



<<< 软件开发 实战 宝典

# PHP

## 开发实战宝典

○ 潘凯华 邹天恩 等编著 ○

在线服务：模块库、案例库、题库、素材库、答疑服务

视频讲解：43小时同步语音视频讲解

模块实战：49个实战模块，55项实验

项目实战：8个项目开发实战案例

DVD语音视频教学光盘：

- 43小时同步视频讲解，现场聆听专业讲授
- 附实例及其源程序，让学习、分析、调试程序更方便
- 赠3个项目案例的视频及源程序，方便拓展训练

本书特色：

- 272个小型实例巩固各项必备技术
- 49个实战模块，55项实验提高综合应用能力
- 5个项目案例提高项目开发的熟练度

在线服务方式：

在线服务网站：[www.mingribook.com](http://www.mingribook.com)

在线QQ服务：100310267 服务热线：400-675-1066



更多资源请访问我的新浪博客(<http://blog.sina.com.cn/ckook>)

B

清华大学出版社

软件开发实战宝典

# PHP 开发实战宝典

# PHP 开发实战宝典

潘凯华 邹天思 等编著

清华大学出版社

清华大学出版社

# 北 京

更多资源请访问我的新浪博客(<http://blog.sina.com>)

清华大学出版社

北京

更多资源请访问我的新浪博客(<http://blog.sina.com.cn/ckook>)

## 内 容 简 介

本书从初学者角度出发，以基础知识为“基石”，以核心技术和高级应用为“梁柱”，最终通过实战项目来检验成果，这就是本书的编写思路。本书突出“基础”、“全面”、“深入”，强调“实战”效果。在介绍技术的同时，各章都提供有示例或稍大一些的实例，同时在各章的结尾通过几个小型项目来综合应用本章所讲解的知识，做到理论与实践相结合。最后提供5个完整的项目实例，详细讲述从前期规划、系统设计，到项目开发的全部实现过程。

全书共分29章，包括初识PHP，PHP环境搭建，PHP语言基础，流程控制语句，函数，字符串，正则表达式，PHP数组，PHP与Web页面交互，PHP与JavaScript交互，日期和时间，会话管理，图形图像，文件系统，面向对象，PHP加密技术，MySQL数据库基本操作，phpMyAdmin图形化管理工具，使用MySQL函数操作数据库，ADODB类库，程序调试与错误处理，XML技术，Ajax技术，Smarty模板技术，博客管理系统，应用Smarty模板开发电子商务平台，办公自动化系统，供求信息网，图书借阅管理系统等内容，最后还在附录中介绍了PHP的编码规则。所有知识都结合具体实例进行介绍，对涉及的程序代码给出了详细的注释，可以使读者轻松领会PHP程序开发的精髓，快速提高开发技能。

本书适合有志于从事软件开发的初学者、高校计算机相关专业学生和毕业生，也可作为软件开发人员的参考手册，或者高校的教学参考书。

本书提供大量的模块实战和项目实战案例，配备43小时视频讲解光盘，提供全部实例源代码，习题都给出了相应的答案。

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

PHP开发实战宝典/潘凯华，邹天思等编著. —北京：清华大学出版社，2010.1  
(软件开发实战宝典)

ISBN 978-7-302-20980-5

I. P… II. ①潘… ②邹… III. PHP语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2009)第163121号

责任编辑：刘利民 郭伟

版式设计：魏远

责任校对：柴燕

责任印制：杨艳

出版发行：清华大学出版社

<http://www.tup.com.cn>

社 总 机：010-62770175

地 址：北京清华大学学研大厦A座

邮 编：100084

邮 购：010-62786544

投稿与读者服务：010-62776969,c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京密云胶印厂

装 订 者：三河市李旗庄少明装订厂

经 销：全国新华书店

开 本：203×260 印 张：42.25 字 数：1124千字

(附DVD视频光盘1张)

版 次：2010年1月第1版 印 次：2010年1月第1次印刷

印 数：1~5000

定 价：79.80元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系  
调换。联系电话：(010)62770177 转 3103 产品编号：034018-01

## 本书编著者名单

主 编：赛奎春 刘志铭 宋 坤

编 著：赛奎春 刘志铭 宋 坤 刘彬彬 刘锐宁 王小科

王国辉 邹天思 梁 水 李钟尉 董大勇 杨 丽

刘中华 陈丹丹 潘凯华 李永强 张振坤 梁 冰

房大伟 吕 双 高春艳 安 剑 孙秀梅 刘 欣

刘龄龄 李 慧



本书是为初学者量身定做的 PHP 入门教材。全书共分 12 章，由浅入深，循序渐进地介绍了 PHP 的基本语法、各种数据类型、控制语句、函数、类与对象、异常处理、文件操作、数据库连接与操作、Web 表单处理、AJAX、AJAX+jQuery、Ajax+PHP、AJAX+MySQL、AJAX+MongoDB、AJAX+Redis、AJAX+Memcached、AJAX+MongoDB+Redis、AJAX+Memcached+Redis、AJAX+MongoDB+Redis+Memcached 等知识。

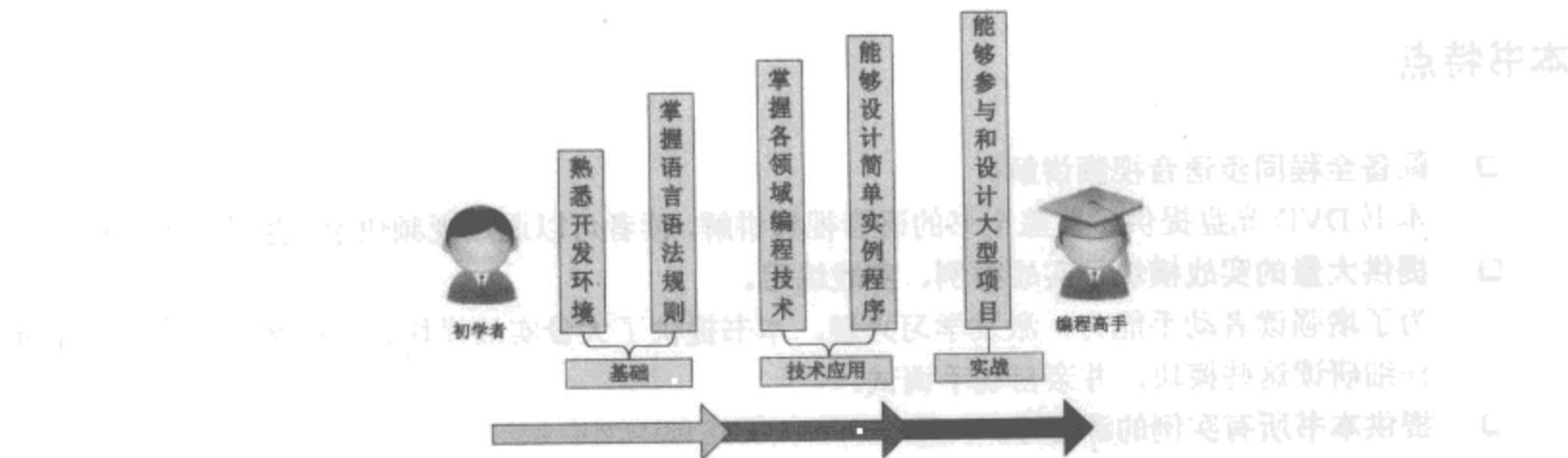
# 前言

## Preface

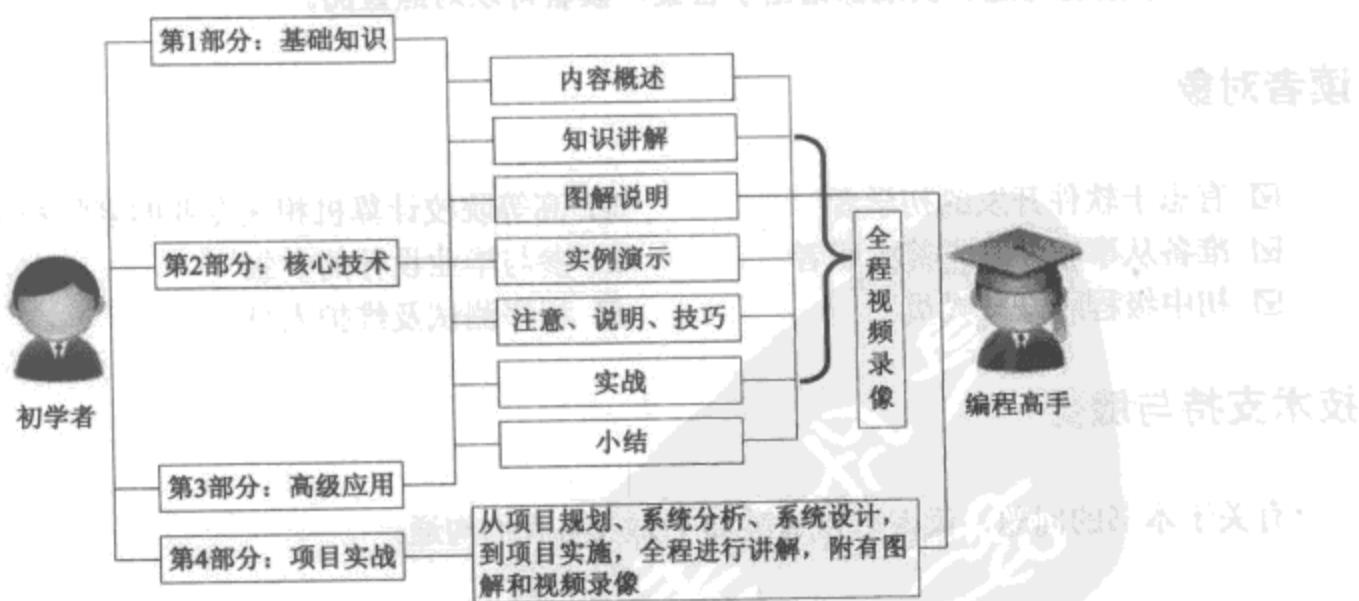
PHP 是全球最普及、应用最广泛的互联网开发语言之一。PHP 语言具有简单、易学、源码开放，可操纵多种主流与非主流的数据库，支持面向对象的编程，支持多种开源框架（如 Zend Framework），支持跨平台的操作，而且完全免费等特点，越来越受到广大程序员的青睐和认同。PHP 目前拥有几百万个用户，其发展速度要快于在它之前的任何一种计算机语言。相信 PHP 一定能够经得起实践的检验，发展成为互联网开发语言中“主流的主流”。

## 本书内容

从初学程序开发的人员步入到编程高手行列通常需要经历 3 个阶段，如下图所示。



本书中的内容也是按照这一规律精心组织的，将从初学者入门到成为编程高手所必备的各类知识分成 4 个部分，大体结构如下图所示。



**第1部分：基础知识。**通过初识 PHP、PHP 环境搭建、PHP 语言基础、流程控制语句、函数、字符串、正则表达式、PHP 数组、PHP 与 Web 页面交互、PHP 与 JavaScript 交互、日期和时间等章节，并结合大量的图示、示例、综合实例以及录像等，帮助读者快速掌握 PHP 语言，并为以后编程奠定坚实的基础。

**第2部分：核心技术。**介绍了 Cookie 与 Session、图形图像、文件系统、面向对象、PHP 加密技术、MySQL 数据库基本基础、phpMyAdmin 图形化管理工具、使用 MySQL 函数操作数据库、ADODB 类库、PHP 程序调试与异常处理等。学习完这一部分，能够开发数据库应用程序和一些中小型的热点模块。

**第3部分：高级应用。**介绍了 XML 技术、Ajax 技术和 Smarty 模板技术。学习完这一部分，能够开发一些实用的网络程序等。

**第4部分：项目实战。**作者精心挑选并开发了 5 个大型完整的项目。这 5 个项目涵盖了 MySQL 数据库、Smarty 模板技术、ADODB 类库、Ajax 等主流技术。运用软件工程的设计思想，让读者学习如何进行网站项目的实践开发。书中按照编写项目计划书→系统设计→数据库设计→创建项目→实现项目→运行项目→解决开发常见问题→发布网站的过程进行介绍，引领读者一步一步亲身体验开发项目的全过程，使读者可以巩固前面所学的知识和技术，积累项目开发经验。

## 本书特点

### □ 配备全程同步语音视频讲解。

本书 DVD 光盘提供了覆盖全书的语音视频讲解，读者可以通过视频快速、直观、轻松地学习。

### □ 提供大量的实战模块、实战案例、实战练习。

为了增强读者动手能力，激发学习兴趣，本书提供了大量实战模块、实战案例，读者一定要仔细研读这些模块，并亲自动手调试。

### □ 提供本书所有实例的源程序。

本书光盘提供了所有实例的源程序，读者不必逐字逐句录入，可直接复制，先比猫画虎试着做一遍，可以提高学习效率，模仿是最快的学习方式。

### □ 本书所有习题和实战都给出了答案，读者可以对照查阅。

## 读者对象

有志于软件开发的初学者

准备从事软件开发的求职者

初中级程序开发人员

高等院校计算机相关专业的老师和学生

参与毕业设计的学生

程序测试及维护人员

## 技术支持与服务

有关于本书的问题，读者可以通过如下方式与我们沟通：

## 前 言

---

1. 登录本技术网站: [www.mingribook.com](http://www.mingribook.com), 查阅相关问题或者留言。
2. 通过邮箱: [tmoonbook@sina.com](mailto:tmoonbook@sina.com) 或 [th\\_press@263.net](mailto:th_press@263.net)。
3. 加入 QQ: 100310063。

最后, 感谢您选择本书, 希望本书能成为您编程路上的领航者。

祝读书快乐!

编 者





# 目 录

Contents

## 第1部分 基础知识

### 第1章 初识 PHP ..... 3

视频讲解：7分钟

1.1 PHP 概述 ..... 4
1.1.1 PHP 是什么 ..... 4
1.1.2 PHP 5 的新特性 ..... 4
1.1.3 PHP 的语言优势 ..... 4
1.1.4 PHP 语言的发展 ..... 5
1.2 PHP 与其他语言的比较 ..... 6
1.3 PHP 语言相关术语 ..... 7
1.4 资深程序员谈如何学习 PHP ..... 8
1.5 常用代码编辑工具 ..... 10
1.6 学习资源 ..... 12
1.6.1 PHP 参考手册 ..... 12
1.6.2 PHP 学习讨论 ..... 12
1.7 成功案例 ..... 13
1.8 本章小结 ..... 14

### 第2章 PHP 环境搭建 ..... 15

视频讲解：80分钟

2.1 搭建 PHP 开发环境的准备工作 ..... 16
2.1.1 在 Windows 下搭建 PHP 开发环境的准备工作 ..... 16
2.1.2 在 Linux 下搭建 PHP 开发环境的准备工作 ..... 16
2.2 Apache 服务器的安装和配置 ..... 16
2.2.1 在 Windows 下安装 Apache 服务器 ..... 16
2.2.2 在 Linux 下安装 Apache 服务器 ..... 19
2.3 PHP 的安装和配置 ..... 20
2.3.1 在 Windows 下安装 PHP ..... 20
2.3.2 在 Linux 下安装 PHP ..... 21

### 2.4 MySQL 服务器的安装和配置 ..... 23

    2.4.1 在 Windows 下安装 MySQL 服务器 ..... 23

    2.4.2 在 Linux 下安装 MySQL 服务器 ..... 26

### 2.5 初学者——应用组合包快速搭建 PHP 环境 ..... 27

### 2.6 第一个 PHP 程序 ..... 29

    2.6.1 使用 Macromedia Dreamweaver 8 编写源程序 ..... 29

    2.6.2 发布和运行 PHP 程序 ..... 31

### 2.7 环境安装常见问题 ..... 31

    2.7.1 Apache 安装常见问题 ..... 31

    2.7.2 PHP 安装常见问题 ..... 32

    2.7.3 MySQL 安装常见问题 ..... 33

### 2.8 实战 ..... 33

### 2.9 本章小结 ..... 35

### 2.10 实战练习 ..... 36

## 第3章 PHP 语言基础 ..... 37

视频讲解：125分钟

3.1 学习语言基础的重要性 ..... 38
3.2 PHP 语法基础 ..... 38
3.2.1 PHP 标记符 ..... 38
3.2.2 PHP 的注释 ..... 39
3.3 PHP 的数据类型 ..... 40
3.3.1 整型 ..... 40
3.3.2 布尔型 ..... 41
3.3.3 浮点型 ..... 41
3.3.4 字符串型 ..... 42
3.3.5 null 类型 ..... 44
3.3.6 数据类型的转换 ..... 45

3.4 PHP 常量 .....	46	4.4.1 使用 break 语句跳出循环 .....	79
3.4.1 声明和使用常量 .....	46	4.4.2 使用 continue 语句跳出循环 .....	81
3.4.2 预定义常量 .....	48	<b>4.5 实战 .....</b>	<b>82</b>
3.5 PHP 变量 .....	49	4.5.1 过滤留言信息中的敏感词 .....	82
3.5.1 变量声明及使用 .....	49	4.5.2 图书信息显示 .....	85
3.5.2 变量作用域 .....	51	4.5.3 应用 switch 语句设计网站主页 .....	86
3.5.3 可变变量 .....	53	<b>4.6 本章小结 .....</b>	<b>88</b>
3.5.4 预定义变量 .....	53	<b>4.7 实战练习 .....</b>	<b>88</b>
3.5.5 变量的生存周期 .....	54		
3.6 PHP 运算符 .....	54	<b>第 5 章 函数 .....</b>	<b>89</b>
3.6.1 字符串运算符 .....	54	<b>■ 视频讲解：75 分钟</b>	
3.6.2 算术运算符 .....	55		
3.6.3 赋值运算符 .....	56	5.1 自定义函数 .....	90
3.6.4 递增递减运算符 .....	57	5.1.1 函数的命名规则 .....	90
3.6.5 位运算符 .....	58	5.1.2 自定义函数格式 .....	90
3.6.6 逻辑运算符 .....	59	5.1.3 调用自定义函数 .....	91
3.6.7 比较运算符 .....	60	5.1.4 函数的参数 .....	91
3.6.8 条件运算符 .....	61	5.1.5 函数返回值 .....	93
3.6.9 运算符的使用规则 .....	61	5.1.6 嵌套调用 .....	93
3.7 PHP 的表达式 .....	62	5.1.7 递归调用 .....	94
<b>3.8 实战 .....</b>	<b>63</b>	5.2 内建函数 .....	94
3.9 本章小结 .....	64	5.3 输出语句 .....	95
3.10 实战练习 .....	64	5.3.1 应用 print 语句输出字符 .....	95
<b>第 4 章 流程控制语句 .....</b>	<b>65</b>	5.3.2 应用 echo 语句输出字符 .....	96
<b>■ 视频讲解：80 分钟</b>		5.3.3 应用 printf 语句格式化输出字符 .....	98
4.1 学前导读 .....	66	5.3.4 应用 sprintf 语句格式化输出字符 .....	99
4.1.1 什么是算法 .....	66	5.4 引用文件 .....	100
4.1.2 算法的描述方法 .....	66	5.4.1 应用 include 语句引用文件 .....	100
4.1.3 程序的 3 种控制结构 .....	67	5.4.2 应用 require 语句引用文件 .....	101
4.2 条件控制语句 .....	69	5.4.3 应用 include_once 语句引用文件 .....	102
4.2.1 if 条件语句 .....	69	5.4.4 应用 require_once 语句引用文件 .....	102
4.2.2 switch...case 分支控制语句 .....	72	5.4.5 include 语句和 require 语句的使用区别 .....	104
4.3 循环控制语句 .....	74	5.4.6 include_once 语句和 require_once 语句的使用区别 .....	104
4.3.1 while 循环语句 .....	74	<b>5.5 实战 .....</b>	<b>104</b>
4.3.2 do...while 循环语句 .....	75	5.5.1 通过自定义函数防止新闻主题信息出现中文乱码 .....	104
4.3.3 for 循环语句 .....	76	5.5.2 应用 include 语句构建在线音乐网站主页 .....	106
4.3.4 foreach 循环 .....	78	<b>5.6 本章小结 .....</b>	<b>108</b>
4.4 跳转控制语句 .....	79	<b>5.7 实战练习 .....</b>	<b>108</b>

<b>第6章 字符串</b>	<b>109</b>
<b>    视频讲解: 76分钟</b>	
6.1 字符串简介	110
6.2 单引号与双引号	110
6.3 定界符	111
6.4 连接字符串	113
6.5 转义、还原字符串	114
6.5.1 手动转义、还原字符串	114
6.5.2 自动转义、还原字符串	114
6.6 获取字符串长度	116
6.7 截取字符串	118
6.8 比较字符串	119
6.8.1 按字节比较	119
6.8.2 按自然排序法比较	120
6.8.3 指定从源字符串的位置比较	121
6.9 检索字符串	121
6.9.1 应用 strstr() 函数检索指定的关键字	121
6.9.2 应用 substr_count() 函数检索子串出现的次数	122
6.10 替换子字符串	124
6.11 去除字符串中空白字符和特殊字符	128
6.11.1 去除字符串左端空白字符和特殊字符	128
6.11.2 去除字符串右端空白字符和特殊字符	129
6.11.3 去除字符串两端空白字符和特殊字符	129
6.12 格式化字符串	130
6.13 分割字符串	131
6.14 字符串与 HTML 相互转换	132
6.15 实战	134
6.15.1 超长文本的分页显示	134
6.15.2 控制页面中输出字符串的长度	136
6.16 本章小结	137
6.17 实战练习	138
<b>第7章 正则表达式</b>	<b>139</b>
<b>    视频讲解: 82分钟</b>	
7.1 正则表达式是什么	140
7.2 正则表达式常用术语	140
7.3 正则表达式语法规则	140
7.3.1 行定位符 (^ 和 \$)	141
7.3.2 字符类 ([])	141
7.3.3 选择字符 ( )	142
7.3.4 连字符 (-)	142
7.3.5 排除字符 ([^])	142
7.3.6 限定符 (? * + {n,m})	142
7.3.7 点字符 (.)	143
7.3.8 转义字符 (\)	143
7.3.9 反斜线 (\)	144
7.3.10 括号字符 (()	145
7.3.11 反向引用	145
7.4 POSIX 扩展正则表达式函数	146
7.4.1 查找字符串	146
7.4.2 替换字符串	146
7.4.3 分割字符串	147
7.5 PCRE 兼容正则表达式函数	147
7.5.1 查找字符串	147
7.5.2 替换字符串	148
7.6 实战	149
7.7 本章小结	152
7.8 实战练习	152
<b>第8章 PHP 数组</b>	<b>153</b>
<b>    视频讲解: 90分钟</b>	
8.1 数组是什么	154
8.2 声明数组	154
8.2.1 数组命名规则	155
8.2.2 通过 PHP 函数创建数组	155
8.2.3 通过数组标识符 “[ ]” 创建数组	156
8.3 数组的类型	157
8.3.1 数字索引数组	157
8.3.2 关联数组	157
8.4 输出数组	158
8.5 数组的构造	159
8.5.1 创建一维数组	159
8.5.2 创建二维数组	159
8.6 遍历数组	160
8.6.1 foreach 结构遍历数组	160
8.6.2 each() 函数遍历数组	161
8.6.3 list() 函数遍历数组	161

8.6.4 for 语句遍历数组 .....	163
8.7 合并数组 .....	163
8.8 字符串与数组的转换 .....	164
8.8.1 应用 explode() 函数将字符串转换成数组 ...	164
8.8.2 应用 implode() 函数将数组转换成一个 新字符串 .....	165
8.9 统计数组元素个数 .....	165
8.10 数组排序 .....	166
8.10.1 数字数组排序 .....	166
8.10.2 关联数组排序 .....	168
8.11 实战 .....	169
8.12 本章小结 .....	170
8.13 实战练习 .....	170
<b>第 9 章 PHP 与 Web 页面交互 .....</b>	<b>171</b>
<b>视频讲解：124 分钟</b>	
9.1 表单 .....	172
9.1.1 创建表单 .....	172
9.1.2 认识表单元素 .....	173
9.2 在 Web 页中创建表单 .....	175
9.3 在 Web 页中嵌入 PHP 脚本 .....	177
9.3.1 在 HTML 标记中嵌入 PHP 脚本 .....	177
9.3.2 为表单元素赋值 .....	178
9.4 提交表单数据 .....	178
9.4.1 POST 方法提交表单数据 .....	179
9.4.2 GET 方法提交表单数据 .....	179
9.4.3 POST 方法与 GET 方法的使用区别 .....	180
9.5 PHP 全局变量 .....	181
9.5.1 \$_POST[] 全局变量 .....	181
9.5.2 \$_GET[] 全局变量 .....	182
9.5.3 \$_SESSION[] 全局变量 .....	183
9.6 对 URL 传递的参数进行编/解码 .....	183
9.6.1 对 URL 传递的参数进行编码 .....	183
9.6.2 对 URL 编码的字符串进行解码 .....	184
9.7 实战 .....	185
9.7.1 发布和查看公告信息 .....	185
9.7.2 图书借阅 .....	187
9.8 本章小结 .....	188
9.9 实战练习 .....	188

<b>第 10 章 PHP 与 JavaScript 交互 .....</b>	<b>189</b>
<b>视频讲解：81 分钟</b>	
10.1 了解 JavaScript .....	190
10.1.1 JavaScript 是什么 .....	190
10.1.2 JavaScript 能做什么 .....	190
10.2 JavaScript 语言基础 .....	190
10.2.1 JavaScript 数据类型 .....	190
10.2.2 JavaScript 变量 .....	191
10.2.3 JavaScript 注释 .....	192
10.3 自定义函数 .....	193
10.4 JavaScript 脚本嵌入方式 .....	194
10.4.1 在 HTML 中嵌入 JavaScript 脚本 .....	194
10.4.2 应用 JavaScript 事件调用自定义函数 .....	195
10.4.3 在 PHP 动态网页中引用 JS 文件 .....	196
10.4.4 解决浏览器不支持 JavaScript 问题 .....	197
10.5 实战 .....	200
10.5.1 应用 JavaScript 脚本制作树状 导航菜单 .....	200
10.5.2 PHP 与 JavaScript 脚本的交互应用 .....	201
10.6 本章小结 .....	206
10.7 实战练习 .....	206
<b>第 11 章 日期和时间 .....</b>	<b>207</b>
<b>视频讲解：93 分钟</b>	
11.1 系统时区设置 .....	208
11.1.1 时区的划分 .....	208
11.1.2 PHP 中的时区设置 .....	208
11.2 UNIX 时间戳 .....	209
11.2.1 什么是时间戳 .....	209
11.2.2 UNIX 时间戳 .....	209
11.2.3 获取指定日期的时间戳 .....	210
11.2.4 获取当前时间戳 .....	211
11.2.5 将英文文本的日期时间描述解析为 UNIX 时间戳 .....	211
11.3 PHP 日期和时间的处理 .....	212
11.3.1 格式化日期和时间 .....	212
11.3.2 获取日期和时间信息 .....	214
11.3.3 获取本地化的日期和时间 .....	215
11.3.4 检验日期和时间的有效性 .....	217

11.4 实战 .....	218
11.4.1 实现倒计时的功能 .....	218
11.4.2 实现生日提示功能 .....	219

11.4.3 计算在线考试用时和剩余时间 .....	222
11.5 本章小结 .....	225
11.6 实战练习 .....	226

## 第2部分 核心技术

### 第12章 会话管理 ..... 229

视频讲解: 117分钟

12.1 Cookie 管理 .....	230
12.1.1 了解 Cookie .....	230
12.1.2 创建 Cookie .....	231
12.1.3 读取 Cookie .....	232
12.1.4 删 除 Cookie .....	233
12.1.5 Cookie 应用的注意事项 .....	234
12.1.6 Cookie 的综合应用——自动登录 .....	235
12.2 Session 管理 .....	237
12.2.1 了解 Session .....	237
12.2.2 启动 Session .....	238
12.2.3 应用 Session .....	239
12.2.4 删 除 Session .....	240
12.2.5 Session 的综合应用——通过 Session 控制页面访问权限 .....	240

### 12.3 Cookie 与 Session 的比较 ..... 242

12.4 实战 .....	242
12.4.1 通过 Cookie 实现单点登录 .....	242
12.4.2 Session 购物车 .....	245
12.5 本章小结 .....	250
12.6 实战练习 .....	250

### 第13章 图形图像 ..... 251

视频讲解: 106分钟

13.1 了解 GD2 函数库 .....	252
13.2 参数设置 .....	252
13.3 Jpgraph 类库 .....	253
13.3.1 Jpgraph 类库简介 .....	253
13.3.2 Jpgraph 的安装 .....	253
13.3.3 Jpgraph 的配置 .....	253
13.4 常用图像处理 .....	254
13.4.1 创建画布 .....	254

13.4.2 颜色处理 .....	255
13.4.3 输出图像 .....	256
13.4.4 绘制基本图形 .....	257
13.4.5 填充几何图形 .....	258
13.4.6 输出文字 .....	259

### 13.5 实战 ..... 260

13.5.1 创建一个简单的图像 .....	260
13.5.2 应用 GD2 函数在照片上添加文字 .....	261
13.5.3 应用图像处理技术生成验证码 .....	262
13.5.4 Jpgraph 创建柱状图展示年度收支情况 ..	264
13.5.5 Jpgraph 创建折线图统计图书销售走势 ..	265
13.5.6 Jpgraph 创建 3D 饼形图展示部门业绩比较 ..	267
13.6 本章小结 .....	268
13.7 实战练习 .....	268

### 第14章 文件系统 ..... 269

视频讲解: 74分钟

14.1 文件处理 .....	270
14.1.1 打开文件 .....	270
14.1.2 读取文件 .....	271
14.1.3 写入文件 .....	274
14.1.4 操作文件 .....	275
14.1.5 关闭文件 .....	276
14.2 文件上传 .....	276
14.2.1 配置 php.ini 文件 .....	277
14.2.2 预定义变量\$_FILES .....	277
14.2.3 文件上传函数 .....	278
14.3 文件处理的高级应用 .....	279
14.3.1 远程文件的访问 .....	279
14.3.2 文件指针 .....	280
14.3.3 锁定文件 .....	282
14.4 目录处理 .....	282

14.4.1 打开目录	283	15.13 实战练习	310
14.4.2 浏览目录	283	<b>第 16 章 PHP 加密技术</b>	
14.4.3 操作目录	284	视频讲解：8分钟	
14.4.4 关闭目录	284	16.1 加密技术简介	312
<b>14.5 实战</b>	<b>285</b>	16.1.1 MD5 算法	312
14.5.1 多图片上传	285	16.1.2 SHA1 算法	313
14.5.2 通过文本文件统计页面访问量	286	16.1.3 DES 算法	313
14.6 本章小结	288	16.1.4 MIME/BASE64 的算法	314
14.7 实战练习	288	<b>16.2 PHP 内置加密函数</b>	314
<b>第 15 章 面向对象</b>	<b>289</b>	16.2.1 使用 crypt() 函数进行加密	314
视频讲解：160分钟			
15.1 面向对象的基本概念	290	16.2.2 使用 md5() 函数进行加密	316
15.1.1 类	290	16.2.3 使用 sha1() 函数进行加密	316
15.1.2 对象	290	16.2.4 使用 base64_encode() 函数进行编码	317
15.1.3 面向对象编程的三大特点	290	16.2.5 使用 base64_decode() 函数进行解码	317
15.2 类与对象	291	<b>16.3 PHP 加密扩展库</b>	317
15.2.1 定义类	291	16.3.1 Mcrypt 扩展库	318
15.2.2 实例化类	292	16.3.2 Mhash 扩展库	320
15.2.3 使用对象	292	<b>16.4 实战</b>	321
15.2.4 成员变量	293	16.5 本章小结	322
15.2.5 成员函数	294	16.6 实战练习	322
15.3 类的权限修饰符	295	<b>第 17 章 MySQL 数据库基本操作</b>	323
15.3.1 public（公有成员）	295	视频讲解：70分钟	
15.3.2 private（私有成员）	296	<b>17.1 MySQL 简介</b>	324
15.3.3 protected（保护成员）	297	17.1.1 MySQL 是什么	324
15.4 构造函数	297	17.1.2 MySQL 特点	324
15.4.1 创建构造函数	298	17.1.3 MySQL 5 支持的特性	325
15.4.2 调用父类构造函数	299	<b>17.2 启动、连接、断开和停止 MySQL 服务器</b>	325
15.5 析构函数	299	17.2.1 启动 MySQL 服务器	325
15.6 使用\$this 变量访问方法和属性	300	17.2.2 连接和断开 MySQL 服务器	326
15.7 static 关键字	301	17.2.3 停止 MySQL 服务器	327
15.8 final 关键字	302	<b>17.3 使用 MySQL 命令行操作数据库</b>	328
15.9 抽象类	303	17.3.1 创建数据库	328
15.10 接口的使用	305	17.3.2 查看数据库	329
<b>15.11 实战</b>	<b>306</b>	17.3.3 选择数据库	329
15.11.1 面向对象的用户登录身份验证	306	17.3.4 删除数据库	329
15.11.2 面向对象的数据分页显示	308	<b>17.4 使用 MySQL 命令行操作数据表</b>	330
15.12 本章小结	310		



17.4.1 创建数据表.....	330	19.2.5 应用 mysql_fetch_object()函数从结果集中获取一行作为对象.....	360
17.4.2 查看表结构.....	331	19.2.6 应用 mysql_fetch_row()函数逐行获取结果集中的每条记录.....	361
17.4.3 修改表结构.....	332	19.2.7 应用 mysql_num_rows()函数获取查询结果集中的记录数.....	363
17.4.4 重命名表.....	333	19.2.8 关闭连接.....	364
17.4.5 删除表.....	334	<b>19.3 PHP 操作 MySQL 数据库.....</b>	365
<b>17.5 MySQL 语句操作 .....</b>	<b>334</b>	19.3.1 应用 insert 命令动态添加公告信息.....	365
17.5.1 插入记录.....	334	19.3.2 应用 select 命令查询公告信息.....	368
17.5.2 浏览数据库记录.....	335	19.3.3 解决截取公告主题乱码问题.....	369
17.5.3 修改记录.....	336	19.3.4 分页显示公告信息.....	371
17.5.4 删除记录.....	336	19.3.5 应用 update 命令动态编辑公告信息.....	374
<b>17.6 实战 .....</b>	<b>337</b>	19.3.6 应用 delete 命令动态删除公告信息.....	376
17.7 本章小结 .....	338	<b>19.4 实战 .....</b>	377
17.8 实战练习 .....	338	19.4.1 输入页码跳转到指定页 .....	377
<b>第 18 章 phpMyAdmin 图形化管理工具 .....</b>	<b>339</b>	19.4.2 图片的分栏分页显示.....	379
<b>    <b>视频讲解：57分钟</b></b>		19.4.3 留言板.....	383
18.1 phpMyAdmin 简介 .....	340	<b>19.5 本章小结 .....</b>	386
18.2 phpMyAdmin 的安装和配置 .....	340	<b>19.6 实战练习 .....</b>	386
18.2.1 Windows 系统下 phpMyAdmin 的安装配置.....	340	<b>第 20 章 ADODB 类库 .....</b>	387
18.2.2 Linux 系统下 phpMyAdmin 的安装配置 ...	341	<b>    <b>视频讲解：96分钟</b></b>	
18.3 phpMyAdmin 的使用 .....	342	20.1 ADODB 类库是什么 .....	388
18.3.1 管理数据库.....	342	20.2 ADODB 支持的数据库 .....	388
18.3.2 管理数据表.....	344	20.3 ADODB 下载与安装 .....	389
18.3.3 管理数据记录.....	346	20.4 ADODB 类库 .....	390
18.3.4 导入导出数据.....	349	20.4.1 连接数据库函数.....	390
<b>18.4 实战 .....</b>	<b>350</b>	20.4.2 操作数据库函数.....	392
18.5 本章小结 .....	351	20.4.3 控制结果集存取方式函数.....	396
18.6 实战练习 .....	352	20.4.4 操作结果集函数.....	397
<b>第 19 章 使用 MySQL 函数操作数据库 .....</b>	<b>353</b>	20.4.5 处理事务函数.....	400
<b>    <b>视频讲解：182分钟</b></b>		20.4.6 生成 HTML 表格函数 .....	400
19.1 PHP 访问 MySQL 数据库的 一般步骤 .....	354	20.4.7 生成下拉列表框函数.....	401
19.2 使用 MySQL 函数库 .....	355	20.4.8 实现分页功能函数.....	402
19.2.1 连接 MySQL 服务器.....	355	20.4.9 错误处理函数.....	403
19.2.2 选择数据库.....	356	<b>20.5 实战 .....</b>	404
19.2.3 执行 SQL 语句 .....	357	20.5.1 实现分页.....	404
19.2.4 应用 mysql_fetch_array()函数从数组 结果集中获取信息.....	358	20.5.2 处理事务.....	405

