

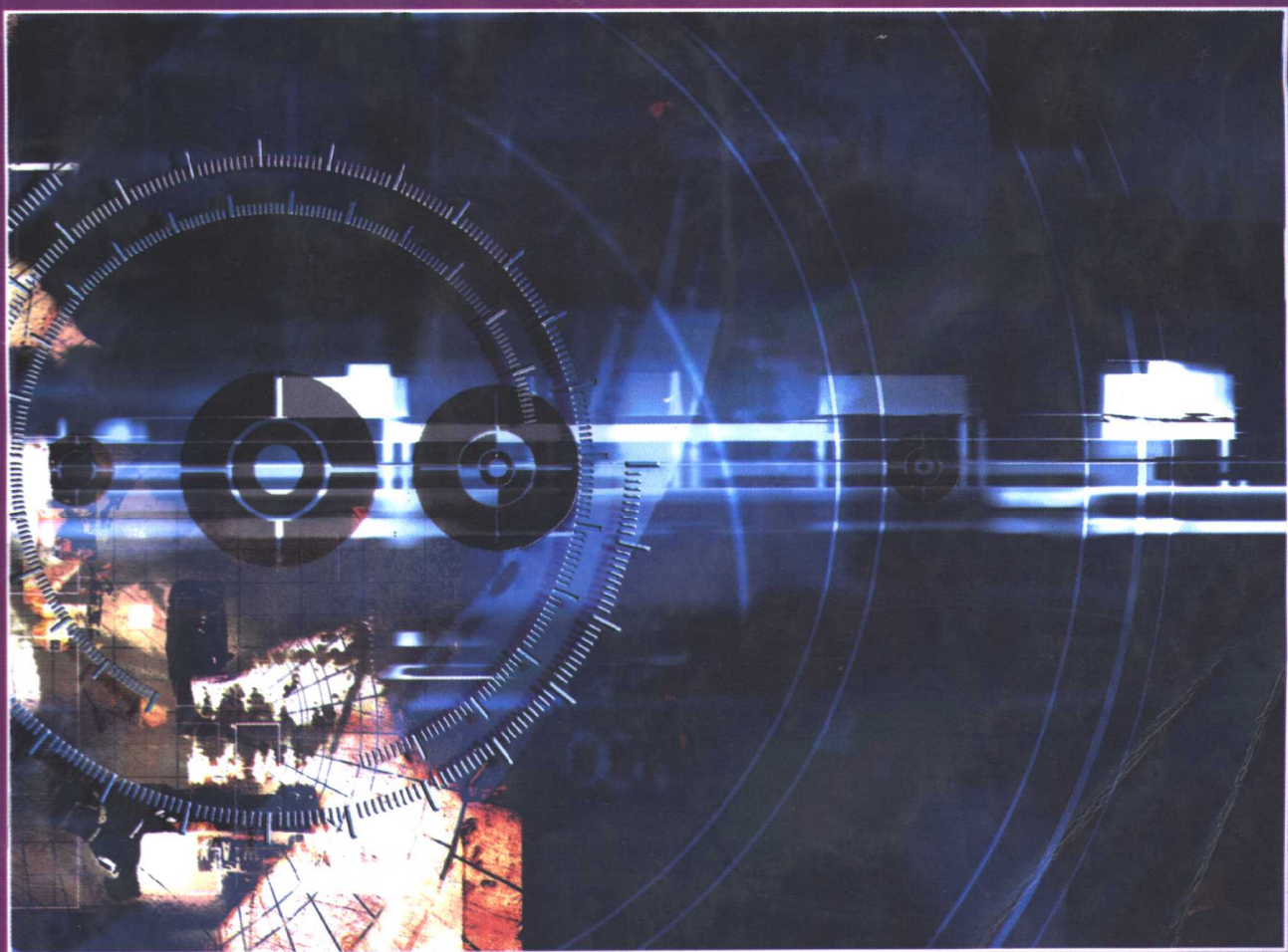
JSP

动态网站建设

高 成 主 编 董长虹 姚德忠 副主编
余 杰 谢 华 编 著

NATIONAL DEFENSE INDUSTRY PRESS

国防工业出版社



动态网站建设丛书

JSP动态网站建设

高 成 主 编

董长虹 姚德忠 副主编

余 杰 谢 华 编 著

国防工业出版社

·北京·

图书在版编目(CIP)数据

JSP 动态网站建设/余杰编著. —北京:国防工业出版社, 2002. 4

(动态网站建设丛书/高成主编)

ISBN 7-118-02807-X

I . J . . . II . 余 . . . III . 因特网 — 程序设计
IV . TP393 . 092

中国版本图书馆 CIP 数据核字(2002)第 006778 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 20½ 470 千字

2002 年 4 月第 1 版 2002 年 4 月北京第 1 次印刷

印数:1—3000 册 定价:28.00 元

(本书如有印装错误,我社负责调换)

