



高等院校规划教材
计算机科学与技术系列

Web 编程技术

厉小军 主 编



机械工业出版社
CHINA MACHINE PRESS



高等院校规划教材·计算机科学与技术系列

Web 编程技术

厉小军 主编

潘云 杨桂君 顾园妍 金晓彤 编著

机械工业出版社

本书从基本的语法和规范入手,以实例为导向,以实践为指导,较为全面地介绍了 JSP 和 Servlet 技术。主要内容包括:Web 编程技术概述,Web 编程技术基础,Servlet 编程,JSP 页面元素,JSP 内建对象,JSP 与 JavaBean,会话管理,MVC 设计模式,并以实际应用案例介绍了 JSP 技术的综合应用。

本书配有电子教案、程序源代码等教学资源,读者可以登录机工教材网(<http://www.cmpedu.com>)进行下载。

本书适合具有一定计算机基础知识的读者阅读,可作为软件工程、计算机科学与技术、电子商务、信息管理与信息系统、网络工程及相近专业本科 Web 编程技术相关课程的教材,也可作为大专院校非计算机专业学习 Web 编程的教学参考书和自学用书,还可供从事 Web 应用软件设计的科研人员参考。

图书在版编目(CIP)数据

Web 编程技术 / 厉小军主编. —北京:机械工业出版社, 2009.2

(高等院校规划教材·计算机科学与技术系列)

ISBN 978-7-111-26136-0

I. W… II. 厉… III. 主页制作—程序设计—高等学校—教材
IV. TP393.092

中国版本图书馆 CIP 数据核字(2009)第 013012 号

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:陈皓

责任印制:杨曦

北京瑞德印刷有限公司印刷(三河市胜利装订厂装订)

2009 年 3 月·第 1 版第 1 次印刷

184mm×260mm·23.5 印张·582 千字

0001—3000 册

标准书号:ISBN 978-7-111-26136-0

定价:39.00 元

凡购本书,如有缺页,倒页,脱页,由本社发行部调换

销售服务热线电话:(010) 68326294 68993821

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379753 88379739

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容更加趋于科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级和省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方式讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

随着 Internet/Intranet 网络技术的发展, Web 系统应用越来越广泛,越来越多的企业利用 Web 技术开发各种各样的应用系统,包括企业门户网站、企业 ERP、电子商务平台、供应链管理系统、客户关系管理系统等。为顺应此趋势,国内许多高等院校的计算机相关专业,纷纷开设 Web 编程技术的相关课程。目前, Web 编程技术主要有两大体系: Sun 公司的 JavaEE 和 Microsoft 公司的 .NET。其中, JavaEE 以 Java 技术为核心,满足 Web 系统对分布性、开放性和平台无关性的要求,获得了众多 IT 厂商和产品的支持。因此,大部分高等院校开设的 Web 编程技术课程都集中在介绍 JavaEE 的相关内容。

JSP 和 Servlet 是 JavaEE 中的基础和核心技术。本书从基本的语法和规范入手,以实例为导向,以实践为指导,深入浅出地讲解 JSP 和 Servlet 技术。本书与其他相关书籍相比,有较好的实用性,其立足 Web 系统开发的实践,以精炼的语言介绍相关技术,并且每部分都提供了参考实例,在最后一章以一个电子商务系统为例,介绍了 JSP 和 Servlet 技术的综合应用。

本书内容如下。

第 1 章“Web 编程技术概述”,介绍主流的 Web 编程技术和 Web 系统的运行环境。

第 2 章“Web 编程技术基础”,介绍 Web 编程基础,包括 HTML 语言和 JavaScript。

第 3 章“Servlet 编程”,介绍 Servlet 的概念和使用。

第 4 章“JSP 页面元素”,介绍 JSP 中的页面元素、编译指令和动作指令。

第 5 章“JSP 内建对象”,介绍 JSP 中的常用对象。

第 6 章“JSP 与 JavaBean”,介绍 JavaBean、JavaBean 数据库访问技术以及 JSP 与 JavaBean 如何结合。

第 7 章“会话管理”,介绍会话跟踪的几种主要技术,包括隐藏域、URL 重写、Cookie 和 Session 对象。

第 8 章“MVC 设计模式”,介绍 MVC 设计模式的概念、优越性和具体实现。

第 9 章“电子商务系统开发示例”,通过实际应用案例,介绍前面各章节的综合应用。

本书适合具有一定计算机基础知识的读者阅读,可作为软件工程、计算机科学与技术、电子商务、信息管理与信息系统、网络工程及相近专业本科 Web 编程技术相关课程的教材,也可作为大专院校非计算机专业学习 Web 编程的教学参考书和自学用书,还可供从事 Web 应用软件设计的科研人员参考。

