

高丛书

宝贝工作室 编著

步步

动态网站开发

# JSP 动态网站开发 实用教程



机械工业出版社  
China Machine Press

**动态网站开发步步高丛书**

# **JSP 动态网站开发实用教程**

宝贝工作室 编著



**机械工业出版社**

JSP 是现在动态网站开发技术中最新也是最为强大的技术，它是基于 Java 的一种脚本开发语言，在国外一经推出即受到了广泛的欢迎。现在在硅谷，JSP+EJB 的开发模式已经成了电子商务网站开发事实上的标准。

虽然 JSP 非常强大，而且已经比较流行，可是在国内，程序员对于 JSP 的理解和认识还只是停留在了解的阶段，很少有网站开发者采用 JSP 来架构自己的动态网站。为了给众多想要采用 JSP 进行动态网站开发的用户提供一个翔实有效的参考资料，我们特编写了此书。

本书主要围绕 JSP 最重要的知识点进行了详细的讲解，无论是入门级的开发者还是经验丰富的程序员，相信都能够从中受益。另外，本书还提供了大量的实例详解，便于读者迅速地掌握 JSP 开发的精髓。

MSB/05

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：于 宁 封面设计：姚 毅

责任印制：路 琳

北京市密云县印刷厂印刷·新华书店北京发行所发行

2001 年 7 月第 1 版第 1 次印刷

787mm×1092mm 1/16·17 印张·418 千字

0 001—5000 册

定价：32.00 元 （1CD，含配套书）

ISBN 7-900066-58-6/TP·55

凡购本书，如有缺页、倒页、脱页，由本社发行部调换  
本社购书热线电话（010）68993821、68326677-2527

# 前 言

随着信息时代的到来, 计算机技术的发展日新月异, 互联网由小到大, 在中国越来越普及。利用互联网特有的跨地域性、低成本、高效率等优势开展电子商务也成为时代发展的必然趋势。随之而来的, 是互联网网络技术, 尤其是基于 Web 的 B/S 系统的设计开发技术也发展得非常迅速。仅仅三、四年时间, 动态网站开发技术就从最早的 CGI(Common GateWay Interface) 技术到微软公司的 ISAPI 以及 Netscape 公司的 NSAPI 技术, 再到这两年的 ASP 或者 PHP, 动态网站开发技术也经历了从复杂到简单、高效, 功能也越来越强大这样一个过程。但是, 现有的动态网站开发技术往往都在某些方面有不少的缺点, 例如: ASP 的跨平台性太差, 离开了微软的东西就毫无用武之地; PHP 运行效率虽然比较高, 设计开发数据库程序非常快, 但其不支持分布式应用的致命弱点导致它只能在中小型网站发挥作用。

1998 年, 在基于 Servlet 技术的基础上, Sun 公司发布了 JSP 0.90, 它比较好地解决了这些问题。首先, JSP 是基于 Java 技术基础的, 继承了 Java 的一整套优点, 例如, 跨平台性、分布式等等。Sun 公司随之推出的用于企业级分布式运算的 J2EE 规范也因其巨大的优点得到市场的承认, 越来越多的公司推出了或准备推出支持 J2EE 规范的各种产品, 越来越多的网站改用 JSP 作为开发手段, JSP 也得到了越来越多的软件的支持, 成为实际上的电子商务系统的开发规范。

为了提高国内 JSP 技术的应用普及程度, 我们特编写了这本书! 本书分为 10 章, 系统地、循序渐进地讲述了如何利用 JSP 技术设计一个交互性的动态网站, 并结合实例详细讲解了一些实际运用的技巧。

第 1、2 章, 对比了常用几种动态网站设计技术的优缺点, 同时搭建起我们自己的开发平台。

第 3 章, 因为 JSP 使用 Java 语言进行编程, 我们讲解了 Java 的语法规则, 重点阐述了面向对象的 Java 程序设计思想, 并附以实例加以说明, 以帮助读者建立起面向对象编程的概念。

第 4、5、6 章讲述了 JSP 的常用语法以及常用的内置对象的使用方法, 并重点讲解用 JSP+JavaBean 技术进行多层应用体系的开发。

第 7 章以 JDBC 为核心重点阐述使用 JSP 开发 Web 数据库系统的方法与常用技巧, 并附以实例说明 JDBC-ODBC bridge、JDBC Driver 等各种驱动形式从配置到使用的方法。