编 委 会

主 编： 高 成

副主编： 董长虹 姚德忠

编委会成员：

潘时龙	高海滨	朱 奇	马 迪	黄志正
郭兴成	陈 洁	崔益亮	张乐乐	张华弟
杨连池	贾立峰	王 震	余 杰	崔 桦
谢 华	郭明玄	张 龙	张建宁	张雪莲
廖 俊	李耀华	李品一	李元瑾	李 瑾
何 熙	王 剑	杨广军	赵 阳	汪 洋

前 言

Web 技术是计算机技术发展最快的领域之一，在这短短的几年中，出现了很多出色的开发语言，其中包括 ASP、PHP、PERL 等，借助这些语言，开发基于 Web 的动态网站从一项只有专业程序员才能胜任的难题，变成对于一般的爱好者也能完成的工作。但这些语言中没有一种能兼顾开发效率和执行速度，同时满足分布式处理的需要，以及有良好的移植性和可扩展性。而这些正是 JSP 的优点，JSP 以强大的 Java 语言作为支持，是一种真正的与平台无关的技术，因此它从一问世便受到了业界的广泛关注，很多公司纷纷宣布支持 JSP，很多软件小组也在自己的产品中加入了支持 JSP 的支持。

JSP (Java Server Pages) 是一种高效的动态网站开发语言，和其他流行的开发语言 (如 ASP、PHP) 相比，JSP 在以下几个方面有了重大的突破：

- (1) 通过 JSP 的扩展标签库和 JavaBean 功能，网站逻辑和网站界面可以完美地分离。
- (2) 使用 Enterprise JavaBean，可以轻松地在 JSP 开发的 Web 中实现事务、安全、会话等企业级应用所需要的功能。
- (3) JDBC2.0 提供了与不同的数据库产品无关的数据库连接方式，更重要的是，数据库连接池提供了一种比普通的数据库连接方式效率高得多的连接方式。

本书从最基础的 JSP 语言开始，循序渐进地介绍了 JSP 开发技术，并涵盖了许多高级主题，如需要在企业级 Web 应用中使用的特性——Enterprise JavaBean、JDBC2.0、数据库连接池和自定义标签库等。

全书由十章及三个附录组成。主要介绍的内容包括：JSP 概述、JSP 运行环境的部署、JSP 语法、JSP 内置对象、Java Servlet 技术、JavaBean 和 EJB、JSP 与数据库的连接、JSP 中的文件操作、JavaMail 和开放源码的论坛项目 Jive。三个附录的内容分别是：Java 语法基础、结构化的数据库查询语言 SQL 以及 JSP 中中文问题的分析与解决。

本书语言简明扼要、内容丰富、涉及面广，既适合初学者阅读，也适合具有一定 JSP 基础的开发人员深入研究使用。

内 容 简 介

JSP (Java Server Pages) 是时下最流行的 Internet/Intranet 开发语言, 是开发基于 Web 的动态网络应用程序最有力的工具之一。

本书从最基础的 JSP 技术开始, 循序渐进地介绍了 JSP 开发技术, 并涵盖了许多高级主题, 如需要在企业级 Web 应用中使用的特性——Enterprise JavaBeans、JDBC2.0、数据库连接池和自定义标签库等等。全书由十章及三个附录组成。主要介绍的内容包括: JSP 概述、JSP 运行环境的部署、JSP 语法、JSP 内置对象、Java Servlet 技术、JavaBean 和 EJB、JSP 与数据库的连接、JSP 中的文件操作、JavaMail 和开放源码的论坛项目 Jive。三个附录的内容分别是: Java 语法基础、结构化的数据库查询语言 SQL 以及 JSP 中中文问题的分析与解决。

本书语言简明扼要、内容丰富、涉及面广, 既适合初学者阅读, 也适合具有一定 JSP 基础的开发人员深入研究使用。

目 录

第 1 章 JSP 技术概览

- 1.1 JSP 技术的发展 1
- 1.2 JSP 与相关的 Java 技术 3
 - 1.2.1 Java Servlet 3
 - 1.2.2 Java Bean 3
 - 1.2.3 JDBC 数据库连接技术 ... 4
- 1.3 JSP 与其他 CGI 语言的比较 4
- 1.4 JSP 的简单描述 5
- 1.5 JSP 技术的未来 7

