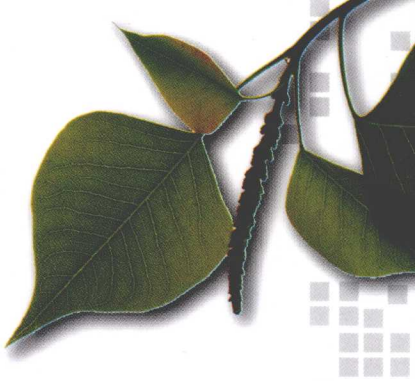
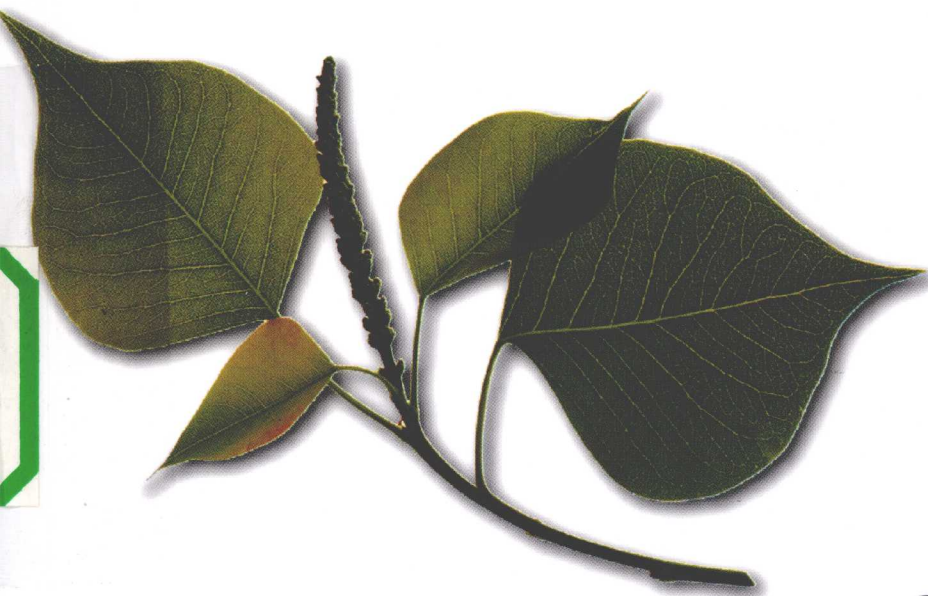




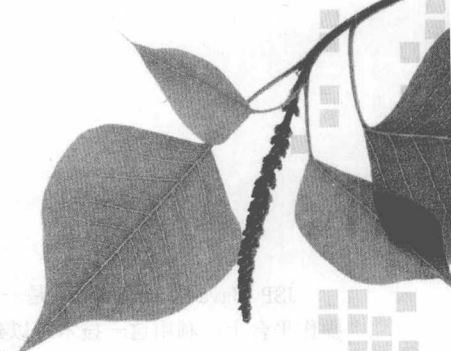
JSP

基础教程 (第2版)

