普通高等院校计算机专业（本科）实用教程系列

JSP实用教程

耿祥义 张跃平 编著

清华大学出版社
http://www.tup.tsinghua.edu.cn

普通高等院校计算机专业（本科）实用教程系列

JSP 实用教程

耿祥义　张跃平　编著

清华大学出版社
北　京

内 容 简 介

JSP 是 Java Server Pages 的缩写，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。利用这一技术可以建立安全、跨平台的先进动态网站。

本书是一本实用教程，配备了大量的例子，叙述详细，通俗易懂，便于自学。针对较难理解的问题，例子都是从简单到复杂，逐步深入，便于读者掌握 JSP 技术。全书分为 9 章，1 至 6 章分别详细地介绍了 JSP 运行环境的配置、JSP 语法、JSP 内置对象、JSP 与文件、JSP 与数据库、JSP 与 JavaBeans 等内容；第 7 章与第 8 章介绍了怎样使用 JSP 技术创建完整的网站。第 9 章讲述 Java Servlet，针对 JSP 与 servelt 结合开发网站做了详细的介绍。本书不仅可以作为 JSP 的培训教材，也适合自学者及网站开发人员参考使用。本书的源程序及免费的 JSP 服务器开发工具 Tomcat4.0 可以在http://www.tupwq.net下载。

图书在版编目(CIP)数据

JSP 实用教程/耿祥义，张跃平编著. —北京：清华大学出版社，2003.5
（普通高等院校计算机专业（本科）实用教程系列）
ISBN 7-302-06524-1

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

中国版本图书馆 CIP 数据核字（2003）第 025956 号

出 版 者：清华大学出版社（北京清华大学学研大厦，邮编：100084）
http://www.tup.tsinghua.edu.cn
http://www.tup.com.cn
责任编辑：魏江江
印 刷 者：北京密云胶印厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：23　**字数**：540 千字
版　　次：2003 年 5 月第 1 版　2003 年 5 月第 1 次印刷
书　　号：ISBN 7-302-06524-1/TP・4894
印　　数：0001~5000
定　　价：32.00 元

序　言

时光更迭、历史嬗递。中国经济带着她足以令世人惊叹的持续高速发展驶入了一个新的世纪，一个新的千年。世纪之初，以微电子、计算机、软件、通信技术为主导的信息技术革命给我们生存的社会所带来的变化令人目不暇接。软件是优化我国产业结构、加速传统产业改造和用信息化带动工业化的基础产业，是体现国家竞争力的战略性产业，是从事知识的提炼、总结、深化和应用的高智型产业；软件关系到国家的安全，是保证我国政治独立、文化不受侵蚀的重要因素；软件也是促进其他学科发展和提升的基础学科；软件作为 20 世纪人类文明进步的最伟大成果之一，代表了先进文化的前进方向。美国政府早在 1992 年“国家关键技术”一文中提出“美国在软件开发和应用上所处的传统领先地位是信息技术及其他重要领域竞争能力的一个关键因素”，“一个成熟的软件制造工业的发展是满足商业与国防对复杂程序日益增长的要求所必需的”，“在很多国家关键技术中，软件是关键的起推动作用（或阻碍作用）的因素”。在 1999 年 1 月美国总统信息技术顾问委员会的报告“21 世纪的信息技术”中指出“从台式计算机、电话系统到股市，我们的经济与社会越来越依赖于软件”，“软件研究为基础研究方面最优先发展的领域”。而软件人才的缺乏和激烈竞争是当前国际的共性问题。各国、各企业都对培养、引进软件人才采取了特殊政策与措施。

为了满足社会对软件人才的需要，为了让更多的人可以更快地学到实用的软件理论、技术与方法。我们编著了《普通高等院校计算机专业（本科）实用教程系列丛书》。本套丛书面向普通高等院校学生，以培养面向 21 世纪计算机专业应用人才（以软件工程师为主）为目标，以简明实用、便于自学、反映计算机技术最新发展和应用为特色，具体归纳为以下几点：