第 8 章, 我们围绕着 JSP 中经常出现的非常典型的问题作为专题进行分析、解决。

第 9 章, 为了更好地结合应用, 我们由浅入深地讲述了常见的动态网站应用项目的 JSP 实现, 并加以分析、引导。

第 10 章, 我们将对 JSP 目前最热的发展方向——在 J2EE 中的应用进行讲述, 同时一步步讲解使用 JBuilder 开发 J2EE 解决方案的过程。

本书由陈伟执笔, 宝贝工作室全体成员为这本书倾注了大量的心血。虽然我们倾心而为, 但是书中错误之处在所难免, 恳请读者们指正。

编 者

# 目 录

## 前言

## 第 1 章 动态网站技术漫谈 ..... 1

### 1.1 动态网站设计技术概述 ..... 1

### 1.2 常见的动态网站设计技术 ..... 2

#### 1.2.1 古典的 CGI ..... 2

#### 1.2.2 强大的 ASP ..... 3

#### 1.2.3 灵巧的 PHP ..... 4

#### 1.2.4 Java Servlet ..... 5

#### 1.2.5 未来之星——JSP ..... 6

### 1.3 JSP 与 ASP 技术的比较 ..... 7

#### 1.3.1 理论体系比较 ..... 7

#### 1.3.2 运行速度比较 ..... 7

#### 1.3.3 运行平台比较 ..... 8

#### 1.3.4 开发效率比较 ..... 8

#### 1.3.5 应用前景 ..... 8

## 第 2 章 学习 JSP 前的准备 ..... 10

### 2.1 JSP 概述 ..... 10

#### 2.1.1 JSP 的发展历史 ..... 10

#### 2.1.2 JSP 与 Servlet 的关系和区别 ..... 10

#### 2.1.3 JSP 与 Java 的关系 ..... 11

#### 2.1.4 JSP 的执行过程 ..... 11

### 2.2 开发平台的选择与建立 ..... 12

#### 2.2.1 平台概述 ..... 12

#### 2.2.2 典型平台的配置 ..... 12

#### 2.2.3 优缺点比较 ..... 24

### 2.3 商业运行平台的选择与建立 ..... 24

#### 2.3.1 平台概述 ..... 24

#### 2.3.2 典型平台的配置 ..... 25

#### 2.3.3 优缺点比较 ..... 35

### 2.4 开发工具介绍 ..... 35

#### 2.4.1 Visual Age ..... 35

#### 2.4.2 JBuilder ..... 36

#### 2.4.3 JDeveloper ..... 38

#### 2.4.4 Visual Cafe ..... 39

## 第 3 章 Java 入门 ..... 40

### 3.1 Java 语言概述 ..... 40

#### 3.1.1 Java 的发展历程 ..... 40

#### 3.1.2 Java 的含义 ..... 40

#### 3.1.3 Java 的优点 ..... 42

#### 3.1.4 Java 相关技术名词介绍 ..... 43

### 3.2 Java 语言介绍 ..... 46

#### 3.2.1 第一个 Java 程序 ..... 46

#### 3.2.2 Java 概述 ..... 47

#### 3.2.3 数据类型 ..... 48

#### 3.2.4 标识符、常量、变量 ..... 49

#### 3.2.5 运算符、表达式 ..... 51

#### 3.2.6 数组的使用 ..... 53

#### 3.2.7 数据类型转换 ..... 55

#### 3.2.8 流程控制 ..... 57

#### 3.2.9 异常处理 ..... 61

### 3.3 Java 的 OO 思想介绍 ..... 68

#### 3.3.1 面向对象的思想介绍 ..... 68

#### 3.3.2 类 ..... 68

#### 3.3.3 定义及使用类 ..... 69

#### 3.3.4 继承 ..... 74

#### 3.3.5 接口 ..... 78

#### 3.3.6 包 ..... 80

## 第 4 章 JSP 语法介绍 ..... 82

### 4.1 第一个 JSP 文件 ..... 82

### 4.2 注释 ..... 83

### 4.3 声明 ..... 86

### 4.4 表达式 ..... 86

### 4.5 脚本代码 ..... 88

|                         |     |                             |     |
|-------------------------|-----|-----------------------------|-----|
| 4.6 包含指令                | 89  | 7.2.2 连接数据库                 | 126 |
| 4.7 页面指令                | 90  | 7.2.3 创建 Statement 对象       | 126 |
| 4.8 自定义指令               | 92  | 7.2.4 使用 SQL 语句访问数据库        | 126 |
| 4.9 JSP 程序示例            | 93  | 7.2.5 对记录集的处理               | 127 |
| 第5章 JSP 中内置对象的使用        | 95  | 7.3 利用 JDBC 开发 Web 数据库程     | 128 |
| 5.1 Request 对象          | 95  | 7.3.1 利用 JDBC-ODBC 桥访问数据库   | 128 |
| 5.2 Response 对象         | 96  | 7.3.2 利用专用 JDBC 访问数据库系统     | 132 |
| 5.3 Out 对象              | 99  | 7.4 利用 JSP + JavaBean 访问数据库 | 135 |
| 5.4 Session 对象          | 103 | 7.4.1 开发数据库访问通用的 JavaBean   | 135 |
| 5.5 Application 对象      | 106 | 7.4.2 利用 JavaBean 访问数据库     | 137 |
| 5.6 Exception 对象        | 108 | 第8章 JSP 开发技巧集               | 139 |
| 第6章 利用 JavaBean 增强 JSP  | 110 | 8.1 常见类编译错误                 | 139 |
| 6.1 什么是 JavaBean        | 110 | 8.2 jar 的用法                 | 140 |
| 6.2 JavaBean 的优点        | 110 | 8.3 JSP 常见问题及技巧             | 143 |
| 6.3 引用 JavaBean         | 111 | 8.4 利用连接池使用数据库              | 145 |
| 6.3.1 引用 JavaBean       | 111 | 8.5 JSP 中文问题                | 150 |
| 6.3.2 设置 JavaBean 的属性   | 111 | 8.5.1 问题分类                  | 150 |
| 6.3.3 使用 JavaBean 的方法   | 112 | 8.5.2 原因分析                  | 153 |
| 6.4 开发 JavaBean         | 112 | 8.5.3 解决方法                  | 156 |
| 6.4.1 开发 JavaBean       | 112 | 8.6 数据库翻页程序的设计              | 157 |
| 6.4.2 配置 JavaBean       | 113 | 8.7 JSP 表单交互程序设计            | 166 |
| 6.5 JavaBean 应用举例       | 114 | 第9章 JSP 应用举例                | 173 |
| 6.5.1 税率计算              | 114 | 9.1 计数器                     | 173 |
| 6.5.2 猜数字游戏             | 116 | 9.2 留言版的制作                  | 174 |
| 6.5.3 ZipBean 的使用       | 120 | 9.3 在线投票系统                  | 180 |
| 第7章 使用 JSP 开发 Web 数据库程  |     | 9.4 利用 JSP 在线发送电子邮件         | 185 |
| 序                       | 122 | 9.5 使用 JSP 上传文件             | 190 |
| 7.1 JDBC 接口技术           | 122 | 9.6 论坛系统                    | 196 |
| 7.1.1 JDBC 简介           | 122 | 9.7 客户端信息的获取                | 218 |
| 7.1.2 JDBC 的分类          | 122 | 9.8 在线购物系统                  | 226 |
| 7.1.3 JDBC API          | 123 | 第10章 JSP 在 J2EE 中的应用        | 232 |
| 7.1.4 JDBC-ODBC Bridge  | 124 | 10.1 J2EE 技术                | 232 |
| 7.1.5 JDBC 的优缺点         | 124 | 10.2 JSP 与 XML              | 235 |
| 7.1.6 JDBC 驱动的安装        | 125 | 10.3 Enterprise JavaBean    | 245 |
| 7.2 利用 JDBC 开发 Web 数据库概 | 125 | 10.4 JSP 在 J2EE 中的重要意义      | 260 |
| 7.2.1 加载驱动程序            | 126 | 附录 本书光盘使用说明                 | 263 |

