



北京科海培训中心

网页设计·网站建设培训系列教材

JSP 编程入门与应用实例



李巍 编著

清华大学出版社



北京科海培训中心

例 8 图

► 网页设计·网站建设培训系列教材

JSP 编程入门与应用实例

李 巍 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书较为全面地介绍了 JSP 技术,根据作者的开发经验,由浅入深、循序渐进地介绍了 JSP 的运行系统,基本概念,语法规范及其相关内容,并提供了大量的应用实例。

全书共分为 13 章,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。在附录中,还对在 JSP 中非常重要的 JSP 的类和 JavaBean 等内容作了进一步的介绍。全书与实际开发结合紧密,使读者感到用 JSP 开发是一件很易上手的工具。

本书是为那些对 Web 开发感兴趣的读者而编写的。不论是 Web 编程的高手,还是初学 Web 开发的网络爱好者,他都能够得到很大的帮助;从实际应用的角度来看,也是一本实用的工具书。

图书在版编目(CIP)数据

JSP 编程入门与应用实例/李巍编著. —北京:清华大学出版社,2000.12

ISBN 7-302-04168-7

I. J… II. 李… III. 计算机网络-程序设计 N.TP393

中国版本图书馆 CIP 数据核字(2000)第 79994 号

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: JSP 编程入门与应用实例

作 者: 李 巍

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

网 址: <http://www.tup.tsinghua.edu.cn>

印刷者: 北京朝阳科普印刷厂

发 行: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 18.125 字数: 441 千字

版 次: 2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

印 数: 0001~5000

书 号: ISBN 7-302-04168-7

定 价: 24.00 元

前 言

1. 网络世界的现状

我们生活在一个日新月异的年代，稍不留神，眼前的一切就已经成为过时的东西了。尤其是计算机技术，自诞生以来给人类的生活方式带来了根本性的变化。在最近几年，Internet 技术又得到了持续、迅猛的发展，已成为计算机产业中的一个技术热点。现在只要大家一进入互联网，那些无穷无尽的内容就会让我们目不暇接，当然那些精彩的界面更会给我们带来美好的享受。可以说，促进 Internet 高速发展的主要因素之一就是 WWW 技术。现在即使是一个普通的老百姓都不会对“网上冲浪”、“Internet”等词汇感到陌生。许多人正在或者正准备学习网络的知识，进行网络开发或者建立自己的个人站点。

实际上，当今活跃的 Web 技术是建立在一系列“活跃”的交互操作上的。通常人们用客户/服务器这个词来描述 Web。这是一个交互的概念，一般把提出请求的一方称为客户端，而把响应请求的一方称为服务器端。这种简单的模型是静态的，他们只能对对方的激励作出响应。而在活跃的 Web 中双方都应该是活跃的，只有这样才能把客户机和服务器结合起来产生最强的交互。在这种交互中，可以平衡工作，并在最合理的地方处理工作。这样就引出了我们所说的动态网页的概念。

在 Web 空间中，有几十万乃至上百万的站点相互之间正在进行着激烈的竞争，他们想尽一切办法来吸引用户的注意力。简单的、静态的页面对用户不会有太大的吸引力。只有动态的、有条理的数据加上友好的、交互性强的界面，最后加上丰富的内容，才能构成用户所乐于访问的界面。当然，数据的自动更新也非常的重要。你可能会说，这怎么听起来像是在讨论应用程序呢？是的，在短短的几年时间里，Web 的面貌已经发生了非常重要的变化。现在，我们可以在 Web 页面中创建应用程序、访问数据库，这样它无论在感觉上、操作中还是实际的用途方面都与 Windows 中的应用程序非常的类似。现在，商家们可以与潜在客户、目前的客户、员工以及其他入中的任何一个进行沟通，并实施一些在线的服务类的商业活动。