20.5.3 缓存函数+ADODB 动态生成静态页.....	406
20.6 本章小结 .....	412
20.7 实战练习 .....	412
<b>第 21 章 程序调试与异常处理.....</b>	<b>413</b>
<b><u>视频讲解：70分钟</u></b>	
21.1 程序基本调试流程 .....	414
21.2 PHP 中的错误类型 .....	414
21.2.1 语法错误.....	415
21.2.2 语义错误.....	417
21.2.3 逻辑错误.....	417
21.2.4 注释错误.....	418
21.2.5 运行错误.....	418
21.3 PHP 错误的调试.....	418
21.3.1 PHP 的错误报告 .....	418
21.3.2 启动错误报告.....	419
21.3.3 使用 print 语句调试程序 .....	419
21.3.4 应用@前缀字符屏蔽 PHP 脚本错误提示... .....	420
21.3.5 使用错误处理器记录日志.....	421
21.4 SQL 错误的调试.....	422
21.4.1 PHP 与 MySQL 连接错误 .....	422
21.4.2 SQL 语句错误 .....	423
<b>21.5 实战 .....</b>	<b>425</b>
21.6 本章小结 .....	427
21.7 实战练习 .....	428

## 第 3 部分

## 高级应用

<b>第 22 章 XML 技术.....</b>	<b>431</b>
<b><u>视频讲解：79分钟</u></b>	
22.1 XML 概述 .....	432
22.2 XML 语法 .....	432
22.2.1 XML 文档结构.....	432
22.2.2 XML 声明.....	433
22.2.3 XML 处理指令.....	433
22.2.4 XML 注释.....	433
22.2.5 XML 元素.....	434
22.2.6 XML 属性.....	436
22.3 创建 XML 文档 .....	436
22.4 SimpleXML .....	437
22.4.1 创建 SimpleXML 对象 .....	438
22.4.2 遍历所有子元素.....	439
22.4.3 遍历所有属性.....	439
22.4.4 访问特定节点元素和属性.....	440
22.4.5 修改 XML 数据 .....	441
22.4.6 保存 XML 文档.....	442
<b>22.5 实战 .....</b>	<b>443</b>
22.5.1 动态创建 XML 文档 .....	443
22.5.2 动态读取 XML 文档 .....	446
22.5.3 通过 XPath 查询 XML 中的数据 .....	448
22.6 本章小结 .....	449
22.7 实战练习 .....	450
<b>第 23 章 Ajax 技术.....</b>	<b>451</b>
<b><u>视频讲解：77分钟</u></b>	
23.1 Ajax 概述 .....	452
23.1.1 什么是 Ajax .....	452
23.1.2 Ajax 的开发模式.....	452
23.1.3 Ajax 的优点 .....	452
23.2 Ajax 技术的组成 .....	453
23.2.1 JavaScript 脚本语言 .....	453
23.2.2 XMLHttpRequest 对象.....	453
23.2.3 XML 语言 .....	458
23.2.4 DOM .....	458
23.2.5 CSS .....	458
23.3 Ajax 与 PHP 的交互 .....	458
23.3.1 通过 GET 方式与 PHP 进行交互.....	459
23.3.2 通过 POST 方式与 PHP 进行交互.....	461
23.4 Ajax 开发常见问题 .....	463
<b>23.5 实战 .....</b>	<b>464</b>
23.5.1 在 PHP 中应用 Ajax 技术检测用户名 .....	465
23.5.2 在 PHP 中应用 Ajax 技术实现博客文章类别添加.....	467

23.5.3 使用 jQuery 制作级联下拉列表框	469	24.3.2 注释	479
23.6 本章小结	471	24.3.3 变量	479
23.7 实战练习	471	24.3.4 修饰变量	482
<b>第 24 章 Smarty 模板技术</b>	<b>473</b>	24.3.5 流程控制	483
<b>视频讲解：102 分钟</b>		24.3.6 内建函数	486
24.1 Smarty 简介	474	24.4 Smarty 程序设计	488
24.1.1 Smarty 模板引擎	474	24.4.1 Smarty 中的常用方法	488
24.1.2 开发模式与运行流程	474	24.4.2 Smarty 的配置变量	489
24.2 Smarty 的安装配置	475	<b>24.5 实战</b>	<b>490</b>
24.2.1 Smarty 下载和安装	475	24.5.1 通过 Smarty 实现网页的动静分离	490
24.2.2 Smarty 配置	475	24.5.2 Smarty 分页类	493
24.2.3 第一个 Smarty 程序	477	24.5.3 网站公告	496
24.3 Smarty 模板设计	478	24.6 本章小结	498
24.3.1 Smarty 模板文件	479	24.7 实战练习	498

## 第 4 部分 项目实战

<b>第 25 章 博客管理系统</b>	<b>501</b>	25.6.4 查询用户信息	515
<b>视频讲解：169 分钟</b>		25.6.5 浏览用户信息	517
25.1 项目描述	502	25.7 文章管理模块设计	518
25.2 系统分析	502	25.7.1 文章管理模块概述	518
25.2.1 需求分析	502	25.7.2 文章管理模块技术分析	519
25.2.2 系统目标	503	25.7.3 添加文章的实现过程	521
25.2.3 系统功能结构	503	25.7.4 文章列表的实现过程	522
25.2.4 系统流程图	504	25.7.5 查看、评论文章的实现过程	524
25.3 软件开发环境	504	25.7.6 删除文章或评论的实现过程	526
25.4 数据库与数据表的设计	505	25.8 图片上传模块设计	527
25.4.1 数据库分析	505	25.8.1 图片上传模块概述	527
25.4.2 数据库 E-R 图设计	505	25.8.2 图片上传模块技术分析	528
25.4.3 创建数据库和数据表	506	25.8.3 图片上传的实现过程	532
25.5 首页设计	508	25.8.4 图片浏览的实现过程	535
25.5.1 首页概述	508	25.8.5 删除图片的实现过程	537
25.5.2 首页技术分析	508	25.9 朋友圈模块设计	538
25.5.3 首页的实现过程	510	25.9.1 朋友圈模块概述	538
25.6 用户管理模块设计	511	25.9.2 朋友圈模块技术分析	538
25.6.1 用户管理模块概述	511	25.9.3 查询好友的实现过程	539
25.6.2 用户注册	511	25.10 项目发布	541
25.6.3 用户登录	514	25.10.1 通过个人服务器发布项目	541

25.10.2 通过租用空间发布项目 .....	541	26.12.2 显示购物车 .....	584
25.11 开发常见问题与解决 .....	543	26.12.3 更改商品数量 .....	586
25.12 本章小结 .....	544	26.12.4 删除商品 .....	587
<b>第 26 章 应用 Smarty 模板开发</b>		26.12.5 保存购物车 .....	589
<b>电子商务平台 .....</b>	<b>545</b>	<b>26.13 收银台模块设计 .....</b>	<b>591</b>
<b>视频讲解：281 分钟</b>		26.13.1 显示订单 .....	592
26.1 项目描述 .....	546	26.13.2 填写订单 .....	592
26.2 系统分析 .....	546	26.13.3 处理订单 .....	593
26.2.1 需求分析 .....	546	26.13.4 反馈订单 .....	595
26.2.2 编写项目计划书 .....	546	26.13.5 查询订单 .....	596
26.3 系统设计 .....	551	26.14 后台首页设计 .....	599
26.3.1 系统目标 .....	551	26.14.1 后台首页布局 .....	599
26.3.2 系统功能结构 .....	551	26.14.2 DIV+JavaScript+CSS 实现树状	
26.3.3 系统流程图 .....	552	菜单 .....	600
26.4 系统开发环境 .....	553	26.15 类别管理模块设计 .....	601
26.5 数据库与数据表的设计 .....	553	26.15.1 添加类别 .....	601
26.5.1 数据库分析 .....	553	26.15.2 查看类别 .....	604
26.5.2 创建数据库和数据表 .....	555	26.16 订单管理模块设计 .....	608
26.6 搭建系统框架 .....	556	26.17 开发的常见问题 .....	611
26.7 公共文件设计 .....	557	26.17.1 解决 Ajax 的乱码问题 .....	611
26.7.1 数据库连接文件 .....	557	26.17.2 使用 JS 脚本获取、输出标签内容 .....	611
26.7.2 Smarty 配置文件 .....	558	26.17.3 使用浮动框架做关联菜单 .....	612
26.7.3 XMLHttpRequest 对象文件 .....	558	26.17.4 禁用页面缓存 .....	612
26.7.4 表单样式文件 .....	558	26.17.5 在新窗口中使用 Session .....	613
26.8 前台首页设计 .....	561	26.17.6 防止站外链接 .....	613
26.9 登录模块设计 .....	562	26.17.7 判断上传文件格式 .....	613
26.9.1 用户注册 .....	563	26.18 网站运行错误分析 .....	614
26.9.2 用户登录 .....	569	26.18.1 数据库导致的错误 .....	614
26.9.3 找回密码 .....	571	26.18.2 Smarty 模板的配置错误 .....	615
26.10 会员信息模块设计 .....	576	26.19 本章小结 .....	616
26.10.1 会员中心 .....	576	<b>第 27 章 办公自动化系统 .....</b>	<b>617</b>
26.10.2 安全退出 .....	579	27.1 项目描述 .....	618
26.11 商品展示模块 .....	579	27.2 技术准备 .....	619
26.11.1 创建 PHP 页 .....	580	27.3 开发过程 .....	620
26.11.2 创建模板页 .....	580	27.3.1 数据库设计 .....	620
26.11.3 js 脚本页面 .....	581	27.3.2 前台首页设计 .....	620
26.12 购物车模块设计 .....	582	27.3.3 人事消息模块设计 .....	621
26.12.1 添加商品 .....	582	27.3.4 考勤管理模块设计 .....	622



27.3.5 后台首页设计 .....	623	附录 A PHP 编码规则 .....	641
27.3.6 部门管理模块设计 .....	623	A.1 PHP File 文件格式 .....	641
27.4 本章小结 .....	624	A.1.1 常规 .....	641
<b>第 28 章 供求信息网 .....</b>	<b>625</b>	A.1.2 缩进 .....	641
28.1 项目描述 .....	626	A.1.3 行的最大长度 .....	641
28.2 技术准备 .....	626	A.1.4 行结束标志 .....	641
28.3 开发过程 .....	627	A.1.5 引号 .....	642
28.3.1 数据库设计 .....	627	A.1.6 运算符、小括号、空格和关键词 .....	642
28.3.2 前台功能设计 .....	627	A.2 命名约定 .....	643
28.3.3 免费供求信息发布模块设计 .....	628	A.2.1 类 .....	643
28.3.4 信息检索模块设计 .....	629	A.2.2 接口 .....	643
28.3.5 后台功能模块设计 .....	630	A.2.3 文件名 .....	643
28.3.6 付费供求信息发布模块设计 .....	631	A.2.4 函数和方法 .....	644
28.4 本章小结 .....	632	A.2.5 变量 .....	645
<b>第 29 章 图书借阅管理系统 .....</b>	<b>633</b>	A.2.6 常量 .....	645
29.1 项目描述 .....	634	A.2.7 数据库表和字段 .....	645
29.2 技术准备 .....	635	A.3 编码风格 .....	646
29.3 开发过程 .....	635	A.3.1 PHP 代码划分 (Demarcation) .....	646
29.3.1 数据库设计 .....	635	A.3.2 字符串 .....	646
29.3.2 功能模块概述 .....	636	A.3.3 数组 .....	647
29.3.3 管理员模块 .....	636	A.3.4 类 .....	647
29.3.4 图书档案管理模块 .....	638	A.3.5 函数和方法 .....	648
29.3.5 图书借还模块 .....	638	A.3.6 控制语句 .....	649
29.4 本章小结 .....	640	A.3.7 数据库 .....	650
		A.3.8 注释文档 .....	651



# 第1部分

## 基础知识

- » 第1章 初识 PHP
- » 第2章 PHP 环境搭建
- » 第3章 PHP 语言基础
- » 第4章 流程控制语句
- » 第5章 函数
- » 第6章 字符串
- » 第7章 正则表达式
- » 第8章 数组
- » 第9章 PHP 与 Web 页面交互
- » 第10章 PHP 与 JavaScript 交互
- » 第11章 日期和时间



# 財咗醫基

PHD 開業 每一趨勢

基層幹事會 PHD 基層幹事會

欲乎始  
如醫

PDG

# 第

# 1 章

## 初识 PHP

( 视频讲解: 7分钟)

PHP 是一种服务器端的嵌入式脚本语言，是一种跨平台、面向对象、HTML 嵌入式的脚本语言。本章将向读者简单介绍 PHP 语言和 PHP 5 的新特性、PHP 的语言优势、常用代码编辑工具，以及学好 PHP 语言的方法等，其主要目的是让读者对 PHP 语言有一个整体的了解和认识，然后再循序渐进地学习，最后达到完全掌握并精通 PHP 语言的目的。

通过阅读本章，可以学习到：

- PHP 是什么
- PHP 5 的新特性
- PHP 的语言优势
- PHP 语言的相关术语
- 学好 PHP 的方法
- PHP 常用的代码编辑工具
- 了解 PHP 的相关学习资源

## 1.1 PHP 概述

### 1.1.1 PHP 是什么

PHP 起源于 1995 年，由 Rasmus Lerdorf 开发。PHP 是 Hypertext Preprocessor（超文本预处理器）的缩写，是一种开源、跨平台、独立于架构、解释型、面向对象、快速安全、简单易学、性能优越的 Web 服务器端动态网页开发语言。目前已有超过 2200 万个网站、1.5 万家公司、450 万程序开发人员在使用 PHP 语言，它是目前动态网页开发中使用最为广泛的语言之一。

### 1.1.2 PHP 5 的新特性

PHP 5 中的对象已经进行了较系统、较全面的调整，下面着重讲述 PHP 5 中新的对象模式。

(1) PHP 5 采用全新的内存管理机制，使其在多线程的环境下可以更有效地运行。

(2) 在 PHP 5 全新的面向对象模型中，引入了全新的面向对象机制。

构造函数和析构函数。

对象的引用。

对象的克隆（clone）。

对象中的私有、公共及受保护模式（public/private 和 protected 关键字）。

接口（Interface）。

抽象类。

\_\_call。

\_\_set 和 \_\_get。

静态成员。

(3) 引入“try…catch”异常处理机制。

(4) 全新的 XML 支持。

### 1.1.3 PHP 的语言优势

PHP 起源于自由软件，即开放源代码软件，使用 PHP 进行 Web 应用程序的开发具有以下语言优势。

(1) 安全性高：PHP 是开源软件，每个人都可以看到所有 PHP 的源代码，程序代码与 Apache 编译在一起的方式也可以让它具有灵活的安全设定，PHP 具有了公认的安全性能。

(2) 跨平台：PHP 几乎支持所有的操作系统平台（如 Win32 或 UNIX/Linux/Macintosh/FreeBSD/OS2 等），并且支持 Apache、IIS 等多种 Web 服务器，并以此广为流行。

(3) 支持广泛的数据库：可操纵多种主流与非主流的数据库，如 MySQL、Access、SQL Server、Oracle、DB2 等，其中 PHP 与 MySQL 是现在最佳的组合，它们的组合可以跨平台运行。

(4) 简单易学：PHP 嵌入在 HTML 语言中，以脚本语言为主，内置丰富函数，语法简单、书写



容易，方便学习掌握。

(5) 执行速度快：占用系统资源少，代码执行速度快。

(6) 开发成本低：在流行的企业应用 LAMP 平台中，Linux、Apache、MySQL 和 PHP 都是免费软件，这种开源免费的框架结构可以为网站经营者节省很大一笔开支。

(7) 模板化：实现程序逻辑与用户界面分离。

(8) 支持面向对象：支持面向对象和过程的两种风格开发，并可向下兼容。面向对象编程（OOP）是当前的软件开发趋势。PHP 对 OOP 提供了良好的支持，可以使用 OOP 的思想来进行 PHP 的高级编程，这对于提高 PHP 编程能力和规划好 Web 开发架构都非常有意义。

(9) 内嵌 Zend 加速引擎，性能稳定快速。

(10) 应用范围广：PHP 技术在 Web 开发的各个方面应用得非常广泛。世界上很多大公司都采用了 PHP 技术，如德意志银行的交易系统，华尔街的股票在线买卖，汉莎航空公司的票务处理，甚至美联储、宇航局都采用了 PHP 技术。

#### 1.1.4 PHP 语言的发展

TIOBE 世界编程语言排行榜在一定程度上体现了编程语言在当前的流行趋势。TIOBE 最新公布了 2009 年 3 月的编程语言排行榜，除 Java、C、C++ 依旧强势占据三甲的位置外，前十名的其他语言排位均有小幅波动。本期值得关注的是 PHP 语言。在本期榜单中，Visual Basic 在坚守了两个月第四名的位置后，本期下降一位，与上期排名第五的 PHP 互换位置。2009 年 3 月份 TIOBE 世界编程语言排行的相关数据说明如图 1.1 所示。

TIOBE 编程语言排行榜衡量了各种编程语言的流行程度。该排行榜每月发布一次，统计数据包括全球范围的软件工程师、培训课程以及第三方供应商，数据来自 Google、MSN 和 Yahoo! 等流行搜索引擎。

 说明：TIOBE 声称，该排行榜并不是要评选最优秀的编程语言或是统计完成编程量最大的语言，但它可以帮助用户了解自己掌握的编程语言是否跟得上时代的发展，并且在开发新的软件系统需要对编程语言做策略性选择时提供参考。

PHP 是在 1994 年由 Rasmus Lerdorf 创建的，最初只是一个简单的用 Perl 语言编写的统计他自己网站访问者数量的程序。后来重新用 C 语言编写，同时可以访问数据库。1995 年，Personal Home Page Tools (PHP Tools) 开始对外发表第一个版本，Rasmus Lerdorf 发布了 PHP 1.0。

此后，越来越多的网站开始使用 PHP，并且强烈要求增加一些特性，如循环语句和数组变量等。在新的成员加入开发行列之后，1995 年的第二版 PHP/FI (Form Interpreter) 加入了对 mSQL 的支持，PHP 2.0 的发布，确立了 PHP 在动态网页开发上的地位。

到 1996 年底，有 15000 个网站使用 PHP/FI。

到 1997 年中，使用 PHP/FI 的网站超过 5 万个。

随着 PHP 5.0 的发布和更多对面向对象的支持，PHP 正不断巩固自己在 Web 开发领域的王者地位。

近几年 PHP 的发展呈现上升趋势，如图 1.2 所示，这也说明了 PHP 语言简单、易学、面向对象和安全等特点正在被更多人所认同。相信新的 PHP 语言将会朝着更加企业化的方向迈进，并且将更适合大型系统的开发。

Position Mar 2009	Position Mar 2008	Delta in Position	Programming Language	Ratings Mar 2009	Delta Mar 2008	Status
1	1	-0	Java	10.797%	+0.09%	A
2	2	-0	C	15.052%	+1.00%	A
3	5	+2	C++	10.357%	+1.09%	A
4	4	-1	PHP	9.482%	-0.41%	A
5	3	+2	Visual Basic	9.285%	-3.32%	A
6	7	+1	Python	5.185%	+0.42%	A
7	8	+1	C#	4.321%	-0.19%	A
8	10	+2	JavaScript	3.807%	+1.27%	A
9	6	+3	Perl	3.419%	-2.79%	A
10	9	+1	Delphi	2.710%	-0.09%	A
11	11	-0	Ruby	2.091%	+0.74%	A
12	12	-0	OC	1.839%	-0.62%	A
13	13	-0	PL/SQL	1.042%	-0.00%	A
14	22	+10	Logo	0.984%	+0.05%	A
15	14	+1	SQL	0.790%	-0.16%	A

图 1.1 2009 年 3 月 TIOBE 世界编程语言排名

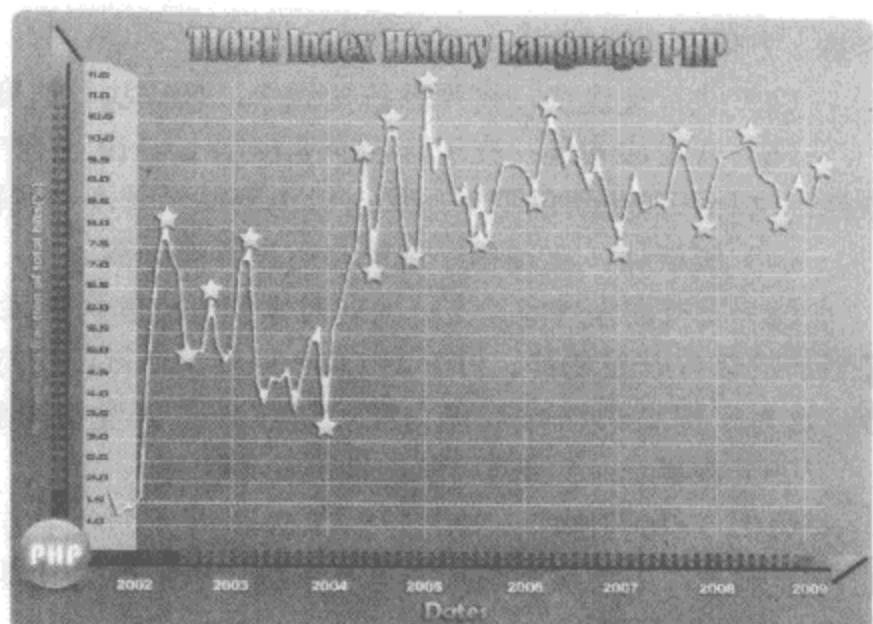


图 1.2 PHP 近几年的语言走势图

说明：图 1.1 和图 1.2 中的数据摘自 <http://www.tiobe.com> 网站。

PHP 凭借其代码开源、完全免费和安全性高等特性，正吸引着越来越多的 Web 开发人员。主流 PHP 5 的诞生、Zend II 引擎的采用，面向对象的支持以及模板化使得 PHP 编程进入了一个新时代，用户数量成稳步上升趋势。PHP 凭借其强大的功能，必将呈现出良好的发展前景。

PHP 无处不在，它可以应用于任何领域，并且已拥有几百万个用户，其发展速度要快于在它之前的任何一种计算机语言。PHP 能够给企业和最终用户带来数不尽的好处。据最新数据统计，全世界有超过 2200 万的网站和 1.5 万家公司在使用 PHP 语言，包括百度、雅虎、Google、YouTube、Digg 等著名网站，也包括汉沙航空电子订票系统、德意志银行的网上银行、华尔街在线的金融信息发布系统等，甚至军队系统这类苛刻的环境。除此之外，PHP 也是企业用来构建服务导向型、创造和混合 Web 融于新一代的综合性商业应用的语言，成为开源商业应用发展的方向。

## 1.2 PHP 与其他语言的比较

就目前的动态网页开发技术而言，除了 PHP 以外，还有 ASP、JSP 和.NET，它们都各有千秋，都有着广泛的用户群，本节将它们进行简单的比较，如表 1.1 所示。

表 1.1 PHP 与其他语言的比较

比较项目	PHP	ASP	JSP	.NET
跨操作系统性	支持	只支持 Win32	支持	只支持 Win32
Web 服务器	多	IIS	很多	IIS
执行效率	快	快	极快	极快
稳定性	高	低	高	高
开发敏捷度	高	高	中	高
支持语言	PHP	VBScript	Java	C#、VB、C++、JScript

续表

比较项目	PHP	ASP	JSP	.NET
函数支持	多	少	中	多
系统安全	高	低	高	高
版本升级	快	慢	慢	一般
难易程度	易	易	难	中

它们都各有所长，可以根据实际需要从中选择一种。不一定要选择最好的，但一定要选择最适合自己的。

### 1.3 PHP 语言相关术语

PHP 不仅是一门语言，而且已慢慢扩展成为一种行业。专业术语的记忆有助于对新语言的学习，特别是在阅读技术文档时，能帮助理解文档中大量的专业术语，这不仅能加快学习的进度，也能使用户更准确地理解原文的意思，以便在学习过程中少走弯路。下面介绍这门语言涉及的一些常用行业术语。

#### ☒ PHP

PHP 是 Hypertext Preprocessor（超文本预处理器）的缩写，是基于服务器端运行的脚本程序语言。可以实现数据库和网页之间的数据交互。PHP 的独特语法混合了 C 语言、Java 语言和 Perl 语言的特点。PHP 语言可以单独运行，也可以嵌入在 HTML 文件中，这样程序员就可以不必完全依赖 HTML 生成网页。因为 PHP 语言的执行都在服务器端，所以在客户端看不到 PHP 代码。

#### ☒ 类

类是对某个对象的定义，是属性和方法的集合，是面向对象编程方式的核心和基础，通过类可以将零散的用于实现某项功能的代码进行有效管理。实际上它本身并不是对象，因为它不存在于内存中。当引用类的代码运行时，类的一个新的实例（即对象）就在内存中创建了。通过类可以在内存中创建多个相同类型的对象。

#### ☒ 对象

对象是类进行实例化后的产物，是一个实体。对象就是类存在于内存中的实例。通过对类进行实例化，就能得到对应的对象。

#### ☒ 函数

函数是一段相对独立的代码，不同的语言有不同的定义方式。在 PHP 中，函数就是有效的 PHP 代码的集合。在设计复杂程序时，可以把程序分为几个函数，不同的函数完成不同的功能。其优点在于各部分充分独立，不仅可以使程序易读、易维护，还可以对代码进行复用。

#### ☒ Session

Session 用中文来解释就是会话期。在计算机专业术语中，Session 是指一个终端用户与交互系统进行通信的时间间隔，通常指从注册进入系统到注销退出系统所经过的时间。因此，Session 实际上是一个特定的时间概念。Session 是将 Session 的信息保存在服务器上，并通过一个 Session ID 来传递客户端的信息。服务器在接收到 Session ID 后根据这个 ID 来提供相关的 Session 信息资源。由于 Session 为服务器存储，远程用户没办法修改 Session 文件的内容。

**Cookie**

**Cookie** 是一种在远程浏览器端储存数据并以此来跟踪和识别用户的机制。简单地说，**Cookie** 是 Web 服务器暂时存储在用户硬盘上的一个文本文件，并随后被 Web 浏览器读取。在 PHP 中通过 `setcookie()` 函数创建 **Cookie**。在创建 **Cookie** 之前，首先必须了解的是，**Cookie** 是 HTTP 头标的组成部分，而头标必须在页面其他内容之前发送，它必须最先输出，即使在 `setcookie()` 函数前输出一个 HTML 标记或 `echo` 语句，甚至一个空行都会导致程序出错。

 **常量**

常量，即不可变的数值或数据项。常量的值通常是在程序运行前定义的，在整个程序运行的过程中，其值是不会改变的。一个常量由英文字母、下划线和数字组成，但数字不能作为首字母出现。常量默认为大小写敏感。按照惯例，常量标识符总是大写的。

 **变量**

和很多语言不同，在 PHP 中使用变量之前不需要声明变量（PHP 4 之前需要声明变量）。PHP 中的变量用一个美元符号“\$”后面跟变量名来表示。PHP 中的变量名是区分大小写的。

 **引用**

在 PHP 中引用就是用不同的名字访问同一个变量内容。变量名和变量内容是不一样的，因此同样的内容可以有不同的名字。

 **构造函数**

构造函数又称为构造器，是类中的一个特殊函数。在 PHP 5 中，构造函数使用统一的名称“`_construct`”进行声明。在使用 `new` 操作符创建对象时，构造函数将会被自动调用。如果一个类没有构造函数，则调用基类的构造函数，如果有则调用当前类构造函数。一般用于在创建对象时进行初始化操作。

 **析构函数**

PHP 5 引入了析构函数的概念，通过定义“`_destruct`”函数来建立析构函数。析构函数与构造函数的功能正好相反，它用于销毁对象。析构函数会在到某个对象的所有引用都被删除或者当对象被显式销毁时执行。

在 PHP 语言的学习过程中，读者应该加强对 PHP 专业术语和其他通用专业术语的学习和积累。

## 1.4 资深程序员谈如何学习 PHP

怎样学好 PHP 语言，这是所有 PHP 程序员都会考虑的问题。其实，每种语言的学习方法都大同小异，相信大家选择 PHP 语言并不是因为它的背景和悠久历史，更重要的是它的实用性。

下面是笔者结合多年的开发实践所总结出来的学习经验，与所有 PHP 程序开发者分享：

- (1) 熟悉 HTML/CSS/JavaScript 等网页基本元素，完成阶段可自行制作完整的网页，对元素属性达到熟悉程度。
- (2) 学会配置 PHP 的开发环境，并选择一种适合自己的开发工具。
- (3) 理解动态语言的概念，运作机制，熟悉 PHP 语法。
- (4) 学习如何将 PHP 与 HTML 结合起来完成简单动态页面。

- (5) 接触 MySQL 数据库，开始设计数据库方面的程序。
- (6) 不断巩固，熟练使用大部分 PHP 常用函数，并理解 OOP、MySQL 优化，以及模板。
- (7) 开发一个功能齐全的动态站点。

这套学习方案是一个循序渐进的过程，初学者不要看到上面的短短几行文字就以为学习起来很简单，任何技术其实都不简单，即使是小吃部的美味菜肴也不是一下子就能做成的。

下面解释一下这个学习过程。

任何网站都是由网页组成的，也就是说想完成网站，必须先学会做网页，因此必须要掌握 HTML 语言，才能为今后制作网站打下基础。掌握静态网页的制作技术是学习开发网站的先决条件。

在学习 HTML、CSS 和 JavaScript 的过程中，边学边做是最有效的方式，当然这一方式对于学习 PHP 同样是最有效的。HTML 中的任何元素都要亲自实践，只有熟悉各个元素的效果之后，才会记忆深刻，而对于书本中的知识只是一味地死记硬背是绝对不行的。大部分新手之所以觉得概念难学，大部分是因为“懒惰”，懒惰是阻止一个人走向成功的最大敌人，所以克服懒惰的习惯，才能更快地学好一门语言。

另外，扎实的基础对于一个程序员来说尤为重要，学习知识切记不可急于求成，必须要打好基础，再进阶学习有一定难度的技术，才能轻松地开发网站。而那些不重视“地基”，一心要打造“空中楼阁”的程序员又怎能开发出好的网络程序呢？所以，建议读者多阅读一些基础教材，了解基本的编程知识。

在熟悉了 HTML/CSS/JavaScript 等网页基本元素后，就要开始接触 PHP 了，首先需要做的就是搭建 PHP 环境和选择一款开发工具。

搭建 PHP 环境的方法有很多，可以自定义安装 Apache 服务器、MySQL 服务器和 PHP，然后手动配置环境。但是对于新手来说，最简单的方法就是使用集成化安装包，因为集成化安装包操作起来非常方便，可以直接安装 Apache 服务器、MySQL 服务器、PHP 和 phpMyAdmin，而且不需要用户手动配置，安装完以后即可直接使用。像这种集成化安装包网络上非常多，如使用 AppServ 集成化安装包（第 2 章有详细介绍）。

PHP 的编辑工具有很多，每款编辑工具都有其各自的优势。在编写程序时，一款好的编辑工具会使程序员编写过程更加轻松、有效和快捷，达到事半功倍的效果。在本章 1.5 节中推荐了几款开发工具，读者可以结合自己的需求进行选择。

在理解了 HTML，并熟悉了 PHP 的概念和语法后，进行 PHP 和 HTML 混合编程应该不成问题，在此期间，可以用 PHP 为自己设计一个漂亮的页面，然后在浏览器上输出此时此刻最想说的话，不要觉得幼稚，虽然你输入的是一小段代码，但是对于你的编程之路，可是迈出了整整一大步。

在高兴之余就必须继续努力了，学习 PHP 与数据库的操作。MySQL 数据库可谓是 PHP 的黄金搭档。等到你发奋努力学会了用 PHP 成功地添加、修改、删除数据时，显然距离成功已经指日可待了。巩固了自己的知识，熟悉了 PHP 和 MySQL 开发的要领之后，开发一个留言本，实现一个计数器，你就能充分体会到成就感了。

此时往往是程序员能否精通 PHP 的关键，为什么这么说呢？因为很多程序员当发现自己可以做很多例子，可以动态操作数据库时，就以为学会了 PHP、精通了 PHP。其实，还早呢！你还没有接触 PHP 的精髓 OOP、XML、Ajax 和模板呢！所以，这时候要不懈努力，一一去接触这些技术，精通这些技术之后，你会发现，PHP 的应用是如此的广泛。

总而言之，PHP 的不断升级会让你有学不完的新技术，这里可以用学无止境来形容学习 PHP 的过程。另外，学习 PHP 还有一个重点中的重点——学习编码规范，以此来养成一个良好的编程习惯。

在学习 PHP 的过程中，会接触到大量的函数，PHP 的函数有上千种，下载一个 PHP 中文手册和 MySQL 手册，或者身边备上一两本 PHP 函数类的相关书籍，就可以快速提高编程效率，解决程序中的异常，使其成为你的编程助手。

其实，学习任何一门语言就是一个多实践，多思考，多请教的过程。不要死记语法，特别是对于刚接触 PHP 语言的程序员，掌握好基本语法，反复实践，仅能读懂书本中的内容和技术是不行的，必须动手编写程序代码，并运行程序、分析运行结构，让大脑对学习内容有个整体的认识和肯定，用自己的方式去思考问题，编写代码来提高编程思想。平时多借鉴网上一些好的功能模块，培养自己的编程思想；多向他人请教，学习他人的编程思想；多与他人沟通技术问题，提高自己的技术和见识，这样可以快速地进入学习状态。借鉴别人成功的代码，绝对是受益匪浅，因为多研究那些经过千锤百炼的经典代码，是提高编程水平的最好方法。另外，目前很多 PHP 书籍都配有视频录像，可以看一些典型的视频录像，从而了解他人的编程思想，快速提高自身的编程水平。

学一门技术最忌急躁，不能遇到技术问题就放弃。遇到问题必须冷静对待，不要让自己的大脑思绪紊乱，保持清醒的头脑才能分析和解决各种问题。可以尝试听歌、散步、玩游戏等活动放松自己。另外还要尝试自己解决，这样可以提高自己的程序调试能力，并能提高解决常见问题的能力。