耿祥义 张跃平 编著



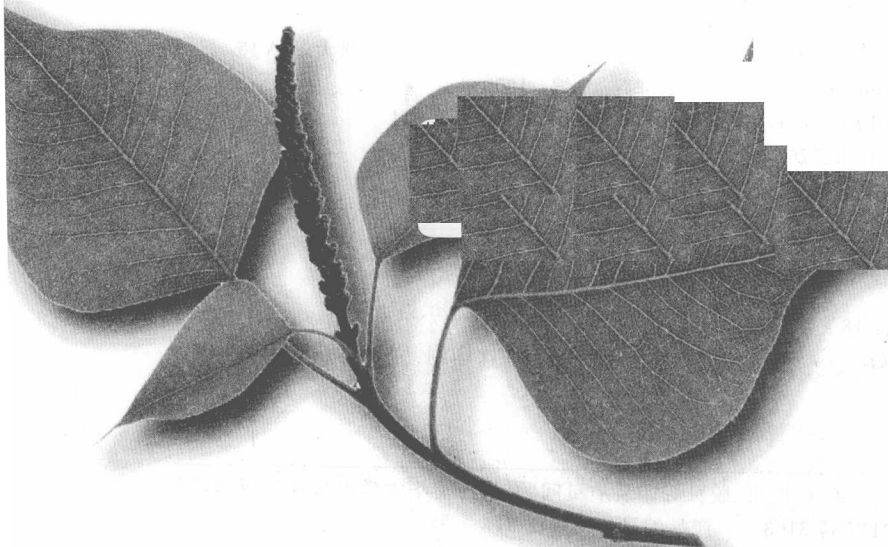
清华大学出版社



JSP

基础教程

耿祥义 张跃平 编著



清华大学出版社
北京

内 容 简 介

JSP (Java Server Pages) 是一种动态网页技术标准, 它可以无缝地运行在 UNIX、Linux 和 Windows 操作平台上。利用这一技术可以建立安全、跨平台的先进动态网站。

本书是《JSP 基础教程》的第 2 版, 增加了一些新的内容, 详细讲解了 JSP 语法和基本的程序设计方法。全书共分 9 章, 包括 JSP 概述、JSP 页面与 JSP 标记、Tag 文件与 Tag 标记、JSP 内置对象、JSP 文件操作、JSP 中使用数据库、JSP 与 JavaBean、Java Servlet 基础, 以及 MVC 模式等重要内容。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及在开发商务网站方面的应用, 使读者快速掌握和运用 JSP 的编程技巧。

本书不仅可以作为各大中专院校的 JSP 培训教材, 也适合自学者及网站开发人员参考使用。

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 基础教程 (第 2 版) / 耿祥义, 张跃平编著. —北京: 清华大学出版社, 2009.10
ISBN 978-7-302-17593-3

I. J… II. ①耿…②张… III. JAVA 语言—主页制作—程序设计—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2008) 第 066611 号

责任编辑: 田在儒

责任校对: 徐俊伟

责任印制: 孟凡玉

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

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 刷 者: 北京市昌平环球印刷厂

装 订 者: 北京国马印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 18 字 数: 447 千字

版 次: 2009 年 10 月第 2 版 印 次: 2009 年 10 月第 1 次印刷

印 数: 1~5000

定 价: 28.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 023242-01

导 读

感谢您选择本书，为了帮助您更好地学习本书的知识，请仔细阅读下面的内容。

本书特色

本书主要针对 JSP 的初学者，重点放在 JSP 中一些重要概念和技术上，特别是 Tag 文件、JavaBean 和 MVC 模式技术。所选例题具有针对性，便于理解 JSP 中的概念和技术，使读者在较短的时间内打下一个好的基础。

读者对象

本书适合具有一定 Java 基础的初学者阅读，适合作为计算机专业的选修课教材和社会培训机构相关专业的培训教材。

写作环境

本书用到的 Tomcat 6.0 和 JDK 1.5 可分别登录 <http://jakarta.apache.org/tomcat> 和 <http://www.java.sun.com> 下载。

软件背景

JSP 是由 Sun 公司倡导、许多公司参与，于 1999 年推出的一种动态网页技术标准。JSP 是基于 Java Servlet 及整个 Java 体系的 Web 开发技术，利用这一技术可以建立安全、跨平台的先进动态网站，这项技术还在不断地更新和优化中。JSP 以 Java 技术为基础，又在许多方面做了改进，具有动态页面与静态页面分离，能够脱离硬件平台的束缚，以及编译后运行等优点，JSP 已经成为 Internet 上的主流开发工具。

学习提示

本书分为 9 章。第 1 章主要介绍 Tomcat 6.0 的安装与配置，通过一个简单的 JSP 页面初识 JSP 概貌。第 2 章详细讲解 JSP 的基本语法，包括程序片、页面指令等重要内容。第 3 章主要讲解 Tag 文件与标记，重点强调了怎样使用 Tag 文件实现代码复用。第 4 章主要讲解 JSP 的内置对象，重点讲解了 Session 会话对象。第 5 章讲解输入输出流技术，重点介绍了对象流，以及文件的上传与下载。第 6 章涉及的内容是数据库，也是 Web 应用开发中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 7 章讲解 JavaBean 的使用，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。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，并按照 MVC 模式给出了易于理解 MVC 设计模式的例子。