为了占领 Internet 这个诱人的市场，各个公司和厂家都拿出了自己的一套解决方案。IBM 推出了 Visual Age for Java 和 Net.Data。前者是一个集成的开发环境，用于测试 Java 程序、JavaBean 或者应用程序；而后者是专门为非编程人员设计的，它可以按照用户的要求生成动态的 Web 页面，它们都可以调用 DB2 强大的数据库功能。著名的数据库厂商 Sybase 也在其开发工具 PowerBuilder 6.0 的版本中加入了 Web 特性，使用户可以在浏览器中直接访问“Data Window”等著名控件，并且可以使用它来开发 Web 应用程序。同样，曾经靠着浏览器在 Internet 上一夜腾飞的网景公司，现在也已推出自己的 Web Server 及完整的 Web 开发工具。所有的这些都说明了一个问题：所有的商家都想在这里占领一席之地；可以说 Web 领域是一个必争之地。

在这种情况下, SUN 公司也不甘落后, 在它的倡导下, 许多公司共同参与建立了一种新的动态网页技术标准——Java Server Pages。SUN 应用 Java 社团性过程开发 JSP 技术。在开发 JSP 规范的过程中, SUN 公司与许许多多主要的 Web 服务器、Web 应用服务器和开发工具供应商, 以及各种各样富有经验的开发团体进行合作, 结果找到了一种适合于应用和页面开发人员的开发方法, 它具有极佳的可移植性和易用性。针对 JSP 的产品, SUN 授权了工具提供商 (如 Macromedia), 结盟公司 (如 Apache, Netscape), 最终用户, 协作商及其他人。最近, SUN 将最新版本的 JSP 和 Java Servlet (JSP 1.1, Java Servlet 2.2) 的源代码发放给 Apache, 以求 JSP 与 Apache 紧密结合、共同发展。Apache, SUN 和许多其他公司及个人公开成立了一个健壮的咨询机构以便任何公司和个人都能免费取得信息。这样, SUN 公司就在这个领域中稳稳地站住了脚跟。

Java Server Pages 作为一门新兴的 Web 技术, 目前国内还没有受到它应有的重视。甚至可以说, 现在国内的许多人对 JSP 还很陌生。其实, 这项技术即使在国外也只不过刚刚起步。我想这种状况, 给我国的 Web 发展创造了一个不可多得的机遇。我们可以借此机会大力发展, 来缩短与国外的差距。

JSP 技术为非专业人员开发高水平的网站提供了良好的工具。但是应该通过什么途径来学习呢? 目前, 在市场上关于这方面的书籍还很少, 专门介绍 JSP 的书籍就更不用说了。只有在网上, 我们可以找到一些介绍 JSP 的技术网站; 但是这些多数都是国外的站点, 即使是一些中文站点, 上面所提供的也多数是对一些外文资料的翻译。这些网站, 几乎成了我们现在学习 JSP 的唯一途径。但是, 并不是每一位网络爱好者都有这样的条件的, 更不用说这种学习方法还要求你有一定的英语水平。正是为了适应广大网络爱好者的需要, 笔者将自己所掌握的有关 JSP 技术方面的知识写成本书, 与大家分享, 使得更多的人加入到我们的行列。

本书在编写过程中, 汲取了很多网站中关于 JSP 技术的精华并加以综合, 形成了一个较完整的体系。内容力求由浅入深, 逐步提高, 无论是对一个网络新手, 还是一个久经沙场的网络高手, 都能够起到一定的指导作用, 都可以从本书中找到一些有益的知识。同时, 在本书中, 笔者还总结了以往学习的经验, 在讲解的时候加入了不少的实例, 希望读者读完本书之后, 既学习到了理论, 又掌握了一些解决问题的方法。

2. 本书内容概述

任何一种新技术都不是独立的, 它需要以前的许多技术的支持。在 JSP 技术中, 你可能会遇到诸如脚本语言、HTML 等一些网页制作方面的问题, 还有一些 Java、Servlet、JavaBean 等编程的内容。即使你是一个新手也不要着急, 其实网页的作者并不需要进行太多的 Java 开发。除了对一些方法调用, 在你的 JSP 网页里应尽可能少用 Java 代码, 因为这里毕竟讲的是网页制作。复杂而重复性高的功能应该放到 JavaBean 中去实现。JSP 技术正是为非专业人员开发高水平的网站提供的一种良好的工具, 为不了解 Servlet 的人提供一种利用简单的扩展标记制作动态网页的技术。所以, 其中用到的 Java、Servlet 的知识都是非常有限, 本书会随着内容的展开, 对相关的内容加以简单的介绍。这样使初学者看了本书

