

普通高等教育“计算机类专业”规划教材

# ASP.NET

# 网站设计教程

刘萍 谢旻旻 主编

李学峰 赵颖 副主编



清华大学出版社

普通高等教育“计算机类专业”规划教材

# ASP.NET 网站设计教程

刘萍 谢旻旻 主编

李学峰 赵颖 副主编

李彩霞 刘琴 潘春花 参编

清华大学出版社  
北京

## 内 容 简 介

全书共 12 章,分为 3 个部分,分别是基础篇、核心篇和实战篇。本书全面而详尽地介绍了微软公司推出的新一代企业级开发平台 ASP.NET。通过本书的学习,读者能够在最短的时间内开发出具有高效性、高可靠性和高可扩展性的网站。

本书主要内容包括 ASP.NET 基础、HTML 简介及使用技巧、C# 语言基础、ASP.NET 常用对象、ASP.NET 服务器控件、数据验证、ADO.NET、数据绑定、ASP.NET 应用程序配置和状态管理以及 ASP.NET AJAX 服务器端编程,最后通过具体的实例详细说明了利用 ASP.NET 开发项目的过程。

本书内容丰富、可操作性强,可作为高等院校本、专科计算机专业或非计算机相关专业的教材,也可作为各类 ASP.NET 培训和广大读者自学与参考的资料。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

ASP.NET 网站设计教程/刘萍,谢旻旻主编. —北京:清华大学出版社,2013.2

(普通高等教育“计算机类专业”规划教材)

ISBN 978-7-302-30972-7

I. ①A… II. ①刘… ②谢… 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(2012)第 301698 号

责任编辑:白立军

封面设计:常雪影

责任校对:梁毅

责任印制:何芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:23

字 数:576千字

版 次:2013年2月第1版

印 次:2013年2月第1次印刷

印 数:1~3000

定 价:39.00元

产品编号:049187-01

ASP.NET 是微软公司推出的企业级网站开发平台,是目前国内外开发中小型企业网站的首选技术。这不仅归功于微软公司 .NET 发展战略的成功实施,更重要的是 ASP.NET 技术本身所具有的无限魅力——功能强大、易学易用、高效快捷。在与其同步推出的集成开发环境 Microsoft Visual Studio 下编程,更如行云流水一般。目前,ASP.NET 应用与开发技术已被很多大中专院校纳入网页设计相关课程的教学之中。

本书作为一本主要用于高校教学的计算机专业教材,由于受篇幅限制,不可能对 ASP.NET 的所有技术细节深挖细掘,但采取了“删繁就简”的基本原则,重点介绍 Web 开发的技术体系,帮助读者建立网站建设知识框架,直观展示实例项目的实际开发技术与操作步骤,剖析实例的技术要点。通过使用本教材,让读者能在实践中逐步培养出真正的软件开发能力。本书各章节都有“实践与练习”部分,提供了许多与讲授内容密切相关的编程场景与习题,其目的是进一步总结归纳知识点,培养读者的应用和探索能力。

规范应该是开发者从开始学习时就应养成的良好编程习惯,但很多人往往忽视了这一点,只将关注的重心放在技术点上,结果反而制约了技术能力的发挥。因为在实际开发工作中,无论是团队开发,还是产品开发,要求都是统一规范,如果到那时再培养自己的规范习惯,欲速则不达了。本书并没有刻意去介绍编程规范,而是将其融入到每一个具体的实例中,使读者在学习过程中,自然而然地感知并强化良好的编程规范习惯。

本书主要采用 ASP.NET 2.0 版本,用 C# 作为后台编程语言,C# 是微软专为 .NET 系统量身定做的语言,越来越多的 .NET 开发者选择了 C# 语言。如果读者已经掌握了 VB、C/C++ 或者 Java 语言,那么 C# 也就理解七八成了。第 3 章会对 C# 进行简单介绍。本书案例的开发工具为 Microsoft Visual Studio 2005,也可以使用完全免费的 Microsoft Visual Web Developer 2008 速成版。

总之,让读者掌握 ASP.NET 这个强大的开发工具,是编写本书的目的。本书主要面向学习 ASP.NET 技术的初、中级读者,特别适合于广大从事 IT 的工作人员,即初、中级网站开发者以及动态网页的设计者或业余爱好者,也可以作为高等院校本、专科计算机专业或非计算机相关专业的教材或参考书。作者多次使用这些内容给本专科的计算机专业和电子信息类专业学生授课,取得了良好的教学效果。

