

21世纪高等学校规划教材 | 计算机科学与技术



Java Web 开发技术教程

李希勇 编著



清华大学出版社



Java Web 开发技术教程

第2版



清华大学出版社

21世纪高等学校规划教材 | 计算



Java Web 开发技术教程

李希勇 编著

清华大学出版社
北京

内 容 简 介

本书是 Java Web 开发的入门与提高教材,全书共分 11 章,分别介绍 Java Web 开发环境的搭建和 JSP 的基础知识、JSP 页面元素和常用的内置对象、XML 的基本概念和使用、Servlet 技术、MVC 设计模式、数据库连接池的概念和 Tomcat 数据库连接池的使用、表达式语言 EL 的使用、标签编程和 JSTL 标签在 JSP 开发中的应用、Ajax 的原理和各种相关技术,并通过案例介绍了 Ajax 技术的常用实现。本书强调实践开发技能的提高,每一章都提供了对应知识内容的案例实训。

本书可作为计算机相关专业和信息管理相关专业 Java Web 程序设计相关课程的教材,也可作为 Web 项目开发人员的自学参考书。

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 13701121933

图书在版编目(CIP)数据

Java Web 开发技术教程/李希勇编著.--北京:清华大学出版社,2014

21 世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-37497-8

I. ①J… II. ①李… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 170842 号

责任编辑:付弘宇 薛 阳

封面设计:傅瑞学

责任校对:李建庄

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>,010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:19 字 数:473 千字

版 次:2014 年 8 月第 1 版 印 次:2014 年 8 月第 1 次印刷

印 数:1~2000

定 价:34.50 元

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21世纪高等学校规划教材·信息管理与信息系统。

(6) 21世纪高等学校规划教材·财经管理与应用。

(7) 21世纪高等学校规划教材·电子商务。

(8) 21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn



前言

当前 Web 应用程序已经逐步成为融合核心业务逻辑的业务处理平台,Web 2.0 概念的普及使其发展更加快速,应用更为广泛。在 Java 技术领域,Java Web 开发空前活跃,许多 IT 行业从业人员都在积极地学习有关 Java Web 的开发技术,市面和网络上也有许多非常好的学习资料,适合各个层次的学习者,然而面对众多学习渠道和资料,学习者特别是初学者往往不知所措,不知如何学习,对在实际开发学习和应用这些技术更是无从下手。

本书是作者在总结多年 Web 应用开发实践、教学实施经验和课程改革经验的基础上编写的,书中将 Java Web 开发按实际开发需要切割成若干知识点,循序渐进地培养学生的 JSP 页面编写能力和 Web 应用项目的开发能力。本书通过大量实例由简入繁地讲解了 Java Web 开发中各种技术的基础知识,每一章都附有针对该章知识点的案例实训,并给出了详细的实训步骤。

本书共分 11 章,详细介绍了 Java Web 应用开发的基础知识和基本技能。第 1 章介绍 Java Web 开发环境的搭建和 JSP 的基础知识,第 2~4 章重点讲解 JSP 的基础语法、JSP 页面元素和常用的内置对象;第 5 章介绍 XML 的基本概念和使用;第 6 章重点讲解 Servlet 的生命周期与对应的操作方法、内置对象在 Servlet 中的应用、过滤器和监视器的基本原理及应用;第 7 章介绍 MVC 设计模式的实现方法;第 8 章讲解数据库连接池的概念和 Tomcat 数据库连接池的使用;第 9 章详细介绍表达式语言 EL 的使用;第 10 章讲解标签编程和 JSTL 标签在 JSP 开发中的应用;第 11 章介绍 Ajax 的原理和各种相关技术,如 CSS、JavaScript、DOM 和 XMLHttpRequest 等,并通过案例介绍了 Ajax 技术的常用实现。

本书用了大量的实例进行讲解,力求通过实例使读者快速掌握 Java Web 应用开发技术。每章后面有精心编写的习题和案例实训,便于读者实践和掌握所学知识。

本书由李希勇编写,参与资料提供、程序调试和书稿校订的还有胡春才、罗晓娟、颜丽和孙正广等,非常感谢他们的辛勤工作。撰写过程中参考了一些优秀的著作、书籍和网站,在此对参考资料的作者表示最诚挚的谢意。在本书的出版过程中,得到了苏啸教授、周锦春教授及许多同事的指导和帮助,同时得到了清华大学出版社的大力支持,在此,向他们表示衷心的感谢!

