

《ASP 网络编程从入门到精通》 清华大学出版社

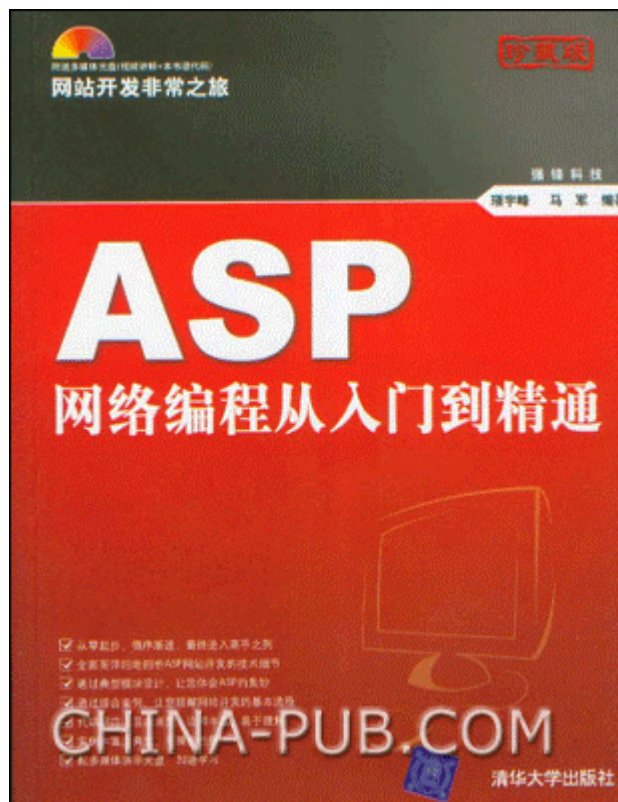
作者：强锋科技

官方网址：<http://www.douban.com/group/19080/>

购书网址：

<http://www.china-pub.com/computers/common/info.asp?id=28800>

<http://www.dearbook.com.cn/book/101886>



特点：

- 面向 ASP 零基础读者，循序渐进
- 全面剖析 ASP 技术细节
- 用代码描述各个知识点，操作性强
- 通过典型模块设计，体会 ASP 的奥妙
- 通过网上商城购物系统，增加项目开发经验
- 配有多媒体光盘，加速学习

适合的读者

- ✓ 网页专业设计制作人员
- ✓ 网站维护人员
- ✓ 网页制作爱好者

- ✓ 大学/大专/中专的教材
- ✓ 社会培训教材
- ✓ 毕业设计的学生

目录

上篇 ASP 网络编程基础

第1章 创建服务器环境

- 1.1 准备互动网页的执行环境
 - 1.1.1 选择ASP的原因
 - 1.1.2 如何执行ASP的程序
- 1.2 架设网站服务器
 - 1.2.1 PWS简介
 - 1.2.2 PWS的安装
 - 1.2.3 PWS的架设
 - 1.2.4 IIS简介
 - 1.2.5 IIS的安装
 - 1.2.6 IIS的架设
 - 1.2.7 IIS的虚拟目录设置
 - 1.2.8 虚拟目录的删除
- 1.3 测试网站服务器
- 1.4 本章小结

第2章 ASP基础

- 2.1 第一个ASP程序
- 2.2 ASP的技术特点和工作原理
 - 2.2.1 ASP的技术特点
 - 2.2.2 ASP的工作原理
 - 2.2.3 ASP提供的内建对象
- 2.3 申请域名
 - 2.3.1 域名的分类
 - 2.3.2 域名申请的流程
- 2.4 发布网站
- 2.5 本章小结

第3章 使用VBScript进行动态网页设计

- 3.1 VBScript语言概述
 - 3.1.1 VBScript语言的特点
 - 3.1.2 在HTML中使用VBScript
 - 3.1.3 在ASP中使用VBScript
- 3.2 VBScript中的变量
 - 3.2.1 简单变量的声明
 - 3.2.2 数组变量的声明
 - 3.2.3 变量的赋值
 - 3.2.4 变量的命名规则
 - 3.2.5 变量的作用域和存活期
- 3.3 VBScript中的运算符
 - 3.3.1 算术运算符

- 3.3.2 比较运算符
 - 3.3.3 逻辑运算符
 - 3.3.4 运算符的优先级
 - 3.4 VBScript中的数据类型
 - 3.5 在VBScript中定义常量
 - 3.6 赋值语句
 - 3.7 条件语句
 - 3.7.1 If...Then语句
 - 3.7.2 If...Then...Else语句
 - 3.7.3 Select...Case语句
 - 3.8 循环语句
 - 3.8.1 Do ...Loop语句
 - 3.8.2 While...Wend语句
 - 3.8.3 For...Next语句
 - 3.8.4 For Each...Next语句
 - 3.8.5 Exit语句
 - 3.9 VBScript中的过程
 - 3.9.1 Sub过程
 - 3.9.2 Function过程
 - 3.10 本章小结
- 第4章 Response和Request对象
 - 4.1 Request对象在网页设计中的应用
 - 4.1.1 Request对象
 - 4.1.2 读取HTTP查询字符串中的数据
 - 4.1.3 读取浏览器提交的表单数据
 - 4.1.4 获得服务器端的环境变量
 - 4.2 Response对象在网页设计中的应用
 - 4.2.1 Response对象
 - 4.2.2 向浏览器发送数据
 - 4.2.3 利用缓冲区输出数据
 - 4.2.4 实现网页的重定位
 - 4.3 在网页中使用Cookie
 - 4.3.1 Cookie的设置
 - 4.3.2 Cookie的使用
 - 4.4 本章小结
- 第5章 Server对象
 - 5.1 处理运行时间较长的脚本
 - 5.2 执行指定路径下的ASP程序
 - 5.3 输出特殊字符
 - 5.4 获得文件的实际物理路径
 - 5.5 创建服务器组件的对象实例
 - 5.6 本章小结
- 第6章 ASP的常用组件
 - 6.1 使用Ad Rotator组件创建一个多彩的广告

- 6.1.1 创建一个广告数据库
 - 6.1.2 在ASP页面中插入广告
 - 6.1.3 处理广告的超级链接
 - 6.2 使用Browser组件获取客户端浏览器信息
 - 6.3 使用Content Linking组件建立网上在线教程的目录
 - 6.4 使用Page Counter组件设计网站计数器
 - 6.5 使用Email组件发送电子邮件
 - 6.5.1 电子邮件的格式
 - 6.5.2 发送电子邮件
 - 6.6 使用FileSystem组件管理文件夹
 - 6.6.1 查找磁盘驱动器以及指定目录下的文件夹和文件
 - 6.6.2 处理文件夹
 - 6.7 使用FileSystem组件管理文件
 - 6.7.1 创建新文件
 - 6.7.2 移动、删除以及复制文件
 - 6.8 使用FileSystem组件读写文件
 - 6.8.1 打开和关闭一个文件并向文件中添加数据
 - 6.8.2 读取文本文件的全文
 - 6.9 本章小结
- 第7章 Application和Session对象
- 7.1 Application对象在网页设计中的应用
 - 7.1.1 设计一个访问计数器
 - 7.1.2 在系统的启动和中止时设置Application对象
 - 7.2 Session对象在网页设计中的应用
 - 7.2.1 记录用户登录信息
 - 7.2.2 Session对象的标志、结束和超时设置
 - 7.2.3 启动和终止Session对象
 - 7.2.4 设置Session对象的超时时间
 - 7.2.5 记录用户在网站上停留的时间
 - 7.3 本章小结
- 第8章 使用ADO组件访问数据库
- 8.1 SQL简介
 - 8.1.1 数据库查询
 - 8.1.2 查询结果排序
 - 8.1.3 插入、删除和更新记录
 - 8.1.4 存储过程
 - 8.2 ADO概述
 - 8.2.1 ADO的对象
 - 8.2.2 ADO的操作流程
 - 8.3 创建ODBC DSN文件
 - 8.3.1 配置Microsoft Access数据库文件的DSN
 - 8.3.2 配置SQL Server数据库文件DSN
 - 8.3.3 连接数据库

- 8.4 使用Connection对象
 - 8.4.1 Connection对象的属性和方法
 - 8.4.2 打开和关闭数据库连接
 - 8.4.3 执行SQL语句
 - 8.4.4 处理事务
 - 8.4.5 设置连接数据源或提交命令的等待时间
- 8.5 使用Command对象
 - 8.5.1 Command对象的属性和方法
 - 8.5.2 执行SQL语句
 - 8.5.3 调用存储过程
- 8.6 使用RecordSet对象
 - 8.6.1 RecordSet对象中的一些概念
 - 8.6.2 RecordSet对象的属性和方法
 - 8.6.3 创建记录集
 - 8.6.4 显示记录集
 - 8.6.5 实现记录集的分页显示
 - 8.6.6 读取记录到数组中
- 8.7 本章小结

