



计算机专业“十二五”规划教材

JSP程序设计教程

莫兵 蒙焕念 黄良永 主编



兵器工业出版社

内容简介

前言

计算机专业“十二五”规划教材

JSP 程序设计教程

莫兵 蒙焕念 黄良永 主编

兵器工业出版社

内容简介

JSP 程序设计技术是目前应用比较广泛的开发 Web 项目的三大动态网页技术之一,是一种动态网页技术标准,具有面向对象、跨平台等特性。尤其是它结合了 Servlet 与 JavaBean 技术,使得页面代码与后台业务逻辑代码分离,解决了过去 Web 开发技术存在的不足,提高了工作效率。全书共分 11 章,具体包括:Web 技术简介、Web 客户端技术、JSP 开发运行环境、JSP 基本语法、JSP 内置对象、数据库操作、JavaBean 组件、JSTL 标签库、Servlet 技术、JSP 开发模式、JSP 高级应用等。

本书适合作为应用型本科院校、高职高专计算机类专业“JSP 程序设计”课程的教材,也可以作为培训教材使用。

图书在版编目(CIP)数据

JSP 程序设计教程 / 莫兵, 蒙焕念, 黄良永主编. —

北京: 兵器工业出版社, 2015.7

ISBN 978-7-5181-0120-7

I. ①J… II. ①莫… ②蒙… ③黄… III. ①JAVA 语言—网页制作工具—高等学校—教材 IV. ①TP312
②TP393.092

中国版本图书馆 CIP 数据核字 (2015) 第 166115 号

出版发行: 兵器工业出版社

发行电话: 010-68962596, 68962591

邮 编: 100089

社 址: 北京市海淀区车道沟 10 号

经 销: 各地新华书店

印 刷: 冯兰庄兴源印刷厂

版 次: 2015 年 8 月第 1 版第 1 次印刷

印 数: 1-3000

责任编辑: 周 琦

封面设计: 赵俊红

责任校对: 郭 芳

责任印制: 王京华

开 本: 787×1092 1/16

印 张: 17.5

字 数: 420 千字

定 价: 38.00 元

(版权所有 翻印必究 印装有误 负责调换)

前言

当前主流的三大动态 Web 开发技术是：JSP、ASP.NET、PHP。其中，JSP（Java Server Page）程序设计技术是目前应用比较广泛的。JSP 是由 Sun Microsystems 公司倡导，多家公司一起参与制定的一种动态网页技术标准。JSP 继承了 Java 的面向对象、跨平台等特性，特别是结合 Servlet 与 JavaBean 技术，使得页面代码与后台业务逻辑代码分离，解决了过去 Web 开发技术存在的不足，提高了工作效率。

本书是笔者在总结多年行业软件开发经验和积极推行项目教学法的基础上编写而成的。以项目驱动教学，主要特色是以“在线学习平台”项目为主线，将书中的各个知识点以案例形式串联起来，通过“讲练结合”，全面、翔实地介绍了 JSP 开发所需的各种知识和技能。

全书共分 11 章，各章知识点分布在三个板块中：一是由第 1、2、3 章构成辅助知识板块，包括 Web 技术简介、Web 客户端技术、JSP 开发运行环境；二是由第 4、5、6、7、9 章构成核心技术板块，包括 JSP 基本语法、JSP 内置对象、数据库操作、JavaBean 组件、Servlet 技术；三是由第 8、10、11 章构成扩展技术板块，包括 JSTL 标签库、JSP 开发模式、JSP 高级应用。

通过对本书的学习，读者可以快速、全面地掌握使用 JSP 开发 Web 应用程序的方法。本书作为“项目驱动、案例教学、理论实践一体化”教学方法的载体，主要有以下特色：

（1）合理的知识结构。在整个专业培养体系中，《JSP 程序设计教程》是《Java 程序设计语言》《数据结构》《数据库技术及应用》《静态网页设计》等的综合应用载体，针对 JSP 技术的特点和要求，将这些前导课程知识点做了取舍，作为复习内容有机地穿插到 JSP 相关技术应用当中，通过已学知识带出新知识，达到温故而知新的效果。