本书作者

耿祥义，1995 年获中国科学技术大学理学博士学位，1997 年中山大学博士后出站，现任大连交通大学软件学院教授。张跃平，大连交通软件学院讲师。

技术支持

如果您在阅读本书的过程中有什么困难，可以登录到本书服务网站，网址是 <http://www.heimofang.com>。这是大家共同交流的平台，在那里会有很多的作者、老师、读者和编辑一起交流，在相关的栏目中发求助帖子，您的问题会很快得到解答。除上述方法外，也可以使用下面的方式寻求技术支持。

✉ 发电子邮件到 tianzr@tup.tsinghua.edu.cn

☎ 打电话 010-62773950

目 录

第 1 章 JSP 概述	1
1.1 什么是 JSP	1
1.2 安装配置 JSP 运行环境	1
1.3 JSP 页面与服务目录	4
1.3.1 JSP 页面	4
1.3.2 Web 服务目录	6
1.4 JSP 运行原理	7
习题一	10
第 2 章 JSP 页面与 JSP 标记	11
2.1 JSP 页面的基本结构	11
2.2 变量和方法的声明	12
2.2.1 声明变量	12
2.2.2 声明方法	14
2.3 Java 程序片	15
2.4 表达式	17
2.5 JSP 中的注释	18
2.6 JSP 指令标记	19
2.6.1 page 指令	19
2.6.2 include 指令标记	23
2.7 JSP 动作标记	26
2.7.1 Include 动作标记	26
2.7.2 param 动作标记	28
2.7.3 forward 动作标记	29
2.7.4 plugin 动作标记	31
2.7.5 useBean 动作标记	32
习题二	32
第 3 章 Tag 文件与 Tag 标记	34
3.1 Tag 文件的结构	34
3.2 Tag 文件的保存	35

3.3	Tag 标记与 Tag 文件的使用	35
3.4	Tag 文件中的常用指令	39
3.5	Tag 标记的嵌套	51
	习题三	52
第 4 章	JSP 内置对象	53
4.1	request 对象	54
4.1.1	获取用户提交的信息	55
4.1.2	处理汉字信息	57
4.1.3	常用方法举例	58
4.1.4	使用 Tag 文件处理有关数据	61
4.1.5	处理 HTML 标记	63
4.2	response 对象	72
4.2.1	动态响应 contentType 属性	73
4.2.2	response 的 HTTP 文件头	75
4.2.3	response 重定向	76
4.2.4	response 的状态行	77
4.3	session 对象	80
4.3.1	session 对象的 Id	80
4.3.2	session 对象与 URL 重写	82
4.3.3	session 对象存储数据	83
4.3.4	在 Tag 文件中使用 session 对象	86
4.3.5	session 对象的生存期限	88
4.3.6	计数器	90
4.4	out 对象	92
4.5	application 对象	93
4.5.1	application 对象的常用方法	94
4.5.2	用 application 制作留言板	94
	习题四	97
第 5 章	JSP 中的文件操作	99
5.1	File 类	99
5.1.1	获取文件的属性	100
5.1.2	创建目录	100
5.1.3	删除文件和目录	103
5.2	使用字节流读写文件	103
5.2.1	FileInputStream 和 FileOutputStream 类	104
5.2.2	BufferedInputStream 和 BufferedOutputStream 类	105

5.3	使用字符流读写文件	107
5.3.1	FileReader 和 FileWriter 类	108
5.3.2	BufferedReader 和 BufferedWriter 类	108
5.4	数据流	110
5.5	对象流	113
5.6	RandomAccessFile 类	120
5.7	文件上传	124
5.8	文件下载	129
	习题五	130
第 6 章	JSP 中使用数据库	132
6.1	SQL Server 2000 数据库管理系统	132
6.2	JDBC	134
6.3	连接数据库的常用方式	134
6.3.1	JDBC-ODBC 桥接器	134
6.3.2	使用纯 Java 数据库驱动程序	140
6.4	查询记录	143
6.4.1	顺序查询	144
6.4.2	随机查询	147
6.4.3	条件查询	151
6.4.4	排序查询	155
6.4.5	模糊查询	157
6.5	更新记录	159
6.6	添加记录	161
6.7	删除记录	164
6.8	常见数据库连接	166
6.8.1	连接 Oracle 数据库	166
6.8.2	连接 Access 数据库	167
6.9	查询 Excel 电子表格	169
6.10	网上投票	170
	习题六	174
第 7 章	JSP 与 JavaBean	175
7.1	编写 JavaBean 和使用 JavaBean	175
7.1.1	bean 的编写与保存	175
7.1.2	使用 bean	177
7.2	获取和修改 bean 的属性	181
7.2.1	getProperty 动作标记	181

