

 大学计算机规划教材

JSP

大学实用教程(第2版)

◆ 耿祥义 张跃平 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

大学计算机规划教材

JSP 大学实用教程

(第2版)

耿祥义 张跃平 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

JSP 是一种动态网页技术标准, 可以建立安全、跨平台的先进动态网站。

本书详细讲解了 JSP 的重要内容, 特别注重结合实例讲解一些难点和关键技术。全书共分 11 章, 内容包括: JSP 简介、JSP 页面、JSP 标记、内置对象、JSP 与 JavaBean、文件操作、数据库操作、Java Servlet、MVC 设计模式, 以及应用实例——网上书城。本书所有知识都结合具体实例进行介绍, 力求详略得当, 突出 JSP 在开发 Web 动态网站方面的强大功能, 使读者快速掌握和运用 JSP 的编程技巧。

本书可以作为高等院校计算机及相关专业的教材, 也适合自学者及网站开发人员参考使用。

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 大学实用教程 / 耿祥义, 张跃平主编. —2 版. —北京: 电子工业出版社, 2012.1

大学计算机规划教材

ISBN 978-7-121-14365-6

I. ① J… II. ① 耿… ② 张… III. ① JAVA 语言—网页制作工具—高等学校—教材

IV. ① TP312 ② TP393.092

中国版本图书馆 CIP 数据核字 (2011) 第 166208 号

策划编辑: 章海涛

责任编辑: 章海涛

特约编辑: 何 雄/张 玉

印 刷: 北京东光印刷厂

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: 18.5 字数: 474 千字

印 次: 2012 年 1 月第 1 次印刷

定 价: 32.00 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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

第 2 版前言

本书是《JSP 大学实用教程》的第 2 版，更新了部分例子和内容，增加了一个实训内容，即第 11 章（实训二：网上书城）。本次修订特别注重教材的可读性和实用性，许多例题都经过精心的考虑，既能帮助理解知识，同时又具有启发性。

全书共分 11 章，分别讲解了 JSP 页面、JSP 标记、内置对象、JSP 与 JavaBean、文件操作、数据库操作、Java Servlet、MVC 设计模式等重要内容。

第 1 章介绍 JSP 重要性，对 Tomcat 服务器的安装与配置进行了详细介绍。

第 2 章讲解 JSP 页面的基本构成以及 JSP 页面的运行原理。

第 3 章讲述常用的 JSP 标记，特别对 Tag 标记给予详细的讲解，使读者认识到代码复用的重要性。

第 4 章讲解内置对象，特别强调了这些内置对象在 JSP 应用开发中的重要性，结合实例使读者掌握内置对象的用法。

第 5 章是 JSP 技术中很重要的内容，即怎样使用 JavaBean 分离数据的显示和处理，这一章给出了许多有一定应用价值的例子。

第 6 章主要讲解怎样使用 Java 中的输入/输出流实现文件的读写操作，在实例上特别强调怎样使用 JavaBean 实现文件的读写操作。

第 7 章涉及的内容是数据库，也是 Web 应用开发的非常重要的一部分内容，特别介绍了各种数据库的连接方式。

第 8 章讲解 Java Servlet，对 Servlet 对象的运行原理给予了细致的讲解，许多例子都是大多数 Web 开发中经常使用的模块。

第 9 章对 Java Servlet 在 MVC 开发模式中的地位给予了重点介绍，并按着 MVC 模式给出了易于理解 MVC 设计模式的例子。

第 10 章是一个完整的网站，完全按着 MVC 模式开发设计，其目的是使读者掌握一般 Web 应用中常用基本模块的开发方法。

第 11 章是一个完整的网站，完全按着 JSP+Tag 模式开发设计，其目的是使读者掌握一般 Web 应用中常用基本模块的开发方法。

希望本书能对读者学习 JSP 有所帮助，并请读者批评指正。

本书为任课教师提供配套的教学资源（包含电子教案、例题源代码、部分习题解答），需要者可登录到[华信教育资源网](http://www.hxedu.com.cn)（<http://www.hxedu.com.cn>），注册之后进行下载。

