

JSP+MySQL+Dreamweaver 动态网站建设全程揭秘

李晓斌 编著

700多个素材源文件

100个实战案例

500多分钟视频教程



Book features



内容全面，循序渐进

遵循从基础到高级，从入门到精通的思路，帮助读者轻松建立起正确的知识体系。



技术专业，易于操作

在技术讲解的同时，配合丰富的操作练习，理论与操作相结合，使读者轻松理解和掌握知识点。



专业作者，如临亲授

作者来自于一线专业公司和高校教师，实践与教学的融合，更清楚学习者的知识需求。



扫码微课，碎片学习

读者可以扫描书中的二维码，利用零碎的时间随时随地通过视频课程进行学习。



源于实践，回归实战

穿插大量的实战操作案例，均来源于实践，通过一步步操作的方式再现给读者。



海量资源，辅助学习

提供立体化的学习资源：所有案例的素材、源文件；多媒体语音视频教程；与教学配套的PPT课件。

KEYWORD

专业学习手册+完全配套资源+PPT课件+6部网页设计电子书+海量辅助设计资源+作者技术交流群

SUBSCRIBE

清华大学出版社

网页设计与开发殿堂之路

JSP+MySQL+ Dreamweaver 动态网站建设全程揭秘

李晓斌 编著

清华大学出版社

北 京

内 容 简 介

本书以 Dreamweaver 为工具, 结合 Apache Tomcat 服务器、JSP 程序语言和 MySQL 数据库, 全面系统地讲解了使用 Dreamweaver 开发 JSP 动态网站的方法和技巧, 并通过多个网站实用系统功能的开发讲解, 使读者能够快速掌握这些网站实用系统功能的实现方法。

本书内容简洁、通俗易懂, 通过知识点与案例相结合的方式, 让读者能够清晰明了地理解书中的相关技术内容, 从而达到理想的学习效果。全书共分 10 章, 包括配置 JSP 网站开发环境、JSP 基础语法、进入 JSP 的世界、JDBC 与 MySQL 数据库的操作、JSP 网站开发基础操作、统计网站访客信息、网站用户登录和注册系统、网站留言板系统、新闻发布管理系统和网站图片管理系统等内容。

本书结构清晰、实例经典、技术实用, 适合动态网页制作的初、中级读者, 也可以作为高等院校动态网页制作课程的教材, 还可以作为网页设计与制作爱好者的自学参考书。

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

JSP+MySQL+Dreamweaver 动态网站建设全程揭秘 / 李晓斌 编著. — 北京: 清华大学出版社, 2019

(网页设计与开发殿堂之路)

ISBN 978-7-302-52696-4

I . ① J… II . ① 李… III . ① JAVA 语言—网页制作工具 ② 关系数据库—数据库管理系统—程序设计
③ 网页制作工具 IV . ① TP393.092 ② TP311.138

中国版本图书馆 CIP 数据核字 (2019) 第 057420 号

责任编辑: 李 磊 焦昭君

封面设计: 王 晨

版式设计: 思创景点

责任校对: 成凤进

责任印制: 沈 露

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

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 19.75 字 数: 571千字

版 次: 2019年8月第1版 印 次: 2019年8月第1次印刷

定 价: 59.80元

产品编号: 077884-01

前言

随着科技的不断进步，网络为人们的工作和生活提供了非常多的便利，几乎已经融入生活的方方面面，而网站是人们获取信息的重要途径，因此如何能快速有效地设计出易用、好管理、互动性强的网站，是网页设计师面临的重要挑战。

网站的建设不仅包括设计静态页面的 HTML、CSS 样式和 JavaScript 脚本代码，还包含很多动态技术，例如常见的 ASP、PHP、JSP 技术等。在各种动态网站开发技术中，Tomcat +MySQL+JSP 组合以其高效性和跨平台性而著称，被誉为“黄金组合”，并得到广泛应用。

本书以 Dreamweaver 作为网站开发工具，详细介绍基于 Apache Tomcat 服务器、MySQL 数据库和 JSP 程序语言的动态网站开发技术，全面讲解使用 Dreamweaver 开发基于 Tomcat +MySQL+JSP 组合的动态网站的方法和技巧。

本书由资深网页设计及教学专家倾力编著，针对 JSP 动态网站开发过程中遇到的各种疑难问题，从实用、便捷的角度出发，由浅入深地介绍 JSP 网站开发的流程，通过大量的实用案例介绍具体的处理方法和技巧，在实例中穿插提示和技巧，并提供解决实际问题的方案，旨在引导读者快速掌握 JSP 动态网站开发，达到学以致用用的目的。全书共分 10 章，各章内容如下。

第 1 章 配置 JSP 网站开发环境，介绍动态网站开发技术、JSP 与 Java 的关系等相关内容，重点讲解 JDK 和 Tomcat 服务器的安装与配置，从而使读者能够在本地计算机中开发和测试 JSP 动态网页。

第 2 章 JSP 基础语法，介绍 JSP 的工作原理和 JSP 程序的基础语法，以及 Java 中的数据类型与变量、类、运算符与表达式、流程控制语句、数组、字符串等相关知识，使读者对 JSP 程序语言有更深入的了解。

第 3 章 进入 JSP 的世界，介绍 JSP 中的编译指令、动作指令以及各种内置对象和常用组件的使用方法，使读者能够进一步熟练地掌握 JSP 程序设计的方法和技巧。

第 4 章 JDBC 与 MySQL 数据库的操作，介绍 MySQL 数据库的安装和配置，以及如何通过 JDBC 来连接 MySQL 数据库，从而实现数据存取操作。

第 5 章 JSP 网站开发基础操作，介绍使用 Dreamweaver 开发 JSP 动态网站的相关面板和操作方法，包括创建 JSP 网站与 MySQL 数据库连接，查询数据库、向数据表中新增记录、更新记录、删除记录等操作。

第 6 章 统计网站访客信息，介绍如何使用 JSP 内置的 request 对象来获取网站访问者的相关信息，并将获取的访客信息写入数据库中，还介绍了使用 JSP 内置的 session 对象来判断是否为新的访客。

第 7 章 网站用户登录和注册系统，介绍网站登录和注册系统的规划以及 MySQL 数据库设计，重点讲解网站用户登录和注册系统中各种动态功能的实现方法。

第 8 章 网站留言板系统，介绍网站留言板系统的规划和 MySQL 数据库设计，重点讲解留言板系统中发表留言、留言回复和留言管理功能的实现方法。



第9章 新闻发布管理系统，介绍新闻发布和管理系统的规划及 MySQL 数据库设计，重点讲解新闻发布和管理系统中前台新闻显示、新闻搜索、后台添加和管理新闻功能的实现方法。

第10章 网站图片管理系统，介绍网站图片管理系统的规划和 MySQL 数据库设计，重点讲解图片管理系统中图片浏览、图片上传、图片显示和图片管理功能的实现方法。

本书由李晓斌编著，另外张晓景、高鹏、胡敏敏、张国勇、贾勇、林秋、胡卫东、姜玉声、周晓丽、郭慧等人也参与了部分编写工作。本书在写作过程中力求严谨，由于作者水平所限，书中难免有疏漏和不足之处，希望广大读者朋友批评、指正，欢迎与我们沟通和交流。QQ 群名称：网页设计与开发交流群；QQ 群号：705894157。

为了方便读者学习，本书为每个实例提供了教学视频，只要扫描一下书中实例名称旁边的二维码，即可直接打开视频进行观看，或者推送到自己的邮箱中下载后进行观看。本书配套的附赠资源中提供了书中所有实例的素材源文件、最终文件、教学视频和 PPT 课件，并附赠海量实用资源。读者在学习时可扫描下面的二维码，然后将内容推送到自己的邮箱中，即可下载获取相应的资源（注意：请将这几个二维码下的压缩文件全部下载完毕后，再进行解压，即可得到完整的文件内容）。



编 者

Search

目录



第1章 配置 JSP 网站开发环境

1.1 动态网站开发技术	1
1.1.1 Web 工作原理	1
1.1.2 静态网页	2
1.1.3 动态网页	2
1.2 了解 JSP 网站开发	5
1.2.1 了解 JSP	5
1.2.2 JSP 与 Java 的关系	5
1.2.3 JSP 的特点	6
1.2.4 JSP 网站开发环境包含的内容	6
1.3 JDK 的安装与配置	7
1.3.1 下载 JDK	7
1.3.2 安装 JDK	8
1.3.3 配置 Java 环境变量	9
1.4 Tomcat 服务器的安装与配置	11
1.4.1 下载 Tomcat 服务器	11
1.4.2 安装 Tomcat 服务器	12
1.4.3 Tomcat 服务器的目录结构	14
1.4.4 Tomcat 服务器的基本操作	16
1.4.5 Tomcat 服务器中的 Web 应用程序文件架构	16
1.5 在 Dreamweaver 中创建 JSP 站点	17
1.5.1 站点文件夹规划	17
1.5.2 JSP 测试服务器	18
实战 使用记事本制作 HTML 页面	18
1.5.3 创建 JSP 页面	19
实战 制作第一个 JSP 网页	19

第2章 JSP 基础语法

2.1 了解 JSP	23
------------	----

2.1.1 JSP 概述	23
2.1.2 JSP 工作原理	24
2.2 JSP 基础语法	25
2.2.1 声明变量或方法	25
2.2.2 JSP 表达式	25
2.2.3 脚本程序	26
2.2.4 标识符	26
2.2.5 JSP 代码中的注释	27
2.3 Java 数据类型与变量	28
2.3.1 基本数据类型	28
2.3.2 数据类型之间的转换	28
2.3.3 变量与常量	29
2.4 Java 数据类型与变量	30
2.4.1 数值类	30
2.4.2 日期时间类	32
2.5 运算符与表达式	33
2.5.1 赋值运算符	33
2.5.2 算术运算符	33
2.5.3 关系运算符	34
2.5.4 逻辑运算符	34
2.5.5 自动递增和递减运算符	34
2.5.6 条件运算符	35
2.5.7 关于表达式	35
2.6 Java 流程控制语句	35
2.6.1 条件语句	35
实战 使用条件语句计算优惠价格	36
实战 使用 switch 语句实现判断	39
2.6.2 循环语句	41
实战 使用循环嵌套输出九九乘法表	43
2.6.3 跳转语句	44
2.7 数组	45



2.7.1 定义数组	45
2.7.2 访问数组	46
2.8 字符串的处理	46
2.8.1 字符串的声明	46
2.8.2 字符串类的常用方法	48
实战 使用 String 类	48

