21世纪高等学校规划教材

JSP CHENGXU SHEJI JIAOCHENG

JSP程序设计教程

龙海侠 主 编
莫壮坚 副主编

中国电力出版社
CHINA ELECTRIC POWER PRESS

21世纪高等学校规划教材

JSP CHENGXU SHEJI JIAOCHENG

JSP程序设计教程

主　编　龙海侠
副主编　莫壮坚
编　写　吴丽华
主　审　李宗民

中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书为21世纪高等学校规划教材。本书由浅入深、循序渐进、较为全面地介绍了JSP的运行系统、基本概念、语法规范及其相关内容，从基本的JSP概述、与其他技术的对比，到JSP的运行系统和模式，逐步深入地对JSP语法规范进行了详细的讲解，并结合应用实例加以巩固。本书分为三大模块。第1篇入门篇包括JSP概述、技术分析，JSP基础学习，JavaBean组件，JSP与Servlet；第2篇数据库操作包括全面解析JDBC、JDBC厂商选择性实现包、JSP数据库操作例程；第3篇为案例精讲。

本书可作为高等本科院校JSP程序设计课程的教材，也可作为高职高专相关课程的教材，还可供专业网页设计人员、网站维护人员及网页制作爱好者参考。

图书在版编目（CIP）数据

JSP 程序设计教程 / 龙海侠主编. —北京：中国电力出版社，2011.6

21世纪高等学校规划教材

ISBN 978-7-5123-1886-1

Ⅰ. ①J… Ⅱ. ①龙… Ⅲ. ①JAVA 语言－网页制作工具－高等学校－教材 Ⅳ. ①TP312 ②TP393.092

中国版本图书馆CIP数据核字（2011）第130555号

中国电力出版社出版、发行

（北京市东城区北京站西街19号 100005 http://www.cepp.sgcc.com.cn）

北京市同江印刷厂印刷

各地新华书店经售

*

2011年8月第一版 2011年8月北京第一次印刷

787毫米×1092毫米 16开本 14.75印张 357千字

定价 **25.00** 元

前　言

JSP（Java Server Page）是由Sun公司在Java语言基础上开发出来的一种动态网页制作技术，它是Java开发阵营中最具代表性的解决方案。JSP不仅具有与Java一样的面向对象性、便利性、跨平台性等优点，还拥有Java Servlet的稳定性，并且可以使用Servlet提供的API、Java Bean及Web开发框架技术，使页面代码与后台处理代码分离，提高工作效率。JSP技术依靠Java语言的稳定、安全、可移植性等优点，已经成为比较流行的Web动态网页开发技术和大中型网站开发的首选，并成为大中专院校的必修课程。

编者结合多年的JSP教学与实践经验，并考虑读者对Java Web应用开发知识和技能需要，本书将JSP基础知识和Java Web应用开发经验融为一体，本书有以下特点：

1. 内容全面而深入。在介绍JSP基本技术的基础上，强化JSP技术的核心和精髓，以及Java Web应用开发整体解决方案知识的讲解。

2. 理论与实践相结合。在介绍JSP技术基本知识的同时，注重实例分析与可操作性，实现了理论与实践的结合。

3. 连贯与系统性。本书强化知识点与案例的前后连贯，使读者能够实现从零基础到入门、精通JSP技术，并具备大中型网站的开发水平。

本书由3篇组成，第1篇入门篇，由第1章JSP概述，第2章JSP基础学习，第3章JavaBean组件和第4章JSP与Servlet等内容组成；第2篇数据库操作，由第5章全面解析JDBC，第6章JDBC厂商选择性实现包和第7章 JSP数据库操作例程等内容组成；第3篇案例精讲，通过学生课绩管理系统全面讲述JSP进行系统开发的流程与关键技术点。

本书由龙海侠主编、莫壮坚副主编，具体分工为：第1～5章由莫壮坚编写，第6～8章由龙海峡编写，参加编写的还有吴丽华、符湛。本书由李宗民主审。在编写过程中，罗志刚、马生全、何书前等提供了很大的帮助。在此，对他们表示衷心的感谢。