# 第 1 章 动态网站技术漫谈

## 1.1 动态网站设计技术概述

随着 Internet 的迅速发展, Web 服务因为其优秀的易用性而成为 Internet 所提供的最重要的功能之一。在早些时候, 用户使用简单的超文本标记语言 (HTML 语言) 来设计网页, 这些简单的 HTML 文档在被设计完成之后保存在 Web 服务器上, 任何连入互联网的用户都可以访问这些页面从而获取网站的信息。然而, 网页设计完成后, 它所提供的信息就不会变化, 人们称这种形式的页面为静态页面。

随着技术的进步, Internet 的进一步普及, 广大互联网用户对 Web 站点的需求越来越多样化, 传统的网页设计模式——静态页面设计已经远远不能满足用户的要求。这时, 用户希望自己的 Web 站点能够根据用户的客户端输入作出不同的反应, 能够与用户进行动态交互。为了解决这个问题, 将传统的编程技术引入互联网络与 Web 技术相结合, 通过在传统的静态页面中加入各种程序和逻辑控制等手段, 从而能够实现用户与服务器之间动态和个性化的交流与互动。我们将这种可以变化的、区别于以前的静态页面的页面称为动态页面, 设计动态页面的技术称为动态网站设计技术。

### 1. 动态网站的功能特点

(1) 支持后台数据库 可以通过简单的动态 Web 程序访问服务端后台数据库, 可动态地将数据库中的信息发布到 Web 网站中去。

(2) 动态、可控的内容显示 所有的静态页面都是事先编写好的、不变的, 因此访问同一页面的用户看到的都将只是相同的内容, 而动态网站可以根据不同的用户做不同的页面显示, 或者对同一个用户的不同阶段做不同的页面显示。

(3) 强大的可扩充性 传统的 Web 网站只能对用户显示静态的页面内容, 而动态网站可以利用一定的程序接口实现各种各样的扩充功能, 例如实现 Internet 提供的 E-mail、Chat 和 FTP 等功能。各种实际应用功能都可以动态网站的形式实现, 无缝地结合到 Web 系统上来。

### 2. Script 脚本语言

传统的 HTML 都是静态的, 为实现网页的动态功能, 通常在 HTML 代码中混合某种程序代码 (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), 由专门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。因此客户端既无法看到所编写的脚本语言代码, 用户的浏览器也不需要任何附加的软件支持就可实现。脚本的解释过程见图 1-1。

目前, 最常用的三种动态网站开发脚本语言有 ASP (Active Server Pages), JSP (Java Server Pages) 和 PHP (Personal Home Page Tools)。

ASP 全名 Active Server Pages, 是微软公司推出的动态网站设计技术。利用它可以产生和运行动态的、交互的且高性能的 Web 服务应用程序。ASP 采用脚本语言 VB Script 或者 Jscript 作为自己的开发语言。

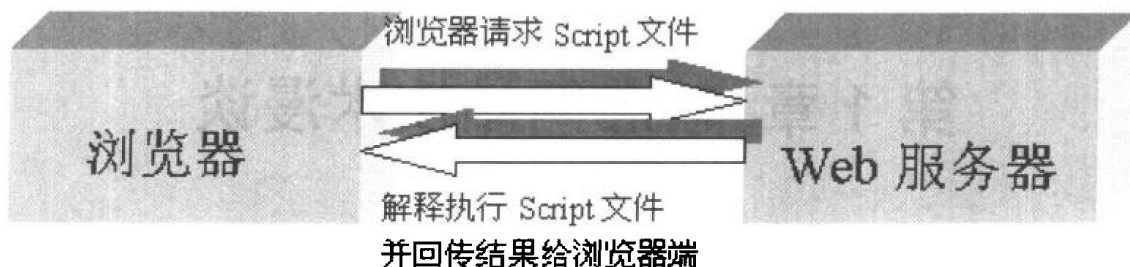


图 1-1 Script 的解释过程

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 等好的语法框架，有以上编程经验的开发人员可以快速地掌握、投入实际应用，并耦合 PHP 自己的特性，使 Web 开发者能够快速地写出动态页面。PHP 支持目前绝大多数数据库，并且免费让所有人使用和更改解释器源代码，读者可以从 PHP 官方站点(<http://www.php.net>)自由下载。

