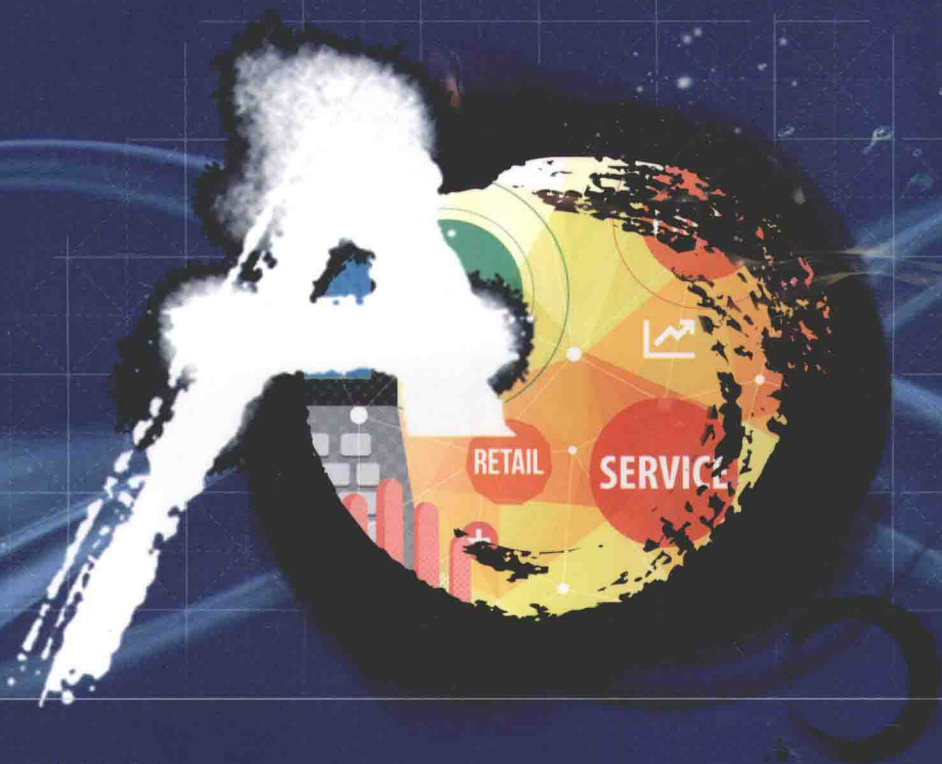




“十二五”普通高等教育本科国家级规划教材



高等学校规划教材

Web程序设计

(第4版)

◎吉根林 顾韵华 主编 ◎吴军华 郑玉 彭作民 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

“十二五”普通高等教育本科国家级规划教材

高等学校规划教材

Web 程序设计

(第4版)

吉根林 顾韵华 主编
吴军华 郑玉 彭作民 编著

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,也是国家级精品课程和优秀教材建设成果。本书主要介绍 Web 程序设计的方法与技术,使读者学会建立网站。全书共 8 章,包括:Web 编程基础知识;Web 应用程序开发与运行环境 Dreamweaver MX 及 Visual Studio 2012;HTML 与 XML;层叠样式表 CSS;Web 客户端程序设计;Web 服务器端程序设计;Web 数据库程序设计;ASP.NET 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,并且免费提供 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、网络工程、软件工程、电子商务、信息管理与信息系统、现代教育技术等相关专业教材,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, 吴军华, 郑玉, 彭作民编著. —4 版. —北京: 电子工业出版社, 2015.7

ISBN 978-7-121-26036-0

I. ①W… II. ①吉… ②顾… ③吴… ④郑… ⑤彭… III. ①网页制作工具—程序设计—高等学校—教材
IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2015) 第 097801 号

策划编辑: 袁 玺 文字编辑: 戴晨辰

责任编辑: 袁 玺

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

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: 17.75 字数: 454.4 千字

版 次: 2002 年 8 月第 1 版

2015 年 7 月第 4 版

印 次: 2015 年 7 月第 1 次印刷

定 价: 40.00 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。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