由于时间仓促，加上作者水平有限，书中难免存在一些不足和错误之处，敬请读者批评指正。本书相关资源可以通过http://jc.cepp.sgcc.com.cn进行下载，并进行交流。

编　者

2011年4月

目 录

第 2 篇 数 据 库 操 作

第3篇 案 例 精 讲

第1篇 入 门 篇

本篇介绍如何使用 Java 服务器页将动态内容传递到 Web 页。本篇是对服务器端脚本语言 JSP 的介绍，它首先介绍了创建代码块，如 JSP 语法、脚本元素、隐含对象和标记库。随后，读者将学习如何构造可靠的 JSP 结构、创建实际的 Web 应用程序、实现应用程序安全测量并将数据库集成到自己的 JSP 结构中。这种循序渐进的自学方法会立刻带读者完成 JSP 的入门。

本篇由 4 章组成。第 1 章对 JSP 进行介绍，讲解了 JSP 的发展状况与历史背景，接着用图解的方式对 JSP 环境安装配置进行讲解，并对 JSP 语法进行介绍，最后通过一个 JSP 程序的执行来使读者对 JSP 有大概的了解。第 2 章主要讲解 JSP 的标签、内置组件、语法及其他运用 JSP 来进行 Web 开发时的基本操作，最后通过一个小实例，对本章所学的内容进行巩固。第 3 章主要讲解 JavaBean 组件技术及其基本属性。第 4 章由几个实用例子讲解 JSP 与 Servlet，通过它们使读者对 JSP 有更深的理解，对 Servlet 技术更加明确。每章最后都有一个小结，可以增加读者的理解。

第1章 JSP 概 述

在本章中，读者将了解 JSP 的发展状况与历史背景、接着用图解的方式对 JSP 环境安装配置进行讲解，对 JSP 语法进行介绍，最后通过一个 JSP 程序的执行来使读者对 JSP 有大概的了解。通过对这章的讲解，可以使读者了解 JSP 的背景和开发设计步骤。

本章中读者需要重点掌握的内容主要有：

（1）JSP 技术的特点；

（2）JSP 环境的配置；

（3）了解 JSP 语法。

1.1 JSP 简介与历史背景

WWW 是目前 Internet 上最主要的信息服务类型，它深入影响了政治、科技、商业以及教育等各个领域的发展和进步。WWW 服务的基础是 HTML 语言，而 JSP 正是开发和维护 Web 站点的一种重要工具，它在 HTML 语言的基础上使用脚本语言对网页的对象模型进行编程，为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。

1.1.1 日新月异的 Web 技术

1989 年，瑞士的欧洲原子物理实验室（CEBN）的科学家们提出了一种通过 Internet 共享文档的方法。这些文档中包含有超文本链接和图形信息。它最具革命性的一步是能在不同的计算机平台之间工作。装有不同操作系统的计算机能存储和显示共享信息。超文本链接也

可以指向不同操作系统的计算机上的文档。

1993 年，美国国家超级计算应用中心（NCSA，National Center for Supercomputing Applications）的 Mare Andressen 成功地开发出 Microsoft Windows 上的 Mosaic 软件，该软件很快成为流行的网页浏览器，并在很大程度上促进了 WWW 的流行。随着网页浏览器的层出不穷，Web 的面貌也随之发生了翻天覆地的变化。但是，在 Web 诞生之初的一段较长的时间里，却一直是完全静态的。也就是说，它只是基于一种简单的传输协议，仅用于向用户发送简单的文本信息。不过，最初设计 WWW 的目的完全是为了共享数据信息，或者是帮助学术界解决一些研究论文的共享。所以，它只是通过超链接将一系列的文件连接起来，这在当时看起来已经足够好了。

在当今的 Web 世界里，有几十万甚至上百万的站点相互之间正在进行着激烈的竞争，它们想尽一切办法来获取用户的注意。简单的、静态的页面是无法完成这个任务的。动态的、有条理的数据和友好的、交互性较强的界面，再加上丰富多彩的内容，才是用户所乐于访问的。当然，与此同时数据的自动更新也是非常重要的。