读者反馈：unicode@phei.com.cn，xygeng0629@sina.com。

作 者

目 录

第 1 章 JSP 简介	1
1.1 什么是 JSP	1
1.2 Tomcat 服务器的安装与配置	1
1.3 测试 JSP 页面	4
1.4 设置 Web 服务目录	5
1.5 设置端口号	7
习题 1	7
第 2 章 JSP 页面	8
2.1 JSP 页面的基本结构	8
2.2 JSP 的运行原理	9
2.3 JSP 页面的成员变量和方法	12
2.4 JSP 页面中的 Java 程序片	13
2.5 JSP 页面中的 Java 表达式	15
2.6 JSP 中的注释	16
2.7 在 JSP 页面中使用 HTML 标记	17
习题 2	23
第 3 章 JSP 标记	25
3.1 指令标记 page	25
3.2 指令标记 include	28
3.3 动作标记 include	29
3.4 动作标记 param	31
3.5 动作标记 forward	31
3.6 动作标记 useBean	33
3.7 Tag 文件与 Tag 标记	33
习题 3	39
第 4 章 内置对象	40
4.1 request 对象	40
4.2 response 对象	46
4.3 session 对象	51
4.4 out 对象	60
4.5 application 对象	61
习题 4	64

- 第 5 章 JSP 与 JavaBean 65
 - 5.1 编写和使用 JavaBean 66
 - 5.1.1 编写 bean 66
 - 5.1.2 使用 bean 67
 - 5.2 获取和修改 bean 的属性 71
 - 5.2.1 动作标记 getProperty 71
 - 5.2.2 动作标记 setProperty 73
 - 5.3 bean 的辅助类 77
 - 5.4 举例 79
 - 5.4.1 三角形 bean 79
 - 5.4.2 四则运算 bean 80
 - 5.4.3 猜数字 bean 82
 - 5.4.4 时间 bean 84
 - 5.4.5 日历 bean 87
 - 5.4.6 播放幻灯片 bean 90
 - 习题 5 91
- 第 6 章 JSP 中的文件操作 93
 - 6.1 获取文件信息 94
 - 6.2 创建、删除 Web 服务目录 96
 - 6.3 读写文件 98
 - 6.3.1 读写文件的常用流 98
 - 6.3.2 读取文件 100
 - 6.3.3 按行读取 103
 - 6.3.4 写文件 105
 - 6.4 标准化考试 106
 - 6.5 文件上传 111
 - 6.6 文件下载 113
 - 习题 6 114
- 第 7 章 在 JSP 中使用数据库 115
 - 7.1 SQL Server 2000 数据库管理系统 116
 - 7.2 JDBC 117
 - 7.3 数据库连接的常用方式 118
 - 7.3.1 JDBC-ODBC 桥接器 118
 - 7.3.2 使用纯 Java 数据库驱动程序 122
 - 7.4 查询操作 125
 - 7.4.1 顺序查询 126
 - 7.4.2 随机查询 132
 - 7.4.3 条件查询 136

7.4.4 排序查询	141
7.4.5 模糊查询	143
7.5 更新、添加与删除操作	145
7.6 使用预处理语句	153
7.6.1 预处理语句优点	153
7.6.2 使用通配符	155
7.7 基于 <code>CachedRowSet</code> 分页显示记录	158
7.8 常见数据库的连接	163
7.8.1 连接 Oracle 数据库	163
7.8.2 连接 MySQL 数据库	164
习题 7	170
第 8 章 Java Servlet 基础	171
8.1 Servlet 对象的创建与使用	172
8.1.1 <code>HttpServlet</code> 类	172
8.1.2 部署 Servlet	172
8.1.3 运行 Servlet	174
8.2 Servlet 工作原理	175
8.3 通过 JSP 页面调用 Servlet	176
8.4 Servlet 的共享变量	179
8.5 <code>doGet()</code> 方法和 <code>doPost()</code> 方法	180
8.6 重定向与转发	183
8.7 会话管理	186
8.7.1 获取用户的会话	186
8.7.2 猜数字	188
习题 8	191
第 9 章 基于 Servlet 的 MVC 模式	192
9.1 MVC 模式介绍	193
9.2 JSP 中的 MVC 模式	193
9.3 模型的生命周期与视图更新	194
9.4 MVC 模式的简单实例	196
9.4.1 计算三角形的面积	196
9.4.2 四则运算	199
9.4.3 读取文件	203
9.4.4 查询数据库	206
习题 9	212
第 10 章 实训一：会员管理系统	213
10.1 系统模块构成	213