本书的作者来自浙江工商大学和浙江财经学院等高校,具有多年讲授 Web 编程技术相关课程的教学经验,能很好地把握学生对这门课程的需求。本书由厉小军主编,参加编写的还有潘云、杨桂君、顾园妍和金晓彤。

限于作者水平,书中难免有不足之处,敬请读者批评指正。

作 者

目 录

出版说明

前言

第 1 章 Web 编程技术概述	1
1.1 WWW 服务	1
1.1.1 WWW 起源	1
1.1.2 WWW 特点	3
1.1.3 WWW 相关概念	4
1.2 常见应用系统的体系结构	7
1.2.1 客户机/服务器结构	7
1.2.2 浏览器/服务器结构	9
1.3 主流 Web 编程技术介绍	10
1.3.1 通用网关接口	10
1.3.2 PHP	11
1.3.3 活动服务器页面	11
1.3.4 Java 服务器页面	12
1.3.5 .NET 框架	12
1.3.6 Java 平台企业版	13
1.4 Web 应用的运行环境	14
1.4.1 Internet 信息服务	14
1.4.2 Tomcat 服务器	15
1.4.3 JBoss 应用服务器	16
1.4.4 WebLogic 服务器	16
1.5 第一个 Web 应用程序	18
1.5.1 准备开发环境	18
1.5.2 管理 Apache Tomcat	23
1.5.3 HelloWorld.jsp 程序	24
1.6 小结	25
1.7 习题	25
第 2 章 Web 编程技术基础	26
2.1 超文本标记语言	26
2.1.1 HTML 基本语法	27
2.1.2 URL 与超链接	28
2.1.3 表格	30
2.1.4 表单与表单元素	34
2.1.5 框架	39

2.1.6	定位标记	41
2.2	客户端脚本语言	42
2.2.1	JavaScript 常用对象	44
2.2.2	JavaScript 中的表单与事件处理	51
2.2.3	JavaScript 中的窗口管理	55
2.3	小结	58
2.4	习题	59
第 3 章	Servlet 编程	60
3.1	Servlet 概述	60
3.1.1	什么是 Servlet	60
3.1.2	Servlet 工作机制	61
3.2	Servlet 生命周期	61
3.2.1	一个简单的 Servlet 例子	61
3.2.2	Servlet 生命周期	63
3.2.3	Servlet API	64
3.3	Servlet 的配置	66
3.4	Servlet 案例	69
3.4.1	输入三角形三条边边长, 求面积	69
3.4.2	登录功能	72
3.5	小结	75
3.6	习题	76
第 4 章	JSP 页面元素	77
4.1	一个典型的 JSP 文件	77
4.2	脚本元素	81
4.2.1	声明	81
4.2.2	Scriptlets	83
4.2.3	表达式	84
4.2.4	声明与 Scriptlets 的比较	85
4.2.5	转义字符	87
4.3	指令元素	89
4.3.1	page 指令	89
4.3.2	include 指令	92
4.3.3	taglib 指令	93
4.4	动作元素	94
4.4.1	<jsp:param>	95
4.4.2	<jsp:include>	95
4.4.3	<jsp:forward>	98
4.4.4	<jsp:useBean>	101
4.4.5	<jsp:setProperty>	103