现在，在 Web 页面内创建应用程序、访问数据库，使其无论在感觉上、操作上以及用途上都与真正的应用程序非常类似。今天，商家们所需要的不再仅仅是一个宣传媒体，而是一个交互性极强的应用平台。利用它，商家可以与潜在的客户、目前的客户、员工以及其他人之间进行沟通，并实施一些在线的服务类商务活动。

1.1.2 什么是 JSP

Java 服务器系统页面（JSP，Java Server Papes）是由 Sun Microsystems 公司倡导的、许多公司参与一起建立的一种动态网页技术标准，其在动态网页的创建中有强大而特殊的功能，它是一种实现普通静态 HTML 和动态 HTML 混合编码的技术。在 Sun 正式发布之后，这种新的 Web 应用开发技术很快便引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。

JSP 是 Java 平台上用于编写包含诸如 HTML、DHTML、XHTML 和 XML 等含有动态生成内容的 Web 页面的应用程序的技术。JSP 技术功能强大、使用灵活，为创建显示动态 Web 内容的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 Web 的应用程序更加容易和快捷，而这些应用程序能够与各种 Web 服务器、Web 应用服务器、浏览器和开发工具共同工作。

许多由 CGI 程序生成的页面大部分仍旧是静态 HTML，动态内容只在页面中有限的几个部分出现。但是包括 Servlet 在内的大多数 CGI 技术及其变种，总是通过程序生成整个页面。JSP 使得我们可以分别创建这两个部分。例如，下面就是一个简单的 JSP 页面。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>欢迎访问</TITLE></HEAD>
<BODY>
<H1>欢迎</H1>
<SMALL>欢迎,
<!-- 首次访问的用户名字为"New User" -->
<!--
```