在本书的编写过程中,尽管本人力求精益求精,但由于时间仓促,加之水平有限,书中难免有疏漏和不足,敬请读者和同行批评指正,可通过邮箱 lixiyong@126.com 与编者交流。

编者

2014 年 6 月



目 录

第 1 章 Java Web 开发环境简介	1
1.1 JSP 基础知识	1
1.1.1 JSP 概念	1
1.1.2 JSP 页面	1
1.1.3 JSP 的运行原理	2
1.2 Java Web 的开发环境	4
1.2.1 JDK 安装配置	5
1.2.2 Tomcat 服务器的搭建	8
1.2.3 使用 MyEclipse 建立一个 Java Web 项目	10
1.2.4 测试 JSP 程序	12
1.3 本章小结	13
习题	14
案例实训	14
第 2 章 JSP 基础语法	16
2.1 JSP 基本结构	16
2.2 变量和方法的声明	17
2.2.1 声明变量	17
2.2.2 声明方法	18
2.2.3 声明类	19
2.3 Java 程序片段	20
2.4 表达式	21
2.5 JSP 注释	22
2.5.1 HTML 注释	22
2.5.2 JSP 注释	23
2.6 本章小结	23
习题	23
案例实训	25
第 3 章 JSP 指令和动作元素	27
3.1 page 指令	27
3.1.1 设置页面编码	28

3.1.2 错误页的设置	29
3.1.3 数据库连接操作	31
3.2 include 指令	34
3.3 <jsp:include>动作标签	35
3.4 <jsp:forward>动作标签	36
3.5 <jsp:useBean>动作标签	37
3.6 <jsp:setProperty>动作标签	39
3.7 <jsp:getProperty>动作标签	42
3.8 本章小结	42
习题	43
案例实训	44
第4章 JSP 常用内置对象	46
4.1 JSP 内置对象及作用域概述	46
4.1.1 JSP 内置对象	46
4.1.2 JSP 的作用域	47
4.2 request 对象	48
4.2.1 获取客户提交的信息	48
4.2.2 处理汉字信息	49
4.2.3 常用方法举例	51
4.2.4 用户注册	53
4.3 response 对象	54
4.3.1 动态响应 contentType 属性	55
4.3.2 response 的 HTTP 文件头	56
4.3.3 response 重定向	57
4.4 session 对象	58
4.4.1 session 对象的 ID	58
4.4.2 session 对象与 URL 重写	58
4.4.3 session 对象常用的方法	58
4.4.4 登录及注销	60
4.5 application 对象	63
4.5.1 application 对象的常用方法	63
4.5.2 用 application 对象制作留言板	63
4.6 out 对象	66
4.7 本章小结	67
习题	68
案例实训	69

第 5 章 XML 简介	73
5.1 认识 XML	73
5.2 XML 解析	78
5.2.1 DOM 解析操作	78
5.2.2 SAX 解析操作	84
5.2.3 XML 解析的好帮手: JDOM	87
5.2.4 最出色的解析工具: DOM4J	90
5.3 本章小结	93
习题	93
案例实训	94
第 6 章 Servlet 技术	99
6.1 Servlet 简介	99
6.1.1 JSP 与 Servlet 的关系	99
6.1.2 Servlet 能够执行的功能	99
6.1.3 Servlet 工作体系结构及生命周期	100
6.2 Servlet 的操作实例	101
6.2.1 使用 Servlet 获取用户提交信息	101
6.2.2 使用 Servlet 实现页面转发和重定向	105
6.2.3 基于 Servlet 的网络通讯录的实现	107
6.3 Servlet 过滤器	119
6.3.1 一个字符过滤器的实现	120
6.3.2 过滤器链的实现	123
6.4 Servlet 监听器	123
6.4.1 实现 Servlet 监听器开发与部署的方法	124
6.4.2 实现 ServletContext 监听器	124
6.4.3 实现 HttpSession 监听器	126
6.4.4 实现 ServletRequest 监听器	130
6.5 本章小结	132
习题	132
案例实训	134
第 7 章 MVC	140
7.1 了解 MVC 模式	140
7.1.1 理解 MVC 设计模式	140
7.1.2 Model 1 介绍	140
7.1.3 基于 MVC 设计模式的 Model 2	141
7.2 为什么要使用 MVC 模式	141