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

《Web 程序设计》前 3 版，承蒙广大读者的支持，被几十所高校选为相关课程教材，至 2014 年 12 月已连续印刷 20 次。在教材出版后的几年中，Web 应用程序开发技术又有了新的发展，同时从服务教学、服务读者的角度看，该教材还需进一步完善。为此，有必要对本书再次进行修订。

本次修订根据我们近年来从事“Web 程序设计”教学的经验与体会及读者的反馈建议，坚持“Web 程序设计”课程既定的教学目标：学会建网站；在保持原书基本风格的基础上，根据 Web 应用程序开发技术的发展趋势，以 ASP.NET 程序设计为主线，介绍 Web 应用程序开发技术，并对部分章节的内容进行了调整优化，进一步提高本书的先进性和实用性。本次修订的具体情况如下：

(1) 调整、优化本书的结构框架，以 ASP.NET 程序设计为主线介绍 Web 应用程序开发技术，从 Web 客户端程序设计、Web 服务器端程序设计、Web 数据库程序设计三个方面介绍 ASP.NET 程序设计的基本内容与相关技术。

(2) 在第 3 版 ASP.NET 程序设计基本内容的基础上，全面介绍了 ASP.NET 程序开发所需的技术，并将 VB.NET 语言更换为 C# 语言。

(3) 删去了“ASP 程序设计”一章，不再介绍 ASP 的知识与技术。

(4) 删去了 VBScript 脚本语言的介绍。

(5) 在“Web 应用程序开发与运行环境”一章中，将 ASP.NET 应用程序开发工具 Visual Studio 2008 版升级为 Visual Studio 2012 版。

(6) 在“ASP.NET 综合应用实例”一章中，删除了一个 ASP 综合应用实例，增加了一个 ASP.NET 综合应用实例，以培养读者对 ASP.NET 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 60~64 学时，上机实验 36~40 学时。全书配有大量例题，还安排了上机实验题，并给出了实验指导，包括实验目的、实验内容及实验步骤。其内容可能比教学时数所允许的稍多一些，可供教师讲课时选取或让学生自学。

本书为任课教师提供 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> 免费注册下载。欢迎任课教师及时向我们反馈授课心得和建议。

本次修订，第 1、2 章分别由南京师范大学吉根林教授、彭作民副教授执笔；第 3、4、5 章由南京信息工程大学顾韵华教授执笔；第 6、7 章分别由南京工业大学郑玉副教授、吴军华副教授执笔；第 8 章由顾韵华和彭作民共同编写；全书由吉根林和顾韵华担任主编，并统稿、定稿。在本次修订过程中，得到了电子工业出版社的支持，在此表示衷心的感谢！

由于作者水平有限，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。作者 E-mail 地址：glji@njnu.edu.cn。

作 者

目 录

第 1 章 Web 编程基础知识	(1)
1.1 什么是 Web	(1)
1.2 Web 的工作原理	(2)
1.3 Internet 网络协议	(3)
1.3.1 TCP/IP 协议	(3)
1.3.2 HTTP 协议	(4)
1.3.3 远程登录协议 Telnet	(4)
1.3.4 文件传输协议 FTP	(5)
1.4 IP 地址、域名和 URL	(5)
1.4.1 IP 地址	(5)
1.4.2 域名	(6)
1.4.3 统一资源定位器 URL	(6)
1.5 动态网页设计技术简介	(7)
1.5.1 PHP	(7)
1.5.2 JSP	(7)
1.5.3 ASP.NET	(8)
1.6 .NET 框架简介	(9)
本章小结	(9)
习题 1	(10)
第 2 章 Web 应用程序开发与运行环境	(11)
2.1 服务器端开发环境	(11)
2.2 客户端开发环境	(11)
2.3 网页设计工具 Dreamweaver MX	(12)
2.3.1 Dreamweaver MX 概览	(12)
2.3.2 Dreamweaver MX 的特性	(12)
2.3.3 Dreamweaver MX 界面介绍	(12)
2.4 Visual Studio.NET 开发工具	(17)
2.4.1 Visual Studio 2012 的安装	(17)
2.4.2 Visual Studio 2012 集成开发环境	(18)
2.4.3 Visual Studio 2012 集成开发环境的使用	(20)
本章小结	(22)
习题 2	(23)
上机实验 2	(23)