10.2	数据库设计	214
10.3	系统管理	214
10.4	会员注册	217
10.5	会员登录	223
10.6	上传照片	227
10.7	浏览会员	232
10.8	修改密码	240
10.9	修改注册信息	244
10.10	退出登录	250
第 11 章	实训二：网上书城	252
11.1	系统主要模块	252
11.2	数据库设计与连接	253
11.3	系统管理	255
11.4	会员注册	257
11.5	会员登录	259
11.6	浏览图书信息	262
11.7	查询图书	265
11.8	查询购物车	268
11.9	订单预览	271
11.10	确认订单	273
11.11	查询订单	276
11.12	查看图书摘要	278
11.13	修改密码	280
11.14	修改注册信息	282
11.15	退出登录	286

第 1 章

JSP 简介

本章导读

- ☆ 知识点：了解 JSP 的来历以及在开发动态网站上的优势。掌握 Tomcat 服务器的安装与配置。
- ☆ 重点：Tomcat 服务器的安装与配置。
- ☆ 难点：学习怎样设置 Web 服务目录。
- ☆ 关键实践：上机编写、保存、运行一个简单的 JSP 页面。

1.1 什么是 JSP

Java 语言以不依赖于平台、面向对象、安全等优良特性成为网络程序设计语言中的佼佼者。目前，许多与 Java 有关的技术得到了广泛的应用和认可，JSP（Java Server Pages）技术就是其中之一。JSP 是基于 Java 语言的一种 Web 应用开发技术，可以建立安全、跨平台的先进动态网站。许多 Web 网站都使用了 JSP 技术。利用 JSP 技术创建的 Web 应用程序可以实现动态页面与静态页面分离，便于 Web 应用程序的扩展和维护。由于 JSP 是基于 Java 语言的 Web 技术，相对其他 Web 技术，JSP 具有脱离硬件平台束缚、编译后运行等优点，已成为 Internet 上的主流 Web 技术之一。

1.2 Tomcat 服务器的安装与配置

网络通信中最常见的模式是 B/S 模式，即需要获取信息的用户使用浏览器向某个服务器发出请求，服务器进行必要的处理后，将有关信息发送给服务器。在 B/S 模式中，服务器上必须有所谓的 Web 服务软件和 Web 应用程序，Web 服务软件负责处理客户对 Web 应用程序的请求，并负责运行管理 Web 应用程序，以满足客户对信息的请求。因此，学习 JSP 需要安装一个支持 JSP 的 Web 服务软件，这样的软件也称为 JSP 引擎。将安装 JSP 引擎的计算机称

为一个支持 JSP 的 Web 服务器。支持 JSP 的 Web 服务器负责运行 JSP，并将运行结果返回给用户，有关 JSP 的运行原理将在本书 2.2 节中讲解。

目前，比较常用的 JSP 引擎包括 Tomcat、JRun 和 Resin，以 Tomcat 的使用最广泛。Tomcat 软件是一个免费的开源 JSP 引擎，也称为 Tomcat 服务器。Tomcat 服务器由 Apache 和 Sun 公司共同开发而成，可以登录到 <http://jakarta.apache.org/tomcat> 免费下载 Tomcat 6.0。登录之后，先在 Download 页面中选择 apache-tomcat-6.0.13，然后在 Binary Distributions 的 Core 页面中选择 Zip 或 Windows Service Installer 即可。如果选择 Zip，将下载 apache-tomcat-6.0.13.zip 文件；如果选择 Windows Service Installer，将下载 apache-tomcat-6.0.13.exe 文件。

本书的重点在于学习 JSP 本身，不涉及比较各种 JSP 引擎的优缺点，因此选择 Tomcat 服务器来学习 JSP。本节主要介绍 Windows 2000/XP 操作系统下 Tomcat 服务器的安装配置。

1. 安装 JDK

安装 Tomcat 之前，首先安装 JDK，这里安装 Sun 公司的 JDK 1.6。假设 JDK 的安装目录是 D:\jdk1.6。安装 JDK 之后需要进行几个环境变量的设置。对于 Windows XP/2000，右键单击“我的电脑”，在弹出的快捷菜单中选择“属性”命令，弹出“系统特性”对话框，再单击其中的“高级选项”，然后单击按钮“环境变量”，分别添加如下系统环境变量：