之后,也不至感到茫然;而如果你是一位 Web 高手,就可以跳过这些段落。但是 JavaBean 的情况有些不同。要想制作出具有真正的可重用性的 Web 应用程序,完成复杂的功能,就离不开 JavaBean 的使用,所以,我们不但专门写了第 9 章讲解 JavaBean 的简单性质,而且在附录中还对其进行了详细的介绍。

由于时间和篇幅的限制,本书对 JSP 技术的一些相关的内容中的一些非本质的东西(如与数据库的连接等),不可能作全面的讲解,而只能做一个概括性的介绍。但是这些内容在实际的应用中占有十分重要的地位,所以要想真正建立一个网站还需要继续学习这些内容,读者可以找一些相关的书籍来参考。本书侧重于 JSP 的知识及应用,其中有大量的例子供读者学习。相信读者读完本书之后,应该有能力自己动手建立一个动态的网站。

全书共分为 13 章,从介绍网络技术的发展动态、JSP 的诞生背景开始,循序渐进的介绍了 JSP 的运行系统和模式, JSP 的基本概念, JSP 的语法、JavaBean 的使用(因为这本书不是以介绍 JavaBean 为目的,所以虽然它在 JSP 中起着极其重要的作用,仍然没有放在正文中详细介绍。)以及如何处理异常情况等内容。最后三章,完全是 JSP 的应用实例,包括网上购物、网上答题、网上聊天和网上论坛三个例子,这几个例子都是有一些深度的综合问题,相信对读者水平的提高有好处。本书的最大特点就是采用了大量的实例,其中还包括程序代码和一些运行界面。

3. 相关资料

Java Server Pages 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准,它在动态网页的建设方面有其强大而特殊的功能。目前的 JSP 规范的版本比较多,有 0.92、0.91、1.0、1.1 等版本。它们之间的差异比较大,真正实现了 JSP 1.0 版本的规范的产品还不多,而原来已经利用 0.9x 版本制作出来的系统也不能轻易放弃,所以现在基本属于三个版本并行发展的情况。而 1.1 版本基本上是与 1.0 版本同时开发的,但是由于 JSP 1.1 需要新的 Servlet API 的支持,所以未来的 JSP 1.1 将会等到新的 Servlet API 公布之后才会推出。本书是以 JSP 1.0 版本规范为准进行讲解的。下面简要介绍一下这几个版本。

JSP 1.1 public draft

这是 JSP 1.1 规范的草案,目前还没有正式的测试版本,只提供规范的草案开放供人们参考,并附有一些参考的源代码,其中除了标准的 JSP 本身的语法标记外,还包括可扩展性的定制语法标记机制,开发者可以自行定义语法标记,而且可以在任何支持 JSP 1.1 规范的引擎上运行。

JSP 1.1 与 JSP 1.0 相比,增加了以下一些新的特色:

- 加强了对 Web 程序的支持,如对 Web 程序的全域状态控制;
- Web 程序的全域事件处理器;
- 还可以与 J2EE 1.0 配合起来工作,在安全、交易和 session 状态等方面整合。
- 如果需要的话,你可以在 <http://java.sun.com/products/jsp/techinfo.html> 下找到它。