学习 PHP 不仅要掌握方法，更多的是付出汗水；同时不能轻言放弃，相信自己，相信自己的选择，更要相信自己的能力，如果自己想放弃，不如想想这句话——努力不一定成功，但放弃必定失败！

最后要叮嘱各位的是——抵御诱惑。本章 1.2 节中讲解的 PHP/ASP/JSP/.NET 的对比也许会让你无所适从，你也许学了一半 PHP，又开始打.NET 的主意，或者说 Java 很强，这个时候绝对不能动摇，要明确自己的学习目标和方向，选择并锁定一门语言，按照自己的既定目标努力学习和认真研究，直到精通。切记不要哪门都学，哪门都不精。哪怕你真想学，也得学会了 PHP，然后再学。见异思迁是最不可取的，狗熊掰玉米就是这个道理，如果经常中途放弃，只能是一无所获，还浪费了宝贵的时间和经历，得不偿失。

因此，必须要强迫自己完成自己的目标，要有坚持不懈、持之以恒的精神。精通一门语言就是这么坚持下来的。

## 1.5 常用代码编辑工具

PHP 的开发工具很多，每种开发工具都有其各自的优势。在编写程序时，一款好的编辑工具会使程序员的编写过程更加轻松、有效和快捷，达到事半功倍的效果。因此读者可根据自己的需求有选择性地使用开发工具。

对于一款好的代码编辑工具，除了具备最基本的代码编辑功能外，一个必备的功能就是语法的高亮显示。应用语法的高亮显示，可以对代码中的不同元素采用不同的颜色进行显示，如关键字用蓝色、对象方法用红色标识等。另外，一款好的代码编辑工具应具备格式排版功能，格式排版功能可以使程序代码的组织结构清晰易懂，并且易于程序员进行程序调试，排除程序的错误异常。

下面介绍几款常用的代码编辑工具。

Macromedia Dreamweaver

Macromedia Dreamweaver 是一款专业的网站开发编辑器。它将可视布局工具、应用程序开发功能和代码编辑支持组合在一起，其功能强大，使得各个层次的开发人员和设计人员都能够快速创建出吸引人的、标准的网站和应用程序。它采用了多种先进的技术，能够快速高效地创建极具表现力和动感效果的网页，使网页创作过程简单无比。同时，Macromedia Dreamweaver 提供了代码自动完成功能，不但可以提高编写速度，而且减少了错误代码出现的几率。Macromedia Dreamweaver 既适用于初学者制作简单的网页，又适用于网站设计师、网站程序员开发各类大型应用程序，极大地方便了程序员对网站的开发与维护。

Macromedia Dreamweaver 从 MX 版本开始支持 PHP+MySQL 的可视化开发，对于初学者确实是较好的选择，因为如果是一般性开发，几乎是可以不用一行代码也可以写出一个程序，而且都是所见即所得的。所包含的特征包括：语法加亮、函数补全，形参提示、全局查找替换、处理 Flash 和图像编辑等。同时，可以为 PHP、ASP 等脚本语言提供辅助支持。

另外，Macromedia Dreamweaver 从第 8 版开始，提供代码折叠功能，可以将一个代码块，如一个方法或者一个类的代码块折叠起来用两行代替，需要时可以再展开。

下载地址：<http://www.adobe.com/downloads/>。

 注意：Macromedia Dreamweaver MX 是首选的网页开发工具。本书所介绍的网页和实例都是使用 Macromedia Dreamweaver 8 编辑的。

 Eclipse

Eclipse 是一款支持各种应用程序开发工具的编辑器，为程序设计师提供了许多强悍的功能。它支持多语言的关键字和语法加亮显示，支持查询结果匹配部分在编辑器中的加亮显示，支持代码格式化功能，还具备强大的调试功能，可以设置断点，使用单步执行方法执行源代码。

官方网站：<http://www.eclipse.org>。

 ZendStudio

ZendStudio 是目前公认的最强大的 PHP 开发工具，具备功能强大的专业编辑工具和调试工具，包括编辑、调试、配置 PHP 程序所需要的客户及服务器组件，支持 PHP 语法加亮显示，尤其是功能齐全的调试功能，让 PHP 错误不再可怕。ZendStudio 是一款收费软件，不过可以免费下载试用版。

下载地址：<http://www zend com/store/products/zend-studio.php>。

 PHPEdit

PHPEdit 是一款 Windows 操作系统下优秀的 PHP 脚本 IDE（集成开发环境）。该软件为快速、便捷地开发 PHP 脚本提供了多种工具，其功能包括：语法关键词高亮；代码提示、浏览；集成 PHP 调试工具；帮助生成器；自定义快捷方式；150 多个脚本命令；键盘模板；报告生成器；快速标记；插件等。

官方网站：<http://phpedit.svoi.net>。

 UltraEdit

UltraEdit 是能够满足用户一切编辑需要的编辑器。UltraEdit 是一套功能强大的文本编辑器，可以编辑文本、十六进制、ASCII 码，可以取代记事本程序，内建英文单字检查、C++ 及 VB 指令突显，可同时编辑多个文件，而且即使开启很大的文件速度也不会变慢。软件附有 HTML 标签颜色显示、搜寻替换以及无限制的还原功能，用户喜欢用其来修改 EXE 或 DLL 文件。

官方网站：<http://www.ultraedit.com.cn>。

## 1.6 学习资源

视频讲解：光盘\TM\Video\1\PHP 的学习资源.exe

作为一种被广泛使用的 Web 应用开发语言，PHP 有着极其丰富的学习资源。

### 1.6.1 PHP 参考手册

学习 PHP 语言，在电脑中必须配备一个 PHP 参考手册，就像在学习汉字时，身边必须配备一本《新华字典》一样。因为 PHP 参考手册是 PHP 语言的“新华字典”，它对 PHP 的函数进行了详细的讲解和说明，并且还给出了一些简单的示例，同时还对 PHP 的安装与配置、语言参考、安全和特点等一些信息进行了介绍。

在 PHP 的 <http://www.php.net/docs.php> 网站上，提供了 PHP 的各种语言、格式和版本的 PHP 参考手册，用户可以进行在线阅读，也可以打包下载。

PHP 参考手册不但对 PHP 的函数进行了解释和说明，而且还提供了快速查找的方法，让用户可以更加方便地查找到指定的函数。PHP 参考手册（下载版）的运行效果如图 1.3 所示。



图 1.3 PHP 参考手册

### 1.6.2 PHP 学习讨论

PHP 的官方网站 (<http://www.php.net>)

提供 PHP 各版本的下载，以及 PHP 手册下载。

喜悦国际村 (<http://www.phpx.com>)

目前国内最火爆的 PHP 讨论社区，注册用户数万，PHP 版面讨论激烈，PHP 高级工程师众多。

PHP 5 研究室 (<http://www.phpv.net>)

关注新技术的研究和跟踪，能够了解新技术，同时能够查询到最新版本的手册。

PHPChina (<http://www.phpchina.com>)

PHPChina 开源社区，有非常多的 PHP 源码，能够从中获取最新的 PHP 信息资源。

 Linux 和 UNIX 中文网站 (<http://www.chinaunix.net>)

全球最大的 Linux 和 UNIX 中文网站，其中的 PHP 版块资料极其丰富。

 PHPWind (<http://www.phpwind.com>)

PHPWind 论坛，是国内发展最迅速的论坛，代码质量比较高，执行速度快。

 Discuz! (<http://www.discuz.com/>)

国内最著名的 PHP 论坛，速度快、稳定性好、安全。易于定制，能够支持非常多的在线人数。目前该论坛已彻底免费。

## 1.7 成功案例

目前，互联网上很多网站的开发都是通过 PHP 语言来完成的，如搜狐、网易和百度等，在这些知名网站的创作开发中都应用到了 PHP 语言。

吉林省明日科技有限公司软件网、图书网、网络学院、编程词典网等也都是用 PHP 语言来完成的。感兴趣的读者可以登录下面的网站，不但可以浏览完全由 PHP 开发的网站，体会 PHP 语言的强大功能，同时也可以通过网站与我们进行交流、互相学习，也可以更深入地了解明日科技，了解我们的编程词典，了解我们的网络学院。

吉林省明日科技有限公司的图书网网址是 <http://www.mingribook.com>，如图 1.4 所示。

吉林省明日科技的网络学院的网址是 <http://www.cccxy.com>，如图 1.5 所示。

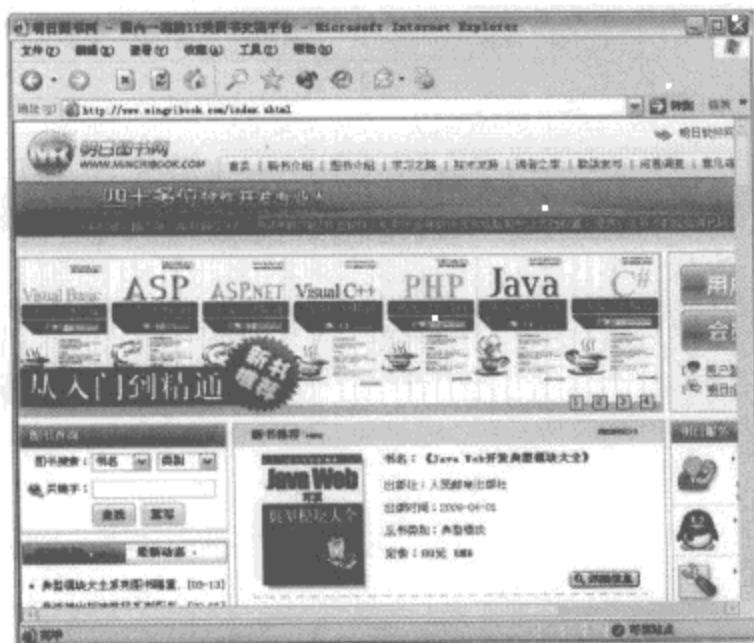


图 1.4 明日科技图书网首页

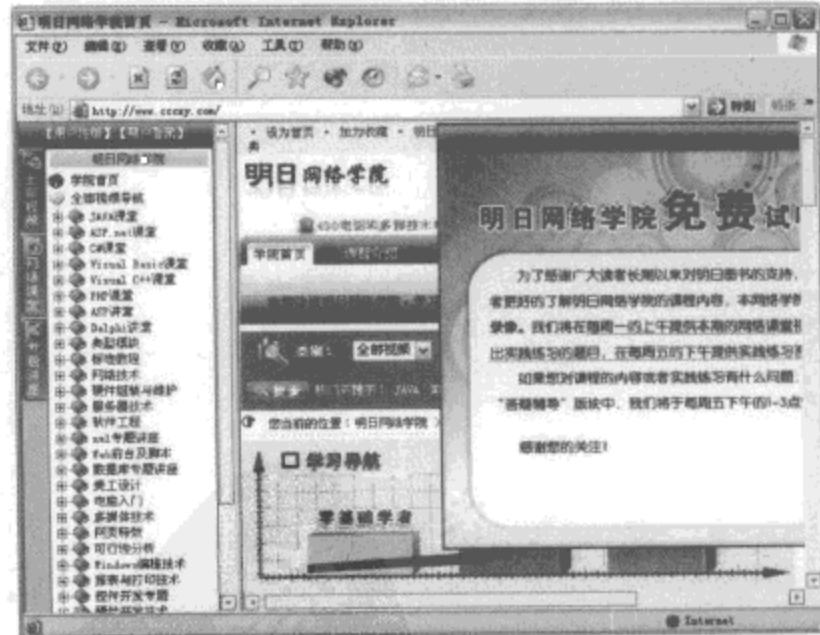


图 1.5 明日网络学院网站首页

吉林省明日科技的论坛网网址是 [http://www.mingribook.com/bbs/bbs\\_index.php](http://www.mingribook.com/bbs/bbs_index.php)，如图 1.6 所示。

吉林省明日科技的编程词典网的网址是 <http://www.mrbccd.com/index.shtml>，如图 1.7 所示。



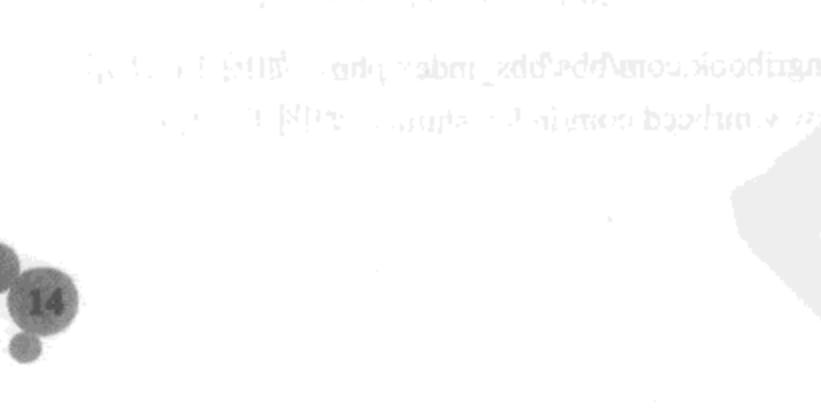
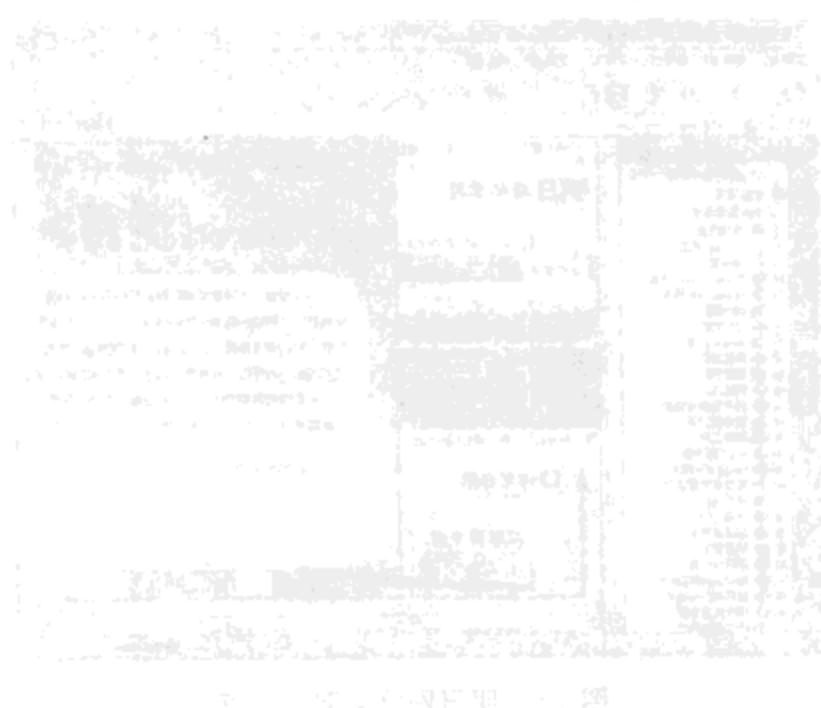
图 1.6 明日科技论坛网首页



图 1.7 明日科技编程词典服务网首页

## 1.8 本章小结

本章重点讲述了什么是 PHP, PHP 的语言优势以及应用领域, 详细介绍了 PHP 5 的新特性, 通过这些内容使读者对 PHP 有一个全面的认识。其次, 指导读者学习 PHP 的方法。另外, 本章的最后还介绍了常用代码编辑工具、网络上的学习资源以及一些经典的 PHP 成功案例, 以方便读者更好地学习 PHP。



# 第 2 章

## PHP 环境搭建

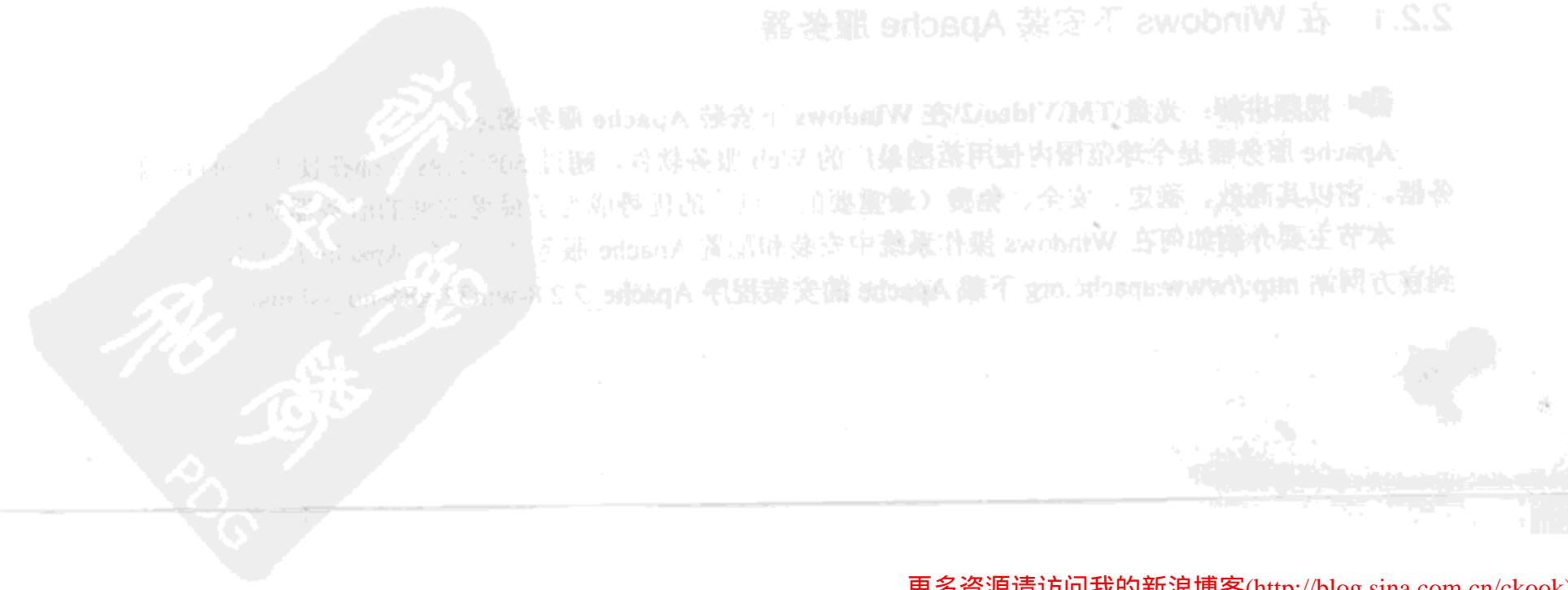
( 视频讲解: 80 分钟)

PHP 是全球最普及、应用最广泛的互联网开发语言之一。学习任何一门编程语言，在开始学习之前都要首先学会搭建和熟悉开发环境，本章将介绍两种操作系统（Windows 和 Linux）下的 Apache 服务器、MySQL 服务器及 PHP 的安装和配置方法。另外还特别为初学者介绍一种简化安装和配置 PHP 环境的组合包。最后，使用 Macromedia Dreamweaver 8 开发 PHP 第一个实例。

通过阅读本章，可以学习到：

- » 如何在 Windows/Linux 下安装和配置 Apache 服务器
- » 如何在 Windows/Linux 下安装和配置 MySQL 服务器
- » 如何在 Windows/Linux 下安装 PHP
- » 如何使用组合包快速搭建 PHP 环境
- » 如何应用开发工具编写、发布和运行第一个 PHP 程序
- » 如何独立解决常见的 PHP 环境安装问题

更多资源请访问我的新浪博客(<http://blog.sina.com.cn/ckook>)



## 2.1 搭建 PHP 开发环境的准备工作

### 2.1.1 在 Windows 下搭建 PHP 开发环境的准备工作

在 Windows 下搭建 PHP 与安装其他的一些软件工具不同。因为 PHP 是从 Linux 移植过来的一种语言，不仅在开发环境上尽量保留着 Linux 的特点（Apache 是 Linux 下的 Web 服务器，地位就像 Windows 下的 IIS；MySQL 也是 Linux 系统中捆绑的数据库），在安装上也被烙上了 Linux 印记。除了正常的安装操作外，还需要在各自的配置文件（.ini、.conf）中进行专门的设置。

安装之前要准备的安装包有：

- Apache\_2.2.8-win32-x86-no\_ssl.msi。下载地址为 <http://httpd.apache.org/download.cgi>。
- php-5.2.5-Win32.zip。下载地址为 <http://www.php.net/downloads.php>。
- mysql-essential-5.0.51a-win32.msi。下载地址为 <http://www.mysql.com/download/>（下载 MySQL 需要注册一个账号）。

### 2.1.2 在 Linux 下搭建 PHP 开发环境的准备工作

在 Linux 下搭建 PHP 环境比 Windows 下要复杂得多。除了 Apache、PHP 等软件外，还要安装一些相关工具，设置必要的参数。而且，要使用 PHP 扩展库，还要进行编译。如本书中使用到的 SOAP、MHASH 等扩展库。这里给出在 Linux 下安装的必要步骤。如果用户在安装过程中遇到特殊的问题，还需要翻阅 Linux 相关的书籍、手册。

安装之前要准备的安装包有：

- httpd-2.2.8.tar.gz。下载地址为 <http://www.apache.org>。
- php-5.2.5.tar.gz。下载地址为 <http://www.php.net/downloads.php>。
- mysql-5.0.51a-Linux-i686.tar.gz。下载地址为 <http://www.mysql.com>。
- libxml2-2.6.26.tar.gz。可在网络上直接搜索该版本进行下载。

## 2.2 Apache 服务器的安装和配置

### 2.2.1 在 Windows 下安装 Apache 服务器

#### ■ 视频讲解：光盘\TM\Video\2\Windows 下安装 Apache.exe

Apache 服务器是全球范围内使用范围最广的 Web 服务软件，超过 50% 的网站都在使用 Apache 服务器，它以其高效、稳定、安全、免费（最重要的一点）的优势成为了最受欢迎的服务器软件。

本节主要介绍如何在 Windows 操作系统中安装和配置 Apache 服务器。安装 Apache 服务器前，应到官方网站 <http://www.apache.org> 下载 Apache 的安装程序 Apache\_2.2.8-win32-x86-no\_ssl.msi。

在 Windows 下安装和配置 Apache 服务器的操作步骤如下。

(1) 双击 Apache\_2.2.8-win32-x86-no\_ssl.msi 文件，弹出欢迎页面。单击 Next 按钮，进入到许可协议页面。

(2) 在许可协议页面，用户需要同意页面中的条款才能继续安装。选中 I accept the terms in the license agreement 单选按钮，页面如图 2.1 所示。单击 Next 按钮进入到下一页。

(3) 本页面是对该程序的一个描述和说明。在了解了相关的信息后，单击 Next 按钮进入到 Server Information 页面。

(4) Server Information 页面需要用户填写域名、服务器名称和管理员 Email。Server Information 页面的填写效果如图 2.2 所示。该页面还有两个单选按钮，如果选中默认的第一个单选按钮，说明该服务器对所有人开放，并且服务器的端口号为 80，这个是推荐选项。第二个单选按钮是指该服务器仅对当前用户开放，并且服务器端口为 8080。这里选中第一个单选按钮。然后单击 Next 按钮进入到下一个页面。

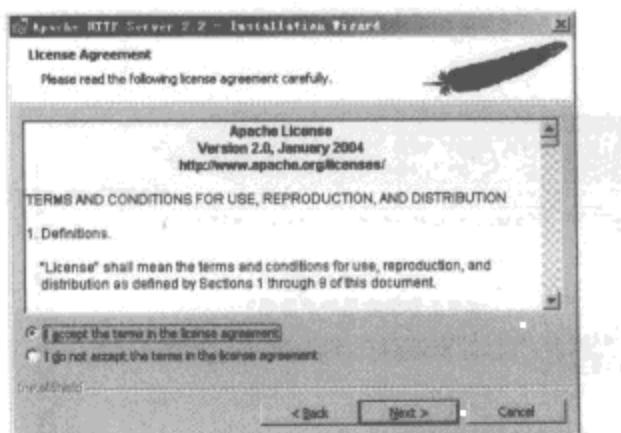


图 2.1 许可协议页面

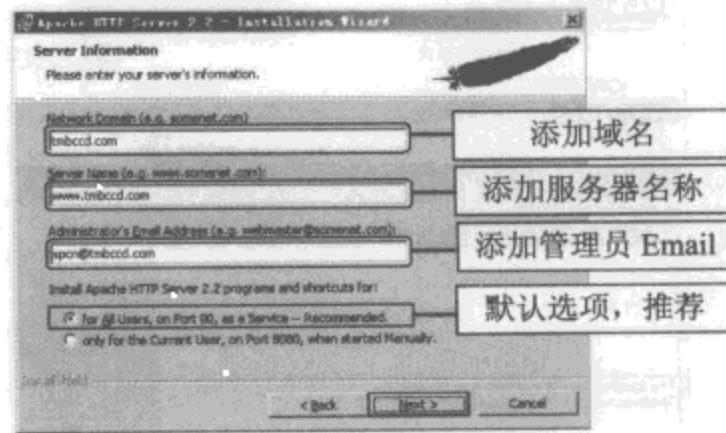


图 2.2 Server Information 页面

**注意：**如果用户的机器安装有“Internet 信息服务（IIS）管理器”，那么必须将此项服务停止，因为 IIS 服务器的默认端口号为 80，同 Apache 服务器默认端口号相同。如果 IIS 服务不停止，就会和 Apache 服务器的端口号产生冲突，Apache 服务器将不能成功安装。

(5) 如图 2.3 所示页面用于选择安装类型。安装类型分为典型安装和自定义安装，通常保持默认选项即可。单击 Next 按钮，进入到路径选取页面。

(6) 在路径选取页面中，单击 Change 按钮可以选择安装路径。这里路径设为“D:\Apache2.2\”，如图 2.4 所示。

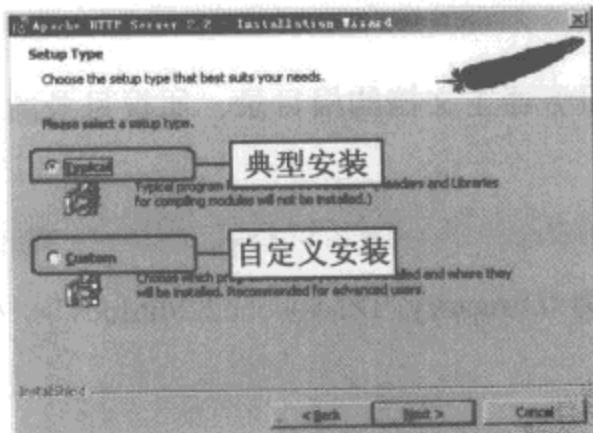


图 2.3 选择安装类型

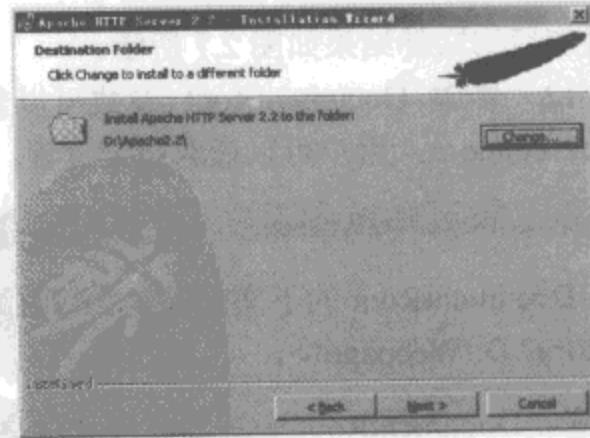


图 2.4 指定路径

(7) 单击 Next 按钮进入文件安装页面。这是 Apache 安装的最后一步，程序开始安装文件。安装结束后，单击 Finish 按钮结束安装程序。

(8) 安装完成后，Apache 服务器会自动开启。在系统托盘区域将出现一个图标，当前 Apache 服务启动时，图标样式为○；服务器未启动时，图标样式为●。

单击 Apache 服务器的启动小图标，将会看到服务器的开启与关闭功能；也可以用鼠标右键单击小图标，在弹出的快捷菜单中选择 Open Apache Monitor 命令，打开 Apache 监控程序，其操作效果如图 2.5 所示。

(9) 服务器开启后，最后需要测试一下服务器。打开 IE 浏览器页面，在地址栏中输入“http://127.0.0.1/”或“http://localhost/”，按 Enter 键后系统会显示如图 2.6 所示的页面，此时说明 Apache 服务器正式安装成功。

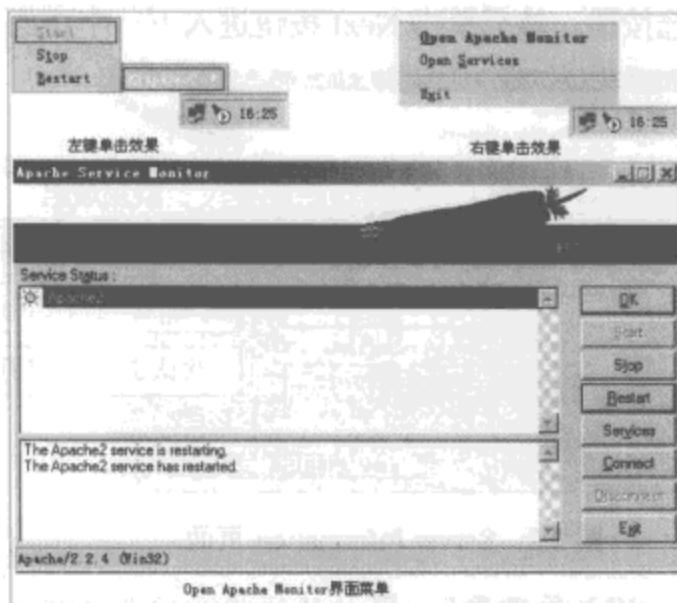


图 2.5 Apache 控制菜单

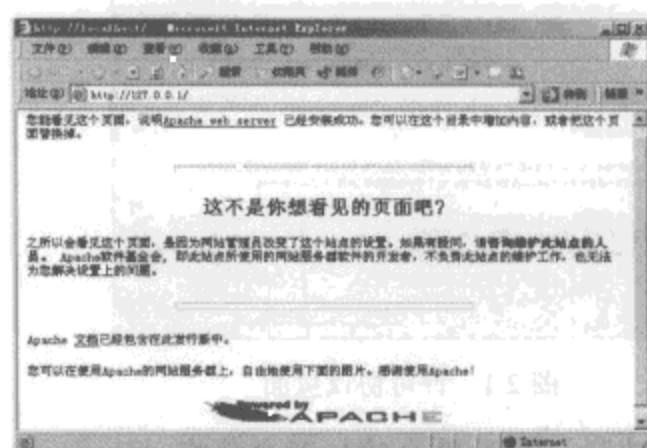


图 2.6 Apache 服务器运行页面

(10) Apache 服务器安装成功后，接下来需要对 Apache 服务器进行配置，以便 Apache 服务器能够识别 PHP 文件。配置 Apache 服务器主要是在 Apache 安装目录下的 conf 子目录中的 httpd.conf 文件中进行的，找到该文件并用记事本等文本编辑器打开该文件。

(11) 定位到 LoadModule 配置块，在 LoadModule 的最后添加如下信息：

```
LoadModule php5_module d:\php5\php5Apache2_2.dll
```

添加后的文件结果如图 2.7 所示。

(12) 修改 DocumentRoot 参数可以修改 Apache 服务器主文档的根目录。原根目录的位置是 Apache2.2\htdocs，用户可以任意指定位置。如：

```
DocumentRoot "D:/Webpage"
```

在 DocumentRoot 的下面间隔约 28 行的位置，有一行为<Directory "D:/Apache2.2/htdocs">，修改为<Directory "D:/Webpage">。

注意：DocumentRoot 和这里的参数值要保持一致。

(13) 添加 Apache 服务器能够识别的 PHP 扩展名。PHP 的扩展名有.php3、.php4、.php、.phtml 等。这里只推荐使用标准的扩展名.php。添加的代码如下：

```
AddType application/x-httdp-php .php
```

添加位置如图 2.8 所示。

```
LoadModule php5_module modules/mod_php5.so
```

图 2.7 加载 dll 文件

```
AddType application/x-httdp-php .php
```

图 2.8 添加 PHP 扩展识别

(14) 默认显示页。Apache 的默认显示页为 index.html。也就是说，在服务器未指名文件时，首先查找 index.html，如果找到 index.html，那么服务器就将加载该文件，否则显示目录内的文件列表。在这里添加一个 PHP 默认页：index.php。更改后的代码如下：

```
DirectoryIndex index.html index.php
```

(15) 修改 Apache 端口号。Apache 的端口号为 80。修改 Listen 选项的值，即可修改端口号。如改为 82，则更改后的代码如下：

```
Listen 82
```

以上配置完成后，重启 Apache 服务器即可。

**◆ 注意：**如果用户的计算机上还有 IIS 服务器，那么可能会因为端口冲突而导致 Apache 无法正常开启。解决的办法是改变其中的一个端口号，或者停止 IIS 服务器。

## 2.2.2 在 Linux 下安装 Apache 服务器

### 视频讲解：光盘\TM\Video\2\在 Linux 下安装 Apache 服务器.exe

在 Linux 下安装 Apache 需要到官方网站 <http://www.apache.org> 下载 Linux 下的 httpd2.2.8.tar.gz 的压缩包。

首先需要打开 Linux 终端（Linux 下几乎所有的软件都需要在终端下安装）。在 RedHat 9 的界面中选择“主菜单” / “系统工具”命令，在弹出的菜单中选择“终端”命令。

在 Linux 下安装和配置 Apache 服务器的操作步骤如下：

- (1) 进入到 Apache 安装文件的目录下，如 /usr/local/work。

```
cd /usr/local/work/
```

- (2) 解压安装包。完成后进入到 httpd2.2.8 目录中。

```
tar xfz httpd2.2.8.tar.gz
cd httpd2.2.8
```

(3) 建立 makefile，将 Apache 服务器安装到 user/local/Apache2 下。

```
/configure --prefix=/usr/local/Apache2 --enable-module=so
```

(4) 编译文件。

```
make
```

(5) 开始安装。

```
make install
```

(6) 安装完成后，将 Apache 服务添加到系统启动项中，重启服务器。

```
/usr/local/Apache2/bin/Apachectl start >> /etc/rc.d/rc.local
/usr/local/Apache2/bin/Apachectl restart
```

(7) 打开 Mozilla 浏览器，在地址栏中输入“<http://localhost/>”，按 Enter 键后如果看到如图 2.9 所示的页面，说明 Apache 服务器已安装成功。

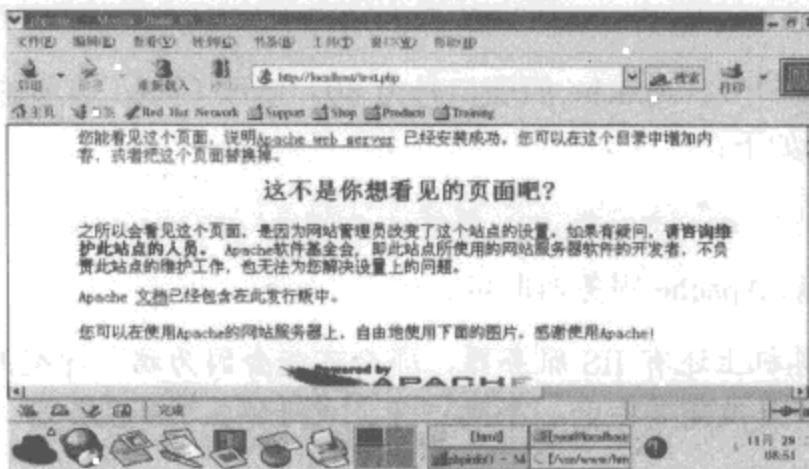


图 2.9 Linux 下的 Apache 服务器安装页面

## 2.3 PHP 的安装和配置

### 2.3.1 在 Windows 下安装 PHP

#### ■ 视频讲解：光盘\TM\Video\2\在 Windows 下安装 PHP.exe

架设基于 PHP 的 Web 服务器，必须安装 PHP。由于 PHP 的代码公开，所以其升级速度较快。安装 PHP 之前应从其官方网站 <http://www.php.net/> 下载最新版本的 PHP 安装程序 `php-5.2.5-Win32.zip`。