JSP 是 Sun 公司推出的新一代站点开发语言，它与目前的 ASP、PHP 等脚本语言显著不同的地方就是 JSP 在服务端是编译后执行的，而不是解释执行。JSP 采用的脚本语言就是 Sun 公司的 Java 语言，JSP 可以在 Servlet 和 JavaBean 的支持下，完成功能强大的站点程序。

在下面的章节中，将对其进行进一步的讲解与比较。

## 1.2 常见的动态网站设计技术

### 1.2.1 古典的 CGI

CGI (Common Gateway Interface) —— 通用网关接口，它是一个 WWW 服务器主机对外服务的标准接口。一般来说，一个 CGI 接口的功能就是在超文本文件和服务器主机应用程序间传递信息，简单地说，CGI 就是存在于 WWW 服务器上的可执行的程序代码，而它的工作就是根据客户的动作、要求产生并传回生成的静态文件。

任何一种程序语言，只要能在服务器主机上利用 CGI 接口来编写应用程序，都可以叫做 CGI 程序语言。目前最为流行的 CGI 程序语言为 Perl 语言，但使用其他的高级程序设计语言也是可以的，例如：VB、C 和 C++ 等。CGI 运行速度快，并且用户可以用任何一种自己熟悉的高级语言来编写 CGI 程序，具有良好的灵活性和可移植性，开发效率也很高。

但是由于 CGI 程序一般以单独程序的形式存在于 Web 服务器上，对每个 CGI 请求，Web 服务器都将重新启动一个新的进程，所以当多个用户并发地访问该 CGI 程序时，Web 服务器的性能下降得非常迅速，严重影响服务器的正常运行。在轻负载的情况下，CGI 的性能与其他方法相比并没有太大差别，但在重负载的情况下，使用 CGI 的 Web 服务器性能远比其他方法要差得多。

下面是用 Perl 语言写的一个“Hello World!”的简单 CGI 示例。

例 1.1 (Hello.cgi):

```
#!/usr/bin/perl  
print "Hello, World";
```

### 1.2.2 强大的 ASP

ASP (Active Server Pages) ——动态服务器页面，它是微软公司推出的为实现基于 Web 应用的动态交互网页制作的一种技术解决方案，一种可以用来取代传统 CGI 的动态、服务器端运行的页面设计技术。从表现形式上来说，ASP 就是一段运行在服务器端的脚本程序代码。利用 ASP 技术，用户可以创建和运行各种动态的交互式 Web 服务器应用程序，如交互式的动态网页，包括使用 HTML 表单收集和处理信息，上传与下载等等。

那么，和传统的 CGI 技术进行比较，ASP 又具有哪些特点呢？

首先，ASP 非常简单、易学，开发效率很高。因为 ASP 是使用 VBScript 等简单易懂的 Script 脚本语言编写，而 VBScript 是 Microsoft Visual Basic 语言的一个子集，同时在其基础之上进行了简化，它采用与 Visual Basic 近似的语法框架，非常简单。这样，对大量的 VB 程序员来说，可以非常容易地掌握 VBScript，并利用其在 Web 网站上快速地开发应用程序。虽然 ASP 默认的脚本编写语言为 VBScript，但是用户也可以根据自己的需要进行选择。只要用户安装了相应的脚本引擎，就几乎可以使用目前所有流行的脚本语言编写 ASP 应用程序。

ASP 的一个最为显著的优势就是对组件和对象技术的充分支持。通过使用 ASP 的组件和对象技术，用户就可以随心所欲地使用对象，例如：调用对象方法，设置对象属性等。此外，用户既可以直接在 ASP 页面中使用 Visual Basic 和 Visual C++ 各种功能强大的 COM 对象，同时还可以创建自己的 COM 对象，直接在 ASP 页面中使用。

因此，通过使用自己的或者第三方开发的各种 COM 对象，可以大大节省开发人员编写代码的数量。举例来说，ASP 提供了一些内置对象，利用这些 ASP 中提供的内置对象，你可以使自己的脚本更加强大；例如，利用 request 对象，您可以从浏览器中接受和发送信息，并可以接收用户在 HTML 表单中的信息，并在一个 ASP 文件中来响应它，而这一过程的实现只需要 2~3 行的代码，非常简单。

ASP 使用的 COM 对象技术基于开放设计环境，用户利用 VB、VC 等工具可以自己定义和制作各种功能的组件，对于 ASP 来说，只需要简单地调用就可以了。这样，动态网页就具有了几乎无限的扩充能力，可以完成任何应用程序所能完成的功能。

当然，有其利必有其弊。ASP 也具有非常明显的缺点——跨平台性非常差。目前情况下，ASP 的运用仅限于 Microsoft Windows 系统之下。换句话说，假设你的操作系统不是 Windows 系列，或者 Web 服务器不是 IIS、PWS，那就不能使用 ASP 了。

总之，由于 ASP 利用了许多流行、高效的技术，如 VBScript、ODBC、COM 等，是一种简单、易学、易用且功能非常强大的动态网站设计技术，同时，由于 Microsoft 公司的关系，从开始推广之初，微软就将 ASP 作为其 Web 服务器产品 IIS 的内置组件捆绑发售，从而为用户直接使用 ASP 带来了极大的方便。因此，ASP 出现后就占据了动态网站设计技术的主流地位，应用得非常广泛。

下面是一个“Hello World!”程序的 ASP 示例。

例 1.2 (Hello.asp)：

```
<html>
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Hello World</title>
</head>
<body>
<%
Response.write "Hello World!"
%>
</body>
</html>
```