变量名: Java_Home	变量值: D:\jdk1.6
变量名: Classpath	变量值: D:\jdk1.6\jre\lib\rt.jar;.;
变量名: Path	变量值: D:\jdk1.6\bin

如果曾经设置过环境变量 Java_Home、Classpath 和 Path，可单击该变量进行编辑操作，将需要的值加入即可，如图 1.1、图 1.2 和图 1.3 所示。

2. 安装与启动 Tomcat 服务器

(1) apache-tomcat-6.0.13.zip 的安装

将下载的 apache-tomcat-6.0.13.zip 解压到磁盘某个分区，如解压到 D:\，解压缩后将出现如图 1.4 所示的目录结构。

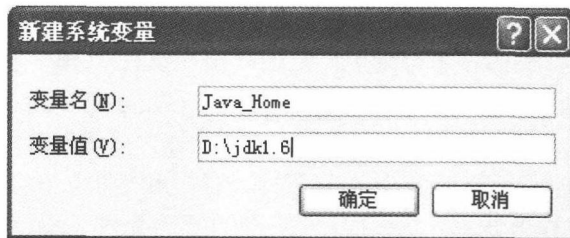


图 1.1 设置 Java_Home

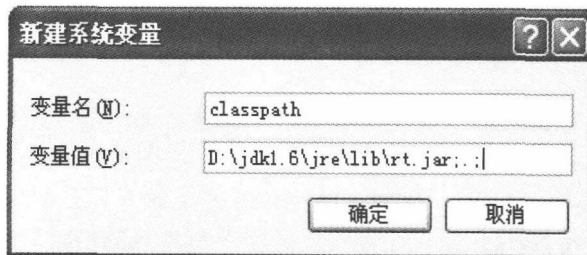


图 1.2 设置 Classpath

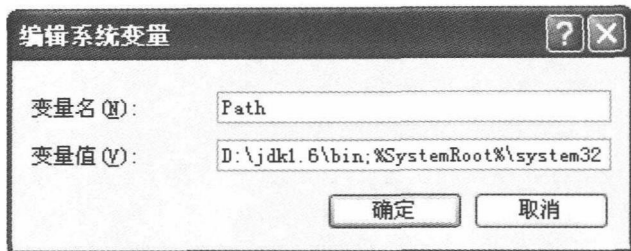


图 1.3 编辑 Path



图 1.4 Tomcat 服务器的目录结构

执行 Tomcat 安装根目录中 bin 文件夹中的 startup.bat 或 tomcat6.exe 来启动 Tomcat 服务器。执行 startup.bat，启动的 Tomcat 服务器会占用一个 MS-DOS 窗口（如图 1.5 所示），如果关闭当前 MS-DOS 窗口，将关闭 Tomcat 服务器。建议使用 startup.bat 启动 Tomcat 服务器，以确保 Tomcat 服务器使用的是 Java_Home 环境变量设置的 JDK。

```

信息: Starting Servlet Engine: Apache Tomcat/6.0.13
2009-1-19 10:14:09 org.apache.coyote.http11.Http11Protocol start
信息: Starting Coyote HTTP/1.1 on http-8080
2009-1-19 10:14:09 org.apache.jk.common.ChannelSocket init
信息: JK: ajp13 listening on /0.0.0.0:8009
2009-1-19 10:14:09 org.apache.jk.server.JkMain start
信息: Jk running ID=0 time=0/63 config=null
2009-1-19 10:14:09 org.apache.catalina.startup.Catalina start
信息: Server startup in 1506 ms

```

图 1.5 启动 Tomcat 服务器

(2) 测试 Tomcat 服务器

在浏览器的地址栏中输入“http://localhost:8080”或“http://127.0.0.1:8080”，会出现如图 1.6 所示的 Tomcat 服务器的测试页面。

