

JSP 程序设计精彩实例



程序设计精彩实例

姜晓铭 刘 波 张亦华 编著

<http://www.juke.com>

<http://www/>

<http://www/>



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

JSP 程序设计精彩实例

姜晓铭 刘 波 张亦华 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Java Server Pages (简称 JSP, 服务器端动态网页) 是由 Sun 微软系统公司 (Sun Microsystems Inc) 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中有其强大而特别的功能。

本书以精炼的语言、清晰的结构, 以及标准的图像对 JSP 进行了全面的介绍。本书内容分为两篇。第一篇讲解了 JSP 程序设计的必备知识; 第二篇以精彩的实例展现了 JSP 在服务器端动态网页开发中的强大功能。本书涉及的实例主要有: 访问计数器、意见箱、在线投票系统、小游戏的制作、用户管理、在线书店、BBS 论坛和聊天室等。

本书内容由浅入深, 既可作为初学者的学习资料, 也可作为中高级编程人员的参考用书。

版权所有, 翻印必究。

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 程序设计精彩实例

作 者: 姜晓铭 刘波 张亦华 编著

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 欧振旭

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印 张:** 21.25 **字 数:** 487 千字

版 次: 2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

书 号: ISBN 7-302-04430-9/TP·2605

印 数: 0001~5000

定 价: 28.00 元

前 言

目前在国外的众多网站，特别是涉及电子商务的网站中，已经大量使用了 JSP 技术。在万维网短暂的历史中，它已经从一个大部分显示静态信息的网络演化到进行股票交易和进行购书操作的动态信息的网络。在 Sun 正式发布 JSP 之后，这种新的 Web 应用开发技术很快引起了人们的关注。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 的说法，JSP 能够适应市场上包括 Apache WebServer、IIS 4.0 在内的 85% 的服务器产品。在 JSP 以前，由于 ASP 的简单易用，许多用户对 ASP 情有独钟。即使是这样，普遍的看法认为，关注 JSP 的发展仍旧很有必要。

编 者

2001 年 3 月

目 录

第一篇 JSP 程序设计必备知识

第 1 章 JSP 入门.....	1
1.1 JSP 初步.....	1
1.1.1 Web 应用程序的发展历史	1
1.1.2 JSP 的运作模式和 JSP 的优点.....	3
1.2 JSP 基本指令.....	5
1.2.1 HTML 注释.....	5
1.2.2 隐藏注释	5
1.2.3 HTML 中嵌入 JSP 脚本的格式	5
1.2.4 声明.....	6
1.2.5 表达式	6
1.2.6 Include 指令	7
1.2.7 Page 指令	7
1.2.8 <jsp:useBean>	10
1.2.9 <jsp:getProperty>.....	12
1.2.10 <jsp:setProperty>	12
1.2.11 Scriptlet.....	15
1.2.12 Taglib 指令	15
1.2.13 <jsp:plugin>	16
1.2.14 <jsp:forward>	19
1.2.15 <applet>.....	20
第 2 章 初试 JSP 程序设计.....	21
2.1 简单的 JSP 网页制作	21
2.2 JSP 与 Java Servlet.....	25
2.2.1 实例一	25
2.2.2 实例二	27
2.3 JSP 与 JavaBeans.....	36
2.3.1 实例一	37
2.3.2 实例二	40

2.4 JSP 与 Applets	43
2.4.1 实例一	44
2.4.2 实例二	47

第二篇 JSP 程序设计实例

第3章 访问计数器	55
3.1 文件构架	55
3.2 设计前的思考	56
3.3 文字模式的访问计数器	56
3.3.1 读取写入文件	57
3.3.2 文字模式的访问计数器	57
3.3.3 用 JavaBeans 模块化的文字模式访问计数器	62
3.4 图形模式的访问计数器	66
第4章 在线投票系统	72
4.1 文件构架	72
4.2 设计前的思考	73
4.3 在线投票系统	74
4.3.1 建立投票表单	74
4.3.2 投票后台处理	78
4.3.3 显示投票结果	83
第5章 意见箱	88
5.1 文件架构	88
5.2 意见箱表单	89
5.2.1 显示欢迎信息	89
5.2.2 表单的客户端确认	90
5.2.3 程序 index.jsp	92
5.3 保存意见箱内容	96
5.3.1 JavaBeans 支持	96
5.3.2 意见的保存	98
5.3.3 查看意见箱	103
5.4 删除操作和发送操作	105
5.4.1 删除操作	105