4.4.6	<jsp:getProperty>	103
4.4.7	<jsp:plugin>、<jsp:params>和<jsp:fallback>	104
4.5	HTML 表单	106
4.5.1	表单	107
4.5.2	文本框	108
4.5.3	列表框	111
4.5.4	单选按钮	115
4.5.5	复选框	117
4.6	综合案例	118
4.6.1	自提交页面	118
4.6.2	用 JSP 实现乘法和除法	120
4.6.3	用户注册	125
4.7	小结	129
4.8	习题	129
第 5 章	JSP 内建对象	131
5.1	out 对象	131
5.2	request 对象	134
5.2.1	getParameter()方法	134
5.2.2	getParameterValues()方法	136
5.2.3	其他方法	138
5.3	response 对象	140
5.4	session 对象	142
5.5	application 对象	145
5.6	pageContext 对象	146
5.7	综合案例	150
5.8	小结	153
5.9	习题	153
第 6 章	JSP 与 JavaBean	155
6.1	JavaBean 概述	155
6.1.1	JavaBean 简介	155
6.1.2	JavaBean 属性的命名	156
6.1.3	一个简单的 JavaBean 例子	156
6.2	JSP 与 JavaBean 的结合	157
6.2.1	创建 JavaBean: <jsp:useBean>	157
6.2.2	简单 JavaBean 属性的设置: <jsp:setProperty>	158
6.2.3	访问 JavaBean: <jsp:getProperty>	159
6.2.4	设置 JavaBean 属性高级技术	159
6.3	JavaBean 的生命周期	163
6.3.1	page 范围的 JavaBean	163

6.3.2	request 范围的 JavaBean	164
6.3.3	session 范围的 JavaBean	165
6.3.4	application 范围的 JavaBean	167
6.3.5	根据条件创建 JavaBean	168
6.4	使用 JavaBean 访问数据库	169
6.4.1	用 JavaBean 封装数据库访问	170
6.4.2	在 JSP 页面中使用 JavaBean 组件进行数据库操作	173
6.5	JavaBean 案例	174
6.6	小结	183
6.7	习题	183
第 7 章	会话管理	185
7.1	隐藏域	185
7.2	URL 重写	190
7.3	Cookie	192
7.4	session 对象	195
7.4.1	创建会话	196
7.4.2	在会话中保存和检索对象	197
7.4.3	销毁会话	198
7.5	综合案例	199
7.5.1	采用 session 实现猜数游戏	199
7.5.2	采用 session 实现页面访问计数器	201
7.6	小结	202
7.7	习题	202
第 8 章	MVC 设计模式	204
8.1	设计模式	204
8.1.1	什么是设计模式	204
8.1.2	MVC 设计模式	204
8.2	JSP Model1 和 JSP Model2	206
8.2.1	JSP Model1	206
8.2.2	JSP Model2	207
8.3	Struts 介绍	208
8.3.1	什么是 Struts	209
8.3.2	Struts 主要的模型组件	211
8.3.3	Struts 的视图组件	216
8.3.4	Struts 的控制器组件	217
8.3.5	标记库简介	223
8.4	MVC 案例	229
8.4.1	以 JSP+Servlet+JavaBean 来实现的 MVC 案例	229
8.4.2	以 Struts 来实现的 MVC 案例	234

8.5	小结	244
8.6	习题	244
第 9 章	电子商务系统开发示例	245
9.1	需求分析	245
9.2	系统结构设计	250
9.3	数据库设计	251
9.4	用户界面设计	256
9.5	对象模型设计	257
9.6	在线图书订购系统前台实现	266
9.6.1	公共基础类的实现	266
9.6.2	图书分类列表的获取	267
9.6.3	首页的实现	272
9.6.4	图书信息的查看	278
9.6.5	用户的注册、登录与注销	291
9.6.6	用户信息管理	301
9.6.7	购物车的实现	305
9.6.8	订单管理	315
9.7	在线图书订购系统后台实现	327
9.7.1	管理员的登录与注销	328
9.7.2	管理待处理订单	333
9.7.3	管理图书分类信息	340
9.7.4	管理图书信息	348
9.8	习题	365
参考文献		366

第1章 Web 编程技术概述

1.1 WWW 服务

WWW (World Wide Web, 简称为 WWW 或 Web) 是一种信息传播工具, 在 WWW 中, 任何用户都可以通过一台连入 Internet 的计算机来获取或发布信息。现在, WWW 常被当做 Internet 的同义词, 但实际上它只是运行在 Internet 上的一种网络服务。和 Internet 的发展史相比, WWW 的发展史要短得多。

1.1.1 WWW 起源

1. 1980 年~1991 年: WWW 的开发

最早关于 WWW 的构想可以追溯到 1980 年英国人 Tim Berners-Lee 为欧洲粒子物理研究中心 (CERN) 开发的一个名为 ENQUIRE 的项目, 这是一个可以处理超文本的在线编辑数据库。尽管它与现在使用的 WWW 大不相同, 但是它们有许多相同的核心思想, 甚至还包括了 Tim Berners-Lee 在 WWW 之后的另一个项目——语义网中的一些构想。