第3章 进入JSP的世界

3.1 JSP 编译指令	50
3.1.1 include(加载指令)	50
3.1.2 page(网页指令)	51
3.1.3 taglib(卷标数据库指令)	52
3.2 JSP 动作指令	52
3.2.1 了解动作指令	52
3.2.2 <jsp:include>	53
实战 使用 <jsp:include> 动作指令包含文件	54
3.2.3 <jsp:forward>	55
实战 使用 <jsp:forward> 动作指令向转向页面传递参数	56
3.2.4 <jsp:useBean>	57
3.2.5 <jsp:setProperty>	57
3.2.6 <jsp:getProerty>	58
3.2.7 <jsp:plugin> 与 <jsp:fallback>	59
3.2.8 <jsp:params> 与 <jsp:param>	59
3.3 JSP 常用内置对象的使用方法	60
3.3.1 JSP 内置对象简介	60
3.3.2 request 对象	61
实战 使用 request 对象获取表单提交的数据	62
3.3.3 response 对象	63
实战 使用 response 对象实现网页重定向	64
3.3.4 session 对象	66
实战 清除指定 session 对象中的内容	68
3.3.5 application 对象	69
实战 使用 application 对象实现网页访问人数的统计	69
3.3.6 out 对象	71
3.3.7 其他内置对象	72
3.4 JavaBean 组件	73
3.4.1 了解 JavaBean	73
3.4.2 JavaBean 的种类	74
3.4.3 JavaBean 规范	74
3.4.4 JavaBean 属性	75
3.5 jspSmartUpload 组件	75

3.5.1 jspSmartUpload 组件的安装	75
3.5.2 jspSmartUpload 组件中的常用类	76
3.5.3 SmartUpload 核心类的使用方法	77
实战 使用 jspSmartUpload 组件实现文件上传	78

第4章 JDBC 与MySQL数据库的操作

4.1 数据库与JDBC技术	82
4.1.1 了解数据库与数据库管理系统	82
4.1.2 关系型数据库	82
4.1.3 JDBC 简介	83
4.1.4 JDBC 驱动程序	84
4.1.5 数据库驱动程序	84
4.2 MySQL 数据库基础	84
4.2.1 了解 MySQL 数据库	84
4.2.2 MySQL 数据库的特点	85
4.2.3 下载 MySQL 数据库	85
4.2.4 MySQL 数据库的安装与配置	87
4.2.5 MySQL 数据库的对象	91
4.2.6 字段的类型	92
4.3 MySQL 数据库的基础操作	94
4.3.1 启动和关闭 MySQL 服务	94
4.3.2 进入和退出 MySQL 管理控制平台	95
实战 进入和退出 MySQL 管理控制平台	95
4.3.3 常用 SQL 语句	96
4.3.4 使用 JSP 连接 MySQL 数据库	98
4.4 MySQL Workbench 数据库图形管理工具	99
4.4.1 MySQL Workbench 简介	99
4.4.2 认识 MySQL Workbench 工作界面	99
4.4.3 使用 MySQL Workbench 创建数据库和数据表	101
实战 创建 MySQL 数据库和数据表	101
4.4.4 添加数据	103
实战 向数据表插入和编辑数据	104
4.4.5 备份与还原数据库	105
实战 使用 MySQL Workbench 进行数据库备份与还原操作	106

第5章 JSP 网站开发基础操作

5.1 使用 Dreamweaver 开发动态网站的关键步骤	109
5.2 Dreamweaver 动态网站开发环境	110
5.2.1 安装 JSP 动态网站开发扩展	110
5.2.2 认识动态网站开发相关面板	112

5.2.3 动态内容源	113
5.3 开发前的准备工作	114
5.3.1 系统功能分析	114
5.3.2 创建 MySQL 数据库	115
5.4 创建网站数据库连接	117
5.4.1 JSP 程序连接数据库服务器的原理	117
5.4.2 下载并安装 MySQL 数据库驱动	118
5.4.3 在 Dreamweaver 中创建动态站点	120
实战 创建图书管理系统站点	120
5.4.4 创建 MySQL 数据库连接	121
实战 创建 JSP 与 MySQL 数据库连接	122
5.5 在 Dreamweaver 中创建并使用数据记录	123
5.5.1 创建记录集	123
实战 创建图书管理系统数据记录集	124
5.5.2 显示数据记录	126
实战 在网页中插入记录集字段	126
5.5.3 “重复区域”服务器行为	126
实战 重复显示多条数据记录	127
5.5.4 “显示区域”服务器行为	127
实战 判断页面中的显示范围	128
5.5.5 “记录集分页”服务器行为	129
实战 添加分页功能	129
5.5.6 记录集导航	130
实战 添加显示计数功能	130
5.5.7 通过超链接传递参数	131
实战 制作详细信息显示页面	131
5.6 数据记录的编辑处理	134
5.6.1 管理页面	134
实战 制作后台数据记录管理页面	134
5.6.2 “插入记录”服务器行为	135
实战 制作添加数据记录页面	136
5.6.3 “更新记录”服务器行为	137
实战 制作更新数据记录页面	137
5.6.4 “删除记录”服务器行为	139
实战 制作删除数据记录页面	139
5.7 JSP 页面出现中文乱码的解决方法	140
5.8 系统功能测试	141
实战 测试图书管理系统功能	141

第 6 章 统计网站访客信息

6.1 系统功能分析	144
6.1.1 网站访客信息统计系统分析	144
6.1.2 相关页面说明	144
6.2 创建系统站点和 MySQL 数据库	145
6.2.1 创建站点	145

实战 创建网站访客信息系统站点	145
6.2.2 创建 MySQL 数据库	146
实战 创建网站访客信息系统数据库	146
6.2.3 创建 MySQL 数据库连接	148
实战 创建网站访客信息系统数据库连接	148
6.3 制作访客信息统计功能	149
6.3.1 “已准备(插入、更新、删除)”服务器 行为	149
实战 实现访客数量统计功能	150
6.3.2 通过 session 对象进行判断	153
实战 加入 session 判断是否为新访客	154
6.3.3 显示访客信息	155
实战 制作查看访客信息页面	156

第 7 章 网站用户登录和注册系统

7.1 系统功能分析	159
7.1.1 网站用户登录和注册系统规划	159
7.1.2 网站用户登录和注册系统相关页面 说明	160
7.2 创建系统站点和 MySQL 数据库	160
7.2.1 用户登录和注册系统站点	160
实战 创建用户登录和注册系统站点	160
7.2.2 创建 MySQL 数据库	161
实战 创建用户登录和注册系统数据库	162
7.2.3 创建 MySQL 数据库连接	164
实战 创建用户登录和注册系统数据库 连接	164
7.3 开发网站用户登录功能	164
7.3.1 “登录用户”服务器行为	164
实战 制作网站用户登录页面	165
7.3.2 登录成功与登录失败的处理	167
实战 制作登录成功与登录失败页面	167
7.3.3 修改用户个人信息	169
实战 制作修改用户个人信息页面	169
7.3.4 个人信息修改成功页面	172
7.3.5 成功退出登录页面	172
7.4 开发网站新用户注册功能	173
7.4.1 实现注册表单验证	173
实战 实现注册表单验证功能	173
7.4.2 新用户注册	175
实战 制作网站新用户注册页面	175
7.4.3 注册成功与注册失败的处理	177
实战 制作注册成功和注册失败页面	177
7.5 开发找回密码功能	177
7.5.1 找回密码	178



实战 制作找回密码页面	178
7.5.2 密码保护问题	179
实战 制作密码保护问题页面	179
7.5.3 找回密码成功	181
实战 制作密码找回结果页面	181
7.6 系统功能测试	183
实战 测试网站用户登录和注册系统功能 ..	183

第8章 网站留言板系统

8.1 系统功能分析	186
8.1.1 网站留言板系统规划	186
8.1.2 网站留言板系统相关页面说明	187
8.2 创建系统站点和 MySQL 数据库	187
8.2.1 网站留言板系统站点	187
实战 创建网站留言板系统站点	187
8.2.2 创建 MySQL 数据库	188
实战 创建网站留言板系统数据库	189
8.2.3 创建 MySQL 数据库连接	193
实战 创建网站留言板系统数据库连接	193
8.3 开发提交留言功能	194
8.3.1 选择用户头像	194
实战 实现选择用户头像功能	194
8.3.2 发表留言	196
实战 制作发表留言页面	196
8.4 开发显示留言功能	198
8.4.1 显示留言	198
实战 制作在留言板首页显示留言功能	198
8.4.2 回复留言	202
实战 制作回复留言表单页面	202
8.4.3 显示留言回复内容	204
实战 制作显示留言回复页面	204
8.4.4 调用显示留言回复页面	207
实战 在留言板首页中调用显示留言回复 页面	207
8.5 开发留言管理功能	208
8.5.1 管理登录	208
实战 制作管理登录页面	208
8.5.2 设置“删除”超链接	210
实战 设置页面中的“删除”超链接是否 可见	210
8.5.3 删除留言	211
实战 制作确认删除留言页面	211
8.5.4 仅管理员可见	212
实战 实现仅管理员可见功能	212
8.6 系统功能测试	213

实战 测试网站留言板系统功能	213
----------------------	-----

第9章 新闻发布管理系统

9.1 系统功能分析	216
9.1.1 新闻发布管理系统规划	216
9.1.2 新闻发布管理系统相关页面说明	217
9.2 创建系统站点和 MySQL 数据库	217
9.2.1 新闻发布管理系统站点	217
实战 创建新闻发布管理系统站点	217
9.2.2 创建 MySQL 数据库	219
实战 创建新闻发布管理系统数据库	219
9.2.3 创建 MySQL 数据库连接	223
实战 创建新闻发布和管理系统数据库 连接	223
9.3 开发新闻显示功能	224
9.3.1 新闻列表	224
实战 制作新闻列表页面	224
9.3.2 新闻分类列表	228
实战 制作新闻分类列表页面	228
9.3.3 新闻搜索	231
实战 实现新闻搜索功能	231
9.3.4 新闻显示页面	235
实战 制作新闻显示页面	235
9.4 开发管理登录页面	236
实战 制作新闻管理登录页面	236
9.5 开发新闻显示功能	237
9.5.1 新闻管理主页面	237
实战 制作新闻管理主页面	237
9.5.2 使用富文本编辑器	240
实战 在 JSP 页面中使用 UEditor 编辑器 ..	240
9.5.3 添加新闻	242
实战 制作添加新闻页面	242
9.5.4 修改新闻	244
实战 制作修改新闻页面	244
9.5.5 删除新闻	247
实战 制作删除新闻页面	247
9.6 开发新闻分类管理功能	249
9.6.1 新闻分类管理	249
实战 制作新闻分类管理页面	249
9.6.2 添加新闻分类	251
实战 制作添加新闻分类页面	251
9.6.3 修改新闻分类	252
实战 制作修改新闻分类页面	252
9.6.4 删除新闻分类	254
实战 制作删除新闻分类页面	254

