

“十二五”普通高等教育本科国家级规划教材

# Web 程序设计

(第5版)

吉根林 顾韵华 主编  
吴军华 殷红先 副主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是“十二五”普通高等教育本科国家级规划教材，以 JSP 程序设计技术为主线介绍 Web 程序设计的方法与技术。全书共 10 章，包括：Web 基础知识和开发运行环境；HTML、XML 和 CSS；JavaScript 程序设计；JSP 基本语法与内置对象；Servlet 与 JavaBean；JSP 数据库应用；JSP 实用组件；表达式语言与标签；Java EE 框架技术基础；JSP 综合应用实例。每章配有大量实例、习题和上机实验题及实验指导，免费提供 PPT 教学课件和程序源代码。

本书可作为高校计算机科学与技术、软件工程、网络工程、电子商务、人工智能、数据科学与大数据技术、信息管理与信息系统、现代教育技术等相关专业的教材，也是 Web 程序开发人员实用的技术参考书。

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

Web 程序设计 / 吉根林，顾韵华主编. —5 版. —北京：电子工业出版社，2019.11

ISBN 978-7-121-36443-3

I. ① W… II. ① 吉… ② 顾… III. ① 网页制作工具—程序设计—高等学校—教材

IV. ① TP393.092.2

中国版本图书馆 CIP 数据核字（2019）第 083134 号

责任编辑：章海涛

印 刷：

装 订：

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

开 本：787×1 092 1/16 印张：22 字数：562 千字

版 次：2002 年 8 月第 1 版

2019 年 11 月第 5 版

印 次：2019 年 11 月第 1 次印刷

定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

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。

本书咨询联系方式：192910558（QQ 群）。

# 前 言

承蒙广大读者的支持，本书自第1版出版以来，被几十所高校选为相关课程教材，已连续印刷几十次。在第4版教材出版后的几年中，Web应用程序开发技术有了新的发展，同时从服务教学、服务读者的角度看，本书还需进一步完善，为此有必要对第4版进行修订。

本书是“十二五”普通高等教育本科国家级规划教材，也是精品课程和优秀教材建设的成果。

本次修订继续保持原书的基本风格，根据Web应用程序开发技术的发展趋势，内容做了较大的调整：

- ① 以 JSP 程序设计技术为主线介绍 Web 应用程序开发技术。
- ② 坚持“Web 程序设计”课程的教学目标：学会建网站。
- ③ 对部分章节的内容进行了优化，进一步提高了本书的先进性和实用性。

本次修订的具体内容如下：

① 调整、优化本书的结构框架，主要介绍 JSP 程序设计的基本内容与相关技术，读者学会利用 JSP 技术开发 Web 应用程序，学会建网站。

② 删除了第4版中有关 ASP.NET 程序设计基本内容和开发技术。

③ 调整 Web 应用程序开发运行环境，使用 JDK、Tomcat、Eclipse 进行 Web 应用程序的开发，介绍了它们的下载、安装与配置方法。

④ 增加了网站安全的知识和技术。

⑤ “综合应用实例”章将 ASP.NET 综合应用实例替换为两个 JSP 综合应用程序开发实例，以培养学生对 JSP 程序设计技术的综合应用能力。

本教材的参考教学时数约为 90~100 学时，其中理论教学 54~60 学时，上机实验 36~40 学时。全书配有大量例题，每章安排了习题和上机实验题，其内容可能比教学时数所允许的分量稍多，可供教师讲课时选取或让学生自学。

本教材为任课教师提供 PPT 教学课件及例题源程序。任课老师可在华信教育资源网 <http://www.hxedu.com.cn> 免费注册后下载。欢迎任课教师及时反馈您的授课心得和建议。

在修订过程中，第1章由南京师范大学吉根林教授执笔；第2、3、4章由南京信息工程大学顾韵华教授执笔；第5、6章分别由南京工业大学吴军华副教授执笔；第7、8、9、10章由南京师范大学殷红先老师编写；全书由吉根林和顾韵华担任主编，并统稿、定稿。本次修订过程中，得到了电子工业出版社的支持，在此表示衷心的感谢！

由于编者水平有限，本书还会存在错误与不足之处，恳请广大读者与同行给予批评指正。  
编者 E-mail 地址：[glji@njnu.edu.cn](mailto:glji@njnu.edu.cn)。

作 者  
于南京



# 目 录

第 1 章 Web 基础知识与开发运行环境 .....	1
1.1 Web 工作原理 .....	1
1.2 Internet 网络协议 .....	3
1.2.1 TCP/IP 协议族 .....	4
1.2.2 HTTP .....	4
1.2.3 Telnet .....	5
1.2.4 FTP .....	5
1.3 IP 地址、域名和 URL .....	5
1.3.1 IP 地址 .....	5
1.3.2 域名 .....	6
1.3.3 URL .....	6
1.4 动态网页设计技术简介 .....	7
1.4.1 PHP .....	7
1.4.2 JSP .....	8
1.4.3 ASP.NET .....	9
1.5 Web 应用程序开发工具与运行环境 .....	9
1.5.1 网站架构 .....	10
1.5.2 JDK 的下载、安装与配置 .....	10
1.5.3 Tomcat 的下载、安装与配置 .....	11
1.5.4 Eclipse 的下载、安装与配置 .....	12
1.6 简单的 Web 应用程序示例 .....	12
1.7 网站安全问题 .....	15
1.7.1 网站攻击手段 .....	15
1.7.2 网站的保护与安全措施 .....	15
本章小结 .....	17
习题 1 .....	18
第 2 章 HTML、XML 和 CSS .....	19
2.1 页面设计概述 .....	19
2.1.1 静态网页 .....	19
2.1.2 动态网页 .....	19
2.1.3 网页的设计风格 .....	20
2.2 超文本标记语言 HTML .....	20
2.2.1 HTML 文档结构 .....	21
2.2.2 HTML 基本标记 .....	23
2.2.3 表格 .....	27
2.2.4 表单 .....	30

2.2.5	框架 (Frame) .....	32
2.3	HTML5.....	35
2.3.1	HTML5 新特性.....	35
2.3.2	HTML5 新功能.....	35
2.3.3	HTML5 网页示例.....	39
2.4	层叠样式表 CSS .....	43
2.4.1	为什么需要层叠样式表 .....	43
2.4.2	样式表的定义和引用 .....	44
2.4.3	样式的优先级.....	49
2.4.4	CSS 属性 .....	51
2.4.5	CSS+DIV 页面布局.....	58
2.5	XML 简介 .....	59
2.5.1	XML 概述.....	59
2.5.2	XML 文档结构.....	61
2.5.3	XML 文档显示.....	63
2.6	应用示例: 个人主页设计.....	64
	本章小结.....	67
	习题 2 .....	67
	上机实验 2.....	67
<b>第 3 章</b>	<b>JavaScript 程序设计.....</b>	<b>71</b>
3.1	脚本语言概述 .....	71
3.1.1	什么是脚本语言 .....	71
3.1.2	JavaScript 的特点 .....	72
3.2	JavaScript 基础 .....	73
3.2.1	JavaScript 程序的编辑和调试.....	73
3.2.2	JavaScript 基本语法 .....	74
3.2.3	JavaScript 函数.....	76
3.2.4	JavaScript 流程控制 .....	78
3.2.5	JavaScript 出错处理 .....	80
3.2.6	JavaScript 表单验证 .....	80
3.2.7	JavaScript 正则表达式 .....	81
3.3	JavaScript 事件 .....	82
3.3.1	JavaScript 事件驱动机制.....	82
3.3.2	JavaScript 常用事件 .....	83
3.3.3	JavaScript 事件触发与处理.....	83
3.3.4	应用示例: 计算器的设计.....	84
3.4	JavaScript 对象 .....	86
3.4.1	对象的定义和引用 .....	86
3.4.2	for..in 和 with 语句.....	87
3.4.3	JavaScript 内置对象 .....	89
3.5	浏览器对象模型及应用 .....	98