第 2 章 JSP 环境的安装与部署

- 2.1 最简洁的平台——JSDK 10
 - 2.1.1 安装 11
 - 2.1.2 配置 11
- 2.2 Resin 服务器平台介绍 12
 - 2.2.1 最快的 JSP 平台 12
 - 2.2.2 安装与启动 13
 - 2.2.3 将 Resin 装入 Microsoft IIS 15
- 2.3 Tomcat 简介 16
 - 2.3.1 Jakarta - Tomcat 计划 ... 16
 - 2.3.2 Tomcat 安装指南 17
 - 2.3.3 Linux 下的 Tomcat 配置 ... 18
 - 2.3.4 Tomcat 的目录结构 19
- 2.4 其他平台简介 20
 - 2.4.1 BEA 公司的 Weblogic 平台 20
 - 2.4.2 Allaire 公司的 JRun 平台 ... 21
 - 2.4.3 IBM 的 WebSphere Server

平台 23

- 2.5 JSP 开发工具 24
 - 2.5.1 EditPlus + JDK 24
 - 2.5.2 使用 JBuilder 开发 JSP ... 26
- 2.6 本章小结 29

第 3 章 JSP 语法介绍

- 3.1 JSP 语法概述 30
- 3.2 JSP 中的注释 32
 - 3.2.1 HTML 注释 32
 - 3.2.2 隐藏注释 32
- 3.3 变量声明与表达式 33
 - 3.3.1 变量与变量声明 33
 - 3.3.2 表达式 34
- 3.4 包含指示、页面指示和标签库指示 35
 - 3.4.1 包含指示 35
 - 3.4.2 页面指示 36
 - 3.4.3 标签库指示 37
- 3.5 Scriptlet 脚本 37
- 3.6 JSP 标识 39
 - 3.6.1 JSP;useBean 标识 39
 - 3.6.2 JSP;setProperty 和 JSP;getProperty 标识 39
 - 3.6.3 JSP;include 标识 40
 - 3.6.4 JSP;forward 标识 41
- 3.7 本章小结 42

第 4 章 JSP 内置对象

- 4.1 request 对象 43

4.2	response 对象	47
4.3	pageContext 对象	48
4.4	session 对象	49
4.5	Application 对象	52
4.6	Out 对象	54
4.7	其他对象	55

第 5 章 Java Servlet

5.1	Servlet 技术简介	56
5.1.1	Servlet 概况	56
5.1.2	Servlet 的功能	58
5.1.3	与传统的 CGI 和类 CGI 技术的比较	58
5.2	Servlet 编程入门	59
5.2.1	第一个 Servlet 程序	59
5.2.2	编译和运行	61
5.2.3	Servlet 的基本结构和生命 周期	62
5.3	Servlet 编程实例	67
5.3.1	处理表单数据	67
5.3.2	处理 Request Header	74
5.3.3	处理 CGI 变量	76
5.3.4	用 Servlet 实现输出	79
5.4	用 Servlet 控制会话	89
5.4.1	Session tracking	90
5.4.2	Cookie	95
5.5	用 Servlet 连接数据库	99
5.5.1	在 Servlet 中连接数 据库	99
5.5.2	使用 JDBC 连接池	103
5.6	本章小结	112

第 6 章 JavaBean 和 EJB

6.1	JavaBean 简介	113
6.1.1	什么是 JavaBean	113
6.1.2	创建 JavaBean	114
6.2	在 JSP 中调用 JavaBean	119

6.2.1	使用 JSP 标识	119
6.2.2	在 JSP 中调用 JavaBean 的范例	120
6.2.3	用 JavaBean 访问数据库	122
6.3	Enterprise JavaBeans	130
6.3.1	Enterprise JavaBeans 的 总体结构	130
6.3.2	Session Beans 与 Entity Beans	130
6.3.3	Enterprise Bean 的开发	131
6.4	Enterprise JavaBeans 的应用	136
6.5	本章小结	138

第 7 章 JSP 与数据库的连接

7.1	JDBC 概述	139
7.2	JDBC 的工作机制	142
7.2.1	JDBC 组件模型	142
7.2.2	JDBC 连接池	150
7.3	用 JDBC 访问数据库	152
7.4	JDBC 应用示例	155
7.4.1	分页显示查询结果	155
7.4.2	留言板	159
7.5	本章小结	166

第 8 章 JSP 中的文件操作

8.1	文件的读取	167
8.2	文件的写入	171
8.3	文件的追加	176
8.4	文件上传举例	178
8.5	本章小结	184

第 9 章 用 JSP 处理电子邮件

9.1	JavaMail 简介	185
9.1.1	布置 JavaMail 的环境	185
9.1.2	测试 JavaMail	186
9.2	用 JavaMail API 发送邮件	192

9.2.1	JavaMail API 核心类介绍	192
9.2.2	发送邮件例程一	194
9.2.3	发送邮件例程二	199
9.3	邮件的查看和接收	207
9.4	本章小结	214

第 10 章 开放源码的论坛项目 Jive

10.1	Jive 概述	215
10.2	Jive 的安装	216
10.3	Jive 的总体结构	221
10.3.1	数据结构	221
10.3.2	基于上述对象的操作	222
10.4	数据库连接和缓存管理	228
10.4.1	取得数据库的连接	228
10.4.2	管理缓存	237
10.5	使用过滤器处理文本	240
10.6	本章小结	243