9.7 系统功能测试	255
实战 测试网站新闻发布管理系统功能	256

第 10 章 网站图片管理系统

10.1 系统功能分析	259
10.1.1 网站图片管理系统规划	259
10.1.2 网站图片管理系统相关页面说明	260
10.2 创建系统站点和 MySQL 数据库	260
10.2.1 网站图片管理系统站点	260
实战 创建网站图片管理系统站点	260
10.2.2 创建 MySQL 数据库	262
实战 创建网站图片管理系统数据库	262
10.2.3 创建 MySQL 数据库连接	266
实战 创建网站图片管理系统数据库连接	266
10.3 开发浏览作品图片功能	267
10.3.1 网站图片管理系统首页	267
实战 制作网站图片管理系统首页	267
10.3.2 作品显示模板	269
实战 制作作品显示模板	269
10.3.3 将作品显示模板嵌入网站图片管理系统 首页	272
实战 将作品显示模板嵌入网站图片管理系统 首页	272
10.3.4 全部作品列表页面	273
实战 制作全部作品列表页面	273
10.3.5 作品分类列表页面	278
实战 制作作品分类列表页面	278

10.3.6 查看作品页面	280
实战 制作查看作品页面	280
10.4 开发作品管理功能	282
10.4.1 后台管理登录	282
实战 制作后台管理登录页面	282
10.4.2 作品管理页面	283
实战 制作注册成功和注册失败页面	283
10.5 开发作品图片上传功能	286
10.5.1 添加 UploadBean 组件	286
10.5.2 上传图片	287
实战 制作上传图片页面	287
10.5.3 添加作品	290
实战 制作注册成功和注册失败页面	290
10.6 开发修改和删除作品功能	293
10.6.1 修改作品	294
实战 制作注册成功和注册失败页面	294
10.6.2 删除作品	297
实战 制作删除作品页面	297
10.7 开发作品分类管理功能	299
10.7.1 作品分类管理	299
实战 制作作品分类管理页面	299
10.7.2 添加作品分类	302
实战 制作注册成功和注册失败页面	302
10.7.3 修改作品分类和删除作品分类	303
10.8 系统功能测试	303
实战 测试网站图片管理系统功能	303

第1章 配置 JSP 网站开发环境



JSP 是一种在服务器端解释执行的动态网页开发技术，如果在本地计算机中进行 JSP 网站的开发及测试工作，则需要在本计算机中搭建 JSP 网站的开发环境。配置 JSP 开发环境的方法很多，但主要工作就是安装和配置 Web 服务器和 JSP 引擎。在本章中将介绍动态网页开发与 JSP 网站开发环境的相关知识，使读者能够更深入地理解 JSP 动态网站开发，并能够在本地计算机中搭建出 JSP 网站开发环境，这也是 JSP 网站开发的第一步。

本章知识点：

- 理解动态网站开发技术
- 了解 JSP 语言的特点及 JSP 开发环境
- 掌握 JDK 的安装与配置
- 掌握 Tomcat 服务器的安装与配置
- 掌握在 Dreamweaver 中创建 JSP 测试服务器站点的方法
- 掌握在 JSP 页面出现中文乱码的解决方法

1.1

动态网站开发技术



静态网页只要在浏览器中打开即可看到网页执行的结果，但是，对于使用了各种服务器端语言 (PHP、ASP 和 JSP 等) 的动态网页，浏览器是无法直接解析这些程序代码的，需要经过 Web 服务器解析后才能看到执行的结果。在学习 PHP 动态网站开发之前，首先要了解动态网站开发的相关技术和原理。

1.1.1 Web 工作原理



互联网是一组彼此连接的计算机，也称为网络。全世界所有计算机通过传输控制协议 (Transmission Control Protocol/Internet Protocol, TCP/IP 协议) 绑定成为一个整体。人们通过互联网可以与千里之外的朋友交流，共同娱乐、共同完成工作，如图 1-1 所示。

万维网，英文全称为 World Wide Web，简称 WWW，是互联网的一个子集，为全世界用户提供信息。万维网是 Internet 上基于客户端 / 服务器端体系结构的分布式多平台的超文本超媒体信息服务系统，它是 Internet 中最主要的信息服务，允许用户在一台计算机上通过 Internet 存取另一台计算机上的信息。

WWW 又称为 3W 或 Web，它作为 Internet 上的新一代用户界面，摒弃了以往纯文本方式的信息交互手段，采用超文本 (Hypertext) 方式工作。利用该技术可以提供全球范围的多媒体信息服务，使企业获取信息的手段有了根本性的改善。

WWW 主要由服务器端 (Server) 和客户端 (Client) 两部分组成。服务器端是信息的提供者，就是存放网页供用户浏览的网站，也称为 Web 服务器。客户端是信息的接收者，通过网络浏览网页的用

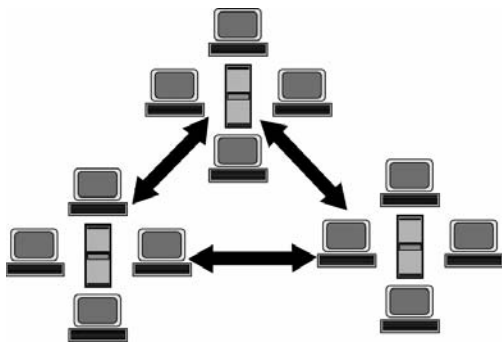


图 1-1



户或计算机的总称，浏览网页的程序称为浏览器。

WWW 中的网页浏览过程，是由客户端的浏览器向服务器端的 Web 服务器发送浏览网页的请求，Web 服务器就会响应该请求并将相应网页传送到客户端的浏览器，然后由浏览器解析和显示网页。

1.1.2 静态网页

静态网页是指客户端的浏览器发送 URL 请求给 WWW 服务器，服务器查找需要的超文本文件，不加任何处理直接下载到客户端，运行在客户端的页面是已经事先做好并存放在服务器中的网页。其页面内容使用的仅仅是标准的 HTML 代码。静态网页通常由纯粹的 HTML 和 CSS 样式构成。

网站制作人员把内容设计成静态网页，访问者只能被动地浏览网站中提供的内容，静态网页的内容不会发生变化，除非设计者修改了网页的内容。静态网页不能实现和浏览网页的用户之间的交互，信息流向是单向的，即从服务器到浏览器，服务器不能根据用户的选择调整返回给用户的内容。

1.1.3 动态网页

网络技术的发展日新月异，许多网页文件的扩展名不再只是 .html，还有 .php、.asp、.jsp 等，这些都是采用动态网页技术制作的。动态网页其实就是建立在浏览器 / 服务器 (B/S) 架构上的服务器端脚程序。在浏览器端显示的网页是服务器端程序运行处理后的结果。

静态网页与动态网页的区别在于 Web 服务器对它们的处理方式不同。当 Web 服务器接收到静态网页的请求时，服务器直接将该页面发送给客户端浏览器，不进行任何处理。如果接收到动态网页的请求，则从 Web 服务器中找到该文件，并将它传递给一个称为应用程序服务器的特殊软件扩展，由它负责解释和执行网页，将执行后的结果传递给客户端浏览器进行显示。

提示

动态网页是与静态网页相对应的，静态网页的 URL 扩展名是以 .htm、.html、.shtml、.xml 等常见形式出现的。而动态网页的 URL 扩展名是以 .asp、.jsp、.php、.perl、.cgi 等形式出现的。

动态网页技术根据程序运行的区域不同，分为客户端动态技术与服务器端动态技术。

1. 客户端动态技术

常见的客户端动态技术包括 JavaScript、VBScript、DHTML、Flash、Java Applet 和 ActiveX 等。客户端动态技术不需要与服务器进行交互，实现动态功能的代码往往采用脚本语言形式直接嵌入网页中。服务器发送给浏览者后，网页在客户端浏览器上直接响应用户的动作，有些应用还需要浏览器安装组件支持。

2. 服务器端动态技术

服务器端动态技术需要与客户端共同参与，客户通过浏览器发出页面请求后，服务器根据 URL 携带的参数运行服务器端程序，产生的结果页面再返回客户端。一般涉及数据库操作的网页（如登录、注册和搜索等）都需要服务器端动态技术程序。动态网页比较注重交互性，即网页会根据用户的要求和选择而动态地改变和响应。将浏览器作为客户端界面，这将是今后 Web 发展的趋势。动态网站上主要是一些页面布局，网页的内容大都存储在数据库中，并可以利用一定的技术使动态网页内容生成静态网页内容，方便网站的优化。

典型的服务器端动态技术有 CGI、ASP/ASP.NET、PHP 和 JSP 等。

1) CGI

CGI 是一种编程标准，它规定了 Web 服务器用其他可执行程序的接口协议标准。CGI 程序通过读取使用者的输入请求，从而产生 HTML 网页。它可以用任何程序设计语言编写。

可以使用不同的程序语言编写适合的 CGI 程序，如 VB、Delphi 或 C/C++ 等。用户将编写好的程序放在 Web 服务器上运行，再将其运行结果通过 Web 服务器传输到客户端的浏览器上。事实上，这样的编制方式比较困难，而且效率低下，因为用户每一次修改程序都必须重新将 CGI 程序编译成可执行文件。

2) ASP/ASP.NET

ASP 是 Active Server Pages 的缩写，是 Microsoft 公司开发的 Web 服务器端脚本开发环境，利用它可以生成动态、高效的 Web 应用程序。

虽然人们习惯于将 ASP 称为 ASP 语言，但从严格意义上讲，ASP 只是为 VBScript 和 JavaScript 等脚本语言提供了一个运行的环境，使开发人员可以在 HTML 代码中使用脚本语言编写程序。当然，ASP 自身也提供了一些非常好用的命令和内置对象。

ASP 程序保存扩展名为 .asp 的文件，一个 ASP 文件相当于一个可执行文件，因此必须放在 Web 服务器上有可执行权限的目录下。当浏览器向 Web 服务器请求调用 ASP 文件时，就启动了 ASP。Web 服务器开始调用 ASP，将被请求的 .asp 文件从头读到底，执行每一个命令，然后动态生成一个 HTML 页面并送到浏览器。由于 ASP 在服务器端解释执行，开发者可以不必考虑浏览器是否支持 ASP，也不必担心程序会被从客户端下载。执行 ASP 文件的过程如图 1-2 所示。