3.5.1	浏览器对象模型 .....	98
3.5.2	Navigator 对象 .....	99
3.5.3	Window 对象 .....	100
3.5.4	Document 对象 .....	103
3.5.5	Form 对象 .....	109
3.5.6	History 和 Location 对象 .....	114
3.5.7	Frame 对象 .....	114
3.5.8	应用示例：用户注册信息合法性检查 .....	116
3.5.9	应用示例：扑克牌游戏程序 .....	120
3.6	HTML DOM .....	123
3.6.1	HTML DOM 概述 .....	123
3.6.2	DOM 节点树 .....	124
3.6.3	DOM 树节点的属性 .....	124
3.6.4	访问 DOM 节点 .....	125
3.7	JavaScript 框架和库 .....	127
	本章小结 .....	127
	习题 3 .....	128
	上机实验 3 .....	128
<b>第 4 章</b>	<b>JSP 基本语法与内置对象 .....</b>	<b>130</b>
4.1	JSP 基本语法 .....	130
4.1.1	JSP 页面 .....	131
4.1.2	JSP 指令 .....	130
4.1.3	JSP 脚本标识 .....	131
4.2	JSP 内置对象 .....	134
4.2.1	Request 对象 .....	135
4.2.2	Response 对象 .....	137
4.2.3	Session 对象 .....	139
4.2.4	Application 对象 .....	143
4.2.5	其他对象 .....	145
4.3	JSP 动作标识 .....	149
4.3.1	include 动作标识 .....	149
4.3.2	forward 动作标识 .....	150
4.3.3	param 动作标识 .....	152
4.4	Cookie 及其应用 .....	153
4.5	应用示例：Web 聊天程序 .....	155
	本章小结 .....	159
	习题 4 .....	159
	上机实验 4 .....	160
<b>第 5 章</b>	<b>Servlet 与 JavaBean .....</b>	<b>162</b>
5.1	Servlet 简介 .....	162
5.2	Servlet 的运行和配置 .....	163

5.2.1	Servlet 的生命周期.....	163
5.2.2	Servlet 配置.....	165
5.3	Servlet API.....	167
5.3.1	Servlet 接口.....	167
5.3.2	ServletConfig 接口.....	168
5.3.3	GenericServlet 类.....	168
5.3.4	HttpServlet 类.....	169
5.4	Servlet 编程.....	170
5.4.1	Servlet 的基本结构.....	170
5.4.2	表单处理.....	171
5.4.3	Servlet 编程示例.....	172
5.5	组件技术和 JavaBean.....	174
5.5.1	JavaBean 简介.....	174
5.5.2	创建和部署 JavaBean.....	176
5.6	JavaBean 的属性.....	177
5.7	在 JSP 中引用 JavaBean.....	179
5.8	应用示例.....	182
	本章小结.....	187
	习题 5.....	187
	上机实验 5.....	188
<b>第 6 章</b>	<b>JSP 数据库应用.....</b>	<b>189</b>
6.1	Web 数据库访问技术.....	189
6.2	数据库语言 SQL.....	191
6.2.1	SQL 概述.....	191
6.2.2	主要 SQL 语句.....	191
6.3	JDBC API.....	193
6.3.1	驱动程序接口 Driver.....	194
6.3.2	驱动程序管理器 DriverManager.....	195
6.3.3	数据库连接接口 Connection.....	195
6.3.4	语句执行接口 Statement 和 PreparedStatement.....	196
6.3.5	结果集接口 ResultSet.....	197
6.4	JDBC 数据库访问.....	198
6.4.1	加载 JDBC 驱动程序.....	198
6.4.2	创建数据库连接.....	199
6.4.3	执行 SQL 语句访问数据库.....	200
6.4.4	数据库访问结果集的处理.....	202
6.4.5	数据库操作中的事务处理.....	203
6.4.6	存储过程的调用.....	204
6.5	JSP 数据库操作.....	206
6.6	SQL 语句注入攻击与防范.....	207
6.6.1	SQL 注入攻击.....	207



6.6.2 避免 SQL 注入攻击.....	208
6.7 应用示例：课程信息查询与修改.....	208
本章小结.....	214
习题 6.....	215
上机实验 6.....	215
<b>第 7 章 JSP 实用组件.....</b>	<b>216</b>
7.1 文件操作.....	216
7.1.1 创建上传对象.....	216
7.1.2 解析上传请求.....	217
7.1.3 FileItem 接口.....	217
7.1.4 ServletFileUpload 类.....	217
7.1.5 DiskFileItemFactory 类.....	218
7.1.6 文件操作示例.....	219
7.2 JSP 动态图表.....	222
7.2.1 JFreeChart 的下载和使用.....	223
7.2.2 JFreeChart 的核心类.....	223
7.2.3 利用 JFreeChart 生成动态图表.....	224
7.2.4 动态图表应用示例.....	224
7.3 JSP 报表.....	228
7.3.1 iText 组件.....	229
7.3.2 应用 iText 组件生成报表.....	229
7.3.3 处理表格.....	231
7.3.4 处理图像.....	235
7.4 Ajax 技术.....	237
7.4.1 Ajax 简介.....	237
7.4.2 Ajax 开发模式.....	238
7.4.3 Ajax 应用示例.....	241
7.4.4 Ajax 开发需要注意的问题.....	243
本章小结.....	245
习题 7.....	245
上机实验 7.....	245
<b>第 8 章 表达式语言和标签.....</b>	<b>247</b>
8.1 EL 表达式.....	247
8.1.1 EL 表达式的语法.....	247
8.1.2 EL 表达式的运算符.....	247
8.1.3 EL 表达式中的隐含对象.....	248
8.1.4 EL 表达式中的保留字.....	249
8.2 JSTL 核心标签库.....	249
8.2.1 表达式标签.....	249
8.2.2 流程控制标签.....	251
8.2.3 循环标签.....	254

8.2.4	URL 标签 .....	256
8.3	SQL 标签库.....	258
8.4	自定义标签库 .....	260
8.4.1	自定义标签处理类.....	260
8.4.2	建立 TLD 文件.....	261
8.4.3	使用自定义标签.....	262
8.4.4	自定义标签使用范例 .....	262
	本章小结.....	264
	习题 8 .....	265
	上机实验 8.....	265
<b>第 9 章</b>	<b>Java EE 框架技术基础 .....</b>	<b>266</b>
9.1	框架技术概述 .....	266
9.1.1	MVC 模型与设计模式 .....	266
9.1.2	Struts2 框架 .....	267
9.1.3	Hibernate 框架.....	268
9.1.4	Spring 框架.....	268
9.2	Struts2 框架.....	269
9.2.1	Struts2 的下载和配置.....	270
9.2.2	Struts2 基础和 struts.xml 的基本配置 .....	270
9.2.3	Action 详解.....	272
9.2.4	值栈和 OGNL 表达式 .....	274
9.2.5	Struts2 的标签库 .....	275
9.2.6	拦截器 .....	280
	本章小结.....	283
	习题 9 .....	283
	上机实验 9.....	283
<b>第 10 章</b>	<b>JSP 综合应用实例.....</b>	<b>284</b>
10.1	留言板.....	284
10.1.1	设计目标 .....	284
10.1.2	设计实体类.....	284
10.1.3	设计数据库处理程序 .....	285
10.1.4	设计留言处理程序 .....	286
10.1.5	设计页面 .....	288
10.1.6	设计字符编码过滤器 .....	291
10.2	教务管理系统 .....	293
10.2.1	系统功能 .....	293
10.2.2	数据库设计.....	294
10.2.3	设计实体类.....	295
10.2.4	文件组织架构.....	296
10.2.5	设计数据库处理程序 .....	297
10.2.6	设计 Action 类.....	299

10.2.7 设计视图 .....	310
10.2.8 设计样式表.....	322
10.2.9 设计配置文件.....	323
本章小结.....	324
附录 A HTML 常用标记和属性 .....	325
附录 B CSS 样式表属性.....	329
附录 C JavaScript 常用对象的属性、方法、事件处理和函数 .....	331
附录 D JSP 内置对象 .....	337
参考文献 .....	340



# 第 1 章 Web 基础知识与开发运行环境

本章介绍开发 Web 程序应该必备的基础知识,包括 Web 的基本概念和工作原理、Internet 网络协议、IP 地址、域名和统一资源定位器 URL、动态网页设计技术、Web 应用程序开发工具与运行环境以及网站安全问题,为学习 Web 程序设计方法和开发技术做好准备。

## 1.1 Web 工作原理

Internet 已成为世界上最大的信息宝库,然而 Internet 上的信息资源既没有统一的目录,也没有统一的组织和系统,这些信息分布在位于世界各地的计算机系统中。人们为了充分利用 Internet 上的信息资源,迫切需要一种方便、快捷的信息浏览和查询工具,在这种情况下,Web 诞生了。

Web,全称为 World Wide Web,缩写为 WWW。Web 有许多译名,如环球网、万维网、全球信息网等。如果有一台计算机与 Internet 相连,不管通过什么方式接入 Internet,任何人都可以通过浏览器(Browser)访问处于 Internet 上任何位置的 Web 站点。但什么是 Web,目前尚无公认的准确定义。简单地说,Web 是一种体系结构,通过它可以访问分布于 Internet 主机上的链接文档。这一说法包含以下几层含义:

① Web 是 Internet 提供的一种服务。尽管这几年 Web 的迅猛发展使得有人甚至误认为 Web 就是 Internet,但事实上,Web 是基于 Internet、采用 Internet 协议的一种体系结构,因而可以访问 Internet 的每个角落。