Apache 服务器顺利启动后，接下来安装 PHP 5。在 Windows 下安装和配置 PHP 的操作步骤如下：

(1) 将 PHP 5 的安装文件 `php-5.2.5-Win32.zip` 解压到相应目录，如 `c:\php`、`d:\php5` 等。这里将其放到 `d:\php5` 目录下。目录结构如图 2.10 所示。



图 2.10 PHP 5 的目录结构

(2) 将该目录下的所有 dll 文件复制到系统盘 Windows\system32 目录下 (Windows 2000 是在 winnt\system32 目录下)。

(3) 将文件 php.ini-dist 复制到系统盘\Windows 目录下，并重新命名为 php.ini。

(4) 打开 php.ini 文件并找到 “extension\_dir = “./”” 这一行，修改为 “extension\_dir = “d:/php5/ext””。

(5) 找到 “;extension=php\_mysql.dll” 这一行，将前面的分号 “;” 去掉。这样，PHP 即可支持 MySQL 数据库。

(6) PHP 配置完成以后，重新启动 Apache 服务器。

(7) 编写一个 PHP 脚本文件，命名为 phpinfo.php，保存在 Apache 服务器的虚拟目录 D:/htdocs 下。PHP 脚本文件的代码如下：

```
<?php
    phpinfo(); //获取 PHP 的配置信息
?>
```

最后在浏览器的地址栏中输入 “http://localhost/phpinfo.php”，如果显示 PHP 的版本相关信息，则说明 PHP 配置成功。

### 2.3.2 在 Linux 下安装 PHP

#### ■ 视频讲解：光盘\TM\Video\2\Linux 下安装 PHP.exe

安装 PHP 5 之前，需要首先查看 libxml 的版本号。如果 libxml 版本号小于 2.5.10，则需要先安装 libxml 高版本。安装 libxml 和 php 5 的步骤如下（不需要安装 libxml，则直接跳到 php 5 的安装步骤即可）：

(1) 将 libxml 和 php 5 复制到/usr/local/work/目录下，并进入到该目录。

```
mv php-5.2.5.tar.gz libxml2-2.6.26.tar.gz /usr/local/work
cd /usr/local/work
```

(2) 分别将 libxml2 和 php 解压。

```
tar xfz libxml2-2.6.26.tar.gz
tar xfz PHP-5.2.5.tar.gz
```

(3) 进入到 libxml 目录，建立 makfile，将 libxml 安装到 /usr/local/libxml2 下。

```
cd libxml2-2.6.62
./configure --prefix=/usr/local/libxml2
```

(4) 编译文件。

```
makefile
```

(5) 开始安装。

```
make install
```

(6) libxml2 安装完毕，开始安装 php 5。进入到 php-5.2.5 目录下。

```
cd ..\php-5.2.5
```

(7) 建立 makefile。

```
./configure --with-apxs2=/usr/local/Apache2/bin/apxs
--with-mysql=/usr/local/mysql
--with-libxml-dir=/usr/local/libxml2
```

(8) 开始编译。

```
make
```

(9) 开始安装。

```
make install
```

(10) 复制 php.ini-dist 或 php.ini-recommended 到 /usr/local/lib 目录，并命名为 php.ini。

```
cp php.ini-dist /usr/local/lib/php.ini
```

(11) 更改 httpd.conf 文件相关设置，该文件位于 /usr/local/Apache2/conf 中。找到该文件中的如下指令行：

```
AddType application/x-gzip .gz .tgz
```

在该指令后加入如下指令：

```
AddType application/x-httpd-php .php
```

重新启动 Apache，并在 Apache 主目录下建立文件 phpinfo.php。

```
<?php
phpinfo();
?>
```

在 Mozilla 浏览器的地址栏中输入 “http://localhost/phpinfo.php”，如果出现如图 2.11 所示的页面，说明 PHP 安装成功。

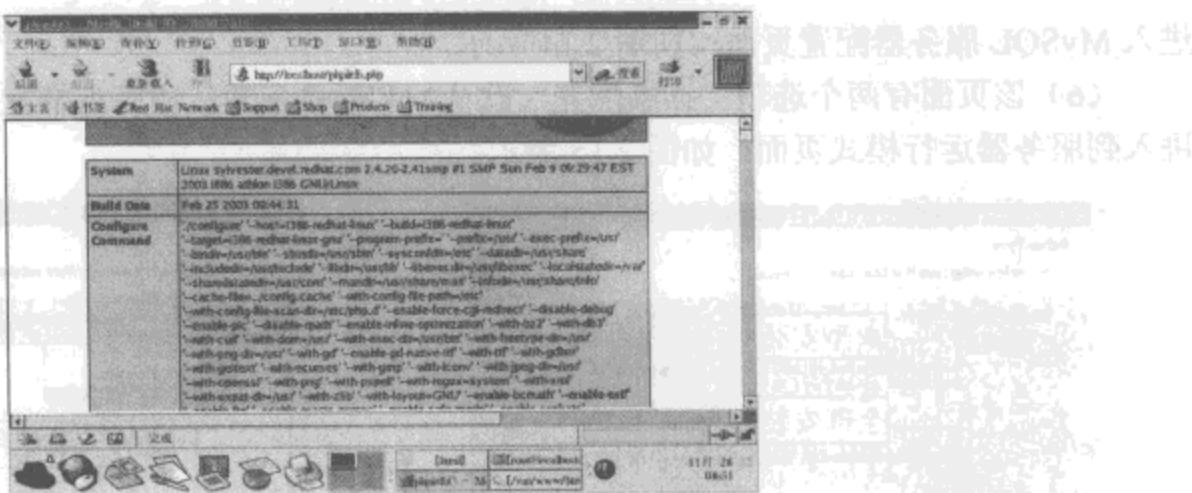


图 2.11 phpinfo 信息

## 2.4 MySQL 服务器的安装和配置

### 2.4.1 在 Windows 下安装 MySQL 服务器

#### ■ 视频讲解：光盘\TM\Video\2\在 Windows 下安装 MySQL 服务器.exe

MySQL 是一款广受欢迎的数据库，由于开源所以市场占有率高，备受 PHP 开发者的青睐，一直被认为是 PHP 的最佳搭档。这是因为 MySQL 不仅是完全网络化的跨平台关系型数据库系统，也是具有客户机/服务器体系结构的分布式数据库管理系统。它具有功能性强、使用简捷、管理方便、运行速度快、版本升级快、安全性高等优点，而且 MySQL 数据库完全免费，从官方网站 <http://www.mysql.com> 即可免费下载到最新版本的 MySQL 安装包 mysql-essential-5.0.51-win32.msi。

**说明：**在学习 MySQL 数据库之前，读者几乎不会接触到数据库知识，也使用不到 MySQL 数据库，因此读者可以暂时不用安装 MySQL 而专心学习 PHP 知识，以免在开始学习 MySQL 数据库时由于时间间隔过长，而忘记当初安装时设置的用户名和密码。

在 Windows 下安装和配置 MySQL 服务器的操作步骤如下。

(1) 双击 MySQL 安装文件 mysql-essential-5.0.51-win32.msi，进入欢迎页面。单击 Next 按钮，进入到 Setup Type 页面。

(2) Setup Type 页面中包含 3 个安装选项，第一项是典型安装，第二项是全部安装。这两个安装的路径不能改变，默认是 E:\Program Files\MySQL\MySQL Server 5.0\（E 盘为系统盘）。第三项是自定义安装，允许用户自定义选择安装组件和安装路径。这里选中 Custom 单选按钮。Setup Type 页面的设置如图 2.12 所示。

(3) 单击 Next 按钮进入到 Custom Setup 页面。选择需要安装的组件，并单击 Change 按钮来选择要安装的目录。Custom Setup 页面的设置如图 2.13 所示。选择完毕后单击 Next 按钮进入到准备安装页面。

(4) 在准备安装页面中显示了用户所选择的安装类型 (type)、路径等信息。如果发现前面的选项设置有误，可以单击 Back 按钮返回到上一个页面重新选择；如果正确，则单击 Install 按钮开始安装文件。

(5) 文件安装完成后，会出现一些关于 MySQL 的功能和版本的介绍。连续单击 Next 按钮，将会

进入 MySQL 服务器配置页面，如图 2.14 所示。

(6) 该页面有两个选项：详细配置（默认）和标准配置。这里保持默认设置。单击 Next 按钮，进入到服务器运行模式页面，如图 2.15 所示。

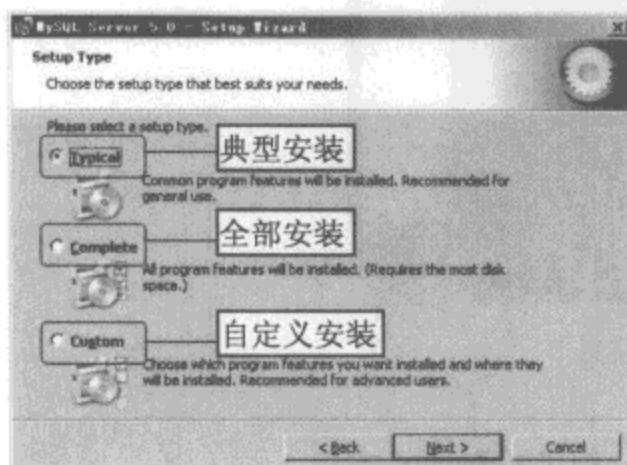


图 2.12 Setup Type 页面

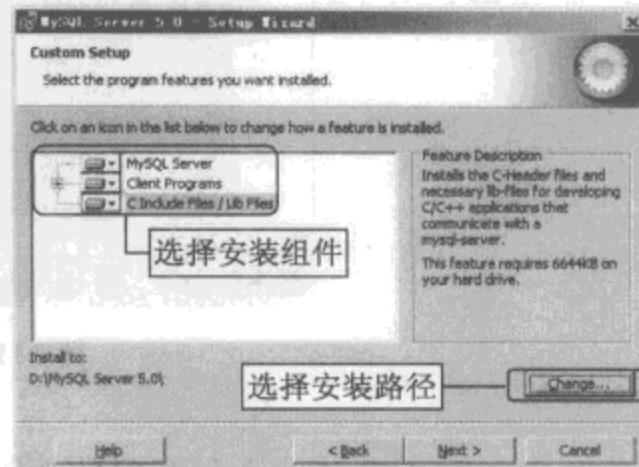


图 2.13 Custom Setup 页面

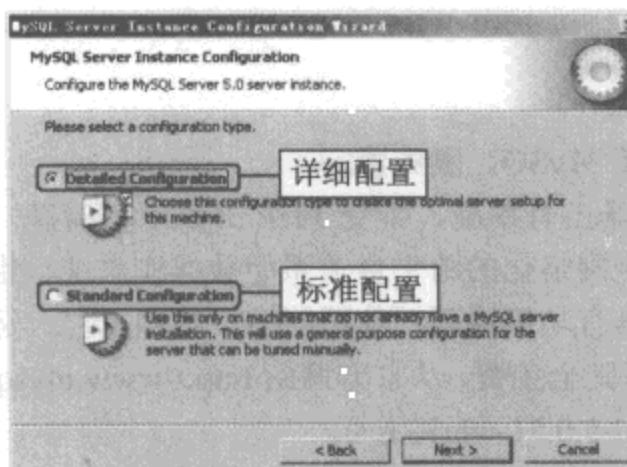


图 2.14 MySQL 服务器配置页面

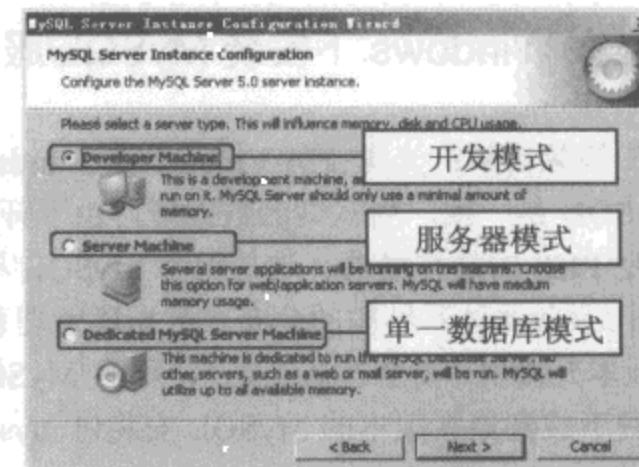


图 2.15 选择服务器运行模式

(7) 该页面中有 3 个选项，这里选择第一个默认项即可（即开发模式，MySQL 服务器占用最小的内存空间，作为本地测试使用完全足够）。选择完毕后，单击 Next 按钮进入选择数据库类型页面。

(8) 本页面有两种数据库类型的选项，第一项是支持 MyISAM、InnoDB 等多种类型库的数据系统；第二项是只支持其中一种类型库。这里选择默认的第一项：Multifunctional Database，支持多种类型库。单击 Next 按钮，如图 2.16 所示。

(9) 进入为 InnoDB 数据文件选择路径页面，这里选择 D 盘下的 MySQL Datafiles 目录。选取分区时要注意所选择分区的剩余空间大小。选择后的页面如图 2.17 所示。单击 Next 按钮。

(10) 进入选择同时连接服务器的最大值的页面，这里可以选择默认的第一项，或者选择第三项自定义连接。第二项的最大连接数为 500。选择后的页面如图 2.18 所示。单击 Next 按钮。

(11) 进入 MySQL 服务器的端口设置页面，默认 3306 即可。选取完毕后单击 Next 按钮。

(12) 进入选择 MySQL 的默认字符集页面。这里选择 GB2312 编码类型。单击 Next 按钮。

(13) 进入选择 MySQL 服务器是否自动运行页面。如果要在 Windows 环境变量 path 中加入 MySQL 执行路径，那么需要选中 Include Bin Directory in Windows PATH 复选框。页面设置如图 2.19 所示。单击 Next 按钮，进入到权限设置页面。

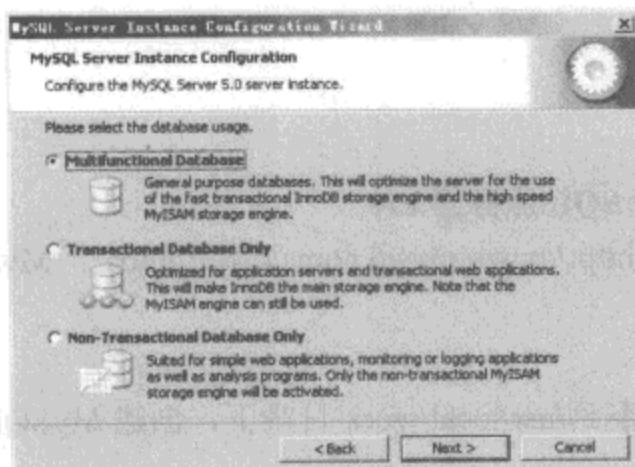


图 2.16 选择数据库类型

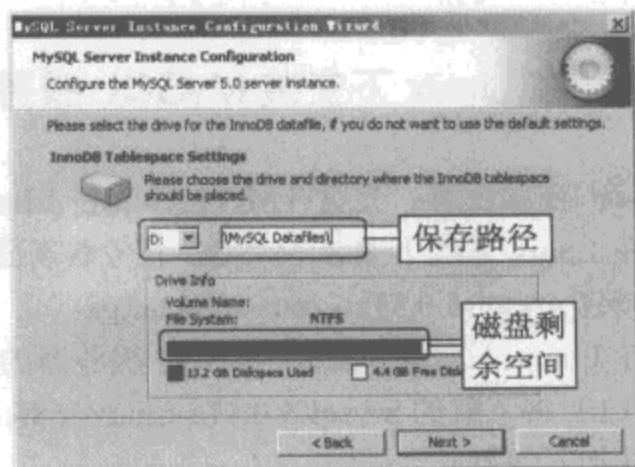


图 2.17 为 InnoDB 数据文件设置路径

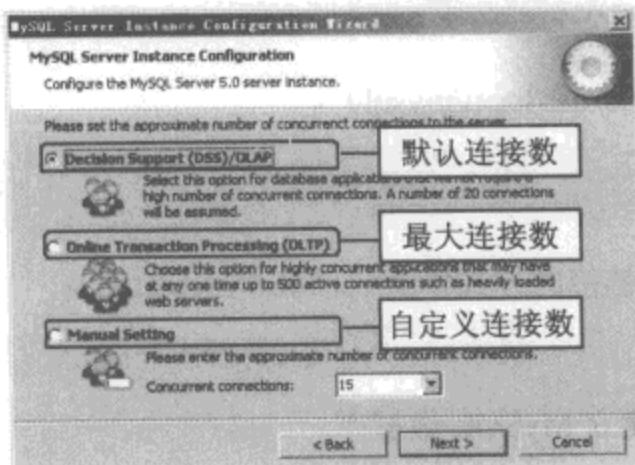


图 2.18 选择同时连接服务器的最大值

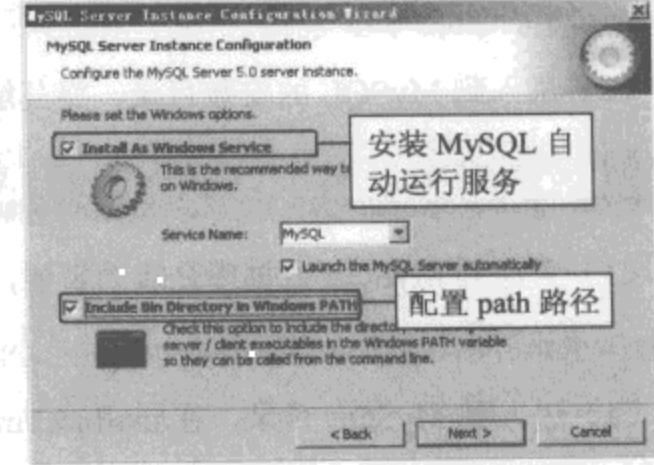


图 2.19 选择 MySQL 服务器的启动方式

(14) 在该页面中可以设置用户登录密码（本书中所有涉及数据库的实例的密码都为 root，所以这里建议也设置为 root，以方便所有 MySQL 数据库实例的运行），在设置密码的下面有一行文本，询问是否允许 root 用户远程登录数据库。如果选中最下面的复选框，则创建一个允许任何人访问数据库的账号，这里不建议选中。页面设置如图 2.20 所示。

(15) 单击 Next 按钮，进入准备执行页面。如果配置没有问题，单击 Execute 按钮开始执行操作，如图 2.21 所示。

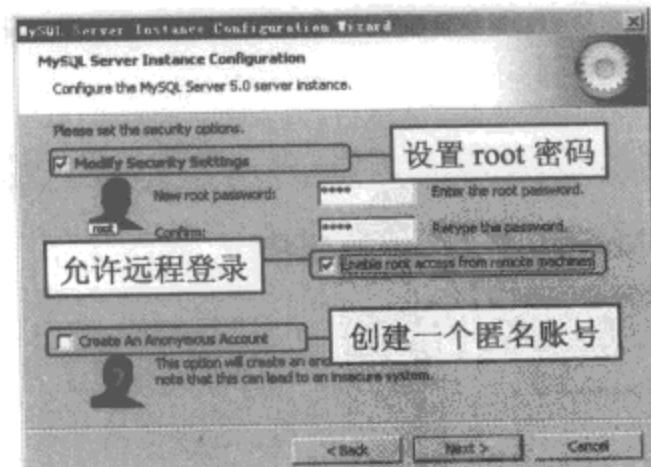


图 2.20 权限设置页面

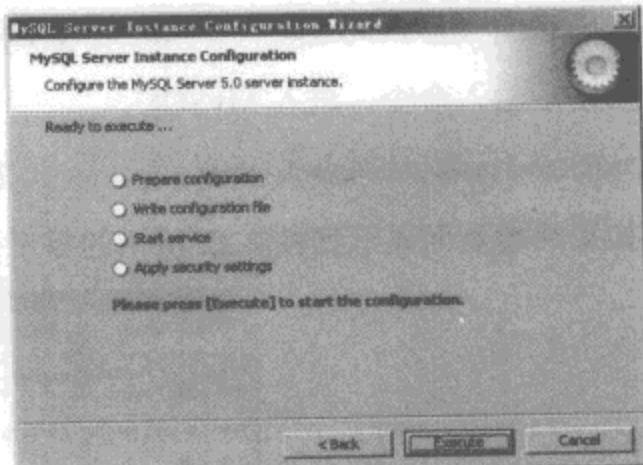


图 2.21 准备执行页面

(16) 安装完成后，单击 Finish 按钮完成 MySQL 服务器的安装。

## 2.4.2 在 Linux 下安装 MySQL 服务器

视频讲解：光盘\TM\Video\2\在Linux下安装MySQL服务器.exe

在 Linux 系统下安装 MySQL 服务器需要到官方网站 <http://www.mysql.com> 下载 Linux 下 MySQL 的安装包 mysql-5.0.51a-Linux-i686.tar.gz。

在 Linux 下安装和配置 MySQL 服务器的操作步骤如下。

(1) 将下载的 mysql-5.0.51a-Linux-i686.tar.gz 文件复制到 /usr/local/work 目录下，创建 MySQL 账号，并加入组群。

```
groupadd mysql  
useradd -g mysql mysql
```

(2) 进入到 MySQL 的安装目录，将其解压（如目录为 /usr/local/mysql）。

```
cd /usr/local/mysql  
tar xfz /usr/local/work/mysql-5.0.51a-Linux-i686.tar.gz
```

(3) 考虑到 MySQL 数据库升级的需要，所以通常以链接的方式建立 /usr/local/mysql 目录。

```
ln -s mysql-5.0.51a-Linux-i686.tar.gz mysql
```

(4) 进入到 MySQL 目录，在 /usr/local/mysql/data 中建立 MySQL 数据库。

```
cd mysql  
scripts/mysql_install_db --user=mysql
```

(5) 修改文件权限。

```
chown -R root  
chown -R mysql data  
chgrp -R mysql
```

(6) 到此 MySQL 安装成功。用户可以通过在终端中输入命令启动 MySQL 服务。

```
/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

启动后输入命令，进入 MySQL。

```
/usr/local/mysql/bin/mysql -uroot
```

如果终端页面显示如图 2.22 所示的提示信息，则说明 MySQL 服务器安装成功。

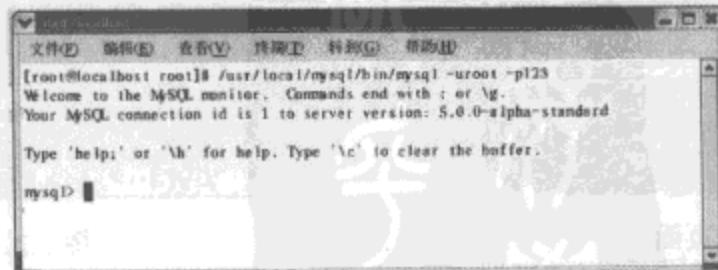


图 2.22 测试 MySQL 是否安装成功

## 2.5 初学者——应用组合包快速搭建 PHP 环境

组合包，就是将 Apache、PHP 和 MySQL 等服务器软件和工具安装配置完成后打包处理。开发人员只要将已配置的套件解压到本地硬盘中即可使用，无须再另行配置。组合包实现了 PHP 开发环境的快速搭建。对于刚刚起步学习 PHP 的程序员，建议采用此方法搭建 PHP 的运行环境。组合包安装简单、速度较快、运行稳定，使用户能将精力更好地集中到 PHP 的学习中。

目前网上流行的组合包有十几种，安装基本都是大同小异。这里推荐 3 种组合包：EasyPHP、AppServ、XAMPP。

建议新手使用 EasyPHP 或 AppServ，两者都是 Apache+MySQL+PHP 开发环境的。而 XAMPP 则相对要复杂一些。不仅可以切换 PHP 4 和 PHP 5，还集成了 perl 开发环境、第三方扩展库等，并且对开发平台进行了优化和整理。如果对 PHP 的开发环境已经有了一定的了解，则推荐使用 XAMPP。

- EasyPHP 下载地址为 <http://www.easyphp.org/>。
- AppServ 下载地址为 <http://www.appservnetwork.com/>。
- XAMPP 下载地址为 <http://www.Apachefriends.org/>。

 注意：要安装这些组合包，必须保证系统中未安装 Apache、PHP 和 MySQL。否则，要先将这些软件卸载后，再开始安装组合包。组合包的安装很简单，只要将程序解压或安装到指定目录后就可以直接使用。

本节以 AppServ 组合包的 AppServ-win32-2.5.7 版本为例，重点讲解该组合包的安装和使用方法。

AppServ 是 PHP 网页架站工具组合包，是将网络上免费的架站资源重新包装成单一的安装程序。它提供了简易、快速的 PHP 运行环境的搭建机制，读者只需按照普通应用软件的安装方式就可以完成 Apache+MySQL+PHP+phpMyAdmin 的安装与配置工作。可以说，AppServ 是初学者的首选。

安装 AppServ 之前应从官方网站 <http://www.appservnetwork.com> 下载 AppServ-win32-2.5.7.exe 安装程序。

### 视频讲解：光盘\TM\Video\2\应用组合包快速搭建 PHP 环境.exe

在 Windows 下应用 AppServ 组合包快速搭建 PHP 开发环境的操作步骤如下。

- (1) 双击 AppServ-win32-2.5.7.exe 文件，打开如图 2.23 所示的 AppServ 启动页面。
- (2) 单击 Next 按钮，打开如图 2.24 所示的 AppServ 安装协议页面。

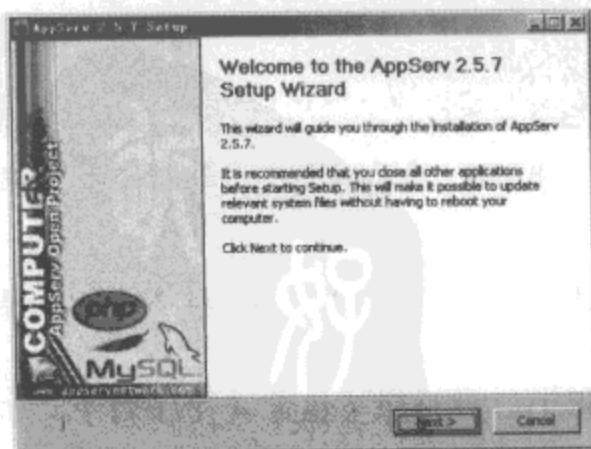


图 2.23 AppServ 启动页面

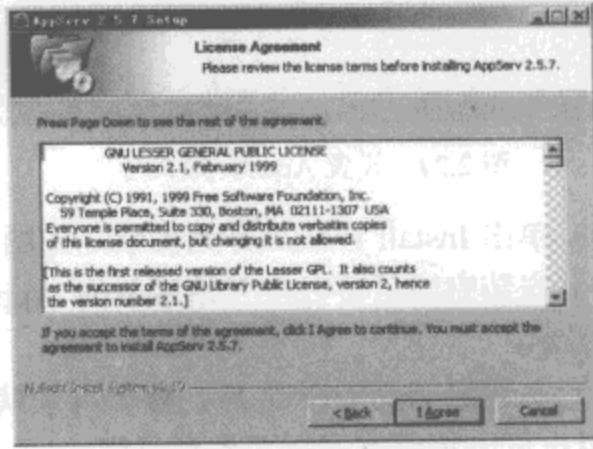


图 2.24 AppServ 安装协议页面

(3) 单击 I Agree 按钮将打开如图 2.25 所示的页面。在该页面中可以设置 AppServ 的安装路径（默认安装路径一般为 C:\AppServ），AppServ 安装完成后 Apache、MySQL、PHP 都将以子目录的形式存储到该目录下。

(4) 单击 Next 按钮将打开如图 2.26 所示的页面。在该页面中可以选择要安装的程序和组件（默认认为全选状态）。

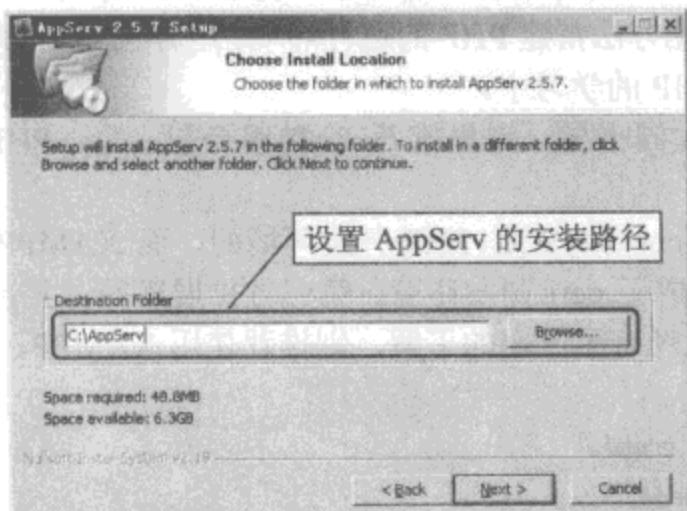


图 2.25 选择 AppServ 安装路径

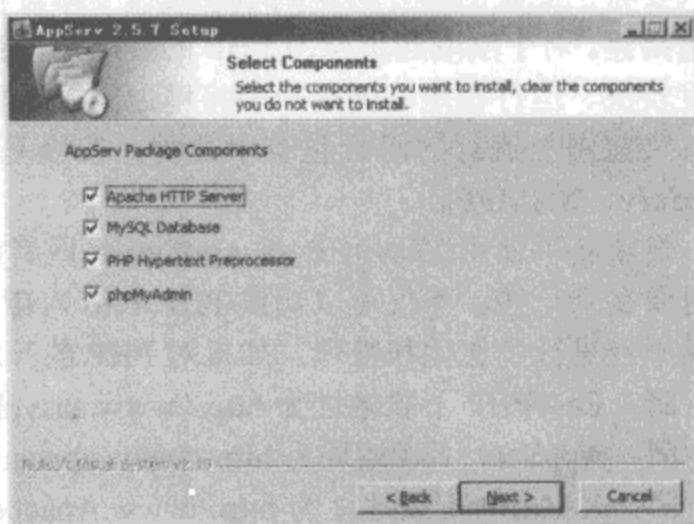


图 2.26 AppServ 安装选项

(5) 单击 Next 按钮，打开如图 2.27 所示的页面。该页面主要用于设置 Apache 的端口号。

(6) 单击 Next 按钮，打开如图 2.28 所示的页面。该页面主要用于对 MySQL 数据库的 root 用户的登录密码及字符集进行设置，这里将字符集设置为 GB2312 Simplified Chinese，表示 MySQL 数据库的字符集将采用简体中文形式。

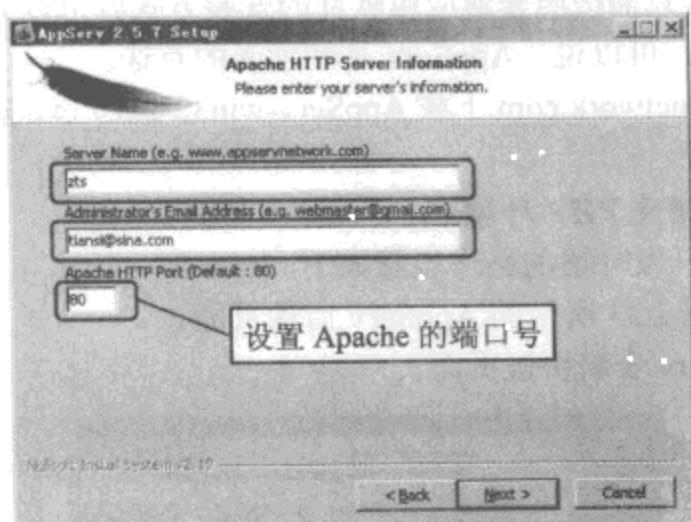


图 2.27 设置 Apache 端口号

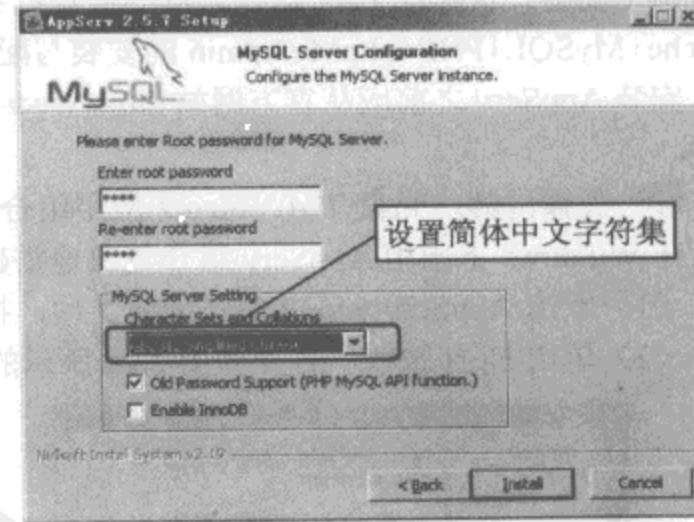


图 2.28 设置 MySQL 登录密码及字符集

(7) 单击 Install 按钮后开始安装，如图 2.29 所示。

(8) 安装完成后可以在开始菜单的 AppServ 相关操作列表中启动 Apache 及 MySQL 服务，如图 2.30 所示。

(9) 安装好 AppServ 之后，整个目录默认为 C:\AppServ，此目录下包含 4 个子目录，如图 2.31 所示，用户可以将所有网页文件存放到 www 目录下。

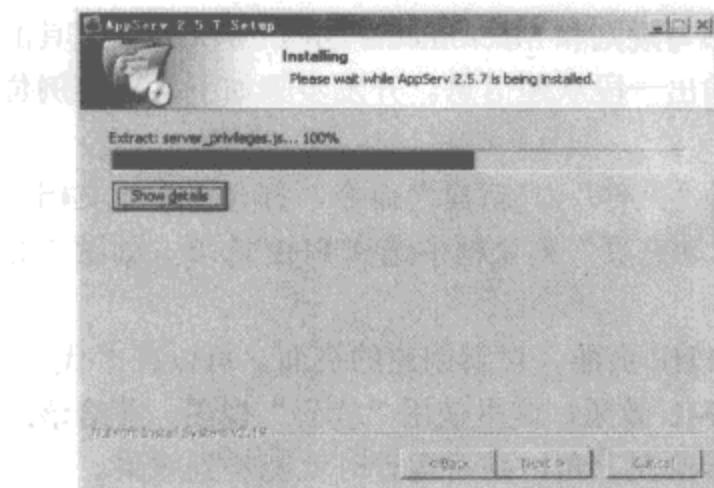


图 2.29 AppServ 安装进度页面

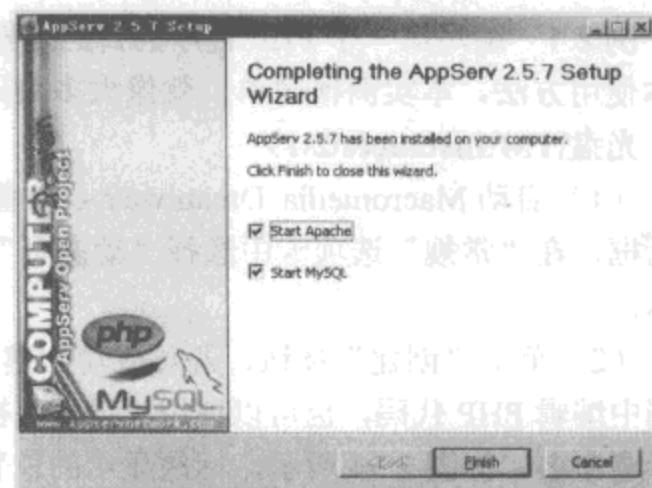


图 2.30 AppServ 安装完成页面

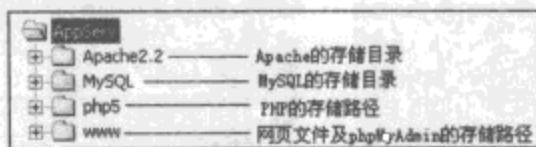


图 2.31 AppServ 目录结构

(10) 打开浏览器，在IE浏览器的地址栏中输入“<http://localhost/>”或者“<http://127.0.0.1/>”，如果打开如图 2.32 所示的页面，则说明 AppServ 安装成功。

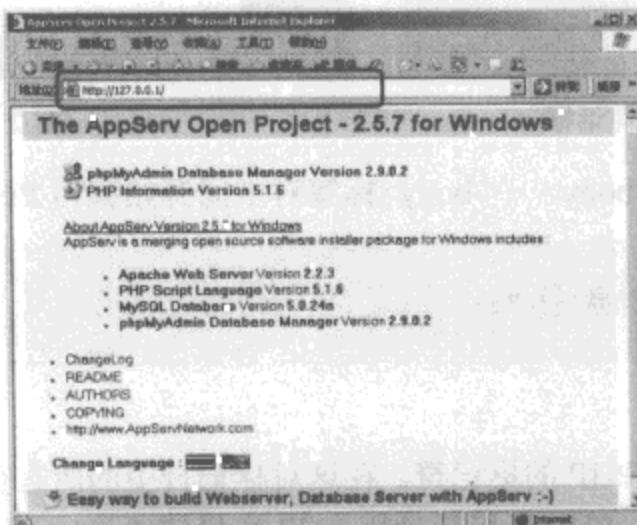


图 2.32 AppServ 测试页

## 2.6 第一个 PHP 程序

视频讲解：光盘\TM\Video\2\第一个 PHP 程序.exe

### 2.6.1 使用 Macromedia Dreamweaver 8 编写源程序

服务器环境配置完成后，接下来应用 Macromedia Dreamweaver 8 开发工具来编写第一个 PHP 程序。

**例 2.1 编写第一个 PHP 程序的目的是熟悉 PHP 的书写规则和 Macromedia Dreamweaver 8 工具的基本使用方法。本实例很简单，就像大多数语言一样，输出一段欢迎信息。开发步骤如下：**（**实例位置：光盘\TM\Instances\2\1**）

(1) 启动 Macromedia Dreamweaver 8 编辑器，选择“文件”/“新建”命令，打开“新建文档”对话框，在“常规”选项卡中选择“动态页”类别，在“动态页”列表框中选择 PHP 选项，如图 2.33 所示。

(2) 单击“创建”按钮，即可成功创建一个动态的 PHP 页面。对新创建的页面，可以在“代码”视图中编辑 PHP 代码，也可以使用“设计”视图查看 HTML 效果。这里使用“代码”视图，并给该页面标题命名。如图 2.34 所示，标题显示的位置是在浏览器的左上角，稍后运行时就能看到效果。

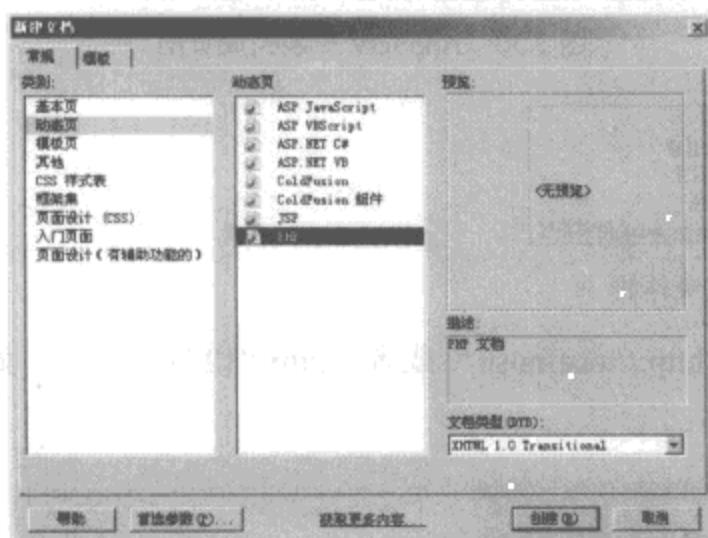


图 2.33 “新建文档”对话框

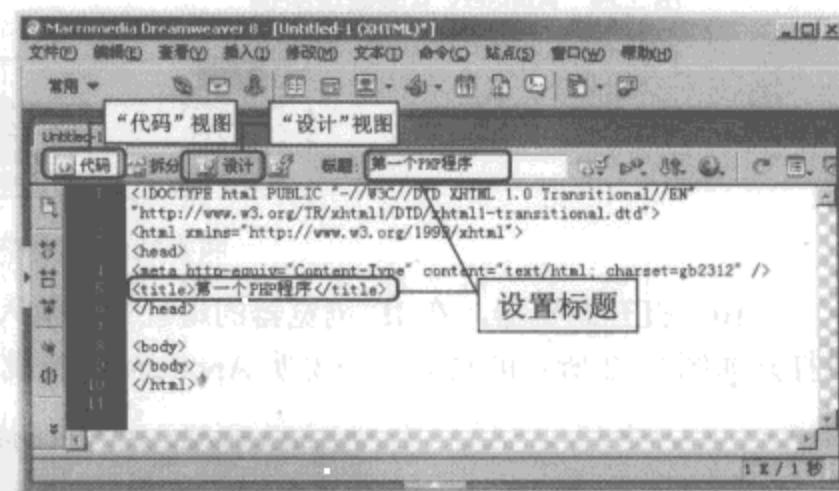


图 2.34 命名标题

(3) 编写 PHP 代码。在<body>…</body>标签对中间开始编写 PHP 代码段，代码如下。