附录一 Java 语言介绍

1.1	编写简单的控制台应用程序	245
1.2	Java 语法介绍	250
1.2.1	基本数据类型	250
1.2.2	Java 运算符	258
1.2.3	流程控制语句	258
1.2.4	Java 异常处理机制	260
1.3	Java 与面向对象程序设计	262
1.3.1	类和对象	262
1.3.2	继承、覆盖与重载	264
1.4	常用程序包介绍	266
1.4.1	java.lang 程序包	266
1.4.2	java.util 程序包	269
1.4.3	java.io 程序包	275
1.5	本章小结	278

附录二 结构化数据库查询语言 SQL

2.1	关系型数据库和 SQL 语言	280
-----	----------------------	-----

2.2	用 SQL 语句对表进行操作	282
2.2.1	创建新表	282
2.2.2	删除和修改表	285
2.3	数据表的增删查改	286
2.3.1	查询数据	286
2.3.2	删除记录	289
2.3.3	修改记录	290
2.3.4	添加记录	290
2.4	创建索引字段	291
2.4.1	索引的概念	291
2.4.2	聚簇索引和非聚簇索引	291
2.4.3	索引属性	292
2.4.4	用 SQL 语句建立索引	293
2.5	使用集合函数	293
2.5.1	统计字段值的数目	294
2.5.2	计算字段的平均值	294
2.5.3	计算字段值的和	295
2.5.4	字段的最小值和最大值	295
2.6	其他常用的 SQL 技术	295
2.6.1	提取数据	296
2.6.2	转换数据	296
2.6.3	操作字符串数据	297
2.6.4	操作日期和时间	299
2.7	数据库查询优化技术	303
2.8	本章小结	308

附录三 Java 中文问题的分析及解决

3.1	汉字编码的常识	310
3.2	Java 中文问题的初步认识	311
3.3	Java 中文问题的表层分析及处理	312
3.4	Java 中文问题的根源分析及解决	313
3.5	关于 Java 中文问题的总结	315

第 1 章 JSP 技术概览

1.1 JSP 技术的发展

随着网络计算的迅猛发展, Internet 正在改变人类社会的方方面面, 而 Java 技术的推广更在其中起着推波助澜的作用。Java 作为新一代的编程语言, 以其良好的移植性和安全性, 正在成为一种 IT 业的主流技术, 一些发达国家甚至将它列入了高等学校的必修课程。

纵观近年来 Java 技术的发展, 其主要趋势体现在以下一些方面。

企业的信息化处理, 例如在基于 Web 技术的电子商务体系中, J2EE+XML 被认为是 B2B 电子商务模式的驱动力。而 B2B 模式将是今后企业生存与发展必不可少的基础, 以 BEA 为首的一大批软件公司纷纷以 J2EE 为基础平台, 构造能适合各种企业需要的电子商务解决方案。

2001 年初夏在旧金山举办的 JavaOne2001 大会, 又在 IT 界掀起了一股 Java 热潮。大会的统计资料表明, 全球超过 85% 的大型企业正在用 Java 开发自己的信息系统。很多大公司如 IBM、Oracle、Motorola、BEA 等都在不遗余力的推广 J2EE 的使用。华尔街金融巨头在 2001 年 2 月在世界贸易中心召开会议, 一致认为下一代企业信息化必将以 J2EE 作为基础框架, 在 IT 业界产生了相当大的震动。

J2EE 是一种利用 Java 平台来简化企业级事务处理的体系结构, 其技术基础就是 Java 核心平台或 Java 2 平台的标准版。J2EE 不仅保留了标准版中的许多优点, 例如“编写一次、到处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等等, 同时还提供了对 EJB(Enterprise Java Beans)、Java Servlet API、JSP(Java Server Pages)以及 XML 技术的全面支持。由于 Java Servlet 和 JSP 都会被编译为 Java Class 运行, 因此执行性能会明显提高。

JSP 是 Java Server Pages 的简称, 是将纯 Java 代码嵌入到 HTML 中实现动态功能的技术。作为 J2EE 体系中的重要一环, JSP 这种新的 Web 应用开发技术很快就引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。按照 Sun 公司的说法, JSP 能够适应市场上包括 Apache WebServer、IIS4.0 在内的 85% 的服务器产品。即使您对 ASP 技术“一往情深”, 关注 JSP 的发展仍然是很有必要的。Java Servlet 和 JSP 中完全使用 Java 语言进行编程, 而 Java 的功能要比 ASP 中使用的 VBScript 要强大得多, 它足以胜任绝大多数的商业事务处理。同时 JSP 代码经过服务器的解释执行后会直接嵌入到 HTML 语言中, 这样就简化了页面的输出, 也保留了脚本语言易于使用的优点。