图 1-2

如下代码为一个简单的 ASP 程序实例。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>简单 ASP 演示程序</title>
</head>
<body>
这是一个简单的 ASP 演示程序，刷新可以显示当前时间：<br><br>
当前时间为：<%response.write Now%>
</body>
</html>
```

<% 和 %> 是 ASP 的定界符，其中的语句可以是 ASP 命令，也可以是 VBScript 脚本程序。response 是 ASP 的内置对象，用于回复浏览器端的请求。response.write 的功能是在当前位置输出指定的数据。Now 是 VBScript 的函数，功能是返回当前的系统日期和时间。

提示

ASP 源文件必须插入 <%...%> 之间，Microsoft 公司针对 JavaScript 推出的 Script 语言即 VBScript，ASP 只能在 Windows 系列的服务器上使用，因此通常使用 VBScript 语言。

ASP.NET 是微软公司近年来开发的以 .NET Framework 为基础的动态网站技术。ASP.NET 是 ASP 的 .NET 版本，是一种编译式的动态技术，执行效率较高，同时支持使用通用语言建立动态网页。



3) PHP

PHP 英文全称为 Hypertext Preprocessor，是一种被广泛应用的开放源代码的多用途脚本语言，它可以嵌入 HTML 中，尤其适合网页开发。

PHP 主要用于服务器端的脚本程序，可以用 PHP 来完成任何其他 CGI 程序能够完成的工作，例如，收集表单数据，生成动态网页或者发送 / 接收 Cookies。但 PHP 的功能远不局限于此，它是一种基于服务器端来创建动态网站的脚本语言，可以用 PHP 和 HTML 生成网站页面。当访问者浏览页面时，服务器端便执行 PHP 的命令并将执行结果发送至访问者的浏览器中，工作机制类似于 ASP 和 ColdFusion，PHP 和它们的不同之处在于，PHP 是开放源码且可跨平台，PHP 可以运行在 Windows 和多种版本的 UNIX 及其他操作系统中。

PHP 的脚本以 `<?php` 开始，以 `?>` 结束，可以把 PHP 的脚本代码放置在文档中的任何位置。

如下代码为一个简单的 PHP 程序实例。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> 简单 PHP 演示程序 </title>
</head>
<body>
<?php
echo " 使用 PHP 输出文字 ";
?>
</body>
</html>
```

在代码中 `<?php ...?>` 部分即 PHP 编程相关的部分。PHP 中的每个代码行都必须以分号结束。分号是一种分隔符，用于把指令集区分开来。

4) JSP

JSP 是由 Sun 公司主导，许多公司一起参与建立的一种动态网页技术标准，英文全称为 Java Service Page。该技术为创建显示动态生成内容的网页页面提供了一个简捷而快速的方法。

JSP 技术的设计目的是使构造基于网页的应用程序更加容易和快捷，而这些应用程序能够与各种网站服务器、应用服务器、浏览器和开发工具共同工作。在传统的 HTML 页面中加入 Java 程序片段和 JSP 标记，就构成了 JSP 网页 (*.jsp)。网站服务器在遇到访问 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 形式返回给访问者。程序片段可以操作数据库、重新定向网页和发送 E-mail 等，这就是建立动态网站所需的功能。所有程序操作都在服务器端执行，网络上传递给客户端的仅是得到的结果，对访问者的浏览器要求比较低。

JSP 的脚本以 `<%` 开始，以 `%>` 结束，可以把 JSP 脚本块放在文档中的任何位置。

如下代码为一个简单的 JSP 程序实例。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> 简单 JSP 演示程序 </title>
</head>
<body>
<b> 今天是: </b>
<% = new java.util.Date() %>
```