```
<?php
echo "欢迎进入 PHP 的世界！！";
?>
```

代码讲解：

- “<?php” 和 “?>” 是 PHP 的标记符。在这对标记符中的所有代码都被当作 PHP 代码来处理。除了这种表示方法外，PHP 还可以使用 ASP 风格的 “<%” 和 SGML 风格的 “<?...?>” 等，在第 3 章中将会详细介绍。
- echo 语句是 PHP 中最常用的语句，主要用于将一个或多个字符串输出至网页，和 ASP 中的 response.write 以及 JSP 中的 out.print 是一个意思，就是将紧跟在后面的字符串或者变量值显示在页面中。每行代码都以分号 “;” 结尾。

**◆ 注意：** 在使用 echo 语句时，应该时刻牢记语句结束处的分号。在 PHP 中，分号是用来分隔语句的。丢失这个 “;” 是读者经常会犯的语法错误，同样也是读者在进行程序的错误处理时，首先应该注意的地方。

在 Macromedia Dreamweaver 8 中输入的 PHP 脚本程序如图 2.35 所示。

(4) 将 PHP 动态页保存到服务器指定的目录以便解析。本章中服务器指定的目录为 D:\appserv\

www\，则将本页保存到 D:\appserv\www\tm\sl\2\1\路径下，命名为 index.php。



图 2.35 在开发工具中输入 PHP 脚本程序

## 2.6.2 发布和运行 PHP 程序

打开 IE 浏览器页面，在地址栏中输入“<http://127.0.0.1/tm/sl/2/1/index.php>”，按 Enter 键后即可查看 index.php 页的执行结果，如图 2.36 所示。

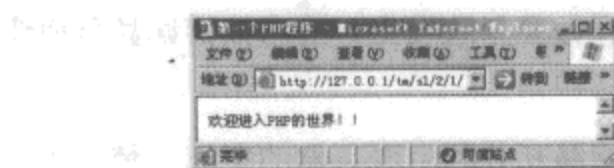


图 2.36 PHP 页面运行结果

## 2.7 环境安装常见问题

### 2.7.1 Apache 安装常见问题

#### 1. 解决 Apache 服务器端口冲突

IIS 服务器、迅雷的默认端口号为 80，同 Apache 服务器默认端口号相同。两者由于采用了相同的端口号 80，因此，在运行网页时就会发生冲突。

如果用户安装了 IIS 服务器，就需要修改 IIS 的默认端口，否则将导致 Apache 服务器无法正常工作。更改 IIS 的默认侦听端口 80，可以在 IIS 的管理器中进行设置，也可以停止 IIS 的服务。

如果用户安装并开启了迅雷软件，就需要关闭该软件，否则端口冲突将会导致运行 PHP 网页程序时出错。

用户也可以在安装 Apache 服务器时将默认的端口号进行更改，从而解决两个服务器或与其他软件共用一个端口号而产生冲突的问题。

#### 2. 更改 Apache 服务器默认存储的文件路径

Apache 服务器的核心配置文件是 httpd.conf，存放路径为“Apache 的安装路径\conf”，用记事本

程序打开该文件，定位到 DocumentRoot，语句如下。

```
DocumentRoot "D:/Webpage"
```

这个语句用于指定网站路径，也就是主页放置的目录。可以使用默认路径，也可以任意指定。需要注意的是，语句的末尾不要加“/”。

同时还要定位到“<Directory "">”一行，在双引号中添加服务器的虚拟路径，这里要与“DocumentRoot”一行中设置相同。

```
<Directory "D:/Webpage">
```

注意：路径的分隔符在 Apache 服务器里写成“/”。

## 2.7.2 PHP 安装常见问题

### 1. PHP 的安装路径

安装文件的路径也要遵循一定的客观原则，为了避免在 Windows 和 Linux 间移植程序时带来的不便，选择 D:\usr\local\php 的目录时要和在 Linux 下的安装目录相匹配。建议最好不要选择中间有空格的目录，如 E:\program Files\PHP，这样做会导致发生一些未知错误甚至崩溃。

### 2. 控制上传文件的大小

在网站开发的过程中，为了确保能够充分利用服务器的空间，禁止上传一些垃圾文件，给网站的维护带来不必要的麻烦，最好对上传文件的大小进行限制，将它控制在有效上传文件大小的范围之内。如果要在 PHP 中实现小文件的上传（2MB 以下），那么无须对 php.ini 配置文件进行修改，使用默认参数即可。但如果想实现完美的上传功能，则一定要对 php.ini 进行一些修改。

Resource Limits，直译就是“资源限制”，包含 3 个参数。该区块不仅是针对上传下载的，还是对全部的文件进行设置。各个参数含义及参数值说明如表 2.1 所示。

表 2.1 Resource Limits 块的参数说明

参 数	说 明
max execution time	每个脚本页面完成执行操作的最大时间，单位是秒。如果设为-1，说明没有限制
max input time	每个脚本页面处理请求数据的最大时间，单位是秒，也可以设为-1
memory limit	一个脚本页所能够消耗的最大内存

post\_max\_size 参数指 PHP 通过表单 POST 所能接收的最大值，包括表单中所有的项。

File Uploads 块是专为文件上传设置的，包含 3 个参数，参数含义及参数值说明如表 2.2 所示。

表 2.2 File Uploads 块的参数说明

参 数	说 明
file_uploads	是否允许 HTTP 上传，默认为 On，即为开启，无须修改
upload tmp dir	文件上传时的临时存储目录。如果没指定就会用系统默认的临时文件夹
upload max filesize	允许上传的文件的最大值

如果想要上传更大的文件，就必须对上述3个区块的参数值进行更改，更改后重新启动Apache服务器即可。

**注意：**如果上传文件超过php.ini文件中设置的值，文件将上传失败。

### 2.7.3 MySQL安装常见问题

在网站运作的过程中，各类错误均不可避免，当数据库连接失败时，除开启MySQL服务检测是否正常运行外，读者还可以检查php.ini文件是否配置正确，以支持MySQL服务。

打开C:\Windows\目录下的php.ini文件，定位到如图2.37所示的代码位置。

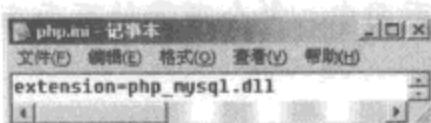


图2.37 修改php.ini文件以支持MySQL数据库

将代码前面的分号删除，然后保存php.ini文件，最后重新启动Apache服务器，即可让PHP支持MySQL数据库。

读者可以运行“<http://127.0.0.1/phpinfo.php>”或“<http://localhost/phpinfo.php>”网址，如果检索到MySQL服务，如图2.38所示，则说明MySQL服务正常运行。

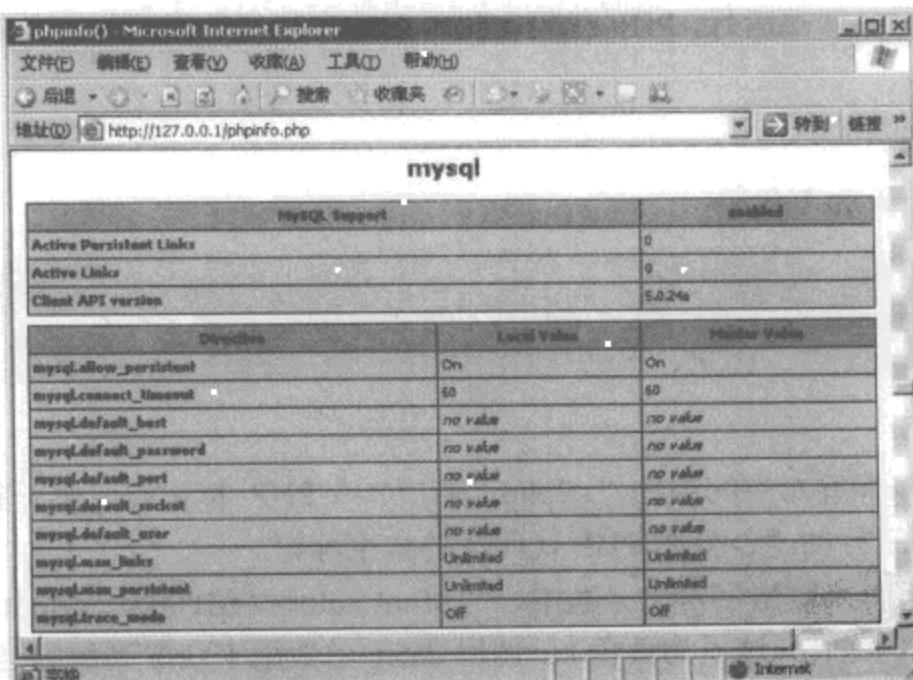


图2.38 测试MySQL服务是否正常运行

## 2.8 实战

**视频讲解：**光盘\TM\Video\2\调用PHP的date()函数动态显示系统时间.exe

**例2.2 应用Macromedia Dreamweaver 8开发工具，通过调用PHP的date()函数，动态显示系统时**

间。(实例位置: 光盘\TM\Instances\2\2)

### 1. 创建站点

编程的第一步就是创建 Web 站点, 在 www 服务器的根目录下创建指定的“TM/sl/2/2”文件夹, 用来存储 PHP 文件和相关资源。

### 2. 使用 Macromedia Dreamweaver 8 创建 PHP 文件

启动 Macromedia Dreamweaver 8 编辑器, 选择“文件”/“新建”命令, 打开“新建文档”对话框, 选择“常规”选项卡中的“动态页”/“PHP”命令, 然后单击“创建”按钮, 即可成功创建一个动态的 PHP 页面。

说明: 将切割的图像嵌入到 PHP 页面中, 并合理设计页面布局。

### 3. 创建 PHP 标记

在<body>标记中使用“<?php ?>”标记符将 PHP 的脚本语言括起来, 而这些 PHP 脚本在浏览器中并不可见, 当用户在 IE 浏览器中输入 URL 地址“http://localhost/PHP Web/index.php”, 服务器端接到请求后, 会使用 PHP 的默认语言解释成标准的 HTML 格式的代码传送到用户的浏览器中。

技巧: 解释生成的 HTML 代码, 用户可以在 IE 浏览器中选择“查看”/“源文件”命令, 打开程序的源代码, 是标准的 HTML 文件。

因为 PHP 代码在服务器端运行, PHP 代码在执行后才产生浏览器能识别的 HTML 语言, 并将其传递给浏览器。

### 4. 编写 PHP 代码

在<body>标记中输入如下代码。

```
<?php
    echo "<font color=#ffff00>北京时间: ".date("Y-m-d H:i:s")."</font>"?
?
```

代码解释:

在<body>…</body>标签对中间添加“<?php”和“?>”PHP 标记符。第一行的“<?php”是 PHP 代码段的开始标记。第三行的“?>”是 PHP 代码段的结束标记。

第二行的 echo 语句用来输出字符串。其中, “<font color=#ffff00>…</font>”代表将包含在该标记间的字符串设置指定的颜色(这里为黄色)。调用 date() 函数用来获取系统的当前时间, 其中“Y-m-d H:i:s”是以指定的格式输出系统的当前日期时间, 并用字符串操作符“.”将多个字符串连接在一起进行输出。

注意: 由于在 PHP 语言中默认设置的是标准的格林威治时间(即采用的是零时区), 所以要获取本地当前的时间必须更改 PHP 语言中的时区设置。更改 PHP 语言中的时区设置有两种方法:  
(1) 修改 php.ini 文件中的设置, 找到[date]下的“;date.timezone =”选项, 将该项修改为“date.timezone =Asia/Hong\_Kong”, 然后重新启动 Apache 服务器。(2) 在程序中, 使用时间/日期函数之前添加函数“date\_default\_timezone\_set("Asia/Hong\_Kong");”即可。

在 Macromedia Dreamweaver 8 中输入 PHP 脚本程序的运行结果如图 2.39 所示。

### 5. 存储文件

在创建好 PHP 文件后，选择“文件”菜单中的“保存”命令，在弹出的“另存为”对话框中，选择已创建的 TM/sl/2/2 文件夹作为 PHP 文件的保存位置，将文件命名为 index.php，然后单击“保存”按钮，如图 2.40 所示。

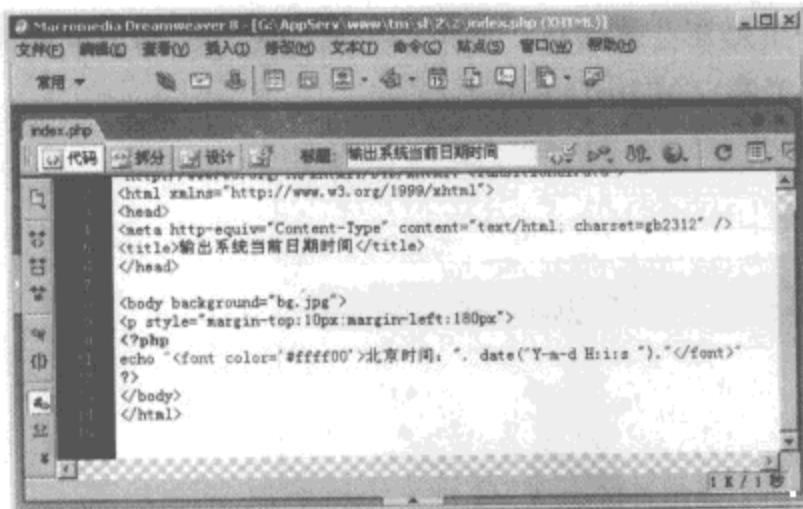


图 2.39 在开发工具中输入 PHP 脚本程序



图 2.40 “另存为”对话框

### 6. 网站运行结果

打开 IE 浏览器，在地址栏中输入 URL 地址“<http://127.0.0.1/tm/sl/2/2/index.php>”，按 Enter 键打开该页面，运行结果如图 2.41 所示。

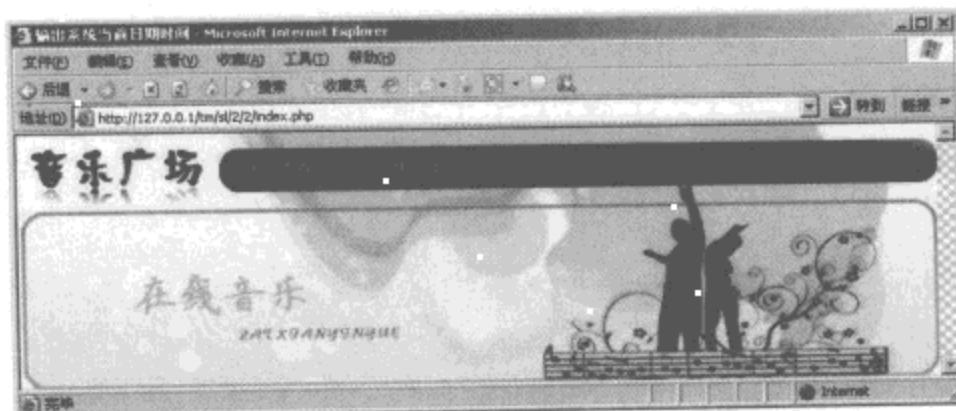


图 2.41 打开网站页面

## 2.9 本章小结

本章主要介绍了 PHP 环境的安装与配置方法。其中，读者要对 PHP 环境的安装与配置重点掌握。通过本章的学习，读者可以轻松解决环境配置的常见问题，并可以独立开发、发布和运行简单的 PHP 程序。对于本章中讲解的在 Windows 和 Linux 系统上 PHP 环境的搭建方法，读者可以根据自身的需求有选择性地阅读。

## 2.10 实战练习

- 尝试开发一个页面，应用 echo 语句输出字符串“我相信通过努力我一定会精通 PHP！”。（答案位置：光盘\TM\Instances\2\3）
  - 尝试开发一个页面，应用 echo 语句输出一条最新公告信息，要求公告后面添加发布日期和时间，输出效果如图 2.42 所示的页面。（答案位置：光盘\TM\Instances\2\4）

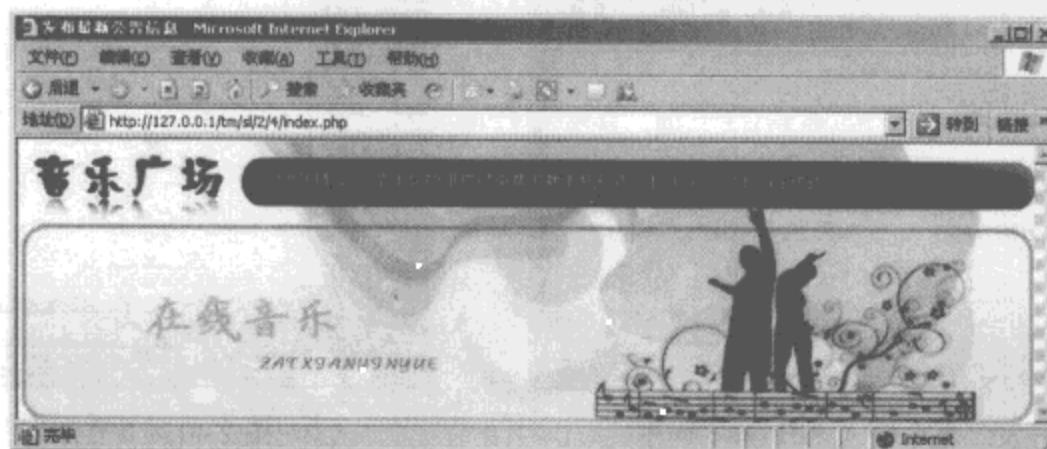


图 2.42 发布最新公告信息

# 第3章 PHP 语言基础

## PHP 语言基础

( 视频讲解: 125分钟)

学习一门语言，首先就要学习这门语言的语法，PHP 也不例外。本章开始学习 PHP 的基础知识，这也是 PHP 的核心内容。不论从事网站制作，还是应用程序开发，没有扎实的基本功是行不通的。开发一个功能模块，如果一边查函数手册一边写程序代码，要用 15 天；但基础好的人只要 3~5 天，甚至更少的时间，孰优孰劣，立见分晓。为了将来应用 PHP 程序开发 Web 站点时“节省”时间，现在就要认真从基础学起，牢牢掌握 PHP 的基础知识。只有做到这一点，才能在以后的开发过程中事半功倍。

通过阅读本章，可以了解到：

- PHP 的语法基础
- PHP 的数据类型
- 如何熟练使用 PHP 的常量、变量
- PHP 各类运算符的应用
- PHP 表达式的应用

### 3.1 学习语言基础的重要性

假如你是一个 PHP 初学者，当你向一位有经验的 PHP 高级程序员请教如何学好 PHP 这门语言的时候，想必他会问你有没有其他编程语言基础，如果说有的话，就能够很快入门，学起来也很轻松。

由此可见，掌握一门语言的基础是多么的重要。可以说，扎实的语言基础是培养和提高程序员编程能力的基本途径。语言基础也是整个 PHP 学习过程中非常重要的一个环节，它在很大程度上决定了程序员整个 PHP 学习的进展和效果。

另外，程序的创造性极强，可以完全按照程序员自己的设计目标和思路组织代码，以及与其他人完全不同的设计方法实现完全相同的功能。但是，编程代码同时又是一项十分严谨的工作，因为语法是严格限定的，代码必须严格遵守语法要求进行书写，任何不符合规范的代码都不能被计算机执行。哪怕是程序员写错一个字母，或者少写一个结束符都会使程序出错。

所以，程序设计语言的语法必须得到必要的重视。语法是 PHP 语言的规则，打好这个基础，循序渐进地学习，实际运用时才会得心应手。

## 3.2 PHP 语法基础

### 3.2.1 PHP 标记符

**视频讲解：**光盘\TM\Video\3\PHP 标记符.exe

PHP 标记符能够让 Web 服务器识别 PHP 代码的开始和结束，两个标记之间的所有文本都会被解释为 PHP，而标记之外的任何文本都会被认为是普通的 HTML，这就是 PHP 标记的作用。PHP 标记符风格迥异，按照风格的不同可划分为以下 4 种。

#### (1) 标准风格

```
<?php
    echo "标准风格的 PHP 标记";
?>
```

这是本书中使用的标记风格，也是推荐读者使用的标记风格。

#### (2) 脚本风格

```
<script language="php">
    echo '这是脚本风格的标记';
</script>
```

在 XHTML 或者 XML 中推荐使用这种标记风格，它符合 XML 语言规范的写法。

#### (3) 简短风格

```
<?
    echo "简短风格的标记";
?>
```

这种标记风格最为简单，输入字符最少，但想要使用它，必须要更改配置文件 `php.ini`。这种标记风格不推荐使用。

#### (4) ASP 风格

```
<%  
echo "ASP 风格的标记";  
%>
```

这种标记风格和 ASP 相同，不推荐使用。

 **注意：**如果使用简短风格 “`<? ?>`” 和 ASP 风格 “`<% %>`”，需要分别在配置文件 `php.ini` 中做如下设置。打开系统盘 Windows（以 Windows 2003 Server 操作系统为例）文件夹下的 `php.ini` 文件，将如下代码段中的“OFF”改为“ON”，更改后的代码如下。

```
short_open_tag = On  
asp_tags = On
```

保存修改后的 `php.ini` 文件，然后重新启动 Apache 服务器，即可支持这两种标记风格。

### 3.2.2 PHP 的注释

#### 视频讲解：光盘\TM\Video\3\PHP 的注释.exe

注释，即代码的解释和说明，一般添加到代码的上方或代码的尾部（添加到代码的尾部时，代码和注释之间加适当数量的`<Tab>`键，以方便程序阅读），用来说明代码或函数的编写人、用途、时间等。注释不会影响到程序的执行，因为在执行时，注释部分会被解释器忽略。

PHP 支持 3 种风格的程序注释。

//单行注释

```
<?php  
    echo 'PHP 编程词典'; //输出字符串（但单行标记后的注释内容不被输出）  
?>
```

/\*...\*/ 多行注释

```
<?php  
/*多行  
注释内容  
不被输出  
*/  
echo '只会看到这句话。';  
?>
```

 **注意：**多行注释是不允许进行嵌套操作的。

**Shell 风格的注释 #**

```
<?php
    echo '这是 Shell 脚本风格的注释';      #这里的內容是看不到的
?>
```

**注意：**在单行注释里的内容不要出现“?>”的标志，因为解释器会认为 PHP 脚本结束，而去执行“?>”后面的代码。例如：

```
<?php
    echo '这样会出错的!!!!!!'      #不会看到?>会看到
?>
```

结果为：这样会出错的!!!!!! 会看到。?>

### 3.3 PHP 的数据类型

**视频讲解：光盘\TM\Video\3\PHP 数据类型.exe**

数据类型的出现是计算机语言发展中最重要的一步。在计算机的世界里，计算机操作的对象是数据，而每一个数据都有其类型，具备相同类型的数据才可以彼此操作。PHP 的常用数据类型可以分成整型、布尔型、浮点型、字符串型和 null 型等。下面进行详细讲解。

#### 3.3.1 整型

整型数据类型只能包含整数。在 32 位的操作系统中，有效的范围是-2147483648~+2147483647。整型数可以用八进制、十进制和十六进制来表示。如果用八进制，数字前面必须加 0；如果用十六进制，则需要加 0x。

**注意：**如果在八进制中出现了非法数字（8 和 9），则后面的数字会被忽略掉。

**例 3.1** 在本例中，分别输出八进制、十进制和十六进制的结果，实例代码如下：（实例位置：光盘\TM\Instances\3\1）

```
<?php
$str1 = 1234;
$str3 = 0x1234;
$str2 = 01234;
echo "数字 1234 不同进制的输出结果: <p>";
echo "八进制的结果是: $str2<br>";
echo "十进制的结果是: $str1<br>";
echo "十六进制的结果是: $str3";
?>
```

运行结果如图 3.1 所示。

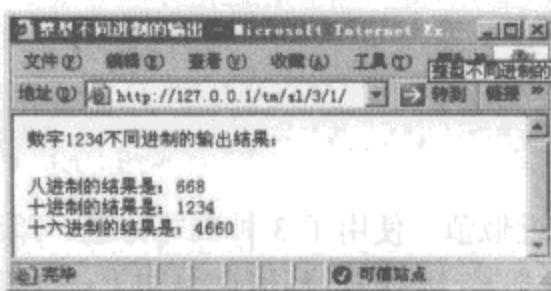


图 3.1 不同进制的输出结果

**◆ 注意：**如果给定的数值超出了 int 所能表示的最大范围，将会被当作 float 型处理，这种情况叫做整数溢出。同样，如果表达式的最后运算结果超出了 int 的范围，也会返回 float 型。

### 3.3.2 布尔型

布尔型是 PHP 中较为常用的数据类型之一。它保存一个 true 或者 false 值，其中 true 和 false 是 PHP 的内部关键字。设定一个布尔型的变量，只需将 true 或者 false 赋值给变量即可。

**例 3.2** 通常布尔型变量都是应用在条件或循环语句的表达式中。下面在 if 条件语句中判断变量 \$boo 中的值是否为 true，如果为 true，则输出“变量\$boo 为真！”，否则输出“变量\$boo 为假！！”，代码如下：（实例位置：光盘\TM\Instances\3\2）

```
<?php
$boo = true; //声明一个 boolean 类型变量，初赋值为 true
if($boo == true) //判断变量$boo 是否为真
    echo '变量$boo 为真!'; //如果变量$boo 的值为真，输出“变量$boo 为真！”
else
    echo '变量$boo 为假!!'; //如果变量$boo 的值为假，则输出“变量$boo 为假！！”
?>
```

结果为：变量\$boo 为真！

**◆ 注意：**在 PHP 中不是只有 false 值才为假的，在一些特殊情况下 boolean 值也被认为是 false。这些特殊情况为：0、0.0、“0”、空白字符串（""）、只声明没有赋值的数组等。

**说明：**美元符号\$是变量的标识符，所有变量都是以\$符开头的，无论是声明变量还是调用变量，\$符都不应遗漏。

### 3.3.3 浮点型

浮点数据类型可用来存储数字，也可用来保存小数。它提供的精度比整型大得多。在 32 位的操作系统中，有效的范围是 1.7E-308~1.7E+308。在 PHP 4.0 以前的版本中，浮点型的标识为 double，也叫双精度浮点数，两者没什么区别。

浮点型数据默认有两种书写格式，一种是标准格式：

3.1415  
-35.8

还有一种是科学记数法格式：

```
3.58E1  
849.72E-3
```

**例 3.3** 本例中输出圆周率的近似值。使用了 3 种书写方法，即圆周率函数、传统书写格式和科学记数法，最后显示在页面上的效果都是一样的，程序代码如下：（实例位置：光盘\TM\Instances\3\3）

```
<?php  
echo '圆周率的 3 种书写方法: <p>';  
echo '圆周率函数: pi() = '. pi() . '<p>'; //调用 pi 函数输出圆周率  
echo '传统书写格式: 3.14159265359 = '. 3.14159265359 . '<p>'; //传统书写格式的浮点数  
echo '科学记数法: 314159265359E-11 = '. 314159265359E-11 . '<p>'; //科学记数法格式的浮点数  
?>
```

运行结果如图 3.2 所示。

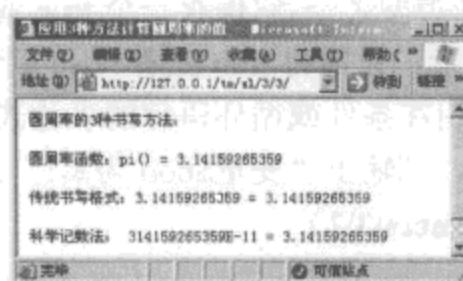


图 3.2 应用 3 种方法计算圆周率的值

**◆ 注意：**浮点型的数值只是一个近似值，所以要尽量避免浮点型之间比较大小，因为最后的结果往往是不准确的。

### 3.3.4 字符串型

字符串是连续的字符序列，由数字、字母和符号组成。字符串中的每个字符只占用一个字节。在 PHP 中，定义字符串有 3 种方式，即单引号 (')、双引号 ("") 和界定符 (<<<)。

#### 1. 单引号

单引号的定义格式如下：

```
<?php  
$str = '字符串';  
?>
```

#### 2. 双引号

双引号的定义格式如下：

```
<?php  
$str="字符串";  
?>
```

**◆ 注意：**两者的不同之处在于，双引号中所包含的变量会自动被替换成实际数值，而在单引号中包含的变量则按普通字符串输出。

**例 3.4** 下面的实例分别应用单引号和双引号来输出同一个变量，其输出的结果完全不同，双引号输出的是变量的值，而单引号输出的是字符串“\$str”。实例代码如下：(实例位置：光盘\TM\Instances\3\4)