本书共 12 章,分为三大部分。

第一部分基础篇,包括 ASP.NET 基础、HTML 简介及使用技巧、C# 语言基础、ASP.NET 常用对象和 ASP.NET 服务器控件;第二部分核心篇,包括数据验证、ADO.NET、数据绑定、ASP.NET 应用程序配置和状态管理以及 ASP.NET AJAX 服务器端编程;第三部

分实战篇,涉及具体的应用程序设计实例,包括新闻发布系统和注册及登录验证模块设计。

其中,第1、2、3、4章由青海民族大学计算机学院教授刘萍编写,第5章由青海广播电视大学开放教育学院讲师赵颖编写,第6章由青海民族大学计算机学院教授李彩霞编写,第7、11、12章由青海民族大学计算机学院讲师谢旻旻编写,第8章由青海广播电视大学副教授李学峰编写,第9章由青海民族大学计算机学院讲师刘琴编写,第10章由青海民族大学计算机学院讲师潘春花编写。

鉴于编者水平所限,书中纰漏和考虑不周之处在所难免,恳请专家和广大读者不吝赐教,批评指正。

编者

2012.12

F O R E W O R D

第 1 章 ASP.NET 基础 /1

1.1 C/S 与 B/S 架构体系 /1

1.1.1 C/S 架构 /1

1.1.2 B/S 架构 /1

1.1.3 C/S 与 B/S 的区别 /2

1.2 静态网页与动态网页 /3

1.2.1 静态网页技术 /3

1.2.2 动态网页技术 /3

1.2.3 静态网页和动态网页的特点比较 /3

1.2.4 动态网页的发展阶段 /4

1.3 .NET Framework 基础 /5

1.3.1 .NET Framework 概述 /5

1.3.2 公共语言运行库 /6

1.3.3 .NET Framework 类库 /6

1.3.4 .NET Framework 的功能 /7

1.4 ASP.NET 环境搭建 /7

1.4.1 安装 Visual Studio 2005 /7

1.4.2 配置 Visual Studio 2005 开发环境 /18

1.5 测试 ASP.NET 环境 /24

1.5.1 认识 Visual Studio .NET 开发环境 /24

1.5.2 第一个 Windows Console 程序 /25

1.5.3 第一个 Windows Form 程序 /25

1.5.4 第一个 ASP.NET 程序 /28

1.6 实践与练习 /29

第 2 章 HTML 简介及使用技巧 /31

2.1 HTML 文档基本结构标签 /31

2.1.1 制作一个基本的网页 /31

2.1.2 HTML 文件的基本结构 /32

2.2 文本和图像标签 /35

2.2.1 常用文本标签 /35

2.2.2 图像标签 /42

2.2.3 超链接标签 /43

2.3	表格	/45
2.3.1	表格基本结构	/45
2.3.2	表格常用标签及属性	/47
2.3.3	表格应用	/49
2.4	表单	/52
2.4.1	表单基本结构	/52
2.4.2	表单常用控件及属性	/53
2.4.3	表单应用	/59
2.5	框架	/60
2.5.1	框架集与框架	/60
2.5.2	框架应用	/63
2.6	CSS 样式表	/64
2.6.1	CSS 基础	/64
2.6.2	样式表的创建	/64
2.6.3	样式表的应用	/70
2.6.4	CSS 各种样式的定义	/71
2.7	HTML 的其他常用标签	/73
2.7.1	嵌入多媒体文件	/73
2.7.2	播放背景音乐	/76
2.7.3	滚动效果	/78
2.8	实践与练习	/79
<b>第3章 C# 语言基础 /81</b>		
3.1	程序结构	/81
3.1.1	命名空间	/81
3.1.2	类	/83
3.1.3	结构	/86
3.2	变量与运算符	/89
3.2.1	变量	/89
3.2.2	运算符	/96
3.3	流程控制	/101
3.3.1	分支语句	/101
3.3.2	循环语句	/110