第3章 HTML 与 XML	(24)
3.1 页面设计概述	(24)
3.2 超文本标记语言 HTML	(25)
3.2.1 HTML 文档结构	(25)
3.2.2 HTML 基本标记	(27)
3.2.3 表格 (Table)	(31)
3.2.4 表单 (Form)	(34)
3.2.5 框架 (Frame)	(36)
3.3 可扩展标记语言 XML	(39)
3.3.1 XML 概述	(39)
3.3.2 XML 文档的编写	(41)
3.3.3 XML 文档的显示	(44)
本章小结	(48)
习题 3	(48)
上机实验 3	(49)
第4章 层叠样式表 CSS	(51)
4.1 样式表的定义和引用	(51)
4.1.1 样式表定义	(51)
4.1.2 样式引用	(52)
4.2 相关标记和属性	(54)
4.2.1 类选择符和 class 属性	(55)
4.2.2 id 选择符和 id 属性	(55)
4.2.3 伪类	(55)
4.2.4 span 标记	(56)
4.2.5 div 标记	(56)
4.3 样式的继承和作用顺序	(56)
4.3.1 样式的继承	(56)
4.3.2 样式的作用顺序	(57)
4.4 CSS 属性	(58)
4.4.1 字体属性	(58)
4.4.2 颜色和背景属性	(59)
4.4.3 文本属性	(61)
4.4.4 方框属性	(62)
4.4.5 列表属性	(63)
4.4.6 定位属性	(64)
4.5 CSS+DIV 页面布局	(66)
4.6 应用实例——设计个人主页	(66)
本章小结	(68)
习题 4	(68)
上机实验 4	(68)

第 5 章 Web 客户端程序设计	(70)
5.1 脚本语言 JavaScript	(70)
5.1.1 什么是脚本语言	(70)
5.1.2 JavaScript 语言概述	(71)
5.1.3 JavaScript 编程基础	(71)
5.1.4 JavaScript 对象	(81)
5.1.5 常用的内建对象和函数	(84)
5.2 浏览器对象模型及应用	(93)
5.2.1 浏览器对象模型	(94)
5.2.2 Navigator 对象	(94)
5.2.3 Window 对象	(95)
5.2.4 Document 对象	(98)
5.2.5 Form 对象	(103)
5.2.6 History 对象和 Location 对象	(108)
5.2.7 Frame 对象	(109)
5.2.8 程序示例——用户注册信息合法性检查	(110)
5.2.9 程序示例——扑克牌游戏程序	(113)
5.3 HTML DOM	(117)
5.3.1 HTML DOM 概述	(117)
5.3.2 DOM 节点树	(117)
5.3.3 DOM 树节点的属性	(118)
5.3.4 访问 DOM 节点	(119)
本章小结	(121)
习题 5	(122)
上机实验 5	(122)
第 6 章 Web 服务器端程序设计	(124)
6.1 初识 ASP.NET	(124)
6.1.1 一个简单的 ASP.NET 程序——用户登录程序	(124)
6.1.2 ASP.NET 程序结构分析	(125)
6.1.3 命名空间	(131)
6.2 C#语言基础	(132)
6.2.1 C#语法规则	(132)
6.2.2 数据类型与变量	(133)
6.2.3 运算符与表达式	(138)
6.2.4 流程控制语句	(142)
6.2.5 C#常用系统类	(147)
6.2.6 C#面向对象的编程	(150)
6.3 服务器控件	(160)
6.3.1 服务器控件的分类	(161)
6.3.2 Web 服务器控件的属性、事件和方法	(161)
6.3.3 标准服务器控件	(163)