（2）近似实战的案例教学。将一个实际的项目适当简化，将难度控制在读者的学习能力范围内。根据本书知识点的分布情况，分解成具有递进关系又相对独立的案例，由浅至深地分布在第 2 章及其之后的各个章节中，每个案例的知识应用对应着这个章节应掌握技术重点，其中很多都是当前 Web 开发中经常使用的，体现了理论学习和具体应用的充分结合。读者学习案例的过程，就是学习 JSP 知识和技能的过程，达到“知识案例的一体化”；读者在完成案例的同时，也就完成了项目的某一功能，达到“案例项目的一体化”。读者能够在掌握基本理论知识的同时，提高综合实践能力。书中的所有程序均由笔者亲自编程调试。

（3）理实一体化教学。打破传统的先理论学习、后上机练习的教学模式，将知识讲

解和技能训练放在同一教学单元和教学地点完成，融“教、学、练”于一体。每一案例的教学均按固定的模式展开，即首先从客户的角度提出【案例需求】，其次分析需求拟定【案例目标】，然后列出实现目标需要哪些知识点作为【案例要点】，最后是实现目标的【制作步骤】。历经多次教师演示—读者模仿的循环，让读者熟悉项目模块设计制作的详细的、完整的过程，体现“在做中学、学以致用”的教学理念。

(4) 阶梯式实践教学。除了各个章节中案例实践，本书精心设置了一些课堂小练习，作为案例实践的补充，也是作为案例实践的进阶前奏。随着章节学习的深入，“在线学习平台”项目业务功能逐步丰富和完善，读者不仅感受到案例实践综合性和难度逐步加大，也体会到一个项目的持续开发伴随着自己技术上的成长而逐渐瓜熟蒂落。让读者通过不断实践，实现编程技能的逐步推进，最终实现与职业能力的“零距离”。

本书由广西交通职业技术学院的莫兵、广西科技师范学院的蒙焕念、广西科技师范学院的黄良永担任主编，由湖南科技职业学院的李建平、河南牧业经济学院的史岳鹏担任副主编。其中，莫兵老师编写了第 4、9、11 章，蒙焕念老师编写了第 1、5、6 章，黄良永老师编写了第 3、7、8 章，李建平老师编写了第 2 章，史岳鹏老师编写了第 10 章。在本书的策划、出版过程中，采纳了许多教学一线教师的意见，得到了同行院校的大力配合，在此表示真诚的谢意。本书的相关资料和售后服务可扫封底的二维码或与 QQ (2436472462) 联系获得。

本书适合作为应用型本科院校、高职高专计算机类专业“JSP 程序设计”课程的教材，也可以作为培训教材使用。由于时间仓促以及编者水平有限，书中难免存在错误和疏漏之处，欢迎广大读者和同仁批评指正。