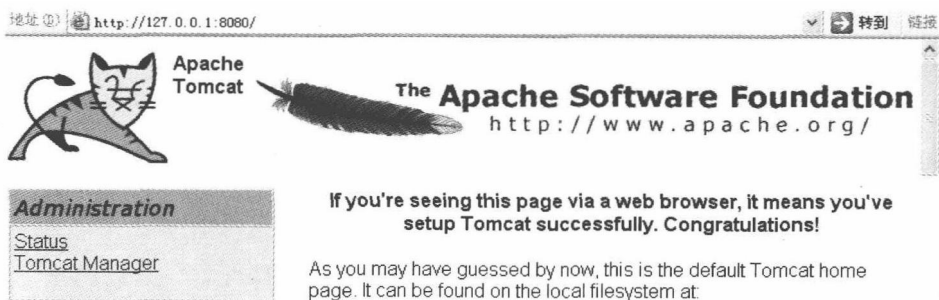


图 1.6 测试 Tomcat 服务器

注意：Tomcat 服务器默认占用 8080 端口，如果 Tomcat 所使用的端口已被占用，Tomcat 服务器将无法启动，有关端口号的配置稍后讲解。

(3) 文件 apache-tomcat-6.0.13.exe 的安装

文件 apache-tomcat-6.0.13.exe 是针对 Windows 的 Tomcat 服务器的，安装后形成的目录结构与 apache-tomcat-6.0.13.zip 完全相同。

双击下载的 apache-tomcat-6.0.13.exe 文件，将出现“安装向导”界面，单击【Next】按钮，接受授权协议后，将出现选择安装方式界面，从中选择“Normal”、“Minimun”、“Custom”或“Full”等安装方式。比如，选择安装方式为“Full”后单击【Next】按钮，将出现选择安

装目录界面，从中可以给出安装 Tomcat 的目录，如输入安装目录为 E:\Tomcat6.0。在选择安装目录时，最好不要使用该界面给出的默认目录，以方便今后使用 Tomcat。在选择安装目录界面中选择安装目录后，单击【Next】按钮，将出现设置端口号和管理密码设置界面。Tomcat 服务器必须占用一个端口号，以便与其他网络程序相区分。该界面提供的 Tomcat 服务器占用的默认端口号是 8080，在安装 Tomcat 时，可以在该界面设置 Tomcat 服务器所占用的端口号，但最好不要使用 1~1024 之间的端口号，以免与其他网络程序发生冲突。另外，在该界面中可以输入管理密码，以便以后管理 Tomcat 服务器。在该界面中，将 Tomcat 服务器的端口设置为默认端口 8080，管理密码设置为 123456。

对于 Windows 2000/NT/XP，在安装 Tomcat 成功时，安装程序会提示用户选择“即刻启动 Tomcat”或“稍后启动 Tomcat”。如果没有选择“即刻启动 Tomcat”，可以通过“开始”→“所有程序”→“Apache tomcat 6.0”→“start Tomcat”启动 Tomcat，也可直接执行 Tomcat 安装目录 bin 文件夹中的 startup.bat 或 tomcat6.exe 来启动 Tomcat，如 E:\Tomcat6.0\bin\tomcat6.exe。注意：若 Tomcat 所使用的端口已被占用，Tomcat 将无法启动。

1.3 测试 JSP 页面

Tomcat 正确启动后，需要测试它是否是一个 JSP 引擎，因为 Tomcat 默认是一个 HTML 引擎。在 1.2 节中，在浏览器的地址栏中输入“http://127.0.0.1:8080”后，Tomcat 将 index.html 的超文本文件发送给浏览器，该超文本文件中并没有 JSP 技术所涉及的内容，因此可以看见 Tomcat 测试页，但并不能保证 Tomcat 的安装就完全正确无误。

我们将使用一个简短的 JSP 页面来测试 Tomcat，在以后的章节里会详细讲解编写 JSP 页面的语法。

简单地说，一个 JSP 页面除了普通的 HTML 标记符外，还可以使用标记符号“<%”、“%>”加入 Java 程序片。JSP 页面文件的扩展名是 .jsp，文件的名称必须符合标识符规定，即名字可以由字母、下划线、美元符号和数字组成，并且第一个字符不能是数字字符。需要特别注意的是，文件名字中的字母是区分大小写的，如 Boy 和 boy 是不同的标识符。

为了明显地区分普通的 HTML 标记和 Java 程序片以及 JSP 标签，我们用大写字母书写普通的 HTML 标记符号。

可以用“记事本”或其他文本编辑器编辑 JSP 页面的源文件。如果使用“记事本”编辑 JSP 页面文件，在保存文件时必须将“保存类型”选择为“所有文件”，将“编码”选择为“ANSI”。如果在保存文件时系统总在文件名后加上“.txt”后缀，那么在保存文件时可以将文件名用“”括起，如图 1.7 所示。