### 1.2.3 灵巧的 PHP

PHP 从 1994 年秋天开始孕育, 它的创始人是 Rasmus Lerdorf。早期没有发布的版本被他用在自己的网页上跟踪有谁来参观过他的在线个人简历。被其他人使用的第一个版本是在 1995 年发布的, 当时叫做 Personal Home Page Tools。它包含了一个非常简单的语法分析引擎, 只能理解一些指定的宏和一些 Home Page 后台的常见功能, 如留言本、计数器和一些其他的素材。在 1995 年中期, PHP 开发小组重写了这个语法分析引擎并且命名为 PHP/FI 2.0 版本。PHP/FI 以惊人的速度发展, 并且其他的人也开始对他的源码加以改进和修改。1997 年, PHP 开发小组在原 PHP/FI2.0 的基础上, 开发出了全新的 PHP3, 并将其置于 GNU 的规约之下(和 Linux 相同), 免费让所有人使用和更改解释器源代码。1999 年, 发布了 PHP4.0 版, 其中 Zend 优化器的加入使其实现编译执行, 运行速度大大提高。

相对于传统的 CGI 语言, PHP 拥有下列优越的特性(以 PHP3 为例说明):

(1) 简单、易于使用 PHP3 的语法结构大部分借用了 C、JAVA 和 Perl 等好的语法框架与语言风格, 有以上编程经验的开发人员可以快速地掌握、投入实际使用。在 PHP3 的程序包中, 有极其详细的安装说明, 任何人都可以按照说明快速地配置好自己的 Web 服务器, 甚至可以在 PHP 的站点上直接下载已经配置好 PHP3 模块的 Web 服务器。同时, 利用已有的编程知识, 也很容易迅速地掌握 PHP 程序的开发。

(2) 运行速度快 PHP3 和 ASP 类似, 也采用在 HTML 文件中嵌入专用的 Script 代码的形式编写, 但 PHP 的解释程序本身作为 Web 服务器的一个模块运行, 这在很大程度上提高了运行时的解析速度。

(3) 优秀的跨平台性 PHP 具有良好的跨平台性, 非常容易进行移植。同样的 PHP 代码可以在 Windows、Unix 和 Linux 等各种系统下的 Web 服务器上正常运行, 支持包括 IIS、Apache 等在内的多种流行的 Web 服务器。在不同的平台上移植 PHP 程序, 不需要更改任何 PHP 代码, 就可以直接在新的平台上使用。

(4) 数据库功能强大 PHP3 支持的数据库极其广泛, 可直接为很多数据库提供连接, 包括 Oracle、Sybase、Postgres、Mysql、Informix、Dbase、Solid 和 Access 等。PHP3 还完全支持 ODBC 接口, 凡是支持 ODBC 接口的数据库, PHP3 都可顺利地对其进行操作。

另外, PHP 提供了丰富的操作函数, 用于数据库的连接以及记录的查询、插入和修改等操作。利用 PHP 编写数据库操作程序非常简单、清晰, 而且对这些数据库的操作都是 PHP3 利用本身自带的库函数完成的, 无需其他附件介入, 因此在 PHP 的实际应用中, 对数据库的访问性能非常高。

(5) 丰富的网络函数 PHP 还提供了丰富的网络函数簇, 利用这些函数, PHP 能轻松地实现与其他协议的交互, 像 IMAP、SNMP、NNTP、POP3 和 HTTP 这些协议等, 开发出功能十分强大的网络程序。

(6) 功能易于扩展 PHP3 不但内置了对文件上传、身份认证、邮件收发和动态图像生成等功能的支持, 还极有远见地提供了对动态 PDF 文档生成、XML 的直接支持。

PHP 是很容易扩展的, 用户可以编写自己的扩展模块。用户可利用自己编写的 DLL 模块, 扩展自己定义的功能。PHP 是大量程序员协同开发的 (就如同 Linux), 所以若有可能, 一旦您编写的功能提交给 PHP 组织被获准, 这些功能将成为 PHP 标准功能的一部分。PHP 的扩充是无限的。

(7) 完全免费 PHP3 是遵循 GNU 规约的, 任何人均可按规约免费使用并进行源码改写, 使用者还可通过 PHP3 的站点、邮件列表等方式获得支持。PHP 和 Linux、Perl 一样, 是一种自由软件, 这使得 PHP 更加灵活, 人们可以不断为 PHP 增加新的功能, 从而使 PHP 永远走在同类技术的前端。

PHP 是由爱好者们自由开发的, 这也就决定了它是免费的、灵活的, 但它同样也具有很大的缺点, 最大的缺点就是不支持未来企业计算的三层结构模型, 缺乏整体架构能力和分布式特性。因此, 它只能作为中、小型网站的开发工具。

下面是一个 “Hello World!” 程序的 PHP 示例。

例 1.3 (Hello.php):

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Hello World</title>
</head>
<body>
<?php echo "Hello World!"?>
</body>
</html>
```

#### 1.2.4 Java Servlet

大家都知道 Java Applet, 那什么是 Servlet 呢?

实际上, Servlet 是用 Java 编写的可以动态地扩展服务器能力的一种 Server 端程序。Servlet 运行于支持 Java 的 Web Server 中, 其主要功能是交互式地浏览和修改数据, 从而生成动态

Web 内容。

Servlet 可以像 CGI、ASP 和 PHP 等脚本一样扩展 Web 服务器功能，但是 Servlet 在 Web 服务器上运行的是编译后的 Java 字节码，只占用很少密集资源，速度也很快，而不像 ASP、PHP 那样是解释执行的。同时由于 Servlet 是用 Java 编写的，所以 Servlet 也是跨平台的。

下面是一个“Hello World!”程序的 Servlet 示例。

例 1.4 (Hello.java):

