

从实际项目实施的角度，结合丰富的开源工具，完整展示JSP开发技术

# 精通 JSP 编程

赵强 编著



包含全部实例代码和开发环境配置的视频演示

Java技术大系

精通

# JSP编程

赵强 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书不仅详细介绍 JSP 2.0 开发的基础知识（包括 Servlet 开发、JSP 语法、使用 JavaBean、自定义标签、JSTL、表达式语言），还从满足实际开发要求的角度出发，深入讲解包括 Web 应用设计模式、数据库开发、日志记录、文件上传下载、图形图表、开发框架、测试、Web 安全性、性能优化等实用技术。本书提供了大量的源码，用户可以参考这些完整展示技术特性的实例，构建自己的应用程序。全书共分为 7 篇 23 章。

本书还在配套光盘中提供一套视频教程，以直观的展现形式，使读者轻松掌握 JSP 开发环境配置及开发工具的用法。

本书可供 Java 程序员作为 JSP 2.0 入门及深入学习的实用参考书，也可供在校师生作为相关课程的教材或实践指导书。

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

精通 JSP 编程 / 赵强编著. —北京：电子工业出版社，2006.3

（Java 技术大系）

ISBN 7-121-02289-3

I. 精… II. 赵… III. JAVA 语言—主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字（2006）第 010907 号

责任编辑：张 毅 zhangyi@phei.com.cn

印 刷：北京东光印刷厂

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

经 销：各地新华书店

开 本：787×1092 1/16 印张：31.5 字数：722 千字

印 次：2006 年 3 月第 1 次印刷

印 数：6000 册 定价：49.00 元（含光盘 1 张）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

# 前 言

欢迎大家选择本书！JSP 作为十分成熟的技术，已经有大量的书籍进行介绍，那么为何还要写这样一本书呢？本书又有哪些特色呢？

笔者通过多年的 J2EE 开发及培训实践，发现掌握 JSP 的语法规则很简单，但是想要用好 JSP，开发出高质量的 JSP 应用程序并不是一件简单的事情。即便掌握了大量的 JSP 相关知识，包括 JSP 开发常用的组件、开发框架、Web 安全性、Web 测试、性能优化等，也未必能保证胜任实际的 JSP 开发工作。出现这种状况的原因很简单——堆砌起来的知识是不能发挥效能的，需要融会贯通，学以致用。本书写作的目的就是希望能够全面讲解以 JSP 为核心的知识框架，并告诉读者如何综合运用所学的知识以解决问题。

本书不仅详细介绍 JSP 2.0 开发的基础知识，还从满足实际开发要求的角度出发，详细讲解包括 Web 应用设计模式、数据库开发、日志记录、文件上传下载、图形图表、开发框架、测试、Web 安全性、性能优化等实用技术。

本书提供大量的源码，用户可以参考这些业务背景虽然简单但完整展示技术特性的实例，构建自己的应用程序。

为使初学者能够快速掌握 JSP 2.0 开发方法，本书还在配套光盘中提供一套视频教程，内容包括配置 Tomcat、WebLogic Server 的安装与配置、MySQL 数据库的用法、使用 JBuilder 进行 JSP 开发等。通过视频这种直观的展现形式，读者可以轻松掌握 JSP 开发环境配置以及开发工具的用法。

## 本书的主要内容

全书共分为 7 篇 23 章。

第一篇“JSP 基础知识”。主要介绍 JSP 开发的一些基本知识，包括 JSP 技术的基本原理、开发环境的配置等内容。

第二篇“JSP 核心技术”。主要介绍 JSP 及 Servlet 的核心技术，包括程序结构、基本概念、调用 JavaBean、保持客户端状态等内容。

第三篇“常用组件”。在 Web 开发中，有一些公共的需求（例如记录日志、文件上传下载）可以使用已有的组件完成。本篇就讲述如何使用 Log4j、jspSmartUpload、JFreeChart 组件，完成日志记录、文件上传下载、绘制图形图表。

第四篇“标签与表达式”。标签是简化 JSP 开发、提高代码重用性的重要手段，本篇讲述如何开发自定义标签，以及如何使用成熟的标签库——JSTL。

第五篇“开发框架”。开发框架是比组件更高一层次的复用技术，基于成熟的开发框架进行开发，可以取得事半功倍的效果，本篇选择了目前比较流行的两个开发框架 Struts 和 Spring 进行介绍。