3.4	集合类型	/120
3.4.1	数组	/121
3.4.2	枚举	/123
3.5	错误和异常处理	/125
3.6	实践与练习	/129
<b>第4章 ASP.NET 常用对象 /135</b>		
4.1	概述	/135
4.1.1	ASP.NET 页面处理过程	/135
4.1.2	ASP.NET 对象概述	/136
4.2	Response 对象	/137
4.2.1	Response 对象简介	/137
4.2.2	向浏览器发送信息	/138
4.2.3	重定向	/138
4.2.4	输出文本文件	/139
4.2.5	设置缓冲区	/139
4.2.6	检查浏览者联机状态	/140
4.2.7	在指定时间段显示网页	/140
4.3	Request 对象	/141
4.3.1	Request 对象的属性和方法	/141
4.3.2	获取表单数据	/143
4.3.3	获取客户端浏览器信息	/144
4.3.4	获取服务器端环境变量	/145
4.3.5	获取当前浏览器网页的路径	/146
4.4	Server 对象	/146
4.4.1	Server 对象的常用属性和方法	/146
4.4.2	HTML 编码和解码	/147
4.4.3	URL 编码和解码	/148
4.4.4	执行指定程序	/150
4.5	Cookie 对象	/151
4.5.1	Cookie 对象概述	/151
4.5.2	Cookie 对象常用属性和方法	/151



4.5.3	使用 Cookie 对象保存和读取客户端信息	/152
4.5.4	创建 Cookie 对象	/152
4.5.5	获取 Cookie 对象	/153
4.6	实践与练习	/154

## 第 5 章 ASP.NET 服务器控件 /156

5.1	HTML 服务器控件	/156
5.1.1	表格	/156
5.1.2	表单	/160
5.1.3	图像	/165
5.2	常用控件	/167
5.2.1	Label 控件	/167
5.2.2	TextBox 控件	/170
5.2.3	Button 控件	/174
5.2.4	LinkButton 控件	/177
5.2.5	ImageButton 控件	/179
5.2.6	HyperLink 控件	/182
5.2.7	ListBox 控件	/184
5.2.8	DropDownList 控件	/187
5.2.9	RadioButton 控件和 RadioButtonList 控件	/191
5.2.10	CheckBox 控件和 CheckBoxList 控件	/195
5.2.11	Image 控件	/201
5.2.12	ImageMap 控件	/202
5.2.13	Panel 容器控件	/206
5.2.14	FileUpload 文件上传控件	/211
5.3	实践与练习	/214

## 第 6 章 数据验证 /216

6.1	数据验证控件	/216
6.1.1	非空数据验证 RequiredFieldValidator	

6.1.1	控件	/218
6.1.2	数据范围验证 RangeValidator 控件	/219
6.1.3	数据比较验证 CompareValidator 控件	/220
6.1.4	数据类型验证	/221
6.1.5	数据格式验证 RegularExpressionValidator 控件	/221
6.1.6	自定义数据验证方法 CustomValidator 控件	/224
6.1.7	页面统一验证 ValidationSummary 控件	/225
6.2	实践与练习	/225
	<b>第7章 ADO.NET</b>	<b>/227</b>
7.1	ADO.NET 简介	/227
7.1.1	ADO.NET 对象模型	/228
7.1.2	ADO.NET 命名空间	/229
7.2	Connection 对象连接数据库	/230
7.2.1	使用 SqlConnection 对象连接 SQL Server 数据库	/231
7.2.2	使用 OleDbConnection 对象连接 OLE DB 数据源	/231
7.3	Command 对象操作数据	/232
7.3.1	查询数据	/233
7.3.2	添加数据	/235
7.3.3	修改数据	/238
7.3.4	删除数据	/241
7.4	结合使用 DataSet 对象和 DataAdapter 对象	/244
7.4.1	使用 DataAdapter 对象填充 DataSet 对象	/244
7.4.2	对 DataSet 中的数据操作	/245

7.5	DataReader 对象读取数据	/247
7.5.1	使用 DataReader 对象读取数据	/248
7.5.2	DataReader 对象和 DataSet 对象的 区别	/249
7.6	实践与练习	/250
<b>第 8 章 数据绑定 /251</b>		
8.1	数据绑定简介	/251
8.1.1	简单数据绑定	/251
8.1.2	用于简单数据绑定控件	/256
8.2	GridView 控件	/260
8.2.1	GridView 控件概述	/260
8.2.2	GridView 控件绑定数据源	/260
8.2.3	GridView 控件外观设置	/262
8.2.4	GridView 控件分页显示数据	/270
8.2.5	GridView 控件中数据排序	/273
8.3	DataList 控件	/276
8.3.1	DataList 控件概述	/276
8.3.2	DataList 控件绑定数据源	/277
8.3.3	使用 SelectedItemTemplate 模板	/279
8.3.4	在 DataList 控件中编辑数据	/282
8.4	Repeater 控件	/284
8.4.1	Repeater 控件概述	/284
8.4.2	在 Repeater 控件中显示数据	/286
8.5	实践与练习	/288
<b>第 9 章 ASP.NET 应用程序配置和状态管理 /289</b>		
9.1	配置 ASP.NET 应用程序的工作环境	/289
9.1.1	IIS 服务器的配置	/289
9.1.2	Web.config 文件解读	/294
9.2	ASP.NET 状态管理	/295
9.2.1	ASP.NET 的生命周期	/295
9.2.2	Application 状态及其应用	/297