② Web 是存储在全世界 Internet 计算机中、数量巨大的文档的集合。或者可以通俗地说,Web 是世界上最大的电子信息仓库。

③ Web 上的海量信息是由彼此关联的文档组成的,这些文档称为主页(Home Page)或页面(Page),是一种超文本(Hypertext)信息,而使其连接在一起的是超链接(Hyperlink)。由于超文本的特性,用户可以看到文本、图形、图像、视频、音频等多媒体信息,这些媒体称为超媒体(Hypermedia)。

④ Web 的内容保存在 Web 站点(Web 服务器)中,用户可通过浏览器访问 Web 站点。因此 Web 是一种基于浏览器/服务器(Browser/Server, B/S)的结构。也就是说,Web 实际上是一种全球性通信系统,通过 Internet 使计算机相互传输基于超媒体的数据信息。

⑤ Web 以一些简单的操作方式(如单击鼠标)连接全球范围的超媒体信息,因此易于使用和普及。基于 Web 开发的各种应用易于跨平台实现,开发成本较低,而且基于 Web 的应用几乎不需要培训用户。

近年来,Web 得到了迅猛的发展,如今的 Web 应用已远远超出了原先对它的设想。它不仅成为 Internet 上最普遍的应用,而且正是由于它的出现,使 Internet 普及和推广的速度大大提高了。

Web 具有以下特点：

① Web 是一种超文本信息系统。Web 的超链接使得 Web 文档不再像书本一样是固定的、线性的，可以从一个位置迅速跳转到另一个位置，从一个主题迅速跳转到另一个相关主题。

② Web 是图形化的和易于导航的。Web 之所以能够迅速流行，一个重要的原因在于它具有在一页上同时显示图形、图像和其他超媒体的性能。在 Web 之前，Internet 上的信息只有文本形式，Web 提供了将图形、图像、音频、视频信息集于一体的特性。同时，Web 非常易于导航，只需从一个链接跳转到另一个链接，就可以在各页面、各站点之间进行浏览。

③ Web 与平台无关。无论系统的软件、硬件平台是什么，都可以通过 Internet 访问 WWW。Web 对系统平台没有限制。

④ Web 是分布式的。对于 Web，没有必要把大量图形、图像、音频、视频信息都放在一起，可以将它们放在不同的站点上，只要通过超链接指向所需的站点，就可以使存放在不同物理位置上的信息实现逻辑上的一体化。对用户来说，这些信息是一体的。

⑤ Web 具有新闻性。Web 站点上的信息是动态的、经常更新的。信息的提供者可以经常对站点上的信息进行更新，所以用户（浏览者）可以得到最新的信息。

⑥ Web 是动态的、交互的。早期的 Web 页面是静态的，用户只能被动浏览。由于开发了多种 Web 动态技术，现在的用户已经能够方便地定制页面。以 ASP（Active Server Pages）、ASP.NET 和 Java 为代表的动态技术使 Web 从静态的页面变成可执行的程序，从而大大提高了 Web 的动态性和交互性。Web 的交互性还表现在它的超链接上，因为通过超链接，用户的浏览顺序和所到站点完全可由用户自行决定

Web 是一种典型的基于浏览器/服务器（Browser/Server，B/S）的体系结构。典型的 B/S 结构将计算机应用分成三个层次，即客户端浏览器层、Web 服务器层和数据库服务器层。B/S 结构有许多优点，简化了客户端的维护，所有应用逻辑都是在 Web 服务器上配置的。B/S 结构突破了传统客户—服务器（Client/Server，C/S）结构中局域网对计算机应用的限制，用户可以在任何地方登录 Web 服务器，按照用户角色执行自己的业务流程。Web 通过 HTTP 实现客户端浏览器和 Web 服务器的信息交换，其基本工作原理如图 1-1 所示。



图 1-1 Web 的基本工作原理

Web 浏览器是一种 Web 客户端程序，用户要浏览 Web 页面，必须在本地计算机上安装浏览器软件。通过在浏览器地址栏中输入 URL 资源地址，将 Web 服务器中特定的网页文件下载到客户端计算机中，并在浏览器中打开。因此，从本质讲，浏览器是一种特定格式的文档阅读器，能根据网页内容，对网页中的各种标记进行解释显示；同时，浏览器是一种程序解释机，如果网页中包含客户端脚本程序，那么浏览器将执行这些客户端脚本代码，从而增强网页的交互性和动态效果。

在 Web 系统中，Web 服务器有两个层面的含义：一是指安装了 Web 服务程序的计算机；二是指 Web 服务器程序，可以管理各种 Web 文件，并为提出 HTTP（HyperText Transfer Protocol，超文本传输协议）请求的浏览器提供 HTTP 响应。要使一台计算机成为一台 Web 服务器，需要配置服务器操作系统，如 UNIX、Windows、Linux 等网络操作系统，并且要安装

专门的信息服务器程序，如 Windows 提供的 Internet 信息服务器（Internet Information Server, IIS）。在大多数情况下，Web 服务器和浏览器处于不同的机器，但它们可以并存在同一台机器上。

Web 服务器向浏览器提供服务的过程大致如下。

<1> 用户打开计算机(客户端),启动浏览器程序(如 Netscape Navigator、Microsoft Internet Explorer 等),并在浏览器中指定一个 URL (Uniform Resource Locator, 统一资源定位器),浏览器便向该 URL 所指向的 Web 服务器发出请求。

<2> Web 服务器(也称为 HTTP 服务器)接到浏览器的请求后,把 URL 转换成页面所在服务器的文件路径名。

<3> 如果 URL 指向的是普通的 HTML (HyperText Markup Language, 超文本标记语言)文档, Web 服务器将直接把它传送给浏览器。HTML 文档中可能包含用 Java、JavaScript、ActiveX、VBScript、C#等编写的小应用程序 (Applet),服务器将它们随 HTML 文档一起传送到浏览器,在浏览器所在的机器上执行。

<4> 如果 HTML 文档中嵌有 JSP 或 ASP.NET 程序,那么 Web 服务器就运行 JSP 或 ASP.NET 程序,并将结果传至浏览器。Web 服务器运行 JSP 或 ASP.NET 程序时,还可能调用数据库服务器和其他服务器。

<5> URL 也可以指向 VRML (Virtual Reality Modeling Language) 文档。只要浏览器中配置有 VRML 插件,或者客户端上已安装 VRML 浏览器,就可以接收 Web 服务器发送的 VRML 文档。

早期的 Web 页面是静态的,用户只能被动浏览。静态页面是用纯 HTML 代码编写的,这些页面的代码保存为.html 或.htm 文件形式。后来,以 ASP、ASP.NET 和 Java 为代表的动态技术使 Web 从静态页面变成可执行的程序,从而产生了动态网页,大大提高了 Web 的动态性和交互性。利用 ASP 或 ASP.NET,服务器可以执行用户用 VBScript、JavaScript 或 C#编写的嵌入 HTML 文档中的程序。通过 ASP 或 ASP.NET 程序,Web 页面可以访问数据库,存取服务器的有关资源,使 Web 页面具有强大的交互能力。Web 的交互性还表现在它的超链接上,因为通过超链接,用户的浏览顺序和所到站点完全可由用户自行决定。

随着技术的不断发展,动态网页的实现一般采用客户端编程和服务器端编程两种程序设计方法。

客户端编程是客户端浏览器下载服务器上的程序来执行有关动态服务工作。程序员把客户端代码编写到 HTML 文件中,当用户提出对某个网页的请求时,这些客户端代码和 HTML 文件代码一起以响应方式返回提出请求的浏览器。常见的客户端编程技术有 VBScript、JavaScript、Java Applet 等。

服务器端编程是将程序员编写的代码保存在服务器上,当用户提出对某个网页的请求时,这个请求要访问的页面代码都在服务器端执行,并把执行结果以 HTML 文件代码的形式传回浏览器,这样浏览器接收的只是程序执行的结果。常见的服务器端编程技术有 PHP、JSP、ASP、ASP.NET。

## 1.2 Internet 网络协议

Internet 是由不同类型、不同规模、独立管理和运行的主机或计算机网络组成的一个全球

性特大网络。Internet 使用的网络协议是 TCP/IP 协议族，凡是接入 Internet 的计算机都必须安装和运行 TCP/IP 协议软件。

## 1.2.1 TCP/IP 协议族

TCP/IP 是一个协议族，其中最重要的是 TCP 和 IP，因此通常将这些协议简称为 TCP/IP (Transmission Control Protocol/Internet Protocol, 传输控制协议/网络协议)。

TCP/IP 把整个网络分成 4 个层次：应用层、传输层、网络层和物理链路层，都建立在硬件基础之上。图 1-2 给出了 TCP/IP 参考模型与 OSI 参考模型的对照。



图 1-2 OSI 参考模型与 TCP/IP 参考模型的对照

应用层是 TCP/IP 参考模型的最高层，向用户提供一些常用应用程序，如电子邮件服务等。应用层包括所有的高层协议，并且总是不断有新的协议加入。应用层协议主要包括：