11.9 级联菜单的设计与实现	267
11.10 自动补全功能	274
11.11 Ajax 存在的问题	282
11.12 本章小结	283
习题	283
案例实训	284
附录 A 习题(选择题)参考答案	288
参考文献	290

10.4.2	流程控制标签	189
10.4.3	循环标签	192
10.4.4	Url 操作标签	196
10.5	l18N 格式标签库	201
10.5.1	数字日期格式化	201
10.5.2	读取消息资源	210
10.5.3	国际化	214
10.6	SQL 标签库	216
10.6.1	设置数据源	216
10.6.2	SQL 操作标签	217
10.7	XML 标签库	223
10.7.1	XML 核心标签库	223
10.7.2	XML 流程控制	225
10.7.3	XML 的文件转换	226
10.8	函数标签库	227
10.9	本章小结	229
	习题	230
	案例实训	231
第 11 章	Ajax 技术	236
11.1	Ajax 的由来	236
11.2	初识 Ajax	237
11.3	关于同步与异步交互方式	238
11.4	Ajax 运行原理	239
11.5	体验 Ajax	242
11.5.1	传统实现方式	242
11.5.2	Ajax 的实现方式	246
11.6	XMLHttpRequest 对象	250
11.6.1	XMLHttpRequest 对象的方法	251
11.6.2	XMLHttpRequest 的属性	252
11.7	Ajax 对不同请求提交方式的处理	254
11.7.1	POST 提交方式的处理	254
11.7.2	GET 提交方式的处理	255
11.7.3	利用 Ajax 技术的页面提示效果	256
11.8	Ajax 核心工作机制	265
11.8.1	初始化对象	265
11.8.2	指定响应处理函数	265
11.8.3	发出 HTTP 请求	266
11.8.4	处理服务器返回的信息	266

7.2.1	使用 Model 1 存在的问题	141
7.2.2	MVC 模式的优势	141
7.3	深入 MVC	142
7.3.1	MVC 处理过程	142
7.3.2	MVC 的适用范围	143
7.4	本章小结	143
	习题	144
	案例实训	144
第 8 章	Tomcat 数据库连接池	151
8.1	数据库连接池概述	151
8.1.1	为什么使用数据库连接池	151
8.1.2	数据库连接池能做什么	151
8.2	数据源操作原理	152
8.3	在 Tomcat 中使用数据库连接池	152
8.4	查找数据源	154
8.5	本章小结	155
	习题	156
	案例实训	156
第 9 章	表达式语言	160
9.1	表达式语言简介	160
9.2	表达式语言的内置对象	161
9.2.1	访问 4 种属性范围的内容	162
9.2.2	调用内置对象操作	163
9.2.3	接收请求参数	164
9.3	集合操作	166
9.4	在 MVC 中应用表达式语言	167
9.5	运算符	170
9.6	本章小结	173
	习题	174
	案例实训	174
第 10 章	标签编程和 JSTL 标签库	178
10.1	标签编程简介	178
10.2	JSTL 简介	181
10.3	安装 JSTL 1.2	182
10.4	核心标签库	182
10.4.1	表达式控制标签	183

第 1 章

Java Web开发环境简介

本章目标:

- 掌握 JSP 编程的基础知识。
- 熟练掌握 Java Web 开发环境的搭建。

1.1 JSP 基础知识

1.1.1 JSP 概念

JSP(Java Server Pages)是由 Sun Microsystems 公司(已被甲骨文公司收购)倡导、许多公司一起参与建立的一种动态网页技术标准,JSP 技术是以 Java 语言作为脚本语言,用于创建可支持跨平台的 Java Web 动态网页技术。在传统的网页 HTML 文件(*.htm, *.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(tag),就构成了 JSP 网页(*.jsp)。

Java 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。JSP 网页中的程序片段可以操作数据库、重新定向网页以及发送 E-mail 等,这些功能就是 Java Web 网站所要实现的。Java Web 开发中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,所以对客户浏览器的要求很低。

JSP 是基于 Java Servlet 技术的,所以它继承了 Java 的一切优点,例如一次编写到处运行、平台无关性、系统的安全性等。

1.1.2 JSP 页面

深入学习 JSP 之前,先看一个 JSP 页面的代码。

【示例代码 1.1】 第一个 JSP 程序。

源文件名称: hello.jsp

```
<% @ page contentType = "text/html" pageEncoding = "GBK" %>
<HTML>
<HEAD><TITLE>Jsp</TITLE>
</HEAD>
<BODY>
<FONT>
```

```

<P>这是我们的第一个 JSP 程序
  <%
    String name = "Tom";
  %>
<BR>
<P>HELLO,<% = name %>
</FONT>
</BODY>
</HTML>

```

运行结果如图 1.1 所示。

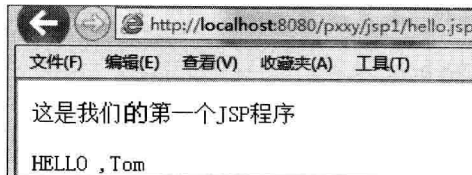


图 1.1 hello.jsp 运行结果

与我们前面学习过的 HTML 语言相比较,上面的 JSP 页面程序在 HTML 代码里面多了<%String name = "Tom";%>这样的代码。一个 JSP 文件就是在传统的 HTML 文件中加入了以“<%”和“%>”包含的 Java 的程序片段形成的以 JSP 为后缀的文件。在上面程序中,JSP 将 HTML 代码和 Java 程序执行后的结果(将 Tom 赋值给了 String 类型的变量 name)一起返回给我们,根据实际需求的不同,我们在 Java 程序中可以实现不同的操作,以建立动态网页的功能。

1.1.3 JSP 的运行原理

JSP 是基于 Java 技术的,当服务器上的一个 JSP 页面被第一次请求执行时,服务器上的 Web 容器首先将 JSP 页面文件转译成一个 Java 文件,再将这个 Java 文件编译生成字节码文件,然后通过执行字节码文件响应客户的请求。而当这个 JSP 页面再次被请求执行时,Web 容器将直接执行这个字节码文件来响应客户,这就是 JSP 程序速度比较快的一个原因,如图 1.2 所示。

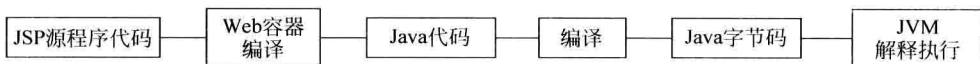


图 1.2 JSP 运行原理

为了更好地理解,下面先看看 JSP 编译成 Java 文件的结果:

```

import javax.servlet.* ;
import javax.servlet.http.* ;
import javax.servlet.jsp.* ;
public final class hello_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {
    private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();
    private static java.util.List _jspx_dependants;