9.2.3	Session 状态及其应用	/299
9.2.4	ViewState 状态及其应用	/302
9.2.5	Session、Application 和 ViewState 的比较	/305
9.3	实践与练习	/306
<b>第 10 章 ASP.NET AJAX 服务器端编程 /307</b>		
10.1	ASP.NET AJAX 基础	/307
10.1.1	ASP.NET AJAX 的基本概念和特点	/307
10.1.2	安装和配置 ASP.NET AJAX	/308
10.2	ScriptManager 控件的使用	/310
10.3	UpdatePanel 控件的使用	/312
10.3.1	UpdatePanel 控件基础	/312
10.3.2	UpdatePanel 控件应用	/313
10.4	UpdateProgress 控件的使用	/317
10.4.1	UpdateProgress 控件基础	/317
10.4.2	UpdateProgress 控件应用	/318
10.5	Timer 控件的使用	/320
10.6	实践与练习	/321
<b>第 11 章 注册及登录验证模块设计 /322</b>		
11.1	系统设计	/322
11.2	关键技术	/322
11.3	开发过程	/325
11.3.1	数据库设计	/325
11.3.2	配置文件 Web.config	/325
11.3.3	公共类	/325
11.3.4	模块设计	/327
<b>第 12 章 新闻发布系统 /335</b>		
12.1	关键技术	/336
12.1.1	GridView 控件中绑定数据并实现分页	/336

12.1.1 数据库设计 /337	12.1.2 DataList 控件中绑定数据并实现 分页 /336
12.2 开发过程 /337	
12.2.1 数据库设计 /337	
12.2.2 配置 Web.config /337	
12.2.3 公共类 /338	
12.2.4 后台登录设计 /339	
12.2.5 后台新闻管理设计 /341	
12.2.6 前台模块设计 /347	
参考文献 /355	

# 第 1 章 ASP.NET 基础

## 1.1 C/S 与 B/S 架构体系

架构的思想源自 1968 年 Dijkstra 的工作。架构设计出现的背景是需要进行超越算法和数据结构一级的设计,以适应软件规模和复杂性的增长。C/S 和 B/S 是当今世界开发模式技术架构的两大主流技术,C/S 是美国 Borland 公司最早研发的、基于客户机/服务器的模式;B/S 是美国微软公司研发的、基于浏览器/服务器的模式。目前,开发出的很多产品都是基于 C/S 或 B/S 技术的。

### 1.1.1 C/S 架构

#### 1. C/S 架构介绍

C/S 架构是一种典型的两层架构,全称是 Client/Server,即客户端/服务器端架构。其客户端包含一个或多个在用户的计算机上运行的程序,而服务器端有两种:一种是数据库服务器端,客户端通过数据库连接访问服务器端的数据;另一种是 Socket 服务器端,服务器端的程序通过 Socket 与客户端的程序通信。

C/S 架构也可以看做是胖客户端架构。因为客户端需要实现绝大多数的业务逻辑和界面展示。这种架构中,作为客户端的部分需要承受很大的压力,因为显示逻辑和事务处理都包含在其中,通过与数据库的交互(通常是 SQL 或存储过程的实现)来达到持久化数据,以此满足实际项目的需要。

#### 2. C/S 架构的优缺点

C/S 架构的优点如下。

- (1) C/S 架构的界面和操作可以很丰富。
- (2) 安全性能可以很容易保证,容易实现多层认证。
- (3) 由于只有一层交互,因此响应速度较快。

C/S 架构的缺点如下。

- (1) 适用面窄,通常用于局域网中。
- (2) 用户群固定。由于程序需要安装才可使用,因此不适合面向一些不可知的用户。
- (3) 维护成本高,发生一次升级,则所有客户端的程序都需要改变。

### 1.1.2 B/S 架构

#### 1. B/S 架构介绍

B/S 架构的全称为 Browser/Server,即浏览器/服务器架构。Browser 指的是 Web 浏览器,极少数事务逻辑在前端实现,但主要事务逻辑在服务器端实现,Browser 客户端、Web App 服务器端和 DB(数据库)端构成所谓的三层架构。B/S 架构的系统无须特别安装,有 Web 浏览器即可。