- ✘ 网络终端协议 Telnet：用于实现互联网中的远程登录功能。
- ✘ 文件传输协议 (File Transfer Protocol, FTP)：实现互联网中交互式文件传输功能。
- ✘ 简单邮件传输协议 (Simple Mail Transfer Protocol, SMTP)：实现互联网中电子邮件的收发功能。
- ✘ 网络文件系统 (Network File System, NFS)：用于网络中不同主机间的文件系统共享。
- ✘ 域名系统 (Domain Name System, DNS)：实现网络设备域名到 IP 地址的映射。
- ✘ 超文本传输协议 (HyperText Transfer Protocol, HTTP)：用于在 Web 浏览器和服务器之间传输 Web 文档。

传输层也叫 TCP 层，主要功能是负责应用进程之间的端 - 端通信。传输层定义了两种协议：传输控制协议 TCP 和用户数据报协议 UDP。

网络层也叫 IP 层，负责处理互联网中计算机之间的通信，向传输层提供统一的数据包。它的主要功能包括 3 方面：处理来自传输层的分组发送请求；处理接收的数据包；处理互连的路径。

物理链路层的主要功能是接收 IP 层的 IP 数据报，通过网络向外发送；接收并处理从网络上传来的物理帧，抽出 IP 数据报，向 IP 发送。物理链路层是主机与网络的实际连接层。

## 1.2.2 HTTP

HTTP 是专门为 Web 设计的一种网络协议，属于 TCP/IP 参考模型中的应用层协议，位于 TCP/IP 的顶层。因此，HTTP 在设计和使用中以 TCP/IP 协议族中的其他协议为基础。例如，HTTP 要通过 DNS 进行域名与 IP 地址的转换，要建立 TCP 链接才能进行文档传输。

Web 浏览器和服务器用 HTTP 来传输 Web 文档。HTTP 基于客户端请求、服务器响应的



工作模式，其定义的事务处理由以下 4 个步骤组成：

- <1> 客户端与服务器建立连接。
- <2> 客户端向服务器提出请求。
- <3> 如果请求被接受，则服务器送回响应，在响应中包括状态码和所需的文件。
- <4> 客户端和服务器断开连接。

### 1.2.3 Telnet

Telnet 是关于远程登录的一个协议。要使用 Telnet，在用户的计算机上需要安装和运行一个名为 Telnet 的程序。在使用 Telnet 时，它又是一个命令。用户可以用 Telnet 命令使用户主机连入 Internet 上任何一台 Telnet 服务器。一般把这台被用户主机调用的服务器称为远程主机。这时候用户主机就成为该远程主机的一个终端。不管这种连接如何复杂，在用户的键盘上输入 Telnet 子命令后，总能在远程主机上得到服务响应，并把结果送到用户的屏幕上。

Internet 上存在成千上万的各种主机（大、中、小型机）或服务器。用户可以通过 Telnet 连入某个主机并成为该主机的终端，进而可访问所需的各种信息，或运行远程主机上的程序来求解各种复杂的问题，一切都是在远程主机上快速执行（而不是将程序调回到用户主机上执行）后再从远程主机返回服务的结果。用户还可以利用 Telnet 连接 Internet 的各种服务器，如 Archie、Gopher、Wais、WWW 及其他服务器，如某图书馆的资料文献服务器等。

用户使用远程主机有两种情况：一种是要求用户有账号才能登录的；另一种是开放的，用户不需拥有自己的账号，即不用口令和用户名就能登录。Internet 上有许多这样的为公众开放的 Telnet 远程服务。

### 1.2.4 FTP

Telnet 让用户主机能以终端方式共享 Internet 上各类主机的资源，却不能把远程主机上的文件复制到用户主机上。有了 FTP 的帮助就能使 Internet 上两台主机间互传（复制）文件。FTP 有一套独立通用的命令（子命令），命令风格与 DOS 命令相似，如 dir 显示目录/文件。实际使用 FTP 时往往会碰到两个难点。第一，并不知道想要复制的文件在哪个 FTP 服务器中，在成千上万个 FTP 服务器中一个个地寻找某个文件犹如大海捞针，此时需要借助某些工具，如 Internet 的 Archie 服务器。第二，要明确传输的文件是什么类型，即确定传输的是二进制文件还是 ASCII 文件。如果文件传输类型不对，复制得到的文件常常是无用的文件。

FTP 既是一种文件传输协议，也是一种服务，提供这种服务的设施叫做 FTP 服务器。有一种 FTP 服务器称为匿名 FTP 服务器，用户不需要拥有口令和用户名就能与匿名 FTP 服务器实现连接并复制文件。Internet 上有许多这样的、为公众开放的匿名 FTP 服务器。

## 1.3 IP 地址、域名和 URL

### 1.3.1 IP 地址

IP 地址是识别 Internet 中主机及网络设备的唯一标识。每个 IP 地址通常分为网络地址和主机地址两部分，其长度为 4 B（字节），共 32 位，由 4 个用“.”分隔的十进制数组成，每

个数不大于 255，如 202.119.106.253。

IP 地址可分成 5 类，其中常用的是如下 3 类。

**A 类：**用于规模很大、主机数目非常多的网络。A 类地址的最高位为 0，接下来的 7 位为网络地址，其余 24 位为主机地址。A 类地址允许组成 126 个网络，每个网络可包含 1700 万台主机。

**B 类：**用于中型和大型网络。B 类地址最高两位为 10，接下来 14 位为网络地址，其余 16 位为主机地址。B 类地址允许组成 16384 个网络，每个网络可包含 65000 台主机。

**C 类：**用于小型本地网络（LAN）。C 类地址最高 3 位为 110，接下来的 21 位为网络地址，其余 8 位为主机地址。

注意，主机地址的末字节不能取 0 和 255 两个数。

## 1.3.2 域名

IP 地址是连网计算机的地址标识，但对大多数人来说，记住很多计算机的 IP 地址并不是一件容易的事，所以 TCP/IP 中提供了域名服务系统（DNS），允许为主机分配字符名称，即域名。在网络通信时由 DNS 自动实现域名与 IP 地址的转换。例如，南京师范大学 Web 服务器的域名为 www.njnu.edu.cn。

Internet 中的域名采用分级命名，其基本结构如下：

计算机名.三级域名.二级域名.顶级域名

域名的结构与管理方式如下。

首先，DNS 将整个 Internet 划分成多个域，称为顶级域，并为每个顶级域规定了国际通用的域名。顶级域名采用两种划分模式，即组织模式和地理模式。有 7 个域对应于组织模式，其余的域对应地理模式，如 cn 代表中国、us 代表美国、jp 代表日本等。

7 个组织模式的顶级域名分配如下：com，商业组织；edu，教育机构；gov，政府部门；mil，军事部门；net，网络中心；org，前述以外的组织；int，国际组织。

其次，Internet 的域名管理机构将顶级域的管理权分派给指定的管理机构，各管理机构对其管理的域继续进行划分，即划分成二级域，并将二级域的管理权授予其下属的管理机构，以此类推，便形成了树形域名结构。管理机构是逐级授权的，所以最终的域名都得到了 Internet 的承认，成为 Internet 中的正式名字。

## 1.3.3 URL

WWW 信息分布在全球，要找到所需信息必须有一种说明该信息存放在哪台计算机的哪个路径下的定位信息。统一资源定位器 URL 就是用来确定某信息位置的方法。

URL 的概念实际上并不复杂，就像指定一个人要说明他的国别、地区、城镇、街道、门牌号一样，URL 指定 Internet 资源位于哪台计算机的哪个目录中。URL 通过定义资源位置的抽象标识来定位网络资源，其格式如下：

<信息服务类型>://<信息资源地址>/<文件路径>

<信息服务类型>是指 Internet 的协议名，包括 ftp（文件传输服务）、http（超文本传输服务）、gopher（Gopher 服务）、mailto（电子邮件地址）、telnet（远程登录服务）、news（提供网络新闻服务）和 wais（提供检索数据库信息服务）。

<信息资源地址>指定一个网络主机的域名或 IP 地址。在有些情况下，主机域名后还要加上端口号，域名与端口号之间用“:”隔开。这里的端口是指操作系统用来辨认特定信息服务的软件端口。一般情况下，服务器程序采用标准的保留端口号，因此用户在 URL 输入中可以省略它们。以下是一些 URL 的例子：

```
http: //www.njnu.edu.cn
http: //www.whitehouse.gov
telnet: //odysseus.circe.com:70
ftp: //ftp.w3.org/pub/www/doc
gopher: //gopher.internet.com
news: //comp.sys.novell
wais: //quake.think.com/directory-of-servers
```

## 1.4 动态网页设计技术简介

早期的 Web 页面是静态的，静态页面是用纯 HTML 编写的。后来，以 PHP、ASP 和 JSP 为代表的动态技术使 Web 从静态页面变成可执行的程序，从而产生了动态网页，大大提高了 Web 的动态性和交互性。