```
public void service(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    ServletOutputStream out=response.getOutputStream();
    response.setContentType("text/html");
    out.println("Hello, World!");
}
```

### 1.2.5 未来之星——JSP

JSP (Java Server Pages) 是 Sun 公司在 1998 年推出的基于 Java Servlet 以及整个 Java 体系的 Web 开发技术，是 Servlet 2.1 API 的扩展。简单地说，JSP 是一种以 Java 为主的跨平台 Web 开发语言，利用这一技术可以建立先进、安全和跨平台的动态网站。

从表现形式上来说，JSP 与微软的 ASP 相似，也是在 HTML 网页中包含了能够生成动态内容的脚本代码，不同的是这个脚本代码是 Java 而不是 VBScript。在用户第一次访问这个页面时，这个 JSP 的页面将会被编译成 Servlet 在 Web 服务器上执行。

其实，在运行时，JSP 是以 Servlet 的形式运行的。但与 Servlet 不同的是，JSP 最大的优点是可以把用户界面设计从系统内容中分离开来，使得设计人员能够在不改变底层动态内容的前提下改变整个网页布局。在 J2EE (Java2 Enterprise Edition) 分布式应用体系结构中，JSP 是多层 (Multi-tier) 体系中表示层 (Presentation Logic Layer) 的表现方式。

如同 ASP 使用 COM 组件实现功能的扩展一样，JSP 也可使用 Java 编写的可重用的、跨平台的组件——JavaBean 或者 EJB (Enterprise JavaBean) 来执行应用程序所要求的更为复杂的处理，而开发 JavaBean 与 EJB 则比开发 COM 组件要容易得多。

下面是一个“Hello World!”程序的 JSP 示例。

例 1.5 (Hello.jsp):

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<%
out.print("Hello World!");
%>
```

```
</body>
```

```
</html>
```

## 1.3 JSP 与 ASP 技术的比较

在以上介绍的几种动态网站设计技术中, CGI 因为其本身的缺点——多进程运行、重负载情况下效率很低而逐渐遭到淘汰, 而 PHP 虽然在数据库设计方面较为方便, 且效率很高, 得到了很大的发展, 但因其技术体系的限制使其不符合分布式应用体系结构, 无法将表示层 (Presentation Logic Layer) 与业务层 (Business Layer) 分离, 不适合开发大型的企业级应用。

JSP 技术与 ASP 技术因其强大的技术支持而成为目前主流的动态网站设计技术, 在市场上占据了越来越大的份额, 那它们二者之间又存在什么异同, 各有什么优缺点呢?

下面就从多个方面对其进行分析比较。

### 1.3.1 理论体系比较

JSP 与 ASP 从完成功能上来讲是相近的, 但它们分属于不同的分布式应用体系。

从分布式应用系统的角度来看, 一个网络项目最少分三层: 核心层 (Data Layer), 业务层 (Business Layer), 表示层 (Presentation Layer), 或者更多层。

Microsoft 推出的 Windows 分布式的互联网应用体系结构 (Windows DNA) 是一个建立现代化的、多层次的分布式计算解决方案的体系结构, 它可以通过任何网络进行传输。Windows DNA 提供了一个集成客户/服务器和基于 Web 应用开发的统一体系。而在 Windows DNA 体系中, 用于解决表示层 (Presentation Layer) 的技术之一就是 ASP 技术。可以认为, Windows DNA 技术的核心就是 ASP+COM/DCOM 技术。

在 Sun 公司推出的 J2EE (Java2 Enterprise Edition) 分布式企业计算体系中, 利用 JavaBean、EJB 技术编写业务层 (Business Layer) 的功能是非常强大的, 但是对于写表示层 (Presentation Layer) 就很不方便, 由 Servlet 发展而来的 JSP 就主要是为了方便书写表示层 (Presentation Layer) 而设计的。通过 JSP 调用 JavaBean, 实现两层的整合, 达到分布式应用。

### 1.3.2 运行速度比较

ASP 的编程语言是 VBScript 之类的脚本语言, JSP 使用的是 Java, 这是两者最明显的区别。此外, ASP 与 JSP 还有一个更为本质的区别: 两种语言引擎用完全不同的方式处理页面中嵌入的程序代码。在 ASP 下, VBScript 代码被 ASP 引擎解释执行; 在 JSP 下, 代码被编译成 Servlet 并由 Java 虚拟机执行, 这种编译操作仅在对 JSP 页面的第一次请求时发生。

用 ASP 与 JSP 分别做循环性能测试及存取 Oracle 数据库性能测试的实验, 在循环性能测试中, JSP 令人吃惊的只用了 4 秒钟就结束了  $20000 \times 20000$  的循环。而 ASP 测试  $20000 \times 20000$  的循环时无法结束, 陷于停顿状态, 最后只好采用  $2000 \times 2000$  循环 (少一个数量级), 用去了 63 秒的时间。在数据库测试中, 两者分别对 Oracle8 进行 1000 次 Insert、Update、Select 和 Delete 的操作, 结果是 JSP 需要 13 秒, ASP 则需要 73 秒。

可以看出在同等条件下, JSP 的性能远远优于 ASP。JSP 在执行以前先被编译成字节码 (Byte Code), 字节码由 Java 虚拟机 (Java Virtual Machine) 解释执行, 比源码解释的效率

高；服务器上还有字节码的 Cache 机制，能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢，因为它存在编译为字节码的过程，以后访问就会快许多。