另一个与实现 WWW 密切相关的工作是 Bob Kahn 和 Vint Cerf 在 1977 年为跨越多个网段而设计的传输控制协议——TCP。1978 年, Internet 协议——IP 被用于和 TCP 一起工作。TCP/IP 在 Internet 中标准地位的确立为后来 WWW 的发展奠定了坚实的基础。

1984 年, Berners-Lee 回到了 CERN, 为了解决来自世界各地的物理学家们共享数据和发布信息的问题, 他于 1989 年 3 月提出了一个建设基于超链接的大型超文本数据库的建议。虽然他的提议并没有受到太多关注, 但是 Berners-Lee 还是决定在他的 NeXT 工作站上开发这个系统, 并且将此系统命名为 World Wide Web, 如图 1-1 所示。

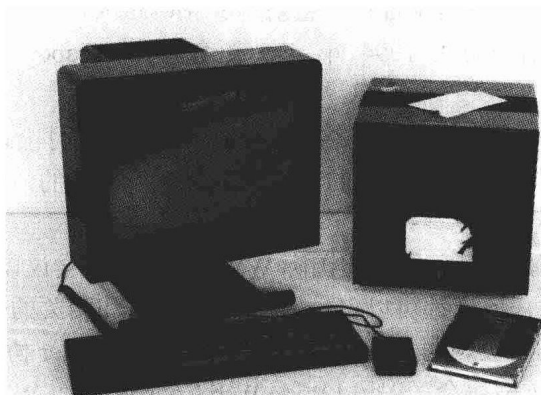


图 1-1 Tim Berners-Lee 在 CERN 使用的 NeXT 工作站成为世界上第一台 Web 服务器

到 1990 年底, Berners-Lee 已经实现了所有 Web 需要的工具, 包括: 第一个 Web 浏览器 (它同时也是一个 Web 编辑器), 第一个 Web 服务器 (info.cern.ch) 和第一个 Web 页面 (描

述该项目的页面)。使用 Berners-Lee 设计的浏览器可以访问 Usenet 新闻组和 FTP 服务,但是它只能在 NeXT 计算机上运行。于是, Nicola Pellow 实现了一个基于文本的浏览器,这个浏览器可以在任何计算机上运行。

美国斯坦福大学线性加速器中心(Stanford Linear Accelerator Center, SLAC)的 Paul Kunz 在 1991 年访问 CERN 的时候,被 Web 所吸引。于是把 Web 软件带回美国,将软件稍做修改后安装在 IBM 大型计算机上,用于发布 SLAC 文档。这是北美大陆的第一台 Web 服务器。

1991 年 8 月 6 日, Berners-Lee 在 Internet 新闻组 alt.hypertext 上发布了一条关于 WWW 的消息。这个日期现在被公认为 WWW 首次在 Internet 上亮相的日子。在 Berners-Lee 发布的简短消息中,他这样描述 WWW:

WWW (World Wide Web) 项目的目的是允许创建指向任何信息的链接。WWW 项目起初是为了让高能物理学家们能够共享数据、新闻和文档,现在我们非常乐意把 WWW 推广到其他领域并且为那些数据提供网关服务器。欢迎合作者的参与!

2. 1992 年~1995 年: WWW 的发展

最早的一批 WWW 用户与 CERN 有着非常密切的联系。它们多数是大学里的理科院系或物理实验室,如费米实验室和斯坦福大学的线性加速器中心。

早期的 Web 站点大多提供了 HTTP 和 Gopher 两种访问方式。有些站点可以使用 WAIS 服务进行检索(类似于现在的搜索引擎),但网络浏览器仍然是字符界面的。直到 1992 年 4 月,芬兰赫尔辛基理工大学发布 Erwise。同年 5 月,美国加州大学伯克利分校的魏培源开发了功能更强的 ViolaWWW 浏览器,该浏览器可以支持嵌入的图像、脚本和动画。无论是 Erwise 还是 ViolaWWW,都运行在 UNIX 的 X Windows 环境下。