5.4.2 发送操作	106
第 6 章 小游戏的制作	109
6.1 文件构架	109
6.2 连城游戏	110
6.2.1 连城游戏制作	110
6.2.2 在浏览器中游戏	117
6.3 猜数字	119
6.3.1 猜数字的制作	119
6.3.2 在浏览器中游戏	122
第 7 章 用户管理	125
7.1 文件架构	125
7.2 连接数据库	126
7.2.1 建立数据库	126
7.2.2 建立数据源	128
7.2.3 调用数据库	130
7.3 用户注册	132
7.3.1 用户注册申请表单	132
7.3.2 确认表单	134
7.3.3 添加到数据库	139
7.4 登录与注销	141
7.4.1 用户登录	141
7.4.2 用户注销	144
7.5 修改用户信息	145
7.5.1 用户信息修改表	145
7.5.2 保存到数据库	149
第 8 章 在线书店	152
8.1 文件架构	152
8.2 数据库准备	153
8.2.1 book 数据表	153
8.2.2 bookOrder 数据表	154
8.2.3 detail 数据表	155
8.3 数据查询	156
8.3.1 在线书店的入口	156
8.3.2 初始化分页信息	158
8.3.3 实现分页	161

8.3.4 查看详细资料.....	168
8.4 购物车.....	170
8.4.1 程序准备.....	171
8.4.2 添加到购物车.....	171
8.4.3 查看购物车.....	174
8.4.4 修改和清空购物车.....	178
8.5 结账.....	179
8.5.1 用户表单.....	180
8.5.2 保存订单.....	185
第9章 BBS 论坛.....	188
9.1 文件架构.....	188
9.2 数据库准备.....	189
9.2.1 bbsUser 数据表.....	189
9.2.2 content 数据表.....	190
9.2.3 board 数据表.....	191
9.2.4 denyId 数据表.....	191
9.3 讨论区.....	192
9.3.1 网站的入口.....	192
9.3.2 全部讨论区.....	197
9.3.3 讨论区文章列表.....	201
9.4 文章管理.....	207
9.4.1 查看文章.....	207
9.4.2 发表文章.....	210
9.4.3 删除文章.....	217
9.5 用户操作.....	218
9.5.1 查看用户列表.....	218
9.5.2 查看用户资料.....	221
9.5.3 查封用户.....	223
第10章 聊天室.....	229
10.1 文件构架.....	229
10.2 数据库的准备.....	230
10.2.1 chatUser 数据表.....	231
10.2.2 chat 数据表.....	231
10.2.3 userlist 数据表.....	232
10.3 聊天室普通程序部分.....	233

10.3.1 用户登录	233
10.3.2 登录界面的后台处理	236
10.3.3 聊天室主界面	244
10.3.4 更改房间	247
10.3.5 发送聊天信息	249
10.3.6 显示聊天信息	257
10.3.7 显示在线用户列表	262
10.3.8 离开聊天室	265
10.4 管理员程序部分	267
10.4.1 登录界面	267
10.4.2 管理聊天室	269
10.4.3 “踢人”出局	274
附录 A Java 语言语法	277
A.1 基本概念	277
A.1.1 标识符与关键字	277
A.1.2 基本数据类型	278
A.1.3 变量与常量	279
A.2 数组与流程控制	280
A.2.1 数组	281
A.2.2 流程控制	284
A.3 类与继承	289
A.3.1 类	289
A.3.2 类的继承	301
A.4 包、接口和异常	304
A.4.1 包	304
A.4.2 接口的介绍	307
A.4.3 异常	310
附录 B SQL 语言	313
B.1 数据定义与数据控制	313
B.1.1 SQL 的数据定义	313
B.1.2 SQL 的数据控制	315
B.2 数据查询语句 (SELECT)	316
B.2.1 SELECT 语句的一般格式	316
B.2.2 SELECT 语句的几种特殊格式	318
B.2.3 筛选数据记录的 WHERE 语句	321