第六篇“开发实践”。本篇主要针对 JSP 开发实践中经常遇到的一些问题展开讨论，包括数据库开发、测试、性能优化、Web 安全等主题。

第七篇“综合实例”。这是本书的最后部分，给出一个运用本书所讲技术实现的综合实例。该实例业务背景很简单，是一个电子商店，但是在技术展示方面是很全面的。该程序使用了 Struts 框架、MySQL 数据库、Log4j 日志组件、Servlet 过滤器、Hiberante 框架等技术。

## 使用本书的条件

本书中的代码均在 JDK 1.4.1、Tomcat 5.0、WebLogic Server 8.1.2 下测试通过，用户可以安装这些软件以运行本书的例程序。

## 光盘内容

本书配套光盘包含了书中的全部源码，以及作者讲解如何配置开发环境、配置数据库连接池、使用 JBuilder 开发 Web 应用的视频资料。

## 勘误与技术支持

我们欢迎读者将书中发现的问题与学习中的疑问反馈给我们：[editor@broadview.com.cn](mailto:editor@broadview.com.cn)，我们核实后会将勘误信息与疑问的解答发布到 [www.broadview.com](http://www.broadview.com) 网站的论坛上，给广大读者提供一个交流的平台。

作 者

# 目 录

<b>第一篇 JSP 基础知识</b>	
<b>第 1 章 JSP 技术概述</b> ..... 2	
1.1 Web 开发技术..... 2	
1.1.1 Web 开发技术的历史..... 2	
1.1.2 主流 Web 开发技术对比..... 3	
1.1.3 JSP 技术的发展趋势..... 7	
1.2 JSP 技术原理..... 8	
1.3 如何成为一名优秀的 JSP 开发人员..... 10	
1.3.1 基本技能..... 11	
1.3.2 高级技术..... 12	
1.4 小结..... 14	
<b>第 2 章 配置开发环境</b> ..... 16	
2.1 Tomcat 的安装与配置..... 16	
2.1.1 Tomcat 的安装..... 16	
2.1.2 Tomcat 的配置与管理..... 18	
2.2 WebLogic Server 的安装与配置..... 22	
2.2.1 安装 BEA WebLogic Server..... 22	
2.2.2 配置开发域及服务器..... 25	
2.2.3 WebLogic Server 的启动与停止..... 31	
2.3 MySQL 数据库的安装与使用..... 32	
2.3.1 MySQL 简介..... 32	
2.3.2 MySQL 的安装与管理..... 33	
2.3.3 MySQL 的常用操作..... 34	
2.3.4 MySQL 的 JDBC 开发..... 36	
2.4 小结..... 37	
<b>第二篇 JSP 核心技术</b>	
<b>第 3 章 Servlet 技术</b> ..... 40	
3.1 Servlet 介绍..... 40	
3.1.1 什么是 Servlet..... 40	
3.1.2 Servlet 的技术特点..... 40	
3.1.3 Servlet 的生命周期..... 41	
3.1.4 开发一个简单的 Servlet..... 42	
3.2 Servlet 编程接口..... 44	
3.2.1 Servlet 类接口的分类..... 44	
3.2.2 与 Servlet 实现相关..... 44	
3.2.3 与 Servlet 配置相关..... 45	
3.2.4 与 Servlet 异常相关..... 46	
3.2.5 与请求和响应相关..... 46	
3.2.6 会话跟踪..... 47	
3.2.7 与 Servlet 上下文相关..... 48	
3.2.8 Servlet 协作相关..... 49	
3.2.9 过滤..... 49	
3.3 使用 HttpServlet 处理客户端请求..... 50	
3.4 获得 Servlet 的初始化参数..... 53	
3.5 Servlet 的配置..... 55	
3.5.1 实例介绍..... 55	
3.5.2 Servlet 的名字、类和初始化参数..... 58	
3.5.3 启动装入优先级..... 58	
3.5.4 Servlet 的映射..... 59	
3.6 Servlet 之间的共享信息..... 59	
3.7 小结..... 60	
<b>第 4 章 JSP 基本语法</b> ..... 61	
4.1 JSP 的构成..... 61	
4.1.1 一个典型的 JSP 文件..... 61	
4.1.2 分析 JSP 文件中的元素..... 62	
4.2 指令元素..... 65	
4.2.1 页面指令..... 66	
4.2.2 include 指令..... 67	
4.2.3 taglib 指令..... 68	