6.4 ASP.NET 的对象	(177)
6.4.1 对象简介	(177)
6.4.2 Page 对象	(178)
6.4.3 Response 对象	(180)
6.4.4 Request 对象	(184)
6.4.5 Application 对象	(188)
6.4.6 Session 对象	(191)
6.4.7 Server 对象	(194)
6.5 ASP.NET 应用举例——建立网上课堂讨论区	(197)
本章小结	(201)
习题 6	(201)
上机实验 6	(202)
第 7 章 Web 数据库程序设计	(204)
7.1 Web 数据库访问技术	(204)
7.2 ODBC 接口	(205)
7.2.1 ODBC 接口概述	(205)
7.2.2 ODBC 的应用	(205)
7.2.3 创建并配置数据源	(206)
7.3 数据库语言 SQL	(207)
7.3.1 SQL 概述	(207)
7.3.2 主要 SQL 语句	(207)
7.4 ADO.NET 数据库组件	(209)
7.4.1 ADO.NET 组件模型	(210)
7.4.2 ADO.NET 的数据访问模式	(211)
7.5 ADO.NET 对象	(213)
7.5.1 Connection 对象	(213)
7.5.2 Command 对象	(215)
7.5.3 DataReader 对象	(219)
7.5.4 DataAdapter 对象	(221)
7.5.5 DataSet 对象	(222)
7.5.6 DataTable 对象	(223)
7.5.7 DataView 对象	(225)
7.6 数据源与 Web 控件的绑定	(227)
7.6.1 数据绑定方法	(227)
7.6.2 Repeater Web 控件绑定	(229)
7.6.3 DataList 控件绑定	(230)
7.6.4 DataGrid 控件绑定	(230)
7.6.5 GridView 控件绑定	(232)
7.7 ADO.NET 数据库访问示例——学生成绩查询与修改	(233)
本章小结	(236)
习题 7	(236)

上机实验 7	(236)
第 8 章 ASP.NET 综合应用实例	(238)
8.1 实例 1——基于数据库的 BBS 论坛管理	(238)
8.1.1 功能设计	(238)
8.1.2 数据库设计	(238)
8.1.3 界面设计	(238)
8.1.4 关键技术	(240)
8.1.5 实现过程	(240)
8.1.6 主要程序代码	(240)
8.2 实例 2——公文管理系统	(246)
8.2.1 系统功能	(246)
8.2.2 数据库设计	(247)
8.2.3 各子系统设计与程序代码	(249)
本章小结	(256)
附录 HTML、JavaScript、CSS、ASP.NET 实用列表	(257)
附录 A HTML 语言常用标记和属性	(257)
附录 B JavaScript 常用对象的属性、方法、事件处理和函数	(260)
附录 C CSS 样式表属性	(266)
附录 D ASP.NET 对象的集合、属性、方法和事件	(268)
参考文献	(273)

第 1 章 Web 编程基础知识

本章介绍开发 Web 程序应具备的基础知识, 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、ASP、ASP.NET、PHP、JSP 等动态网页设计技术以及 .NET 框架, 为在本课程中学习 Web 程序设计方法和开发技术做好准备。

1.1 什么是 Web

现在 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 Internet 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, 不管它是通过什么方式连入 Internet 的, 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, 目前尚无公认的确切定义。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, 因而它可以访问 Internet 的每一个角落。

(2) Web 是存储在全世界 Internet 计算机中的、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), 它是一种超文本 (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, 简称 B/S) 的结构。也就是说, Web 实际上是一种全球性通信系统, 它通过 Internet 使计算机相互传送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, 它易于使用和普及。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 的超文本链接使得 Web 文档不再像书本一样是固定的、线性的, 而是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, 一个很重要的原因就在于它

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 则提供将图形、图像、音频、视频信息集于一体的特性。同时，Web 是非常易于导航的，只需要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了。

(3) Web 与平台无关。无论系统的软、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，动态服务器页面）、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。