WWW 的发展在 1993 年伴随着 Mosaic 浏览器的发布出现了转折点。Mosaic 是一个图形界面的 Web 浏览器,它是由美国伊利诺斯大学国家超级计算机应用中心(National Center for Supercomputing Applications, NCSA)的一个项目小组开发的。最初, NCSA 只是建立了一个 Web 站点。到了 1992 年 12 月,伊利诺斯大学的两名在 NCSA 工作的学生 Andreessen 和 Eric Bina 开始了 Mosaic 的开发工作。1993 年 2 月,他们发布了 X Window 环境下的网络浏览器 Mosaic。Mosaic 以其对多媒体信息的支持和众多新功能而大受欢迎。Andreessen 在毕业后和硅谷图形公司(SGI)的前总裁 Jim Clark 一起创建了 Mosaic 通信公司,开始了 Mosaic 浏览器的商业开发。Mosaic 通信公司在 1994 年 4 月更名为 Netscape,并且将浏览器也更名为 Netscape Navigator。

第一个 Microsoft Windows 环境下的网络浏览器是 1993 年 6 月正式发布的 Cello。它是由 Thomas.R.Bruce 为美国康奈尔大学法学院的法律信息研究所开发的,因为大多数的律师使用的是 Windows 而不是 UNIX。

1994 年 5 月, CERN 主持召开了第一届国际 WWW 大会,该会议以后每年都会召开一次。在这次会议上, CERN 同意任何人都可以自由地使用 WWW 和相关的代码,并且不用为此支付版权。这么做也是为了回应美国明尼苏达大学声称将要收取 Gopher 协议许可费的消息。

到了 1994 年 9 月,万维网协会(World Wide Web Consortium, W3C)在美国麻省理工学院成立, Tim Berners-Lee 出任主席一职。

3. 1996 年~1998 年: WWW 的商业化

到 1996 年,对于大多数商业公司而言,创建一个 Web 站点已经成为一件必须要做的事

情了。虽然一开始大家关注的都还只是在 Web 上免费发布信息以及通过 Web 便捷地获取信息，但是很快 Web 的交互性就直接导致了基于 Web 的商业行为（即电子商务）和世界范围内的多用户通信。Web 以及在此之上产生的众多新概念激发了许多人的热情和想象力，他们很快意识到新的商业模式将与 Web 密不可分，许多人迫不及待地投身其中并希望从中掘得第一桶金。

4. 1999 年～2001 年：.com 的兴起和繁荣

1998 年～1999 年的低利率刺激了当时西方资本市场的投资行为，许多拥有创业计划和管理能力的新兴企业家由此可以获得他们所需要的风险投资，大批具有 Internet 背景的.com 公司应运而生。网络技术可以与 19 世纪 40 年代的铁路建设、20 世纪 20 年代的无线电技术、20 世纪 50 年代的电子管技术、20 世纪 60 年代的分时计算机以及 20 世纪 80 年代的家庭计算机和生物技术等新技术带来的繁荣景象相媲美。

2001 年，在经历数年的过度繁荣之后，互联网泡沫逐渐破灭。许多.com 公司在用光了投资人的风险投资资金后却没有给投资人带来任何赢利，因此纷纷退市。

5. 2002 年至今：无处不在的 WWW

泡沫的破灭并没有延缓互联网前进的脚步，相反，它促进互联网以一种更加稳健和理性的方式发展。今天，WWW 使得全世界的人们以史无前例的巨大规模相互交流。距离遥远或不同阶层的人们可以通过网络建立起紧密的联系，使彼此思想境界得到升华，甚至改变他们对待事务的态度及精神。情感经历、政治观点、文化习惯、表达方式、商业建议、艺术、摄影和文学都可以以人类历史上从来没有过的低投入实现数据共享。尽管 WWW 还存在着许多不足，但不可否认它是历史上影响最深远、最广泛的传播媒介。信息经由 WWW 和 Internet 来传播，使得用户可以通过比传统方式更有效率的方式查询信息资源、相互联系。这在很大程度上加快了人类进步的步伐。

1.1.2 WWW 特点

WWW 是一种超文本信息系统，它的一个主要概念就是超文本链接。WWW 使得文本不再像一本书一样是固定的线性的，而是可以从一个位置跳转到另外的位置。用户可以从其中获取更多的信息，也可以跳转到别的主题上。如果想要了解某一个主题的内容，只要在这个主题上用鼠标单击一下，就可以跳转到包含这一主题的文档上。

WWW 是图形化且易于导航的。它之所以非常流行的一个重要原因就在于它可以在一页上同时显示色彩丰富的图形和文本。在 WWW 之前，Internet 上的信息只有文本形式。WWW 可以将图形、音频和视频信息集合于一体。同时，WWW 又是非常易于导航的，只需从一个链接跳到另一个链接，就可以实现在各页或各站点之间的浏览了。