在一般情况下, 构建电子商务系统可以使用单纯的 JSP 技术, 比如仅用 JSP 就可以

很容易的实现一个功能齐全的购物车，但是如果要想完成大型的商业应用程序，综合以下这几种相互补充的技术往往可以提高整个系统的效率和稳定性。

1. Java Servlet

JSP 技术的发展壮大了 Java Servlet 技术，事实上，JSP 引擎在运行 JSP 时会把 JSP 页面翻译成 Servlet，因此 Servlet 在网络编程世界也是非常流行的，因为它在理论上和实践上都是可以完全取代 CGI 脚本的，Servlet 也能产生动态的网页，它是通过把静态的 HTML 与数据库查询或事务性服务提供的内容混合起来实现的，但 Servlet 程序员在输出动态数据的同时也需要输出大量静态的 HTML 代码，这是一件相当烦琐而且极易出错的工作。JSP 则是通过在 HTML 页面中内嵌 Java 代码这条途径来实现生成动态网页的目的，程序员只需要集中于自己的逻辑处理，而不必理会那些烦琐的 HTML 代码，这样就极大的增加了基于 Java 的网络体系结构的灵活性。

从另一个角度来讲，在开发一个大型的网络应用程序时，前台的静态页面和后台的处理逻辑往往会由不同的人来书写，我们希望尽量将他们的工作分离开，每个人只需要专注于自己擅长的部分，那么就得将事务处理从 JSP 页面中独立出来，这时 Servlet 便显示出优势，它由专门的程序员来开发，而网页设计师只需要调用它们，做一些简单的输入输出操作就可以了，完全不用理会其中复杂的处理逻辑。同时，Servlet 在执行时比 JSP 代码少了一道解释的工序，安全性和稳定性也更好一些。所以在逻辑处理比较复杂的情况下，选择这种分离的模式会更适合一点。

2. Java Bean

将 JSP 和 Java Bean 组件技术结合起来也是一种有效的开发模式，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 标识或者小脚本来生成页面上的动态内容，而这些生成内容的逻辑就可以被封装在 Java Bean 组件中，并且捆绑在小脚本里。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，而不影响内容的生成。在服务器端，JSP 引擎解释 JSP 标识和小脚本，生成所请求的内容（例如，通过访问 Java Bean 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。

大多数 JSP 页面都依赖于可重用的、跨平台的组件（Java Bean 或者 Enterprise Java Bean 组件）来执行应用程序所要求的更为复杂的处理。开发人员可以重用现有的组件来完成相似的功能，这种方法大大加速了大型系统的开发。

3. 标识库

标识库是另一项与 JSP 密切相关的技术。Web 页面开发人员不会都是熟悉脚本语言的编程人员。Java Server Pages 技术封装了许多功能，它们通过不同的标识来实现。标准的 JSP 标识能够访问和实例化 Java Bean 组件、设置或者检索组件属性、下载 Applet 等等。

通过开发定制化的标识库，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识库，这使得 Web 页面开发人员能够使用熟悉的工具与与标识相似的执行特定功能的构件来工作。

总之, JSP 技术很容易整合到多种应用体系结构中, 作为 J2EE (Java 2 企业版体系结构) 的一个组成部分, JSP 技术能够支持高度复杂的基于 Web 的应用。随着越来越多的供应商将 JSP 支持添加到他们的产品中, 相信 JSP 将会成为企业级应用的标准规范之一。

1.2 JSP 与相关的 Java 技术

在大多数的应用中, JSP 通常不能单独完成所有工作, 它需要相关的 Java 技术作为支持才能完成整个系统的构建, 在这一节里就将介绍这些与 JSP 相关的 Java 技术。

1.2.1 Java Servlet

Servlet 在国内的翻译方式很多, 本书将直接采用 Servlet 这个名称, 而不做任何翻译。Servlet 和传统的 CGI 程序和 ISAPI、NSAPI 等 Web 程序开发工具的作用是基本相同的, 在使用 Java Servlet 以后, 用户不必再使用效率低下的 CGI 方式, 也不必使用只能在某个固定 Web 服务器平台运行 API 方式来动态生成 Web 页面。许多 Web 服务器都支持 Servlet, 即使不直接支持 Servlet 的 Web 服务器也可以通过附加的应用服务器和模块来支持它。得益于 Java 跨平台的特性, Servlet 也是平台无关的, 实际上, 只要符合 Java Servlet 规范, Servlet 是完全平台无关且是与 Web 服务器无关的。由于 Java Servlet 内部是以线程方式提供服务, 不必对于每个请求都启动一个进程, 并且利用多线程机制可以同时为多个请求服务, 因此 Java Servlet 效率非常高。