中篇 网络项目模块设计

第9章 搜索引擎

- 9.1 关键技术的实现
 - 9.1.1 模糊搜索技术
 - 9.1.2 多个关键字查询
- 9.2 通过其他大型搜索网站建立的搜索引擎
- 9.3 一个功能基本完善的搜索引擎
 - 9.3.1 系统简介
 - 9.3.2 数据库设计
 - 9.3.3 系统文件组成
 - 9.3.4 网站搜索模块
 - 9.3.5 分类目录模块
 - 9.3.6 网站登录模块
 - 9.3.7 网站修改模块
 - 9.3.8 管理员登录模块
 - 9.3.9 网站管理模块
 - 9.3.10 网站审核模块
 - 9.3.11 分类管理模块
 - 9.3.12 退出系统模块
- 9.4 本章小结

第10章 留言板

- 10.1 关键技术的实现
 - 10.1.1 复选框选项的获取
 - 10.1.2 复选框全部选中的实现
- 10.2 一个功能比较完善的留言板系统

- 10.2.1 系统简介
- 10.2.2 数据库设计
- 10.2.3 系统文件组成
- 10.2.4 查看留言模块
- 10.2.5 发表留言模块
- 10.2.6 管理员登录模块
- 10.2.7 留言管理模块
- 10.2.8 留言回复修改模块
- 10.2.9 留言删除模块
- 10.2.10 留言设置模块
- 10.2.11 退出登录模块
- 10.3 本章小结
- 第11章 网站流量统计系统
 - 11.1 关键技术的实现
 - 11.1.1 ASP工程文件Global.asa
 - 11.1.2 使用Application对象创建简单的网站计数器
 - 11.1.3 用session对象创建简单的网站当前在线人数计数器
 - 11.1.4 使用PageCounter组件创建简单的网站计数器
 - 11.1.5 使用FileSystem组件创建简单网站计数器
 - 11.2 改进的计数器
 - 11.2.1 图形化的计数器
 - 11.2.2 使用数据库的网站计数器
 - 11.3 网站流量统计系统的实现
 - 11.3.1 系统简介
 - 11.3.2 数据库设计
 - 11.3.3 系统文件构成
 - 11.3.4 计数器界面实现
 - 11.3.5 详细流量统计的实现
 - 11.4 本章小结
- 第12章 用户管理及用户登录
 - 12.1 关键技术的实现
 - 12.1.1 验证码简介
 - 12.1.2 md5数据加密简介
 - 12.1.3 表单验证的实现
 - 12.1.4 表单递交的合法性检测
 - 12.2 一个最基本的注册模块
 - 12.3 用户管理及用户登录系统实现
 - 12.3.1 系统简介
 - 12.3.2 数据库设计
 - 12.3.3 系统文件组成
 - 12.3.4 用户登录模块
 - 12.3.5 限制普通用户的权限
 - 12.3.6 系统主界面
 - 12.3.7 用户管理模块

- 12.3.8 添加用户模块
 - 12.3.9 修改用户模块
 - 12.3.10 删除用户模块
 - 12.3.11 修改密码模块
 - 12.3.12 退出登录模块
- 12.4 本章小结
- 第13章 发送邮件
 - 13.1 发送邮件简介
 - 13.1.1 电子邮件简介
 - 13.1.2 SMTP简介
 - 13.2 关键技术实现
 - 13.2.1 普通邮件的发送
 - 13.2.2 带附件邮件的发送
 - 13.2.3 使用CDONTS组件发送邮件
 - 13.2.4 使用JMail组件发送邮件
 - 13.2.5 使用w3 JMail组件发送邮件
 - 13.2.6 使用ASPEmail组件发送邮件
 - 13.2.7 ASPMail
 - 13.3 邮件发送系统的实现
 - 13.3.1 系统简介
 - 13.3.2 数据库设计
 - 13.3.3 系统文件构成
 - 13.3.4 设计用户登录模块
 - 13.3.5 设计用户注册模块
 - 13.3.6 邮件管理主界面模块
 - 13.3.7 邮件发送模块
 - 13.3.8 已发送邮件查看模块
 - 13.3.9 邮件系统设置模块
 - 13.3.10 基本信息修改模块
 - 13.3.11 邮件地址管理模块
 - 13.3.12 退出登录模块
 - 13.4 本章小结
- 第14章 聊天室
 - 14.1 关键技术的实现
 - 14.1.1 网页的定时刷新
 - 14.1.2 使用Application和Session对象刷新聊天室
 - 14.1.3 私聊的实现
 - 14.2 一个功能比较完善的聊天室的实现
 - 14.2.1 系统简介
 - 14.2.2 数据库设计
 - 14.2.3 系统文件组成
 - 14.2.4 用户登录模块
 - 14.2.5 用户注册模块
 - 14.2.6 聊天室主界面

- 14.2.7 在线用户显示页面
 - 14.2.8 聊天发言提交页面
 - 14.2.9 聊天内容显示页面
 - 14.2.10 用户退出页面
- 14.3 本章小结
- 第15章 网络投票
 - 15.1 系统简介
 - 15.2 关键技术的实现
 - 15.2.1 所投选项的获取
 - 15.2.2 禁止重复投票
 - 15.3 一个简单投票计数器的实现
 - 15.3.1 投票模块
 - 15.3.2 计数模块
 - 15.3.3 显示模块
 - 15.4 一个功能完善的投票系统的实现
 - 15.4.1 系统介绍
 - 15.4.2 数据库设计
 - 15.4.3 系统文件构成
 - 15.4.4 用户投票模块
 - 15.4.5 投票统计模块
 - 15.4.6 结果显示模块
 - 15.4.7 管理员登录模块
 - 15.4.8 设置当前投票模块
 - 15.4.9 增加投票模块
 - 15.4.10 修改投票模块
 - 15.4.11 删除投票模块
 - 15.4.12 信息修改模块
 - 15.5 本章小结

下篇 完整网站系统构建实例

第 16 章 网上商城购物系统

- 16.1 总体系统设计
 - 16.1.1 设计目标
 - 16.1.2 设计方案
 - 16.1.3 模块功能
- 16.2 数据库分析
 - 16.2.1 数据库的需求
 - 16.2.2 数据库ER图分析
 - 16.2.3 数据库设计
- 16.3 导航条应用
 - 16.3.1 首部及尾部导航条
 - 16.3.1 ASP#include指令

- 16.3.2 代码的实现
- 16.4 注册和登录
 - 16.4.1 填写注册信息
 - 16.4.2 提交注册信息
 - 16.4.3 用户登录和注销
 - 16.4.4 MD5加密算法介绍
 - 16.4.5 实现验证码的登录
- 16.5 用户管理中心
 - 16.5.1 进入用户中心
 - 16.5.2 修改个人资料
 - 16.5.3 修改个人密码
 - 16.5.4 取回用户密码
 - 16.5.5 “我的订单”界面
 - 16.5.6 我的购物车的实现
 - 16.5.7 收货人信息
- 16.6 商品查询
 - 16.6.1 简单查询
 - 16.6.2 高级查询窗口
 - 16.6.3 高级查询处理
- 16.7 销售信息统计
 - 16.7.1 销售排行榜榜
 - 16.7.2 关注排行榜榜
- 16.8 收藏和购买商品
 - 16.8.1 浏览商品
 - 16.8.2 购买商品
 - 16.8.3 填写收货人信息
 - 16.8.4 订单提交
- 16.9 后台管理员界面
 - 16.9.1 管理界面菜单栏
 - 16.9.2 管理界面工作区
- 16.10 商品分类管理
 - 16.10.1 商品大类信息浏览
 - 16.10.2 修改商品大类信息
 - 16.10.3 添加商品大类
 - 16.10.4 商品小类信息浏览
 - 16.10.5 商品小类信息修改和删除
 - 16.10.6 添加商品小类
 - 16.10.7 商品类别转移
- 16.11 添加及修改商品
 - 16.11.1 添加新的商品
 - 16.11.2 处理新的商品入库
 - 16.11.3 浏览商品
 - 16.11.4 修改商品的信息
 - 16.11.5 删除所选商品