1.2 Web 的工作原理

Web 是一种典型的基于浏览器/服务器（Browser/Server，简称 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，它简化了客户端的维护，所有的应用逻辑都是在 Web 服务器上配置的。B/S 结构突破了传统客户机/服务器（Client/Server，简称 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 协议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，必须在本地计算机中安装浏览器软件。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，它能根据网页内容，对网页中的各种标记进行解释显示；同时，浏览器也是一种程序解释机，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，并且还要安装专门的信息服务器程序，如 Windows 所提供的 IIS（Internet Information Server，互联网信息服务）。大多数情况下，Web 服务器和浏览器处于不同的机器中，但它们也可以并存在同一台机器中。

Web 服务器向浏览器提供服务的过程大致可以归纳为以下步骤。

(1) 用户打开计算机(客户机),启动浏览器程序(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 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),服务器也将它们随 HTML 文档一道传送给浏览器,在浏览器所在的机器中执行。

(4) 如果 HTML 文档中嵌有 ASP 或 ASP.NET 程序,那么 Web 服务器就运行 ASP 或 ASP.NET 程序,并将结果传送给浏览器。Web 服务器运行 ASP 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, 虚拟现实建模语言)文档。只要浏览器中配置有 VRML 插件,或者客户机中已安装 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。

(1) 客户端编程就是客户端浏览器下载服务器中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。

(2) 服务器端编程就是将程序员编写的代码保存在服务器中,当用户提出对某个网页的请求时,这个请求所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。

1.3 Internet 网络协议

Internet 是由各种不同类型、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,凡是连入 Internet 的计算机都必须安装和运行 TCP/IP 协议软件。

1.3.1 TCP/IP 协议

TCP/IP 协议是一个协议集,其中最重要的是 TCP 协议和 IP 协议,因此,通常将这些协议简称为 TCP/IP 协议。

TCP/IP 协议把整个网络分成 4 个层次:应用层、传输层、网络层和物理链路层。它们都建立

在硬件基础之上。图 1-2 给出了 OSI 参考模型与 TCP/IP 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

(1) 应用层。它是 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。应用层主要有以下协议：

- ① 网络终端协议 Telnet //用于实现互联网中的远程登录功能；
- ② 文件传输协议 FTP //用于实现互联网中交互式文件传输功能；
- ③ 简单电子邮件协议 SMTP //用于实现互联网中电子邮件收发功能；
- ④ 网络文件系统 NFS //用于网络中不同主机间的文件系统共享；
- ⑤ 域名服务系统 DNS //用于实现网络设备域名到 IP 地址的映射服务；
- ⑥ 超文本传输协议 HTTP //用于在 Web 浏览器和服务端之间传输 Web 文档。

(2) 传输层。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。

(3) 网络层。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报文。它的主要功能包括以下三个方面：

- ① 处理来自传输层的分组发送请求；
- ② 处理接收的数据报文；
- ③ 处理互连的路径。

(4) 物理链路层。它的主要功能是接收 IP 层的 IP 数据报文，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 数据报文，向 IP 发送。该层是主机与网络的实际连接层。

1.3.2 HTTP 协议

HTTP 是专门为 Web 设计的一种网络协议，它属于 TCP/IP 参考模型中的应用层协议，位于 TCP/IP 协议的顶层。因此，它在设计和使用中以 TCP/IP 协议集中的其他协议为基础。例如，它要通过 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 浏览器和服务端用 HTTP 协议来传输 Web 文档。HTTP 基于客户端请求、服务端响应的工作模式，其定义的事务处理由以下 4 个步骤组成：

- (1) 客户端与服务端建立连接；
- (2) 客户端向服务端提出请求；
- (3) 如果请求被接受，则服务端送回响应，在响应中包括状态码和所需的文件；
- (4) 客户端和服务端断开连接。