```
<?php
    $str= '我是小小字符串'; //声明一个字符串变量
    echo "$str"; //用双引号输出的是变量的值
    echo "<p>";
    echo '$str'; //用单引号输出的是字符串
?

```

运行结果如图 3.3 所示。

两者之间另一不同点是对转义字符的使用。使用单引号(')时，只要对单引号行转义就可以了，但使用双引号("")时，还要注意“\"”、“\$”等字符的使用。这些特殊字符都要通过转义符“\”来显示。常用的转义字符如表 3.1 所示。

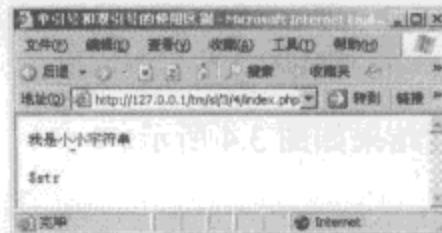


图 3.3 单引号和双引号的使用区别

表 3.1 转义字符

转义字符	输出
\n	换行 (LF 或 ASCII 字符 0x0A (10))
\r	回车 (CR 或 ASCII 字符 0x0D (13))
\t	水平制表符 (HT 或 ASCII 字符 0x09 (9))
\\"	反斜杠
\\$	美元符号
\'	单引号
\"	双引号
\[0-7]{1,3}	此正则表达式序列匹配一个用八进制符号表示的字符，如\467
\x[0-9A-Fa-f]{1,2}	此正则表达式序列匹配一个用十六进制符号表示的字符，如\x9f

\n 和\r 在 Windows 系统中没有什么区别，都可以当作回车符。但在 Linux 系统中则是两种效果，在 Linux 中，\n 表示换到下一行，却不会回到行首；而\r 表示光标会回到行首，但仍然在本行。如果读者使用 Linux 操作系统，可以尝试一下这两个转义字符的应用。

**◆ 注意：**如果对非转移字符使用了 \"，那么在输出时，“\"”也会跟着一起被输出。

**说明：**在定义简单的字符串时，使用单引号是一个更加合适的处理方式。如果使用双引号，PHP 将花费一些时间来处理字符串的转义和变量的解析。因此，建议在定义字符串时，如果没有特别的要求，应尽量使用单引号。

### 3. 界定符

界定符(<<<)是从 PHP 4.0 开始支持的。在使用时后面接一个标识符，然后是字符串，最后以同样的标识符结束字符串。

界定符的格式如下：

```
$string = <<< str
要输出的字符串
str
```

其中 str 为指定的标识符。

**例 3.5** 下面应用界定符输出变量中的值，可以看到，它和双引号没什么区别，包含的变量也被替换成实际数值，实例代码如下：（实例位置：光盘\TM\Instances\3\5）

```
<?php
$str="学习 PHP 语言要永不言弃！"; //定义变量$str
echo <<<str
    这和双引号没有什么区别，\$str 同样可以被输出出来。<p>
    \$str 的内容为：$str
str;
?>
```

运行结果如图 3.4 所示。

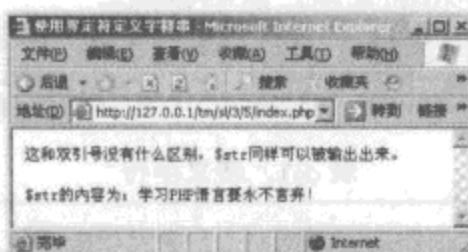


图 3.4 使用界定符定义字符串

**注意：**结束标识符必须单独另起一行，并且不允许有空格。在标识符前后有其他符号或字符时，也会发生错误。在界定符开始标记和结束标记中间不允许添加注释，否则在运行程序时，会弹出如“Parse error: parse error, unexpected T\_SL in G:\AppServ\www\tm\sl\3\5\index.php on line 10”的错误提示。

### 3.3.5 null 类型

关键字 null（空）实际上表示一个变量没有被赋值。空值（null）不区分大小写，null 和 NULL 效果是一样的。被赋予空值的情况有以下 3 种。

- 没有赋任何值。
- 被赋予 null 值。
- 被 unset() 函数处理过的变量。

判断变量是否为空值，可以应用 is\_null() 函数进行检测。

**例 3.6** 下面来看一个具体实例。字符串 str1 被赋值为 null，str2 根本没有声明和赋值，所以也输出 null，最后的 str3 虽然被赋予了初值，但被 unset() 函数处理后，也变为 null 型。unset() 函数的作用就是从内存中删除变量。实例代码如下：（实例位置：光盘\TM\Instances\3\6）

```
<?php
echo "变量(\$str1)直接赋值为 null: ";
echo "变量(\$str2)未声明,其值为: ";
echo "变量(\$str3)先被赋值为 'PHP',然后被unset(),其值为: "
```

```

$str1 = null; //变量$str1 被赋空值
$str3 = "str"; //变量$str3 被赋值 str
if(is_null($str1)) //判断$str1 是否为空
    echo "str1 = null";
echo "<p>变量($str2)未被赋值: "; //判断$str2 是否为空
if(is_null($str2))
    echo "str2 = null";
echo "<p>被unset()函数处理过的变量($str3): ";
unset($str3); //释放$str3
if(is_null($str3)) //判断$str3 是否为空
    echo "str3 = null";
?>

```

运行结果如图 3.5 所示。

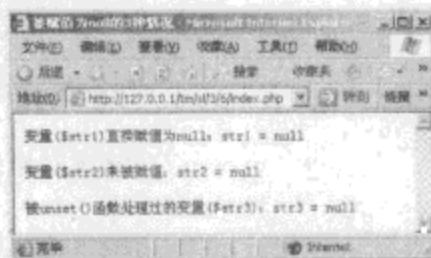


图 3.5 被赋值为 null 的 3 种情况

说明：is\_null() 函数用来判断变量是否为 null，该函数返回一个 bool 型，如果变量为 null，则返回 true，否则返回 false。unset() 函数用来销毁指定的变量。

注意：从 PHP 4.0 开始，unset() 函数就不再有返回值，所以不要试图获取或输出 unset()。

### 3.3.6 数据类型的转换

虽然 PHP 是弱类型语言，但有时仍然需要用到类型转换。PHP 中的类型转换和 C 语言一样简单，只须在变量前加上用括号括起来的类型名称即可。允许转换的类型如表 3.2 所示。

表 3.2 类型强制转换

转换操作符	转换类型	举例
(boolean)	转换成布尔型	(boolean)\$num、(boolean)\$str
(string)	转换成字符型	(string)\$boo、(string)\$flo
(integer)	转换成整型	(integer)\$boo、(integer)\$str
(float)	转换成浮点型	(float)\$str、(float)\$str
(array)	转换成数组	(array)\$str
(object)	转换成对象	(object)\$str

注意：在进行类型转换的过程中应该注意——转换成 boolean 型时，null、0 和未赋值的变量或数组会被转换为 false，其他的为真；转换成整型时，布尔型的 false 转换为 0，true 转换为 1，浮点型的小数部分被舍去，字符型如果以数字开头，就截取到非数字位，否则输出 0。

类型转换还可以通过 settype() 函数来完成，该函数可以将指定的变量转换成指定的数据类型。

```
bool settype ( mixed var, string type )
```

参数 var 为指定的变量，参数 type 为指定的类型，参数 type 有 7 个可选值：boolean、float、integer、array、null、object 和 string。如果转换成功则返回 true，否则返回 false。

当字符串转换为整型或浮点型时，如果字符是以数字开头的，就会先把数字部分转换为整型，再舍去后面的字符串。如果数字中含有小数点，则会取到小数点前一位。

**例 3.7 应用上面的方法将指定的字符串转换为其他类型，比较一下，看看转换后的数据有什么不同，实例代码如下：**（实例位置：光盘\TM\Instances\3\7）

```
<?php
$num = "2.568475E2";
echo "原始 string 型: ".$num; //输出原始字符串
echo "<br>";
echo "转换为 boolean 型: ".(boolean)$num; //将字符串转换为布尔型并输出
echo "<br>";
echo "转换为 integer 型: ".(integer)$num; //将字符串转换为整型并输出
echo "<br>";
echo "转换为 float 型: ".(float)$num; //将字符串转换为浮点型并输出
echo "<br>";
echo "转换为 array 型: ".(array)$num; //将字符串转换为数组型并输出
?>
```

运行结果如图 3.6 所示。

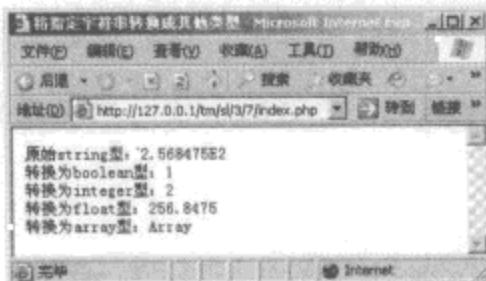


图 3.6 将指定字符串转换成其他类型

## 3.4 PHP 常量

**视频讲解：**光盘\TM\Video\3\PHP 常量.exe

本节介绍 PHP 的常量，包括常量的声明和使用，以及预定义常量。

### 3.4.1 声明和使用常量

常量可以理解为值不变的变量。常量值被定义后，在脚本的其他任何地方都不能改变。一个常量由英文字母、下划线和数字组成，但数字不能作为首字母出现。

在 PHP 中使用 define() 函数来定义常量，该函数的语法格式为：

```
define(string constant_name,mixed value,case_sensitive=true)
```

参数说明如下。

- constant\_name: 必选参数, 常量名称, 即标志符。
- value: 必选参数, 常量的值。
- case\_sensitive: 可选参数, 指定是否大小写敏感, 设定为 true 表示不敏感。默认状态表示大小写敏感。

获取常量值有两种方法: 一种是使用常量名直接获取值。另一种方法是使用 constant() 函数, 这和直接使用常量名输出的效果是一样的。但函数可以动态输出不同的常量, 在使用上要灵活、方便得多。函数的语法格式为:

```
mixed constant(string const_name)
```

参数 const\_name 为要获取常量的名称, 也可为存储常量名的变量。如果成功则返回常量的值, 失败则提示错误信息常量没有被定义。

要判断一个常量是否已经定义, 可以使用 defined() 函数。函数的语法格式为:

```
bool defined(string constant_name);
```

参数 constant\_name 为要获取常量的名称, 成功则返回 true, 否则返回 false。

例 3.8 为了便于读者更好地理解如何定义常量, 这里给出一个定义常量的实例。在这个实例中应用上述的 3 个函数: define() 函数、constant() 函数和 defined() 函数, 通过 define() 函数来定义一个常量, 使用 constant() 函数来动态获取常量的值, 应用 defined() 函数来判断常量是否被定义, 代码如下: (实例位置: 光盘\TM\Instances\3\8)

```
<?php
define ("MESSAGE","PHP 编程词典"); // 定义常量, 并设置大小写敏感
echo MESSAGE."<BR>"; // 输出常量 MESSAGE
echo Message."<BR>"; // 输出"Message", 表示没有该常量
define ("COUNT","PHP 编程词典, 让您编程无忧",true); // 定义常量 COUNT
echo COUNT."<BR>"; // 输出常量 COUNT
echo Count."<BR>"; // 输出常量 COUNT, 因为设定大小写不敏感
$name = "count";
echo constant ($name)."<BR>"; // 输出常量 COUNT
if (defined ("MESSAGE")){ // 如果定义返回 true, 使用 echo 输出显示信息
    echo "PHP 编程词典, 易查、易学、易用的高效编程工具! ";
}
?>
```

运行结果如图 3.7 所示。

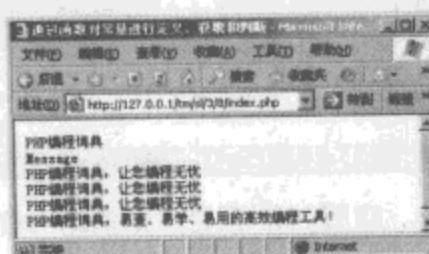


图 3.7 通过函数对常量进行定义、获取和判断

### 3.4.2 预定义常量

PHP 中可以使用预定义常量获取 PHP 中的信息。常用的预定义常量如表 3.3 所示。

表 3.3 PHP 的预定义常量

常量名	功能
FILE	默认常量，PHP 程序文件名
LINE	默认常量，PHP 程序行数
PHP_VERSION	内建常量，PHP 程序的版本，如 3.0.8_dev
PHP_OS	内建常量，执行 PHP 解析器的操作系统名称，如 Windows
TRUE	这个常量是一个真值 (true)
FALSE	这个常量是一个假值 (false)
NULL	一个 null 值
E_ERROR	这个常量指到最近的错误处
E_WARNING	这个常量指到最近的警告处
E_PARSE	这个常量指解析语法有潜在问题处
E_NOTICE	这个常量为发生不寻常的提示但不一定是错误处

 注意：“\_\_FILE\_\_” 和 “\_\_LINE\_\_” 中的 “\_\_” 是两条下划线，而不是一条 “\_”。

 说明：表中以 “E\_” 开头的预定义常量，是 PHP 的错误调试部分。如果想详细了解相关内容，可参考 error\_reporting() 函数。

**例 3.9** 预定义常量与用户自定义常量在使用上没什么差别。下面应用预定义常量来输出 PHP 中的信息，实例代码如下：（实例位置：光盘\TM\Instances\3\9）

```
<?php
echo "当前文件路径: __FILE__"; //输出 "__FILE__" 常量
echo "<br>当前行数: __LINE__"; //输出 "__LINE__" 常量
echo "<br>当前 PHP 版本信息: ".PHP_VERSION; //输出 PHP 版本信息
echo "<br> 当前操作系统: ".PHP_OS; //输出系统信息
?>
```

运行结果如图 3.8 所示。

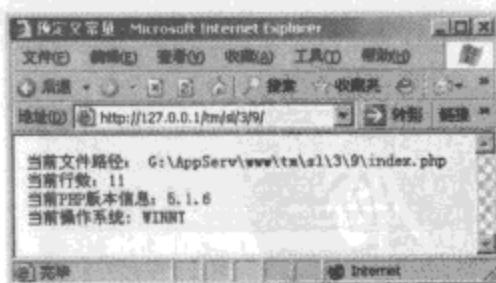


图 3.8 应用 PHP 预定义常量输出信息

 注意：根据每个用户操作系统和软件版本的不同，该程序的运行结果也不一定相同。

## 3.5 PHP 变量

### ■ 视频讲解：光盘\TM\Video\3\PHP 变量.exe

其值可以改变的量称为变量。变量为开发人员提供了一个有名字的内存存储区，程序中可以通过变量名对内存存储区进行读、写操作。为了确定每个变量内存存储区的大小，存储区中可以存放数据范围，以及变量可以使用的运算符。系统为程序中的每一个变量分配一个存储单元，变量名实质上就是计算机内存单元的命名。因此，借助变量名就可以访问内存中的数据。

### 3.5.1 变量声明及使用

和很多语言不同，在 PHP 中使用变量之前不需要声明变量（PHP 4.0 之前需要声明变量），只需为变量赋值即可。PHP 中的变量名称用\$和标识符表示，变量名是区分大小写的。

PHP 中的变量名称遵循以下约定：

- ✓ 在 PHP 中的变量名是区分大小写的。
- ✓ 变量名必须是以美元符号 (\$) 开始。
- ✓ 变量名开头可以以下划线开始。
- ✓ 变量名不能以数字字符开头。
- ✓ 变量名可以包含一些扩展字符（如重音拉丁字母），但不能包含非法扩展字符（如汉字字符和汉字字母）。

 技巧：声明的变量不可以与已有的变量重名，否则将引起冲突。变量的名称应采用能反映变量含义的名称，以利于提高程序的可读性。只要能明确反映变量的含义，可以使用英文单词、单词缩写、拼音（尽量使用英文单词），如 \$book\_name, \$user\_age, \$shop\_price 等，必要时，也可以将变量的类型包含在变量名中，如 \$book\_id\_int，这样可以直接根据变量名称了解变量的类型。

在程序中使用变量前，需要为变量赋值。PHP 中变量的定义非常简单灵活。在定义变量时，不需要指定变量的类型，PHP 自动根据对变量的赋值决定其类型。变量的赋值是通过使用赋值运算符“=”实现的。在定义变量时也可以直接为变量赋值，此时称之为变量的初始化。

例 3.10 下面的代码定义了一个整型变量 n\_sum，将其赋值为 100；定义一个布尔型变量；定义一个空字符串。（实例位置：光盘\TM\Instances\3\10）

```
<?php
    $n_sum = 100; // 定义一个整型变量，并进行初始化
    $str1=false; // 定义一个布尔型变量，并进行初始化
    $str2=" "; // 定义一个空字符串
?>
```

**◆ 注意：**在定义变量时，要养成良好的编程习惯，在定义变量前要对其定义初始值。如果在定义变量时没有指定变量的初始值，那么在使用变量时，PHP 会根据变量在语句中所处的位置确定其类型，并采用该类型的默认值。字符串的初始值为空值；整型的初始值为 0；布尔型的初始值为 false。

对变量赋值时，要遵循变量命名规则，如下面的变量命名是合法的。

```
<?php
    $helpsoft="PHP 编程词典";
    $_book="计算机图书 ";
?

```

下面的变量命名则是非法的。

```
<?php
    $5_str="编程词典";           //变量名不能以数字字符开头
    $@zts = "zts";              //变量名不能以其他字符开头
?

```

**◆ 注意：** PHP 中的变量名称区分大小写，而函数名称不区分大小写。

除了直接赋值外，还有两种方式来给变量声明或赋值。一种是变量间的赋值。

**例 3.11** 变量间的赋值是指赋值后，两个变量使用各自的内存，互不干扰，实例代码如下：（实例位置：光盘\TM\Instances\3\11）

```
<?php
$str1 = "PHP 编程词典";          //声明变量$str1
$str2 = $str1;                    //使用$str1 来初始化$str2
$str1 = "我喜欢学 PHP";          //改变变量$str1 的值
echo $str2;                      //输出变量$str2 的值
?
```

结果为：PHP 编程词典

另一种是引用赋值。从 PHP 4.0 开始，PHP 引入了“引用赋值”的概念。引用赋值是指用不同的名字访问同一个变量内容，当改变其中一个变量的值时，另一个也跟着发生变化。使用&符号来表示引用。

**例 3.12** 本例中，变量\$str2 是变量\$str 的引用，当给变量\$str 赋值后，\$str2 的值也会跟着发生变化。实例代码如下：（实例位置：光盘\TM\Instances\3\12）

```
<?php
$str = "学习 PHP 很轻松";          //声明变量$str
$str2 = & $str;                    //使用引用赋值，这是$str2 已经赋值成为“学习 PHP 很轻松”
$str = "我要大声的告诉你：$str";   //重新给$str 赋值
echo $str2;                        //输出变量$str2
echo "<p>";
echo $str;                          //输出变量$str
?

```

运行结果如图 3.9 所示。

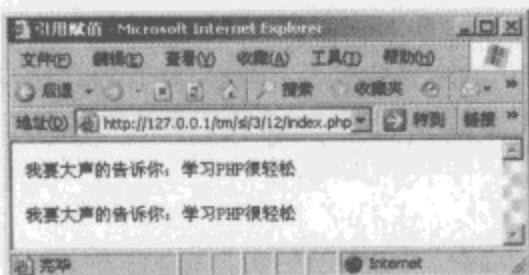


图 3.9 引用赋值

引用变量并不是复制一个变量给另一个变量，而是将两个变量指向同一个内容，可以理解为将同一个变量的地址传递给另一个变量。引用后，两个变量完全相同。当对其中的任意一个变量的内容进行更改时，另一个变量的内容也会随之更改。

引用和复制的区别在于：复制是将原变量内容复制下来，开辟一个新的内存空间来保存。而引用则是给变量的内容再命一个名字。也可以这样理解，一些论坛的版主，博客的博主，登录网站时发表帖子或文章时一般不会留真名，而是用笔名，这个笔名就可以看作是一个引用，是其身份的代表。

### 3.5.2 变量作用域

变量的作用域是指该变量在程序中可以被使用的区域。变量在使用时，要符合变量的定义规则。变量必须在有效范围内使用，如果变量超出有效范围，变量也就失去其意义了。按作用域可以将变量分为全局变量、局部变量和静态变量。变量作用域的说明如表 3.4 所示。

表 3.4 变量作用域

作用域	说 明
全局变量	即被定义在所有函数以外的变量，其作用域是整个 PHP 文件，但是在用户自定义函数内部是不可用的。想在用户自定义函数内部使用全局变量，要使用 <code>global</code> 关键词声明，或者通过使用全局数组 <code>\$globals</code> 进行访问
局部变量	即在函数的内部定义的变量，这些变量只限于在函数内部使用，在函数外部不能被使用
静态变量	能够在函数调用结束后仍保留变量值，当再次回到其作用域时，又可以继续使用原来的值。而一般变量是在函数调用结束后，其存储的数据值将被清除，所占的内存空间被释放。使用静态变量时，先要用关键字 <code>static</code> 来声明变量，需要把关键字 <code>static</code> 放在要定义的变量之前

在函数的内部定义的变量，其作用域是所在函数。如果在函数外赋值，将被认为是完全不同的另一个变量。在退出声明变量的函数时，该变量及相应的值就会被撤销。

**例 3.13** 下面在自定义函数中应用全局变量与局部变量进行对比。在本实例中定义两个全局变量 `$zy` 和 `$zyy`，在用户自定义函数 `lxt()` 中，如果想要在第 6 行和第 8 行调用它们，而程序输出的结果是“我喜欢 PHP 语言”，因为在第 7 行用 `global` 关键字声明了全局变量 `$zyy`。而第 5 行由于定义了局部变量，因此输出的结果为“我喜欢”，其中第 5 行的 `$zy` 和第 2 行的 `$zy` 没有任何关系。实例代码如下：（实例位置：光盘\TM\Instances\3\13）

```
<?php
$zy = "你好。";
```

```

$zyy = "PHP 语言";
function lxt () {
    $zy="我喜欢";           //定义局部变量
    echo $zy;               //输出的是局部变量的内容，而非全局
    global $zyy;            //应用关键字 global 在函数内部定义全局变量
    echo $zyy."<br>";       //此处调用$zyy
}
lxt ();
?>

```

运行结果：我喜欢 PHP 语言

**◆ 注意：**因为默认情况下全局变量和局部变量的作用域是不相交的，所以，在函数内部可以定义与全局变量同名的变量。全局变量可以在程序中的任何地方访问。但是在用户自定义函数内部是不可用的，想在用户自定义函数内部使用全局变量，要使用 `global` 关键字声明。

静态变量在函数内部定义，只局限于函数内部使用，但却具有和程序文件相同的生命周期。也就是说，静态变量一旦被定义，则在当前程序文件结束之前一直存在。

静态变量通过在变量前使用关键词 `static` 声明变量，格式如下。

```
static $str;
```

**例 3.14** 下面应用静态变量和普通变量同时输出一个数据，看两者功能有什么不同。实例代码如下：（实例位置：光盘\TM\Instances\3\14）

```

<?php
function zdy () {
    static $message = 0;           //初始化静态变量
    $message+=1;                  //静态变量加 1
    echo $message." ";           //输出静态变量
}
function zdy1() {
    $message = 0;                 //声明函数内部变量（局部变量）
    $message += 1;                //局部变量加 1
    echo $message." ";           //输出局部变量
}
for ($i=0 ; $i<10 ; $i++)      zdy();        //输出 1~10
echo "<br>";
for ($i=0 ; $i<10 ; $i++)      zdy1();       //输出 10 个 1
?>

```

运行结果如图 3.10 所示。

自定义函数 `zdy()` 是输出 1~10 的 10 个数字，而 `zdy1()` 函数则输出的是 10 个 1。因为自定义函数 `zdy()` 含有静态变量，而函数 `zdy1()` 是一个普通变量。初始化都为 0，再分别使用 `for` 循环调用两个函数，结果是静态变量的函数 `zdy()` 在被调用后保留了 `$message` 中的值，静态变量的初始化只是

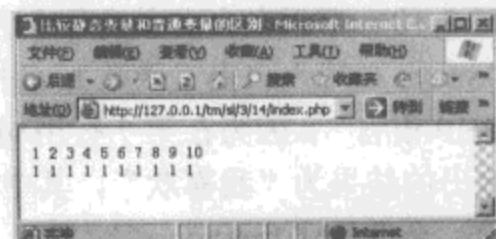


图 3.10 比较静态变量和普通变量的区别

在第一次遇到时被执行，以后就不再对其进行初始化操作了，将会略过第三行代码不执行；而普通变量的函数 zdy1()在被调用后，其变量\$message 失去了原来的值，重新被初始化为 0。

### 3.5.3 可变变量

可变变量是一种独特的变量，变量的名称并不是预先定义好的，而是动态地设置和使用。可变变量一般是指使用一个变量的值作为另一个变量的名称，所以可变变量又称为变量的变量。可变变量通过在一个变量名称前使用两个“\$”符号实现。

**例 3.15** 下面应用可变变量实现动态改变变量的名称。首先定义两个变量\$change\_name 和\$php，并且输出变量\$change\_name 的值，然后应用可变变量来改变变量\$change\_name 的名称，最后输出改变名称后的变量值，程序代码如下：（实例位置：光盘\TM\Instances\3\15）

```
<?php
$change_name = "php"; //声明变量$change_name
$php = "编程的关键因素在于学好语言基础!";
echo $change_name; //输出变量$change_name
echo $$change_name; //通过可变变量输出$php 的值
?>
```

结果为：php 编程的关键因素在于学好语言基础！

### 3.5.4 预定义变量

PHP 还提供了很多非常实用的预定义变量，通过这些预定义变量可以获取到用户会话、用户操作系统的环境和本地操作系统的环境等信息。常用的预定义变量如表 3.5 所示。

表 3.5 预定义变量

变量的名称	说 明
\$ SERVER['SERVER_ADDR']	当前运行脚本所在服务器的 IP 地址
\$_SERVER['SERVER_NAME']	当前运行脚本所在服务器主机的名称。如果该脚本运行在一个虚拟主机上，则该名称是由那个虚拟主机所设置的值决定
\$_SERVER['REQUEST_METHOD']	访问页面时的请求方法，如 GET、HEAD、POST、PUT。如果请求的方式是 HEAD，则 PHP 脚本将在送出头信息后终止（这意味着在产生任何输出后，不再有输出缓冲）
\$ SERVER['REMOTE_ADDR']	正在浏览当前页面用户的 IP 地址
\$_SERVER['REMOTE_HOST']	正在浏览当前页面用户的主机名。反向域名解析基于该用户的 REMOTE ADDR
\$_SERVER['REMOTE_PORT']	用户连接到服务器时所使用的端口
\$_SERVER['SCRIPT_FILENAME']	当前执行脚本的绝对路径名。注意，如果脚本在 CLI 中被执行，作为相对路径，如 file.php 或者 ./file.php，\$_SERVER['SCRIPT_FILENAME'] 将包含用户指定的相对路径

续表

变量的名称	说 明
<code>\$_SERVER['SERVER_PORT']</code>	服务器所使用的端口。默认为 80。如果使用 SSL 安全连接，则这个值为用户设置的 HTTP 端口
<code>\$_SERVER['REMOTE_ADDR']</code>	正在浏览当前页面用户的 IP 地址
<code>\$_SERVER['DOCUMENT_ROOT']</code>	当前运行脚本所在的文档根目录。在服务器配置文件中定义
<code>\$_COOKIE</code>	通过 <code>HTTPCookie</code> 传递到脚本的信息。这些 cookie 多数是由执行 PHP 脚本时通过 <code>setcookie()</code> 函数设置的
<code>\$_SESSION</code>	包含与所有会话变量有关的信息。 <code>\$_SESSION</code> 变量主要应用于会话控制和页面之间值的传递
<code>\$_POST</code>	包含通过 POST 方法传递的参数的相关信息。主要用于获取通过 POST 方法提交的数据
<code>\$_GET</code>	包含通过 GET 方法传递的参数的相关信息。主要用于获取通过 GET 方法提交的数据
<code>\$GLOBALS</code>	由所有已定义全局变量组成的数组。变量名就是该数组的索引。它可以被称为所有超级变量的超级集合

### 3.5.5 变量的生存周期

变量存在的时间称为生存周期。即从变量被声明的那一刻起，直到脚本运行结束。对于过程级变量，其存活期仅是该过程运行的时间，该过程结束后，变量随之消失。在执行过程中，局部变量是理想的临时存储空间。在不同过程中可以使用同名的局部变量，这是因为每个局部变量只被声明它的过程识别。

## 3.6 PHP 运算符

### 视频讲解：光盘\TM\Video\3\PHP 运算符.exe

运算符是用来对变量、常量或数据进行计算的符号；它对一个值或一组值执行一个指定的操作。PHP 的运算符包括：字符串运算符、算术运算符、赋值运算符、递增或递减运算符、位运算符、逻辑运算符、比较运算符和条件运算符。下面进行详细讲解。

### 3.6.1 字符串运算符

字符串运算符只有一个，即英文的句号“.”。它将两个字符串连接起来，结合到一起形成一个新的字符串。使用过 C 或 Java 的读者要注意了，这里的“+”号，只做赋值运算符使用，而不能做字符串运算符。

**例 3.16** 下面应用一个实例，来看一下“.”和“+”两者之间的区别。当使用“.”时，变量 \$m 和 \$n 两个字符串组成一个新的字符“18.26e9”，当使用“+”时，PHP 会认为这是一次运算。如果“+”

号的两边有字符类型，则自动转换为整型，如果是字母，则输出为0，如果是以数字开头的字符串，则会截取字符串头部的数字，再进行运算，实例代码如下：（实例位置：光盘\TM\Instances\3\16）

```
<?php
$n = "18.26e";
//声明一个字符串变量，以数字开头
$m = 9;
//声明一个整型变量
$nm = $n.$m;
//使用“.”运算符将两个变量连接
echo $nm."\t";
$mn = $n + $m;
//使用“+”运算符将两个变量连接
echo $mn;
?>
```

结果为：18.26e9 27.26

### 3.6.2 算术运算符

算术运算符用来连接运算表达式。算术运算符包括加（+）、减（-）、乘（\*）、除（/）、取模（%）等运算符。常用的算术运算符如表3.6所示。

表3.6 常用的算术运算符

算术操作符	说 明	举例（定义\$a=5, \$b=3）	结 果
+	加法运算	\$a + \$b	值为8
-	减法运算	\$a - \$b	值为2
*	乘法运算	\$a * \$b	值为15
/	除法运算	\$a / \$b	值为1.66
%	取模运算	\$a % \$b	值为2

说明：在算术运算符中使用%求余，如果被除数（\$a）是负数的话，那么取得的结果也是一个负值。

例3.17 在本例中分别应用算术运算符对表达式进行计算，实例代码如下：（实例位置：光盘\TM\Instances\3\17）

```
<?php
$a = 15;
//声明变量$a
$b = -8;
//声明变量$b
$c = 26;
//声明变量$c
echo "\$a = ".$a."  
";
//输出变量$a
echo "\$b = ".$b."  
";
//输出变量$b
echo "\$c = ".$c."  
";
//输出变量$c
echo "\$a + \$b = ".($a + $b)."  
";
//计算变量$a 加 $b 的值
echo "\$a - \$b = ".($a - $b)."  
";
//计算变量$a 减 $b 的值
echo "\$a * \$b = ".($a * $b)."  
";
//计算 $a 乘以 $b 的值
echo "\$a / \$b = ".($a / $b)."  
";
//计算 $a 除以 $b 的值
echo "\$a \% \$c = ".($a % $c);
//计算 $a 和 $c 的余数，被除数为15
?>
```

运行结果如图3.11所示。

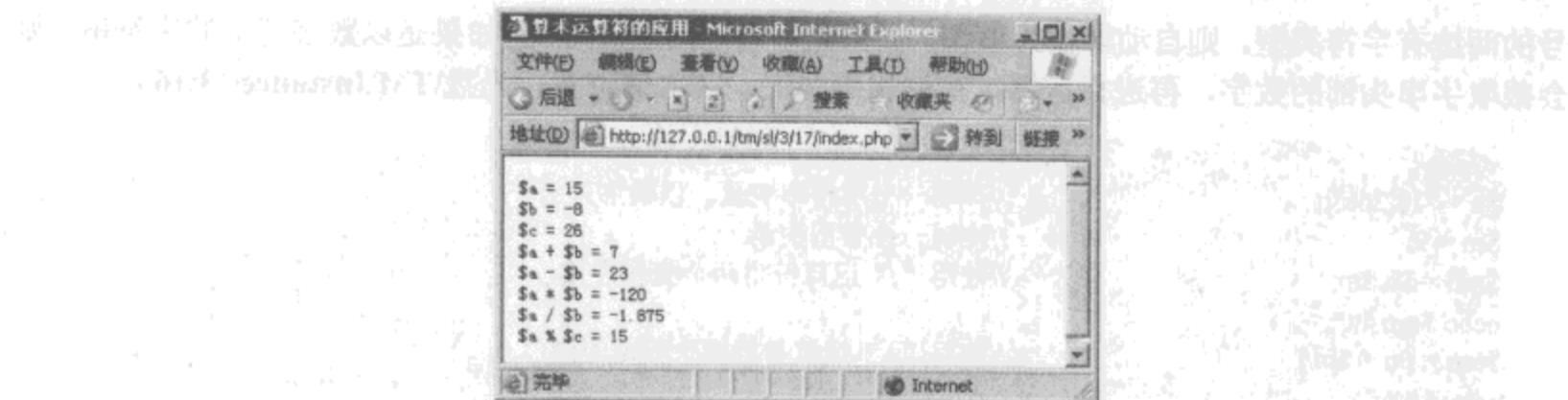


图 3.11 算术运算符的应用

### 3.6.3 赋值运算符

最基本的赋值运算符是“=”，用于对变量进行赋值，而其他运算符可以和赋值运算符“=”联合使用，构成组合赋值运算符。赋值运算符是把基本赋值运算符“=”右边的值赋给左边的变量或者常量。在 PHP 中的赋值运算符如表 3.7 所示。

表 3.7 常用赋值运算符

操作	符 号	举 例	展 开 形 式	意 义
赋值	=	\$a=b	\$a=3	将右边表达式的值赋给左边的变量
加	+=	\$a+= b	\$a=\$a + b	将运算符左边的变量加上右边表达式的值赋给左边的变量
减	-=	\$a-= b	\$a=\$a-b	将运算符左边的变量减去右边表达式的值赋给左边的变量
乘	*=	\$a*= b	\$a=\$a * b	将运算符左边的变量乘以右边表达式的值赋给左边的变量
除	/=	\$a/= b	\$a=\$a / b	将运算符左边的变量除以右边表达式的值赋给左边的变量
连接字符	.=	\$a.= b	\$a=\$a. b	将右边的字符加到左边
取余数	%=	\$a% = b	\$a=\$a % b	将运算符左边的变量用右边表达式的值求模，并将结果赋给左边的变量

**例 3.18** 应用赋值运算符给指定的变量赋值，并计算各表达式的值，代码如下：（实例位置：光盘\TM\Instances\3\18）