- 16.12 管理商品订单和评论
 - 16.12.1 查看订单信息
 - 16.12.2 查询订单信息
 - 16.12.3 修改订单状态
 - 16.12.4 浏览所有评论
 - 16.12.5 审核或者删除评论
- 16.13 用户管理
 - 16.13.1 前台用户信息
 - 16.13.2 查询指定用户
 - 16.13.3 修改所选用户信息
 - 16.13.4 删除指定用户
 - 16.13.5 后台用户浏览
 - 16.13.6 后台用户修改和删除
 - 16.13.7 添加后台用户
 - 16.13.8 权限控制
- 16.14 信息管理
 - 16.14.1 添加新闻
 - 16.14.2 新闻修改删除
 - 16.14.3 首页公告设置
 - 16.14.4 查看意见反馈
 - 16.14.5 留言板块管理
- 16.16 系统发布
- 16.17 本章小结



第 2 章

ASP 基础

- » 第一个 ASP 程序
- » ASP 的技术特点和工作原理
- » 申请域名
- » 发布网站

在本章，将讲述 ASP 的一些基础知识，帮助读者了解 ASP 的特点、工作原理和对象构成。为了便于后期学习和调试，本章首先讲解如何申请域名以及如何申请网站空间。本章内容不多，但是以后学习 ASP 的基础，读者要熟练掌握。

2.1 第一个 ASP 程序

下面是一个简单的 ASP 程序，它在网页上显示五个“Hello World”字符串，并且使字体变得越来越大。

<code><html></code>	'<html>是HTML文件的起始标志
<code><head></code>	'HTML文件头部起始标志
<code><title>Hello World</title></code>	'设置网页标题
<code></head></code>	'完成对HTML文件头部的定义
<code><body></code>	'HTML文件主体部分起始标志
<code><%</code>	'表示下面要插入ASP代码
<code>For i=2 to 6</code>	'ASP中的循环语句
<code>%></code>	
<code><font size=<%=i%>></code>	'设置字体大小
<code>Hello World!</code>	'要显示的文字
<code>
</code>	'换行
<code></code>	'结束要求的字体大小
<code><%</code>	
<code>Next</code>	'For循环结束标志
<code>%></code>	
<code></body></code>	'完成对HTML文件主体部分的定义
<code></html></code>	'完成对HTML文件的定义

程序运行结果如图 2-1 所示。



图2-1 程序运行结果

在上面程序中，使用“<% %>”符号将 ASP 脚本程序代码包括起来。

但是在浏览器端看到的并不是上面的程序代码。因为 ASP 代码是运行在服务器端的，ASP 代码执行后产生浏览器能够识别的 HTML 代码，并返回给用户浏览器。浏览器端看到的程序源代码如下：

```
<html>
<head>
  <title>Hello World</title>
</head>
<body>
  <font size=2>
    Hello World!<br>
  </font>
  <font size=3>
    Hello World!<br>
  </font>
  <font size=4>
    Hello World!<br>
  </font>
  <font size=5>
    Hello World!<br>
  </font>
```

```
<font size=6>
Hello World!<br>
</font>
</body>
</html>
```

那么究竟是怎样执行这个 ASP 程序的呢？

首先是将编写好的程序保存为 hello.asp 文件并存放于 Web 服务器主目录下。在第一章 IIS 的架设中，Web 服务器主目录设为“系统盘:\inetpub\wwwroot”，在此，可以将其设为“d:\inetpub\wwwroot”。将文件放到主目录下之后，就可以在浏览器的 IE 地址栏输入如下的网址来执行上面的 ASP 程序。

http://localhost/hello.asp

服务器收到这个请求后，使用 ASP 的默认语言解释器解释 ASP 程序，而并不将 ASP 程序传到用户浏览器。在用户端，浏览器上显示的页面是解释之后的结果页面，而解释之后生成的 HTML 文件代码就是上面列举的程序源代码，是标准的 HTML 格式的代码。

2.2 ASP 的技术特点和工作原理

这一节将介绍 ASP 的一些基础知识，包括 ASP 的技术特点、工作原理以及内建对象。

2.2.1 ASP 的技术特点

ASP 内含于 IIS 之中，通过 ASP 大家可以结合 HTML 网页、ASP 指令集和 ActiveX 控件建立动态、交互且高效的 Web 服务器应用程序。由于 ASP 是与脚本语言无关的，因此不用担心客户端的浏览器是否能运行用户所编写的代码。当程序执行完毕后，服务器只将执行的结果返回给客户端浏览器，这样就减轻了客户端浏览器的负担，大大提高了交互的速度。归纳 ASP 的特点，大致有以下几条：

无需编译 ASP脚本集成于HTML当中，容易生成，无需编译或链接即可直接解释执行。

易于生成 使用普通的文本编辑器(如WINDOWS下的记事本)就可以进行.asp页面的编辑设计。

独立于浏览器 用户端只要使用可执行HTML码的浏览器，即可浏览ASP所设计的网页内容。因为ASP的运行是与浏览器无关的。

面向对象 在ASP脚本中可以方便地引用系统组件和ASP的内置组件，还能通过定制ActiveX服务器组件来扩充功能。

完成网站应用程序快 使用了VBScript等简单的脚本语言，可以快速完成网站应用程序的编写。

源代码不会外漏 ASP脚本在服务器端执行，传到用户浏览器的只是ASP的执行结果所生成的常规HTML代码,这样可保证辛辛苦苦编写出来的程序代码不会被他人盗取。

2.2.2 ASP 的工作原理

比较用户通过客户端浏览器请求 ASP 程序的过程和访问普通 HTML 页面的过程中的不同，可以清楚的了解 ASP 的工作原理。

对 HTML 页面的访问一般用户都比较了解。首先将 HTML 文件的 Internet 地址输入到客户端浏览器地址栏，然后浏览器将网页请求发送到 Web 服务器，Web 服务器收到请求之后，通过扩展名.html 或者.htm 判断出 HTML 文件的请求，然后将适当的 HTML 文件从服务器

磁盘中取出并返回到客户端浏览器。在客户端，浏览器对 HTML 文件解释之后将解释的结果显示出来，用户看到的就是这个结果。

而 ASP 的工作原理是，当用户申请一个*.asp 文件时，Web 服务器响应该请求，并调用服务器上的 ASP 解释器，解释被申请的文件，生成相对简单的页面返回给用户浏览器。这些生成的页面是纯 HTML 文件，所以一般浏览器都能够浏览这些页面，而实际的用户申请的*.asp 文件并不直接返回给浏览器。。

2.2.3 ASP 提供的内建对象

在目前的 ASP 版本中，主要有六个内建对象。这六个内建对象提供了许多方法和属性，大大方便了用户编写 Web 应用程序。下面将简单介绍这六个内建对象。

Request对象：读取用户信息。用来取得任何由HTTP请求传递过来的信息，包括使用POST和GET传递的参数，以及从服务器和客户端认证所传递的Cookie等。

Response对象：传送信息给用户。可以使用它的方法输出信息到浏览器，或将使用者转移到另一个URL，并可以控制内容形态和设定Cookie值。

Server对象：控制ASP的执行环境。提供存取Web服务器的方法和属性。

Session对象：存储用户对话的相关信息。此对象仅适用于一个用户，可以用它来记录该用户的一些信息，并为每一个用户保留一个SessionID。

Application对象：用来为应用程序的所有用户设置属性，并且传递信息给用户。

ObjectContext对象：提供在页面内进行事务处理的功能。

使用以上的内建对象，可以得到浏览器传过来的信息，向浏览器输出信息，记录单一用户，创建全体用户操作量，以及创建组件等。在后面的章节中，将具体介绍它们的用法。

2.3 申请域名

所谓的域名就是 Internet 网络上的一个服务器或一个网络系统的名字，在全世界，没有重复的域名。

2.3.1 域名的分类

域名的形式是以若干个英文字母或数字组成，由“.”分隔成几部分，如 SOHU.com 就是一个域名。注册域名之后，其他人才能通过网络访问该网站。那么如何才能注册一个域名呢？现在网上有许多网站提供了域名注册服务。总的来说域名注册有免费和收费两种。建议申请收费域名，推荐使用中国互联网络信息中心授权的顶级.CN 域名注册机构——新网互联（dns.com.cn）注册域名。

域名注册时，首先要明确注册什么样的域名，然后根据网上提供的域名注册步骤就可以完成域名注册了。完成域名注册并开通后，就可以使用自己的域名建立网站了。

注意：在网上注册域名时，域名的命名要符合下面的共同规则。

域名中只能包含以下字符：

- 26个英文字母；
- “0, 1, 2, 3, 4, 5, 6, 7, 8, 9”十个数字；
- “-”（英文中的连词号）

域名中字符的组合规则：

- 在域名中，不区分英文字母的大小写；
- 对于一个域名的长度是有一定限制的。

CN 下域名命名的规则为：

遵照域名命名的全部共同规则；

只能注册三级域名，三级域名用字母（A-Z，a-z，大小写等价）、数字（0-9）和连接符（-）组成，各级域名之间用实点（.）连接，三级域名长度不得超过20个字符；

不得使用，或限制使用以下名称（下表列出了一些注册此类域名时需要提供的材料）：

- 注册含有“CHINA”、“CHINESE”、“CN”、“NATIONAL”等要经国家有关部门（指部级以上单位）正式批准；
- 公众知晓的其他国家或者地区名称、外国地名、国际组织名称不得使用；
- 县级以上（含县级）行政区划名称的全称或者缩写要经相关县级以上（含县级）人民政府正式批准；
- 行业名称或者商品的通用名称不得使用；
- 他人已在中国注册过的企业名称或者商标名称不得使用；
- 对国家、社会或者公共利益有损害的名称不得使用。

