实现过程	286
11.7	写九宫格日记模块设计	287
11.7.1	写九宫格日记概述	287
11.7.2	应用jQuery让PNG图片在IE 6	
	下背景透明	287
11.7.3	填写日记信息的实现过程	288
11.7.4	预览生成的日记图片的	
	实现过程	292
11.7.5	保存日记图片的实现过程	296
11.8	项目发布	298
11.9	小结	299

第12章 课程设计一——在线投票系统 300

12.1	课程设计的目的	301
------	---------	-----

12.2	设计思路	301
12.2.1	显示投票选项的设计思路	301
12.2.2	参与投票的设计思路	301
12.2.3	显示投票结果的设计思路	302
12.3	设计过程	302
12.3.1	数据表的设计	302
12.3.2	值JavaBean的设计	303
12.3.3	数据库操作类的编写	304
12.3.4	工具类的编写	309
12.3.5	显示投票选项的设计	309
12.3.6	参与投票的设计	311
12.3.7	查看结果的设计	313
12.4	小结	315

第13章 课程设计二——Ajax聊天室 316

13.1	课程设计的目的	317
13.2	设计思路	317
13.3	设计过程	317
13.3.1	用户JavaBean的编写	317
13.3.2	登录页面的设计	318
13.3.3	聊天室主页面设计	319
13.3.4	在线人员列表的设计	319
13.3.5	用户发言的设计	321
13.3.6	显示聊天内容的设计	322
13.3.7	退出聊天室的设计	324
13.4	小结	325

参考文献 326

第1章

JSP概述

本章要点

- JSP概念 ■
- JSP特点 ■
- JSP处理过程 ■
- JSP开发环境搭建 ■
- JSP开发工具 ■

■ 本章介绍JSP技术的相关概念以及如何开发JSP程序，主要内容包括JSP技术的概述、JSP技术特征、JSP的处理过程、JSP开发环境的搭建、JSP开发工具介绍、JSP程序开发模式等。通过本章的学习，读者应了解什么是JSP、JSP技术的特征和处理过程，并掌握JDK的安装与配置、Tomcat的安装与配置、Eclipse开发工具的使用等。尤其要深刻理解在JSP程序开发模式中，JSP+Servlet+JavaBean开发模式的设计及模式中各技术所扮演的角色。

1.1 JSP技术概述

在了解JSP技术之前，先向读者介绍与JSP技术相关的一些概念，这样有助于读者阅读后面的内容。

1. Java语言

Java语言是由Sun公司于1995年推出的编程语言，一经推出，就赢得了业界的一致好评，并受到了广泛关注。Java语言适用于Internet环境，目前已成为开发Internet应用的主要语言之一。它具有简单、面向对象、可移植性、分布性、解释器通用性、稳健、多线程、安全和高性能等优点。其中最重要的就是实现了跨平台运行，这使得应用Java开发的程序可以方便地移植到不同的操作系统中运行。

Java语言是完全面向对象的编程语言，它的语法规则和C++类似，但Java语言对C++进行了简化和提高。例如，C++中的指针和多重继承通常会使程序变得复杂，而Java语言通过接口取代了多重继承，并取消了指针、内存的申请和释放等影响系统安全的部分。

在Java语言中，最小的单位是类，不允许在类外面定义变量和方法，所以就不存在所谓的“全局变量”这一概念。在Java类中定义的变量和方法分别称为成员变量和成员方法，其中成员变量也叫作类的属性。在定义这些类的成员时，需要通过权限修饰符来声明它们的使用范围。

Java语言编写的程序应被保存为后缀名为.java的文件，然后编译成后缀名为.class的字节码文件，最终通过执行该字节码文件执行Java程序。

2. Servlet技术

Servlet是在JSP之前就存在的运行在服务端的一种Java技术，它是用Java语言编写的服务器端程序，Java语言能够实现的功能，Servlet基本上都可以实现（除图形界面外）。Servlet主要用于处理Http请求，并将处理的结果传递给浏览器生成动态Web页面。Servlet具有可移植（可在多种系统平台和服务器平台下运行）、功能强大、安全、可扩展和灵活等优点。