4.2.4 表达式语言 .....	69	6.2.1 使用<useBean>动作 .....	107
4.3 脚本元素 .....	69	6.2.2 使用<setProperty>和 <getProperty>动作 .....	111
4.3.1 声明 (Declaration) .....	69	6.3 小结 .....	116
4.3.2 表达式 (Expression) .....	70	<b>第 7 章 保持客户端状态</b> .....	117
4.3.3 Scriptlet .....	70	7.1 使用会话保持客户端状态 .....	117
4.4 动作元素 .....	72	7.1.1 会话的实现原理 .....	117
4.4.1 <jsp:param> .....	73	7.1.2 HttpSession 接口 .....	119
4.4.2 <jsp:include> .....	73	7.1.3 会话与浏览器的关系 .....	120
4.4.3 <jsp:forward> .....	75	7.2 使用 session 对象 .....	120
4.4.4 <jsp:useBean> .....	77	7.2.1 什么是 session 对象 .....	120
4.4.5 <jsp:setProperty> .....	77	7.2.2 如何使用 URL 重写 .....	122
4.4.6 <jsp:getProperty> .....	79	7.2.3 会话使用实例 .....	123
4.4.7 <jsp:plugin> .....	79	7.3 会话管理 .....	129
4.4.8 <jsp:fallback> .....	83	7.4 小结 .....	130
4.5 小结 .....	84	<b>第 8 章 Servlet 的高级开发</b> .....	131
<b>第 5 章 JSP 内建对象</b> .....	85	8.1 Servlet 过滤器 .....	131
5.1 内建对象介绍 .....	85	8.1.1 Servlet 过滤器的基本原理 .....	131
5.2 输入输出控制对象 .....	86	8.1.2 Servlet 过滤器开发实例 .....	132
5.2.1 out 对象 .....	86	8.2 Servlet 事件监听 .....	135
5.2.2 request 对象 .....	88	8.2.1 对 Servlet 上下文进行监听 .....	135
5.2.3 response 对象 .....	90	8.2.2 监听 Http 会话 .....	137
5.3 作用域对象 .....	91	8.2.3 对请求监听 .....	140
5.3.1 作用域对象简介 .....	91	8.3 JSP 与 Servlet 的结合使用 .....	142
5.3.2 page 对象 .....	93	8.3.1 模式一 .....	142
5.3.3 pageContext 对象 .....	93	8.3.2 模式二 .....	143
5.3.4 session 对象 .....	95	8.4 小结 .....	150
5.3.5 application 对象 .....	97	<b>第三篇 常用组件</b>	
5.4 流转控制对象 .....	99	<b>第 9 章 文件的上传和下载组件</b> .....	152
5.5 异常处理对象 .....	99	9.1 jspSmartUpload 组件 .....	152
5.6 初始化参数对象 .....	101	9.1.1 jspSmartUpload 组件的安装 与配置 .....	152
5.7 小结 .....	102	9.1.2 jspSmartUpload 组件的主要 对象 .....	153
<b>第 6 章 使用 JavaBean</b> .....	103		
6.1 JavaBean 简介 .....	103		
6.1.1 JavaBean 的组成 .....	103		
6.1.2 JavaBean 的属性 .....	104		
6.2 在 JSP 中使用 JavaBean .....	107		





## 第五篇 开发框架

第 15 章 Struts 框架	270
15.1 Struts 框架的基本原理	270
15.2 Struts 开发实例	280
15.2.1 Struts 初级实例	280
15.2.2 Struts 中级实例	291
15.2.3 Struts 高级实例	302
15.3 小结	315
第 16 章 Spring 框架	316
16.1 Spring 框架简介	316
16.1.1 Spring 的特性	316
16.1.2 Spring 模块	317
16.2 Spring MVC 框架的基本原理	319
16.3 Spring MVC 框架开发实例	320
16.3.1 Spring MVC 实例 ——springapp1	320
16.3.2 Spring MVC 综合实例 ——SpringApp	324
16.3 小结	329

## 第六篇 开发实践

第 17 章 数据库应用开发	332
17.1 JDBC 编程接口	332
17.1.1 JDBC 简介	332
17.1.2 JDBC 的工作方式	333
17.1.3 JDBC 驱动程序	333
17.1.4 JDBC 的主要对象	335
17.1.5 JDBC 开发的基本步骤	338
17.1.6 执行 SQL 语句	339
17.1.7 处理结果集	341
17.1.8 关闭连接	341
17.2 使用数据库连接池	342
17.2.1 连接池的工作原理	342