B.3 数据操作语句	324
B.3.1 Delete 命令: 删除数据	325
B.3.2 Update 命令: 更新数据	325
B.3.3 Insert Into 命令: 增加数据	326
B.3.4 Select Into 命令: 建立新数据表	327

第一篇 JSP 程序设计必备知识

第 1 章 JSP 入门

主要内容：

- Web 应用程序的发展历史
- JSP 的运作模式
- JSP 的优点
- JSP 的基本语法

本章将向读者简要介绍 JSP 的一些初步知识，以及 JSP 的基本指令和 Java 语言的一些基本语法。

1.1 JSP 初步

JSP 是由 Sun 微软系统公司（Sun Microsystems Inc）倡导、许多公司参与一起建立的一种动态网页技术标准，其在动态网页的建设中有其强大而特别的功能。目前在国外的众多网站特别是涉及电子商务的网站中，已经大量使用了 JSP 技术。

JSP（Java Server Pages，服务器端动态网页）既然名为“服务器端”动态网页，可想而知，JSP 必定是在“服务器”建立的动态网页。更明确地说，JSP 是能在 Web Server（尤其是 JSWDK）端整合 Java 语言至 HTML 网页的环境中，利用 HTML 网页内含的 Java 程序代码取代原有的 CGI、ISAPI 或者 IDC 的程序，以便执行原有 CGI/WinCGI、ISAPI 的功能。

在进入 JSP 的殿堂之前，先纵观整个 Web Server 应用程序的发展历史，再认识 JSP 的开发环境以及运作模式。

1.1.1 Web 应用程序的发展历史

在万维网短暂的历史中，它已经从一个大部分显示静态信息的网络演化到进行股票

易和进行购书操作的一个动态信息的网络。在各种各样的应用程序中，对于可能使用的基于 Web 的客户端，看上去没有任何限制。

基于浏览器客户端的应用程序比传统的基于客户/服务器的应用程序的优势在于：几乎没有限制的客户端访问和极其简化的应用程序部署和管理（要更新一个应用程序，管理人员只需要更改一个基于服务器的程序，而不是成千上万的安装在客户端的应用程序）。这样，软件工业正迅速地向建造基于浏览器客户端的多层次应用程序迈进。

这些快速增长的、基于 Web 的精巧应用程序要求开发技术上的改进。静态 HTML 对于显示相对静态的内容是不错的选择，新的挑战在于创建交互的、基于 Web 的应用程序。在这些程序中，页面的内容是基于用户的请求或者系统的状态，而不是预先定义的文字。

对于这个问题的一个早期解决方案是使用 CGI-BIN 接口；开发人员编写与接口相关的单独程序，以及基于 Web 的应用程序，后者通过 Web 服务器来调用前者。这个方案有着严重的扩展性问题——每个新的 CGI 要求在服务器上新增一个进程。如果多个用户并发地访问该程序，这些进程将消耗该 Web 服务器所有的可用资源，并且系统性能降低到极其低下的地步。

某些 Web 服务器供应商已经尝试通过为他们的服务器提供“插件”和 API 来简化 Web 应用程序的开发。这些解决方案与特定的 Web 服务器相关，但却不能解决跨平台操作的问题。例如，微软的 ASP TM 技术使得在 Web 页面上创建动态内容更加容易，但是也只能工作在微软的 IIS 和 Personal Web Server 上。

当然，还存在其他的解决方案，但是它们都不能使一个普通的页面设计者能够轻易地掌握。例如，Java Servlet 技术就可以使得用 Java 语言编写交互的应用程序的服务器端的代码变得容易，一个 Java Servlet 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）。开发人员需要编写出这样的 Servlet，以接收来自 Web 浏览器的 HTTP 请求，动态地生成响应（可能要查询数据库来完成这项请求），然后发送包含 HTML 或 XML 文档的响应到浏览器。