1.3.3 远程登录协议 Telnet

Telnet 是关于远程登录的一个协议。要使用 Telnet，在用户的计算机中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 PC 键盘上输入一个 Telnet 子命令后,总能在远程主机上得到服务响应,并把结果送回到用户的 PC 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,进而用户可访问所需的各种信息,或运行远程主机中的程序来求解各种复杂的问题,一切都是在远程主机中快速执行(而不是将程序调回到用户主机中执行)后再从远程主机返回服务的结果。用户还可以利用 Telnet 连到 Internet 的各种服务器,如 Archie、Gopher、Wais、WWW 及其他服务器,比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,用户无须拥有自己的账号,即不用口令和用户名就能登录。在 Internet 上有许多这样的为公众开放的 Telnet 远程服务。

1.3.4 文件传输协议 FTP

Telnet 让用户主机能以终端方式共享 Internet 上各类主机的资源,却不能把远程主机中的文件复制到用户主机中。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 为显示目录/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 上的 Archie 服务器。第二,要明确传送的文件是什么类型的,即确定传送的是二进制文件还是 ASCII 码文件。如果文件传送类型不对,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作 FTP 服务器。有一种 FTP 服务器称为匿名 FTP 服务器,用户无须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。在 Internet 上有许多这样的、为公众开放的匿名 FTP 服务器。

1.4 IP 地址、域名和 URL

1.4.1 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 1 700 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 16 384 个网络,每个网络可包含 65 000 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 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。

1.4.2 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 协议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，其余的域对应于地理模式，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      //上述以外的组织；
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，依此类推，便形成了树形域名结构。由于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。

1.4.3 统一资源定位器 URL

WWW 信息分布在全球，要找到所需信息就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。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 `waits`（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冒号隔开。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，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

1.5 动态网页设计技术简介

随着网络技术的不断发展,单纯的静态网页已经远远不能满足 Internet 发展的需要。早期,动态网页使用的主要是 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 是一种功能强大的服务器端脚本编程环境,它是微软公司的产品,从 Windows NT Server 操作系统开始就附带这种脚本编程环境。1996 年底,微软公司推出了 ASP1.0,它内含于 IIS3.0 之中。1998 年,微软推出了 ASP2.0。2000 年,微软公司发布了 Windows 2000 操作系统,这个版本给我们带来了 IIS5.0 和 ASP3.0。ASP 最大的好处是可以包含 HTML 标记,也可以直接存取数据库以及使用 ActiveX 控件,它采用脚本语言 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。

1.5.1 PHP

PHP (Hypertext Preprocessor, 超文本预处理器) 是一种跨平台的服务器端嵌入式脚本语言,它是一种易于学习和使用的服务器端脚本语言,嵌入 HTML 文件,大量地借用 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 能够快速地产出动态网页的脚本代码。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 系统。

(2) 强大的数据库操作功能。PHP 提供丰富的数据库操作函数,它为各种流行数据库,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 语言共同使用;它与 HTML 语言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。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。

1.5.2 JSP

JSP (Java Server Pages, Java 服务器页面) 是 Sun 公司于 1999 年 6 月推出的网站开发语言。它是基于 Java Servlet 及整个 Java 体系的 Web 开发技术,利用这一技术可以建立先进、安全和跨平台的动态网站。它完全解决了目前 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。使用 JSP 标记或小脚本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 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。

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 页面依赖于可重用的、跨平台的组件来执行应用程序所要求的复杂处理，如使用 JavaBeans 或 Enterprise JavaBeans™ 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，Java 数据库连接）。很多数据库系统，如 Oracle、Sybase、MS SQL Server 和 MS 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 公司还开发了 JDBC-ODBC bridge，使用此项技术，Java 程序就可以访问带有 ODBC 驱动程序的数据库了。

1.5.3 ASP.NET

ASP.NET 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。它在结构上与前面的版本大不相同，它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。

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。ASP.NET 采用全新的编程环境，代表了技术发展的主流方向。从深层次说，ASP.NET 与 ASP 的主要区别体现在以下三个方面。

(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 程序完成之后，在每次请求时都要解释执行。这就意味着，它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 则是建立在 .NET 框架之上的，它可以使用 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 代码和 HTML 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