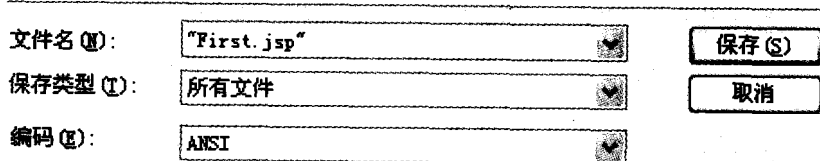


图 1.7 JSP 文件的保存

必须将编写好的 JSP 页面文件保存到 Tomcat 服务器的一个 Web 服务目录中。如果 Tomcat 的安装目录是 E:\Tomcat6.0，那么 Tomcat 服务器的 Web 服务目录的根目录是

E:\Tomcat6.0\webapps\Root。

为了测试 JSP 页面，可以将编写好的 JSP 页面文件保存或复制到上述根目录中。

【例 1-1】 制作一个简单的 JSP 页面 first.jsp，将其保存到根目录中。在浏览器的地址栏中输入“http://127.0.0.1:8080/first.jsp”，如果 Tomcat 服务器和 JDK 都安装正确，就会出现如图 1.8 所示的页面。

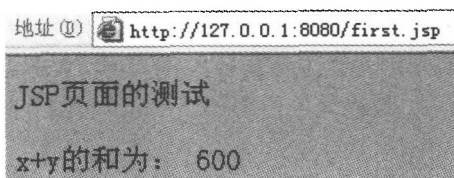


图 1.8 测试 JSP 页面

如果出现错误提示：

```
org.apache.jasper.JasperException: Unable to compile class for JSP
```

请关闭 Tomcat 服务器，然后运行 startup.bat（该文件在 Tomcat 安装目录的 bin 文件夹中），重新启动 Tomcat 服务器，以确保 Tomcat 服务器使用的 JDK 是 Java_Home 设置的 JDK。

first.jsp

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML><BODY bgcolor=cyan>
<FONT size=4>
<P>JSP 页面的测试
  <% int x=100,y=500,z;
    z=x+y;
    %>
<P>x+y 的和为: <%= z %>
</FONT></BODY>
</HTML>
```

1.4 设置 Web 服务目录

为了让客户通过浏览器访问一个 Tomcat 服务器上的 JSP 页面，必须将编写好的 JSP 页面文件保存到该 Tomcat 服务器的某个 Web 服务目录中。

1. 根目录

如果 Tomcat 服务器的安装目录是 E:\Tomcat6.0，则 Tomcat 的 Web 服务目录的根目录是 E:\Tomcat6.0\webapps\Root。

用户如果准备访问根目录中的 JSP 页面，在浏览器中输入 Tomcat 服务器的 IP 地址（或域名）、端口号和 JSP 页面的名字即可（必须省略 Web 根目录的名字），如 Tomcat 服务器的 IP 地址是 192.168.1.200，根目录中存放的 JSP 页面的名字是 A.jsp，那么用户在浏览器输入的内容是“http://192.168.1.200:8080/A.jsp”。

也许用户没有为 Tomcat 服务器所在的机器设置过一个有效的 IP 地址，为了调试 JSP 页面，可以打开 Tomcat 服务器上的浏览器，在地址栏中输入“http://127.0.0.1:8080/A.jsp”。

2. 已有的 Web 服务目录

Tomcat 服务器安装目录 webapps 下的任何一个子目录都可以作为一个 Web 服务目录。

安装 Tomcat 服务器后, webapps 目录下有如下子目录: jsp-examples、balancer、servlets-examples、servlets-examples 和 tomcat-docs, 也可以在目录 webapps 下再新建子目录, 如子目录 Dalian。

如果将 JSP 页面文件 A.jsp 保存到目录 webapps 下的服务目录中, 那么应当在浏览器的地址栏中输入 Tomcat 服务器的 IP 地址 (或域名)、端口号、Web 服务目录和 JSP 页面的名字。如 A.jsp 保存到 jsp-examples 中, 则输入的内容如下“http://127.0.0.1:8080/jsp-examples/A.jsp”。

3. 建立新的 Web 服务目录

可以将 Tomcat 服务器所在计算机的某个目录设置成一个 Web 服务目录, 并为该 Web 服务目录指定虚拟目录, 即隐藏 Web 服务目录的实际位置, 用户只能通过虚拟目录访问 Web 服务目录中的 JSP 页面。

修改 Tomcat 服务器安装目录下 conf 目录中的 server.xml 文件, 来设置新的 Web 服务目录。假设要将 D:\MyJsp\star 和 C:\sun 作为 Web 服务目录, 并让用户分别使用 hello 和 moon 虚拟目录访问 Web 服务目录 D:\MyJsp\star 和 C:\sun 下的 JSP 页面, 则应先用记事本打开文件夹 Tomcat6.0\conf 中的主配置文件 server.xml, 找到出现 “</HOST>” 的部分 (server.xml 文件尾部), 然后在 “</HOST>” 的前面加入如下内容:

```
<Context path="/hello" docBase="D:/MyJsp/star" debug="0" reloadable="true" />
<Context path="/moon" docBase="C:/sun" debug="0" reloadable="true" />
```

注意: XML 文件是区分大小写的, 不可以将 <Context> 写成 <context>。

主配置文件 server.xml 修改后, 必须重新启动 Tomcat 服务器。这样, 用户就可以将 JSP 页面存放到目录 D:\MyJsp\star 或 C:\sun 中, 可以通过虚拟目录 hello 或 moon 访问 JSP 页面, 如将 A.jsp 保存到目录 D:\MyJsp\star 或 C:\sun 中, 应在浏览器地址栏中输入 “http://127.0.0.1:8080/hello/A.jsp” 或 “http://127.0.0.1:8080/moon/A.jsp”。

注意: 在学习或使用 JSP 时, 不提倡将所有的 JSP 页面都存放在 Tomcat 服务器的根目录中, 应当善于建立新的 Web 应用目录, 以便有效地管理 JSP 页面。

4. 相对目录

Web 服务目录下的目录称为该 Web 服务目录下的相对服务目录。例如, 可以在 Web 服务目录 C:\sun 下再建立子目录 image, 将文件 B.jsp 保存到 image 目录中, 则可以在浏览器的地址栏中输入 “http://127.0.0.1:8080/moon/image/B.jsp” 来访问 B.jsp。Web 服务目录下的 JSP 页面可以通过相对路径来访问子目录中的 JSP 页面, 如 Web 服务目录 sun 通过相对路径访问子目录中的 JSP 页面 image/B.jsp (不可以写成 “/image/B.jsp”, “/” 代表根目录, “/image/B.jsp” 中的 image 将代表一个 Web 服务目录, 而不再是 Web 服务目录 sun 的一个子目录)。

【例 1-2】 将 A.jsp 保存到 Web 服务目录 C:\sun 中, 将 B.jsp 保存到 sun 的相对目录 image 中。A.jsp 中通过超链接访问 B.jsp, 如下所示:

A.jsp

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

```

<A href=image/B.jsp > 链接到 B.jsp</A>
</BODY></HTML>
<%@ page contentType="text/html;charset=GB2312" %>

```

B.jsp

```

<HTML><BODY bgcolor=cyan>
<FONT size=8>
<P> 1 到 100 的连续和
    <% int sum=0;
        for(int i=1;i<=100;i++){
            sum=sum+i;
        }
        out.println("sum="+sum);
    %>
</FONT>
</BODY></HTML>

```

1.5 设置端口号

8080 是 Tomcat 服务器的默认端口号。可以通过修改 Tomcat 服务器的 conf 目录下的主配置文件 server.xml 来更改端口号。用记事本打开 server.xml 文件，找到如下部分：