B/S 架构中,显示逻辑交给了 Web 浏览器,事务处理逻辑则放在 Web App 上,这样就避免了庞大的胖客户端,减少了客户端的压力。因为客户端包含的逻辑很少,因此也被称为瘦客户端。

## 2. B/S 架构的优缺点

B/S 架构的优点如下。

- (1) 客户端无须安装,有 Web 浏览器即可。
- (2) B/S 架构可以直接放在广域网上,通过一定的权限控制实现多客户访问的目的,交互性较强。
- (3) BS 架构无须升级多个客户端,升级服务器即可。

B/S 架构的缺点如下。

- (1) 在跨浏览器上,B/S 架构不尽如人意。
- (2) 表现要达到 C/S 程序的程度需要花费不少精力。
- (3) 在速度和安全性上需要花费巨大的设计成本,这是 B/S 架构的最大问题。

### 1.1.3 C/S 与 B/S 的区别

C/S 是建立在局域网基础上的,B/S 是建立在广域网基础上的。主要区别在下面几个方面。

#### 1. 硬件环境不同

C/S 一般建立在专用网络上,小范围网络环境,局域网之间再通过专门服务器提供连接和数据交换服务;B/S 建立在广域网之上,不必是专门的网络硬件环境,例如,电话上网,租用设备,信息可以自己管理,有着比 C/S 更强的适应范围,一般只要有操作系统和浏览器就行。

#### 2. 对安全要求不同

C/S 一般面向相对固定的用户群,对信息安全的控制能力很强,一般高度机密的信息系统采用 C/S 结构较为适宜,而仅通过 B/S 发布部分可公开信息;B/S 建立在广域网之上,对安全的控制能力相对较弱,面向的是不可知的用户群。

#### 3. 对程序架构不同

C/S 程序更加注重流程,可以对权限多层次校验,对系统运行速度可以较少考虑。B/S 对安全以及访问速度的多重考虑,建立在需要更加优化的基础之上,比 C/S 有更高的要求。B/S 结构的程序架构是发展趋势,从 Microsoft 的 .NET 系列的 BizTalk、Exchange 等,全面支持网络的构件搭建的系统。Sun 和 IBM 推出的 JavaBean 构件技术等,使 B/S 更加成熟。

#### 4. 软件重用不同

C/S 程序可以整体性考虑,构件的重用性达不到 B/S 结构的要求;B/S 对多重结构要求构件具有相对独立的功能,能够相对较好地重用这些构件。

#### 5. 系统维护不同

C/S 程序由于整体性要求,必须整体考察以处理出现的问题和系统升级,升级较难,有可能需要再做一个全新的系统;B/S 构件的组成方便了构件个别更换,可以实现系统的无缝升级,使系统维护开销减到最小,用户从网上自己下载安装就可以实现升级。

## 6. 处理问题不同

C/S 程序处理用户界面是固定的,在相同区域安全要求高,需求与操作系统相关,都是相同的系统;B/S 建立在广域网上,面向不同的用户群,地域分散,与操作系统关系最小。

## 7. 用户接口不同

C/S 多是建立在 Windows 平台上,表现方法有限,对程序员普遍要求较高;B/S 建立在浏览器上,有更加丰富和生动的表现方式与用户交流,并且开发成本较低。

# 1.2 静态网页与动态网页

## 1.2.1 静态网页技术

网页一般分为静态网页和动态网页两大类,相应的网页开发技术也就分为静态网页技术和动态网页技术。

静态网页是指用纯 HTML 代码编写的网页,并保存为 .html 或 .htm 的文件形式。这种用纯 HTML 代码编写的网页在制作完成后,任何人在任何时候采用任何方式浏览该页面,所看到的浏览结果都是相同的。因此,这种网页的内容更新较为烦琐,必须是设计制作好之后用专门的软件上传到服务器上才能更新。例如,网站的栏目有些是长时间不变的,像这样的页面宜采用静态网页来实现,且浏览的速度比较快。静态网页适合于一些产品规格恒定,不轻易变更的小规模公司,其作用主要是用于配合传统媒体进行广告宣传,适用于一般更新较少的展示型网站。

在 HTML 格式的网页上,也可以出现各种动态的效果,如 GIF 格式的动画、Flash、滚动字幕等,这些“动态效果”只是视觉上的,与下面介绍的动态网页是不同的概念。