1. 讲透基本理论、基本原理、方法和技术，在写法上力求叙述详细，算法具体，通俗易懂，便于自学。

2. 理论结合实际。计算机是一门实践性很强的科学，丛书贯彻从实践中来到实践中去的原则，许多技术理论结合实例讲，以便于学习和理解。

3. 本丛书形成完整的体系，每本教材既有相对独立性，又有相互衔接和呼应，为总的培养目标服务。

4. 每本教材都配以习题和实验，在各教学阶段安排课程设计或大作业，培养学生的实战能力与创新精神。习题和实验可以制作成光盘。

新世纪曙光激人向上，催人奋进。江总书记在十五届五中全会上的讲话：“大力推进国民经济和社会信息化，是覆盖现代化建设全局的战略举措。以信息化带动工业化，发挥优势，实现社会生产力的跨越式发展。”指明了我国信息界前进的方向。21 世纪日趋开放的国策与更加迅速发展的科技会托起祖国更加辉煌灿烂的明天。

孙家广

2001 年 3 月

普通高等院校计算机专业（本科）实用教程系列

编　委　会

前　言

Java 不依赖平台的特点使得它受到了广泛的关注，许多和 Java 相关的优秀技术不断出现，JSP（Java Server Pages）就是其中之一。JSP 是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术，利用这一技术可以建立动态的、高性能的、安全的、跨平台的先进动态网站。1999 年，Sun 公司利用自己在 Java 语言上的特权和优势，正式公布 JSP 技术标准，标志着一种更加先进的动态网页技术从此诞生。

本书是一本实用教程，配备了大量的例子，叙述详细，通俗易懂，便于自学。

本书不仅详细地介绍了 JSP 的主要内容，而且对 JSP 中最重要的两个内容: Javabeans 以及 Java Servlet 分章给予了详细和重点的讲述，避免了重点讲 JSP 而忽略 Java Servlet 或只重点讲 Java Servlet 而忽略了 JSP。针对较难理解的问题，例子都是从简单到复杂，逐步深入，便于读者掌握 JSP 技术。全书分为 9 章，1 至 5 章分别详细地介绍了 JSP 运行环境的配置、JSP 语法、JSP 内置对象、JSP 与文件、JSP 与数据库等内容；第 6 章详细地介绍了怎样在 JSP 使用 JavaBeans，这部分内容是 JSP 最核心和精彩的部分。按照 Sun 公司的定义，JavaBeans 是一个可重复使用的软件组件，可以实现代码的重复利用。JavaBeans 易编写、易维护、易使用，可以在任何安装了 Java 运行环境的平台上的使用，而不需要重新编译。在 JSP 页面中调用 Javabeans，可有效的分离静态工作部分和动态工作部分。第 7 章、第 8 章讲述怎样使用 JSP+Javabeans 技术创建完整的网站，通过这两章的学习，可以举一反三建立一些中小型的网站。在第 9 章，详细地介绍了 servlet 的运行原理，以及 JSP 页面中怎样调用一个 servlet 完成动态数据的处理。SUN 公司以 Java Servlet 为基础，推出了 Java Server Page。JSP 提供了 Java Servlet 的几乎所有好处，当一个客户请求一个 JSP 页面时，JSP 引擎根据 JSP 页面生成一个 Java 文件，即一个 servlet。通过第 9 章的学习，不仅对于深刻理解 JSP 有一定的帮助，而且通过学习 servlet，还能选择使用 JSP+JavaBeans+servlet 的模式来开发 Web 应用程序。尽管用 JSP 支持 JavaBeans 这一特点，可以有效的管理页面的静态部分和页面的动态部分，但通过 servlet 技术，也可以在一个 JSP 页面中调用一个 servlet 完成动态数据的处理，而让 JSP 页面本身处理静态的信息。因此，开发一个 Web 应用有两种模式可以选择：JSP+JavaBeans 和 JSP+JavaBeans+servlet