```
 out.println 用来输出内容
 -->
<% out.println(Utils.getUserNameFromCookie(request)); %>
要设置账号信息，请点击
<A HREF="Account-Settings.html">这里</A></SMALL>
<P>
页面的其余内容。
</BODY></HTML>
```

此页面显示结果如图 1-1 所示。

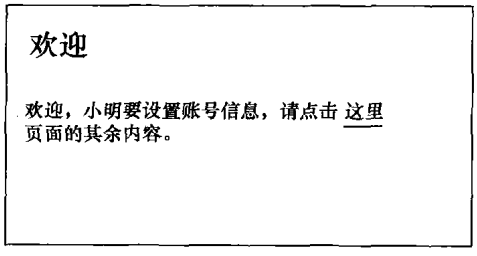

图 1-1 简单的 JSP 页面

1.1.3 JSP 技术的显著优点

1. 把应用程序内容与页面显示分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 XML 标记来设计和格式化最终页面，使用 JSP 标记或者 Scriptlet 来生成页面上的根据请求而变化的动态内容（例如，请求用户信息或者特定的一种商品的价格）。生成内容的逻辑被封装在标记和 JavaBean 组件中，并且捆绑在 Scriptlet 中，所有的脚本程序在服务器端运行。这样 Web 管理人员和页面设计者，能够编辑和使用 JSP 页面，而不影响内容的生成。

在服务器端，由 JSP 引擎解释 JSP 标记和 Scriptlet，生成所请求的内容（例如，通过访问 JavaBean 组件，使用 JDBC 技术访问数据库，或者包含文件），并且将结果以 HTML（或者 XML）页面的形式发送回浏览器。这样既可以保护作者自己的程序代码，又可以保证任何基于 HTML 的 Web 浏览器的完全可用性。

与 Servlet 相比，JSP 能提供所有 Servlet 的功能，但它比用 println 编写和修改 HTML 更方便。此外，还可以更明确地进行分工，Web 页面设计人员编写 HTML，只需要留出地方让 Servlet 程序员插入动态部分即可。

2. 一次编写，到处运行

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，而且所有的 JSP 页面都要被编译成为 Servlet，JSP 页面就具有 Java 技术的所有优点，包括健壮的存储管理和安全性等。当然其中最重要的一点就是“一次编写，到处运行”。

JSP 技术是与设计平台完全无关的，包括它的动态 Web 页面、它的 Web 服务器和底层的服务器组件。用户可以在任何平台上编写 JSP 页面，在任何 Web 服务器或者 Web 应用服务器上运行，或者通过任何 Web 浏览器访问。还可以在任何平台上建立服务器组件并且在任何服务器上运行它们，目前主要是 JavaBean 和 Servlet。有了这个优点，随着越来越多的供应商将 JSP 支持添加到他们的产品中，用户就可以使用自己所选择的服务器和工具，更改工具或服务器并不会影响到当前的应用。

3. 强调可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件（JavaBeans 或者企业版的 JavaBeans 组件）来执行应用程序中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，或者使得这些组件为更多的使用者或者客户团体所使用。这些组件有助于将网页的设计与逻辑程序的编写分开，节约了开发时间，同时充分发挥了 Java 和其他脚本语言的跨平台的能力和灵活性。基于组件的方法加速了总体的开发过程，并且使得各种组织能在他们现

有的技能和优化结果的开发努力中得到平衡。

4. 采用标记简化页面的开发

Web 页面开发人员不一定都是熟悉脚本语言的编程人员。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 标记中进行动态内容生成时所必需的。标准的 JSP 标记能够访问和实例化 JavaBean 组件、设置或者检索组件属性、下载 MySQL，以及执行用其他方法难于编码和耗时的功能。

1.1.4 JSP 和类似技术的比较

1. 类似技术简介

目前，最常用的三种动态网页语言是 ASP（Active Server Pages）、JSP（JavaServer Pages）和 PHP（Hypertext Preprocessor）。

ASP 是一个 Web 服务器端的开发环境，利用它可以产生和执行动态的、互动的、高性能的 Web 服务应用程序。ASP 采用脚本语言 VBScript（Java script）作为自己的开发语言。

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法，并耦合自己的特性，使 Web 开发者能够快速地写出动态产生页面。它支持目前绝大多数数据库。还有一点，PHP 是完全免费的，用户可以从 PHP 官方站点（http://www.PHP.net）自由下载。而且用户可以不受限制地获得源码，甚至可以从中加进用户自己需要的特色。

JSP 是 Sun 公司推出的新一代网站开发语言，Sun 公司借助自己在 Java 上的不凡造诣，在 Java 应用程序和 Java Applet 之外，又有新的硕果，就是 JSP，JSP 可以在 Serverlet 和 JavaBean 的支持下，完成功能强大的站点程序。