03/2005/07

JSP 1.0

这是目前各个厂商都在加紧追赶的规范版本，也是本书中所依据的规范版本。本书将详细介绍它的规范内容和使用。

如果需要的话，你可以在 <http://java.sun.com/products/jsp> 下找到它。

JSP 0.92

这是 JSP 的整体架构最完善的规范版本，为后来的 JSP 1.0 规范的建立提供了许多重要的概念，同时也是目前最受欢迎的规范版本。虽然在 JSP 1.0 规范中，对 JSP 的语法作了大量的修改，但是这些全都是为了让它更加正规化，而绝不是因为 JSP 0.92 中的功能不够完善。甚至，有些 JSP 0.92 中的功能在 JSP 1.0 中还没有实现，或者是被作为可选择性的内容，而放到 JSP 1.1 中才要求实现。

如果需要的话，你可以在 <http://java.sun.com/products/jsp/jsp092.html> 下找到它。

JSP 0.91

这是被最多的实现引擎支持的规范版本。目前，大多数在 JSP 规范制定前的版本的引擎都至少能够支持它。

如果需要的话，你可以在 http://www.burridge.net/jsp/Spec91/jsp_spec.html 找到它。

4. 致 谢

长期以来我希望能够和广大的网络开发爱好者交流一下自己的经验和体会。这次应科海编辑室之邀，编写这本 JSP 技术开发的书，实在是满足了我这个心愿。但是由于是初次尝试，而且 JSP 技术本身正处于发展阶段，还有很多不确定因素。所以在书中可能有很多的不足之处，希望广大的读者加以指正。我一定非常感激。

其实，在编写的过程中，我同其他的网络开发者一样，也曾经遇到了一些技术性的问题。在这些方面要感谢我的好朋友刘苗，正是在他的支持和帮助下，我才能顺利完成这本书的写作。同时，还要感谢我的家人和朋友的支持，在他们的鼓励和支持下，我克服了种种困难，终于能够如愿以偿。

目 录

第 1 章 JSP 简介	1
1.1 日新月异的 Web 技术	1
1.2 什么是 JSP	1
1.3 JSP 的形式	3
1.4 其他动态网页技术	5
1.4.1 CGI 与 JSP	5
1.4.2 ASP 与 JSP	5
1.4.3 Servlet 与 JSP	9
1.5 展望未来	10
1.6 小结	10
第 2 章 JSP 的实现与运行方式	12
2.1 JSP 的三种实现模型	12
2.1.1 一个简单的模型	12
2.1.2 Servlet 的灵活应用	13
2.1.3 采用企业版的 JavaBean 技术的可扩展处理	13
2.2 JSP 运行环境——JSDK	14
2.2.1 运行 JSDK Server	14
2.2.2 配置 JSDK Server 以运行 JSP	15
2.2.3 JSDK 运行 JSP 时的出错处理方法	16
2.3 编写一个简单的 JSP 页面	16
2.4 JSP 引擎	20
2.4.1 JSP 模式	20
2.4.2 JSP 页面实现类	22
2.4.3 缓冲	25
2.5 小结	26
第 3 章 JSP 的基本概念	27
3.1 对象	27
3.1.1 对象域	27
3.1.2 对象和变量	28
3.1.3 隐含对象	28
3.1.4 pageContext 对象	29
3.1.5 对象的应用	29
3.2 Web 应用程序	30
3.2.1 定义	30
3.2.2 Web 应用程序与 JSP	30
3.2.3 URL 映射	31
3.2.4 实现相对 URL 的规范	32

3.2.5	应用程序和 ServletContexts	32
3.2.6	session	33
3.3	应用模式	35
3.3.1	简单的 2-1/2 层应用模式	35
3.3.2	N 层应用模式	35
3.3.3	松弛的应用程序对	35
3.3.4	重定向请求	36
3.3.5	包含请求	36
3.4	小结	37
第 4 章	JSP 语法基础	38
4.1	JSP 总体语法规则	38
4.1.1	JSP 元素和模板数据	38
4.1.2	JSP 元素的语法	39
4.1.3	开始和结束的标记	39
4.1.4	空元素	39
4.1.5	属性值	40
4.1.6	空白区域	40
4.2	Quoting 规则	41
4.2.1	脚本元素的 Quoting 规则	41
4.2.2	模板数据的 Quoting 规则	42
4.2.3	属性值的 Quoting 规则	42
4.3	JSP 语法分类	43
4.3.1	指令和动作	43
4.3.2	脚本元素	43
4.3.3	标记扩展机制	43
4.4	注释	44
4.4.1	生成在输出到客户端的文件中的注释	44
4.4.2	JSP 注释	44
4.5	小结	45
第 5 章	指令类语法	47
5.1	page 指令	47
5.1.1	属性	48
5.1.2	同步问题	55
5.1.3	指定内容类型	55
5.1.4	传输本地内容	55
5.2	include 指令	56
5.3	taglib 指令	58
5.4	一个简单的欢迎界面及其所用指令分析	59
5.4.1	一个简单的欢迎界面	60
5.4.2	详细分析	62