本书面向的读者是具有一定 Java 语言基础和初步 HTML 语言基础的人。本书不仅适合高等院校电子类专业的选修课教材，也适合计算机培训学校选做教材以及网站开发人员参考使用。

目　录

第 1 章　JSP 简介

1.1　什么是 JSP

JSP 是 Java Server Pages 的缩写，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 以及整个 Java 体系的 Web 开发技术，利用这一技术可以建立安全的、跨平台的先进动态网站，这项技术还在不断地被更新和优化。用户可能对 Microsoft 的 ASP(Active Server Pages)比较熟悉，ASP 也是一个 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 的编程语言是 VBScript 和 JavaScript，JSP 使用的是 Java。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，以及编译后运行等优点，完全克服了 ASP 的脚本级执行的缺点，因而会逐渐成为 Internet 上的主流开发工具。

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　JSP 页面

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 页面文件,简单地说，一个 JSP 页面除了普通的 HTML 标记符外，再使用标记符号“<%”、“%>”加入 Java 程序片。一个 JSP 页面文件的扩展名是 jsp，文件的名字必须符合标识符规定，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 程序片段以及 JSP 标签，本书中用大写字母书写普通的 HTML 标记符号。

下面的例子 1 是一个简单的 JSP 页面。

例子 1 (效果如图 1.1 所示)

Example1_1.jsp

```
<%@ page contentType="text/html;charset=GB2312"%>
<HTML>
<BODY BGCOLOR=cyan>
<FONT Size=1>
<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：
<BR>
  <%=sum%>
</FONT>
</BODY>
<HTML>
```

用浏览器访问该 JSP 页面的效果如图 1.1 所示。

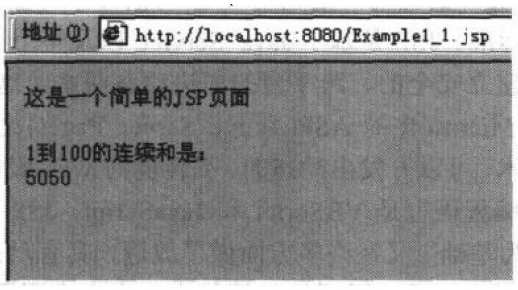

图 1.1　简单的 JSP 页面

1.3　JSP 的运行原理

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，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，JSP 引擎将直接执行这个字节码文件来响应客户，这也是 JSP 比 ASP 运行速度快的一个原因。而 JSP 页面的首次执行往往由服务器管理者来执行。这个字节码文件的主要工作是：

(1) 把 JSP 页面中普通的 HTML 标记符号(页面的静态部分)交给客户的浏览器负责显示。

(2) 执行“<%”和“%>”之间的 Java 程序片(JSP 页面中的动态部分)，并把执行结果交给客户的浏览器显示。

(3) 当多个客户请求一个 JSP 页面时，JSP 引擎为每个客户启动一个线程而不是启动一个进程，这些线程由 JSP 引擎服务器来管理，与传统的 CGI 为每个客户启动一个进程相比较，其效率要高得多。