经国家有关部门（指部级以上单位）正式批准和相关县级以上（含县级）人民政府正式批准是指，相关机构要出据书面文件表示同意 XXXX 单位注册 XXX 域名。如：要申请 beijing.com.cn 域名，则需提供北京市人民政府的批文。

2.3.2 域名申请的流程

1. 个人用户申请流程

个人空间的申请可以根据个人用户的实际情况申请一些免费的域名作为网络个人标识。

（1）根据个人的情况选择确定申请域名的类型。这里使用中华网的免费的域名作为例子为读者介绍域名申请的过程。

（2）确定申请的域名后需要对所要申请的域名进行查询，确定域名是否已经被其他的公司或个人注册，如果没有被注册，便可以申请注册。进入域名申请界面，单击【查询】按钮，如图 2-1 所示。



图2-1 域名申请界面

（4）如果域名可以使用，则会弹出如图 2-2 所示的界面，证明用户可以使用所申请的域名。



图2-2 域名查询界面

（5）单击【注册】链接，会要求用户填写一张租用域名的用户的详细信息的电子版的表格，如图 2-3 所示。

注意：如果是公司申请的情况，需要和域名申请公司签署合同，标明双方的权利和义务。个人申请则可以使用域名申请公司的注册信息作为维护双方权利，确定双方义务的文件。

(6) 单击【提交】按钮，弹出如图 2-4 所示的完成界面，即证明用户已经申请到所需要的域名。

用户名：	<input type="text"/>	*《即域名http://www.kdnet.com/用户名，2-12个英文字母》
密码：	<input type="password"/>	*《您管理空间时所需的密码》
确认密码：	<input type="password"/>	*《您管理空间时所需的密码》
真实姓名：	<input type="text"/>	*《必须是2-4个中文字符，使用英文姓名或拼音字母将不被通过》
性别：	男 <input type="checkbox"/> 女 <input type="checkbox"/>	*
出生日期：	1990 年 01 月 01 日	*《您需要它来验证您输入信息的准确性》
身份证号码：	<input type="text"/>	*《请填写您真实的身份证号》
联系地址：	<input type="text"/>	*《请务必认真填写您的详细地址》
邮政编码：	<input type="text"/>	*《所在地区的邮政编码》
电话：	<input type="text"/>	*《可选择填写》
电子邮箱：	<input type="text"/>	*《请填写您的邮箱，请不要用hotmail的邮箱》
网站分类：	<input type="text"/>	*《正确选择您的网站类型》
网站介绍：	<input type="text"/>	*《可选择填写》
网站名称：	<input type="text"/>	*《注：网站主题/标题/名称》
<input type="button" value="提交"/> <input type="button" value="重置"/>		

图2-3 用户信息表格



图2-4 完成界面

2. 企业用户申请流程

企业用户和个人用户的最大区别在于企业用户需要交纳一定的费用来保证域名服务的稳定。需要注意的有如下几点。

- 企业用户的申请和个人用户的申请的区别在于企业用户需要和域名申请公司签署相应的合同并缴纳一定的费用。
- 在确认合同生效后，使用域名的公司需要向域名申请公司缴纳费用。如果是在网站上直接申请，可以通过邮局汇款或者银行转账的形式向对方付款。用户应妥善保管汇款回持或者银行回持。
- 在域名申请公司受到用户所缴纳的费用后，会开通域名服务，这是域名即可以使用了。

2.4 发布网站

网站发布包括局域网上的网站发布和 Internet 上的网站发布。在局域网内发布网站比较简单。只要在 Windows 2000 上安装 IIS 或在 Windows 98 上安装 PWS，局域网内的其他用户就可以直接通过用户的机器名或 IP 地址访问用户的网站。

在 Internet 上发布网站，首先要申请网站空间，如果已经申请域名，则将所有 HTML 文件或 ASP 文件上传到域名对应的服务器主目录上，然后其他人就可以通过该域名直接访问用户的网站了。

申请域名后，就可以申请网站空间。建议用户选择信誉较好的的网络公司进行网络空间申请的工作。具体流程如下。

1. 个人申请空间

个人空间的申请可以根据个人用户的实际情况申请一些免费的空间作为个人网站数据存储基地。

(1) 根据个人情况选择确定申请网站空间的类型。其中包括要提供什么样的服务,使用何种数据库。在根据调查的结果决定使用空间的类型大小、操作系统。

(2) 选择出租网络空间的公司,在网络上有很多可以申请网络空间的公司,进入该公司的网站或者通过电话可以联系该公司。本文中以中华网的免费的域名作为例子介绍空间申请的流程。

(3) 单击【注册拥有强大功能的个人主页】后进入免费空间申请的界面,如图 2-5 所示。

(4) 阅读并同意公司的条款后,单击【同意】按钮,弹出【个人空间注册信息】窗口,如图 2-6 所示。



图2-5 免费空间申请界面

图2-6 【个人空间注册信息】窗口

(5) 正确填写所有要求的信息后,单击【注册】按钮,弹出如图 2-7 所示的完成界面,即证明用户已经申请到所需要的空间。



图2-7 完成界面

(6) 用户使用网络公司提供的用户空间的 FTP 账号和密码,上传网站文件。同时,在用户使用的注册邮箱中会收到网络公司发送的电子邮件。可以通过电子邮件了解自己所申请的网路空间的详细信息。

注意：由于同时使用浙江都市网的尔击域名和网络空间,所以不需要对空间和域名作绑定。如果所申请的空间、域名不属于一家公司可以要求域名服务商作域名和空间的绑定。

2. 企业申请空间

企业用户和个人用户的最大区别在于企业用户需要交纳一定的费用来保证网络服务的稳定。公司申请空间时,需要和出租空间的公司签署合同,标明双方的权利和义务。在确认合同生

效后,使用网络空间的公司需要向出租网络空间的公司缴纳费用。如果是在网站上直接申请,可以通过邮局汇款或者银行转账的形式向对方付款。

在出租网络空间的公司受到用户所缴纳的费用后,会在公司的服务器上为用户划分一块用户指定大小的空间,同时会将用户空间的 Ftp 账号和密码发送给用户,这样用户就可以使用这个账号和密码上传网站文件了。将所申请的空间的 IP 地址通知域名申请空间的公司,绑定域名和 IP 地址,这样当网络用户输入域名时就可以直接访问网站了。

2.5 本章小结

在这一章,读者了解到了 ASP 的一些基础知识,包括它的技术特点、工作原理和内建对象。最后读者了解如何注册域名和发布网站。读者应着重把握 ASP 如何进行实现网页的动态更新、域名和网络空间的获取。

第14章

聊天室

- » 对页面进行定时刷新
- » 用户登录和注册模块的实现
- » 显示在线用户
- » 聊天信息的提交和显示
- » 一个功能完善的聊天室的实现

经常上网而没有去过聊天室的人是非常少的。在当今网络时代，聊天室给大家带来了很方便。通过聊天室，可以跟身在远方的好朋友尽情聊天，也可以结交认识网上的很多朋友。特别是一些专业聊天室，如求职等，在里面能够及时的获得许多有用信息。下面将介绍如何编程实现聊天室。首先来看一下聊天室实现的关键技术。

14.1 关键技术的实现

聊天室一般涉及到的技术包括聊天内容的实时刷新显示和网友私聊的实现。这一节将讲述这两方面内容。

14.1.1 网页的定时刷新

在聊天室中，浏览器要不停的刷新，读取最新的用户发言。可以使用下面的语句实现网页的自动刷新。

```
<meta http-equiv="refresh" content="4">
```

这是一条最简单的页面自动刷新脚本，它是 HTML 脚本，其意思是每 4 秒钟刷新一次，执行读取数据库中用户发言并在页面显示的操作。下面的例子在代码中使用了这一语句。每 4 秒钟在页面上显示一次最新时间。具体代码如下：

```
<html>
<head>
  <meta http-equiv="refresh" content="4">           ' 设定刷新时间间隔
  <title>网页定时刷新</title>                     ' 设定网页标题
</head>
<body>
  现在的时间是：<%=now()%>                        ' 显示当前时间
</body>
</html>
```