5.5 小结.....	62
第 6 章 动作语法.....	63
6.1 动作标记的属性.....	63
6.1.1 请求时属性值.....	63
6.1.2 id 属性.....	64
6.1.3 scope 属性.....	65
6.2 <jsp:useBean>.....	66
6.2.1 执行过程.....	66
6.2.2 语法.....	67
6.2.3 属性.....	68
6.3 <jsp:setProperty>.....	69
6.3.1 语法.....	70
6.3.2 属性.....	70
6.4 <jsp:getProperty>.....	71
6.4.1 语法.....	71
6.4.2 属性.....	72
6.5 <jsp:include>.....	72
6.5.1 语法.....	72
6.5.2 属性.....	73
6.5.3 包含机制的总结.....	73
6.6 <jsp:forward>.....	73
6.6.1 语法.....	74
6.6.2 属性.....	74
6.7 <jsp:plugin>.....	74
6.7.1 语法.....	75
6.7.2 属性.....	76
6.8 使用 JSP 动作处理 HTML 表单.....	77
6.8.1 如何处理表单中的数据.....	77
6.8.2 怎样创建一个表单.....	78
6.8.3 一个简单的应用程序.....	78
6.8.4 详细说明.....	81
6.9 小结.....	84
第 7 章 脚本语法.....	85
7.1 声明.....	85
7.1.1 语法.....	85
7.1.2 使用范围.....	86
7.2 小脚本.....	86
7.2.1 语法.....	86
7.2.2 用法.....	87
7.3 表达式.....	88

7.3.1 语法	88
7.4 简单的猜数游戏.....	88
7.4.1 简单的猜数游戏.....	89
7.4.2 详细分析	93
7.5 小结.....	94
第 8 章 XML 概述.....	95
8.1 什么是 XML.....	95
8.2 为什么要用 XML 表示.....	96
8.3 文档类型.....	97
8.3.1 jsp:root 元素	97
8.3.2 公共 ID	98
8.4 指令.....	98
8.4.1 page 指令.....	98
8.4.2 include 指令.....	98
8.4.3 taglib 指令	99
8.5 脚本元素.....	99
8.5.1 声明	99
8.5.2 Scriptlet.....	99
8.5.3 表达式	99
8.6 动作.....	100
8.7 把 JSP 页面转化成 XML 文档	100
8.7.1 quoting 规则	100
8.7.2 请求时属性表达式.....	100
8.8 XML 文档的 DTD.....	101
8.9 小结.....	102
第 9 章 JavaBean	103
9.1 使用 JavaBean	103
9.2 JavaBean 的活动域.....	104
9.3 JavaBean 的生命周期.....	106
9.3.1 建立规则	106
9.3.2 何时被删除	106
9.4 小结.....	106
第 10 章 错误和异常处理	108
10.1 错误处理.....	108
10.1.1 翻译时的错误处理.....	108
10.1.2 客户请求时的错误处理.....	108
10.2 异常处理.....	109
10.2.1 如何增加一个错误页面.....	109
10.2.2 一个 email 查找的例子	110