三者都提供在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。但 JSP 代码被编译成 Servlet 并由 Java 虚拟机解释执行，这种编译操作仅在对 JSP 页面的第一次请求时发生。在 ASP、PHP 和 JSP 环境下，HTML 代码主要负责描述信息的显示样式，而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器，而 ASP、PHP 和 JSP 页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到 HTML 代码中，然后一起发送给浏览器。ASP、PHP 和 JSP 三者都是面向 Web 服务器的技术，客户端浏览器不需要任何附加的软件支持。

2. 技术特点对比

（1）ASP 的技术特点。

1）使用 VBScript、JScript 等简单易懂的脚本语言，结合 HTML 代码，即可快速地完成网站的应用程序。

2）无须编译、容易编写，可在服务器端直接执行。

3）使用普通的文本编辑器，如 Windows 的记事本，即可进行编辑设计。

4）与浏览器无关（Browser Independence），客户端只要使用可执行 HTML 码的浏览器，即可浏览 ASP 所设计的网页内容。ASP 所使用的脚本语言（VBScript、JScript）均在 Web 服务器端执行，客户端的浏览器不需要执行这些脚本语言。

5）ASP 能与任何 ActiveX Scripting 语言兼容。除了可使用 VB Script 或 JScript 语言来设计外，还通过 plug in 的方式，使用由第三方所提供的其他脚本语言，如 REXX、Perl 和 Tcl 等。脚本引擎是处理脚本程序的 COM（Component Object Model）对象。

6）可使用服务器端的脚本来产生客户端的脚本。

7）ActiveX Server Components（ActiveX 服务器组件）具有无限可扩充性。可以使用 Visual Basic、Java、Visual C++和 COBOL 等程序设计语言来编写用户所需要的 ActiveX Server Components。

（2）PHP 的技术特点。PHP 可以编译成具有与许多数据库相连接的函数。PHP 与 MySQL 是现在绝佳的群组合。用户还可以自己编写外围的函数去间接存取数据库。通过这样的途径，当用户更换使用的数据库时，可以轻松地修改编码以适应这样的变化。PHP LIB 就是最常用的可以提供一般事务需要的一系列数据库。但 PHP 提供的数据库接口支持彼此不统一，例如对 Oracle、MySQL 和 Sybase 的接口，彼此都不一样。这也是 PHP 的一个弱点。

（3）JSP 的技术特点。

1）将内容的产生和显示进行分离。使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面。使用 JSP 标识或者小脚本来产生页面上的动态内容。产生内容的逻辑被封装在标识和 JavaBeans 群组件中，并且捆绑在小脚本中，所有的脚本在服务器端执行。如果核心逻辑被封装在标识和 Beans 中，那么其他人，如 Web 管理人员和页面设计者，能够编辑和使用 JSP 页面，而不影响内容的产生。在服务器端，JSP 引擎解释 JSP 标识，产生所请求的内容（例如，通过存取 JavaBeans 群组件，使用 JDBC 技术存取数据库），并且将结果以 HTML（或者 XML）页面的形式发送回浏览器。这既有助于作者保护自己的代码，而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2）强调可重用的群组件。绝大多数 JSP 页面依赖于可重用且跨平台的组件（如 JavaBeans 或者 Enterprise JavaBeans）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，或者使得这些组件为更多的使用者或者用户团体所使用。基于组件的方法加速了总体开发过程，并且使得各种群组织在他们现有的技能和优化结果的开发努力中得到平衡。

3）采用标识简化页面开发。

Web 页面开发人员不会都是熟悉脚本语言的程序设计人员。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产生所需要的。标准的 JSP 标识能够存取和实例化 JavaBeans 组件，设定或者检索群组件属性，下载 MySQL，以及执行用其他方法更难于编码和耗时的功能。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建立自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。JSP 技术很容易整合到多种应用体系结构中，以利用现存的工具和技巧，并且扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分，以及 Java 2EE 的一个成员，JSP 技术能够支持高度复杂的基于 Web 的应用。由于 JSP 页面的内置脚本语言是基于 Java 程序设计语言的，而且所有的 JSP 页面都被编译成为 Java Servlet，JSP 页面就具有 Java 技术的所有好处，包括健壮的存储管理和安全性以及 Java 程序设计语言“一次编写，各处执行”的特点。随着越来越多的供货商将 JSP 支持加入到他们的产品中，用户可以使用自己所选择的服务器和工具，修改工具或服务器并不影响目前的应用。