下面是 JSP 引擎生成的 Example1_1.Jsp 的 Java 文件，其中把 JSP 引擎交给客户端负责显示的内容做了(***)注释。

```
package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;
public class first1$jsp extends HttpJspBase {
    static {
    }
    public first1$jsp( ) {
    }
    private static boolean _jspx_inited = false;
    public final void _jspx_init() throws org.apache.jasper.runtime.JspException {
    }
    public void _jspService(HttpServletRequest request,
   HttpServletResponse response)
        throws java.io.IOException, ServletException {
        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        String  _value = null;
        try {
           if (_jspx_inited == false) {
                synchronized (this) {
                   if (_jspx_inited == false) {
                       _jspx_init();
                       _jspx_inited = true;
                   }
               }
           }
      _jspxFactory = JspFactory.getDefaultFactory();
       response.setContentType("text/html;charset=GB2312");
       pageContext = _jspxFactory.getPageContext(this, request, response,"",
       true, 8192, true);
       application = pageContext.getServletContext();
       config = pageContext.getServletConfig();
       session = pageContext.getSession();
       out = pageContext.getOut();
  (***)out.write("\r\n<HTML>\r\n<BODY>\r\n<P>这是一个简单的 JSP 页面\r\n");
            int i, sum=0;
                for(i=1;i<=100;i++)
                   { sum=sum+i;
                   }
(***)       out.write("\r\n<P>  1 到 100 的连续和是: \r\n<BR>\r\n   ");
(***)       out.print(sum );
(***)       out.write("\r\n</BODY>\r\n<HTML>\r\n");
       } catch (Throwable t) {
           if (out != null && out.getBufferSize() != 0)
              out.clearBuffer();
```

```
            if (pageContext != null) pageContext.handlePageException(t);
        } finally {
            if (_jspxFactory != null) _jspxFactory.releasePageContext
            (pageContext);
        }
    }
}
```

下面是客户端浏览器查看到的 Example1_1.jsp 的源代码。

```
<HTML>
<BODY BGCOLOR=cyan>
<FONT Size=1>
<P>这是一个简单的 JSP 页面
<P>  1 到 100 的连续和是：
<BR>
  5050
</FONT>
</BODY>
<HTML>
```

1.4 安装配置 JSP 运行环境

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。这样，Jakarta—Tomcat 就诞生了(Jakarta 是 JSP 项目的最初名称)。目前，Tomcat 能和大部分主流服务器一起高效率地工作。

本节重点讲述 Window98/Window2000 操作系统下 Tomcat 服务器的安装配置。

安装 Tomcat 之前，必须首先安装 JDK，这里安装 Sun 公司的 JDK1.3。

假设 JDK 的安装目录是 C:\Jdk1.3。然后，解压缩 jakarta-tomcat-4.0.zip 文件，该文件可从 Sun 公司的网站 http://java.sun.com 或 http://jakarta.Apache.org 免费得到。假设解压缩文件到 D:\Tomcat，这时可以得到如图 1.2 所示的目录结构。

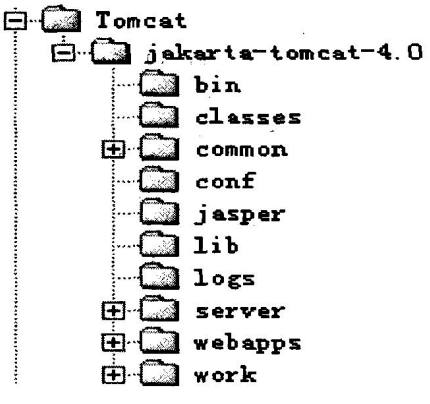

图 1.2 Tomcat 引擎目录结构

在启动 Tomcat 服务器之前，还需要进行几个环境变量的设置。

对于 window2000，用鼠标右键单击“我的电脑”，弹出菜单，然后选择属性，弹出“系统特性”对话框，再单击该对话框中的高级选项，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：C:\jdk1.3。
变量名：TOMCAT_HOME，变量值：D:\Tomcat\jakarta-tomcat-4.0。
变量名：CLASSPATH，变量值：C:\jdk1.3\jre\lib\rt.jar;.;。
变量名：PATH，变量值：C:\jdk1.3\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。

对于 Win9x，用记事本编辑 Autoexec.bat 文件，将如下的设置语句加入即可。

```
PATH  C:\jdk1.3.1_01\bin;
SET  CLASSPATH=.;C:\jdk1.3.1_01\jre\lib\rt.jar;
SET  TOMCAT_HOME=D:\Tomcat\jakarta-tomcat-4.0
SET  JAVA_HOME=C:\jdk1.3.1_01
```

现在，就可以启动 Tomcat 服务器了。执行 Tomcat\jakarta-tomcat-4.0\bin 下的 startup.bat，出现如图 1.3 所示的窗口，表明服务器已经启动。