```

```
private javax.el.ExpressionFactory _el_expressionfactory;
private org.apache.AnnotationProcessor _jsp_annotationprocessor;
public Object getDependants() {
    return _jspx_dependants;
}
public void _jspInit() {
    _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig().
getServletContext()).getExpressionFactory();
    _jsp_annotationprocessor = (org.apache.AnnotationProcessor) getServletConfig().
getServletContext().getAttribute(org.apache.AnnotationProcessor.class.getName());
}
public void _jspDestroy() {
}
public void _jspService(HttpServletRequest request, HttpServletResponse response)
    throws java.io.IOException, ServletException {
    PageContext pageContext = null;
    HttpSession session = null;
    ServletContext application = null;
    ServletConfig config = null;
    JspWriter out = null;
    Object page = this;
    JspWriter _jspx_out = null;
    PageContext _jspx_page_context = null;
    try {
        response.setContentType("text/html;charset = GBK");
        pageContext = _jspxFactory.getPageContext(this, request, response,
            null, true, 8192, true);
        _jspx_page_context = pageContext;
        application = pageContext.getServletContext();
        config = pageContext.getServletConfig();
        session = pageContext.getSession();
        out = pageContext.getOut();
        _jspx_out = out;
        out.write("\r\n");
        out.write("<HTML>\r\n");
        out.write("<HEAD><TITLE>Jsp</TITLE>\r\n");
        out.write("</HEAD>\r\n");
        out.write("<BODY>\r\n");
        out.write("<FONT>\r\n");
        out.write("<P>这是我们的第一个 JSP 程序\r\n");
        out.write(" ");
        String name = "Tom";
        out.write("\r\n");
        out.write("<BR>\r\n");
        out.write("<p>HELLO,");
        out.print(name);
        out.write("\r\n");
        out.write("</FONT>\r\n");
        out.write("</BODY>\r\n");
        out.write("</HTML>\r\n");
    } catch (Throwable t) {
```



```
        if (!(t instanceof SkipPageException)){
            out = _jspx_out;
            if (out != null && out.getBufferSize() != 0)
                try { out.clearBuffer(); } catch (java.io.IOException e) {}
            if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
        }
    } finally {
        _jspxFactory.releasePageContext(_jspx_page_context);
    }
}
}
```

上面加粗的部分就是 Web 容器响应客户端的代码。下面是在客户端查看到的代码：

```
<HTML>
<HEAD><TITLE> Jsp</TITLE>
</HEAD>
<BODY>
<FONT>
<P>这是我们的第一个 JSP 程序
<BR>
<p>HELLO, Tom
</FONT>
</BODY>
</HTML>
```

JSP 程序被编译成 Java 代码的工作是由 Web 容器来完成的。以 Tomcat 为例,如果 JSP 页面是在 Root 目录中被访问的,可以查看 Tomcat 安装目录下的 \work\Catalina\localhost_org\apache\jsp 目录,找到 JSP 引擎生成的 JSP 页面的 Java 文件和编译后的字节码文件。如果 JSP 页面存放在我们给出的 Web 文件夹中,例如 myweb,那么可以在 Tomcat 安装目录下的 \work\Catalina\localhost_org\apache\jsp\myweb 目录下找到 JSP 引擎生成的 JSP 页面的 Java 文件和编译后的字节码文件。

1.2 Java Web 的开发环境

在了解了 JSP 技术的工作原理之后,下面重点介绍 Java Web 的运行和开发环境。

基于 Java Web 项目的运行,在服务器端和客户端都必须有相应的环境。服务器端必须安装 Java 虚拟机以及与 Servlet 兼容的 Web 服务器,客户端则只要有 Web 浏览器就可以。现在主流的 Web 服务器有多种: Apache 的 Tomcat、Bea 的 WebLogic 等,其中 Tomcat 是一种开源的项目,是学习的很好选择。本章将以 Tomcat 为例,讲述它的基本安装和配置。

要进行可视化的 Java Web 项目开发,则需要借助 Java 集成开发环境 (IDE),如 Eclipse 或 MyEclipse。其中 MyEclipse 扩展了 Eclipse 的功能,集成了 Web 服务、程序框架和数据库等各种插件。数据库系统选择方面,Java Web 项目开发一般选择 Oracle 或开源的 MySQL,本书使用数据库的案例中一般采用开源的 MySQL。