使用上面语句实现网页的自动刷新，这对于一个小型聊天室来说还是可以的，但是对于一个大型的、人数比较多的聊天室，使用上面语句自动刷新会造成时间和资源的浪费。

14.1.2 使用 Application 和 Session 对象刷新聊天室

从第十一章可以了解到，Application 对象代表了一个 Active Server 应用程序，也就是一个 Web 网页，Session 对象则表示一个用户，代表一个用户对这个页面的一次访问，通过 Application 对象可以让访问它的所有用户共享信息，并可以在 Web 服务器运行期间持久地保存数据，而 Session 对象也可以在用户的一次访问期间持久地保持数据，利用这两个对象，就可以十分方便地刷新聊天室了。

具体原理如下：在网站开始运行，用户还没有进入聊天室发言时，初始化 Application 变量 Gchatnum 为 0，Gchatnum 为用户发言次数，用户发言一次，则 Application("Gchatnum")加 1。当某个用户登录进入聊天室后，创建该用户的一个 Session 变量 Lchatnum。并将 Application("Gchatnum")的值赋给 Session("Lchatnum")，该用户在聊天室聊天过程中，不断判断 Application("Gchatnum")的值与 Session("Lchatnum")的值是否相同，如果不相同，则将数据库中 Application("Gchatnum")>Session("Lchatnum")的聊天记录读出并显示在该用户的浏览器上。

在下一节讲述的聊天室的实现中，聊天内容的刷新就是采用这种方法。这里将不再具体举例。

14.1.3 私聊的实现

在聊天室里闲聊的时候，有时候有些话不想让同一聊天室内其他的人看见，那么聊天室就必须支持私聊的功能。实现私聊的方法很多，下面将介绍利用 Session 和 Application 对象来实

现私聊。

首先来看一下私聊实现的原理：私聊要求发言者和发言对象都能看到私聊的内容，另外，当私聊的对象是所有人时，则所有的人都能看到私聊的内容。因此，可以设置几个 Application 变量来保存私聊信息。例如，Application(“chat”)保存私聊的内容，Application(“chatto”)保存私聊的对象，Application(“owner”)保存发言者。当用户登录聊天室时，使用 Session(“user”)记录用户名，Session(“chat”)记录该用户是否私聊，初始化为 False，表示用户没有私聊。当用户要私聊时，则记录私聊信息，并将 Session(“chat”) 置为 True，表示有私聊。语句如下：

```
'私聊内容
Application("chat")="<font color=blue>"&发言者用户名&"</font> <I>悄悄地对
<font color=#00bb00>&nbsp;"&私聊对象用户名&" </font>耳语</I>:
<font color=#0000aa> "&私聊内容&"</font><BR><BR>"
Application("chatto") = 私聊对象用户名      '私聊对象
Application("owner") = 发言者用户名          '发言者
Session("chat") = "True"                     '存在私聊
```

当 Application 中保存私聊信息后，用户浏览器端会判断 Session(“user”)的值与 Application(“chatto”)或 Application(“owner”)的值是否相同，如果相同，则在浏览器端显示私聊内容。如果不相同，但是 Application(“chatto”)是所有人，则仍然显示私聊内容。否则不会显示任何私聊信息。

以上为私聊实现的原理，其代码的实现将在下面的程序实现中具体讲述。

14.2 一个功能比较完善的聊天室的实现

这一节将实现一个聊天室。在该聊天室内，用户能够看到最新的聊天内容，并且能够实现与某一网友的私聊。

14.2.1 系统简介

该聊天室能够实现多用户注册。该系统主要完成的功能是：

用户登录后，

能够聊天发言；

能够与指定的用户进行私聊；

该系统主要包括三个模块：用户登录模块、新用户注册模块和已注册用户的聊天模块。

用户登录模块要求用户输入用户账号、用户密码和验证码，完成用户的登录。

新用户注册模块要求用户输入登录的用户名、密码、真实姓名和电子邮箱。

已注册用户聊天模块能够完成用户的聊天发言和私聊。

该系统的结构如图 14-1 所示。



图14-1 系统设计结构图

14.2.2 数据库设计

本系统数据库采用 Access 数据库。根据系统的结构图可以看出，该系统用到三个数据表：User 数据表、Online 数据表和 chat 数据表。User 数据表用来保存用户注册信息，包括用户名、密码、真实姓名和电子邮箱四个字段；Online 数据表用来保存在线用户信息，包括用户名和登录时间；chat 数据表用来保存用户聊天内容，包括发言人用户名、发言对象用户名、发言时间和发言内容。

User 数据表的结构见表 14-1。

表14-1 数据表User的结构

字段名称	字段类型	主关键字	说明
ID	自动编号	*	编号
UserName	文本		用户名
Password	文本		用户密码
Name	文本		用户真实姓名
Email	文本		用户电子邮箱

Online 数据表的结构见表 14-2。

表14-2 数据表Online的结构

字段名称	字段类型	主关键字	说明
ID	自动编号	*	编号
UserName	文本		用户名
OnlineTime	日期/时间		登录时间

数据表 chat 的结构见 14-3。

表14-3 数据表chat的结构

字段名称	字段类型	主关键字	说明
ID	自动编号	*	编号
FromName	文本		发言人用户名
ToName	文本		发言对象用户名
PostTime	日期/时间		发言时间
chat	文本		发言内容

14.2.3 系统文件组成

由系统结构图可以看出，该系统主要由以下文件组成：

用户登录模块：

index.asp文件要求用户输入用户名、密码和验证码，完成用户登录。

用户注册模块：

register.asp文件要求用户输入用户账号、密码、确认密码、真实姓名和电子邮箱，完成用户的注册。

已注册用户的聊天室模块：

main.asp文件为聊天室主界面文件。

chatroom.asp文件为聊天室布局文件，该聊天室的基本布局是左侧显示在线用户名，右侧上面部分显示聊天内容，下面用于用户输入聊天内容并发送。

showname.asp文件显示当前在线用户名单。

postmsg.asp文件实现用户输入和发送聊天内容。

showmsg.asp文件显示用户登录之后的所有聊天内容。

topmsg.asp文件比较简单，是为了在聊天室主界面上方显示该聊天室的名称。

logout.asp文件为用户退出登录文件。

公共文件

- conn.asp文件完成与数据库的连接。
- md5.asp文件完成用户密码的加密。
- procedure..asp文件中包含邮件地址格式确认过程。

在该系统中，除了这几个主要文件外，还有 function.asp、safecode..asp、canvas.asp 和 font.asp 文件，这四个文件完成用户登录时验证码的生成与显示。

14.2.4 用户登录模块

用户登录该聊天室后，首先进入登录界面 index.asp。index.asp 文件的设计界面如图 14-2 所示，其中包含表单 form1，用于输入管理员的用户名、密码和验证码。



图14-2 index.asp文件的设计界面

在该界面上，输入用户名的文本域为 UserName，输入密码的文本域为 UserPwd，输入验证码的文本域为 confirm，验证码的生成是调用 function.asp 文件中的 GetSafeCode()过程实现

的。

定义表单的 form1 的语句如下：

```
'check()过程检查用户名、密码和验证码是否输入
<form name ="form1" method = "Post" action = "index.asp?act=login" _
onsubmit = "return Check()">
```