3. 应用范围对比

ASP 是 Microsoft 开发的动态网页语言，也继承了微软产品的一贯传统，只能执行于微软的服务器产品，即 WindowsNT 下的 IIS（Internet MySQL Server）和 Windows 98 下的 PWS

（Personal Web Server）上。UNIX 下也有 ChiliSoft 的组件来支持 ASP，但是 ASP 本身的功能有限，必须通过 ASP+COM 的群组合来扩充，UNIX 下的 COM 实现起来非常困难。

PHP 3 可在 Windows、UNIX 和 Linux 的 Web 服务器上正常执行，还支持 IIS 和 Apache 等一般的 Web 服务器，用户更换平台时，无需变换 PHP 3 代码，可即拿即用。

JSP 同 PHP 3 类似，几乎可以执行于所有平台，如 Win NT、Linux 和 UNIX。在 NT 下 IIS 通过一个外加服务器，如 JRUN 或者 ServletExec，就能支持 JSP。知名的 Web 服务器 Apache 也能够支持 JSP。由于 Apache 广泛应用在 NT、UNIX 和 Linux 上，因此 JSP 有更广泛的执行平台。虽然现在 NT 操作系统占了很大的市场份额，但是在服务器方面 UNIX 的优势仍然很大，而新崛起的 Linux 更是来势不小。从一个平台移植到另外一个平台，JSP 和 JavaBean 甚至不用重新编译，因为 Java 字节码都是标准的、与平台无关的。

4. 性能比较

有人做过试验，对这三种语言分别做回圈性能测试及存取 Oracle 数据库测试。在循环性能测试中，JSP 只用了令人吃惊的 4s 就结束了 20 000*20 000 的回圈。而 ASP、PHP 测试的是 2000*2000 循环（少一个数量级），却分别用了 63s 和 84s（参考 PHP LIB）。数据库测试中，三者分别对 Oracle 8 进行 1000 次 Insert、Update、Select 和 Delete，JSP 需要 13s，PHP 需要 69s，ASP 则需要 73s。

5. 前景分析

目前在国内，PHP 与 ASP 应用最为广泛。而 JSP 由于是一种较新的技术，国内采用的较少。但在国外，JSP 已经是比较流行的一种技术，尤其是电子商务类的网站，多采用 JSP。采用 PHP 的网站有新浪网（Sina）、中国人（Chinaren）等，但由于 PHP 本身存在的一些缺点，使得它不适合应用于大型电子商务站点，而更适合一些小型的商业站点。首先，PHP 缺乏规模支持。其次，缺乏多层结构支持。对于大负荷站点，解决方法只有一个：分布计算。数据库、应用逻辑层、表示逻辑层彼此分开，而且同层也可以根据流量分开，群组成二维数组。而 PHP 则缺乏这种支持。还有上面提到过的一点，PHP 提供的数据库接口支持不统一，这就使得它不适合运用在电子商务中。

ASP 和 JSP 则没有以上缺陷，ASP 可以通过 Microsoft Windows 的 COM/DCOM 获得 ActiveX 规模支持，通过 DCOM 和 Transcation Server 获得结构支持；JSP 可以通过 Sun Java 的 Java Class 和 EJB 获得规模支持，通过 EJB/CORBA 以及众多厂商的 Application Server 获得结构支持。

三者中，JSP 应该是未来发展的趋势。世界上一些大的电子商务解决方案提供商都采用 JSP/Servlet。比较出名的有 IBM 的 E－business，它的核心是采用 JSP/Servlet 的 Web Sphere。它们都是通过 CGI 来提供支持的。但自从它推出了 Enfinity，一个采用 JSP/Servlet 的电子商务 Application Server 后，就声言不再开发传统软件了。

总之，ASP、PHP 和 JSP 三者都有相当数量的支持者，由此也可以看出三者各有所长。正在学习或使用动态页面的朋友可根据三者的特点选择一种适合自己的语言。

1.2 JSP 环境安装配置

JSP 环境配置有好多种，下面我们就用 Tomcat 下的配置来介绍。

1.2.1 Tomcat 下 JSP 环境的配置

1. 下载 j2sdk 和 Tomcat

到 Sun 官方站点（http://java.sun.com/j2se/1.4.2/download.html）下载 j2sdk，注意下载版本为 Windows Offline Installation 的 SDK，同时最好下载 J2SE 1.4.2 Documentation，然后到 Tomcat 官方站点（http://www.apache.org/dist/jakarta/Tomcat-5/）下载 Tomcat（下载较稳定的 5.0.x 版本的 Tomcat）。

2. 安装和配置用户的 j2sdk 和 Tomcat

执行 j2sdk 和 Tomcat 的安装程序，按默认设置进行安装即可。

（1）安装 j2sdk 以后，需要配置环境变量，在我的电脑→属性→高级→环境变量→系统变量中添加以下环境变量（假定用户的 j2sdk 安装在 C:\j2sdk1.4.2）：