7.2.2	setProperty 动作标记	182
7.3	bean 的辅助类	186
7.4	使用 bean 的简单例子	187
7.4.1	三角形 bean	187
7.4.2	猜数字 bean	189
7.4.3	日历 bean	191
7.4.4	四则运算 bean	194
7.4.5	浏览图片 bean	196
7.5	JavaBean 与文件操作	198
7.5.1	读文件 bean	198
7.5.2	写文件 bean	201
7.5.3	上传文件 bean	203
7.6	JavaBean 与数据库操作	206
7.6.1	查询记录 bean	206
7.6.2	分页显示记录 bean	209
7.7	标准化考试 bean	214
	习题七	217
第 8 章	Java Servlet 基础	219
8.1	Servlet 类与 servlet 对象	219
	Servlet 类及字节码的保存	219
8.2	编写 web.xml	221
8.3	servlet 对象的创建与运行	223
8.4	servlet 对象的工作原理	223
8.4.1	servlet 对象的生命周期	223
8.4.2	init 方法	224
8.4.3	service 方法	224
8.4.4	destroy 方法	225
8.5	通过 JSP 页面调用 servlet	225
8.5.1	通过表单向 servlet 提交数据	225
8.5.2	通过超链接访问 servlet	227
8.6	共享变量	228
8.7	doGet 和 doPost 方法	230
8.8	重定向与转发	232
8.8.1	sendRedirect 方法	232
8.8.2	RequestDispatcher 对象	232
8.9	使用 session	235
	获取用户的会话	236

习题八	238
第 9 章 MVC 模式	240
9.1 MVC 模式介绍	240
9.2 JSP 中的 MVC 模式	241
9.3 模型的生命周期与视图更新	242
9.3.1 request 周期的 JavaBean	242
9.3.2 session 周期的 JavaBean	243
9.3.3 application 周期的 JavaBean	244
9.4 MVC 模式的简单实例	245
9.4.1 计算三角形和梯形的面积	245
9.4.2 简单的计算器	250
9.5 MVC 模式与文件操作	255
9.6 MVC 模式与数据库操作	259
9.7 MVC 模式与注册登录	266
9.7.1 注册	267
9.7.2 登录	272
9.7.3 验证	277
习题九	278

第 1 章 JSP 概述

1.1 什么是 JSP

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 应用程序，这样的应用程序负责处理用户的请求，满足用户对信息的请求。

JSP 是 Web 服务器端开发技术，利用这一技术可以建立安全、跨平台的 Web 应用程序。JSP 的安全性和跨平台得益于 Java 语言，这是因为 Java 语言具有不依赖于平台、面向对象和安全等优良特性，已经成为网络程序设计的佼佼者。许多和 Java 有关的技术得到了广泛的应用和认可，JSP 技术就是其中之一。读者可能对 Microsoft 的 ASP (Active Server Pages) 比较熟悉，ASP 也是一项 Web 服务器端的开发技术，可以开发出动态的、高性能的 Web 服务应用程序。JSP 和 ASP 技术非常相似。ASP 使用的是 VBScript 脚本语言，而 JSP 使用的是 Java 编程语言。与 ASP 相比，JSP 以 Java 技术为基础，又在许多方面做了改进，具有动态页面与静态页面分离、能够脱离硬件平台的束缚及编译后运行等优点，完全克服了 ASP 的脚本级执行的缺点。JSP 已经成为开发动态网站的主流技术。

需要强调的一点是：要想真正地掌握 JSP 技术，必须有较好的 Java 语言基础，以及 HTML 语言方面的知识。

1.2 安装配置 JSP 运行环境

JSP 的核心内容就是编写 JSP 页面（有关 JSP 页面的内容见本章 1.3 节和第 2 章的 2.1 节）。对于 JSP 页面，需要一个应用程序来运行它，我们将这样的应用程序称作 JSP 引擎或 JSP 容器。将安装 JSP 引擎的计算机称作一个支持 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 1.4 节讲解。