但 Java Servlet 也不是没有缺点, 和传统的 CGI、ISAPI、NSAPI 方式相同, Java Servlet 也是利用输出 HTML 语句来实现动态网页的, 如果单纯用 Java Servlet 来开发整个网站, 动态部分和静态页面的整合过程简直就是一场恶梦。这就是 Sun 公司推出 Java Server Pages 的原因。

Java Servlet 是 JSP 技术的基础, 在大型的 Web 应用程序开发中, 通常需要 Java Servlet 和 JSP 配合才能完成, 这里只是简单介绍了 Servlet 的相关知识, 关于 Servlet 的开发将在第五章讲述。

1.2.2 Java Bean

Java Bean 是 Java 的可重用组件技术。ASP 通过 COM 来扩充复杂的功能, 如文件上载、发送 Email 以及将业务处理或复杂计算分离出来成为独立可重复利用的模块。JSP 通过 Java Bean 实现了同样的功能扩充。JSP 对于在 Web 应用中集成 Java Bean 组件提供了完善的支持。这种支持不仅能缩短开发时间 (可以直接利用现有成熟的组件, 避免了重复开发), 也为 JSP 应用带来了更多的可伸缩性。Java Bean 组件可以用来执行复杂的计算任务, 或负责与数据库交互等。

在实际的 JSP 开发过程中, 读者将会发现, 和传统的 ASP 或 PHP 页面相比, JSP 页面是非常简洁的, 由于 Java Bean 开发起来简单, 又可以利用 Java 语言的强大功能, 许多动态页面处理过程实际上被封装到了 Java Bean 中。

1.2.3 JDBC 数据库连接技术

JDBC 是用于执行 SQL 语句的 Java 应用程序接口, 由一组用 Java 语言编写的类与接口组成, 在 JSP 中将使用 JDBC 来访问数据库。JDBC 是一种规范, 它让各数据库厂商为 Java 程序员提供标准的数据库访问类和接口, 这样就使得开发独立于 DBMS 的 Java 应用程序成为可能。一般的 Java 开发工具都带有 JDBC-ODBC 桥驱动程序, 这样, 只要是能够使用 ODBC 访问的数据库系统, 也就能够使用 JDBC 访问了。

本书将在第七章详细介绍 JDBC 技术。

1.3 JSP 与其他 CGI 语言的比较

在互联网技术迅猛发展的今天, 流行的开发 Web 应用程序的方法大致有: ASP、PHP、CGI 和 JSP。其中 CGI 是早期的 Web 应用开发方法, 开发难度较大, 同时不易于扩展, 逐渐为 ASP、PHP 和 JSP 所取代。其中 JSP 技术迅速成为后起之秀, 在 Web 开发领域独领风骚。JSP 基于强大的 Java 语言, 并与 J2EE 紧密结合, 可快速创建高性能的 Web 应用程序。它将内容的生成和显示进行分离, 使 Web 开发人员能专注于应用逻辑的处理; 它强调利用可重用的组件 (Java Bean、EJB), 加速总体开发速度, 提高系统质量; 它采用标识简化页面编程, 降低开发人员的工作难度。

JSP 与 ASP 和 PHP 相比, 主要优势在于以下几点。

1. 平台和服务器的独立性

JSP 可以运行在任何符合 Java 语法结构的环境中, 取代了过去依附于单一平台或开发商的 Web 开发技术, JSP 能够运行在大多数的 Web 服务器上, 并且支持来自多家开发商提供的各种各样的工具包。

而 ASP 是基于 ActiveX 控件技术的, 因此它基本上是局限于微软的操作系统平台之上。ASP 的主要工作环境是微软的 IIS 服务器, 又因 ActiveX 对象具有平台特性, 所以 ASP 技术不能在跨平台的 Web 服务器上工作。

2. 开放的开发过程, 开放的源代码

自 1995 年以来, Sun 公司开始使用开放过程的方法同国际 Java 组织合作开发和修改了 Java 技术的规范。最近, Sun 公司将最新版本的 JSP 和 Java Servlet (JSP 1.1, Java Servlet 2.2) 的源代码发放给 Apache, 以求 JSP 与 Apache 能够紧密的相互发展。Apache, Sun 和许多其他的公司及个人公开成立一个咨询机构以便任何公司和个人都能免费取得信息 (详见: <http://jakarta.apache.org>)。

JSP 毫无疑问已经取得成功, 并将随 Java 组织不断开放扩大继续完善。相反, ASP 技术仅依靠微软本身的推动, 其发展是建立在独占的、封闭的开发过程基础之上。

3. 开发难度上的优势

ASP 和 JSP 技术都能使开发者实现通过点击网页中的组件制作交互式的、动态的内容。ASP 仅支持组件对象模型 COM, 而 JSP 技术提供的组件都是基于 Java Bean 技术或 JSP 标签库。由此可以看出两者虽有相同之处, 但其区别是很明显的。