早期，动态网页设计主要使用 CGI（Common Gateway Interface，公共网关接口）技术，可以使用不同的语言编写合适的 CGI 程序，如 Visual Basic、C/C++等。虽然 CGI 技术已经发展成熟且功能强大，但由于编程困难、效率较低、修改复杂等缺陷，因此 CGI 技术已被淘汰。

ASP 为动态服务器网页（Active Server Pages）的简称，是一种功能强大的服务器端脚本编程环境。1996 年底，微软公司推出了 ASP 1.0，内含于 IIS 3.0（Microsoft Internet Information Server 3.0）中。1998 年，微软推出了 ASP 2.0。2000 年，微软公司发布了 Windows 2000 操作系统，其中包含 IIS 5.0 和 ASP 3.0。ASP 最大的好处是可以包含 HTML 标签，也可以直接存取数据库以及使用 ActiveX 控件，采用脚本语言 VBScript、JavaScript 作为开发语言，利用 HTML 网页、ASP 指令和 ActiveX 组件建立动态、交互的 Web 服务器应用程序。由于 ASP.NET 的出现，与 ASP 相比，ASP.NET 在功能、效率等方面都具有优势，因此目前 ASP 基本不再使用。

目前比较受关注的动态网页设计技术主要有 PHP、JSP、ASP.NET。

### 1.4.1 PHP

PHP（Hypertext Preprocessor，超文本预处理器）于 1994 年由 Rasmus Lerdorf 创建，刚开始是 Rasmus Lerdorf 为了维护个人网页而制作的一个简单的用 Perl 语言编写的程序。1995 年发布 PHP 1.0，之后发布了多个版本，2008 年发布了 PHP 5。PHP 是一种跨平台的服务器端嵌入式脚本语言，是一种易于学习和使用的服务器端脚本语言，嵌入 HTML 文件，大量借用 C、Java 和 Perl 语言的语法，并耦合 PHP 本身的特性，形成了自己的独特风格。PHP 支持目前绝大多数的数据库，Web 开发者使用 PHP 能够快速地写出生成动态网页的脚本代码，只需很少的编程知识就能建立一个真正交互的 Web 站点。PHP 是完全免费的，可以从 PHP 官方网站（<http://www.php.net>）自由下载，可以不受限制地获得源代码，并可加入自己需要的功能。

PHP 具有如下特点:

- ① 支持多种系统平台, 包括 Windows、UNIX 和 Linux。
- ② 强大的数据库操作功能。PHP 提供丰富的数据库操作函数, 为各种流行数据库, 包括 Linux 平台的 PostgreSQL、MySQL、Solid 及 Oracle, Windows 平台的 SQL Server, 都设计了专门的函数, 使操作这些数据库十分方便。
- ③ 易于与现有的网页融合。与 ASP、JSP 一样, PHP 也可结合 HTML 使用; 与 HTML 具有非常好的兼容性, 使用者可以直接在脚本代码中加入 HTML 标记, 或者在 HTML 标记中加入脚本代码, 从而更好地实现页面控制, 提供更加丰富的功能。
- ④ 具有丰富的功能。PHP 提供结构化特性、面向对象设计、数据库处理、网络接口使用及安全编码机制等全面的功能。
- ⑤ 可移植性好。通过 PHP, 只需要进行很少的修改就可将整个网站从一个平台移植到另一个平台上, 如从 Windows 平台移植到 UNIX 平台。

## 1.4.2 JSP

JSP (Java Server Pages) 是 Sun 公司于 1999 年 6 月推出的网站开发语言, 基于 Java Servlet 及整个 Java 体系的 Web 开发技术, 可以建立先进、安全和跨平台的动态网站。JSP 完全解决了目前 ASP、PHP 的一个通病——脚本级执行。

JSP 与 ASP 在技术方面有许多相似之处。两者都是为实现 Web 动态交互网页制作而提供的技术支持环境, 都能帮助程序开发人员实现应用程序的编制与自带组件的网页设计, 都能替代 CGI 使网站建设与发展变得简单又快捷。由于它们来源于不同的技术规范, 因而其实现的基础不同, 即对 Web 服务器平台的要求不同。基于 JSP 技术的应用程序比基于 ASP 的应用程序更易于维护和管理。

JSP 技术具有以下优点。

① 内容生成与显示分离。使用 JSP 技术, Web 页面开发人员可以使用 HTML 或 XML 标记来设计页面, 来生成页面上的动态内容 (内容是动态的, 但可根据用户请求而变化)。动态生成的内容被封装在标记和 JavaBean 组件中, 并且捆绑在小脚本中, 所有的脚本在服务器端运行。在服务器端, 使用 JSP 引擎来解释 JSP 标记和小脚本, 生成所请求的内容, 并将结果以 HTML 或 XML 页面形式发送回浏览器。这有助于作者保护自己的代码, 又能保证任何基于 HTML 的 Web 浏览器的完全可用性。

② 可重用的组件。绝大多数 JSP 页面依赖可重用的、跨平台的组件来执行应用程序所要求的复杂处理, 如使用 JavaBean 或 Enterprise JavaBean 组件。开发人员可以共享各种组件, 这种基于组件的方法提高了系统的开发效率。

③ 采用标记简化页面开发。JSP 技术使用 XML 标记封装了许多与动态内容生成相关的功能, 页面开发人员使用这些标记就可以进行设计, 而不必进行编程。

④ 适应更广泛的平台。JSP+JavaBean 可以在大多数 Web 服务器平台下使用。著名的 Web 服务器 Apache 能够很好地支持 JSP, 由于 Apache 广泛应用在 Windows、UNIX 和 Linux 操作系统上, 因此 JSP 有更广泛的运行平台。

⑤ 易于连接数据库。Java 中连接数据库的技术是 JDBC (Java DataBase Connectivity)。很多数据库系统, 如 Oracle、Sybase、Microsoft SQL Server、Access 等, 都带有 JDBC 驱动程序, Java 程序通过 JDBC 驱动程序与数据库相连, 执行查询数据、提取数据等操作。另外,

Sun 公司开发了 JDBC-ODBC bridge, Java 程序就可以访问带有 ODBC 驱动程序的数据库了。

### 1.4.3 ASP.NET

ASP.NET 又称为 ASP+, 是微软公司于 2001 年推出的一种用于创建 Web 应用程序的编程模型。它在结构上几乎完全是基于组件和模块化的。Web 应用程序的开发人员使用这个开发环境可以实现更加模块化、功能更强大的应用程序。

ASP.NET 具备开发网站应用程序的一切解决方案, 包括验证、缓存、状态管理、调试和部署等全部功能。在代码编写方面特色是将页面逻辑和业务逻辑分开, 分离程序代码与显示的内容, 让丰富多彩的网页更容易编写。在 ASP.NET 中, 所有程序保存在服务器端, 由服务器编译执行。当第一次执行一个程序时进行编译, 当再次执行这个程序时, 就在服务器端直接执行它的已编译好的程序代码, 因而 ASP.NET 程序的执行速度有较大的提高。对于实现同样功能的程序, ASP.NET 使用的代码量比 ASP 要小得多。

从深层次说, ASP.NET 与 ASP 的主要区别体现在以下 3 方面。

① 效率。ASP 是一个脚本编程环境, 只能用 VBScript 或 JavaScript 这样的非模块化语言来编写。当 ASP 程序完成后, 在每次请求时都要解释执行。这意味着, ASP 在使用其他语言编写大量组件的时候会遇到困难, 并且无法实现对操作系统的底层操作。ASP.NET 则建立在 .NET 框架上, 可以使用 Visual Basic、C#、J# 这样的模块化程序设计语言, 并且它在第一次执行时进行编译, 之后的执行不需要重新编译就可以直接运行, 所以速度和效率比 ASP 提高很多。

② 可重用性。在编写 ASP 应用程序时, ASP 代码与 HTML 代码混合在一起。只要需要, 就可以在任意的位置插入一段代码来实现特定的功能。这种方法表面上看起来很方便, 但实际上会产生大量烦琐的页面, 很难让人读懂, 导致代码维护困难。ASP.NET 则可以实现代码与内容的完全分离, 使得维护更方便。

③ 代码量。ASP 对所有要实现的功能均需要通过编写代码来实现。例如, 为了保证一个用户数据提交页面的友好性, 当用户输入错误时应显示错误的位置, 并尽量把用户原来的输入显示在控件中。对于这样的应用, ASP 需要程序员编写大量的代码才能实现。在 ASP.NET 中, 程序员只要预先说明, 就可以自动实现这样的功能。所以相对来说, 要实现同样的功能, 使用 ASP.NET 比使用 ASP 的代码量要小得多。