自从 JSP 发布以后，出现了各式各样的 JSP 引擎。1999 年 10 月 Sun 公司将 Java Server Page 1.1 代码交给 Apache 组织，Apache 组织对 JSP 进行了实用研究，并将这个服务器项目称为 Tomcat。从此，著名的 Web 服务器 Apache 开始支持 JSP。目前，Tomcat 能和

大部分主流服务器一起高效率地工作。可以登录 <http://jakarta.apache.org/tomcat> 免费下载 Tomcat。登录之后，首先在 Download 中选择 Tomcat 6.x，然后在 Binary Distributions 的 Core 中选择 Zip 或 Windows Service Installer 即可。如果选择 Zip 将下载 apache-tomcat-6.0.13.zip；如果选择 Windows Service Installer 将下载 apache-tomcat-6.0.13.exe。

本章重点讲述在 Windows XP/Windows 2000 操作系统下 Tomcat 的安装与配置，将安装了 Tomcat 的计算机称做一个 Tomcat 服务器。

1. 安装 JDK

安装 Tomcat 之前，首先安装 JDK，这里安装 Sun 公司的 JDK 1.5。假设 JDK 的安装目录是：

D:\Jdk1.5。

安装 JDK 之后需要进行几个环境变量的设置。对于 Windows XP/2000，用鼠标右键单击“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。选择该对话框中的“高级”选项卡，然后单击“环境变量”按钮，分别添加如下的系统环境变量。

变量名：JAVA_HOME，变量值：D:\jdk1.5。

变量名：classpath，变量值：D:\jdk1.5\jre\lib\rt.jar;.;。

变量名：Path，变量值：D:\jdk1.5\bin。

如果曾经设置过环境变量 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。

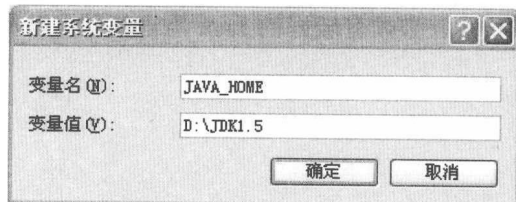


图 1-1 设置 JAVA_HOME

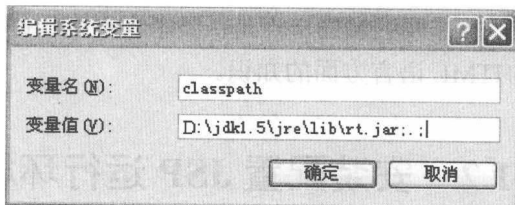


图 1-2 编辑 classpath

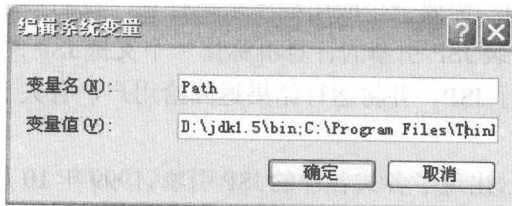


图 1-3 编辑 Path

2. 安装与启动 Tomcat 服务器

1/ apache-tomcat-6.0.13.zip 的安装

操作步骤:

① 将下载的 apache-tomcat-6.0.13.zip 解压到磁盘某个分区, 比如解压到 D:\, 解压后将出现如图 1-4 所示的目录结构。



图 1-4 Tomcat 服务器目录结构

② 执行 Tomcat 安装根目录 bin 文件夹中的 startup.bat 或 tomcat6.exe 来启动 Tomcat 服务器。执行 startup.bat 启动 Tomcat 服务器会占用一个 MS-DOS 窗口 (如图 1-5 所示的界面), 如果关闭当前 MS-DOS 窗口, 则将关闭 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。

```
2007-8-6 9:14:48 org.apache.catalina.core.StandardService start
信息: Starting service Catalina
2007-8-6 9:14:48 org.apache.catalina.core.StandardEngine start
信息: Starting Servlet Engine: Apache Tomcat/6.0.13
2007-8-6 9:14:50 org.apache.coyote.http11.Http11Protocol start
信息: Starting Coyote HTTP/1.1 on http-8080
2007-8-6 9:14:50 org.apache.jk.common.ChannelSocket init
信息: JK: ajp13 listening on /0.0.0.0:8009
2007-8-6 9:14:50 org.apache.jk.server.JkMain start
信息: Jk running ID=0 time=0/63 config=null
2007-8-6 9:14:51 org.apache.catalina.startup.Catalina start
信息: Server startup in 2176 ms
```