```
<% = new java.util.Date() %>
</body>
</html>
```

提示

JSP 几乎可以运行在所有的服务器系统上,对客户端浏览器要求也很低。JSP 支持 85% 以上的操作系统,除了 Windows 外,它还支持 Linux 和 UNIX 等操作系统。

1.2 了解 JSP 网站开发

HTML 页面可以直接使用浏览器进行浏览,并不需要通过服务器解析。但是,浏览器只能看懂 HTML、VBScript、JavaScript 这些客户端语言,对于各种服务器端语言(ASP、PHP、JSP 等)的网页,浏览器无法解析这些程序代码。对于服务器端语言,则需要经过 Web 服务器解析成 HTML。

所以在开发 JSP 网站之前,需要在本地计算机中搭建能够对 JSP 程序进行编译的开发环境,这样才能在本地计算机中开发 JSP 网站页面。

1.2.1 了解 JSP

JSP 是原 Sun 公司推出的新一代网站开发语言,在 2010 年 Sun 公司被 Oracle 公司收购。JSP 技术类似 ASP 技术,它是在传统的网页 HTML 文件中插入 Java 脚本程序 (Scriptlet) 和 JSP 标记 (Tag),从而形成 JSP 文件。使用 JSP 开发的 Web 应用程序是跨平台的,既能在 Windows 操作系统下运行,也能在其他的操作系统下运行,如 Linux、UNIX 等。

目前,JSP 已经是主流的服务器端动态网页技术,尤其是电子商务类的网站大多采用 JSP 程序语言进行开发。JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成。服务器在页面被客户端请求以后对这些 Java 代码进行处理,然后才翻译成 HTML 页面返回给客户端的浏览器。JSP 具备了 Java 技术的简单易用,完全地面向对象,具有平台无关性且安全可靠等众多优势。

提示

Java Servlet 是 JSP 的技术基础,大型的 Web 应用程序开发需要 Java Servlet 和 JSP 配合才能完成。

1.2.2 JSP 与 Java 的关系

在介绍 JSP 之前,必须先介绍一下 Java 程序语言,因为 JSP 动态网页技术是 Java 家庭的一部分,JSP 又能去调用 JavaBean 和 Enterprise Java Beans(EJB)。

Java 程序语言是由 Sun 公司于 1991 年开发的,Java 语言是面向对象的语言,它的起源是 C++,所以 Java 语言沿用大部分 C++ 的语法。

基于网络应用的发展,现在的 Java 版本为 Java 2,按照使用平台的不同再分为 3 个层级:分别是企业级应用的 Java 2 Enterprise Edition(J2EE)、客户端程序所需要的开发环境 Java 2 Standard Edition(J2SE),以及嵌入式系统应用的 Java 2 Micro Edition(J2ME)。J2EE、J2SE 和 J2ME 都是 Java 2 开发平台的工具,其中 J2EE 属于服务器端的开发工具,而 J2SE 属于一般客户端的开发环境,而 J2ME 则属于家电产品或 PDA 等支持 Java 语言的电子产品的开发工具。J2SE 包括 Java 2 SDK 标准版与 Java 2 Runtime 标准版,它着重于单机环境下 Java 程序的开发。如果要使用到网络环境,建议使用 J2EE。



JSP 可以说是 Java 在开发网页应用程序中的一种技术, 并且其语法也是以 Java 的 Servlet 为基础。

1.2.3 JSP 的特点



JSP 作为一种服务器端的脚本语言, 其特点主要表现在以下几个方面。

1. 跨平台运行

JSP 技术以 Java 为基础, 可以沿用 Java 强大的 API 功能, 并且无论是在何种平台下, 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、跨服务器的特点。例如, 如今流行的 Web 服务器 Apache 同样能够支持 JSP, 而且 Apache 服务器支持多种平台, 从而使 JSP 可以在更多的平台上运行。

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 中提供的 JDBC 来连接数据库, 就不用担心平台变更时的代码移植问题。

2. 分离内容生成和显示

使用 JSP 技术, Web 页面开发人员可以使用 HTML 或 XML 标识来设计静态页面, 使用 Java 脚本程序和 JSP 标记来生成页面中的动态内容, 生成内容的逻辑被封装在标识和 JavaBean 组件中, 并且捆绑在 Java 脚本程序中, 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 JavaBean 中, 那么其开发人员, 例如 Web 管理人员和页面设计者, 能够编辑和使用 JSP 页面, 而不影响页面内容的生成。

3. 可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件 (JavaBean 或者企业级 JavaBean 组件) 来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件, 或者使这些组件为更多的使用者或者客户团体所使用。

4. 采用标识简化页面开发

JSP 技术封装了许多功能, 这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所需要的。

标准的 JSP 标识能够访问和实例化 JavaBean 组件, 通过开发定制化标识库, JSP 技术是可以扩展的。第三方开发人员和其他人员可以为常用功能创建自己的标识库, 这使 Web 页面开发人员能够使用熟悉的工具来执行特殊的功能, 从而更加轻松地完成任务。

5. 采用预编译提高访问速度

作为 Java 平台的一部分, JSP 拥有 Java 编程语言“一次编写, 各处运行”的特点。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, 则服务器只需要直接调用这些已经被编译好的代码, 这样大大提高了页面的访问速度。

1.2.4 JSP 网站开发环境包含的内容

完整的 JSP 网站开发环境涉及操作系统、Java 开发工具包、Web 服务器和数据库。

本书所使用的开发环境为 Windows 7 + Dreamweaver CC(网页制作软件) + JDK(Java 开发工具包) + Tomcat(Web 服务器) + MySQL(数据库)。

1. JDK

JDK 的英文全称为 Java Develop Kit, 是 Java 开发工具包, 其中包含运行 Java 程序所必需的 Java 虚拟机、Java 类库和开发工具等。在使用 JSP 开发网站之前, 首先必须安装 JDK。

2. Tomcat 服务器

Tomcat 是 Apache-Jakarta 软件组织开发的一个服务器软件包，是一个小型的、轻量级的、支持 JSP 和 Servlet 技术的 Web 服务器。Tomcat 既可以嵌入 Apache 使用，也可以作为独立的 Web 服务器使用，而且还具有作为商业 Java Web 应用容器的特征，它已经成为学习开发 JSP 应用的首选。

3. MySQL 数据库

MySQL 数据库是一个开放源码的小型关系数据库管理系统，由于其具有体积小、速度快、成本低等优点，目前被广泛应用于 Internet 的中小型网站中。MySQL 数据库是一个真正的多用户、多线程的 SQL 数据库服务器。由于 MySQL 数据库源代码的开放性和稳定性，并且可与 JSP 完美结合，很多网站使用它们进行 Web 开发，有关 MySQL 数据库的具体内容将在第 4 章中进行详细介绍。

1.3

JDK 的安装与配置

JDK 是一切 Java 应用程序的基础，可以说，所有的 Java 应用程序都是构建在 JDK 之上的，其核心是一组 Java API。本节将向读者介绍 JDK 的下载、安装和配置操作。

1.3.1 下载 JDK

JDK 是 Java Development Kit 的缩写，中文称为 Java 开发工具包。它是一种用于构建在 Java 平台上发布的应用程序、applet 和组件的开发环境。JDK 是一切 Java 应用程序的基础，所有的 Java 应用程序都是构建在 JDK 之上的，它是一组 API，也可以说是一些 Java Class。要成为一个 Java 程序员，JDK 是必不可少的基本工具。因此，编写 JSP 动态页面，也必须了解 JDK。

Sun 公司在 2010 年被 Oracle 公司收购，现在可以在 Oracle 公司的官方网站中下载 JDK。

打开浏览器，在地址栏中输入 Oracle 公司的官方网站地址 www.oracle.com，进入网站首页，在导航菜单中选择 Java 中的 Java SE，如图 1-3 所示。进入 Java SE 的介绍页面，单击“Java SE 平台”超链接，如图 1-4 所示。



图 1-3



图 1-4

进入 Java SE 平台页面，切换到 Downloads 选项卡中，即可看到 Java SE 最新版本和历史版本的相关信息以及下载按钮，如图 1-5 所示。在该页面中找到 Java SE 8u171/8u172 版本，单击该版本 JDK 选项的 DOWNLOAD 按钮，如图 1-6 所示。

进入 Java SE 8u171/8u172 版本的下载页面中，在该页面中提供了针对众多不同操作系统的 JDK 安装文件，在下载 JDK 之前，选中 Accept License Agreement 单选按钮，接受许可协议，如图 1-7 所示。然后根据所使用的操作系统来选择下载不同的 JDK 版本，笔者使用的是 64 位 Windows 7 操作系统，所以这里下载 Windows X64 版本的 JDK，如图 1-8 所示。如果读者使用的是 32 位的 Windows 操作系统，就需要下载 Windows X86 版本的 JDK。



图 1-5



图 1-6



图 1-7



图 1-8

单击 jdk-8u171-windows-x64.exe 文件名称, 显示下载提示, 如图 1-9 所示。单击“保存”按钮, 即可下载针对 64 位 Windows 操作系统的 JDK 安装程序。下载完成后, 可以在文件保存位置看到所下载的 jdk-8u171-windows-x64.exe 安装程序文件, 如图 1-10 所示。



图 1-9



图 1-10

1.3.2 安装 JDK

完成 JDK 工具的下载后, 即可安装 JDK, 其安装方法和步骤与其他软件的安装基本相同。

双击刚下载的 jdk-8u171-windows-x64.exe 版本安装程序, 显示 JDK 安装欢迎界面, 如图 1-11 所示。单击“下一步”按钮, 切换到定制安装界面, 在该界面中可以设置需要同时安装的工具, 以及程序的安装目录, 建议使用默认目录, 如图 1-12 所示。

单击“下一步”按钮, 即可开始安装 JDK, 显示安装进度, 如图 1-13 所示。当 JDK 安装完成后, 自动弹出安装 JRE(Java SE Runtime Environment, Java 运行环境)的选项, 可以修改其安装目录, 建议使用默认目录, 如图 1-14 所示。



图 1-11



图 1-12



图 1-13



图 1-14

单击“下一步”按钮，即可开始安装并显示安装进度，如图 1-15 所示。安装完成后，显示完成界面，如图 1-16 所示。在该界面中单击“后续步骤”按钮，在弹出的浏览器窗口中可以显示 JDK 的相关文档说明，单击“关闭”按钮，完成 JDK 的安装。