```
<?php
    $a = 10; //声明变量$a
    $b = 8; //声明变量$b
    echo "\$a = ".$a."<br>"; //输出变量$a
    echo "\$b = ".$b."<br>"; //输出变量$b
    echo "\$a += \$b = " . ($a += $b). "<br>"; //计算变量$a 加$b 的值
    echo "\$a -= \$b = " . ($a -= $b). "<br>"; //计算变量$a 减$b 的值
    echo "\$a *= \$b = " . ($a *= $b). "<br>"; //计算$a 乘以$b 的值
    echo "\$a /= \$b = " . ($a /= $b). "<br>"; //计算$a 除以$b 的值
    echo "\$a %= \$b = " . ($a %= $b); //计算$a 和$b 的余数
?>
```

运行结果如图 3.12 所示。

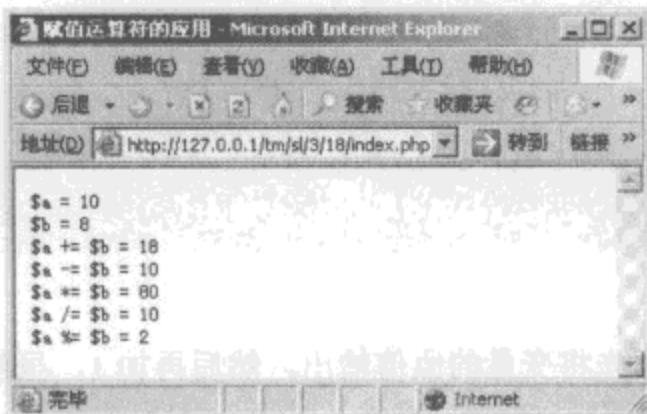


图 3.12 赋值运算符的应用

### 3.6.4 递增递减运算符

算术运算符适合在有两个或者两个以上不同操作数的场合使用，但是，当只有一个操作数时，使用算术运算符是没有必要的。这时，就可以使用“++”或者“--”运算符了，即递增或递减运算符。

递增或递减运算符有两种使用方法，一种是先将变量增加或者减少 1 后再将值赋给原变量，称为前置递增或递减运算符（也称前置自增自减运算符）；另一种是将运算符放在变量后面，即先返回变量的当前值，然后变量的当前值增加或者减少 1，称为后置递增或递减运算符（后置自增自减运算符）。

#### 1. 前置递增运算符

前置递增运算符是指使用前置“++”运算符将变量加 1 再将值赋给原变量。例如：

```
<?php
$a=8;
echo ++$a;
?>
```

结果为：9

该程序的运行结果为 9，也就是首先把变量 \$a 加 1，再将结果赋给原变量，所以运行本程序后，原变量 \$a 的值已经变成 9。

#### 2. 前置递减运算符

前置递减运算符是指使用前置“--”运算符将变量减 1 再将值赋给原变量。例如：

```
<?php
$a=8;
echo --$a;
?>
```

结果为：7

该程序的运行结果为 7，也就是首先把变量 \$a 减 1，再将结果赋给原变量，所以运行本程序后，原变量 \$a 的值已经变成 7。

#### 3. 后置递增运算符

后置递增运算符是指先返回变量的当前值，然后再使用后置“++”运算符将变量的当前值再加 1。

例如：

```
<?php
    $a=8;
    echo $a++;
?>
```

结果为：8

该程序的运行结果为 8，即先将变量的原值输出，然后再加 1，虽然显示输出的是 8，但实际上变量 \$a 的值已经是 9 了。

#### 4. 后置递减运算符

后置递减运算符是指先返回变量的当前值，然后使用后置 “--” 运算符将变量的当前值再减 1。

例如：

```
<?php
    $a=8;
    echo $a--;
?>
```

结果为：8

该程序的运行结果为 8，即先将变量的原值输出，然后再减 1，虽然显示输出的是 8，但实际上变量 \$a 的值已经是 7 了。

### 3.6.5 位运算符

位运算符是指对二进制位从低位到高位对齐后进行运算。在 PHP 中的位运算符如表 3.8 所示。

表 3.8 位运算符

符 号	作 用	举 例
&	按位与	\$m & \$n
	按位或	\$m   \$n
^	按位异或	\$m ^ \$n
~	按位取反	\$m ~ \$n
<<	向左移位	\$m << \$n
>>	向右移位	\$m >> \$n

例 3.19 应用位运算符对变量中的值进行位运算操作，实例代码如下：(实例位置：光盘\TM\Instances\3\19)

```
<?php
$m = 10;
$n = 16;
$mn = $m & $n;           //位与
echo $mn . "\t";
$mn = $m | $n;           //位或
```

```

echo $mn . "\t";
$mn = $m ^ $n;           //位异或
echo $mn . "\t";
$mn = ~$m;               //位取反
echo $mn;
?>

```

结果为： 0 26 26 -11

### 3.6.6 逻辑运算符

逻辑运算符用来组合逻辑运算的结果，是程序设计中一组非常重要的运算符。PHP 的逻辑运算符如表 3.9 所示。

表 3.9 PHP 的逻辑运算符

逻辑运算符	说 明	结 果 为 真
&& 或 and	逻辑与，只有当两个操作数的值都为 true 时，\$m && \$n 的值才为 true	当 \$m 和 \$n 都为真时
或 or	逻辑或，只要两个操作数中有一个值为 true，\$m    \$n 的值就为 true	当 \$m 为真或者 \$n 为真时
xor	逻辑异或，当两个操作数的值为一真一假时，\$m xor \$n 的值为 true	当 \$m、\$n 一真一假时
!	逻辑非，如果其单一操作数为 true，则返回 false；否则返回 true	当 \$m 为假时

在逻辑运算符中，逻辑与和逻辑或这两个运算符有 4 种运算符号（&&、and 和 ||、or），其中属于同一个逻辑结构的两个运算符号（如&& 和 and）之间却有着不同的优先级。

**例 3.20** 下面分别应用逻辑或中的运算符号“||”和“or”进行相同的判断，但是因为同一逻辑结构的两个运算符（“||”和“or”）的优先级不同，输出的结果也不同。实例代码如下：（实例位置：光盘\TM\Instances\3\20）

```

<?php
$i = true;           //声明一个布尔型变量 $i，赋值为真
$j = true;           //再声明一个布尔变量 $j，赋值也为真
$z = false;          //最后再声明一个初值为假的布尔变量 $z
if($i or $j and $z)
    echo "true";     //这是用 or 做判断
                    //如果 if 表达式为真，输出 true
else
    echo "false";    //否则输出 false
    echo "<br>";
if($i || $j and $z)
    echo "true";     //这是用 || 做判断
                    //如果表达式为真，输出 true
else
    echo "false";    //如果表达式为假，输出 false
?>

```

结果为：

```
true  
false
```

**◆ 注意：**可以看到，两个 if 语句，除了 or 和 || 不同之外，其他完全一样，但最后的结果却是南辕北辙，正好相反。在实际应用中要多注意这样的细节。

### 3.6.7 比较运算符

比较运算符就是对变量或表达式的结果进行大小、真假等比较，如果比较结果为真，则返回 true；如果为假，则返回 false。PHP 中的比较运算符如表 3.10 所示。

表 3.10 PHP 的比较运算符

运 算 符	说 明	举 例 (定 义 \$m=5, \$n=3)	结 果
<	小 于	\$m < \$n	false
>	大 于	\$m > \$n	true
<=	小 于 等 于	\$m <= \$n	false
>=	大 于 等 于	\$m >= \$n	true
==	相 等	\$m == \$n	false
!=	不 等	\$m != \$n	true
====	恒 等	\$m === \$n	false
!==	非 恒 等	\$m !== \$n	true

比较运算“====”和“!==”不太常见。“\$m === \$n”，说明 \$m 和 \$n 不只是数值上相等，而且两者的类型也一样。“!=”和“====”的含义差不多，“\$m != \$n”是指 \$m 和 \$n 或者数值不相等，或者类型不相等。

**例 3.21** 下面应用比较运算符对变量中的值进行比较，设置变量“\$value = "100"”，变量的类型为字符串型，将变量\$value 与数字 100 进行比较，会发现比较的结果非常有趣。其中使用的 var\_dump 函数是系统函数，作用是输出变量的相关信息。实例代码如下：（实例位置：光盘\TM\Instances\3\21）

```
<?php  
$value="100"; //声明一个字符串变量$value  
echo "$value = \"$value\"";  
echo "<p>$value==100: ";  
var_dump($value==100); //结果为: bool(true)  
echo "<p>$value==true: ";  
var_dump($value==true); //结果为: bool(true)  
echo "<p>$value!=null: ";  
var_dump($value!=null); //结果为: bool(true)  
echo "<p>$value==false: ";  
var_dump($value==false); //结果为: bool(false)  
echo "<p>$value === 100: ";  
var_dump($value==100); //结果为: bool(false)  
echo "<p>$value==true: ";
```

```

var_dump($value==true);           //结果为: bool(false)
echo "<p>(10/2.0 != 5): ";
var_dump(10/2.0 != 5);           //结果为: bool(true)
?>

```

运行结果如图 3.13 所示。



图 3.13 比较运算符的应用

### 3.6.8 条件运算符

条件运算符用于根据一个表达式在另两个表达式中选择一个，而不是用来在两个语句或者程序中选择。条件运算符也称三元运算符，条件运算符的使用最好放在括号里。

**例 3.22** 下面应用条件运算符实现一个简单的判断功能，如果变量 \$age 的值大于等于 18，则输出“小明是成年人”，否则输出“小明未成年”，实例代码如下：（实例位置：光盘\TM\Instances\3\22）

```

<?php
$age=16;
echo ($age>=18)?小明是成年人: 小明未成年;
?>

```

运行结果：小明未成年

### 3.6.9 运算符的使用规则

所谓运算符的优先级，是指在应用中哪一个运算符先计算，哪一个后计算，与数学的四则运算遵循的“先乘除，后加减”是一个道理。

PHP 的运算符在运算中遵循的规则是：优先级高的运算先做，优先级低的操作后做，同一优先级的操作按照从左到右的顺序进行。也可以像四则运算那样使用小括号，括号内的运算最先进行。PHP 运算符优先级的一览表如表 3.11 所示。

表 3.11 运算符的优先级

优 先 级 别	运 算 符
1	or、and、xor
2	=、+=、-=、*=、/=、.=、%=?
3	、&&
4	、^
5	&、.
6	++、--（递增或递减运算符）
7	/、*、%
8	<<、>>
9	++、--
10	+、-（正、负号运算符）、!、~
11	==、!=、<>
12	<、<=、>、>=
13	?:
14	->
15	=>

### 3.7 PHP 的表达式

将运算符和操作数连接起来的式子称为表达式。表达式是构成 PHP 程序语言的基本元素，也是 PHP 最重要的组成元素。根据运算符的不同，表达式可以分为算术表达式、字符串表达式、关系表达式、赋值表达式以及逻辑表达式等。

在 PHP 语言中，几乎所写的任何语句都是表达式。最基本的表达式形式是常量和变量，如“\$a=8”，即表示将值 8 赋给变量 \$a。

表达式是通过具体的代码来实现的，是多个符号集合起来组成的代码，而这些符号只是一些对 PHP 解释程序有具体含义的最小单元。它们可以是变量名、函数名、运算符、字符串、数值和括号等。如下面的代码：

```
<?php
"fine";
$a = "php";
?>
```

这就是由两个表达式组成的脚本，即“fine”和“\$a="php"”。此外，还可以进行连续赋值，例如：

```
<?php
$b = $a = 8;
?>
```

因为 PHP 赋值操作的顺序是由右到左的，所以变量 \$b 和 \$a 都被赋值 8。

在 PHP 的代码中，使用分号 “;” 来区分表达式，表达式也可以包含在括号内。可以这样理解：一个表达式再加上一个分号，就是一条 PHP 语句。

表达式的应用范围非常广泛，可以调用一个数组、创建一个类、给变量赋值等。

**◆ 注意：**在编写程序时，应该注意表达式后面的这个分号 “;”，不要漏写，这是一个出现频率很高的错误点。

## 3.8 实战

### ■ 视频讲解：光盘\TM\Video\3\判断指定的年份是否为闰年.exe

记得一位老红军曾说过：“没上过战场的军人算不上真正的军人”。这句话是有一定道理的。从程序员的角度来说，只有理论而没有实际开发能力的程序员，不能够算是程序员。本节将通过两个综合实例运用前面所学知识解决开发中的一些问题，以此来巩固大家所学的知识，做到“理论与实践相结合”。

下面通过前面所讲的基础知识结合一个实例来演示如何在 PHP 中应用运算符和表达式。

**例 3.23** 下面应用运算符和表达式来判断用户指定的年份是否是闰年。闰年的判断方式是，能够被 4 整除不能够被 100 整除，或者能够被 100 整除并且能被 400 整除的年份为闰年。本实例的完整代码如下：（实例位置：光盘\TM\Instances\3\23）

```
<table width="392" height="101" border="1" align="center" cellpadding="1" cellspacing="1" bordercolor="#009900" bgcolor="#CCFF00">
<tr>
<td height="25" colspan="2" align="center" class="STYLE5">判断指定的年份是否为闰年</td>
</tr>
<tr>
<td width="160" height="30" bgcolor="#FFFF99"><span class="STYLE3">请输入一个四位数的年份:</span></td>
<td width="219" bgcolor="#FFFF99">
<form name="form1" method="post" action="">
<table width="196" height="29" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="136"><input name="txt_year" type="text" value="" size="20"> </td>
<td width="60">&ampnbsp
<input type="submit" name="Button" value="计算"></td>
</tr>
</table>
</form>
</td>
</tr>
<tr>
<td height="37" colspan="2" align="center">
<?php
if($_POST[Button]!=""){
$year=$_POST[txt_year];
}
?>

```

```

        $result = ($year%4==0 && $year%100!=0)||($year%400==0)?$year."是闰年":$year."不是闰年";
        echo $result;
    }
?>
        </td>
    </tr>
</table>

```

运行本实例，在文本框中输入指定的年份，单击“计算”按钮，即可求出该年份是否为闰年，运行结果如图 3.14 所示。



图 3.14 判断指定的年份是否为闰年

### 3.9 本章小结

本章主要介绍了 PHP 语言的基础知识，重点讲解了 PHP 的语法基础、数据类型、常量、变量、运算符和表达式。基础知识是一门语言的核心，希望初学者能静下心来，牢牢掌握本章的知识，只有基础扎实，才能真正迈向精通 PHP 的门槛。

## 3.10 实战练习

1. 应用字符串运算符输出如图 3.15 所示的结果。（答案位置：光盘\TM\Instances\3\24）

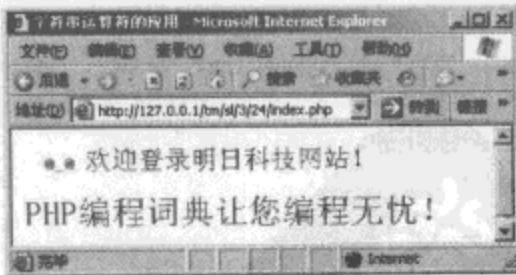


图 3.15 字符串运算符的应用

2. 在开发动态网站中的用户注册模块时，其中用户注册的密码一般都要填写一个确认密码，来加强输入信息的准确性。下面在用户注册模块中，应用比较运算符验证两次输入的密码是否一致。如果相等则输出注册成功信息，否则将输出错误提示信息。（答案位置：光盘\TM\Instances\3\25）

# 第 4 章

## 流程控制语句

( 视频讲解: 80 分钟)

条件控制语句和循环控制语句是两种基本的语法结构，它们都是用来控制程序执行流程的，也是构成程序的主要语法基础。通常，语句是按顺序执行的，即执行完当前的语句之后，执行下一条语句。但是这种简单的顺序执行方式能够解决的问题很有限，现实生活中的许多逻辑使用顺序执行的方式是不能够被描述的。例如，“如果明天不下雨，我们就去放风筝，否则我们呆在家里练习书法”。在程序中需要根据条件判断来确定是“放风筝”还是“呆在家里练习书法”，两种结果不可能同时存在，这里只能应用条件控制语句实现这个功能，而循环控制语句不能够确定选择哪种结果。因此，开发人员可以按照实际问题的逻辑思路选择性地使用流程控制语句来编写程序代码。

通过阅读本章，可以学习到：

- if 条件控制语句的应用
- if 多分支条件控制语句的应用
- switch…case 分支控制语句
- while 循环语句
- do…while 循环语句
- for 循环语句
- foreach 循环语句
- break、continue 跳转控制语句的应用

## 4.1 学前导读

程序由一条条语句组成，每条语句都被用来实现一个具体的任务。一般情况下，一段程序代码是顺序执行的，即从头到尾按顺序逐行执行。顺序执行是程序最为基本最为简单的结构。但有时却需要在某种条件下有选择地执行指定的操作，或者重复地执行某一类程序，就是所谓的程序流程的控制问题。流程控制语句包括条件控制语句、循环控制语句和跳转语句。下面分别进行详细介绍。

### 4.1.1 什么是算法

算法是指为解决某一个问题而采取的步骤和方法的描述。在程序设计中的算法是指对计算机工作的步骤和方法的描述。

算法的每一个步骤都是严格规定好的，是能够被计算机识别并正确执行的，并且每一个步骤都能够被计算机理解为一个或一组唯一的动作，而不使计算机产生歧义。算法必须有开始和结束，并且必须保证算法规定的每一个步骤最终都能够被完成。

下面通过一个例子对算法进行说明。

交换变量 M、N 中的内容。

计算机本身不能够直接执行这个操作，交换两个变量的最传统的方法就是借用第三方变量作中间量。算法可以表述如下：

- (1) 将变量 M 的内容赋给中间变量。
- (2) 将变量 N 的内容赋给变量 M。
- (3) 将中间变量存放的内容赋给变量 N。

最终算法可以写成：

- (1) O  $\leftarrow$  M。
- (2) M  $\leftarrow$  N。
- (3) N  $\leftarrow$  O。

算法，实际上就是用自然语言或者其他方式描述的一个计算机程序，编写计算机程序也就是把用某种方式描述的算法，使用程序设计的语言描述出来。

### 4.1.2 算法的描述方法

算法的描述方法有很多，有自然语言、传统流程图、N-S 结构化流程图和伪代码。自然语言就是在 4.1.1 节中使用的表述方法。使用自然语言进行描述通俗易懂，但容易产生歧义，而且在表示分支和循环时很不方便。下面将对其他 3 种描述方式分别进行介绍。

#### 1. 传统流程图

传统流程图是一种用图形来表示算法的描述方法，使用不同的几何图形来表示不同性质的操作，使用流程线来表示算法的执行方式。它具有直观、形象、易于理解的优点，应用比较广泛。

传统流程图由以下几部分组成。

-  起止框：算法的开始和结束。
-  处理框：初始化或运算赋值等操作。
-  输入输出框：表示输入输出数据。
-  判断框：根据一个条件成立与否，决定执行两种不同操作的其中一个。
-  连接点：用于流程分支的连接。
-  流程线：表示程序执行的方向。

将流程图框用流程线及连接点连接起来组成一个完整的传统流程图。传统流程图的具体形式在 4.1.3 节会作详细介绍。

## 2. N-S 结构化流程图

N-S 流程图将传统流程图中的流程线去掉，将全部算法写在一个矩形框中，这个框包含其他从属于它的从属框。当程序算法比较繁琐时，一般采用 N-S 结构化流程图。N-S 结构化流程图更有利于程序设计的结构化。

## 3. 伪代码

伪代码用介于自然语言和计算机语言之间的文字和符号对算法进行描述。如同一篇文章，自上而下地写下来，每一行（或几行）表示一个基本操作。使用伪代码描述算法并无固定、严格的语法规则，只要把意思表达清楚即可。例如，使用伪代码描述“判断变量 M 是否等于 0”的算法如下。

```

如果 M 等于 0
    输出字符串“变量 M 等于 0”。
否则
    输出字符串“变量 M 不等于 0”。

```

### 4.1.3 程序的 3 种控制结构

在编程的过程中，所有的操作都是在按照某种结构有条不紊地进行，纠其根源程序的控制结构大致可以分为 3 种：顺序结构、选择结构和循环结构。在对这 3 种结构的使用中，几乎很少有哪个程序是单独使用某一种结构来完成某个操作，基本上都是其中的两种或者 3 种结构结合使用。为了帮助读者更好地应用这 3 种程序控制结构，下面对其进行详细介绍。

#### 1. 顺序结构

顺序结构是最简单最基本的结构方式，各流程框依次按顺序执行。其传统流程图的表示方式与 N-S 结构化流程图的表示方式分别如图 4.1 和图 4.2 所示。其执行顺序为：开始→语句 1→语句 2→……→结束。

#### 2. 选择（分支）结构

选择结构就是对给定条件进行判断，条件为真时执行一个分支，条件为假时执行另一个分支。其传统流程图表示方式与 N-S 结构化流程图表示方式分别如图 4.3 和图 4.4 所示。



图 4.1 顺序结构传统流程图

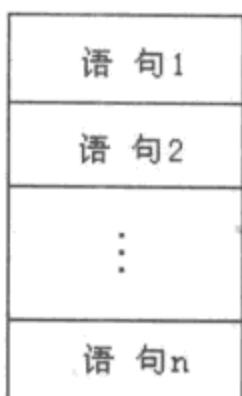


图 4.2 N-S 结构化流程图

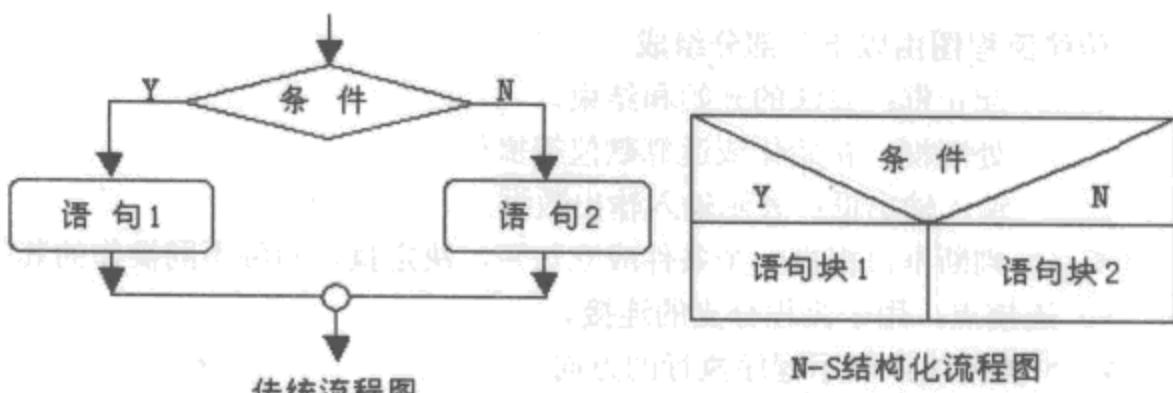


图 4.3 条件成立与否都执行语句或语句块

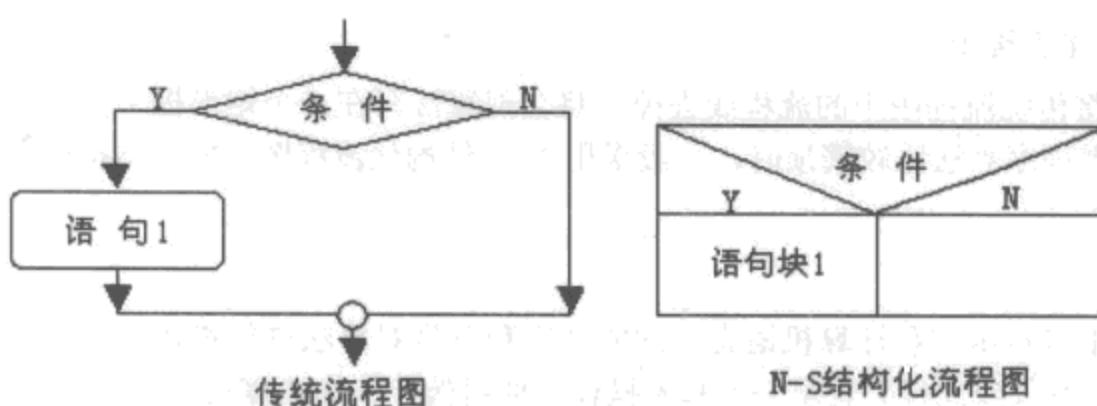


图 4.4 条件为否则不执行语句或语句块

### 3. 循环结构

循环结构可以根据需要多次重复执行一行或者多行代码。循环结构分为两种：前测试型循环和后测试型循环。

前测试型循环，先判断后执行。当条件为真时反复执行语句或语句块，条件为假时，跳出循环，继续执行循环后面的语句，流程图如图 4.5 所示。

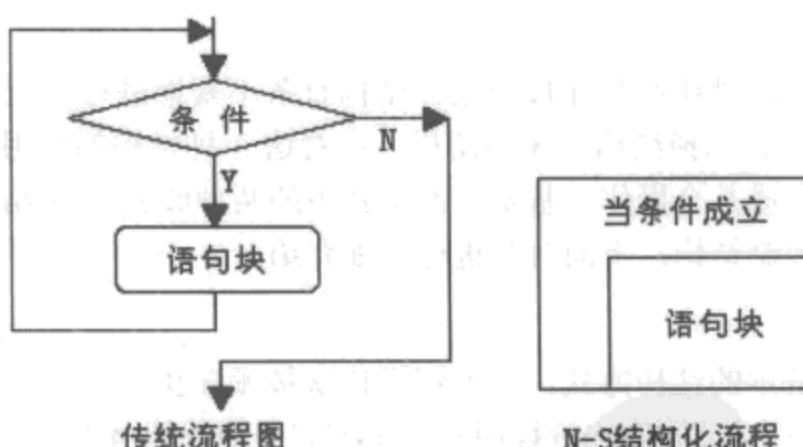


图 4.5 前测试型循环流程图

后测试型循环，先执行后判断。先执行语句或语句块，再进行条件判断，直到条件为假时，跳出循环，继续执行循环后面的语句，否则一直执行语句或语句块，流程图如图 4.6 所示。

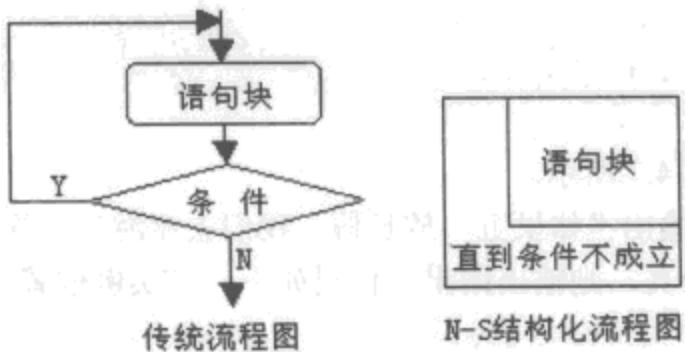


图 4.6 后测试型循环流程图

在 PHP 中，大多数情况下程序不会是简单的顺序结构，而是以这 3 种结构的组合形式出现。其中的顺序结构很容易理解，就是直接输出程序运行结果，而选择和循环结构则需要以下 3 种控制语句来实现。

- 条件控制语句：if、else、elseif 和 switch 语句。
- 循环控制语句：while、do…while、for 和 foreach 语句。
- 跳转控制语句：break 和 continue 语句。

在下面的章节中将对这 3 种控制语句进行详细的讲解。

## 4.2 条件控制语句

### ■ 视频讲解：光盘\TM\Video\4\条件控制语句.exe

条件控制语句就是以一定的条件作为依据，根据判断的结果确定只执行某一部分的代码，而与其并列的其他部分代码则不执行。

在 PHP 中，条件控制语句可分为以下两种类型：

- if 条件语句。
- switch…case 分支控制语句。

### 4.2.1 if 条件语句

#### 1. if 语句

if 语句是最简单的条件判定语句，它对某段程序的执行附加一个条件，如果条件成立，就执行这段程序；否则就跳过这段程序去执行下面的程序。

if 语句的格式为：

```
if(expr)
    statement;
```

如果表达式 expr 的值为 true，那么就顺序执行 statement 语句；否则，就跳过该条语句再往下执行。如果需要执行的语句不只一条，那么可以使用“{}”（在“{}”中的语句，被称之为语句组），格式为：

```
if(expr){
    statement1;
```

```
statement2;
...
}
```

if 条件语句的流程图如图 4.7 所示。

**例 4.1** 在本例中，实现输出“如果五一放长假，我们去旅游”。首先为变量赋一个逻辑值，然后判断这个值是否为真，如果为真，则输出结果，代码如下：（实例位置：光盘\TM\Instances\4\1）

```
<?php
$day_51=true; //为变量赋予一个逻辑值
if ($day_51==true){ //判断变量的逻辑值是否为真
    echo "五一放长假，我们去旅游"; //输出字符串
}
?>
```

运行结果如图 4.8 所示。

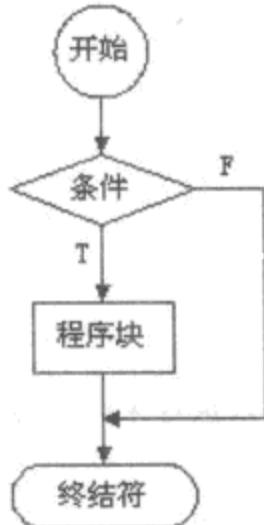


图 4.7 if 语句流程图

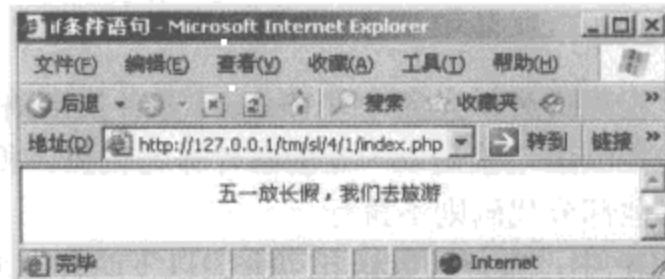


图 4.8 if 语句的应用

## 2. If...else 语句

大多数时候，总是需要在满足某个条件时执行一条语句，而在不满足该条件时执行其他语句。为了在 if 语句中描述这种情况，if 语句中提供了 else 子句，else 子句表示的自然语句是“否则”的意思，其语法格式为：

```
if(expr){
    statement1;
}else{
    statement2;
}
```

当表达式 expr 的值为 true 时，执行 statement1 语句；如果表达式 expr 的值为 false，则执行 statement2 语句。if...else 语句的流程图如图 4.9 所示。

**例 4.2** 在许多问题的逻辑关系中，经常会出现依据一个条件是否成立会导致两种不同的结果。例如，“如果五一放长假，我们去旅游，否则我们就去逛商场”。首先为变量赋一个逻辑值，然后判断这个值是否为 true，如果为 true，则输出“五一长假我们去旅游”；否则输出“五一短假我们去逛商场”，

根据不同的结果显示不同的字符串。实例代码如下：（实例位置：光盘\TM\Instances\4\2）

```
<?php
$day_51=false; //为变量赋予一个逻辑值
if ($day_51==true){ //判断变量的逻辑值是否为真
    echo "五一长假我们去旅游"; //输出字符串
}
else{
    echo "五一短假我们去逛商场"; //输出字符串
}
?>
```

结果为：五一短假我们去逛商场

### 3. 嵌套的 elseif 结构语句

使用 if 语句和 else 子句能够描述一些复杂的逻辑问题，但是有时并不能够完整地表达人们的语义。考虑这样一种情况，if…else 语句只能选择两种结果：要么执行真，要么执行假。但如果现在有两种以上的选择该怎么办呢？例如，小明的考试成绩，如果是 90 分以上，则为“优秀”；如果是 60~90 分之间，则为“良好”；如果低于 60 分，则为“不及格”。这时，可以使用 elseif（也可以写做 else if）语句来执行，该语法格式为：

```
if(expr1){
    statement 1;
}else if(expr2){
    statement 2;
}...
else{
    statement n;
}
```

elseif 语句的流程图如图 4.10 所示。

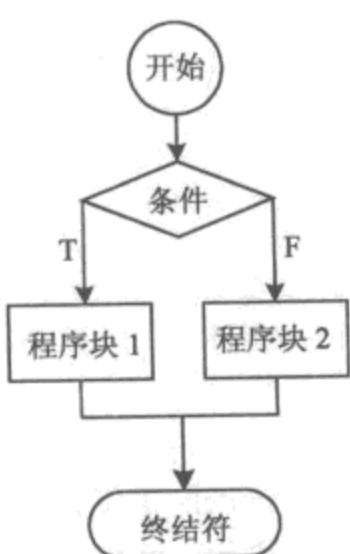


图 4.9 if...else 语句流程图

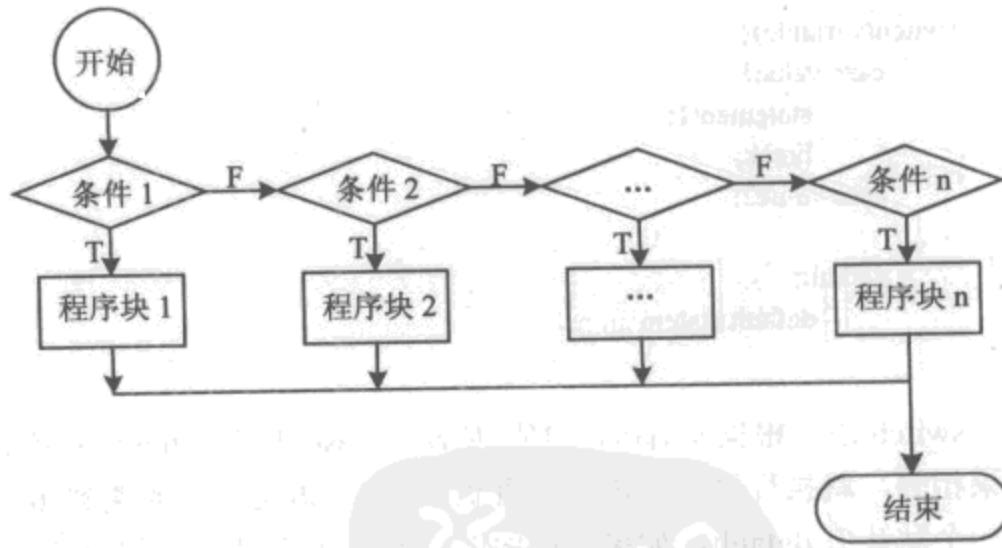


图 4.10 elseif 语句的流程图

**例 4.3** 本例通过 elseif 语句，判断小明的考试情况。如果成绩大于等于 90，为优秀；如果成绩大

于等于 80，则为良好；如果成绩大于等于 60，则为及格，否则为不及格。实例代码如下：（实例位置：光盘\TM\Instances\4\3）

```
<?php
    $score=95;
    if ($score >=90){
        echo "小明期末考试成绩优秀";
    }elseif($score<90 && $score>=80){
        echo "小明期末考试成绩良好";
    }elseif($score<80 && $score>=60){
        echo "小明期末考试成绩及格";
    }else{
        echo "小明期末考试成绩不及格";
    }
?>
```

//设置小明期末考试的默认值  
//判断小明的期末考试成绩是否在 90 分以上  
//如果是，说明小明期末考试成绩优秀  
//否则判断小明期末考试成绩是否在 80~90 之间  
//如果是，说明小明期末考试成绩良好  
//否则判断小明期末考试成绩是否在 60~80 之间  
//如果是，说明小明期末考试成绩及格  
//如果两个判断都是 false，则输出默认值  
//说明小明期末考试成绩不及格

结果为：小明期末考试成绩优秀

 注意：if 语句和 elseif 语句的执行条件是表达式的值为真，而 else 执行条件是表达式的值为假。这里的表达式的值不等于变量的值。

#### 4.2.2 switch…case 分支控制语句

##### 视频讲解：光盘\TM\Video\4\switch…case 分支控制语句.exe

在程序设计中，所有依据条件作出判定的问题，都可以用前面所介绍的不同类型的 if 语句来解决。不过，在用 if…else 语句处理多个条件的判定问题时，组成条件的表达式在每一个 elseif 语句中都要计算一次，显得繁琐臃肿。为了避免 if 语句的冗长，提高程序的可读性，可以使用 switch 分支控制语句。PHP 提供 switch…case 多分支控制语句，对于某些多项选择场合，会使程序代码更加简洁、易读。

switch 语句的语法格式如下：