图 1-5 启动 Tomcat 服务器

2/ jakarta-tomcat-6.0.13.exe 的安装

apache-tomcat-6.0.13.exe 文件是针对 MS-Windows 的 Tomcat 服务器, 安装后形成的 Tomcat 服务器的目录结构和 apache-tomcat-6.0.13.zip 安装的基本相同。

单击下载的 apache-tomcat-6.0.13.exe, 将出现“安装向导”界面, 单击其中的 Next 按钮, 接受授权协议后, 将出现选择“安装方式”的界面。在“安装方式”界面中选择 Normal、Minimum、Custom 或 Full 之一, 然后按照安装向导的提示进行安装即可, 不再赘述。

3/ 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。

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

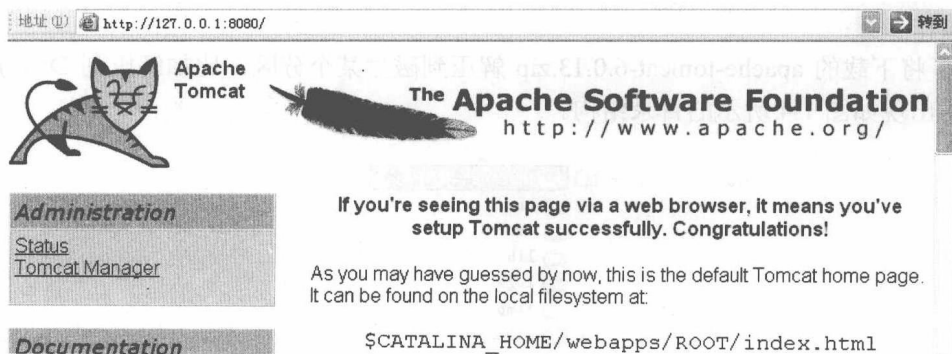


图 1-6 测试 Tomcat 服务器

4/ 配置端口

8080 是 Tomcat 服务器默认占用的端口。可以通过修改 Tomcat 服务器安装目录中 conf 文件下的主配置文件 server.xml 来更改端口号。用记事本打开 server.xml 文件，找到出现下列代码的部分：

```
<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，并重新启动 Tomcat 服务器即可，比如将 8080 更改为 9090 等。

如果 Tomcat 服务器所在的计算机没有启动其他的占有 80 号端口的网络程序，也可以将 Tomcat 服务器的端口号设置为 80，这样用户在访问 Tomcat 服务器时可以省略端口号，例如 http://127.0.0.1/。

1.3 JSP 页面与服务目录

1.3.1 JSP 页面

我们将在后面的章节里详细地学习编写 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，还使用标记符号“<%”，“%>”加入 Java 程序片。一个 JSP 页面按文本文件保存，扩展名是.jsp。例如，使用文本编辑器“记事本”编辑 JSP 页

面, 在保存 JSP 页面时, 必须将“保存类型”选择为“所有文件”, 将“编码”选择为 ANSI。如果保存 JSP 页面时, 你的计算机系统总是给文件名尾加上.txt, 那么你在保存 JSP 页面时可以将文件名用双引号括起, 如图 1-7 所示。

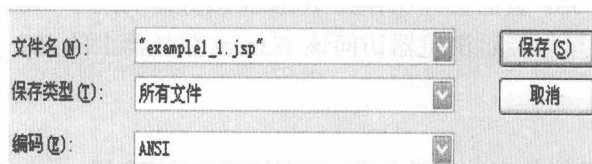


图 1-7 JSP 页面文件名

在保存 JSP 页面时, 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。需要注意的是, JSP 技术基于 Java 语言, 名字区分大小写。

为了明显地区分普通的 HTML 标记和 Java 程序片段及 JSP 标签, 我们用大写字母书写普通的 HTML 标记符号 (这不是必须的, HTML 标记不区分大小写)。可以用记事本或更好的文本编辑器来编辑 JSP 页面的源文件。

下面的例子 1 是一个简单的 JSP 页面。

例子 1 (效果如图 1-8 所示)

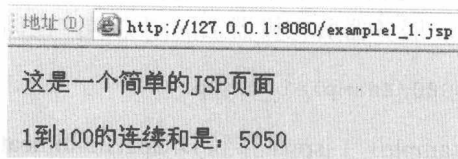


图 1-8 简单的 JSP 页面

Example1_1.jsp