图 1-15



图 1-16

1.3.3 配置 Java 环境变量

完成 JDK 的安装后，还需要设置其在编译和运行时所需要使用的环境变量。

在操作系统桌面上的“计算机”图标上单击鼠标右键，在弹出的菜单中选择“属性”命令，弹出属性窗口，如图 1-17 所示。在窗口左侧单击“高级系统设置”选项，显示“系统属性”对话框中的“高级”选项卡，如图 1-18 所示。

单击“环境变量”按钮，弹出“环境变量”对话框，如图 1-19 所示。在该对话框中单击“系统变量”选项下方的“新建”按钮，弹出“新建系统变量”对话框，设置“变量名”为 JAVA_HOME，“变量值”为 C:\Program Files\Java\jdk1.8.0_171，如图 1-20 所示。



图 1-17



图 1-18



图 1-19

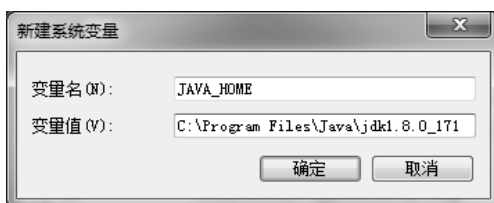


图 1-20

提示

这里所添加的名为 JAVA_HOME 的系统变量的“变量值”设置，实际上是 JDK 的安装目录，我们在安装 JDK 时采用了默认的安装目录，其默认会安装在 C:\Program Files 文件夹中。如果用户在安装 JDK 时修改了安装目录，则这里的“变量值”设置会有所不同。另外，如果所安装的 JDK 版本不同，其 JDK 安装目录的文件夹名称也会有所不同。

单击“确定”按钮，添加系统变量。在“环境变量”对话框上面的列表框中双击 Path 系统变量，弹出“编辑系统变量”对话框，在变量值的末尾添加一个变量值“%JAVA_HOME%\bin;”，如图 1-21 所示。单击“确定”按钮，完成 Path 系统变量的编辑。

技巧

变量值结尾的英文分号一定要写，如果原来的 Path 系统变量的变量值结尾没有分号，则用户需要添加一个分号，起到分隔的作用。

单击“系统变量”选项下方的“新建”按钮，弹出“新建系统变量”对话框，设置“变量名”为 CLASSPATH，“变量值”为“%JAVA_HOME%\bin;”，如图 1-22 所示。单击“确定”按钮，完成名为 CLASSPATH 的系统变量的添加。



图 1-21

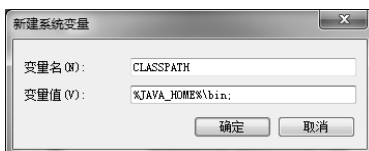


图 1-22

单击“确定”按钮，完成“环境变量”对话框的设置，单击“确定”按钮，关闭“系统属性”对话框，完成 Java 环境变量的配置。

完成 Java 环境变量的配置后，还需要测试 Java 环境变量的配置是否正确。在操作系统中单击左下角的“开始”按钮，在弹出的系统菜单的文本框中输入 cmd 命令，如图 1-23 所示。按 Enter 键，打开系统 DOS 窗口，如图 1-24 所示。



图 1-23

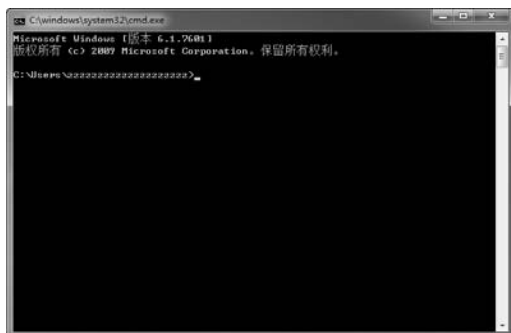


图 1-24

在 DOS 窗口中分别输入 java 和 javac 命令进行测试, 如果显示出如图 1-25 和图 1-26 所示的界面, 则说明 JDK 配置成功。

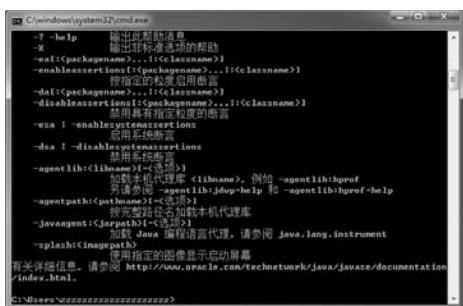


图 1-25



图 1-26

1.4 Tomcat 服务器的安装与配置

Tomcat 是一个免费的开源 Servlet 容器, 由 Apache 公司、原 Sun 公司和其他一些公司及个人共同开发。在本节中将向读者介绍 Tomcat 服务器的下载、安装和配置操作。

1.4.1 下载 Tomcat 服务器

由于 Java 的跨平台特性, 基于 Java 的 Tomcat 也具有跨平台性。用户可以从 Apache 官方网站免费下载 Tomcat 服务器的安装程序, Tomcat 提供了各种平台的版本供用户下载。

打开浏览器, 在地址栏中输入 Tomcat 服务器的官方地址 <http://tomcat.apache.org>, 进入 Tomcat 服务器的官方页面, 如图 1-27 所示。在该页面中显示了 Tomcat 服务器的相关介绍文字内容, 向下拖动页面, 选择需要下载的版本, 单击 Download 按钮, 如图 1-28 所示。

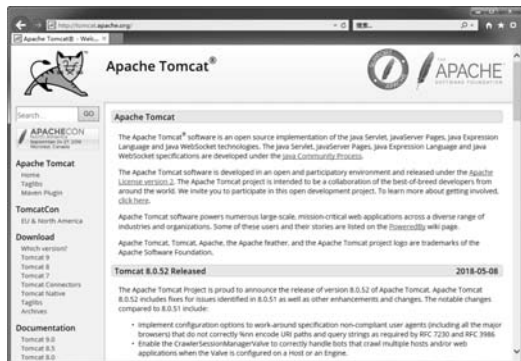


图 1-27



图 1-28



进入 Tomcat 8 版本的介绍页面，在 Quick Navigation 选项栏中单击 Browse 超链接，如图 1-29 所示。在弹出的窗口中显示 Tomcat 8 所包含的不同版本的服务器文件夹，这里选择 v8.0.52 版本，如图 1-30 所示。



图 1-29

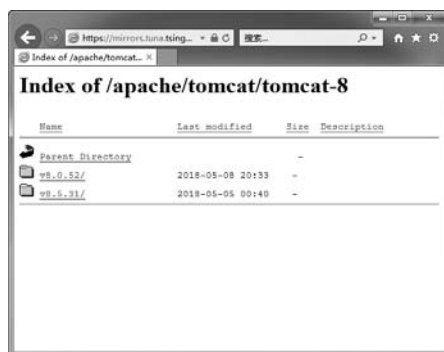


图 1-30

单击 v8.0.52 超链接，进入该版本 Tomcat 服务器的文件及版本说明页面，如图 1-31 所示。单击 bin 超链接，进入该文件夹中，为用户提供了多种针对不同平台的版本，如图 1-32 所示。

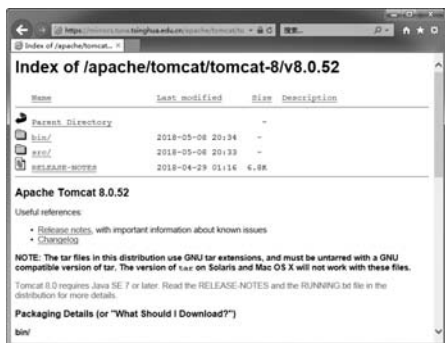


图 1-31

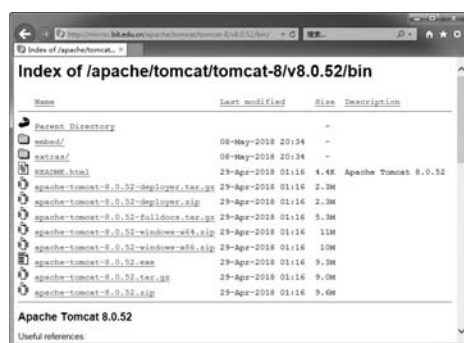


图 1-32

Tomcat 服务器包括安装版和解压版两种。如果用户选择解压版，则根据自己所使用的操作系统来选择下载对应的解压版 Tomcat 服务器。推荐下载安装版的 Tomcat 服务器，这里单击名称为 apache-tomcat-8.0.52.exe 超链接，即可下载安装版本的 Tomcat 服务器，如图 1-33 所示。下载完成后，可以在文件保存位置看到所下载的 apache-tomcat-8.0.52.exe 安装程序文件，如图 1-34 所示。



图 1-33



图 1-34

1.4.2 安装 Tomcat 服务器

在安装 Tomcat 服务器之前必须已经安装了 JDK，否则 Tomcat 服务器不能定位安装需要使用的 Java 运行环境 JRE。需要特别说明的是，Tomcat 服务器的 Web 服务默认端口是 8080。

双击刚下载的 apache-tomcat-8.0.52.exe 安装程序，显示安装欢迎界面，如图 1-35 所示。单

击 Next 按钮，显示该软件的安装许可协议，如图 1-36 所示。



图 1-35

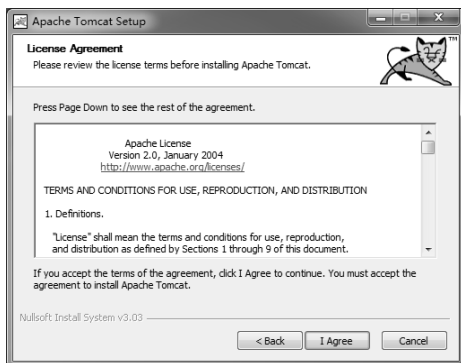


图 1-36

单击 I Agree 按钮，同意软件安装许可协议，切换到选择安装模式的界面，在下拉列表中选择 Full 选项，选择完全安装模式，如图 1-37 所示。单击 Next 按钮，切换到 Tomcat 服务器的基本设置界面，勾选 Create shortcuts for all users 复选框，其他保持默认设置，如图 1-38 所示。

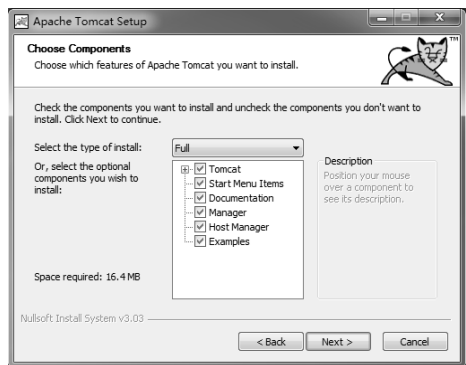


图 1-37

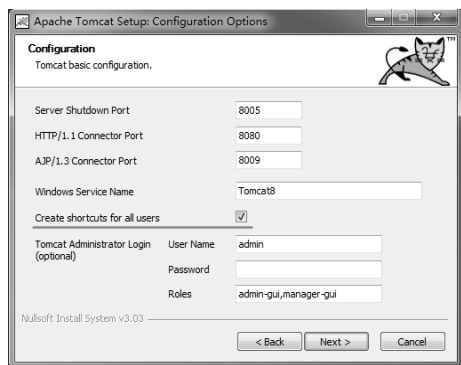


图 1-38

提示

在 Tomcat 服务器的基础设置界面中，Server Shutdown Port 选项用于设置 Tomcat 服务器的关机端口，默认为 8005；HTTP/1.1 Connector Port 选项用于设置 http 访问端口，默认为 8080；AJP/1.3 Connector Port 选项用于设置 AJP 协议端口，默认为 8009；Windows Service Name 选项用于设置服务器名称；Create shortcuts for all users 选项，选中该选项，表示所有用户都可以使用该 Tomcat 服务器；Tomcat Administrator Login(optional) 选项区，用于设置 Tomcat 服务器管理员的用户名和密码，默认为空。

单击 Next 按钮，显示设置 Java 虚拟机界面，安装程序能够自动定位到 Java 虚拟机的安装路径，如图 1-39 所示。单击 Next 按钮，切换到 Tomcat 服务器安装目录设置界面，可以设置 Tomcat 测试服务器的安装位置，如图 1-40 所示。

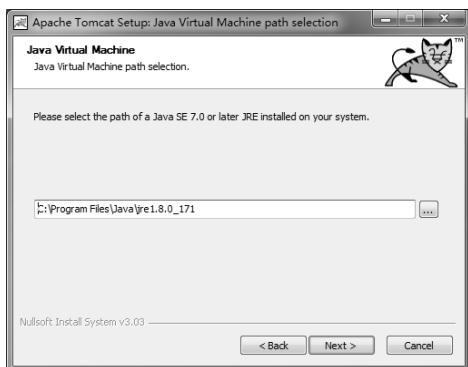


图 1-39

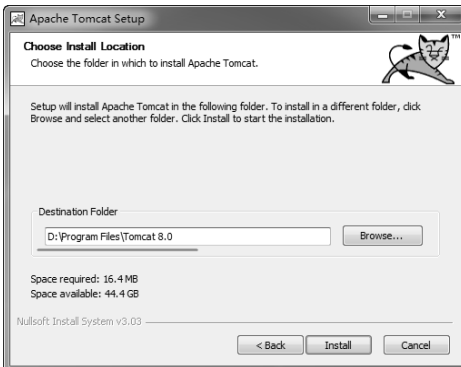


图 1-40



技巧



Tomcat 服务器默认的安装路径位于 C:\Program Files\Apache Software Foundation\ 目录中, 路径比较深, 用户可以将其安装到本地磁盘的根目录中, 便于查找和操作。

单击 Next 按钮, 即可开始安装 Tomcat 服务器, 显示安装进度, 如图 1-41 所示。安装完成后自动切换到安装完成界面, 如图 1-42 所示。

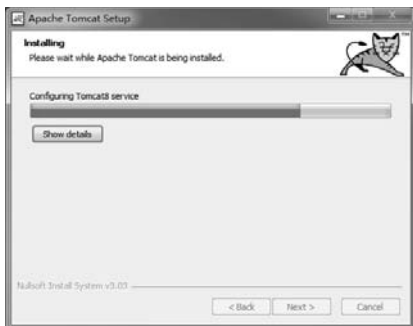


图 1-41

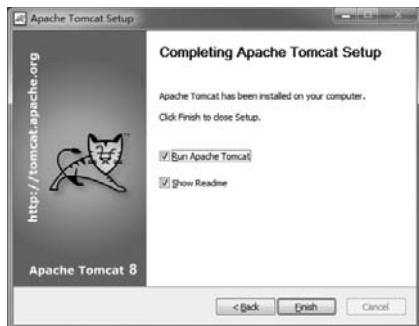


图 1-42

在安装完成界面中勾选 Run Apache Tomcat 复选框, 单击 Finish 按钮, 完成 Tomcat 服务器的安装, 并启动 Tomcat 服务器, 如图 1-43 所示。Tomcat 服务器启动完成后, 在操作系统桌面右下角的通知区域中可以看到 Tomcat 服务器启动图标, 如图 1-44 所示。

用户如果需要测试服务器是否安装成功, 可以打开浏览器窗口, 在地址栏中输入 <http://localhost:8080>, 访问 Tomcat 服务器的默认页面, 如果显示如图 1-45 所示的 Tomcat 服务器介绍页面, 则表示 Tomcat 服务器安装成功并且成功地启动服务。

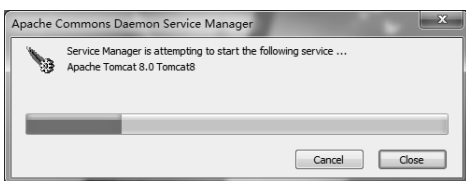


图 1-43



图 1-44

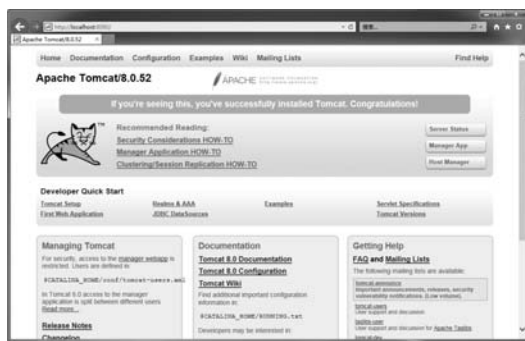


图 1-45

1.4.3 Tomcat 服务器的目录结构



完成 Tomcat 服务器的安装后, 在 Tomcat 服务器的安装目录中包含多个文件夹和文件, 首先需要了解各文件夹的意义和用途。例如, 本书的 Tomcat 服务器安装在 D 盘的 Program Files 文件夹中, 所以 Tomcat 服务器的主目录为 D:\Program Files\Tomcat 8.0, 打开该文件夹, 可以看到 Tomcat 服务器主目录中的文件夹, 如图 1-46 所示。

Tomcat 服务器主目录中各文件夹的意义和用途说明如表 1-1 所示。



图 1-46

表 1-1 Tomcat 服务器目录说明

文件夹	说明
bin	在该文件夹中存放启动和关闭 Tomcat 服务器的相关编辑文件
conf	在该文件夹中存放服务器的各种配置文件，包括 server.xml、web.xml 等
lib	在该文件夹中存放的 JAR 文件和类文件，能够被各目录中的 JSP 页面和 Tomcat 服务器系统程序访问
logs	在该文件夹中存放 Tomcat 服务器的日志档案
temp	该文件夹用于存储 Tomcat 服务器的各种临时文件
webapps	该文件夹用于存放 Web 应用程序文件，例如 JSP 应用程序、Servlet 应用程序和默认 Web 服务目录 ROOT
work	该文件夹用于存储 JSP 页面转换为 servlet 文件和字节码文件