17.2.2 配置连接池	343
17.2.3 使用数据源获得连接	350
17.2.4 使用连接池的 Web 应用实例	351
17.3 数据分页显示	354
17.4 小结	362
第 18 章 Web 安全性	363
18.1 为 Web 资源设置安全约束	363
18.1.1 定义<security-constraint>元素	364
18.1.2 在 web.xml 中定义<security-role>元素	364
18.1.3 在 web.xml 中定义<login-config>元素	365
18.2 Tomcat 下的安全域	366
18.3 WebLogic Server 下的安全域	368
18.4 JSP 应用安全性漏洞及对策	370
18.4.1 跨站脚本 (cross-site scripting)	370
18.4.2 HTML 注射	372
18.4.3 SQL 注射	373
18.4.4 使用 JavaBean 带来的问题	374
18.5 小结	375

第 19 章 Web 系统测试	376
19.1 Web 系统测试的基本理论	376
19.1.1 功能测试	376
19.1.2 性能测试	377
19.1.3 单元测试	378
19.1.4 其他测试	379
19.1.5 常用的测试工具	379
19.2 使用 JUnit 进行单元测试	381
19.3 使用 JMeter 测试性能	384
19.3.1 JMeter 的特性	384
19.3.2 JMeter 的安装与配置	385
19.3.3 建立测试计划 (Test Plan)	386

19.3.4	开始执行测试计划	389
19.3.5	数据库性能测试	389
19.3.6	JNEstore 的性能测试	390
19.4	小结	394
<b>第 20 章</b>	<b>JSP 性能优化</b>	<b>395</b>
20.1	JSP 性能优化	395
20.1.1	使用 jspInit()方法	395
20.1.2	优化输出操作	397
20.1.3	控制 session 对象	398
20.1.4	在 useBean 动作中使用 合适的范围	399
20.1.5	采用 forward()方法重定向	399
20.1.6	选择正确的包含方式	400
20.2	使用缓存工具	400
20.2.1	OSCache 的安装与配置	401
20.2.2	使用 OSCache 缓存	401
20.3	小结	404

## 第七篇 综合实例

<b>第 21 章</b>	<b>需求分析及技术方案</b>	<b>406</b>
21.1	业务背景介绍	406
21.2	需求分析	407
21.3	架构设计	408
21.4	JNEstore 的功能演示	410
21.5	小结	412

<b>第 22 章</b>	<b>基于 Struts 的 Web 层开发</b>	<b>413</b>
22.1	定义 JNEstore 的配置文件	413
22.2	建立应用程序初始页面	415
22.3	建立显示图书的功能模块	417
22.4	建立用户登录功能模块	421
22.5	建立新用户注册功能模块	424
22.6	建立购物车管理功能模块	428
22.7	建立订单处理功能模块	431
22.8	建立退出应用程序功能 模块	433
22.9	建立图书查找功能模块	433
22.10	应用程序国际化	435
22.11	小结	436
<b>第 23 章</b>	<b>业务层开发</b>	<b>437</b>
23.1	JNEstore 的持久层解决	
23.1	方案	437
23.1.1	Java 持久层常用技术 对比分析	437
23.1.2	JNEstore 的持久层解决方案	439
23.2	使用 Hibernate 实现持久化	440
23.2.1	Hibernate 简介	441
23.2.2	Hibernate 开发的基本技术	441
23.2.3	Hibernate 开发的高级技术	472
23.2.4	实现 JNEstore 的持久层	475
23.3	小结	489

# 第一篇

## JSP 基础知识

本篇内容包括：

第 1 章 JSP 技术概述——主要介绍 JSP 技术的发展过程、与同类技术的对比、JSP 的原理，最后，向 JSP 初学者提供了一份建议，使初学者了解一个优秀的 JSP 开发者应具备的素质。

第 2 章 配置开发环境——主要介绍如何安装、配置应用服务器 Tomcat、WebLogic Server 及 MySQL 数据库。

# 第 1 章 JSP 技术概述

---

目前，JSP 是一种主流的动态网页实现技术，随着 Java 技术的发展，JSP 技术的内容也极大丰富起来。为了让大家对这项技术有一个完整、清晰的了解，本章将重点讲述 JSP 技术的发展过程、实现原理，并通过一个 JSP 的实例展示 JSP 文件的结构。在本章的最后，向 JSP 的初学者提出了一份建议，使初学者了解一个优秀的 JSP 开发者应具备的素质。

本章的主要内容有：

- Web 开发技术介绍
- JSP 技术原理
- 如何成为优秀的 JSP 开发人员

## 1.1 Web 开发技术

随着 Internet 的发展，动态网页技术随之诞生，CGI、ASP、JSP、PHP 等技术陆续出现。本节将对目前主流的动态网页技术进行对比分析，从中来了解 JSP 技术的突出优势。在未来很长一段时间里，JSP 仍然是开发企业级 Web 应用程序的首选。