1. WWW 与平台无关

无论用户的系统平台是什么，都可以通过 Internet 访问 WWW。浏览 WWW 对用户的系统平台没有特别的限制。无论从 Windows、UNIX、Linux 或 Macintosh，或者别的平台，用户都可以访问 WWW。对 WWW 的访问是通过一种叫做浏览器（Browser）的软件来实现的，如 Mozilla 的 Firefox、Opera 或者 Microsoft 的 Internet Explore 等。

2. WWW 是分布式的

大量的图形、音频和视频信息会占用相当大的磁盘空间，用户甚至无法预知信息的多少。

对于 Web 而言，没有必要把所有信息都放在一起。信息可以放在不同的站点上，只需要在浏览器中指明这个站点就可以了。这种机制使得物理上并不一定在一个站点上的信息在逻辑上实现了一体化，即从用户的角度来看，这些信息是一体的。

3. WWW 是动态的

各 Web 站点的信息提供者可以经常对网站上的信息进行更新，并尽量保证信息的时效性。因此 Web 站点上的信息是动态和不断更新的，这需由信息的提供者来保证。

4. WWW 是交互的

Web 的交互性首先表现在它的超链接上。用户的浏览顺序和所到站点完全由他自己决定。另外，通过服务器端动态网页（Active Server Pages, ASP）的形式可以从服务器方获得动态的信息：用户通过填写表单向服务器提交请求，服务器根据用户的请求返回相应信息。

1.1.3 WWW 相关概念

WWW 的核心部分是由以下 3 个标准构成的。

- 1) 统一资源标识符（URI）。
- 2) 超文本传输协议（HTTP）。
- 3) 超文本标记语言（HTML）。

除了这 3 个标准之外，WWW 还包含了一些其他概念，下面对所有这些概念进行基本介绍。

1. 统一资源标识符

统一资源标识符（Uniform Resource Identifier, URI）是互联网的一个协议要素，可以通过它来定位任何远程或本地的可用资源，通常包括 HTML 文档、图像、视频片段和程序等资源。

URI 的通用格式如下。

协议:[/] [[用户名[:密码]@]主机名[:端口号]]/资源路径]

其中协议指的是访问资源使用的协议名称，常见的有 http、ftp、mailto 和 file 等。协议和主机名称或 IP 地址间使用半角冒号（:）连接，包含资源路径的冒号后还有两个半角斜杠（//）。

主机名称或 IP 地址用于在网络中查找服务器并和它连接（TCP 要求建立连接，UDP 不要求，但大部分协议是采用 TCP 的），以此来获取资源。有时服务器采用的不是网络协议的默认端口，这就需要在主机名称后加上端口号，并使用半角冒号连接，但这不是必须的。因为在大部分情况下，服务器都会采用协议默认的端口号。连接的用户名和密码并非在所有情况下都要求，况且在要求用户名的同时也不一定要求密码。通常用户名和密码及主机名称之间使用半角 @ 符连接，而用户名和密码之间则使用半角冒号。常见要求用户名的情况是用 mailto 协议来发送邮件，而 ftp 通常情况下也要求用户名及密码，即使用用户名和密码为空，客户端也会发送 anonymous 和一个邮箱地址当做用户名和密码。在 file 协议下访问的通常是本机的资源，在这种情况下不存在主机名称或 IP 地址，而是由协议直接连接本地地址。

资源路径即资源存放在服务器或本地计算机的物理或虚拟位置。

2. 超文本传输协议

超文本传输协议（HyperText Transfer Protocol, HTTP）是互联网上应用最为广泛的一种

网络传输协议。所有的 WWW 文档都必须遵守这个协议。设计 HTTP 的最初目的是为了提供一种发布和接收 HTML 页面的方法。

HTTP 的发展是万维网协会（W3C）和 Internet 工作小组合作的结果，在 RFC 2616 中定义了 HTTP 1.1 这个今天普遍使用的版本。

HTTP 被用于客户端和服务端间的请求和应答。一个 HTTP 客户端（如 Web 浏览器），通过建立一个到远程主机特殊端口（默认端口为 80）的连接，初始化一个请求。一个 HTTP 服务器通过监听特殊端口，等待客户端发送一个请求序列。接收到请求序列后，服务器会发回一个回复及一个消息，此消息的主体可能是被请求的文件、错误消息或其他信息。