编者

2015 年 6 月

此为试读，需要完整PDF请访问：www.ertongbook.com

目 录

第1章 Web 技术简介1

1.1 Web 基础知识.....1

1.1.1 Web 发展过程.....1

1.1.2 Web 动态网页技术.....2

1.1.3 Web 应用程序工作原理.....2

1.2 JSP 技术概述.....3

1.2.1 JSP 技术主要优点.....3

1.2.2 JSP 技术基本部分.....4

1.3 本章小结.....5

1.4 本章练习.....6

第2章 Web 客户端技术8

2.1 HTML.....8

2.1.1 HTML 标签与属性.....8

2.1.2 HTML 文档.....12

2.1.3 案例一：导航和框架的应用13

2.1.4 案例二：表格与表单的应用15

2.2 页面布局.....17

2.2.1 常用的页面布局方法.....18

2.2.2 CSS 简介.....18

2.2.3 CSS 基本语法.....18

2.2.4 在网页中加载 CSS 样式表.....21

2.2.5 DIV 布局和 CSS 盒子模型.....22

2.2.6 案例三：CSS+DIV 页面布局...24

2.2.7 案例四：CSS 制作横向菜单...27

2.3 JavaScript29

2.3.1 JavaScript 基础.....30

2.3.2 JavaScript 对象.....30

2.3.3 JavaScript 事件.....32

2.3.4 在网页中使用 JavaScript34

2.3.5 案例五：JavaScript 进行数据验证.....36

2.4 本章小结.....38

2.5 本章练习.....39

第3章 JSP 开发运行环境41

3.1 JSP 开发运行环境介绍.....41

3.1.1 JDK 介绍.....41

3.1.2 Tomcat 介绍.....42

3.1.3 MyEclipse 介绍.....43

3.2 JSP 运行环境搭建.....43

3.2.1 JDK 安装和配置.....43

3.2.2 Tomcat 安装和配置.....45

3.2.3 MyEclipse 安装和配置.....47

3.2.4 MyEclipse 优化及使用.....49

3.3 案例一：用 MyEclipse 开发 JSP.....54

3.3.1 Web 服务目录.....56

3.3.2 JSP 运行机制.....57

3.4 本章小结.....58

3.5 本章练习.....58

第4章 JSP 基本语法60

4.1 JSP 页面基本构成.....60

4.2 JSP 中的注释.....61

4.3 指令元素.....62

4.3.1 page 指令.....62

4.3.2 include 指令.....64

4.3.3 taglib 指令.....65

4.3.4 案例一：指令元素在 JSP 开发中应用.....65

4.4 脚本元素.....67

4.4.1 小脚本（Scriptlet）.....67

4.4.2 声明（Declaration）.....67

4.4.3 表达式（Expression）.....68

4.4.4 案例二：脚本元素在 JSP 开发中应用.....68

4.5 动作元素.....71

4.5.1 包含文件<jsp:include>	72	查询制作	134
4.5.2 网页跳转<jsp:forward>	72	6.3 本章小结	137
4.5.3 使用 Applet 插件		6.4 本章练习	138
<jsp: plugin >	73	第 7 章 JavaBean 组件	140
4.6 本章小结	75	7.1 认识 JavaBean	140
4.7 本章练习	75	7.2 编写 JavaBean 遵循的原则	141
第 5 章 JSP 内置对象	77	7.3 在 Jsp 中使用 JavaBean	142
5.1 JSP 内置对象的基本知识	77	7.3.1 使用 JavaBean 的优点	142
5.1.1 JSP 内置对象的分类	77	7.3.2 使用<jsp:useBean>动作	
5.1.2 内置对象属性保存范围	77	元素	143
5.2 JSP 内置对象的常用方法	78	7.3.3 JavaBean 运行机制	144
5.2.1 out 请求对象	78	7.3.4 JavaBean 字节码文件存放	145
5.2.2 request 请求对象	79	7.4 JavaBean 属性操作	146
5.2.3 response 响应对象	81	7.4.1 设置 JavaBean 属性值	146
5.2.4 session 会话对象	84	7.4.2 访问 JavaBean 属性值	149
5.2.5 application 应用程序对象	85	7.5 案例一：在页头动态显示当前	
5.2.6 其他对象	86	学期	149
5.3 JSP 内置对象的案例	89	7.6 案例二：网上调查用户兴趣	154
5.3.1 案例一：request 对象处理		7.7 本章小结	156
用户登录数据	89	7.8 本章练习	156
5.3.2 案例二：session 对象处理		第 8 章 JSTL 标签库	159
用户登录数据	93	8.1 JSTL 标签库概述	159
5.4 本章小结	96	8.1.1 什么是 JSTL	159
5.5 本章练习	97	8.1.2 JSTL 小例子	160
第 6 章 数据库操作	99	8.2 EL 表达式	161
6.1 JDBC 技术	99	8.2.1 获取数据	162
6.1.1 JDBC 简介	99	8.2.2 执行运算	164
6.1.2 JDBC 的体系结构	100	8.2.3 获得 Web 开发常用对象	166
6.1.2 JDBC 驱动程序	101	8.2.4 调用 Java 方法	171
6.1.3 JDBC API 常用类和接口	103	8.2.5 JSTL 中的常用 EL 函数	174
6.2 数据库操作案例	113	8.3 JSTL 核心标签	175
6.2.1 案例一：连接数据库实现		8.3.1 表达式控制标签使用	175
系统登录	113	8.3.2 流程控制标签使用	176
6.2.2 案例二：课程信息操作		8.3.3 循环标签使用	177
主界面设计	115	8.3.4 URL 操作标签使用	179
6.2.3 案例三：对课程信息编辑		8.4 I18N 格式化标签	182
模块制作	122	8.5 其他标签	185
6.2.4 案例四：对课程信息分页		8.5.1 SQL 标签	185