可以看到，表单提交后，仍将在 index.asp 文件中执行，处理表单提交的数据。index.asp 文件中具体的处理表单的代码如下。首先检查用户名、密码和验证码是否输入正确，如果正确将转到 main.asp 页面。

```
<%
If Request.QueryString("act") ="login" Then      '用户登录
    user = Request.Form("UserName")              '读取从表单传递过来的用户名数据
    pass = Request.Form("Password")              '读取从表单传递过来的密码数据
    If user<>" " and pass<>" " Then                '用户名和密码已经填写
        '用函数IsNumeric()判断用户输入的验证码是否为数字
        If not IsNumeric(Request.Form("confirm")) Then
            '如果用户填写的验证码不是数字，则提示用户，并返回到登录窗口
            Response.Write "<script>alert('你输入的验证码为非数字！');_
                            window.location.href('index.asp');</script>"
            Response.End
        End If
        'Session("SafeCode")中保存程序生成的验证码，比较生成的验证码与用户输入的验证码
        '是否相同
        If (int(Session("SafeCode"))=int(Request.Form("confirm"))) Then
            '生成RecordSet对象
            Set rs=Server.CreateObject("Adodb.RecordSet")
            '从数据表User中读取UserName的值为user和用户Pass的值为md5(pass)的
            '记录。md5()函数是一个加密函数，将管理员输入的密码加密后与数据库中的值进
            '行比较
            sql="Select * from User where UserName='&user&'" and _
                Password='&md5(pass)&'"
            rs.open sql,conn,1,3                    '读取满足条件的记录保存在rs中
            If not(rs.bof and rs.eof) Then            '满足条件的记录存在
                '输入的用户名和密码正确
                If md5(pass)=rs("Password") and user=rs("UserName") Then
                    '判断登录的用户是否已在线
                    sql = "Select * from online where UserName = '&user&'"
                    Set OLrs = Server.CreateObject("Adodb.RecordSet")
                    OLrs.open sql,conn,1,1
                    '用户已在线
                    If not (OLrs.EOF and OLrs.BOF) Then
                        '给出提示，并返回到前一页
                        Response.Write("<script>alert('该用户已经存在！');_
                                    history.back();</script>")
                        Response.End
                    Else
                        '用户不在线
                        'Session变量user中保存当前登录的用户名
                        Session("user")=user
                        'Session变量chat保存用户私聊内容
                        Session("chat") = ""
                        'Application("Gchatnum")中的值为当前聊天总记录数
                        Session("Lchatnum") = Application("Gchatnum")
                        OLrs.close                    '关闭RecordSet对象
                        Set OLrs=nothing
                        rs.Close
                        Set rs = nothing
                        '登录成功，转到页面main.asp
```

```

        Response.Redirect "main.asp"
    End If
Else
    '登录不成功，提示用户，并转到页面index.asp
    Response.Write "<script>alert('登陆失败，用户名或密码_
        错误！');window.location.href('index.asp');</script>"
    End If
    conn.Close                '关闭数据库连接
    Set conn=nothing
Else
    '满足条件的记录不存在
    Response.Write "<script>alert( '登陆失败！用户名或密码不存在！_
        ');window.location.href('index.asp');</script>"
    End If
    conn.Close                '关闭数据库连接
    Set conn=nothing
Else
    '验证码输入错误
    Response.Write "<script>alert('登陆失败！验证码输入错误！');_
        window.location.href('index.asp');</script>"
    End If
Else
    '用户名和密码没有输入
    Response.Write "<script>alert('登陆失败！用户名和密码不能为空！');_
        window.location.href('index.asp');</script>"
    End If
%>

```

用于检查用户名、密码和验证码是否输入的 Check()方法的代码如下：

```

<script language="VBScript">
    function check()                '检查用户名、密码和验证码是否输入
        If form1.username.value="" Then '用户名文本域的值为空
            alert("用户名不能为空!") '给出提示
            form1.username.focus() '将用户名文本域置为输入焦点
            return false
        End If
        If form1.password.value="" Then '密码文本域的值为空
            alert("密码不能为空!")
            form1.password.focus() '将密码文本域置为输入的焦点
            return false
        End If
        If form1.confirm.value="" Then '验证码文本域的值为空
            alert("验证码不能为空!")
            form1.confirm.focus() '将验证码文本域置为输入焦点
            return false
        End If
        return true
    End Function
</script>

```

上面是 index.asp 文件的部分代码，index.asp 文件的完整代码运行步骤如下：

- (1) 使用代码<!--#include file="conn.asp"-->连结数据库。
- (2) 使用代码<!--#include file="md5.asp"-->定义 md5()加密函数。
- (3) 使用代码<!--#include file="Function.asp"-->定义生成验证码的函数。
- (4) 使用 Request 对象的 QueryString()方法获取 act 的值，如果 act 的值为 login，则检查用户输入的用户名、密码和验证码是否正确，如果正确，判断登录用户是否已经在线，如果没有在线，则转到聊天室主界面 admin.asp。否则退回到登录界面。
- (5) 如果 act 的值不是 login，则使用 HTML 代码生成用户登录界面，定义表单 form1。index.asp 文件的运行结果如图 14-3 所示。



图14-3 聊天室开始界面

在 index.asp 文件中，如果是新用户，界面下方给出了一个【新用户注册】的超级链接，用户可以单击该链接进入用户注册界面。

14.2.5 用户注册模块

如果是新用户，则由 index.asp 页面转到 register.asp 页面进行新用户的注册。在该页面，用户需要填写登录系统时的用户账号、密码、真实姓名和电子邮箱。register.asp 文件的界面设计如图 14-4 所示。其中包含表单 registerform，用于输入用户需要填写的信息。

在该界面上，输入用户账号的文本域为 UserName，输入用户密码的文本域为 Password，输入确认密码的文本域为 Pwd，输入用户真实姓名的文本域为 Name，输入电子邮箱的文本域为 Email。定义表单 registerform 的代码如下：

```
<Form method="POST" action="register.asp?result=successful" _
name = registerform>
```



图14-4 用户注册页面设计

从代码得知，当表单提交后，将在 register.asp 文件中处理表单提交的数据。在 register.asp 文件中定义表单处理过程 save()过程，首先判断用户输入信息的有效性，然后将输入的信息保存到数据库中。save()过程的具体处理代码如下：

```
<%
```

```

Sub save()                                '定义过程
Dim UserName,UserPwd,Pwd,Name,Email      '声明变量
'读取从表单传递过来的用户名数据,Trim()函数去掉字符串的前导与后续空格
UserName=TRIM(Request.Form("UserName"))
UserPwd=TRIM(Request.Form("UserPwd"))    '读取从表单传递过来的密码数据
Pwd=TRIM(Request.Form("Pwd"))            '读取从表单传来的确认密码数据
'读取从表单传递过来的真实姓名数据
Name=TRIM(Request.Form("Name"))
'读取从表单传递过来的电子邮箱数据
Email=TRIM(Request.Form("Email"))
'用户输入不完全
If UserName="" or UserPwd="" or Pwd="" or Name="" or Email="" Then
    '给出提示,返回注册页面.history.back()返回注册页面
    Response.Write "<script>alert('错误:请输入所有信息');_
        history.back();</script>"
    Response.End
End If
if UserPwd<>Pwd Then                      '两次输入的密码不同
    '给出提示,并返回注册页面
    Response.Write "<script>alert('两次输入的密码不相同!');_
        history.back(); </script>"
    Response.End
End If
'IsValidEmail()过程判断用户输入的的发件人的邮件地址格式是否正确
If NOT IsValidEmail(Email) Then
    '邮件地址格式错误,则给出提示,并返回到注册页面
    Response.Write "<script>alert('错误:邮件地址格式错误!');_
        history.back();</script>"
    Response.End
End If
'判断数据库中是否存在用户输入的用户名
Set HVrs = Server.Createobject("adodb.recordset")
'从数据库中查找UserName列的值为UserName的记录
sql = "Select * from user where UserName = '"&UserName&"'"
HVrs.Open sql,conn,1,1
If not (HVrs.EOF and HVrs.BOF) Then      '存在满足条件的记录
    '用户名在数据库中存在,则给出提示,并返回注册页面
    Response.Write "<script>alert('该用户名已经存在!请重新填写用户名');_
        history.back();</script>"
    Response.End
Else                                     '不存在满足条件的记录
    '在数据库中插入一条新记录
    Set rs=Server.Createobject("adodb.recordset")
    sql="select UserName>Password,Name,Email from User"
    rs.open sql,conn,1,3
    rs.AddNew
    rs("UserName")=UserName              '用户登录名
    rs("Password")=md5(UserPwd)          '用户登录密码
    rs("Name")=Name                      '用户真实姓名
    rs("Email")=Email                   '用户电子邮箱
    rs.update                            '更新数据库
    rs.close
End If
HVrs.Close
Set HVrs = nothing
End Sub%>

```

当用户注册成功后,会出现如图 14-5 所示的设置成功窗口。用户单击【完成】超级链接,

就进入系统登录界面。



图14-5 注册成功界面

14.2.6 聊天室主界面

聊天室主界面 main.asp 是一个多窗口的页面。该页面分为两部分：窗口的上面部分显示聊天室名称，由 topmsg.asp 文件实现；下面部分显示聊天室在线用户列表、聊天内容和用户聊天发表，这部分是由文件 chatroom.asp 实现的。chatroom.asp 文件也是一个多窗口的文件。它由左侧的在线用户列表文件 showname.asp、右上侧的聊天内容显示文件 showmsg.asp 和右下侧的聊天内容提交文件 postmsg.asp 三部分组成。聊天室主界面的布局如图 14-6 所示。