HTTP 并不局限于使用网络协议（TCP/IP）及其相关支持层，事实上，它可以在任何其他互联网协议之上执行，或者在其他网络上执行。HTTP 只要求可靠的传输，任何能够提供这种保证的协议都可以被其使用。

与其他基于 TCP 的协议不同的是，在一个特殊的请求（或者请求的相关序列）完成后，HTTP 连接通常被中断。这对网页设计者来说，会产生一些问题。

HTTP 存在一个安全版本，即 HTTPS。HTTPS 支持任何加密算法，不过此加密算法须能被页面双方所理解。

3. 超文本标记语言

超文本标记语言（HyperText Markup Language, HTML）是为网页创建和其他可在网页浏览器中看到的信息而设计的一种标记语言。HTML 被用来结构化信息——如标题、段落和列表等，也可用来在一定程度上描述文档的外观和语义。

包含 HTML 内容的文件最常用的扩展名是.html，但是像 DOS 这样的旧操作系统限制扩展名为最多 3 个字符，所以也可以使用.htm 作为扩展名。

早期的 HTML 语法被定义成较松散的规则，以有助于不熟悉网络出版的人使用。网页浏览器接受了这个现实，并且可以显示语法不严格的网页。随着时间的流逝，官方标准渐渐趋于严格，但是许多网络浏览器继续支持显示一些不合乎标准的 HTML 文档。

4. Web 服务器

Web 服务器也称为 WWW 服务器，主要功能是提供网上信息浏览服务。常用的 Web 服务器软件主要有两个：Microsoft 的互联网信息服务器（Internet Information Server, IIS）和 Apache 的 HTTP 服务器。

Web 服务器可以处理 HTTP。当 Web 服务器接收到一个 HTTP 请求之后，它会返回一个 HTTP 响应（例如，送回一个 HTML 页面）。为了处理请求，Web 服务器的响应形式可以包括一个静态页面或图片，进行页面跳转或者把动态响应的产生委托给一些其他的程序。

5. 浏览器

浏览器是一种可以显示 Web 服务器或文件系统的 HTML 文件内容的软件，并让用户与这些文件进行交互。浏览器主要通过 HTTP 与 Web 服务器交互并获取网页，这些网页由 URL 指定，文件格式通常为 HTML。大部分的浏览器支持除了 HTML 之外的广泛的格式。例如，JPEG、PNG 和 GIF 等图像格式，并且能够扩展支持众多的插件（plug-ins）。另外，许多浏览器还支持其他的 URL 类型及其相应的协议，如 FTP、Gopher 和 HTTPS。

个人电脑上常见的浏览器包括 Microsoft 的 Internet Explorer 和 Opera, Mozilla 的 Firefox, 在苹果电脑中使用的 Safari。

6. JavaScript

JavaScript 是一种面向对象的脚本语言，网景通讯公司最初在它的 Netscape Navigator 2.0 产品上设计并实现，原名 LiveScript，目前 JavaScript 已经成为在 WWW 上广泛使用的用于动态网页的编程语言。

7. 层叠样式表

层叠样式表 (Cascading Style Sheets, CSS) 是 W3C 定义和维护的标准，是一种为用标记语言书写的文档 (如 HTML 文档或 XML 应用) 添加样式 (字体、间距和颜色等) 的语言。目前最新版本是 CSS 2.1，为 W3C 的候选推荐标准。下一版本 CSS 3 仍然在开发过程中。

一个网页的使用者可以使用 CSS 来决定文件的颜色、字体和排版等显示特性。CSS 最主要的目的是将文件的结构与文件的显示分离开来。这种分离使得文件的可读性加强，文件的结构更加灵活，网页的使用者可以自己决定文件的显示。另外，文件的结构也大大简化了。

8. Web 1.0

Web 1.0 这个词实际上是在 Web 2.0 一词之后出现的。它并没有一个准确的定义，和 Web 2.0 的界限也很模糊，一般情况下指的是 Web 2.0 热潮之前的 Web，主要包括在 1994 年~2004 年之间建立的 Web 站点。这一类站点通常以信息发布作为主要功能。

在 2006 年 11 月的科技网络创新高峰会议 (Technet Summit) 上，Netflix 公司的创始人兼首席执行官 Reed Hastings 对 Web 1.0~Web 3.0 给出了一个简单定义：