8.5.2 XML 标签	189	第 10 章 JSP 开发模式	221
8.6 案例一：用 JSTL 优化首页脚本	191	10.1 JSP 主要开放模式	221
8.7 案例二：用 JSTL 优化课程主页脚本	194	10.1.1 Model 1 模式	221
8.8 本章小结	196	10.1.2 Model 2 模式	222
8.9 本章练习	197	10.1.3 两种模式比较	223
第 9 章 Servlet 技术	198	10.2 MVC 模式	223
9.1 认识 Servlet	198	10.2.1 MVC 模式原理	223
9.1.1 什么是 Servlet	198	10.2.2 JSP+JavaBean+Servlet 实现 MVC	225
9.1.2 Servlet 优点	199	10.3 案例一：利用 MVC 模式改进模块	226
9.1.3 Servlet 用途	199	10.4 本章小结	229
9.2 开发一个简单的 Servlet	200	10.5 本章练习	229
9.2.1 编写 Servlet	200	第 11 章 JSP 高级应用	232
9.2.2 配置 Web.xml	202	11.1 文件的上传与下载	232
9.2.3 调用 Servlet	203	11.1.1 相关知识	232
9.3 Servlet 运行原理	204	11.1.2 实例演示	233
9.3.1 HTTP 及其请求方法	204	11.2 登录验证过滤器	239
9.3.2 Servlet 与 Servlet 容器	205	11.2.1 相关知识	239
9.3.3 Servlet 的生命周期	206	11.2.2 实例演示	241
9.4 Servlet API	207	11.3 在线用户统计	247
9.4.1 Javax.servlet 常用接口及类	207	11.3.1 相关知识	247
9.4.2 Javax.servlet.http 常用接口及类	208	11.3.2 实例演示	248
9.4.3 Servlet 的方法	209	11.4 海量数据查询分页	256
9.5 案例一：Servlet 改进课程主页脚本	211	11.4.1 相关知识	256
9.6 本章小结	218	11.4.2 实例演示	258
9.7 本章练习	219	11.5 本章小结	270
		11.6 本章练习	270

第 1 章 Web 技术简介

【本章学习目标】

- 了解 Web 技术的发展历史
- 对 JSP 技术等 Web 动态网页技术有大致的认识
- 理解 C/S 模式 B/S 模式的特点

1.1 Web 基础知识

通过了解 Web 技术发展过程，充分认识 Web 动态网页技术对当今社会的影响，大致了解 Web 动态网页技术特点和发展趋势。

1.1.1 Web 发展过程

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均能存储和显示共享信息。同时，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 Andreessen 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。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问题。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。

1.1.2 Web 动态网页技术

Web 动态网页技术是运行在服务器端的 Web 应用程序,程序根据用户的请求在服务器端进行动态处理后,把处理的结果以 HTML 文件格式返回给客户端。目前流行的 Web 动态网页开发技术有 ASP.NET、PHP、JSP。

(1) **ASP.NET**: 是微软公司继 ASP 之后推出的新一代动态网站开发技术,ASP.NET 是基于 .NET 框架平台 (Microsoft .NET Framework), 用户可以选择 .NET 框架下自己喜欢的语言进行开发。ASP.NET 技术是 ASP 技术的更新,也是微软公司目前主推的技术。但是由于微软的产品永远都会受到平台的限制。往往只用于小型项目的开发。

(2) **PHP**: 是超文本预处理器 (Hypertext Preprocessor) 英文缩写,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 语言的语法,并结合自身的特性,使 Web 开发者能够迅速地写出动态页面。Web 服务器软件采用 Apache 软件基金会的一个开放源码的网页服务器 Apache HTTP Server (简称 Apache)。

(3) **JSP**: 是 Java 服务器页面 (Java Server Pages) 英文缩写,使用 Java 完成的动态 Web 开发,代码风格和 ASP 类似,都属于在 HTML 代码中嵌入 Java 代码以实现功能,由于 Java 语言的跨平台特性,所以 JSP 不会受到操作系统或开发平台的制约,而且有多种 Web 服务器软件可以支持,如 Tomcat、WebLogic、JBoss、WebSphere 等,所以经常在大中型开发中使用。

1.1.3 Web 应用程序工作原理

JSP 页面是运行在服务器端的一种 Web 应用程序。在学习 JSP 技术前,先了解一下 Web 应用程序的工作原理。

目前在 Internet 上信息大多以网页形式存储在服务器上,通过浏览器获取网页内容,这是一种典型的 B/S (Browser/Server, 浏览器/服务器) 模式。它是随着 Internet 技术的兴起,对 C/S (Client/Server, 客户机/服务器) 模式应用的扩展。

B/S 模式的工作过程是:把请求(即为一个网址)发送到服务器端响应的 Web 应用程序(静态网页和动态网页),Web 程序接收请求后进行相关处理,并由 Web 程序把结果以文本或网页的形式,返回到浏览器。因此当访问 Web 站点时,我们所看到的所有内容都是在浏览器中显示的,是在 HTTP 协议和 B/S 结构的基础上实现的。HTTP 协议是 WWW (World Wide Web) 信息高速公路上传输信息的超文本传输协议。具体体现在 URL (Uniform Resource Location) 统一资源定位符,它的格式为 `http://host[:port][abs_path]`,解释如下:

- **http** 表示要通过 HTTP 协议来定位网络资源;
- **host** 表示存有该资源的 Internet 主机域名或主机 IP 地址(有时也包括端口号 port);
- **abs_path** 表示主机资源的具体地址,如目录和文件名等。

第一部分和第二部分之间用 “://” 符号隔开,第二部分和第三部分用 “/” 符号隔开。第一部分和第二部分是不可缺少的,第三部分有时可以省略,即目录和文件名可以省略,但 “/” 符号不能省略。

下面有 3 种形式的 URL:

- **www.haha.com**（最终浏览器将此形式自动转为以下第二种形式访问，并使用默认的端口 80）。
- **http://www.haha.org/**。
- **http://192.168.0.104:8080/index.jsp**（使用 IP 地址和指定端口号访问资源，由于使用的服务器是 Tomcat，它的默认端口是 8080）。

Web 应用程序工作原理如图 1-1 所示。

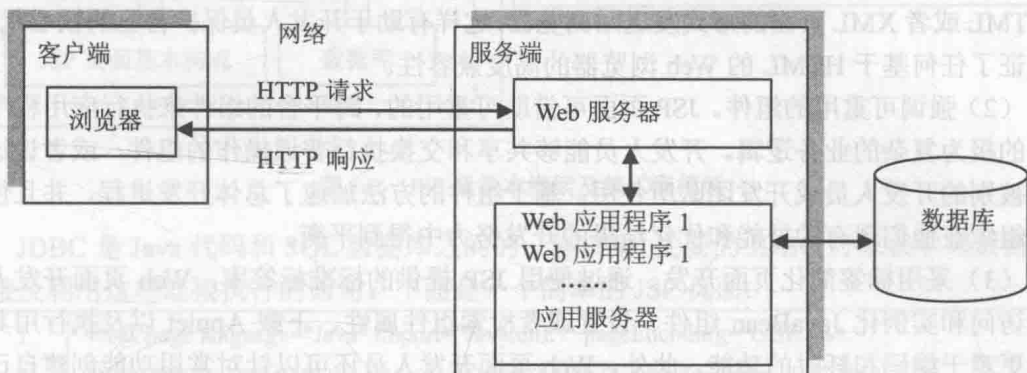


图 1-1 Web 应用程序工作原理

在这种结构下，用户工作界面是通过 IE 浏览器来实现的。B/S 模式最大的优点是运行维护比较简便，能实现不同的人员、从不同的地点、以不同的接入方式（例如 LAN、WAN、Internet/Intranet 等）访问和操作共同的数据，联网成本低；最大的缺点是对外网环境依赖性太强，由于各种原因引起外网中断都会造成系统瘫痪。由于是架构在互联网上的，安全技术要求比 C/S 模式高，响应速度较 C/S 模式慢。

1.2 JSP 技术概述

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，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，就是 JSP 是一种实现普通静态 HTML 和动态 HTML 混合编码的技术。它实现了 HTML 语法中的 Java 扩张（以 `<% , %>` 形式）。

1.2.1 JSP 技术主要优点

JSP 技术以 Java 语言为基础，继承了 Java 语言的许多优点，因此使用 JSP 开发动态网站十分方便，开发效率较高。JSP 以 Servlet 技术为基础，又在许多方面做了改进。同时，JSP 利用跨平台运行的 JavaBean 组件，可以方便地实现组件复用，进一步提高了开发效率。因此，JSP 已成为目前网站建设主流的动态网站开发技术之一。具体来说，JSP 技术主要有以下几个优点：

- （1）将内容的生成和显示分离。借助 JSP 技术，Web 页面开发人员可以使用 HTML



或者 XML 标签来设计和格式化最终页面,使用 JSP 标签或者脚本程序来生成动态 Web 页面的内容(页面的内容是根据请求来变化的,例如,请求账户信息或者特定的一本书的价格)。生成内容的逻辑被封装在标签和 JavaBean 组件中,并且捆绑在脚本程序中,所有的脚本程序在服务器端运行。因此,其他人能够方便地编辑和使用 JSP 页面,而不影响内容的生成。

在服务器端,由 JSP 引擎负责 JSP 标识和脚本程序,生成所请求的内容,并且将结果以 HTML 或者 XML 页面的形式发送回浏览器,这样有助于开发人员保护自己的核心代码,又保证了任何基于 HTML 的 Web 浏览器的高度兼容性。