```

<Connector port="8080" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true"
/>

```

将其中的 port="8080"更改为新的端口号即可，如将“8080”更改为“9080”等。

如果 Tomcat 服务器所在的计算机没有启动其他占用端口号 80 的网络程序，也可以将 Tomcat 服务器的端口号设置为 80，这样用户在访问 JSP 页面时可以省略端口号，如“http://127.0.0.1/first.jsp”。

习 题 1

1. 安装 JDK 后应当进行哪些设置？
2. 运行 startup.bat 文件启动 Tomcat 服务器的好处是什么？
3. Hello.jsp 和 hello.jsp 是否是相同的 JSP 文件名字？
4. 请在 C:\下建立一个名字为 Game 的目录，并将该目录设置成一个 Web 服务目录，然后编写一个简单的 JSP 页面，保存到该目录中，让用户使用虚拟目录 moon 访问该 JSP 页面。
5. 怎样访问 Web 服务目录子目录中的 JSP 页面？
6. 如果想修改 Tomcat 服务器的端口号，应当修改哪个文件？能否将端口号修改为 80？

第 2 章

JSP 页面

本章导读

- ☆ 知识点：掌握 JSP 页面的基本结构以及运行原理。
- ☆ 重点：学习怎样声明 JSP 页面的成员变量、方法；掌握怎样使用 Java 程序片和 Java 表达式。
- ☆ 难点：掌握 Java 程序片的执行原理。
- ☆ 关键实践：上机编写两个 JSP 页面。一个 JSP 页面重点测试 JSP 页面的成员变量和局部变量的不同，另一个 JSP 页面着重尝试怎样分解一个 Java 程序片。

2.1 JSP 页面的基本结构

在传统的 HTML 页面文件中加入 Java 程序片和 JSP 标签，就构成了一个 JSP 页面。一个 JSP 页面可由 5 种元素组合而成：① 普通的 HTML 标记符；② JSP 标记，如指令标记、动作标记；③ 成员变量和方法；④ Java 程序片；⑤ Java 表达式。

Tomcat 服务器的安装目录下 webapps 的子目录都可以作为一个 Web 服务目录，在 webapps 目录下新建一个目录 chapter2。除非特别约定，本章例子中的 JSP 页面均保存在目录 chapter2 中。

【例 2-1】（效果如图 2.1 所示）

```
程序片创建Date对象:  
Thu Feb 15 08:57:00 CST 2007  
在下一行输出和:  
146
```

图 2.1 简单的 JSP 页面

example2_1.jsp


```

<%@ page contentType="text/html;charset=GB2312" %> <!-- JSP 指令标记 -->
<%@ page import="java.util.Date" %> <!-- JSP 指令标记 -->
<%! Date date;                                //数据声明
    int sum;
    public int add(int m,int n) {                //方法声明
        return m+n;
    }
%>
<HTML><BODY bgcolor=cyan> <!--HTML 标记 -->
<FONT size=4><P>程序片创建 Date 对象:
    <% date=new Date();                          //Java 程序片
        out.println("<BR>" +date);
        sum=add(12,34);
    %>
<BR>在下一行输出和: <BR>
    <%= sum+100 %> <!-- Java 表达式 -->
</FONT></BODY></HTML>

```

2.2 JSP 的运行原理

当 Tomcat 服务器上的一个 JSP 页面第一次被请求执行时, Tomcat 服务器将启动一个线程。该线程的任务是, 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, 并将该字节码文件加载到内存中, 然后执行字节码文件响应客户的请求, 该线程完成任务后, 线程进入死亡状态。这个字节码的功能如下:

- ⊙ 把 JSP 页面中普通的 HTML 标记符号, 交给客户的浏览器执行显示。
- ⊙ JSP 标记、数据和方法声明、Java 程序片由 Tomcat 服务器负责执行, 将需要显示的结果发送给客户的浏览器。
- ⊙ Java 表达式由 Tomcat 服务器负责计算, 将结果转化为字符串, 交给客户的浏览器负责显示。

被加载到内存中的字节码将常驻内存, 当这个 JSP 页面再次被请求执行时, Tomcat 服务器将再启动一个线程, 直接执行常驻内存的字节码文件来响应客户。这也是 JSP 比 ASP 速度快的一个原因。而 JSP 页面的首次执行往往由服务器管理者来执行。

当多个客户请求一个 JSP 页面时, Tomcat 服务器为每个客户启动一个线程, 该线程负责执行常驻内存的字节码文件来响应相应客户的请求。这些线程由 Tomcat 服务器来管理, 将 CPU 的使用权在各线程之间快速切换, 以保证每个线程都有机会执行字节码文件 (如图 2.2 所示), 这与传统的 CGI 为每个客户启动一个进程相比较, 效率要高得多。

注: 如果对 JSP 页面进行了修改、保存, 那么 Tomcat 服务器会生成新的字节码。

下面是 Tomcat 服务器生成的 example2_1.jsp 的 Java 文件, 我们把 Tomcat 服务器交给客户端浏览器负责显示的内容做了注释 (§)。如果 JSP 页面保存在目录 Root 中, 则在 Tomcat 服务器下的目录 work\Catalina\localhost_org\apache\jsp 中, 可以找到 Tomcat 服务器生成的 JSP 页面对应的 Java 文件和编译 Java 文件得到的字节码文件。