10.2.3 在 Bean 中处理异常	125
10.2.4 从一个页面中调用错误页面	129
10.2.5 编写错误页面	129
10.2.6 编写简单的跟踪机制	129
10.3 小结	130
第 11 章 实例一	131
11.1 网上购物	131
11.2 网上测试	145
第 12 章 实例二	156
第 13 章 实例三	193
13.1 概述	193
13.2 连接到数据库	193
13.3 代码及说明	204
13.4 编程小结	248
附录 A JSP 的历史与未来	250
附录 B JSP 的类	253
附录 C JavaBean	259
附录 D JSP 语法卡片	271
附录 E JSP 相关资源	275

第 1 章 JSP 简介

WWW 是目前 Internet 上最主要的信息服务类型，它深入影响到政治、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，而在本书中我们要学习的 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 日新月异的 Web 技术

在 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毫无疑问的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（National Center for Supercomputing Applications, NCSA）的 Marc 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是足够好的事了。

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，他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的媒体，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、员工以及其他人员之间进行沟通，并实施一些在线的服务类商务活动。

1.2 什么是 JSP

JSP 是 Java Server Pages（Java 服务器系统页面）的缩写，它是由 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。在 Sun 正式发布 Java Server Pages 之后，这种新的 Web 应用开发技术很快便引起了人们的关注。Java Server Pages 为创建高度动态的 Web 应用提供了一个独特的开发环境。按照 Sun 的说法，Java Server Pages 能够适应市场上包括 Apache WebServer、IIS 在内的 85% 的服务器产品。

Java Server Pages(以下简称为 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。

JSP 技术有以下几个显著的优点：

- **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。

- **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。你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。你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，你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。

- 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开发过程，并且使得各种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。

- 采用标记简化页面的开发

Web 页面开发人员不一定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 Applet，以及执行用其他方法更难于编码和耗时的功能。

通过开发定制化的标记库，JSP 技术是可以扩展的。今后，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记库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标记一样的可以执行特定功能的组件来工作。

1.3 JSP 的形式

一个 JSP 页面从外表看上去，与 HTML 和 XML 并没有什么区别，只是多了一些使用 Java 语言编写的、JSP 引擎能够处理和抽取的额外的组件而已。通常，JSP 组件创建插入最终页面的文本。JSP 中的 Java 代码是通过“<%”和“%>”符号加入到 HTML 代码中间的。

程序 1-1 是一个简单的 JSP 页面的例子。它虽然非常简单，但是其中包含了普通的 HTML 和很多的 JSP 组件。其中包括对时钟 JavaBean 组件的调用；对一个外部文件的包含（用于著作权信息）以及 JSP 表达式和 Scriptlet 等。在这个例子中可以打印出带有年、月的日期，并且还会根据浏览时的时间使用“Good Morning”和“Good Afternoon”来对你表示欢迎。

程序 1-1 一个简单的 JSP 页面

```
<HTML>

<%@ page language="java" imports="com.wombat.JSP.*" %>

<H1>Welcome</H1>

<P>Today is </P>
```



```
<jsp:useBean id="clock" class="calendar.jspCalendar" />

<UL>
<LI>Day: <%=clock.getDayOfMonth() %>
<LI>Year: <%=clock.getYear() %>
</UL>

<% if (Calendar.getInstance().get(Calendar.AM_PM) == Calendar.AM) { %>
Good Morning
<% } else { %>
Good Afternoon
<% } %>
<%@ include file="copyright.html" %>
</HTML>
```

在这段 JSP 源文件中包含了以下的一些内容:

- 一个 JSP 指令可以将有关的信息传送到 JSP 引擎。在这个例子中,

```
<%@ page language="java" imports="com.wombat.JSP.*" %>
```