## 1.2.2 动态网页技术

动态网页就是根据用户的请求,由服务器动态生成的网页,用户发出请求后,从服务器上获得生成的动态结果,并以网页的形式显示在浏览器中。在浏览器发出请求指令之前,网页中的内容其实并不存在,这就是其动态名称的由来。换句话说,浏览器中看到的网页代码原先并不存在,而是由服务器生成,根据不同用户的不同要求,服务器返回的页面可能并不一致。

例如,在 Google 上搜索信息时,得到的一个搜索结果的页面,该页面的内容是经过整合后变成静态网页返回,而且是动态生成的。

动态网页的网页文件里包含程序代码,通过后台数据库与 Web 服务器的信息交互,由后台数据库提供实时数据更新和数据查询服务。这种网页的后缀名称一般根据不同的程序设计语言不同,如常见的有 .ASP、.JSP、.PHP、.PERL、.CGI 等形式为后缀。动态网页能够根据不同时间和不同访问者而显示不同的内容。常见的有 BBS、留言板、论坛、聊天室、计数器、校友录和购物系统等通常用动态网页实现。动态网页的制作比较复杂,需要用到 ASP、PHP、JSP 和 ASP.NET 等专门的动态网页设计语言。

## 1.2.3 静态网页和动态网页的特点比较

静态网页和动态网页各有特点,网站采用动态网页还是静态网页主要取决于网站的功能需求和网站内容的多少,如果网站功能比较简单,内容更新量不是很大,采用纯静态网页



的方式会更简单,反之,一般要采用动态网页技术来实现。

静态网页是网站建设的基础,静态网页和动态网页之间也并不矛盾,为了网站适应搜索引擎检索的需要,即使采用动态网站技术,也可以将网页内容转化为静态网页发布。

动态网站也可以采用动静结合的原则,适合采用动态网页的地方用动态网页,如果有必要使用静态网页,则可以考虑用静态网页的方法来实现。在同一个网站上,动态网页内容和静态网页内容同时存在也是很常见的。

### 1. 静态网页的特点

(1) 静态网页的每个网页都有一个固定的 URL,且网页 URL 以 HTM、HTML、SHTML 等常见形式为后缀。

(2) 网页内容一经发布到网站服务器上,无论是否有用户访问,每个静态网页的内容都是保存在网站服务器上的,也就是说,静态网页是实实在在保存在服务器上的文件,每个网页都是一个独立的文件。

(3) 静态网页的内容相对稳定,因此容易被搜索引擎检索。

(4) 静态网页的交互性较差,在功能方面有较大的限制。

(5) 静态网页没有数据库的支持,在网站制作和维护方面工作量较大,因此,当网站信息量很大时完全依靠静态网页制作方式比较困难。

### 2. 动态网页的特点

(1) 动态网页以数据库技术为基础,可以大大降低网站维护的工作量。

(2) 采用动态网页技术的网站可以实现更多的功能,如用户注册、用户登录、在线调查、用户管理、订单管理等。

(3) 动态网页实际上并不是独立存在于服务器上的网页文件,只有当用户请求时服务器才返回一个完整的网页。

## 1.2.4 动态网页的发展阶段

与静态网页相比,动态网页的处理上多了一个处理代码的过程。用什么方式来处理代码,在不同的历史时期采用了不同的技术,大体上可以划分为 3 个阶段。

### 1. CGI 阶段

CGI(Common Gateway Interface)代表服务器端的一种通用(标准)接口。每当服务器接到客户更新数据的要求以后,利用这个接口去启动外部应用程序来完成各类计算、处理或访问数据库的工作,处理完后将结果返回 Web 服务器,再返回浏览器。外部应用程序是用 C、C++、Perl、Pascal、Java 或其他语言编写的程序,程序运行在独立的地址空间中。随着 ISAPI(用于 Internet Explorer 浏览器)和 NSAPI(用于 Netscape 浏览器)技术的出现,外部应用程序改用动态链接库(DLL),被载入 Web 服务器的地址空间运行,并且用“线程”代替“进程”,因而显著地提高了运行效率。但不论是 CGI 还是 ISAPI 或 NSAPI,都需要编写外部应用程序,而编写外部应用程序并不是一件容易的事情。从开发人员的角度来讲,这种开发方式并没有带来开发上的方便。

### 2. 脚本语言阶段

该阶段出现了许多优秀的脚本语言,如 ASP、PHP、JSP 等。脚本语言的出现极大地简化了动态网站开发的难度,特别是 ASP 和 PHP 学习简单、功能强大,成为许多网站开发者