图14-6 聊天室主界面

在 HTML 中 ,设计多窗口页面时经常使用三个元素 FrameSet、Framehe 和 NoFrame。FrameSet 元素定义窗框集，它是一个整体框架；Frame 元素定义每一个框架；NoFrame 元素来显示没有窗框时的信息。因此 main.asp 文件和 chatroom.asp 文件的结构布局代码如下：

main.asp 文件中的聊天室布局代码：

```
'frameSet定义框架集，将窗口定义为一个整体框架
<frameset rows="15%,*" frameborder="1" border="1" framespacing="0">
```

```

' frame定义一个框架，并设置该框架的名称(name)和该框架内要显示的文件
<frame src = "topmsg.asp" name = "topframe">
<frame src="chatroom.asp?myself=<%=Session("user")%>" name="include">
<noframes>                                ' 显示没有框架时的信息
<body>
</body>
</noframes>                                ' 结束无框架定义
</frameset>                                ' 结束框架集定义
<frameset>                                ' 定义框架集
<noframes><body>
</body></noframes>
</frameset>                                ' 结束框架集定义

```

chatroom.asp 文件中的聊天室布局代码：

```

<html>
<head>
<title>聊天室</title>                    ' 设置网页标题
</head>
' frameSet定义框架集，并设置其行高(rows)、框架边框(frameborder)、边框(border)和
' 边框间距(framespacing)
<frameset rows="*" frameborder="1" border="1" framespacing="0">
' 定义框架集，并设置其列宽(cols)、框架边框(frameborder)、边框(border)和边框间距
' (framespacing)
<frameset cols="150,*" frameborder="1" border="1" framespacing="0">
' 定义框架，并设置其边框(border)、该框架内显示的文件(src)、名称(name)、是否有滚动
' 框(scrolling)和禁止调整框架尺寸(noresize)。该框架内显示在线用户列表。
<frame border=1 src="showname.asp?myself=<%=request._
  querystring("myself")%>" name="leftFrame" scrolling="yes" noresize>
<frameset rows="*,100" frameborder="1" border="1" framespacing="0">
' 该框架内显示聊天内容
<frame border = 1 src="showmsg.asp" name="showmsg" scrolling = "yes">
' 该框架内为用户发言提交
<frame border = 1 src="postmsg.asp?myself=<%=request._
  querystring("myself")%>&toname=<%=request.QueryString("toname")%>"_
  name="bottomFrame" scrolling="no" noresize>
</frameset>                                ' 结束框架集定义
</frameset>
</frameset>
<noframes><body>                            ' 显示没有框架时的信息
</body></noframes>
</html>

```

为了防止未登录用户直接进入聊天室，在 main.asp 文件开头添加 Session("user")是否为空的判断语句，如果不为空，则将登录用户添加到 online 数据表中，具体代码如下：

```

<%
If Session("User")<>" Then                    ' 用户登录
' 把用户名添加到在线名单中
set rsx=Server.CreateObject("ADODB.RecordSet")    ' 创建RecordSet对象
' 从数据库中查询id为空的记录
sqlx="select * from online where id is null"
' 执行查询，将结果保存在rsx中
rsx.open sqlx,conn,1,3
rsx.addnew                                        ' 添加一条新记录
rsx("onlinetime")=time()                        ' 登录时间
rsx("username")=Session("user")                ' 用户名
rsx.update                                        ' 更新数据库
rsx.close                                        ' 关闭RecordSet对象
Set rsx = nothing
%>

```

```

.....
<%
Else
    '给出提示，返回到前一页
    Response.Write("<script>alert('你还没有登录!');_
                    history.back();</script>")
    '终止当前对ASP文件的执行，并将现有结果返回给客户端
    Response.End
End If
%>

```

'窗口布局代码

'用户没有登录

14.2.7 在线用户显示页面

用户进入聊天室后，在聊天室的左边能够看到当前在线用户列表。在用户名上单击鼠标左键，就可以向该用户发言了。在线用户显示页面是由 showname.asp 文件完成的，其具体代码如下：

```

<html>
<head>
    <title>用户列表</title>
</head>
<body>
    <table height = "85%" valign = top>
        <tr height = 10>
            <td valign = top>
                在线网友
                <hr>
            </td>
        </tr>
        <tr valign = top>
            <td valign = top align = center>
                <!-- #include file="conn.asp" -->
                <%
                    '显示最新在线名单
                    '创建RecordSet对象
                    set rsout=Server.CreateObject("ADODB.RecordSet")
                    '从数据库中查询所有记录，按照id降序排列
                    sqlout="select * from online where id ORDER BY id DESC"
                    '执行查询操作，将结果保存在rsout中
                    rsout.open sqlout,conn,1,1
                    Do While not rsout.eof
                        '建立超级链接
                        response.write ("<a href = chatroom.asp?myself="&Session("user")_
                                    &"&toname=")
                        response.Write(rsout("username"))
                        response.Write(" target = 'include'>")
                        response.Write(rsout("username"))
                        response.Write("</a>")
                        response.Write("<br>")
                        rsout.movenext
                    loop
                    response.Write("<br>")
                %>
            </td>
        </tr>
    </tr>

```

'设置网页标题

'循环显示所有记录

'显示用户名

'换行

'指向下一条记录

'换行

```

        <td></td>
    </tr>
</table>
<table border=0 cellpadding=0 cellspacing=0 width="86%" height="10%">
    <tr>
        <td></td>
    </tr>
</table>
</body>
</html>

```

程序执行结果如图 14-7 所示。

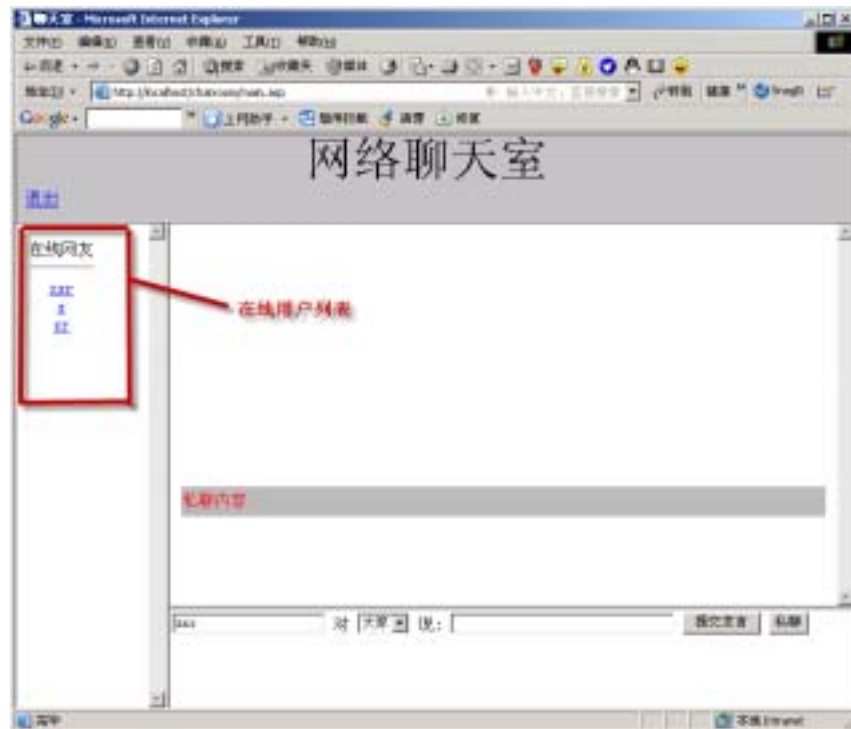


图14-7 在线用户显示列表

14.2.8 聊天发言提交页面

用户登录聊天室之后，就可以跟别人聊天了。聊天发言提交页面由 postmsg.asp 文件实现。其界面设计如图 14-8 所示。界面中包含一个表单 form1，用来让用户选择聊天对象和输入聊天内容。表单 form1 的定义语句如下：

```

<form action="postmsg.asp?act=addmsg&myself=_
<%=request.querystring("myself")%>&toname=_
<%=request.QueryString("toname")%>" method="post" name="form1" >

```



图14-8 postmsg.asp文件的界面设计

在表单 form1 中定义了一个文本域 FromName，为只读类型，显示当前登录用户名。定义文本域 msg，用于用户输入聊天内容。还定义了一个下拉框 rname，用于用户从下拉框中选择聊天对象。在聊天发言提交页面显示时，程序会从数据库中读取当前在线用户名单，并在该下拉框中显示前 50 位。具体代码如下：

```
<select name="rname">                                '定义下拉框

<%
'没有选定聊天对象，则显示大家
If request.QueryString("toname") = "" Then
%>
<option>大家</option>
<%
Else                                                    '选定聊天对象，则显示该对象
%>
<option><%=request.QueryString("toname")%></option>
<%
End If
'显示在线人数
set rsout=Server.CreateObject("ADODB.RecordSet") '创建RecordSet对象
'从数据苦中查许所有记录，按照id降序排列
sqlout="select * from online where id ORDER BY id DESC"
'执行查询，将结果保存在rsout中
rsout.open sqlout,conn,1,1
'在下拉框中最多显示50个用户名
for i=1 to 50
%>
'在线用户名
<option><%=response.Write(rsout("username"))%></option>
<%
rsout.movenext                                          '指向下一条记录
'到达最后一条记录则跳出For循环
if rsout.EOF Then Exit For
next
%>

</select>
```