### 1.1.1 Web 开发技术的历史

Web 这个 Internet 上最热门的应用架构是由 Tim Berners-Lee 发明的。1990 年 11 月，第一个 Web 服务器 `nxoc01.cern.ch` 开始运行，Tim Berners-Lee 在自己编写的图形化 Web 浏览器“WorldWideWeb”上看到了最早的 Web 页面。1991 年，CERN（欧洲粒子物理研究中心，European Particle Physics Laboratory）正式发布了 Web 技术标准。目前，与 Web 相关的各种技术标准都由著名的 W3C 组织（World Wide Web Consortium）进行管理和维护。

从技术层面看，Web 架构的精华有三处：用超文本技术（HTML）实现信息与信息的连接；用统一资源定位技术（URI）实现全球信息的精确定位；用新的应用层协议（HTTP）实现分布式信息共享。这三个特点无一不与信息的分发、获取和利用有关。

Web 服务端的开发技术也是由静态向动态逐渐发展、完善起来的。最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器。一种名为 SSI（通常称为服务器端嵌入，Server Side Includes）的技术可以让 Web 服务器在返回 HTML 文件前，更新 HTML 文件的某些内容，但其功能非常有限。第一种真正使服务器能根据运行时的具体情况动态生成 HTML 页面的技术是大名鼎鼎的 CGI

(公共网关接口, Common Gateway Interface)。1993年, CGI 1.0 的标准草案由 NCSA (National Center for Supercomputing Applications) 提出; 1995年, NCSA 开始制定 CGI 1.1 标准; 1997年, CGI 1.2 也被纳入了议事日程。CGI 技术允许服务端的应用程序根据客户端的请求动态生成 HTML 页面, 这使客户端和服务端的动态信息交换成为了可能。随着 CGI 技术的普及, 聊天室、论坛、电子商务、信息查询、全文检索等 Web 应用蓬勃发展起来, 人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。

早期的 CGI 程序大多是编译后的可执行程序, 其编程语言可以是 C、C++、Pascal 等任何通用的程序设计语言。为了简化 CGI 程序的修改、编译和发布过程, 人们开始探寻用脚本语言实现 CGI 应用的可行方式。在此方面, 不能不提的是 Larry Wall 于 1987 年发明的 Perl 语言。Perl 结合了 C 语言的高效以及 SH、AWK 等脚本语言的便捷, 似乎天生就适合于编写 CGI 程序。1995年, 第一个用 Perl 编写的 CGI 程序问世。很快, Perl 在 CGI 编程领域的风头就盖过了它的前辈 C 语言。随后, Python 等著名的脚本语言也陆续加入了 CGI 编程语言的行列。

1994年, Rasmus Lerdorf 发明了专用于 Web 服务端编程的 PHP (Personal Home Page Tools) 语言。与以往的 CGI 程序不同, PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面, Web 应用的开发者可以用一种更加简便、快捷的方式实现动态 Web 功能。1996年, Microsoft 借鉴 PHP 的思想, 在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。ASP 使用的脚本语言是我们熟悉的 VBScript 和 JavaScript。借助 Microsoft Visual Studio 等开发工具在市场上的成功, ASP 迅速成为 Windows 系统下 Web 服务端的主流开发技术。

当然, 以 Sun 公司为首的 Java 阵营也不会示弱。1997年, Servlet 技术问世, 1998年, JSP 技术诞生。Servlet 和 JSP 的组合 (还可以加上 JavaBean 技术) 让 Java 开发者同时拥有了类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能, 此外, Java 的运行时编译技术也大大提高了 Servlet 和 JSP 的执行效率, 这也正是 Servlet 和 JSP 被后来的 J2EE 平台吸纳为核心技术的原因之一。

## 1.1.2 主流 Web 开发技术对比

目前, 开发 Web 应用主要使用三种技术: JSP、ASP/ASP.NET 和 PHP。这三种技术都是将程序代码嵌入 HTML 中, 由 HTML 负责信息的表示, 由程序代码负责处理的逻辑, 其执行结果被重新嵌入到 HTML 后一起发送给客户浏览器。

下面对这三种技术进行对比分析。

### 1. 两大开发平台 (J2EE 与 .NET)

Web 服务端开发技术的完善使开发复杂的 Web 应用成为了可能。在此起彼伏的电子商务大潮中, 为了适应企业级应用开发的各种复杂需求, 并给最终用户提供更可靠、更完善的信息服务, 两个最重要的企业级开发平台——J2EE 和 .NET 在 2000 年前后分别诞生于 Java 和 Windows 阵营, 它们随即就在企业级 Web 开发领域展开了较量。