Web 1.0 是通过拨号网络访问的，平均只有 50Kbit/s 的带宽；Web 2.0 的平均带宽有 1Mbit/s；Web 3.0 将具有 10Mbit/s 的带宽，它将是全视频的 Web。

典型的 Web 1.0 站点通常具有以下设计元素。

- 1) 静态页面。
- 2) 使用框架或框架集。
- 3) 使用一些诸如<marquee>标记这样的专有 HTML 扩展标记。
- 4) 提供在线客户留言簿。
- 5) 使用 Gif 图片作为网站的按钮。

9. Web 2.0

从 Web 技术的角度来看，Web 2.0 代表的是一种 Web 设计和开发的趋势——第二代基于 Web 的网上社区和基于主机的网络服务，如社交网络 (Social-Networking)、Wiki、博客 (Blog)、协作标签 (Folksonomies) 等。所有这些新事物都力图促进人与人之间的交流、共享、协作和创造性。Web 2.0 一词在 2004 年第一次 O'Reilly Media Web 2.0 会议后开始流行起来，虽然从表面上看这似乎是一个 WWW 的新版本，但它实际上并不意味着技术规范的更新，它改变的是软件开发人员和最终用户使用 Web 的方式。O'Reilly 公司的创始人和首席执行官 Tim O'Reilly 这样定义 Web 2.0：Web 2.0 是计算机工业中的商业革命，引起这场革命的原因是商业活动开始以互联网为平台，以及在这个新平台上寻找成功之路的努力和尝试。另一来自 IBM 公司的社交网络分析家 Dario de Giudibus 提出了一个不同的定义，这个定义更侧重于交互性和系统架构的实现。他是这样定义 Web 2.0 的：Web 2.0 是一个面向知识的环境，在这个环境中，内容来自于人们彼此的交互行为，并且这些内容在一个面向服务的架构中被网络应用程序发布、管理和使用。

1.2 常见应用系统的体系结构

所谓应用系统的体系结构，指的是系统的整体设计及系统各个组成部分之间的关系。对于系统的体系结构究竟应该包含哪些内容并没有严格定义，不同的组织对此存在着各自的描述。例如，美国国家标准学会（ANSI）和国际电气电子工程师学会（IEEE）认为，体系结构应该包括系统的基本组织结构、构成系统的组件及其相互关系、系统所处的环境及系统的设计和变更原则。

就计算机软件开发而言，其常用的体系结构包括客户机/服务器结构和浏览器/服务器结构两种类型。

1.2.1 客户机/服务器结构

客户机/服务器（Client/Server）结构通常应用于计算机网络环境下，它把客户端和服务端分离开来，每一个连到网络上的客户或服务器都可以被称作“节点”。最基本的客户机/服务器结构只包含两类节点：客户和服务器，它有时也被称为二层（Two-tier）结构。在这种结构中，设备可以共享文件和资源。

每一个客户端软件的实例都可以向一个或多个连在网络上的服务器发送数据访问请求。在服务器端，服务器接收这些请求，对它们进行处理，然后将处理结果返回给客户端。客户机/服务器结构在各种网络应用中得到广泛的应用，虽然这些应用的表现形式各异，但它们的本质是一样的。例如，常见的客户端包括网络浏览器、FTP 客户端、电子邮件客户端程序和网络游戏客户端程序，常见的服务器则包括 Web 服务器、FTP 服务器、电子邮件服务器和数据库服务器等。

客户机/服务器结构图如图 1-2 所示。

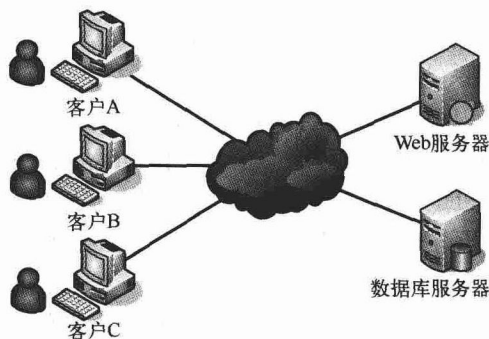


图 1-2 客户机/服务器结构图

客户端的特性如下。

- 1) 客户端是发出请求的一方。
- 2) 需要等待并且接收响应。
- 3) 同一时刻通常只连接数量不多的服务器。
- 4) 通常使用图形用户界面直接与最终用户打交道。