```
<%@ page contentType="text/html;charset=GB2312"%>
<HTML>
<BODY BGCOLOR=yellow>
<FONT Size=3>
<P>这是一个简单的 JSP 页面
    <%
        int i, sum=0;
        for (i=1;i<=100;i++)
        {
            sum=sum+i;
        }
    %>
<P> 1 到 100 的连续和是: <%=sum %>
</FONT>
</BODY>
</HTML>
```

1.3.2 Web 服务目录

必须将编写好的 JSP 页面文件保存到 Tomcat 服务器的某个 Web 服务目录中，只有这样，远程的用户才可以通过浏览器访问该 Tomcat 服务器上的 JSP 页面。

1. 根目录

如果 Tomcat 服务器的安装目录是 D:\apache-tomcat-6.0.13, 那么 Tomcat 的 Web 服务目录的根目录是:

```
D:\apache-tomcat-6.0.13\webapps\Root.
```

用户如果准备访问根目录中的 JSP 页面，可以在浏览器中输入 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），例如，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 Example1_1.jsp，那么用户在浏览器中输入的内容是：

```
http://192.168.1.200:8080/Example1_1.jsp.
```

也许你没有为 Tomcat 服务器所在的机器设置过一个有效的 IP 地址，那么为了调试 JSP 页面，可以打开 Tomcat 服务器所在机器上的浏览器，在浏览器的地址栏中输入：

```
http://127.0.0.1:8080/Example1_1.jsp.
```

我们将例子 1 中的 Example1_1.jsp 保存到 D:\apache-tomcat-6.0.13\webapps\Root 中，用浏览器访问 Example1_1.jsp 的效果如图 1-8 所示。

2. 已有的 Web 服务目录

Tomcat 服务器安装目录的 webapps 目录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 目录下已有 jsp-Examples、balancer、servlets-Examples、servlets-Examples 和 tomcat-docs 子目录。可以在 webapps 目录下新建子目录，如 MyBook 子目录，那么 MyBook 就成为一个 Web 服务目录。如果将 JSP 页面文件 Example1_1.jsp 保存到 webapps 目录下的 Web 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，比如，Example1_1.jsp 保存到 jsp-Examples 中，输入的内容为：

```
http://127.0.0.1:8080/jsp-Examples/Example1_1.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。

通过修改 Tomcat 服务器安装目录下 conf 文件夹中的 server.xml 文件来设置新的 Web 服务目录。假设要将 D:\MyJsp\star 及 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。首先用记事本打开 conf 文件夹中的主配置文件 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:

```
<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,就可以将 JSP 页面存放到 D:\MyJsp\star 或 C:\sun 中,用户可以通过虚拟目录 hello 或 moon 访问 JSP 页面,例如,Example1_1.jsp 保存到 D:\MyJsp\star 或 C:\sun 中,在浏览器地址栏中输入:

```
http://127.0.0.1:8080/hello/Example1_1.jsp,
```

或

```
http://127.0.0.1:8080/moon/Example1_1.jsp。
```

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 Web 服务目录。例如,可以在 Web 服务目录 D:\MyJsp\star 下再建立一个子目录 image,将 Example1_1.jsp 文件保存到 image 中,那么可以在浏览器的地址栏中输入:

```
http://127.0.0.1:8080/hello/image/Example1_1.jsp
```

来访问 Example1_1.jsp。

1.4 JSP 运行原理

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 JSP 引擎首先将 JSP 页面文件转译成一个 Java 文件,并编译这个 Java 文件生成字节码文件,然后执行字节码文件响应用户的请求。而当这个 JSP 页面再次被请求执行时,JSP 引擎将直接执行字节码文件来响应用户,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。字节码文件的主要工作有以下几方面。

- 把 JSP 页面中的 HTML 标记符号(页面的静态部分)交给用户的浏览器负责显示。
- 负责处理 JSP 标记,并将有关的处理结果发送到用户的浏览器。
- 执行“<%”和“%>”之间的 Java 程序片(JSP 页面中的动态部分),并把执行