(1) JSP 标签的可扩充性

尽管 ASP 和 JSP 都使用标签与脚本技术来制作动态 Web 网页, JSP 技术能够使开发者扩展 JSP 标签得以应用, JSP 开发者能定制标签库, 所以网页制作者充分利用与 XML 兼容的标签技术强大的功能, 大大减少对脚本语言的依赖。由于定制标签技术, 使网页制作者降低了制作网页和向多个网页扩充关键功能的复杂程度。

(2) JSP 跨平台的可重用性

JSP 的开发人员在开发过程中一直关注可重用性。JSP 组件 (Enterprise Java Bean, Java Bean, 或定制的 JSP 标签) 都是跨平台可重用的。企业级 Java Bean 组件可以访问传统的数据库, 并能以分布式系统模式工作于 Unix 和 Windows 平台。JSP 技术的标签可扩充功能为开发人员提供了简便的、与 XML 兼容的接口以及共享网页的打包功能, 使其完全的工业标准化。

这种基于组件的模式有效提高了应用程序的开发效率, 因为这种模式能够使开发人员利用快捷的子组件快速创建模板应用程序, 然后再整合一些附加功能以后便可使用。像这样有效的方法在 JSP 中无处不在, 并可将其打包成一个 Java Bean 或一个工业标准化的组件。

4. JSP 依托了功能强大的 Java 语言

JSP 技术是用 Java 语言作为脚本语言的, 而 ASP 网页使用微软的 VBScript 或 JScript。Java 是成熟的、强大的、易扩充的编程语言, 远优于基于 BASIC 的脚本语言。JSP 利用 Java 技术并且被编译为 Java Servlet, JSP 网页为整个服务器端的 Java 库单元提供了一个接口来服务于 HTTP 的应用程序。

Java 使开发人员的工作在其他方面也变得一样容易、简单。例如, Java 语言通过提供防止内存的泄漏的方法, 在内存管理方面也能大显身手。加之, JSP 为应用提供了健壮异常事件处理机制。

(1) 易于维护

基于 JSP 技术的应用程序比基于 ASP 的应用程序易于维护和管理。

脚本语言都能很好服务于小的应用程序, 但不能适应大型的、复杂的应用程序。因为, Java 是结构化的, 它比较容易创建和维护庞大的、组件化的应用程序。

JSP 突出的组件技术实现了处理逻辑和数据输出的分离。企业级的 Java Bean 结构整合了企业逻辑, 例如数据库的访问、安全及事务完整性。

因为 JSP 技术是一种开放的, 跨平台的结构, 因此, Web 服务器、平台及其他的组件能很容易升级或切换, 不会影响 JSP 基本的应用程序。这一特点使 JSP 能够适应各种 Web 应用程序不断的变化和发展。

(2) 企业级产品的多样性

J2EE 是适用于企业级应用程序的 Java 结构, 作为 J2EE 的一部分, JSP 可访问所有 J2EE 的组件, 包括 Java Bean, 企业级 Java Bean 及 Java Servlet。JSP 网页都能完全编译成为 Servlet, 所以它具有强大的功能和更好的灵活性。

1.4 JSP 的简单描述

JSP 能将网页的动态部分与静态部分有效分开, JSP 看上去与普通的 HTML 代码十分

相似。实际上，只要用任一网页制作工具，编写规则的 HTML，然后通过专门的标签将动态部分包含进来就可以了。标签绝大部分是以<%开始，以%>结束的。

下面是一个 JSP 片段：

Hello Mr.

```
<I><%= request.getParameter("name") %></I>
```

网页的显示情况是"Hello Mr. Robinson"。

文件的扩展名通常是.jsp，并且可以放在普通静态网页存放的目录下。虽然写的更像静态网页而不是 Servlet，但其实质上还是要转化成 Servlet，即静态网页的数据输出和 Servlet 服务。这个转换是在最开始调用时完成的，需要注意的是，许多 Web 服务器支持匿名服务，因此，表面上看是调用 HTML 文件，实际上则可能是 Servlet 或者 JSP。

标准 JSP 支持以下几种标签：

- (1) 编译器指引 <%@ 编译器指引 %>
- (2) 预定义 <%! 预定义 %>
- (3) 运算式 <%= 运算式 %>
- (4) 程序代码 <% 程序代码 %>
- (5) 注解 <%-- 注解 --%>

下面来看一个简单的 JSP 页面，从下面的代码清单中可以看到，JSP 页面除了比普通 HTML 页面多一些 Java 代码外，两者具有基本相同的结构。Java 代码是通过<% 和 %>符号加入到 HTML 代码中间的，它的主要功能是生成并显示一个从 0 到 9 的字符串。在这个字符串的前面和后面都是一些通过 HTML 代码输出的文本。

```
<HTML>
<HEAD>< TITLE>JSP 页面 </TITLE></HEAD>
<BODY>
<%@ page language="Java" %>
<%! String str="0"; %>
<% for (int i=1; i < 10; i++) {
    str = str + i;
} %>
JSP 输出之前。
<P>
<%= str %>
<P>
JSP 输出之后。
</BODY>
</HTML>
```