J2EE 是纯粹基于 Java 的解决方案。1998年, Sun 发布了 EJB 1.0 标准。EJB 为企业

级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。至此，J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年，Sun 正式发布了 J2EE 的第一个版本。紧接着，遵循 J2EE 标准，为企业级应用提供支撑平台的各类应用服务软件争先恐后地涌现了出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里最为成功的商业软件平台。随着开源运动的兴起，JBoss 等开源世界里的应用服务新秀也吸引了许多用户的注意力。到 2003 年时，Sun 的 J2EE 已经升级到了 1.4 版，其中三个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。至此，J2EE 体系及相关的软件产品已经成为了 Web 服务端开发的一个强有力的支撑环境。

和 J2EE 不同的是，Microsoft 的 .NET 平台是一个强调多语言间交互的通用运行环境。尽管设计者试图以 .NET 平台作为绝大多数 Windows 应用的首选运行环境，但 .NET 首先吸引的却是 Web 开发者的目光。2001 年，ECMA 通过了 Microsoft 提交的 C# 语言和 CLI 标准，这两种技术标准奠定了 .NET 平台的基石，它们也于 2003 年成为了 ISO 的国际标准。2002 年，Microsoft 正式发布 .NET Framework 和 Visual Studio .NET 开发环境。早在 .NET 发布之前，就已经有许多 Windows 平台的 Web 开发者迫不及待地利用 Beta 版本开发 Web 应用了。这大概是因为，.NET 平台及相关的开发环境不但为 Web 服务端应用提供了一个支持多种语言的、通用的运行平台，而且还引入了 ASP.NET 这样一种全新的 Web 开发技术。ASP.NET 超越了 ASP 的局限，可以使用 VB.NET、C# 等编译型语言，支持 Web Form、.NET Server Control、ADO.NET 等高级特性。客观地讲，.NET 平台，尤其是 .NET 平台中的 ASP.NET 的确不失为 Web 开发技术在 Windows 平台上的一个集大成者。

## 2. JSP 与 ASP.NET

长时间以来，ASP 和 JSP 一直是 Web 开发技术领域的一对竞争对手，如今，ASP.NET 接过了 ASP 的衣钵继续和 JSP 竞争。那么，这两种技术各自有什么优缺点，谁会是最后的赢家呢？

ASP（目前升级为 ASP.NET）高举着微软 .NET 这杆大旗，势头正猛；而 JSP 得益于 J2EE 的树大根深，也毫不示弱，一直在试图开枝展叶。作为一名开发人员，我们自然不必卷入这场争斗中，也不必厚此薄彼。但是，我们必须对 ASP、JSP 有一个比较全面的了解，这样才能在需要的时候选择最合适的工具。

### (1) 语言与性能

ASP（ASP.NET）来源于微软，自然带有鲜明的微软痕迹，它主要应用于 Windows 平台，虽然添加组件后也可以用于 Linux 平台，但是，很多功能都没办法实现。所以，从某种程度上说，ASP（ASP.NET）只适用于 Windows 平台。其支持的编程语言除了 VBScript、JavaScript 外，还有 VB、C++、C# 等（微软方面称支持 20 多种编程语言）。从执行效率上说，ASP.NET 和 ASP 相比，有了很大改进。这是由于在 ASP.NET 下，程序语言先编译成 MSIL，然后 MSIL 再编译成机器码加以执行（MSIL 是一种介于高级程序语言与机器码之间的中间语言）。在用户第一次访问网页的时候，ASP.NET 程序被编译成 MSIL，然后再由 xspisapi.dll 将 MSIL 编译执行。此后 MSIL 被缓存下来，如果有其他用户访问这个页面，网络就可以直接将其 MSIL 编译后执行，因此其执行速度会有

一定提升。而此前的 ASP 每次被访问都要对源代码进行解释执行，效率较低。

JSP 作为 J2EE 的一个重要组成部分，只支持 Java 语言。和 C++ 比较起来，Java 牺牲效率换来了平台无关性，Java 可以在大多数操作系统上运行而不需要重新编译，因为 Java 的编译代码是一种字节代码，在运行时由操作系统上的 Java 虚拟机（Java Virtual Machine，简称 JVM）来解释执行，也正是由于 Java 虚拟机的存在，才实现了跨平台的目的。在性能上，由于 JSP 实质上最终要转化为 Servlet（运行于服务器端的 Java 类，可以响应客户端请求）来执行，而 Servlet 比传统的 Java 程序（Applets, Java App）要快，它们在服务器端运行，不需要加载复杂的 GUI，所以响应很快。Servlet 的字节代码在客户请求时才执行，尽管当首次调用 Servlet 时会有几秒钟的加载时间，但后续的请求响应非常迅速，因为服务器已经缓存了运行的 Servlet（这一点上 ASP.NET 借鉴了 JSP）。当前的 JSP 服务器一般都带有 Java 即时编译器（JIT），因此，JSP 的执行比每次都要解释执行的 ASP 代码要快得多，尤其是在代码中存在循环操作时，JSP 的速度要快 1~2 个数量级；但和 ASP.NET 相比，性能上的优势不是那么明显。