(2) 强调可重用的组件。JSP 页面可借助可重用的、跨平台的组件来执行应用程序所要求的极为复杂的业务逻辑。开发人员能够共享和交换执行普通操作的组件,或者让这些组件被别的开发人员或开发团队所使用。基于组件的方法加速了总体开发进程,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。

(3) 采用标签简化页面开发。通过使用 JSP 提供的标准标签库,Web 页面开发人员能够访问和实例化 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。此外,Web 页面开发人员还可以针对常用功能创建自己的标签库。

(4) 一次编写、处处运行。由于 JSP 页面的内置脚本是基于 Java 语言的,而且所有的 JSP 页面都被编译为 Java Servlet,因此 JSP 页面就具有 Java 技术的所有优点,包括健壮性和安全性等。作为 Java 平台的一部分,JSP 拥有 Java 语言“一次编写,处处运行”的特点。JSP 几乎可以运行于所有的平台,如 Linux、Unix 等。

(5) 更高的效率 and 安全性。JSP 程序在执行之前先被编译成字节码文件,字节码文件由 Java 虚拟机解释执行,比源代码解释的效率 high。此外,服务器端还有字节码的 Cache 机构,能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢,因为它被编译成 Cache,以后就快得多了。同时,JSP 源程序不大可能被下载,特别是 Javabeen 程序,完全可能放到不对外目录中。

1.2.2 JSP 技术基本部分

JSP 技术最基本内容包括:页面基本构成和内置对象,若对数据库操作,还涉及 JDBC。由这些内容构成 JSP 最基本框架,再由基本框架扩展到更高级框架,如图 1-2 所示。

页面基本构成包括:HTML 标记代码、JavaScript 语句、CSS 层叠样式、JSP 基本语法。

JSP 语法包括:Java 程序片段、JSP 内置标记、JSP 自定义标记等。

JSP 中一共预定义了 9 个内置对象,分别为: request、response、session、application、out、pageContext、config、page、exception,常用的有 out、request、response、session 等。

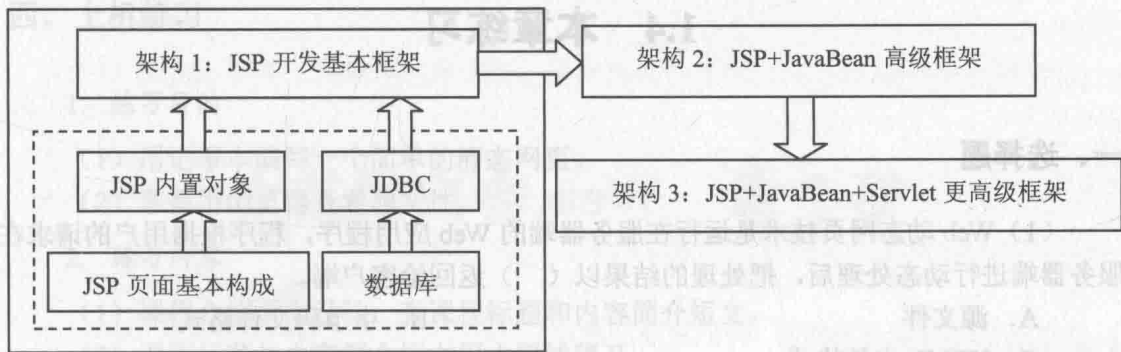


图 1-2 JSP 最基本框架及其扩展框架

JDBC 是 Java 代码和 SQL 数据库之间的一座桥梁。主要的 JDBC 对象表示同数据库的连接及利用这些连接执行的语句。下面是一个简单的 JSP 页面：

```

1  <%@ page language="Java" import="Java.util.*" pageEncoding="GBK"%>
2  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional/EN">
3  <html>
4  <head> <title>我的第一个 JSP</title> </head>
5  <body>
6      <h1>我的第一个 JSP</h1><small>
7          <% int iCnt = 1; iCnt = iCnt+5;
8              out.println("用内置对象 out 提供的方法显示 iCnt="+iCnt); %>
9          <br/>用脚本元素显示 iCnt=<%=iCnt%>
10         </small><p>页面的其余内容</p>
11 </body></html>