```
JAVA_HOME=C:\j2sdk1.4.2
Classpath=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;(.;一定不能少，因为它代表当前路径)
Path=%JAVA_HOME%\bin
```

接着可以写一个简单的 Java 程序来测试 j2sdk 是否已安装成功：

```
public class example1{
public static void main(String args[]){
System.out.println("This is a test program.");
}
}
```

将上面的这段程序保存为文件名为 example1.java 的文件。开始→运行→cmd，打开命令提示符窗口，cd 到用户的 example1.java 所在目录，然后输入下面的命令：

```
javac example1.java          //java 的编译命令 javac
java example1                //执行 Test.java 类
```

此时如果看到打印出来“This is a test program.”说明安装成功了，如果没有打印出这句话，则需要仔细检查配置情况。

（2）安装 Tomcat 后，在我的电脑→属性→高级→环境变量→系统变量中添加以下环境变量（假定用户的 Tomcat 安装在 C:\Tomcat5）：

```
CATALINA_HOME=C:\Tomcat5;
CATALINA_BASE=C:\Tomcat5;
```

然后修改环境变量中的 classpath，把 Tomcat 安装目录下的 common\lib 下的 Servlet.jar 追加到 classpath 中去，修改后的 classpath 如下：

```
classpath=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;%CATALINA_HOME%\common\lib\Servlet.jar;
```

接着启动 Tomcat，在 IE 中访问“http://localhost:8080”，如果看到 Tomcat 的欢迎页面说明安装成功了。

注：8080 为 Tomcat 使用的端口，可以在配置文件 Tomcat 目录下的 conf\server.xml 进行修改。

1.2.2 建立自己的 JSP 工作目录

具体步骤如下：

（1）到 Tomcat 的安装目录的 webapps 目录，可以看到 ROOT、examples 和 Tomcat-docs

之类 Tomcat 自带的的目录；

（2）在 webapps 目录下新建一个目录，起名叫 myapp；

（3）myapp 下新建一个目录 WEB-INF，注意，目录名称是区分大小写的；

（4）WEB-INF 下新建一个文件 WEB.xml（也可从 examples 目录下的 webapp 下复制过来），内容如下：

```
<?xml version="1.0" encoding ="ISO 8859-1"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
<display-name>My Web Application</display-name>
<description>
A application for test.
</description>
</web-app>
```

（5）在 myapp 下新建一个测试的 JSP 页面，文件名为 index.jsp，文件内容如下：

```
<html><body>
<center>
Now time is: <%=new java.util.Date()%>
</center>
</body></html>
```

（6）重新启动 Tomcat；

（7）打开浏览器，输入“http://localhost:8080/myapp/index.jsp”，如果看到当前时间说明就成功了。

1.3 JSP 语 法 介 绍

这里简要介绍 JSP 的语法，意为让读者对 JSP 有大概的了解。更为详细的介绍见第 2 章。

1.3.1 JSP 页面中的元素

JSP 使得我们能够分离页面的静态 HTML 和动态部分。HTML 可以用任何通常使用的 Web 制作工具编写，编写方式也和原来的一样；动态部分的代码放入特殊标记之内，大部分以“<%”开始，以“%>”结束。例如，下面是一个 JSP 页面的片断，如果我们用“http://host/test1.31.jsp?title=Core+Web+Programming”这个 URL 打开该页面，则结果显示“Thanks for ordering Core Web Programming”。

test1.31.jsp 源程序如下：