### 1.3.3 运行平台比较

在目前的情况下，支持 ASP 的 Web 服务器很有限，只有微软的 IIS 或 PWS（个人 Web 服务器），只能运行在 Windows 的平台上。而 JSP 可以运行在 IIS、Apache 等任何 Web 服务器上，可以运行在包括 Windows、Unix、Linux 和 Solaris 的等绝大部分系统平台上。可见 JSP 的运行平台更加广泛，跨平台性、可移植性也更好。

### 1.3.4 开发效率比较

ASP 与 JSP 都为嵌入式脚本语言，开发效率都比较高，但 JSP 技术的一些特点使得开发更加迅速、快捷。

（1）内容的生成和显示分离 因为在 JSP 页面中，生成内容的逻辑封装在处于业务层的 JavaBean 或 EJB 中，然后通过嵌入页面的脚本代码生成具体的内容，具体显示则是由页面文件（可能是 HTML，也可能是 XML）负责完成的。网页内容的生成和显示是分离的，这就意味着 Web 设计人员可以方便地设计页面，而不影响内容的生成，而程序设计者只需要修改相应的业务逻辑，而不用管显示的形式。这样，对于一个大型的分布式应用系统来说，非常有利于协作开发。

（2）可复用的组件 JSP 系统开发时是采用表示层与业务层分离的形式进行的，业务逻辑都封装在可跨平台使用的 JavaBean 或 EJB 组件中，由于 Java 良好的可移植性，开发人员可以相互利用常用的组件，提高开发效率。

（3）JSP 可定制标签库 用户使用一些 HTML 的标签和嵌入的脚本来进行动态 Web 网站的开发。JSP 的开发者能够定制自己的标签库（Tag Lib），使得 Web 页面设计人员能够非常形象地利用开发者所设计的构件，而不需要懂关于程序的知识。

（4）JSP 的组件使用更方便 ASP 通过 COM 来扩充复杂的功能，如文件上传、发送 E-mail 以及将业务处理或者复杂计算分离出来成为独立可重复利用的模块。JSP 通过 JavaBean 实现了同样的功能扩充。

在开发方面，COM 的开发远比 JavaBean 复杂和繁琐，学会开发 COM 必须掌握烦琐的 API 库。而 JavaBean 就简单多了，会 Java 语言的人即可迅速掌握 JavaBean 的开发。

在组件的发布方式上，COM 必须在服务器上注册，如果修改了 COM 程序，就必须重新注册，甚至必须关机和重新启动。JavaBean 则不需要注册，放在 ClassPath 包含的目录中就行了。如果 JavaBean 进行了修改，则 JSWDK 和 Tomcat 现在还需要关闭和重新运行（但不是关机），但开发者已经许诺将在以后的版本中做到不需要关闭服务器。

另外，JavaBean 是完全的面向对象开发，可以针对不同的业务处理功能方便地建立一整套可重复利用的对象库，例如用户权限控制、E-mail 自动回复等等，代码复用程度很高。

### 1.3.5 应用前景

目前来说一般的脚本语言都能很好服务于小型的应用，而不能适应大型的、复杂的应用。由于 Java 是结构化、面向对象的，因此 JSP 相对容易创建和维护庞大的、组件化应用程序。

JSP 突出的组件技术使修改内容而不影响逻辑或修改逻辑而不影响内容变得很容易实现。企业级的 JavaBeans 结构整合了企业逻辑，例如数据库的访问，安全，事务完整性与独立性，做到真正的分布式应用。

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，且不会影响 JSP 基本的应用程序。这一特点使得 JSP 能够适用各种 Web 应用程序不断的变化和发展。

Java2 平台即企业版（J2EE）是适用于多企业应用程序的 Java 结构，作为 J2EE 的部分，JSP 网页可访问所有 J2EE 的组件，包括 JavaBean，企业 JavaBean（EJB）及 Java Servlets。

通过 J2EE，JSP 网页能够用许多方式同企业系统交互访问。J2EE 支持两种 CORBA 规范的技术：Java IDL 和 RMI-IIOP。因此，在 EJB 技术支持下，JSP 网页可通过运用高级的、对象映射的方式访问数据库。

因为 JSP 技术是基于 Java 的开放性过程的产品，所以它能够广泛支持不同提供商提供的工具、Web 服务器和应用程序的服务，这样能够使用户选择最佳的开发方法，选择最适合他们的应用程序开发的工具包，同时，有效地保护用户在代码和人员培训上的投资。

## 第 2 章 学习 JSP 前的准备

### 2.1 JSP 概述

#### 2.1.1 JSP 的发展历史

JSP (Java Server Pages) 是 Sun 公司在 1998 年推出的基于 Java Servlet 以及整个 Java 体系的 Web 开发技术, 是 Servlet 2.1 API 的扩展。1998 年, Sun 公司发布了 JSP 0.90 标准; 1999 年 1 月, 发布了 JSP 0.92 标准, 同时推出支持 JSP 的 Web 服务器 Java Web Server2, JSP 迅速流行; 1999 年 11 月, Sun 发布 JSP 1.1 标准, 同时推出 JSWDK1.0.1 与 Servlet2.2 版。2000 年 1 月, Apache 开发组织的 Jakarta 项目小组发布了支持 Servlet2.2 和 JSP 1.1 的开发软件包——Tomcat。利用这个软件, Web 开发者与设计者能在多层结构的企业级应用系统中更加方便、快速地开发动态页面与程序。2000 年 9 月, Sun 发布 JSP1.2 草案和 Servlet2.3 版。