## 1.5 Web 应用程序开发与运行环境

Web 应用系统需要 Web 浏览器、Web 服务器、开发工具包和数据库。

用 JSP 开发 Web 应用系统, 对浏览器的要求并不是很高, 任何支持 HTML 的浏览器都可以。

Web 服务器是运行及发布 Web 应用的大容器, 只有将开发的 Web 项目放置到该容器中, 才能使网络中的所有用户通过浏览器进行访问。开发 JSP 应用采用的服务器主要是 Servlet 兼容的 Web 服务器, 比较常用的有 WebLogic、WebSphere 和 Tomcat 等。Tomcat 服务器最为流行, 它是 Apache-Jakarta 开源项目中的一个子项目, 是一个小型的、轻量级的、支持 JSP 和 Servlet 技术的 Web 服务器, 已经成为学习开发 JSP 应用的首选。本书示例均在 Tomcat 上调

试运行。

Eclipse 是一个基于 Java 的、开放源码的、可扩展的应用开发平台，为编程人员提供了一流的 Java 集成开发环境（Integrated Development Environment, IDE）。Eclipse 是一个可以用于构建集成 Web 和应用程序开发工具的平台，因此本书采用 Eclipse 作为 IDE 开发工具。

任何项目的开发应用几乎都需要使用数据库，用来存储需要的信息、数据。JSP 支持所有流行的数据库，如 Oracle、SQL Server、Access 或 MySQL 数据库，本书示例均在 SQL Server 或 MySQL 上调试运行。

## 1.5.1 网站架构

Web 应用程序是一种使用 HTTP 作为核心通信协议、通过互联网让 Web 浏览器和服务器通信的计算机程序，因此一个网站需要考虑硬件架构和软件架构。硬件架构包括服务器、路由器、交换机、网络等设施。服务器可能根据客户需求分析、网站的目标群体、访问流量、应用大小、复杂程度又分为图片服务器、页面服务器、数据库服务器、应用服务器、日志服务器等。

软件架构包括操作系统、应用服务器、数据库、域名、开发工具以及系统框架的选择。

一个简单的 Web 网站仅需要一台 Windows 计算机，安装 JDK 作为 Java 运行环境，安装 Tomcat 作为 Web 服务器和 MySQL 作为数据库服务器，将应用部署在 Tomcat 服务器上即可。当然，如果互联网上的 Web 用户要能访问这个网站，就需要为这个网站申请一个 IP 地址或域名。

## 1.5.2 JDK 的下载、安装与配置

JDK（Java Develop Kit，Java 开发工具包）包括运行 Java 程序所需的 JRE 环境及开发过程中常用的库文件，是 Tomcat 作为 Web 服务器必须具备的。

在使用 JSP 设计页面前，首先必须安装 JDK，目前 JDK 的最新版本为 JDK 11，下载地址为 <http://www.oracle.com/technetwork/java/javase/downloads/>，有解压版和安装版两种，建议初学者选择安装版。选择合适的版本后，安装过程比较简单，只需要按照默认设置即可安装成功。如果是解压版，必须配置环境变量。

用 Eclipse 作为开发环境一般不需要对 JDK 进行配置。如果手工编译 Java 程序 JavaBean、Servlet 等，则需要配置 Java 开发环境 path 和 classpath。

配置过程如下：

右击“我的电脑”，在弹出的快捷菜单中选择“属性→高级→环境变量”，在“系统环境变量”中，新建变量名“JAVA\_HOME”，值为 JDK 的安装路径，一般是“C:\Program Files\Java\jdk1.8.0\_91”。

然后在“系统环境变量”中编辑或新建变量名“path”及“classpath”。其中，path 的值为“.;%JAVA\_HOME%\bin\;”，classpath 的值为“.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\dt.jar; %JAVA\_HOME%\lib\tools.jar”。

在 Windows 中打开命令行窗口，输入 javac，若出现如图 1-3 所示的界面，即表明 JDK 安装、配置成功。

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\yinhx>javac
用法: javac <options> <source files>
其中, 可能的选项包括:
-g 生成所有调试信息
-g:none 不生成任何调试信息
-g:(lines,vars,source) 只生成某些调试信息
-nowarn 不生成任何警告
-verbose 输出有关编译器正在执行的操作的消息
-deprecation 输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径> 指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 覆盖引导类文件的位置
-extdirs <目录> 覆盖所安装扩展的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-proc:(none,only) 控制是否执行注释处理和/或编译。
-processor <class1>[,<class2>,<class3>...] 要运行的注释处理程序的名称; 绕过默认
的搜索进程
-processorpath <路径> 指定查找注释处理程序的位置
-parameters 生成元数据以用于方法参数的反射
-d <目录> 指定放置生成的类文件的位置
-s <目录> 指定放置生成的源文件的位置
-h <目录> 指定放置生成的本机标头文件的位置
-implicit:(none,class) 指定是否为隐式引用文件生成类文件

```

图 1-3 javac 使用帮助

### 1.5.3 Tomcat 的下载、安装与配置

Tomcat 是 Apache 软件基金会 (Apache Software Foundation) 的 Jakarta 项目中的一个核心项目, 由 Apache 其他一些公司及个人共同开发而成, 是一个小型的、轻量级的、支持 JSP 和 Servlet 技术的 Web 服务器, 而且是一个免费的开放源代码的 Web 应用服务器。Tomcat 已经成为学习开发 JSP 应用的首选 Web 应用服务器。

首先到网站 <https://tomcat.apache.org/> 下载 Tomcat, 目前最新版本为 Tomcat 9.0, 有解压版和安装版两种, 建议初学者选择安装版。选择合适的版本后, 安装过程比较简单, 只需要按照默认设置即可安装成功。

启动 Tomcat, 在浏览器中输入 <http://localhost:8080>, 若出现如图 1-4 所示的界面, 即表明 Tomcat 安装、配置成功。

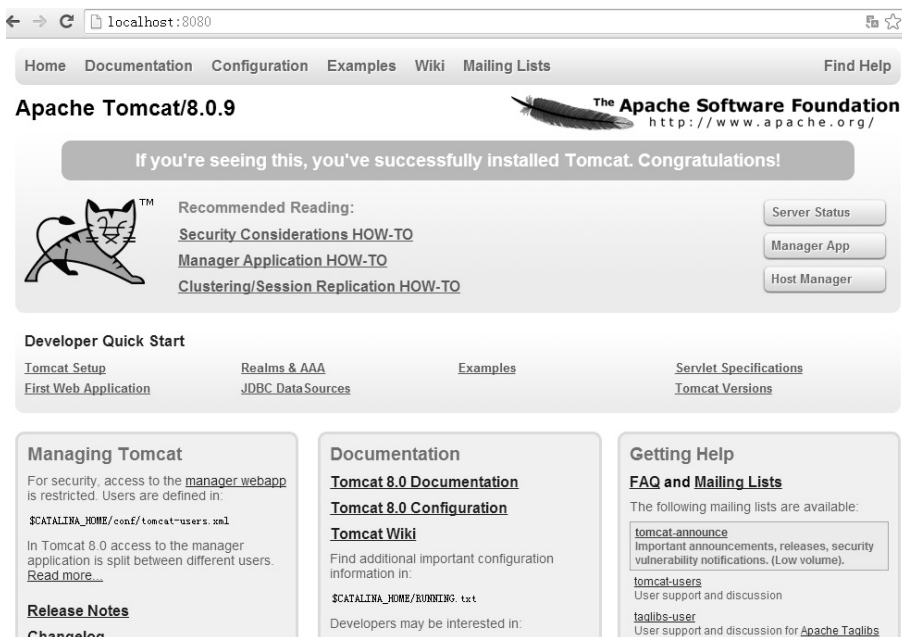


图 1-4 Tomcat 启动界面

## 1.5.4 Eclipse 的下载、安装与配置

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。实际上，Eclipse 只是一个框架和一组服务，用于通过插件组件构建开发环境。现在的 Eclipse 附带了一个标准的插件集，为编程人员提供了一流的 Java 集成开发环境 (Integrated Development Environment, IDE)，是一个可以用于构建集成 Web 和应用程序开发工具的平台。

打开网页 <https://www.eclipse.org/downloads/> 下载 Eclipse，如 eclipse-java-photon-R-win32-x86\_64.zip，解压后不需配置即可使用。

启动 Eclipse，在解压后的 eclipse 目录中，双击 eclipse.exe，设置 workspace 路径，然后进入如图 1-5 所示的工作界面。



图 1-5 Eclipse 工作界面

## 1.6 简单的 Web 应用程序示例

JDK、Tomcat、Eclipse 安装成功后，就可以开发 Web 应用程序了。注意，在用 Eclipse 运行 JSP 程序前，应该将 Tomcat 停止运行，因为 Eclipse 自身内置了一个 Tomcat，否则会引起 8080 端口冲突。

### 1. 创建第一个 Web 项目 Hello

创建一个 Web 项目的步骤如下。

<1> 进入 Eclipse 工作界面，选择 File 菜单的 New 命令，选择 Dynamic Web Project，创建一个项目名，项目名为 ch\_1，其他用默认值，如图 1-6 所示。

<2> 在项目资源管理器 Project Explorer 中右击工程名 ch\_1，在弹出的快捷菜单中选择“New→JSP File”，新建一个 JSP 文件 index.jsp，路径采用默认值 WebContent。这时系统会提供一个 JSP 模板。

<3> 在页面模板中，输入如下代码：



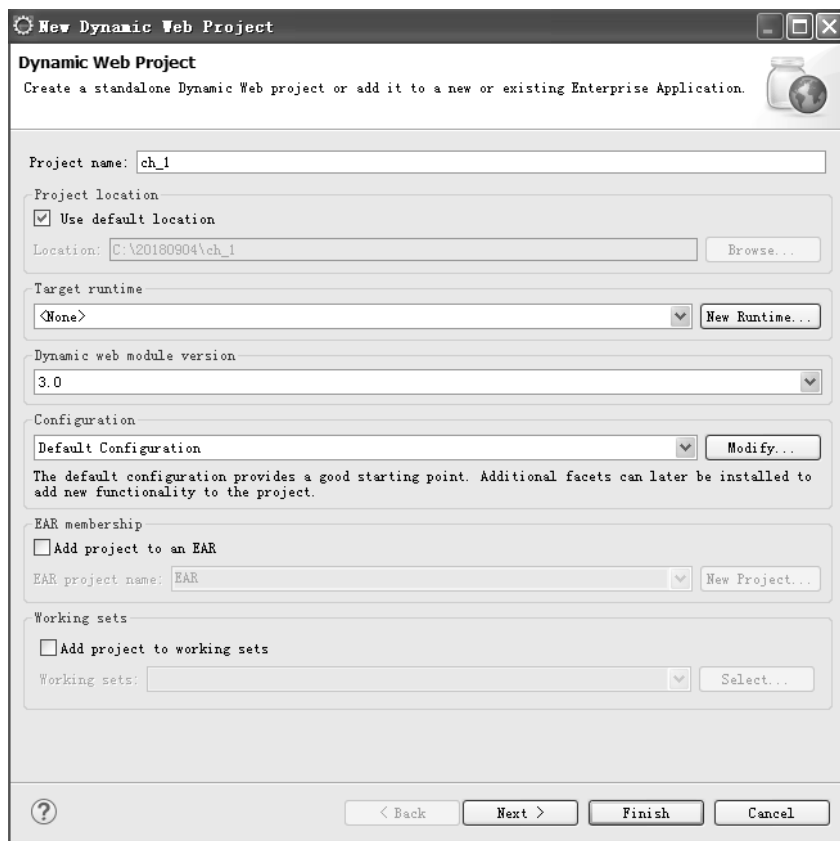


图 1-6 创建 Dynamic Web Project 项目

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>这是标题</title></head>
<body>
<h1>你好，这是第一个 JSP 页面</h1>
</body></html>
```