### （2）组件模型

在 ASP（ASP.NET）中可以通过 COM 对象来封装商业逻辑，或者完成高强度的计算。这种模型使得页面的开发快速、简洁。COM 可以用符合标准对象模型的任何语言开发，同时因为组件可以由编译语言（如 C++、C）构成，而不是解释型的脚本语言（VBScript、JScript），所以性能上也有不少的改进。但是，COM 组件最大的缺点就是过于复杂，需要经过大量的学习和培训，即使已经具有了较深厚的 C++ 功底也无法轻松胜任。而且，在 ASP 中还存在组件注册的问题，常常会有不同版本的 DLL 覆盖的现象发生，从而使得 ASP 网页的部署相当复杂。不过，这个问题早就引起了微软的重视；后来的 ASP.NET 针对此缺陷做了不少改进，现在只要用“xcopy”的方式将相关的文件复制到目的机器就可以了。

和微软的 COM 组件相对应，在 JSP 中由 JavaBeans 来完成一些预定义功能的封装。JavaBeans 只能用 Java 语言开发，但是，它更容易开发，因为一旦掌握了 Java，了解了 JavaBeans 的结构，开发 JavaBeans 是非常容易的。另外，和 COM 相比，如果修改了 Java Beans 也不需要重新注册。开发者常常可以创建满足业务逻辑的完整的 JavaBeans 库，让非编程人员使用该库来开发动态网站。

### （3）程序结构

从程序结构上说，ASP 和 JSP 在“< %...% >”中间嵌入代码，而 ASP.NET 中代码是写在“< Script >...< /Script >”之间的。在对 Session（会话）、Application（应用）、include（文件包括）等的管理上，ASP（ASP.NET）和 JSP 有相似的处理办法，但是在具体的细节上有不少区别。

为了跟踪用户的操作状态，ASP.NET（ASP）应用 Session 对象，而 JSP 使用一个叫 HttpSession 的对象来实现同样的功能。Session 的信息保存在服务器端，而 Session 的 ID 保存在客户机的 Cookie 中。如果客户机禁止 Cookie，SessionID 就必须带在 URL 后面。Session 一般在服务器上设置了一个过期时间，当客户停止活动后自动失效。Session 中保存和检索的信息不能是如（int、double）等基本数据类型，而必须是 Java 的相应对象。

另外，在 ASP.NET 中，Session 得到了进一步完善。由于 ASP 中 Session 只能存在于一台服务器上，用户的访问不能跨主机，否则无法识别出是否是同一个访问者。针对这一点，在 ASP.NET 使用了 Server farm，它可以将许多主机组织在一起，给用户一个一致的映象，从而使得 Session 的信息可以跨进程、跨机器。

另外，在 Web 应用中，服务器需要管理面向整个应用的参数，使得每个客户都能获得同样的参数值。为此，ASP.NET 使用了 Application 对象，而 JSP 则使用了 ServletContext 对象，但其操作方法和 Session 相同。

ASP.NET 使用 ODBC 或 ADO.NET（ASP 通过 ADO 对象）连接数据库，而 Java 通过一个叫 JDBC 的 API 接口连接数据库。目标数据库通过 JDBC 驱动程序，使得 Java 可以用标准的方式访问数据库。JDBC 不使用服务器端的数据源，只要有 JDBC 驱动程序，Java 就可以访问数据库了。如果一个特定的数据库没有 JDBC 驱动程序，而只有 ODBC 驱动程序，Java 就会提供一个 JDBC-ODBC 桥来将 JDBC 调用转化为 ODBC 调用。所有的 Java 编译器都带有一个免费的 JDBC-ODBC 桥。理论上，这个桥可以访问任何常见的数据库产品。