这个 JSP 页面可以分成几个部分来分析。

首先是 JSP 指令。它描述的是页面的基本信息，如所使用的语言、是否维持会话状态、是否使用缓冲等。JSP 指令由<%@ 开始，以%> 结束。在本例中，指令“<%@ page language="Java" %>”只简单地定义了本例使用的是 Java 语言（当前，在 JSP 规范中 Java

是唯一被支持的语言)。

接下来的是 JSP 声明。JSP 声明可以看成是定义类这一层次的变量和方法的地方。JSP 声明由 `< %!` 开始，以 `%>` 结束。如本例中的 “`< %! String str="0"; %>`” 定义了一个字符串变量。在每一项声明的后面都必须有一个分号，就像在普通 Java 类中声明成员变量一样。

位于 `<%` 和 `%>` 之间的代码块是描述 JSP 页面处理逻辑的 Java 代码，如本例中的 for 循环所示。

最后，位于 `< %=` 和 `%>` 之间的代码称为 JSP 表达式，如本例中的 “`< %= str %>`” 所示。JSP 表达式提供了一种将 JSP 生成的数值嵌入 HTML 页面的简单方法。

除了 HTML 标准语法，JSP 以下面三种方式嵌入到页面中：脚本元素、指令语法、动作语法。脚本元素中包含标准的 Java 代码，指令语法控制 Servlet 的整个结构，而动作语法可以规范要用到的元件（如：Bean），另外还能控制 JSP 引擎的行为。

1.5 JSP 技术的未来

JSP 技术被设计为一个开放的、可扩展的建立动态 Web 页面的标准。开发人员可以使用 JSP 页面来创建可移植的 Web 应用，在不同的 Web 和应用服务器上为不同的场合运行。

通过与业界的广泛合作，Sun 公司保证了 JSP 的规范的开放性和可移植性。JSP 可以使用任何客户机和服务器平台，在任何地方编写和部署它们。将来，开发工具供应商和其他厂商将通过为专门的功能提供客户化的标识库而扩展平台的功能。

JSP 目前的版本通过一个核心标识集、隐含对象以及开始创建动态 Web 页面所需的基本功能构成了开发的基础。越来越多的厂商都在开发支持这一标准的工具集合以及运行平台。

在今后的版本中，JSP 将给予 XML、客户化标识更多的支持，同时提高与 J2EE 的集成度，并且供应商们可能会有选择地扩展和扩充在 JSP 规范中基本的、必需的功能。我们有理由相信 JSP 将成为诸如电子商务、分布计算等网络计算领域的统一标准。

第 2 章 JSP 环境的安装与部署

JSP 是基于 Java Servlet 的技术, 是 Servlet2.1 API 的扩展, 因此支持 Servlet 的新版本平台都应该支持 JSP。要学习 Servlet 和 JSP 开发必须准备一个符合 Java Servlet 2.1/2.2 和 Java Server Pages1.0/1.1 规范的开发环境。

首先需要安装一个支持 Java Servlet 的 Web 服务器, 或者在现有的 Web 服务器上安装支持 Servlet 的软件包, 如果使用的是最新的 Web 服务器或应用服务器, 那么很可能它已经有了所有的支持软件。

下面是目前支持 Java Servlet 的 Web 服务器或应用服务器的最新列表 (表 2-1)。

其中 Server 表示服务器, Add-on Engine 表示附加引擎, Server and Add-on Engine 则表示兼有服务器和附加引擎两者的性质。

表 2-1 支持 Java Servlet 的服务器

产 品	产品类型	支持 Servlet 的版本	支持 JSP 的版本
Apache Tomcat	Server and Add-on Engine		
	3.2.1	2.2	1.1
	4.0 beta	2.3 PFD2	1.2 PFD2
Apache Web Server Jserv	Server and Add-on Engine	2.0	No
ATG Dynamo Application Server	Server	2.2	1.1
BEA Weblogic Server	Server	2.2	1.1
Borland AppServer	Server	2.2	1.1
Brokat Server Technologies	Server	2.2	1.1
Caucho Technology Resin	Add-on Engine	2.2	1.1
EasyThings Web Server	Server	2.2	1.1
ExOffice Intalio	Server	2.2	1.1
Gefion Software WAICoolRunner	Add-on Engine	2.2	1.1
Gefion Software Lite WebServer	Server	2.2	1.1
HP Bluestone Total-e-Server	Server	2.2	1.1
IBM WebSphere Application Server 4.0	Server	2.2	1.1
IONA iPortal Application Server	Server	2.2	1.1
iPlanet Application Server 6.0	Server	2.2	1.1