<4> 在代码编辑窗口单击右键，在弹出的快捷菜单中选择“Run As→Run on Server”，在出现的 Run on Server 界面中选择 Tomcat 运行程序。运行结果如图 1-7 所示。



图 1-7 运行结果

<5> 在 Eclipse 中第一次运行时要求配置 Tomcat 的路径。在 Run on Server 界面中选择 Tomcat 后，在 Tomcat installation directory 中选择 Tomcat 的安装目录，本书是 C:\Program Files\Apache Software Foundation\Tomcat 8.0，如图 1-8 所示。

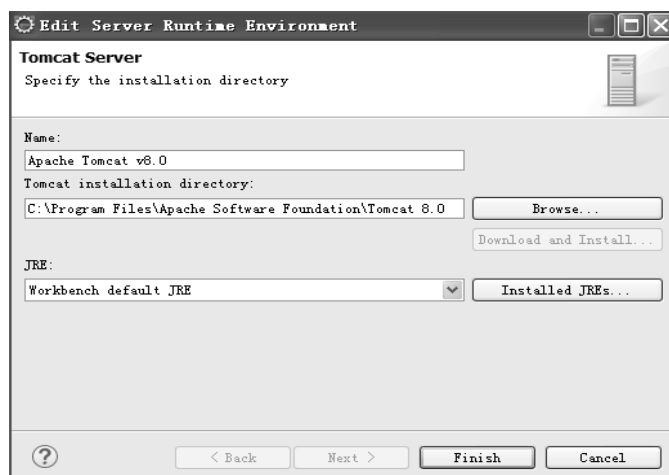


图 1-8 配置 Tomcat 路径

## 2. 目录结构

<1> 在 Eclipse 开发环境的项目名 ch\_1 下面，要保证创建的文件都在 WebContent 中，否则程序中跳转时或引用资源（如图片、CSS、JSP 文件等）需要指明所在路径。

<2> 项目中的 Java 程序在 src 子目录中。

<3> 编译后的字节码在 build 子目录中的 classes 子目录中。

<4> WebContent 下面有一个名为 WEB-INF 的子目录，注意必须大写。在 WEB-INF 目录中有一个 lib 子目录，这些都是新建项目时自动创建的，如图 1-9 所示。Web.xml 文件必须位于 WEB-INF 目录中，项目需要的 JAR 包必须复制到 lib 子目录中（如 SQL Server 数据库包 sqljdbc4.jar、Oracle 数据库包 ojdbc6.jar、标签文件 javax.servlet.jsp.jstl-1.2.1.jar 等）。

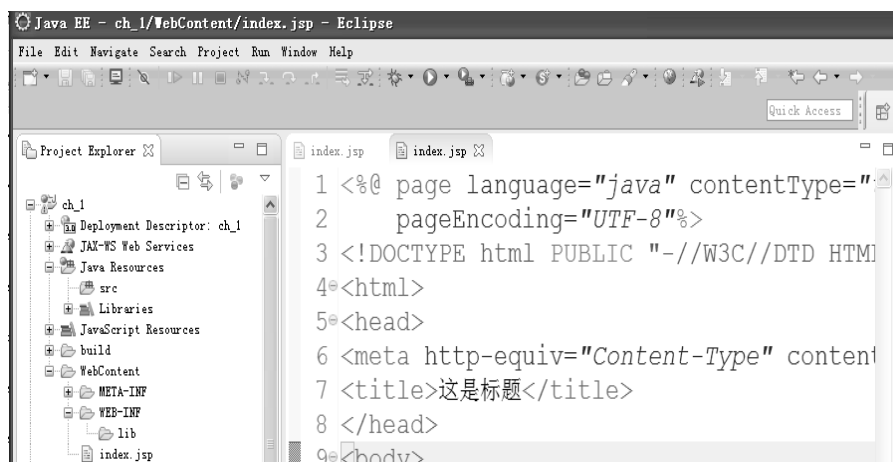


图 1-9 目录结构

## 3. 应用程序部署

在 Eclipse 中项目成功开发后，需要发布到 Web 服务器上。

最简单的就是在 Tomcat 安装目录中的 webapps 子目录中新建一个子目录，然后将 Eclipse 项目中的 WebContent 中的文件包括目录全部复制到新建的子目录中，并将 Eclipse 项目中 build 子目录中的 classes 子目录复制到新建的子目录中的 WEB-INF 中。

启动 Tomcat，打开浏览器，输入 [http://localhost:8080/ch\\_1/index.jsp](http://localhost:8080/ch_1/index.jsp)，即可打开 index.jsp

网页，其中 `ch_1` 是在 `webapps` 中新建的子目录名，如果要打开的文件是 `index.jsp`，则可以省略不输入，运行结果见图 1-7。

## 1.7 网站安全问题

网站安全是一个非常重要的问题。我们要采取有效措施保护网站安全，要采取一系列的防御措施防止网站受到入侵者对其进行挂马、篡改网页等。网站设计者更多地考虑满足用户应用需求，很少考虑网站应用开发过程中所存在的漏洞，很多网站设计开发者、网站维护人员对网站攻防技术了解甚少；在正常使用过程中，即便存在安全漏洞，使用者也并不会察觉。

### 1.7.1 网站攻击手段

#### 1. SQL 注入

对于与后台数据库产生交互的网页，如果没有对用户输入数据的合法性进行全面的判断，就会使应用程序存在安全隐患。攻击者可以在提交正常数据的 URL 或者表单输入框中提交一段精心构造的数据库查询代码，使后台应用执行攻击的 SQL 代码，攻击者根据程序返回的结果，获得某些他想要得知的敏感数据，如管理员密码、保密商业资料等。