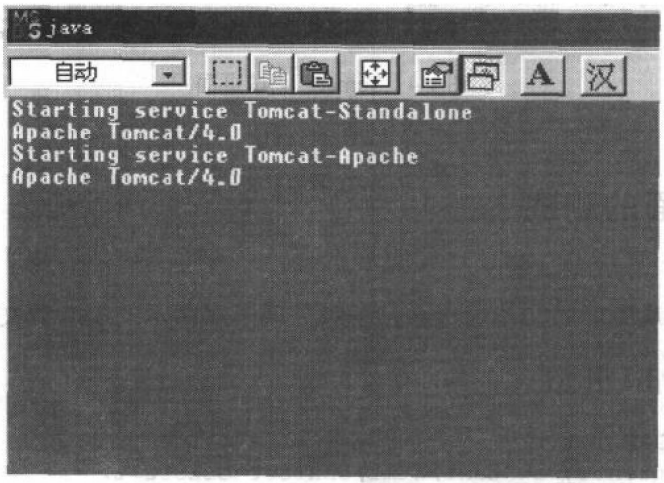

图 1.3　Tomcat 服务器启动后的窗口

在浏览器的地址栏中键入 http://localhost:8080，会出现如图 1.4 所示的 Tomcat 测试页面。

注：Tomcat 服务器内置 web 服务。

注：如果 Tomcat 不能启动，请首先检查环境变量的设置是否正确，对于 Win9x，如果出现“out of environment space”的错误提示，就需要修改 MS-DOS 属性，将属性中内存的初始环境更改为 4096。

注：8080 是 Tomcat 服务器的默认端口号。可以通过修改 Tomcat\jakarta-tomcat-4.0\conf 文件下的主配置文件 server.xml，更改端口号。用记事本打开 server.xml 文件，找到以下文字部分：

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
```

```
<Connector className="org.apache.catalina.connector.http.HttpConnector"
           port="8080" minProcessors="5" maxProcessors="75"
           enableLookups="true" redirectPort="8443"
           acceptCount="10" debug="0" connectionTimeout="60000"/>
```

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

注：可以通过执行 Tomcat\jakarta-tomcat-4.0\bin 下的 shutdown.bat 来关闭 Tomcat 服务器。

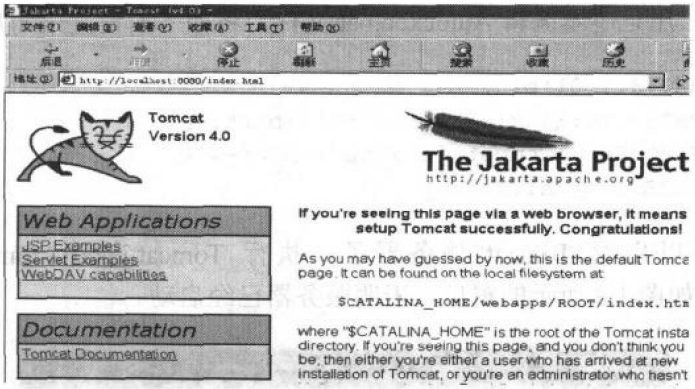

图 1.4 Tomcat 测试页面

1.5 JSP 页面的测试

(1) 用记事本或更好的文本编辑器，编辑如例子 2 所示的 JSP 源文件。

例子 2 (效果如图 1.5 所示)

Example1_2.jsp

```
<%@ page contentType="text/html;charset=GB2312"%>
<%@ page import="java.util.*"%>
<HTML>
<BODY>
<P> 现在的时间是:
    <%  Date date=new Date();
    %>
<BR>
  <%=date%>
</BODY>
<HTML>
```

(2) 将 JSP 文件命名为 Example1_2.jsp，保存到 Tomcat\Jakarta-tomcat-4.0\webapps\root 下。在浏览器的地址栏中输入 http://localhost:8080/ Example1_2.jsp，对 JSP 页面进行测试，出现如图 1.5 所示的效果。