在 Tomcat 服务器目录中双击 conf 文件夹，进入该文件夹中，在该文件夹中存放 Tomcat 服务器的相关配置文件，其中最重要的是 server.xml，如图 1-47 所示。使用记事本打开 server.xml 文件，如图 1-48 所示，在该文件中可以配置 Web 服务的端口、会话过期时间和虚拟主机等。



图 1-47

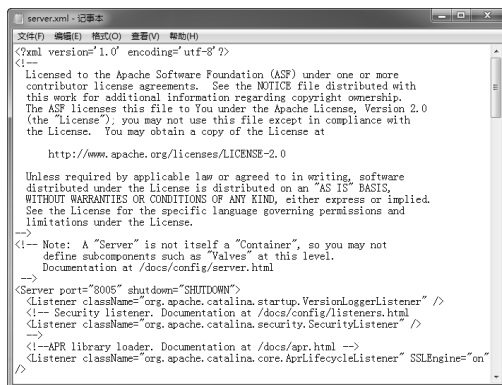


图 1-48

如果需要修改 Web 服务默认的端口，则可以在 server.xml 文件中按快捷键 Ctrl+F，弹出“查找”对话框，输入关键词 Connector，如图 1-49 所示。单击“查找下一个”按钮，即可在该配置文件中定位到 Web 服务默认端口的设置代码，如图 1-50 所示。



图 1-49



图 1-50

在此处将 port="8080" 修改为自己需要的端口即可，但是需要注意的是，对 server.xml 配置文件进行修改后必须重新启动 Tomcat 服务器，才能使修改后的配置生效。

提示

这里只是给读者演示如何修改 Tomcat 服务器的 Web 服务默认端口，本书依旧使用默认的 8080 端口，建议读者也采用默认访问端口。



1.4.4 Tomcat 服务器的基本操作



在完成 Tomcat 服务器的安装后会启动 Tomcat 服务，如果该图标没有在系统桌面右上角的通知区域中显示，用户可以在操作系统中执行“开始”>“所有程序”>Apache Tomcat 8.0 Tomcat8>Monitor Tomcat 命令，如图 1-51 所示。在系统桌面右下角的通知区域中会显示 Tomcat 服务器的图标**D**，如图 1-52 所示。



图 1-51



图 1-52

如果当前的 Tomcat 服务器处于关闭服务的状态(服务图标显示为**D**)，在该图标上单击鼠标右键，在弹出的快捷菜单中选择 Start service 命令，如图 1-53 所示，即可启动 Tomcat 服务。

如果当前的 Tomcat 服务器处于开启服务的状态(服务图标显示为**D**)，在该图标上单击鼠标右键，在弹出的快捷菜单中选择 Stop service 命令，如图 1-54 所示，即可关闭 Tomcat 服务。

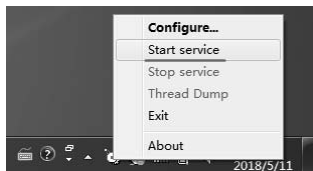


图 1-53

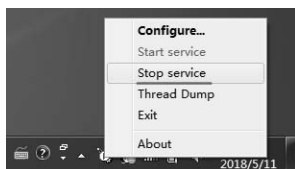


图 1-54

1.4.5 Tomcat 服务器中的 Web 应用程序文件架构



在 Tomcat 服务器中，所有的 JSP 页面都必须放置在 Tomcat 服务器根目录的 webapps 文件夹中，才能在 Tomcat 服务器中对该 JSP 页面进行解释执行。

打开 webapps 文件夹，可以看到在该文件夹中包含一个 ROOT 文件夹，如图 1-55 所示。该文件夹用来存放 Tomcat 默认的 index.jsp 页面，里面包含基本的配置，今后制作好的网页就可以放在其中。

ROOT 文件夹中包含一个 WEB-INF 文件夹，这是一个非常重要的目录，它代表构建网站的基本配置，如图 1-56 所示。今后自己开发的 Java 程序经编译后生成的 class 文件可以存放在该目录下，大大地扩展网页需要用到的函数。



图 1-55

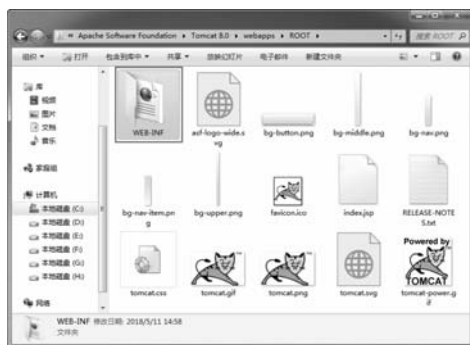


图 1-56

但是有人要问了，可不可以自己建立一个独立于 ROOT 目录之外的网站文件夹呢？答案是肯定的，只是需要配置好文件夹里面的设置。另外，创建的独立于 ROOT 目录之外的网站根目录，必须包含 WEB-INF 文件夹。我们并不建议初学者将站点文件夹设置在 ROOT 目录之外，因为这样操作

还需要进行另外的配置, 相对比较麻烦, 建议初学者在 ROOT 目录中创建网站根目录。

提示

一个完整的 Web 应用程序必须具备 WEB-INF 目录, 该目录中通常包含一个结构与部署说明文件 web.xml, 它是一个 XML 文件, 记载了一个应用程序所有的组成信息, 如果用户的网站程序中只有 JSP 和 HTML 两种文件, 则通常可以不用理会这个文件。

1.5 在 Dreamweaver 中创建 JSP 站点

根据前面介绍的内容就能在本地计算机中完成 JSP 开发环境的搭建, 已经可以开发 JSP 网站了, 但是如果使用 Dreamweaver 来开发 JSP 网站, 还需要在 Dreamweaver 中创建动态网站站点, 使 Dreamweaver 清除网站目录和测试服务器路径等信息。

1.5.1 站点文件夹规划

在开发制作网站之前, 需要将设计制作好的网站静态页面等相应的内容放置在本地计算机硬盘上, 为了方便使用 Tomcat 服务器对所开发的 JSP 页面进行测试, 需要将网站站点内容放置在 Tomcat 服务器的默认网站目录中 (Tomcat 服务器目录中的 webapps\ROOT), 本书的路径为 D:\Program Files\Tomcat 8.0\webapps\ROOT 目录, 再创建合理的文件夹来管理站点文件。

1. 合理的文件夹规划

在本地站点中应该使用文件夹合理构建文档的结构。首先为站点创建一个主要文件夹, 然后在其中再创建多个子文件夹, 最后将文档分类存放在相应的文件夹中。

例如, 可以在 images 文件夹中放置网站页面需要用到的图片, 在 style 文件夹中放置网站页面用到的 CSS 样式表文件, 在 admin 文件夹中放置后台管理程序页面等, 如图 1-57 所示。

2. 合理的文件命名

在网站的开发制作过程中, 可能需要创建较多的文件, 这就需要为各文件命名合理的文件名称。合理的文件命名主要有两个好处, 一是看到网页的文件名, 就可以大致了解该网页的主要内容; 二是当网站的规模变得很大时, 也可以很容易地找到相应的文件进行修改或更新。

合理的文件命名, 主要有以下几点要求。

- (1) 尽量使用短名称作为文件命名, 避免文件名称过长, 不便于记忆。
- (2) 避免使用中文的文件夹和文件名, 许多 Internet 服务器使用的是英文操作系统, 对中文的文件夹和文件名支持都不好, 而且浏览网站的用户也有可能使用的是英文操作系统, 中文的文件夹和文件名同样可能导致浏览错误或访问失败。
- (3) 建议在站点的规划和创建过程中, 全部使用小写的文件名称。很多服务器采用 UNIX 操作系统, 该操作系统是区分文件名称大小写的。

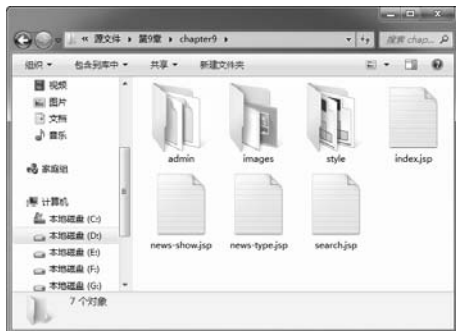


图 1-57

技巧

在创建 JSP 网站页面的过程中, 所有文件夹和文件名称一定要使用英文或者数字名称, 不能使用中文名称来命名, 否则会导致 Tomcat 服务器不能正常支持该站点。



3. 保持本地和远程站点为相同的结构

保持本地和远程站点为相同的结构是指在本地站点中规划设计的网站文件结构要与上传到 Internet 服务器上被人们浏览的网站文件结构相同。这样在本地站点上的文件夹和文件上的操作，都可以与远程站点上的文件夹和文件相对应。

1.5.2 JSP 测试服务器

要创建动态的 JSP 网站程序，就必须首先在 Dreamweaver 中定义 JSP 测试服务器站点，在这个步骤中需要告诉 Dreamweaver 关于网站的一些必要的基本信息。网站的完整规划与建设，能够让网页设计师和网站应用程序开发人员，甚至是后续负责维护更新的管理员，很轻易地熟悉与其相关的工作，以及找到必需的页面或文件。

实战 使用记事本制作 HTML 页面

最终文件：无 视频：视频\第1章\1-5-2.mp4



01 打开 Tomcat 服务器的默认网站根目录，本书所安装的 Tomcat 服务器的默认网站根目录是 D:\Program Files\Tomcat 8.0\webapps\ROOT，如图 1-58 所示。在默认网站根目录中创建站点文件夹，这里将文件夹命名为 chapter1，如图 1-59 所示。



图 1-58

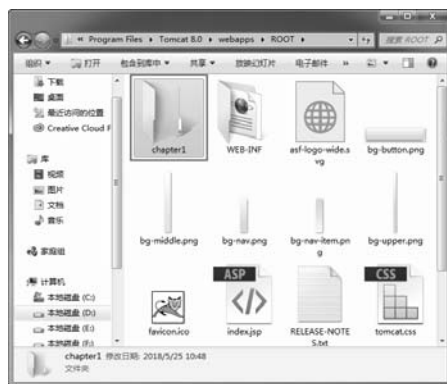


图 1-59

02 打开 Dreamweaver，执行“站点”>“新建站点”命令，弹出“站点设置对象”对话框，设置“站点名称”为 chapter1，“本地站点文件夹”为 D:\Program Files\Tomcat 8.0\webapps\ROOT\chapter1\，如图 1-60 所示。在对话框左侧单击“服务器”选项，切换到服务器选项设置界面，如图 1-61 所示。



图 1-60



图 1-61

03 单击“添加新服务器”按钮, 弹出服务器设置对话框，在“连接方法”下拉列表中选择“本地/网络”选项，对相关选项进行设置，如图 1-62 所示。单击“高级”按钮，切换到“高级”选项卡中，在“服务器模型”下拉列表中选择 JSP 选项，如图 1-63 所示。

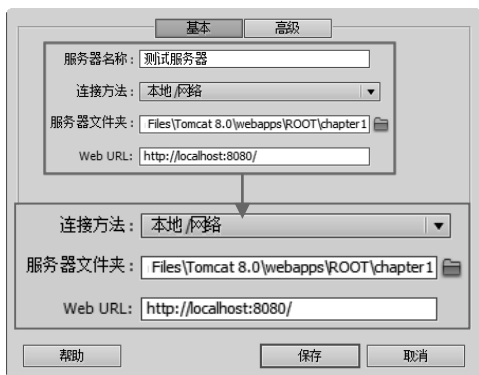


图 1-62

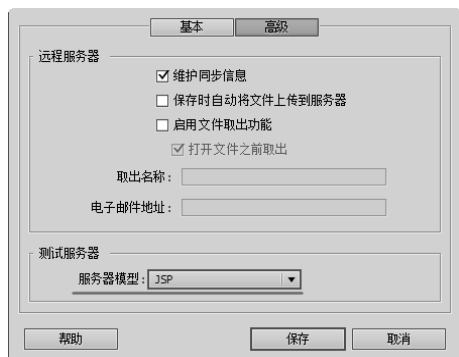


图 1-63

提示

“服务器文件夹”选项用于指定本地计算机中测试服务器的默认网站目录。Web URL 选项用于设置访问该站点的地址，Tomcat 服务器的默认网站访问地址是 http://localhost:8080/。

04 单击“保存”按钮，保存服务器选项设置，返回“站点设置对象”对话框，在刚刚添加的测试服务器选项中，勾选“测试”复选框，如图 1-64 所示。单击“保存”按钮，完成 JSP 网站测试服务器的定义，在“文件”面板中显示当前所创建的站点，如图 1-65 所示。



图 1-64



图 1-65

提示

在本地计算机中已经安装了 Tomcat 测试服务器，可以在本地计算机中测试 JSP 网页，所以不需要设置远程服务器信息，设置好本地信息和测试服务器之后，单击“保存”按钮，关闭“站点设置对象”对话框，这样就完成了 Dreamweaver 测试 JSP 网页服务器站点的创建和设置。

1.5.3 创建 JSP 页面

如果读者对网页编程有所了解，应该知道在编写网页程序时，可以将动态语言代码嵌入 HTML 代码中，嵌入在 HTML 代码中的动态程序语言代码需要使用特殊的符号进行包含，JSP 也是这样，可以将 JSP 程序代码直接嵌入 HTML 页面中执行。

在 Dreamweaver 中创建 JSP 页面与创建 HTML 页面一样方便和快捷，只需要创建一个 HTML 页面将其保存为 .jsp 文件即可。下面使用 Dreamweaver 创建一个 JSP 页面，并在该页面中输出相应的文字和当前系统时间。

实战 制作第一个 JSP 网页

最终文件：最终文件\第 1 章\chapter1\index.jsp

视频：视频\第 1 章\1-5-3.mp4





01 执行“文件”>“新建”命令，弹出“新建文档”对话框，选择 HTML 选项，如图 1-66 所示。单击“创建”按钮，新建 HTML 页面，执行“文件”>“保存”命令，将该文件保存在站点文件夹 chapter1 中，并命名为 index.jsp，如图 1-67 所示。



图 1-66



图 1-67

02 转换到代码视图中，可以看到页面的 HTML 代码，在 <title> 与 </title> 标签之间输入页面的标题，如图 1-68 所示。在 <body> 与 </body> 标签之间输入相应的 JSP 代码，如图 1-69 所示。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>制作第一个JSP网页</title>
</head>

<body>
</body>
</html>
```