采用这种方法，整个网页必须都在 Java Servlet 中制作。如果开发人员或者 Web 管理人员想要调整页面显示，他们就不得不编辑并重新编译该 Java Servlet，即使该 Java Servlet 在逻辑上已经能够运行。采用这种方法，生成带有动态内容的页面仍然需要应用程序的开发技巧。

很显然，目前所需要的是一个业界范围内的创建动态内容页面的解决方案。这个方案将解决当前方案所解决不了的问题，如：

- 能够在任何 Web 或应用程序服务器上运行。
- 将应用程序逻辑和页面显示分离。
- 能够快速地进行开发和测试。
- 简化开发基于 Web 的交互式应用程序的过程。

JSP 技术就是设计用来满足这些要求的解决方案。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。Sun Microsystems 开发出这个规范来整合及平衡已经存在的对 Java 编程环境（例如，Java Servlet 和 JavaBeansTM）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。

所谓的 JSP 网页 (*.jsp)，就是在传统的网页 HTML 文件 (*.htm,*.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标记 (tag) 而构成的。Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页，以及发送 E-mail 等，这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，对客户浏览器的要求最低，可以实现无 Plug-in、无 ActiveX、无 Java Applet，甚至无 Frame。JSP 在动态网页的建设中有其强大而特别的功能。

在 Sun 正式发布 JSP 之后，这种新的 Web 应用开发技术很快引起了人们的关注。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 的说法，JSP 能够适应市场上包括 Apache WebServer、IIS 4.0 在内的 85% 的服务器产品。在 JSP 以前，由于 ASP 的简单易用，许多用户对 ASP 情有独钟。即使是这样，普遍的看法认为，关注 JSP 的发展仍旧很有必要。

1.1.2 JSP 的运作模式和 JSP 的优点

JSP 能在 Web Server (尤其是 JSWDK) 端整合 Java 语言至 HTML 网页的环境中，然后利用 HTML 网页内含的 Java 程序代码取代原有的 CGI、ISAPI 或者 IDC 的程序，以便执行原有 CGI/WinCGI、ISAPI 的功能。

1. JSP 的运作模式

相对应于 Client 端 (指的是浏览器端的 HTML 文件) 内嵌的描述语言，Sun 公司提供的 JSWDK-1.0.1 也支持类似的描述语言，它便是 Java 语言。先来看看 JSP 的运作模式 (如图 1-1 所示)。

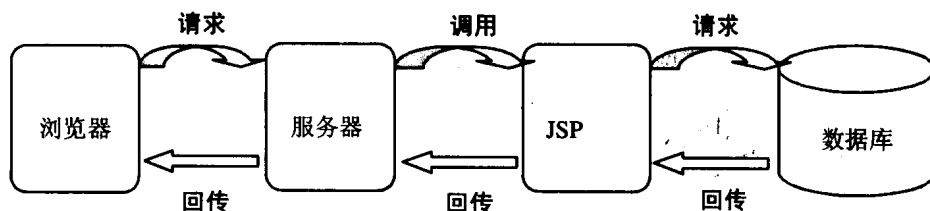


图 1-1 JSP 运作模式

由于 JSP 放置在 Web 服务器上，它在解析使用者由表单 (Form) 传送过来的字段数据后，接着通过适当的逻辑生成标准 HTML 文件，然后传给客户端，使用者看到的是一般符合 HTML 格式的文件内容。因为 JSP 是在 JSWDK-1.0.1 上执行的，所以无论使用者使用的是哪一种平台下的浏览器，皆能欣赏由 JSP 产生的网页内容。

2. JSP 的优点

JSP 与 ASP、PHP 相比有下列优点：

(1) 将内容的生成和显示进行分离:

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 标识或者小脚本来生成页面上的动态内容(内容是根据请求而变化的,例如请求账户信息或者特定的一瓶酒的价格),其生成内容的逻辑被封装在标识和 JavaBeans 组件中,并且捆绑在小脚本中,所有的脚本在服务器端运行。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 JavaBeans 组件、使用 JDBC 技术访问数据库等),并且将结果以 HTML(或者 XML)页面的形式发送回浏览器。这有助于作者既保护自己的代码,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 Enterprise JavaBeans 组件)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客户团体所使用。基于组件的方法加速了总体开发过程,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。

(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编程人员。Java Server Pages 技术封装了许多功能,这些功能是在生成易用的、与 JSP 相关的 XML 标识的动态内容时所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 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如同标识一样的工具来执行特定功能的构件。

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 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。

(4) 健壮性与安全性:

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。

(5) 良好的移植性:

作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,各处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中,可以使用自己所选择的服务器和工具,而且更改工具或服务器并不影响当前的应用。

(6) 企业级的扩展性和性能:

当与 Java 2 平台、企业版(J2EE)和 Enterprise JavaBeans 技术整合时,JSP 页面将

1.2 JSP 基本指令

服务器 JSP 页面中的程序代码的识别就是依靠一些基本指令完成的，所以下面将详细地介绍一下这些指令。这在下面的实例中都将用到，希望读者能够很好地掌握。

1.2.1 HTML 注释

HTML 注释的作用是产生一个注释发送到客户端。JSP 页面中的 HTML 注释跟其他 HTML 注释非常相似，它就是一段文本，并且在浏览器端用查看源代码功能可以看得到，也就是说，HTML 注释的内容是会传给浏览器的。

注释的 JSP 语法的基本格式如下：

```
<!-- 注释(comment) [<%= 表达式(expression)%>] -->
```

在这个格式中，是以<!--符号开始，而以-->符号结束。前面的注释部分是不会被服务器编译器所编译的，而后面的表达式部分将会被服务器编译器所编译，是真正意义上的“程序”。

1.2.2 隐藏注释

与上面的注释说明不同，隐藏注释是不会将其内容传给浏览器的。也就是说，隐藏注释说明是写在 JSP 程序中，但是不被传送到客户端，JSP 会忽略隐藏注释的文本行，所以隐藏注释通常用来注释不愿意被别人看到的内容。

JSP 不处理任何<%--和--%>之间的字符，这段注释也不会被发送到客户端，当然在客户端查看源代码也发现不了它。

在使用--%>结束注释之前可以使用任何符号，如果想在注释中用--%>，那么可以用--%>避开。

隐藏注释的语法格式如下：

```
<%-- 注释 (comment) --%>
```

在这个格式中，是以<%符号开始，以%>符号结束，它与注释语法是不同的，读者一定要将其区分开来。

1.2.3 HTML 中嵌入 JSP 脚本的格式

在 HTML 中嵌入 JSP 脚本，是实现页面动态化的基础，JSP 脚本才可以真正称得上是“程序”。

在 HTML 中可以嵌入 JSP 脚本语言, 使得网页具有动态的效果, 其基本格式如下:

`<%=代码段%>`

代码段可以包括任意长的代码, 声明任意多的变量和方法或者包括任意多的表达式。编写 JSP 程序最好的方法, 就是先处理好纯 HTML 格式的 Web 页面之后, 再将程序嵌入 Web 页面。这种方法, 可以让您在开发上达到事半功倍的效果。

1.2.4 声明

每一个程序语言都会出现变量和方法, JSP 程序也不例外。要让编译器知道这些变量和方法是合法的, 就必须依靠声明的作用。

声明指在 JSP 程序中声明合法的变量和方法。要在 JSP 程序使用变量和方法, 必须首先声明它, 不然将会出错, 可以一次性声明多个变量和方法, 只要以“;”结尾就行, 也可以一次只声明一个变量和方法, 但是这些声明在 Java 语言中必须是合法的。

当声明方法或变量时, 请注意以下的一些规则:

(1) 声明必须以“;”结尾 (Scriptlet 有同样的规则, 但是表达式就不同了)。

(2) 可以直接使用在 `<% @ page %>` 中包含进来的已经声明的变量和方法, 不需要对它们重新进行声明。

(3) 一个声明仅在一个页面中有效。

JSP 声明语法的基本格式如下:

`<%! 声明 1 (declaration1) ; [ 声明 2 (declaration2) ; ]+ ... %>`

这个声明是以符号 `<%!` 开始的, 以符号 `%>` 结束的。这又与前面的不同, 希望读者注意区分。

1.2.5 表达式

前面说过 JSP 是通过在 HTML 中加入 JSP 脚本可以实现动态页面的效果, 因此有了脚本语言就不能不介绍表达式。

表达式元素表示的是一个在脚本语言中被定义的表达式, 在运行后自动转化为字符串, 然后插入到这个表达式在 JSP 文件的显示位置。因为这个表达式的值已经转化为字符串, 所以你能在一行文本中插入这个表达式 (形式和 ASP 完全一样)。

当在 JSP 中使用表达式时请记住以下几点:

(1) 你不能用一个分号 (“;”) 来作为表达式的结束符, 但是同样的表达式用在 Scriptlet 中就需要以分号来结尾了。

(2) 这个表达式元素能够包括任何在 Java Language Specification 中有效的表达式。

(3) 有时候表达式也能作为其他 JSP 元素的属性值。一个表达式能够变得很复杂, 它可以由一个或多个表达式组成, 这些表达式的顺序是从左到右。

表达式的 JSP 语法的基本格式如下：

```
<%= 表达式 (expression) %>
```

这个格式对读者来说，应该不是陌生的，其实在第一节就已经接触过这种格式。这个格式是以符号`<%`开始，而以符号`%>`结束。

1.2.6 Include 指令

Include 指令的作用是在 JSP 中包含一个静态文件，同时解析这个文件中的 JSP 语句。

Include 指令将会在 JSP 编译时插入一个包含文本或代码的文件，当使用 **Include** 指令时，这个包含的过程就当是静态的。静态的包含就是指将插入这个被包含的文件到 JSP 文件中，这个包含的文件可以是 JSP 文件、HTML 文件或文本文件。如果包含的是 JSP 文件，将会执行这个包含的 JSP 的文件中的代码。

如果仅仅只是用 **Include** 指令来包含一个静态文件。那么这个包含的文件所执行的结果将会插入到 JSP 文件中放 **Include** 指令的地方。一旦执行了包含文件，那么主 JSP 文件的过程将会恢复，继续执行下一行。

这个被包含文件可以是 html 文件、jsp 文件、文本文件，或者只是一段 Java 代码，但是需要注意在这个包含文件中不能使用`<html>`、`</html>`、`<body>`或`</body>`标记，因为这将会影响在原 JSP 文件中同样的标记，这样做有时会导致错误。

Include 指令包含的文件的路径名一般来说是指相对路径，不需要什么端口、协议和域名，如果这个路径以“/”开头，那么这个路径主要是参照 JSP 应用的上下关系路径，如果路径是以文件名或目录名开头，那么这个路径就是正在使用的 JSP 文件的当前路径。

Include 指令的基本语法格式如下：

```
<%@ include file="relativeURL" %>
```

在这个格式中，是以符号`<%@`开始的，而以符号`%>`结束的。读者要特别注意。

1.2.7 Page 指令

Page 指令的作用是定义 JSP 文件中的全局属性。**Page** 指令作用于整个 JSP 页面，同样包括静态的包含文件。但是 **Page** 指令不能作用于动态的包含文件，比如`<jsp:include>`。

你可以在一个页面中用上多个 **Page** 指令，但是其中的属性只能用一次，不过也有个例外，那就是 **Import** 属性。因为 **Import** 属性和 Java 中的 **Import** 语句差不多（参照 Java 基本语法），所以你就能多用此属性几次了。

无论把 **Page** 指令放在 JSP 文件的哪个地方，它的作用范围都是整个 JSP 页面。不过，为了 JSP 程序的可读性，以及好的编程习惯，最好还是把它放在 JSP 文件的顶部。

下面在介绍 **Page** 的基本语法之前，先来看看 **Page** 指令的 11 个基本属性。