在JSP中用到的Servlet通常都继承自`javax.servlet.http.HttpServlet`类，在该类中实现了用来处理Http请求的大部分功能。

JSP是在Servlet的基础上开发的一种新的技术，所以JSP与Servlet有着密不可分的关系。JSP页面在执行过程中会被转换为Servlet，然后由服务器执行该Servlet。

关于Servlet技术的相关介绍请参看本书的第6章。

3. JavaBean技术

JavaBean是根据特殊的规范编写的普通的Java类，可称它们为“独立的组件”。每一个JavaBean实现一个特定的功能，通过合理地组织具有不同功能的JavaBean，可以快速地生成一个全新的应用程序。如果将这个应用程序比作一辆汽车，那么程序中的JavaBean就好比组成这辆汽车的不同零件。对于程序开发人员来说，JavaBean的最大优点就是充分提高了代码的可重用性，并且对程序的后期维护和扩展起到了积极的作用。

JavaBean可按功能划分为可视化和不可可视化两种。可视化JavaBean主要应用在图形界面编程的领域中，在JSP中通常应用的是不可可视化JavaBean。应用该种JavaBean可用来封装各种业务逻辑，例如连接数据库、获取当前时间等。这样，当在开发程序的过程中需要连接数据库或实现其他功能时，就可直接在JSP页面或Servlet中调用实现该功能的JavaBean来实现。



JSP技术
概述

通过应用JavaBean,可以很好地将业务逻辑和前台显示代码分离,这大大提高了代码的可读性和易维护性。

关于JavaBean技术的相关介绍请参看本书的第5章。

4. JSP技术

Java Server Pages(JSP)是由Sun公司倡导,与多个公司共同建立的一种技术标准,它建立在Servlet之上。应用JSP,程序员或非程序员可以高效率地创建Web应用程序,并使得开发的Web应用程序具有安全性高、跨平台等优点。

JSP是运行在服务器端的脚本语言之一,与其他服务器端的脚本语言一样,是用来开发动态网页的一种技术。

JSP页面由传统的HTML代码和嵌入到其中的Java代码组成。当用户请求一个JSP页面时,服务器会执行这些Java代码,然后将结果与页面中的静态部分相结合返回给客户端浏览器。JSP页面中还包含了各种特殊的JSP元素,通过这些元素可以访问其他的动态内容并将它们嵌入到页面中,例如访问JavaBean组件的<jsp:useBean>动作元素。程序员还可以通过编写自己的元素来实现特定的功能,开发出更为强大的Web应用程序。

JSP是在Servlet的基础上开发的技术,它继承了Java Servlet的各项优秀功能。而Java Servlet是作为Java的一种解决方案,在制作网页的过程中,它继承了Java的所有特性。因此JSP同样继承了Java技术的简单、便利、面向对象、跨平台和安全可靠等优点,比起其他服务器脚本语言,JSP更加简单、迅速和便利。在JSP中利用JavaBean和JSP元素,可以有效地将静态的HTML代码和动态数据区分开来,给程序的修改和扩展带来了很大方便。

5. JSP在JavaWeb开发中的地位

Web应用程序大体上可以分为两种,即静态网站和动态网站。早期的Web应用主要是静态页面的浏览,即静态网站。这些网站使用HTML语言来编写,放在Web服务器上,用户使用浏览器通过HTTP协议请求服务器上的Web页面;服务器上的Web服务器将接收到的用户请求处理后,再发送给客户端浏览器,显示给用户。

在开发Web应用程序时,通常需要应用客户端和服务器两方面的技术。其中,客户端应用的技术主要用于展现信息内容,而服务器应用的技术,则主要用于进行业务逻辑的处理和数据库的交互。想要开发动态网站,就需要用到服务器端技术。而JSP技术就是服务器端应用的技术,JSP页面中的HTML代码用来显示静态内容部分,嵌入到页面中的Java代码与JSP标记用来生成动态的内容部分。JSP允许程序员编写自己的标签库来完成应用程序的特定要求。JSP可以被预编译,提高了程序的运行速度。另外,JSP开发的应用程序经过一次编译后,便可以随时随地运行。所以在绝大部分系统平台中,代码无需作修改便可支持JSP在任何服务器中运行。