简单地说, JSP 是一种以 Java 为基础的跨平台 Web 开发脚本语言, 利用这一技术可以建立先进、安全和跨平台的动态网站。

#### 2.1.2 JSP 与 Servlet 的关系和区别

自 Java 诞生以来, 因为 Java 语言本身的优点, 利用 Java 语言来开发 Server 端的应用程式是非常方便的。1997 年, Sun 公司提出了 Java Servlet 技术, Servlet 可以用来扩充 Web Server 的基本功能, 最常使用的就是增加 Web Server 的互动性, 它提供了强有力且效率较高的技术来取代传统的 CGI (Common Gateway Interface) 的功能 (然而, Servlet 不仅仅是为 HTTP 设计的, 实际上 Servlet 就是运行在网络服务器上的一种特定的应用程序, 它也可以根据使用者的要求设计自己的协议; Servlet 也不仅仅只用在 Web Server 上)。

Servlet 实际上就是一个基于 Java 技术的运行在服务器端的程序 (与 Applet 不同, 后者运行在浏览器端)。开发人员能够编写出这样的 Servlet 程序, 以接收来自 Web 浏览器的 HTTP 请求, 动态地生成相应响应 (可能要查询数据库来完成这项请求), 然后发送 HTML 或 XML 文档的响应到浏览器。

和传统的 Server 端扩展机制, 例如: CGI 相比, Servlet 有如下优点:

- 1) CGI 脚本快, 因为 Servlet 采用了不同的处理模式。
- 2) Servlet 使用的 API 标准为大多数 Web Server 所接受。

3) 因为是 Java 语言开发的, 所以 Servlet 拥有 Java 的所有优点, 包括易于开发和与平台无关等。

但采用 Servlet 开发 Web 系统, 整个业务处理逻辑及页面表示都必须在 Servlet 中实现, 程序设计开发人员或者页面开发人员修改了文件, 就不得不重新编译该 Servlet。这样, 对于页面开发人员与程序设计人员来说, 不利于分工协作, 设计与更新基于 Servlet 的 Web 应用程序将成为一件非常棘手的事情。

因此, 针对 Servlet 所具有的缺点, 需要一个业界范围新的创建动态页面的解决方案。这个新方案将解决当前方案所受到的限制, 具有如下的特点:

- 1) 能够在任何 Web 或应用程序服务器上运行。
- 2) 将应用程序逻辑和页面显示分离。
- 3) 能够快速地进行开发和测试。

4) 简化开发基于 Web 的交互式应用程序的过程。Sun 公司在 Servlet 技术的基础上开发出 JSP 技术来满足这样的要求。JSP 规范是 Web 服务器、应用服务器、交易系统以及开发工具供应商间广泛合作的结果。这个规范来整合和平衡已经存在的对 Java 编程环境(例如, Java Servlet 和 JavaBeans™)进行支持的技术和工具, 其结果是产生了一种新的、开发基于 Web 应用程序的方法, 给予使用基于组件应用逻辑的页面设计者以强大的功能。

JSP 是 Servlet API 的一个扩展。事实上, JSP 网页被编译成 Servlet 运行, 所以它们也具有 Servlet 的所有优势, 包括访问 Java API。与 Servlet 相比, JSP 有以下优势:

1) JSP 能使业务逻辑和页面显示分离。JSP 可以结合 JavaBean 技术实现表示层(Presentation Layer)与业务层(Business Layer)的分离, 利于实现企业级的分布式应用。同时调用 JavaBean 组件也非常简单, JSP 标签可以完全管理这些组件, 避免网页设计人员复杂化应用程序。

2) 开发人员可以提供定制化的 JSP 标签库。由于 JSP 网页能够根据需要自动进行编译, Web 设计人员无需重新编译应用程序逻辑就可以改变表述代码。这也使得 JSP 与 Servlet 相比成为一种可以更灵活生成动态 Web 内容的方法。对于开发人员来说, 页面设计人员不必知道 Java 编程语言或任何 Servlet 代码, 他们能够集中精力去编写 HTML 代码, 达到更好的表示效果; 编程人员就可以集中精力去建立对象和应用程序, 实现更强大的功能。

因此, 可以认为 JSP 是 Servlet 的一种技术扩展, JSP 继承了 Servlet 的所有优点(因为它本身就是以 Servlet 的形式运行的), 同时解决了 Servlet 设计表示层的种种问题, 简化 Web 系统的页面开发。

### 2.1.3 JSP 与 Java 的关系

对于 JSP 与 Java 的关系, 可以从以下两个角度来理解。

1) JSP 文件实际上就是包含了用 Java 语言编写的脚本代码的 HTML 文件, 是由 Java 代码、HTML 代码以及一些分隔符标签混合而成的。类似于 ASP 是利用 VBScript 作为其脚本语言一样, Java 语言是 JSP 所采用的进行程序设计的脚本语言。

2) 另一方面, JSP 文件实际上是被编译成 Servlet 在 Web 服务器上执行的。JSP 技术是 Servlet 技术的扩展, 而 Servlet 实际上是一种特殊的运行在服务器端的 Java 程序。所以 JSP 也就是用 Java 语言编写的一种特殊的应用程序。

### 2.1.4 JSP 的执行过程

在某个 JSP 文件第一次获得请求时, 不管请求来自于客户端浏览器还是服务器上的 Servlet, JSP 文件首先将被 JSP 引擎(JSP engine)转换成为一个 Servlet。而这个引擎本身也是一个 Servlet。

JSP 文件在编译过程中如果被发现有语法错误, 转换过程将中断, 并向客户端发出