指出该页面即将访问的一些 Java 编程语言的扩展的位置。指令必须被放置在相应的标记中。

- 固定模板数据,也就是所有 JSP 引擎不能识别的标记,它们将随着结果页面被发送。通常,这些标记是 HTML 标记或者 XML 标记。在上面的例子中包括无序列表和<H1>标记。
- JSP 动作或者标记。这些标记通常作为标准的或定制的标记被实现,并且使用 XML 标记语法。在这个例子中,在<jsp:useBean>标记中实例化了服务器端的一个 JavaBean——clock。
- 表达式,也就是在标记之间的要求 JSP 引擎计算的内容。在上面的列表项中,时钟组件(clock)的 Day 和 Year 属性值作为字符串返回,并且作为输出插入到 JSP 文件中。在上面的例子中,第一个列表项是日期,第二个是年份。
- Scriptlet 是执行不为标记所支持的功能或者将所有的东西捆绑在一起的短小的脚本语言程序片断。JSP 1.0 的内置脚本语言是基于 Java 语言的。在上面示例中的 Scriptlet 用来确定现在是上午还是下午,并且据此来欢迎用户。

这个例子可能很小,但是它所包含的技术却很全面。在实际应用中,可以将关键的处理封装在服务器端的组件中,这样 Web 开发人员就能够使用熟悉的语法和工具很容易地访问这些信息。基于 Java 的 Scriptlet 为执行不同的功能提供了一种灵活的方式,而且不要求扩展脚本语言。JSP 页面作为一个整体是可读和可理解的,这就使得查找或者预防问题以及共享工作内容变得更加容易。

1.4 其他动态网页技术

飞速发展的精巧的基于 Web 的应用程序要求开发技术上的革新。静态的 HTML 对于显示相对静态的内容是不错的选择；但是新的发展方向却在于创建交互式的基于 Web 的应用程序。在这些程序中，页面的内容基于用户的请求或者是系统的状态，而不是预先定义的文字，这就为动态网页技术的出现和发展创造了广泛的空间。

1.4.1 CGI 与 JSP

对于创建动态 Web 页面的问题，一个早期的解决方案是使用 CGI-BIN 接口。

对 CGI (Common Gateway Interface 即公共网关接口) 相信许多人都不会陌生，在动态网页出现的早期，几乎所有网络上的复杂的功能都是由 CGI 实现的。开发人员需要编写与接口相关的单独的程序，以及基于 Web 的应用程序。这些应用程序通过 Web 服务器来调用有关接口的程序。

这种解决方案不仅对开发人员有很高的要求而且还存在着严重的可扩展性问题。因为每个新的 CGI 都要求在服务器上新增加一个进程。如果多个用户并发地访问该程序，这些进程将消耗该 Web 服务器上的所有的可用资源，而且整个系统性能也将降低到极其低下的程度。

无论是现在的 ASP 还是 JSP 技术，它们都直接与 Web 服务器相连接，不需要通过既占时间又占资源的标准输入输出接口来执行程序后，再将数据回传，所以比 CGI 的运行效率要高得多。另外 CGI 的 Web 程序也难以维持它们的架构，因为 CGI 没有专门的技术可以将一组 CGI 程序组织成 Web 程序，并在互相之间共享资源。而对于 ASP 与 JSP 技术，它们与 Web 服务器一起启动、一起执行、直到关闭，因此很容易编写程序管理和维护连接的系统。所以 CGI 不适于发展复杂的 Web 程序。

1.4.2 ASP 与 JSP

Active Server Pages (活动服务器页面，以下简称为 ASP) 是由 Microsoft 提出的动态网页解决方案。它可以完成：产生动态网页；处理 HTML 表单；创建数据库驱动的网页；跟踪用户会话；创建可搜索网页和将定制组件同网页结合在一起等众多的功能。

ASP 和 JSP 有很多相似之处。因为，它们两者都提供在 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。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 和 JSP 页面还需要附加的语言引擎来分析和执行程序代码。分析和执行完毕以后，程序代码的执行结果被重新嵌入到 HTML 代码当中，然后一起发送给浏览器。ASP 和 JSP 都是面向 Web 服务器的技术，客户端的浏览器不需要任何附加的软件支持。总而言之，两者都是为基于 Web 应用实现动态交互网页制作提供的技术环境支持；两者都能够为程序开发人员提供实现应用程序的编写与自带组件设计网页从逻辑上分离的技术；两者都能够替代 CGI 使网站建设与发展变得较为简单与快捷。