1.2 JSP技术特征

本节将介绍JSP在开发Web应用程序时的一些特点,如跨平台、分离静态内容和动态内容、可重复使用的组件、沿用了Java Servlet的所有功能以及能预编译等。

1. 跨平台

JSP是以Java为基础开发的,所以它不仅可以沿用Java强大的API功能,而且不管是在何种平台下,只要服务器支持JSP,就可以运行使用JSP开发的Web应用程序,体现了它的跨平台、跨服务器特点。例如在Windows NT下的IIS通过JRUN或ServletExec插件就能支持JSP。如今最流行的Web服务器Apache同样能够支持JSP,而且Apache支持多种平台,从而使得JSP可以在多个平台上运行。

在数据库操作中,因为JDBC同样是独立于平台的,所以在JSP中使用Java API提供的JDBC来连接数据



JSP技术
特征

库时,就不用担心平台变更时的代码移植问题。正是因为Java的这种特征,使得应用JSP开发的Web应用程序能够很简单地运用到不同的平台上。

2. 分离静态内容和动态内容

在前面提到的Java Servlet,对于开发Web应用程序而言是一种很好的技术。但同时面临着一个问题:所有的内容必须在Java代码中来完成,包括HTML代码同样要嵌入到程序代码中来生成静态的内容。即使因HTML代码出现小问题,也需要有熟悉Java Servlet的程序员来解决。

JSP弥补了Java Servlet在工作中的不足。使用JSP,程序员可以使用HTML或XML标记来设计和格式化静态的内容部分,使用JSP标记及JavaBean组件或者小脚本程序来制作动态内容部分。服务器将执行JSP标记和小脚本程序,并将结果与页面中的静态部分结合后以HTML页面的形式发送给客户端浏览器。程序员可以将一些业务逻辑封装到JavaBean组件中,Web页面的设计人员可以利用程序员开发的JavaBean组件和JSP标记来制作出动态页面,而且不会影响到内容的生成。

静态内容与动态内容的明确分离,是将以Java Servlet开发Web应用发展为以JSP开发Web应用的重要因素之一。

3. 可重复使用的组件

JavaBean组件是JSP中不可缺少的重要组成部分之一,程序通过JavaBean组件来执行所要求的更为复杂的运算。JavaBean组件不仅可以应用于JSP中,同样适用于其他的Java应用程序中。这种特性使得开发人员之间可以共享JavaBean组件,加快了应用程序的总体开发进程。

同样,JSP的标准标签和自定义标签与JavaBean组件一样可以一次生成重复使用。这些标签都是通过编写的程序代码来实现特定功能的,在使用它们时与通常在页面中用到的HTML标记用法相同。这样可以将一个复杂而且需要出现多次的操作简单化,大大提高了工作效率。

4. 沿用了Java Servlet的所有功能

相对于Java Servlet来说,使用从Java Servlet发展而来的JSP技术开发Web应用更加简单易学,并且JSP同样提供了Java Servlet所有的特性。实际上服务器在执行JSP文件时先将其转换为Servlet代码,然后再对其进行编译,可以说JSP就是Servlet,创建一个JSP文件其实就是创建一个Servlet文件的简化操作。理所当然,Servlet中的所有特性在JSP中同样可以使用。

5. 预编译

预编译是JSP的另一个重要特性。JSP页面在被服务器执行前,都是已经被编译好的,并且通常只进行一次编译,即在JSP页面被第一次请求时进行编译,在后续的请求中如果JSP页面没有被修改过,服务器只需要直接调用这些已经被编译好的代码,这大大提高了访问速度。

1.3 JSP的处理过程

当客户端浏览器向服务器发出请求要访问一个JSP页面时,服务器根据该请求加载相应的JSP页面,并对该页面进行编译,然后执行。JSP的处理过程如图1-1所示。

- (1) 客户端通过浏览器向服务器发出请求,在该请求中包含了请求资源的路径,这样当服务器接收到该请求后就可以知道被请求的资源。
- (2) 服务器根据接收到的客户端请求来加载被请求的JSP文件。
- (3) Web服务器中的JSP引擎会将加载的JSP文件转化为Servlet。
- (4) JSP引擎将生成的Servlet代码编译成Class文件。
- (5) 服务器执行这个Class文件。
- (6) 最后服务器将执行结果发送给浏览器进行显示。



JSP的处理
过程