```
Thanks for ordering
<I><%= request.getParameter("title") %></I>
```

JSP 页面文件通常以.jsp 为扩展名，而且可以安装到任何能够存放普通 Web 页面的地方。虽然从代码编写来看，JSP 页面更像普通 Web 页面而不像 Servlet，但实际上，JSP 最终会被转换成正规的 Servlet，静态 HTML 直接输出到和 Servlet Service 方法关联的输出流。JSP 到 Servlet 的转换过程一般在出现第一次页面请求时进行。因此，如果希望第一个用户不会由于

JSP 页面转换成 Servlet 而等待太长的时间，希望 Servlet 已经正确地编译并装载，可以在安装 JSP 页面之后自己请求这个页面，这样 JSP 页面就转换成 Servlet 了。

另外也请注意，许多 Web 服务器允许定义别名，所以一个看起来指向 HTML 文件的 URL 实际上可能指向 Servlet 或 JSP 页面。

除了普通HTML代码之外，嵌入JSP页面的其他成分主要有以下三种：脚本元素（Scripting Element）、指令（Directive）和动作（Action）。脚本元素用来嵌入 Java 代码，这些 Java 代码将成为转换成的 Servlet 的一部分；JSP 指令用来从整体上控制 Servlet 的结构；动作用来引入现有的组件或者控制 JSP 引擎的行为。为了简化脚本元素，JSP 定义了一组可以直接使用的变量（预定义变量），如前面代码片断中的 request。

1.3.2 JSP 语法概要

1. JSP 表达式

<%= expression %>用于计算表达式并输出结果。等价的 XML 表达为：

```
<jsp:expression>
expression
</jsp:expression>
```

可以使用的预定义变量包括：request、response、out、session、application、config 和 pageContext。这些预定义变量也可以在 JSP Scriptlet 中使用。

JSP Scriptlet <% code %>用于将代码插入到 Service 方法。等价的 XML 表达为：

```
<jsp:scriptlet>
code
</jsp:scriptlet>
```

2. JSP 声明

<%! code %>用于将代码插入到 Servlet 类（在 Service 方法之外）。等价的 XML 表达为：

```
<jsp:declaration>
code
</jsp:declaration>
```

3. page 指令

<%@ page att="val" %>是作用于 Servlet 引擎的全局性指令。等价的 XML 表达为：

```
<jsp:Directive.page att="val"\>
```

其合法的属性如下：

```
import="package.class"
contentType="MIME-Type"
isThreadSafe="true|false"
session="true|false"
buffer="size kb|none"
autoflush="true|false"
extends="package.class"
info="message"
```

```
errorPage="URL"
isErrorPage="true|false"
language="java"
```

4. include 指令

<%@ include file="URL" %>用于当 JSP 转换成 Servlet 时，指定应当包含的本地系统上的文件。等价的 XML 表达为：

```
<jsp:Directive.include  file="URL"\>
```

其中 URL 必须是相对 URL。利用 jsp:include 动作可以在请求的时候（而不是 JSP 转换成 Servlet 时）引入文件。

5. JSP 注释

<%-- comment --%>用于注释，在 JSP 转换成 Servlet 时被忽略。如果要把注释嵌入结果 HTML 文档，使用普通的 HTML 注释标记<-- comment -->。

6. jsp:include 动作

<jsp:include page="relative URL" flush="true"/>用于当 Servlet 被请求时，引入指定的文件。如果希望在页面转换的时候包含某个文件，使用 JSP include 指令。

注意：在某些服务器上，被包含文件必须是 HTML 文件或 JSP 文件，具体由服务器决定（通常根据文件扩展名判断）。

7. jsp:useBean 动作

<jsp:useBean att=val*/>或者<jsp:useBean att=val*>…</jsp:useBean>用于寻找或实例化一个 Java Bean。可能的属性包括：

```
id="name"
scope="page|request
|session|application"
class="package.class"
type="package.class"
beanName="package.class"
```

8. jsp:setProperty 动作

<jsp:setProperty att=val*/>用于设置 Bean 的属性。合法的属性包括：

```
name="BeanName"
property="propertyName|*"
param="parameterName"
value="val"
```

9. jsp:getProperty 动作

<jsp:getProperty name="propertyName" value="val"/>用于提取并输出 Bean 的属性。jsp:forward 动作<jsp:forward page="relative URL"/>把请求转到另外一个页面。

10. jsp:plugin 动作

jsp:plugin 动作 <jsp:plugin attribute="value"*>用来根据浏览器的类型，插入通过 Java 插件运行 Java Applet 所必需的 OBJECT 或 EMBED 元素。