图 1-68

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>制作第一个JSP网页</title>
</head>

<body>
<%
out.print("<center><h1>欢迎学习JSP网站开发! </h1>
<h2>当前的系统时间为: </h2></center>");
out.println("<hr>");
%>
</body>
</html>
```

图 1-69

提示

嵌入 HTML 代码中的 JSP 程序代码必须使用“<%”和“%>”包含，JSP 中的每个代码行都必须以分号结束，分号是一种分隔符，用于把指令集区分开来。

03 执行“文件”>“保存”命令，保存该页面。在浏览器地址栏中输入地址 localhost:8080/chapter1/index.jsp，测试该 JSP 页面，发现页面中输出的中文字符都会显示为乱码，如图 1-70 所示。返回页面代码中，在所有代码之前添加相应的 JSP 字符声明设置代码，如图 1-71 所示。



图 1-70

```
<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@ page contentType="text/html; charset=utf-8"%>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>制作第一个JSP网页</title>
</head>
```

图 1-71

提示

此处是在页面头部所添加的代码中，pageEncoding 属性设置的是 JSP 文件本身在本地保存时的编码方式；contentType 属性中的 charset 设置的是服务器发送网页内容给客户端时所使用的编码方式。

04 保存该 JSP 页面，在测试服务器中预览该 JSP 页面，可以看到 JSP 页面执行的效果，正确输出中文字符，如图 1-72 所示。返回网页代码中，在页面头部添加代码，导入 java.util.Date 时间类和 java.text.SimpleDateFormat 简单时间格式类，如图 1-73 所示。



图 1-72

```
<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.util.Date"%>
<%@ page import="java.text.SimpleDateFormat"%>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>制作第一个JSP网页</title>
</head>
```

图 1-73

05 在 <body> 与 </body> 标签之间添加输出当前系统日期时间的 JSP 代码，如图 1-74 所示。返回 Dreamweaver 设计视图中，可以看到 JSP 代码在设计视图中显示为 JSP 代码图标，如图 1-75 所示。

```
<body>
<%
out.print("<center><h1>欢迎学习JSP网站开发! </h1><h2>当前的系统时间为: </h2></center>");
out.println("<br>");
%>
<%
SimpleDateFormat date=new SimpleDateFormat("yyyy-MM-dd H:m:s");
String postdate=date.format(new Date());
%>
<center><%=postdate%></center>
</body>
```

图 1-74



图 1-75

提示

从第一次访问一个 JSP 页面开始，到这个 JSP 页面的内容被发送到客户端显示，这个 JSP 页面需要经过 3 次编码转换。

第一阶段是 JSP 编辑成 .java，它会根据 pageEncoding 属性的设置来读取 JSP，结果是由指定的编辑方式翻译成统一的 UTF-8 Java 源码（即 .java），如果 pageEncoding 设置错了，或者没有进行设置，就会导致出现中文乱码。

第二阶段是由 JavaC 的 Java 源码至 Java byteCode 的编译，无论 JSP 页面编写时是什么编码方式，经过这个阶段的结果全部是 UTF-8 的 encoding 的 Java 源码。JavaC 使用 UTF-8 的 encoding 读取 Java 源码，编译成 UTF-8 encoding 的二进制码（即 .class），这是 JVM 对常数字串在二进制码（Java Encoding）内表达的规范。

第三阶段是 Tomcat 服务器载入和执行阶段二得到的 Java 二进制码，向客户端输出结果，也就是客户端所看到的，这时隐藏在阶段一和阶段二的参数 contentType 就发挥了功效。

技巧

因为在页面中通过 Java 类来生成系统时间，所以首先要导入 java.util.Date 时间类和 java.text.SimpleDateFormat 简单时间格式类，然后才能在代码中生成时间实例 date 并调用 format 时间格式化函数。另外，在 JSP 代码中可以使用 <%= 变量名 %> 的方式来输出变量的值。

06 完成该 JSP 页面的制作，完整的代码如下。

```
<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.util.Date"%>
<%@ page import="java.text.SimpleDateFormat"%>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>制作第一个 JSP 网页</title>
</head>
```



```
<body>
<%
    out.print("<center><h1> 欢迎学习 JSP 网站开发! </h1><h2> 当前的系统时间为: </h2>
</center>");
    out.println("<hr>");
    %>
<%
    SimpleDateFormat date=new SimpleDateFormat("yyyy-MM-dd H:m:s");
    String postdate=date.format(new Date());
    %>
    <center><%=postdate%></center>
</body>
</html>
```

提示

JSP 代码被嵌入 HTML 代码中, 必须被 JSP 服务器编译解析后, 将解析后的结果输出到客户端的浏览器中, 才能正常显示 JSP 页面的结果。

07 执行“文件”>“保存”命令, 保存该 JSP 页面, 打开浏览器窗口, 在地址栏中输入地址 localhost:8080/chapter1/index.jsp, 可以看到该 JSP 页面的执行结果, 如图 1-76 所示。在网页上单击鼠标右键, 在弹出的快捷菜单中选择“查看源”命令, 然后在弹出的对话框中可以看到该 JSP 被服务器编译执行后的代码全部是静态网页代码, 如图 1-77 所示。



图 1-76



图 1-77

提示

在对 JSP 网页进行测试之前, 还需要确保 Tomcat 服务已经正常启动, 否则服务器将无法解析 JSP 网页代码。

技巧



只要在 Dreamweaver 中创建站点时正确地设置了 JSP 的测试服务器, 那么在测试 JSP 网页时也可以直接使用 Dreamweaver 中的预览功能, 而不需要在浏览器地址栏中手动输入测试页面地址。

第2章

JSP 基础语法



JSP(Java Server Page) 是运行于服务器端的脚本语言之一，是 Java 阵营中最具有代表性的解决方案。JSP 是由原 Sun 公司倡导、许多公司参与建立的一种动态网页技术。本章将介绍一些 JSP 的基础语法，包括 JSP 的数据类型与变量、Java 类的使用方法、运算符与表达式、流程控制语句等，通过学习这些基础知识使读者能够更加深入地了解 JSP。

本章知识点：

- 理解 JSP 的工作原理
- 理解并掌握 JSP 的基础语法
- 了解 Java 数据类型与变量
- 了解 Java 类的常用方法
- 理解并掌握 Java 中各种运算符的使用方法
- 掌握 Java 中流程控制语句的使用方法
- 了解数组和字符串类的使用

2.1

了解 JSP



使用 JSP 技术，不仅能制作出像 HTML 一样的静态网页，还能制作包含动态数据的网页。JSP 网页是在静态 HTML 文件中加入 Java 脚本程序 (Scriptlet) 和 JSP 标记 (Tag) 构成的，尤其适合动态网站的开发。

2.1.1 JSP 概述



在传统的 HTML 页面 (*.html、*.htm) 中加入 Java 程序片段和 JSP 标记 (Tag)，就构成了 JSP 网页 (*.jsp)，Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的 Java 程序片段，然后将执行结果以 HTML 格式返回给客户。

Java 程序片段可以操作数据库，可以重定向网页，可以发送 E-mail 等，这就是创建动态网站所需要的功能。所有程序操作都是在服务器端执行，网络上传送给客户端的仅仅是程序执行后的结果，对客户端浏览器几乎没有任何的要求。

JSP 与 Microsoft 的 ASP 技术非常相似，ASP 的编程语言是 VBScript 之类的脚本语言，JSP 使用的是 Java，这是两者最明显的区别。此外，ASP 与 JSP 还有一个更本质的区别，即两种语言引擎使用完全不同的方式来处理页面中嵌入的程序代码。在 ASP 语言中，VBScript 脚本代码被 ASP 引擎解释执行，网页每一次请求，ASP 引擎必须解释执行一次；在 JSP 语言中，Java 代码被编译成 Servlet 并由 Java 虚拟机执行，这种编译操作仅在对 JSP 页面的第一次请求时发生。

事实证明，Java Servlet 是一种开发 Web 应用的理想构架。JSP 以 Servlet 技术为基础，又在许多方面做出改进。利用跨平台运行的 Java Bean 组件，JSP 为分离处理逻辑与显示样式提供了卓越的解决方案。