```

【例子说明】第 1 行为 JSP 指令元素，page 指令用于定义与整个 JSP 页面相关的属性，对整个页面有效。第 7 行第一个<%…%>是 JSP 脚本小程序元素，其中，第一句定义了一个 int 变量，名为 iCnt，并赋初值，第二句对变量 iCnt 做加法操作，第三句用 out 对象提供的 println 方法在网页显示内容。第 9 行第二个<% …%>是 JSP 表达式元素，作用与 out 对象提供的 println 方法一样。

1.3 本章小结

本章介绍了 Web 技术的发展历史，特别是对 Web 动态技术发展趋势做了较为详细的介绍，重点介绍了 JSP 动态网页技术所包含的内容和特点，初步接触一个简单的 JSP 页面例子，为后面深入学习做一个较为充分的铺垫。从 JSP 基本技术构成看，学好 Web 动态网页技术前提条件是，必须学好静态网页技术。

1.4 本章练习

一、选择题

(1) Web 动态网页技术是运行在服务器端的 Web 应用程序，程序根据用户的请求在服务器端进行动态处理后，把处理的结果以 () 返回给客户端。

- A. 源文件
- B. 字节码文件格式
- C. HTML 文件格式
- D. 什么都不做

(2) B/S 模式与 C/S 模式比较，下列哪项说法是对的 ()。

- A. 安全性方面 B/S 比 C/S 更容易实现
- B. C/S 模式对外网环境依赖性太强
- C. C/S 的响应速度比 B/S 慢些
- D. B/S 模式维护简单

(3) JSP 有多种 Web 服务器软件可以支持，下列哪项不是 ()。

- A. IIS
- B. WebSphere
- C. Tomcat
- D. WebLogic

(4) 下列关于动态网页与静态网页的区别描述是错误的 ()

- A. 静态网页服务器端返回的 HTML 文件是事先存储好的
- B. 动态网页服务器端返回的 HTML 文件是程序生成的
- C. 网页中有动画播放，所以它是动态网页
- D. 动态网页除了有程序代码，还有 HTML 标签

(5) URL 是 Internet 中的资源的命名机制，URL 由三个部分组成 ()。

- A. 协议、Internet 主机域名或 IP 地址、文件名
- B. 主机、Internet 主机域名或 IP 地址和文件名、协议
- C. 协议、文件名、主机名
- D. 协议、文件名、IP 地址

二、判断题

(1) 在 Web 诞生之初的一段较长的时间里，一直是完全静态的。()

(2) CGI 是最早实现的动态 Web 技术，执行效率很高。()

(3) ASP.NET 是微软公司目前主推的 Web 动态网页技术，该技术具有跨平台特性。()

(4) B/S 模式的工作过程是：客户端发出请求，服务器处理并对客户端响应。()

(5) Apache 是 ASP.NET 技术用于开发网站所支持的 Web 服务器。()

三、简答题

(1) JSP 技术主要优点有哪些？

(2) JSP 页面基本构成包括哪些？



四、上机练习

1. 练习目的

- (1) 用记事本编写一个简单的静态网页。
- (2) 掌握用浏览器查看源文件。

2. 练习内容

- (1) 课程介绍页面设计，有课程标题和内容简介短文。
- (2) 课程标题与内容简介短文用水平线隔开。

表 1-1 常用 HTML 标签的属性

属性	常用属性	属性说明
文档标题		1. HTML 标签 (HTML tag) 是 HTML 文档中用于定义网页内容的符号。HTML 标签通常由尖括号 (< 和 >) 包围，有些标签还需要属性。HTML 标签分为单标签 (如) 和双标签 (如 和)。HTML 标签的属性用于指定标签的某些特性，如类 (class)、ID (id)、样式 (style) 等。HTML 标签的属性通常写在标签的左尖括号内，属性名和属性值之间用等号 (=) 连接。属性值可以用双引号 (") 或单引号 (') 包围。HTML 标签的属性用于指定标签的某些特性，如类 (class)、ID (id)、样式 (style) 等。HTML 标签的属性通常写在标签的左尖括号内，属性名和属性值之间用等号 (=) 连接。属性值可以用双引号 (") 或单引号 (') 包围。
<html> ... </html>		1. HTML 标签 (HTML tag) 是 HTML 文档中用于定义网页内容的符号。HTML 标签通常由尖括号 (< 和 >) 包围，有些标签还需要属性。HTML 标签分为单标签 (如) 和双标签 (如 和)。HTML 标签的属性用于指定标签的某些特性，如类 (class)、ID (id)、样式 (style) 等。HTML 标签的属性通常写在标签的左尖括号内，属性名和属性值之间用等号 (=) 连接。属性值可以用双引号 (") 或单引号 (') 包围。
...		2. HTML 标签 (HTML tag) 是 HTML 文档中用于定义网页内容的符号。HTML 标签通常由尖括号 (< 和 >) 包围，有些标签还需要属性。HTML 标签分为单标签 (如) 和双标签 (如 和)。HTML 标签的属性用于指定标签的某些特性，如类 (class)、ID (id)、样式 (style) 等。HTML 标签的属性通常写在标签的左尖括号内，属性名和属性值之间用等号 (=) 连接。属性值可以用双引号 (") 或单引号 (') 包围。
	alt 属性	定义文本的颜色或背景色等属性