当用户选定聊天对象，输入聊天内容之后，可以单击【提交留言】按钮或【私聊】按钮，将内容提交。从上面的表单定义语句可知，提交的表单数据仍将在 postmsg.asp 文件中执行。在该文件中，首先读取 act 的值，如果 act 的值为 addmsg，则判断用户单击的是【提交留言】按钮还是【私聊】按钮，单击的按钮不同执行的操作也不同，如果是【提交留言】则将聊天内容及其其他信息保存在数据库中，以便所有的人都能看到该内容；如果是【私聊】，则将私聊的信息保存在 Application 变量中，只允许部分人看到留言。具体的执行代码如下：

```
<%
ac=request.QueryString("act")                        '读取act的值
if ac="addmsg" then                                   'act的值为addmsg
    If Request.Form("hidden")<>" Then                '私聊
        '清空Session("chat")中的内容
        Session("chat") = ""
        '把要显示的私聊信息保存在Application("chat")中
        Application("chat") = "<font size = 2>"&request.Form("username")&"_
                                悄悄地对"&request.form("rname")&"说:_
                                <font color = black>"&request.form("msg")_
                                &"</font>"&time()&"</font>"
```

```

        Application("chatto") = request.form("rname") '聊天对象
        Application("owner") = request.Form("username") '发言者
Else '不是私聊
    '创建RecordSet对象
    set rs=server.createobject("adodb.recordset")
    sql="select * from chat" '查询数据库
    rs.open sql,conn,1,3 '执行查询
    rs.addnew '添加一条新记录
    rs("FromName")=request.Form("username") '发言者
    rs("ToName")=request.form("rname") '聊天对象
    rs("chat")=request.form("msg") '显示的私聊信息
    rs("posttime")=time() '提交时间
    rs.update '更新数据库
    rs.Close '关闭RecordSet对象
    set rs=nothing
    '增加一条聊天记录,将Application("Gchatnum")的值加1
    Application("Gchatnum") = Application("Gchatnum")+1
End If
end if
%>

```

postmsg.asp 文件完整的代码执行步骤如下：

- (1) 使用代码<!-- #include file="conn.asp" -->连接数据库。
 - (2) 使用 HTML 代码生成页面，并从数据库中读取在线用户名，将前面 50 位用户添加到下拉框中。
 - (3) 读取 act 的值。
 - (4) 判断 act 的值。如果 act 的值为 addmsg，则判断用户单击的是哪一个按钮。
 - (5) 如果用户单击【私聊】按钮，则把私聊要显示的内容和私聊的发言者和发言对象保存在 Application 变量中。
 - (6) 如果用户单击【提交留言】按钮，则将用户的聊天内容以及聊天发言者和发言对象、发言时间保存在数据库中。
- 程序的具体执行效果请看下面一节。

14.2.9 聊天内容显示页面

聊天内容显示页面由 showmsg.asp 文件实现。在该页面，可以显示公共聊天内容和私聊内容。具体代码如下：

```

<!--#include file="conn.asp"--> '包含文件conn.asp
<HTML>
<HEAD>
    <TITLE>聊天室</TITLE> '设置网页标题
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=gb2312">
    '网页自动刷新语句，在此为每隔2秒刷新一次
    <META HTTP-EQUIV="refresh" CONTENT="2">
</HEAD>
<BODY>
    '生成表格并设置其宽度(width)、高度(height)和边框(border)
    <table width="100%" height="70%" border="0" >
        <tr>
            <td valign = top>
                <%
                '显示聊天内容
                'Session("Lchatnum") 为用户登录时聊天总记录数，Application("Gchatnum")为

```

```

'当前聊天总记录数, Session("Lchatnum")<Application("Gchatnum")表明用户
'登录之后, 该聊天室有用户聊天, 则显示登录之后的聊天记录
If Session("Lchatnum")<Application("Gchatnum") Then
    start = Session("Lchatnum")+1
    '使用For语句循环显示所有聊天记录
    for i = Application("Gchatnum") to start step -1
        '创建RecordSet对象
        set rs=Server.CreateObject("ADODB.RecordSet")
        '从数据库中查询id列的值为i的记录, 并将结果保存在rs中
        rs.Open "Select * from chat where id=" & i, conn, 1, 3
    %>
    '建立对该用户发言的超级链接
    <a href = chatroom.asp?myself=<%=Session("user")%>&_
toname=<%=rs("FromName")%>target="include">
        <%response.Write(rs("FromName"))%>      '发言者
    </a>
    对
    '建立对该用户发言的超级链接
    <a href = chatroom.asp?myself=<%=Session("user")%>&_
toname=<%=rs("ToName")%>target="include">
        <%response.Write(rs("ToName"))%>      '发言对象
    </a>
    说:
    <font color = black>                                '字体颜色
    <%
        response.Write(rs("chat"))                '聊天内容
    %>
    </font>
    <%
        response.Write(rs("posttime")&"<br>")      '发言时间
    %>
    <br>
    <%
        rs.close()                                '关闭RecordSet对象
        set rs=nothing
    Next
End If
%>
</td>
</tr>
</table>

<table height = "30%" width = "100%">                '该表格内显示私聊内容
    '定义表行, 并设置其宽度(width)、垂直对齐方式(valign)、高度(height)和背景颜
    '色(bgcolor)
    <tr width = "10" valign = center height = "30%" bgcolor = "#bbbbbb">
        <td><font color = red>私聊内容</font></td>
    </tr>
    <tr>
        <td valign = top>
            <%
                'Session("chat")中保存用户浏览器上的私聊内容, Application("chat")中保
                '存当前私聊内容, 二者不相等, 说明有新的私聊信息
                If Session("chat") <> Application("chat") Then
                    '登录用户为私聊发言者或者为私聊发言对象, 或者私聊对象为所有人, 则显示私聊
                    '内容
                    If Session("user") = Application("owner") or
                        Session("user") = Application("chatto") or

```

```

Application("chatto") = "大家" Then
    Response.Write Application("chat")      '显示私聊内容
    '保存用户可以看到的私聊内容
    Session("chat") = Application("chat")
End If

Else                                         '不满足条件
    Response.Write Session("chat")         '显示用户私聊内容
End If
%>
</td>
</tr>
</table>
</BODY>
</HTML>

```

当用户刚登录时，聊天内容显示界面上什么都没有，由 14.2.7 节图 14-7 可以看出。随着用户在该聊天室的停留，其他用户从该用户登录之后的公共聊天内容和该用户的聊天内容都将显示在界面上。

图 14-9 为用户 xr 登录一段时间之后的聊天内容。



图14-9 登录一段时间之后的聊天室

当用户输入聊天内容，然后单击【私聊】按钮，用户聊天内容将显示在下方的私聊内容区，并且只能由私聊双方看到。

下面为用户 x 向用户 zxr 发送私聊信息“呵呵，不告诉你”的显示结果。下面图 14-10 到 14-12 分别为用户 x、用户 zxr 和用户 xr 所看到的界面。



图14-10 用户x看到的界面



图14-11 用户zxr看到的界面



图14-12 用户xr看到的界面

14.2.10 用户退出页面

在聊天室主界面，单击【退出】超级链接，将执行 `logout.asp` 文件。退出登录的操作包括以下内容：

- 将当前登录用户从数据库中删除。
- 将 `Session("user")` 设置为空。
- 将 `Session("chat")` 设置为空。
- 将 `Session("Lchatnum")` 置为0。
- 将网页转到 `index.asp` 用户登录文件。

`logout.asp` 文件的代码如下：

```
<!--#include file = "conn.asp"-->
<%
    ' 创建RecordSet对象
    set rsdell=Server.CreateObject("ADODB.RecordSet")
    ' 将当前登录用户从在线用户数据表中删除
    sqldell="delete * from online where username='"&Session("user")&"'"
    rsdell.open sqldell,conn,1,3
    Session("user") = ""
    Session("chat") = ""
    Session("Lchatnum") = 0
    Response.Redirect "index.asp"
%>
```

14.3 本章小结

本章讲述了如何实现一个聊天室，加深了对 Application 对象和 Session 对象的了解，对网页的布局也有了更深的认识。

不过，本章的聊天室虽然已经实现了一些基本的功能，但还存在这不足，例如，私聊信息的存储、用户信息的查看等等，都没有实现。读者在编程的时候可以再进一步思考一下，实现更多的功能。