总而言之，由于 JSP 模型是在 ASP 之后定义的，它借用了 ASP 的许多优点，如 Session、Application 等对象。同时 JSP 使用灵活而强大的 Java 语言，而不是低效的脚本语言，使得 JSP 与 ASP 相比，具有明显优势。但是，ASP.NET 又是在借鉴 JSP 的优点后推出的，这样 ASP.NET 又具有后发优势。目前看来，ASP.NET 和 JSP 这两种技术仍然在不断完善，两者间的竞争还将继续下去，因为 .NET、J2EE 平台之争没有结束。

### 3. PHP

PHP 是一种跨平台的服务器端嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法，并耦合 PHP 自己的特性，使 Web 开发者能够快速写出动态页面。它支持目前绝大多数数据库。还有一点，PHP 是完全免费的，开发者可以从 PHP 官方站点 (<http://www.php.net>) 自由下载，而且可以不受限制地获得源码，甚至可以从中加进自己需要的特色。下面对 PHP 的主要技术特性进行介绍。

#### (1) 数据库连接

PHP 可以编译成具有与许多数据库相连接的函数（PHP 与 MySQL 是绝佳的组合），还可以自己编写外围的函数去间接存取数据库。当更换使用的数据库时，通过这样的途径可以轻松地更改编码以适应这样的变化。PHPLIB 就是最常用的可以提供一般事务需要的一系列基库。但 PHP 提供的数据库接口支持彼此不统一，比如对 Oracle、MySQL、Sybase 的接口，彼此都不一样。这也是 PHP 的一个弱点。

#### (2) 面向对象编程

PHP 提供了类和对象。基于 Web 的编程工作非常需要面向对象编程能力，而 PHP 正好支持构造器、提取类等。

#### (3) 运行平台

PHP 可在 Windows、UNIX、Linux 的 Web 服务器上正常运行，还支持 IIS、Apache 等通用 Web 服务器，用户更换平台时，无需变换 PHP 代码。



#### (4) 缺陷

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,它缺乏多层结构支持。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组成二维阵列。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,这就使得它不适合在电子商务中运用。

### 1.1.3 JSP 技术的发展趋势

随着软件技术的发展,JSP 技术的内容也不断丰富,例如,使用 XML 进行 Web 系统的数据交换,使用 Web 开发框架等。

#### 1. XML 与 Web Services

如果说 HTML 语言给 Web 世界赋予了无限生机的话,那么,XML 语言的出现大概就可以算成是 Web 的一次新生了。按照 Tim Berners-Lee 的说法,Web 是一个“信息空间”。HTML 语言具有较强的表现力,但也存在结构过于灵活、语法不规范的弱点。当信息都以 HTML 语言的面貌出现时,Web 这个信息空间是杂乱无章、没有秩序的。为了让 Web 世界里的所有信息都有章可循、有法可依,我们需要一种更为规范、更能够体现信息特点的语言。

1996 年,W3C 在 SGML 语言的基础上提出了 XML (eXtensible Markup Language) 语言草案。1998 年,W3C 正式发布了 XML 1.0 标准。XML 语言对信息的格式和表达方法做了最大程度的规范,应用软件可以按照统一的方式处理所有 XML 信息。这样一来,信息在整个 Web 世界里的共享和交换就有了技术上的保障。HTML 语言关心的是信息的表现形式,而 XML 语言关心的是信息本身的格式和数据内容。从这个意义上说,XML 语言不但可以将客户端的信息展现技术提高到一个新的层次,而且可以显著提高服务端信息获取、生成、发布和共享的能力。为了将 XML 信息转换为 HTML 等不同的信息展现形式,1999 年,W3C 制定出了 XSLT 标准。同一年,IE 5.0 增加了对 XML 和 XSLT 的支持。

现在,网站的开发者可以直接使用 XML 语言发布信息了。针对不同的应用领域,人们还制定了许多专门的 XML 规范。例如,2001 年 W3C 发布的 SVG (Scalable Vector Graphics) 1.0 标准就是一种用 XML 语言表达的、全新的二维矢量图形格式。开发者可以用 SVG 格式描述大多数已有的 Flash 动画。与 Flash 格式相比,符合 XML 标准的 SVG 格式显然更有利于信息交换和共享。

Web 本身就是一个最大的分布式应用系统。对于分布式开发而言,XML 技术也大有用武之地。一个明显的事实是,如果能让分布式应用借助 XML 格式交换信息,那么,以往横亘在分布式架构上的信息交换难题也就迎刃而解了。1999 年,W3C 和相关的企业开始讨论设计基于 XML 的通信协议,2000 年,W3C 发布了 SOAP (Simple Object Access Protocol) 协议的 1.1 版。人们把利用 SOAP 协议传递 XML 信息的分布式应用模型称为