第2章 Web 客户端技术

【本章学习目标】

- 了解 HTML 基础知识
- 掌握使用常用的网页编辑工具设计网页
- 掌握 DIV+CSS 标准布局的方法
- 学会使用客户端脚本语言改善的网页的动态性

2.1 HTML

HTML 是英文 Hyper Text Mark-up Language (文本标签语言) 的缩写, 它规定了自己的语法规则, 用来表示比“文本”更丰富的意义, 例如图片、表格、链接等。浏览器 (IE、FireFox 等) 软件知道 HTML 语言的语法, 可以用来查看 HTML 文档。目前互联网上的绝大部分网页都是使用 HTML 编写的, 其特点如下:

- HTML 是用来描述网页的一种语言;
- HTML 不是一种编程语言, 而是一种标签语言 (Mark-up language);
- 标签语言是一套标签 (Mark-up tag);
- HTML 使用标签来描述网页。

2.1.1 HTML 标签与属性

1. HTML 标签 (HTML tag) 的特点

HTML 标签 (HTML tag) 的特点如下:

- HTML 标签对大小写不敏感, 万维网联盟 (W3C) 在 HTML 4 中推荐使用小写;
- HTML 标签是由尖括号包围的关键词, 例如 <html>;
- HTML 标签通常是成对出现的, 例如 和 ;
- HTML 标签对中的第一个标签是开始标签, 第二个标签是结束标签;
- HTML 标签从开始标签 (Start tag) 到结束标签 (End tag) 的所有代码, 我们称之为 HTML 元素, 大多数 HTML 元素可以嵌套 (可以包含其他 HTML 元素);
- HTML 标签存在少量不成对出现的情况, 例如
 和 , 没有内容的 HTML 元素被称为空元素。

2. HTML 属性

大多数 HTML 标签（元素）拥有属性。属性提供了有关 HTML 元素的更多的信息，其特点如下：

- 属性总是以名称/值对的形式出现，例如：`name="value"`；
- 属性总是在 HTML 元素的开始标签中规定；
- 属性和属性值对大小写不敏感，不过万维网联盟在其 HTML 4 推荐标准中推荐小写的属性/属性值；
- 属性值应该始终被包括在引号内，双引号是最常用的，若属性值本身含有双引号，那么可使用单引号包括属性值；
- 属性可以设置多个，且之间无先后次序，可选，若省略则采用默认值，格式如下：
`<标签名 属性 1 属性 2 属性 3...> 内容 </标签名>`。

例如：`<h1>` 定义标题的开始，`<h1 align="center">`拥有关于对齐方式的附加信息。
`<body>` 定义 HTML 文档的主体，`<body bgcolor="yellow">`拥有关于背景颜色的附加信息。
`<table>`定义 HTML 表格，`<table border="1">` 拥有关于表格边框的附加信息。
适用于大多数 HTML 标签的属性如表 2-1 所示。

表 2-1 适用于大多数 HTML 标签的属性

属性	值	描述
class	classname	规定元素的类名（classname）
id	id	规定元素的唯一 id
style	style_definition	规定元素的行内样式（inline style）
title	text	规定元素的额外信息（可在工具提示中显示）

(3) HTML 常用的标签如表 2-2 所示。

表 2-2 HTML 常用标签

标签	常用属性	功能说明
文档标签		
<code><html> ... </html></code>		定义文档，描述网页
<code><head> ... </head></code>		定义文档头部，包含整个文档的一般信息，例如文档标题、文档整体描述、文档关键字等
<code><title> ... </title></code>		定义文档标题，显示在浏览器的标题栏中。
<code><body> ...</body></code>	Bgcolor 背景色 Background 背景图案	定义文档主体，显示可见的页面内容
<code><!-- 注释--></code>		定义注释，不在网页上显示
文本标签		
<code> ...</code>	color 颜色	定义文本的颜色、尺寸、字体