#### 2. 跨站脚本攻击

由于网页可以包含由服务器生成的并且由客户机浏览器解释的文本和 HTML 标记，如果不可信的内容被引入动态页面中，则无论是网站还是客户机，都没有足够的信息识别这种情况并采取保护措施。攻击者如果知道某一网站上的应用程序接收跨站点脚本的提交，就可以在网提交可以完成攻击的脚本，如 JavaScript、VBScript、ActiveX、HTML 或 Flash 等内容，普通用户一旦点击了网页上这些攻击者提交的脚本，就会在用户客户机上执行，完成从截获账户、更改用户设置、窃取和篡改 cookie 到虚假广告在内的种种攻击行为。

随着攻击向应用层发展，传统网络安全设备不能有效的解决目前的安全威胁，网络中的应用部署面临的安全问题必须通过一种全新设计的高性能防护应用层攻击的安全防火墙——应用防火墙来解决。应用防火墙通过执行应用会话内部的请求来处理应用层。应用防火墙专门保护 Web 应用通信流和所有相关的应用资源免受利用 Web 协议发动的攻击。应用防火墙可以阻止将应用行为用于恶意目的的浏览器和 HTTP 攻击。这些攻击包括利用特殊字符或通配符修改数据的数据攻击，设法得到命令串或逻辑语句的逻辑内容攻击，以及以账户、文件或主机为主要目标的目标攻击。

#### 3. DNS 攻击

黑客利用常见的洪水攻击，阻击 DNS 服务器，从而达到域名解析失败，IP 地址被转向，导致网站服务器无法访问。

### 1.7.2 网站的保护与安全措施

#### 1. 安全配置

关闭不必要的服务，安装操作系统的最新补丁，将服务升级到最新版本并安装所有补丁，根据服务提供者的安全建议进行配置，这些措施将有助于服务器本身的安全。

## 2. 防火墙技术

防火墙是指一个由软件和硬件设备组合而成、在内部网和外部网之间、专用网与公网之间的边界上构造的保护屏障，保护内部网免受非法用户的侵入，防火墙主要由服务访问规则、验证工具、包过滤和应用网关 4 部分组成，是一个位于计算机与所连接的网络之间的软件或硬件。安装必要的防火墙可以阻止各种扫描工具的试探与信息收集，甚至阻止来自某些特定 IP 地址的机器链接，给服务器增加一个防护层。

## 3. 漏洞扫描

现在很多网站都存在 SQL 注入漏洞、上传漏洞等漏洞，黑客可以通过这些漏洞进行 SQL 注入进行攻击，通过上传漏洞进行木马上传等。所以网站安全检测的重要一步就是网站的漏洞检测。有些在线的网站漏洞检测工具可以免费进行漏洞扫描和网站安全检测。

## 4. 网站木马检测

网站被挂马是非常普遍的事情，也是最头疼的一件事。所以在网站安全检测中，网站是否被挂木马是重要的指标。最简单的检测网站是否有挂木马的方法是，可以直接提交 URL 到杀毒软件在线安全中心进行木马检测。如有网址被挂木马，则要暂时关闭网站，及时清除木马或木马链接的页面地址。

## 5. 入侵检测

利用入侵检测系统进行实时监控，发现正在进行的攻击行为和攻击前的试探行为，记录黑客的来源和攻击的方法步骤。

## 6. 优化代码

优化网站代码，避免 SQL 注入等攻击手段，检查代码中可能出现的漏洞，经常对代码进行测试和维护。

## 7. 建立网络信息安全管理体

建立网络信息安全管理体可以强化员工的信息安全意识，规范组织信息安全行为，在信息系统受到侵袭时，确保业务持续开展并将损失降到最低程度。

网络信息安全管理体一般包括四方面：第一是总体方针；第二是安全管理组织体系；第三是涵盖物理、网络、系统、应用、数据等方面的统一安全策略，分别从物理安全、网络安全、系统安全、应用安全、数据安全、病毒防护、安全教育、应急恢复、口令管理、安全审计、系统开发、第三方安全等方面提出了规范的安全策略要求；第四是可操作的安全管理制度、操作规范和流程。

在建网站的时候如何保证网站的安全性呢？下面进行简单介绍。

### (1) 开源程序的安全性保障

开源程序的源码是公开的，如果不做一些设置很容易被不法分子利用并攻击，使用开源建站必须关注系统的漏洞问题，关注开源系统的升级和补丁，及时把漏洞补上。

### (2) 网站的账号信息及地址的修改

开源程序的网站后台地址都是有规律的，在建站完成后一定要把后台地址进行修改，这样会增加被攻击性的难度，同时设置验证码和账号密码的难度。

### (3) 信息更新的安全性

网站建立以后，后期往往会进行维护，有时会通过 FTP 或 SSH 等工具连接到网站服务器

的文件目录，把修改后的源码文件进行上传，在进行网站相关文件上传时，一定要事先检查文件的安全性，对类似 JS、可执行文件 EXE、可执行脚本 SH 等必须加以检查，防止恶意文件上传到网站的服务器目录。

#### （4）服务器的安全性

网站的服务器安全是网站运维人员的职责，但是很多网站没有人负责服务器的安全维护。一些相关的服务器安全设置是必须要做的，包括服务器的防火墙需要正常开启、服务器的登录账号和密码的强度必须做好、服务器的故障告警提醒等，其中服务器的故障告警指的是当服务器出现故障的时候，站长需要能够短时间内收到告警提醒，以便在网站出现故障的时候可以短时间内尽快恢复。

#### （5）网站日志的分析检查

网站的访问日志可以在服务器和网站的后台查看，其中服务器的日志可以在 Web 服务器相关日志文件中查看，如 apache、tomcat 等；建议在网站的后台设置相关的访问记录功能，以便于在网站出现安全问题时可以快速排查到原因，如可以在网站的后台记录用户访问的 IP 来源、访问次数、停留时间、访问的页面等。

## 本章小结

本章主要介绍了 Web 编程的基础知识，包括 Web 的基本概念和工作原理、Internet 网络协议、IP 地址、域名和 URL、ASP、ASP.NET、PHP、JSP 等动态网页设计技术。

Internet 是由不同类型、不同规模、独立管理和运行的主机或计算机网络组成的一个全球性特大网络。Internet 使用的网络协议是 TCP/IP，凡是接入 Internet 的计算机都必须安装和运行 TCP/IP 软件。TCP/IP 是一个协议族，其应用层主要有超文本传输协议 HTTP、远程登录协议 Telnet、文件传输协议 FTP 和域名服务系统 DNS 等。

Web 是一种基于 B/S 模式、采用 Internet 协议的体系结构，是一种基于 Internet 的超文本信息系统。早期的 Web 页面是静态的，静态页面是用纯 HTML 编写的。后来，以 PHP、ASP 和 JSP 为代表的动态技术使 Web 从静态页面变成可执行的程序，从而产生了动态网页，大大提高了 Web 的动态性和交互性。

动态网页的实现一般采用客户端编程和服务器端编程两种程序设计方法。客户端编程是客户端浏览器下载服务器的程序来执行有关动态服务工作。常见的客户端编程技术有 VBScript、JavaScript、Java Applet 等。服务器端编程是将程序员编写的代码保存在服务器上，当用户提出对某个网页的请求时，这个请求要访问的页面代码都在服务器端执行，并把执行结果以 HTML 文件的形式传回浏览器。常见的服务器端编程技术有 PHP、JSP、ASP 和 ASP.NET。

IP 地址是识别 Internet 中主机及网络设备的唯一标识。但对大多数人来说，记住很多计算机的 IP 地址并不是一件容易的事，所以产生了域名服务系统 DNS，允许为主机分配字符名称，即域名。在网络通信时，由 DNS 自动实现域名与 IP 地址的转换。WWW 信息分布在全球，要找到所需信息必须有一种说明该信息存放在哪台计算机的哪个路径下的定位信息。统一资源定位器 URL 是用来确定某信息位置的方法。

网站安全是一个非常重要的问题。有的网站防御措施过于落后，常常受到黑客的攻击。黑客的攻击手段包括 SQL 注入、跨站脚本攻击、DNS 攻击等。为了防止黑客的攻击，可以采

取安全配置、防火墙、漏洞扫描、木马检测、入侵检测、代码优化等一系列网站保护与安全措施，同时建立网络信息安全管理体系统。

## 习 题 1

- 1.1 简述 Web 的特点及应用。
- 1.2 描述 Web 服务器向浏览器提供服务的基本过程。
- 1.3 请列举主要的动态网页设计技术。
- 1.4 TCP/IP 参考模型分成哪几层？每层的主要功能是什么？
- 1.5 解释下列网络协议的作用：  
Telnet    SMTP    FTP    DNS    HTTP    TCP    IP
- 1.6 名词解释：  
域名    IP 地址    URL    Web    PHP    JSP    ASP    ASP.NET
- 1.7 攻击网站主要有哪些手段？网站的保护与安全措施有哪些？
- 1.8 尝试 JDK 的下载、安装与配置操作。
- 1.9 尝试 Tomcat 的下载、安装与配置操作。
- 1.10 尝试 Eclipse 的下载、安装与配置操作。