```
switch(variable){
    case value1:
        statement1;
        break;
    case value2:
    ...
    default:
        default statement n;
}
```

switch 语句根据 variable 的值，依次与 case 中的 value 值相比较，如果不相等，继续查找下一个 case；如果相等，就执行对应的语句，直到 switch 语句结束或遇到 break 为止。一般 switch 语句的结尾都有一个默认值 default，如果在前面的 case 中没有找到相符合的条件，则输出默认语句，这和 else 语句类似。

switch 语句的流程图如图 4.11 所示。

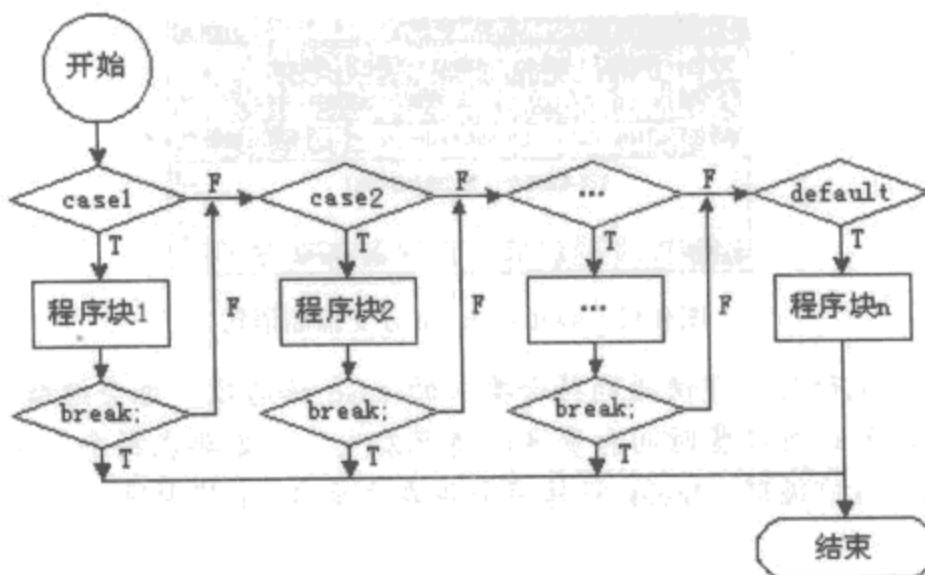


图 4.11 switch 语句流程图

**例 4.4** 本例将使用 switch 语句来输出当天为星期几，并根据不同的日期输出不同的贴心提醒警句，实例代码如下：（实例位置：光盘\TM\Instances\4\4）

```

<?php
setlocale(LC_TIME,"chs");           //设置本地环境
$weekday = strftime("%A");          //声明变量$weekday 的值
switch ($weekday){                  //switch 语句，判断$weekday 的值
    case "星期一":                 //如果变量的值为“星期一”
        echo "今天是$weekday , 新的一周开始了!";
        break;
    case "星期二":                 //如果变量的值为“星期二”
        echo "今天是$weekday , 时刻保持良好的工作状态!";
        break;
    case "星期三":                 //如果变量的值为“星期三”
        echo "今天是$weekday , 劳动者是最美的人，努力工作哟!";
        break;
    case "星期四":                 //如果变量的值为“星期四”
        echo "今天是$weekday , 勤奋才能创造绩效，加油! ) ";
        break;
    case "星期五":                 //如果变量的值为“星期五”
        echo "今天是$weekday , 一定要出色地完成本周工作哟!";
        break;
    case "星期六":                 //如果变量的值为“星期六”
        echo "今天是$weekday , 可以睡到自然醒!";
        break;
    default:                       //默认值
        echo "今天是$weekday , 呵呵，轻松地玩上一天!";
        break;
}
?>
  
```

运行结果如图 4.12 所示。

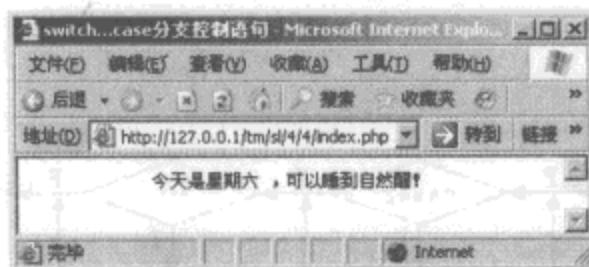


图 4.12 switch...case 分支控制语句

**◆ 注意：**switch 语句在执行时，即使遇到符合要求的 case 语句段，也会继续往下执行，直到 switch 结束。为了避免这种浪费时间和资源的情况发生，一定要在每个 case 语句段后添加 break 跳转语句跳出当前循环。break 跳转语句将在本章 4.4.1 节中进行详细介绍。

## 4.3 循环控制语句

在实际应用中，经常会遇到一些操作并不复杂，但需要反复多次处理的问题。对于这类问题，如果用顺序结构的程序来处理，将是十分繁琐的，有时候是难以实现的。为此，PHP 提供了循环语句来实现循环结构的程序设计。

循环控制语句是指能够按照一定的条件重复执行某段功能代码的代码结构。循环控制语句分为以下 4 种：

- while 循环语句。
- do...while 循环语句。
- for 循环语句。
- foreach 循环语句。

### 4.3.1 while 循环语句

#### ◆ 视频讲解：光盘\TM\Video\4\while 循环语句.exe

while 循环语句是 PHP 中最简单的循环控制语句。while 循环语句根据某一条件进行判断，决定是否执行循环，其语法格式如下：

```
while (expr){
    statement
}
```

当表达式 expr 的值为 true 时，将执行 statement 语句，执行结束后，再返回到 expr 表达式继续进行判断。直到表达式的值为 false，才跳出循环，执行大括号后面的语句。

while 循环语句的流程图如图 4.13 所示。



图 4.13 while 循环语句的流程图

**例 4.5** 应用 while 循环语句输出 3 个表格。首先初始化\$num 变量，赋初始值为 1；当\$num 等于 1 时，while 条件表达式的值为真，这时输出一个表格；然后将变量\$num 自加 1，再判断 while 条件表达式的值是否为真，如果为真，则继续执行循环体的内容，输出一个表格。直到\$num 变量大于 3 时，while 条件表达式的值为假，则退出循环。实例代码如下：（实例位置：光盘\TM\Instances\4\5）

```

<?php
$num = 1;
while($num <=3){
    echo"<table width='300' border='1' cellpadding='3' cellspacing='0' bordercolor='#FFCC33' bgcolor='#CCFF33>
<tr><td>编程词典上市了，抢购中...</td></tr></table>";
    $num++;
}
?>
  
```

运行结果如图 4.14 所示。

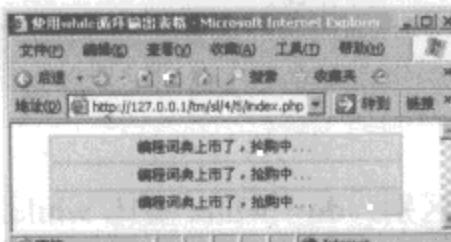


图 4.14 while 循环语句的应用

### 4.3.2 do…while 循环语句

#### ■ 视频讲解：光盘\TM\Video\4\do…while 循环语句.exe

while 语句还有另一种表示形式，do…while。do…while 循环语句和 while 循环语句非常相似，只是 do…while 循环语句在循环底部检测循环表达式，而不是在循环的顶部进行检测。

do…while 循环语句的语法格式如下：

```

do{
    statement
}while(expr);
  
```

`do...while` 语句先执行 statement 语句，然后再对表达式进行判断。如果表达式的值为 `false`，则跳出循环。因此应用 `do...while` 循环语句时该语句的循环体至少被执行一次。

`do...while` 语句的流程控制图如图 4.15 所示。

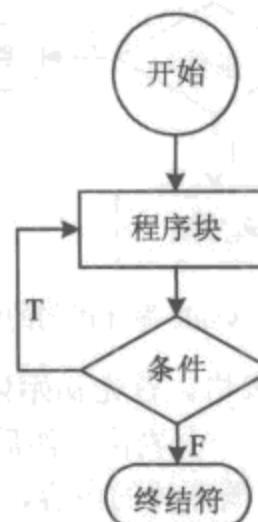


图 4.15 `do...while` 语句的控制流程图

**例 4.6** 通过两个语句进行对比，来比较 `while` 循环语句和 `do...while` 循环语句在应用上的不同点，代码如下：（实例位置：光盘\TM\Instances\4\6）

```

<?php
    $num = 1; // 声明一个整型变量$num
    while($num != 1){ // 使用 while 循环输出
        echo "你看不到我噢！";
    }
    do{ // 使用 do...while 循环输出
        echo "看到我了吧！";
    }while($num != 1); // 由于 while 表达式的条件是后判断的，因此该字符串至少输出一次
?>
  
```

结果为：看到我了吧！

从上面的代码中可以看出两者的区别：`do...while` 要比 `while` 语句多循环一次。当 `while` 表达式的值为假时，`while` 循环直接跳出当前循环，而 `do...while` 语句则是先执行一遍程序块，然后再对表达式进行判断。

**◆ 注意：** `do...while` 语句结尾处的 `while` 语句括号后面有一个分号“;”，为了养成良好的编程习惯，建议读者在书写的過程中不要将其遗漏。

### 4.3.3 for 循环语句

**◆ 视频讲解：**光盘\TM\Video\4\for 循环语句.exe

`for` 循环是 PHP 中最复杂的循环结构。`for` 循环语句能够按照已知的循环次数进行循环操作，主要应用于多条件情况下的循环操作。如果在单一条件下使用 `for` 循环语句就有些不合适。这一点从该语句的语法中就可以看出，其条件的表达式有 3 个。它的语法格式为：



```
for (expr1; expr2; expr3){
    statement;
}
```

其中, expr1 为变量初始赋值; expr2 为循环条件, 即在每次循环开始前求值。如果值为真, 则执行 statement; 否则, 跳出循环, 继续往下执行。expr3 为变量递增或递减, 即每次循环后被执行。for 循环语句的流程图如图 4.16 所示。

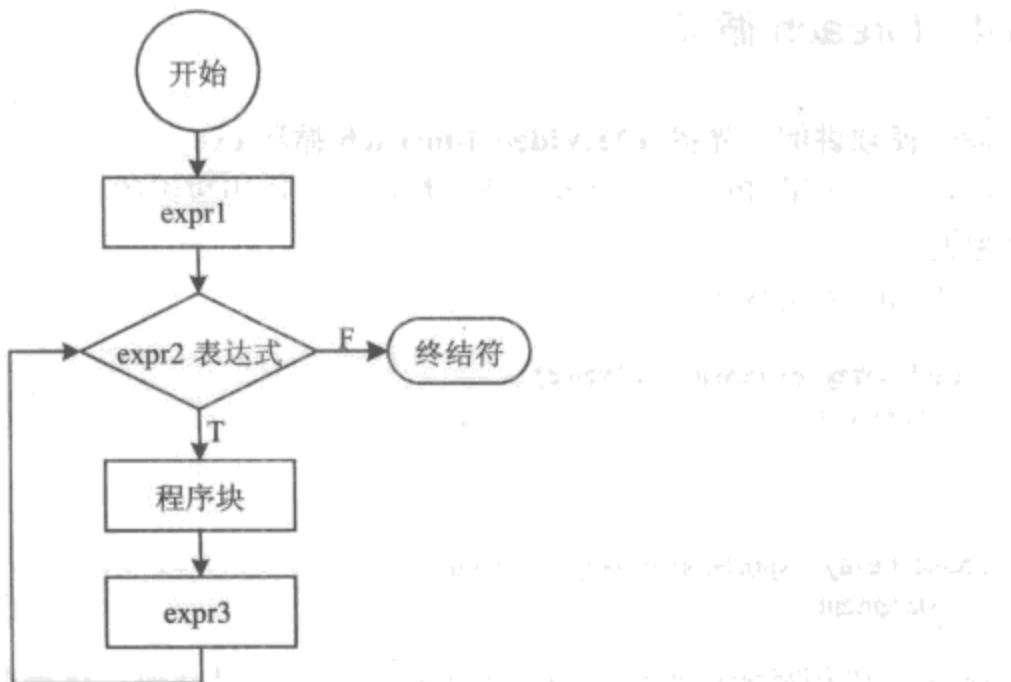


图 4.16 for 循环语句的流程图

#### 例 4.7 应用 for 循环计算 100 的阶乘, 代码如下: (实例位置: 光盘\TM\Instances\4\7)

```
<?php
$sum = 1; //声明整型变量$sum
for ($i = 1;$i <=100;$i++){
    $sum *= $i; //当$i≤100 时, 执行该表达式
}
echo "100! = ".$sum; //输出该表达式
?>
```

结果为: 100! = 9.33262154439E+157

在 for 语句中无论采用循环变量递增或递减的方式, 有一个前提, 就是一定要保证循环能够结束, 无期限的循环(死循环)将导致程序的崩溃。在循环变量自增的例子中, 循环的条件是 \$i ≤ 100, 由于在每次循环后 \$i 的值会加 1, 当 \$i 的值大于 100 时, 循环就结束了。

上面的代码采用了循环变量递增的方式。当然, 也可以采用倒序的方式, 以循环变量递减的方式编写程序。例如:

```
<?php
$sum = 1; //声明整型变量$sum
for ($i = 100;$i >0;$i--){
    $sum *= $i; //当$i>0 时, 执行该表达式
}
```

```
echo "100! = ".$sum; //输出该表达式
?>
```

结果为：100! = 9.33262154439E+157

在循环变量自减的例子中，循环条件是*\$i>0*，在每次循环后*\$i* 的值会减 1，当*\$i>0* 时，循环就结束了。

#### 4.3.4 foreach 循环

##### 视频讲解：光盘\TM\Video\4\foreach 循环.exe

`foreach` 循环在 PHP 4.0 中擅长处理数组，是遍历数组的一种简单方法。在 PHP 5.0 中，新增加了对对象的支持。

该语句的语法格式为：

```
foreach (array_expression as $value)
    statement
```

或

```
foreach (array_expression as $key => $value)
    statement
```

`foreach` 语句将遍历数组 `array_expression`，每次循环时，将当前数组中的值赋给 `$value`（或是 `$key` 和 `$value`），同时，数组指针向后移动，直到遍历结束。当使用 `foreach` 语句时，数组指针将自动被重置，所以不需要手动设置指针位置。

 注意：`foreach` 语句不支持用“@”来禁止错误信息。

例 4.8 `foreach` 语句的应用很广泛，其主要功能就是处理数组，下面就应用 `foreach` 语句来处理一个数组，实现输出购物车中商品的功能。这里假设将购物车中的商品存储于指定的数组中，然后通过 `foreach` 语句来输出购物车中的商品信息。程序代码如下：（实例位置：光盘\TM\Instances\4\8）

```
<?php
$name = array("1"=>"品牌笔记本电脑","2"=>"高档男士衬衫","3"=>"品牌 3G 手机","4"=>"高档女士挎包");
$price = array("1"=>"4998 元","2"=>"588 元","3"=>"4666 元","4"=>"698 元");
$counts = array("1"=>1,"2"=>1,"3"=>2,"4"=>1);
echo '<table width="580" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#FF0000">
<tr>
    <td width="145" align="center" bgcolor="#FFFFFF" class="STYLE1">商品名称</td>
    <td width="145" align="center" bgcolor="#FFFFFF" class="STYLE1">价格</td>
    <td width="145" align="center" bgcolor="#FFFFFF" class="STYLE1">数量</td>
    <td width="145" align="center" bgcolor="#FFFFFF" class="STYLE1">金额</td>
</tr>';
foreach($name as $key=>$value){ //以 book 数组做循环，输出键和值
echo '<tr>
    <td height="25" align="center" bgcolor="#FFFFFF" class="STYLE2">' . $value . '</td>
    <td align="center" bgcolor="#FFFFFF" class="STYLE2">' . $price[$key] . '</td>
```

```

</td align="center" bgcolor="#FFFFFF" class="STYLE2">.$counts[$key].</td>
</td align="center" bgcolor="#FFFFFF" class="STYLE2">.$counts[$key]*$price[$key].</td>
</tr>';
}
echo '</table>';
?>

```

运行结果如图 4.17 所示。

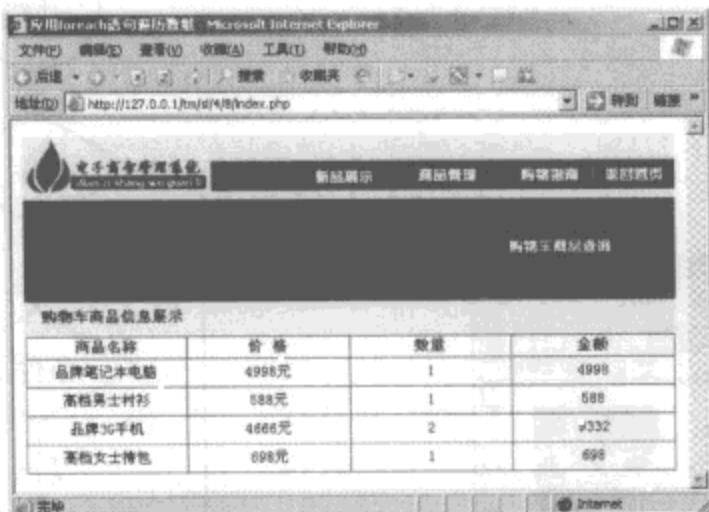


图 4.17 应用 foreach 语句输出购物车中的商品

技巧：当试图使 foreach 语句用于其他数据类型或者未初始化的变量时会产生错误。为了避免这个问题，最好使用 is\_array() 函数先来判断变量是否为数组类型，如果是，再进行其他操作。

## 4.4 跳转控制语句

### 4.4.1 使用 break 语句跳出循环

break 关键字可以终止当前的循环，包括 while、do…while、for、foreach 和 switch 在内的所有控制语句。

例 4.9 在本例中，应用 for 循环控制语句声明变量 \$i，循环输出 4 个表情头像，只有当变量 \$i 等于 4 时，才使用 break 跳转控制语句跳转 for 循环，代码如下：（实例位置：光盘\TM\Instances\4\9）

```

<?php
    for($i=1;$i<=4;$i++){
        //应用 for 循环控制语句输出表情头像
        if($i==4){
            //判断变量是否等于 4
            break; //如果等于 4，使用 break 语句跳转循环
        }
    ?>
    <input type="radio" name="head" value="<?php echo("images/".$i.".jpg");?>" />
    " width="90" height="90" id="head"/>
    <?php
    ?
?>

```

运行结果如图 4.18 所示。

**break** 语句不仅可以跳出当前的循环，还可以指定跳出几重循环。格式为：

```
break n;
```

参数 n 指定要跳出的循环数量。**break** 关键字的流程图如图 4.19 所示。



图 4.18 应用 **break** 跳转控制语句跳出循环

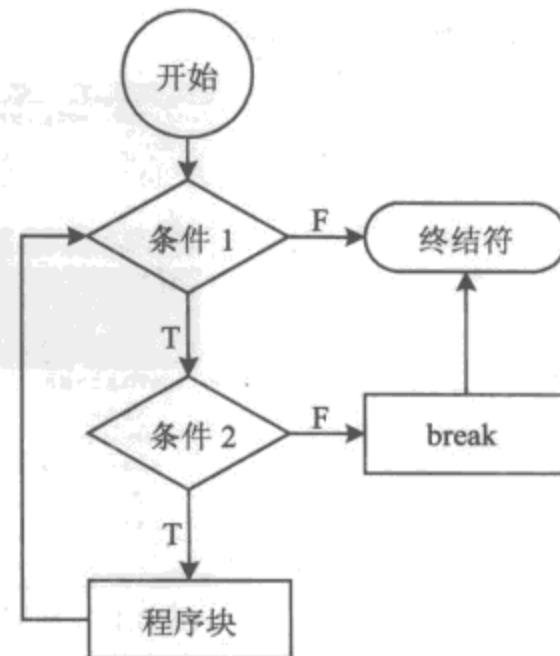


图 4.19 **break** 关键字的流程图

**例 4.10** 在本例中，共有 3 层循环，最外层的 while 循环和中间层的 for 循环是无限循环，最里面并列两个 for 循环：程序首先执行第一个 for 循环，当变量 \$i 等于 3 时，跳出当前循环（一重循环），继续执行第二个 for 循环，当第二个 for 循环中的变量 \$j 等于 4 时，将直接跳出最外层循环。实例代码如下：（实例位置：光盘\TM\Instances\4\10）

```

<?php
while(true){
    for(;;){
        for($i=1;$i<=4;$i++){
            echo $i." <img src='images/$i.jpg'>". "\t";
            if($i == 3){
                echo "<p>变量\$i 等于 3，跳出一重循环。</p>";
                break 1;
            }
        }
        for($j = 1; $j < 5; $j++){
            echo $j." <img src='images/$j.jpg'>". "\t";
            if($j == 4){
                echo "<p>变量\$j 等于 4，跳出最外层循环。</p>";
                break 3;
            }
        }
    }
    echo "看不到我吧，呵呵!";
}

```

```

}
?>

```

运行结果如图 4.20 所示。



图 4.20 使用 break 关键字跳出多重循环

#### 4.4.2 使用 continue 语句跳出循环

在程序执行 break 后，程序将跳出循环执行流程，而开始继续执行循环体的后续语句。continue 跳转语句的作用没有 break 那么强大，只能终止本次循环，而进入到下一次循环中。在执行 continue 语句后，程序将结束本次循环的执行，并开始下一轮循环的执行操作。continue 也可以指定跳出几重循环。continue 跳转语句的流程图如图 4.21 所示。

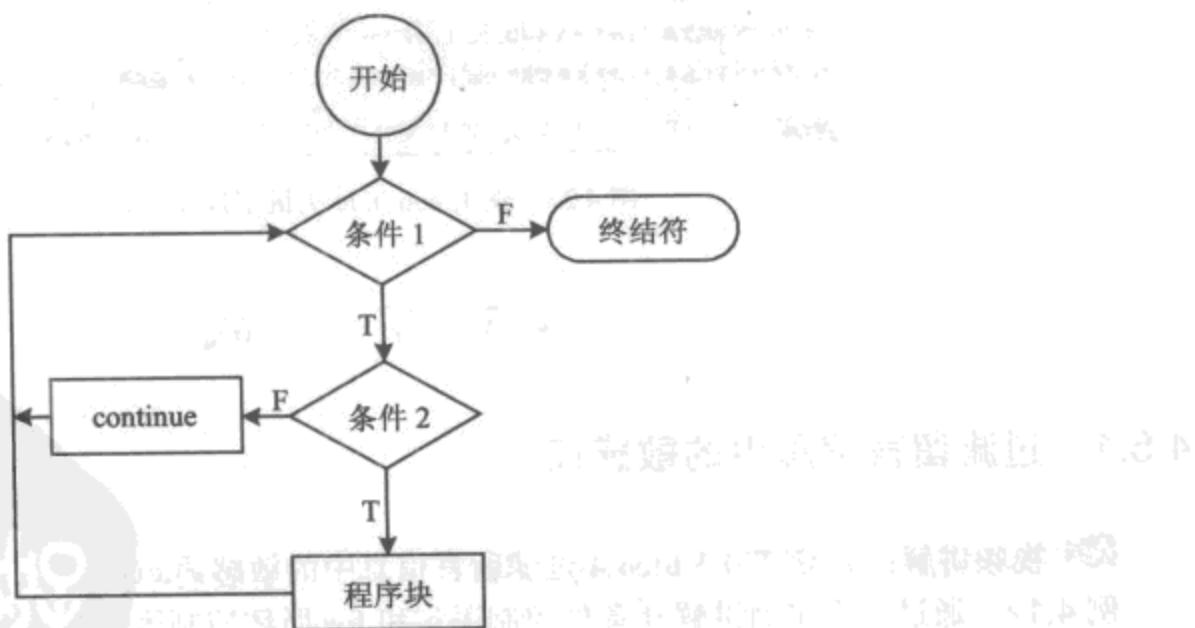


图 4.21 continue 跳转语句的流程图

**例 4.11** 本例将使用 for 循环输出数组变量。如果变量的数组下标为偶数，则只输出一个空行，如果是奇数，才继续输出。在最里面的循环中，判断当前数组下标是否等于 \$i，如果不相等，输出数

组变量，否则跳到最外重循环。代码如下：（实例位置：光盘\TM\Instances\4\11）

```
<?php
$arr = array("PHP 程序开发范例宝典","PHP 典型模块开发大全","PHP 函数参考大全","PHP 项目开发全程实录",
"PHP 从入门到精通","PHP 网络编程自学手册");           //声明一个数组变量$arr
for($i = 0; $i < 6; $i++){                                //使用 for 循环
    echo "<br>";
    if($i % 2 == 0){                                     //如果 $i 的值为偶数，则跳出本次循环
        continue;
    }
    for(;;){                                         //无限循环
        for($j = 0; $j < count($arr); $j++){          //再次使用 for 循环输出数组变量
            if($j == $i){                            //如果当前输出的数组下标等于 $i
                continue 3;                          //跳出最外重循环
            }else{
                echo $arr[$j]."\t|";                  //输出表达式
            }
        }
    }
    echo "你永远都看不到我噢!";
}
?>
```

运行结果如图 4.22 所示。

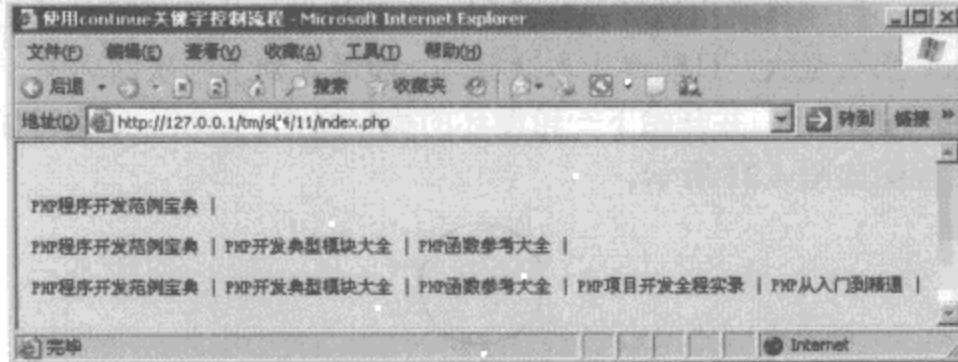


图 4.22 使用 continue 关键字控制流程

## 4.5 实战

### 4.5.1 过滤留言信息中的敏感词

视频讲解：光盘\TM\Video\4\过滤留言信息中的敏感词.exe

例 4.12 通过一个实例讲解 if 条件控制语句和 for 循环控制语句在 Web 程序中的应用。很多论坛、评论、贴吧都有过滤敏感词的功能。所谓的“敏感词”是指网站管理者为了正常经营发展、减少不必要的纷争、消除不良影响，而人为设定的一些较易引起人们歧义思维的词汇。本实例应用 file() 函数读取存储在文本文件中的敏感词汇，并将其存储在\$file\_word 数组中。然后，应用 for 循环语句自动读取

数组元素（敏感词），最后通过正则表达式 preg\_match() 函数检验用户提交的留言信息中是否含有敏感词。（实例位置：光盘\TM\Instances\4\12）

- (1) 应用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php。
- (2) 添加一个 form 表单、两个文本框、一个编辑框、一个提交按钮和一个重置按钮，并应用表格对表单元素进行合理的布局，程序代码如下。

```
<table width="761" border="0" align="center" cellpadding="0" cellspacing="0" bordercolor="#FEFEFE" bgcolor="#FFFFFF">
    <form action="" method="post" name="form1" id="form1">
        <tr>
            <td width="761" align="center" bgcolor="#F9F8EF"><table width="749" border="0" align="center" cellpadding="0" cellspacing="0" style="BORDER-COLLAPSE: collapse">
                <tr>
                    <td width="749" height="57" background="images/a_03.jpg">&nbsp;&nbsp;</td>
                </tr>
                <tr>
                    <td height="36" colspan="3" align="left" background="images/a_05.jpg" bgcolor="#F9F8EF" scope="col">姓 名：
                        <input name="user_name" id="user_name" value="匿名" maxlength="64" type="text" />
                        <span style="COLOR: #ff0000">*</span></td>
                    </tr>
                    <tr>
                        <td height="36" colspan="3" align="left" background="images/a_05.jpg" bgcolor="#F9F8EF" style="border-top: 1px solid black">标 题：
                            <input maxlength="64" size="30" name="title" type="text" />
                            <span style="COLOR: #ff0000">*</span></td>
                        </tr>
                        <tr>
                            <td height="126" colspan="3" align="left" background="images/a_05.jpg" bgcolor="#F9F8EF">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;内 容：
                                <textarea name="content" cols="60" rows="8" id="content" style="background:url(/images/mrbcccd.gif)"></textarea>
                                <span style="COLOR: #ff0000">*</span></td>
                            </tr>
                            <tr>
                                <td height="40" colspan="3" align="center" background="images/a_05.jpg" bgcolor="#F9F8EF">
                                    <input name="submit" type="submit" class="btn1" id="submit" value="签写留言"/>
                                    <input name="reset" type="reset" class="btn1" value="清除留言" />
                                </td>
                            </tr>
                        </table>
                    </td>
                </tr>
                <tr>
                    <td width="761" align="center" bgcolor="#F9F8EF">
                        <table width="749" border="0" align="center" cellpadding="0" cellspacing="0">
                            <tr>
                                <td width="703" height="31" align="center" background="images/a_07.jpg">&nbsp;</td>
                            </tr>
                        </table>
                    </td>
                </tr>
            </table>
        </form>
    </td>
</tr>
```

```

</table></td>
</tr>
</form>
</table>

```

(3) 当用户发表留言信息后, 提交留言信息时, 通过正则表达式 `preg_match()` 函数将留言信息与存储在数组中的敏感词进行对比, 检验用户提交的留言信息中是否含有敏感词。如果留言信息中含有敏感词, 那么将弹出提示信息, 否则留言信息发布成功。实现敏感词过滤的完整代码如下。

```

<?php
if($_POST){
    if(is_file("./filterwords.txt")){
        $filter_word = file("./filterwords.txt");
        //判断给定文件名是否为一个正常的文件
        $str=$_POST['content'];
        for($i=0;$i<count($filter_word);$i++){
            //把整个文件读入一个数组中
            if(preg_match("/".trim($filter_word[$i])."/i",$str)){
                //应用 for 循环语句对敏感词进行判断
                //应用正则表达式, 判断传递的留言信息中
                //是否含有敏感词
                echo "<script> alert('留言信息中包含敏感词! ');history.back(-1);</script>";
                exit;
            }
        }
    }
}
?>

```

技巧: `preg_match()` 函数中的 “/i”, 是指在进行敏感词汇比较时不区分字母大小写。

在 IE 浏览器中输入地址, 按 Enter 键, 在用户提交留言信息后, 系统将签写的留言信息与本模块设定的敏感词进行一一对比, 如果在用户签写的留言信息中存在敏感词, 如“喜欢”, 就会弹出留言中含有敏感词的警告信息, 效果如图 4.23 所示。

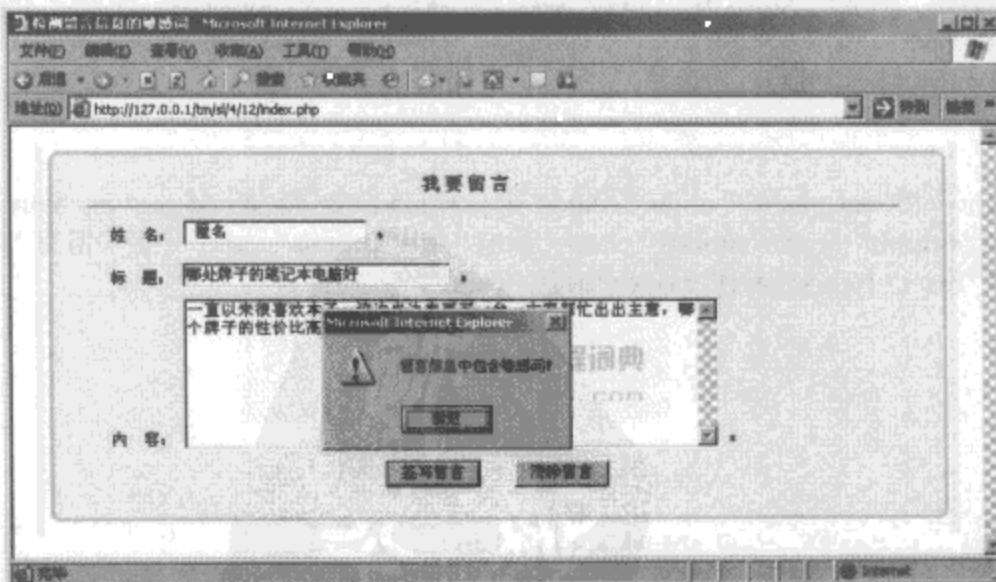


图 4.23 过滤敏感词“喜欢”

## 4.5.2 图书信息显示

### 视频讲解：光盘\TM\Video\4\图书信息显示.exe

**例 4.13** 在本实例中，综合了前面各范例中介绍的有关表单中的各个元素，来实现各个表单元素的综合应用。本实例主要应用表单元素实现一个发布公告信息的功能，通过将表单信息提交到数据处理页获取表单元素的值，实现查看公告信息的功能。（实例位置：光盘\TM\Instances\4\13）

- (1) 应用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php。
- (2) 在页面中，应用 do…while 循环控制语句输出图书信息表中的图书信息。程序代码如下。

```
<table width="200" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
    <td></td>
</tr>
</table>
<table width="703" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
    <td bgcolor="#EF5E0F" >
        <table width="703" border="0" align="center" cellpadding="0" cellspacing="1">
            <tr>
                <td height="25" colspan="5" bgcolor="#FFFFFF">&ampnbsp</td>
            </tr>
            <tr>
                <td width="187" height="25" bgcolor="#F39A40"><div align="center">书名</div></td>
                <td width="173" bgcolor="#F39A40"><div align="center">作者</div></td>
                <td width="101" bgcolor="#F39A40"><div align="center">价格</div></td>
                <td width="131" bgcolor="#F39A40"><div align="center">出版社</div></td>
                <td width="105" bgcolor="#F39A40"><div align="center">出版时间</div></td>
            </tr>
        </table>
    <?php
        $conn=mysql_connect("localhost","root","root");
        mysql_select_db("db_books",$conn);
        mysql_query("set names gb2312");
        include_once("conn.php");
        $sql=mysql_query("select * from tb_book order by id desc",$conn);
        $info=mysql_fetch_array($sql);
        if($info==false) {
            echo "暂无图书信息！";
        }
        else{
            do {
                ?>
            <tr>
                <td height="25" ><div align="center"><?php echo $info[book_name];?></div></td>
                <td height="25" ><div align="center"><?php echo $info[book_author];?></div></td>
                <td height="25" ><div align="center"><?php echo $info[book_price];?></div></td>
                <td height="25" ><div align="center"><?php echo $info[book_publish];?></div></td>
            </tr>
        } while($info=$sql);
    }
    </?php>
</td>
</tr>
</table>
```

```

<td height="25" ><div align="center"><?php echo $info[book_date];?></div></td>
</tr>
<?php
    }while($info=mysql_fetch_array($sql));
}
?>
</table>
</td>
</tr>
</table>

```

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 4.24 所示。

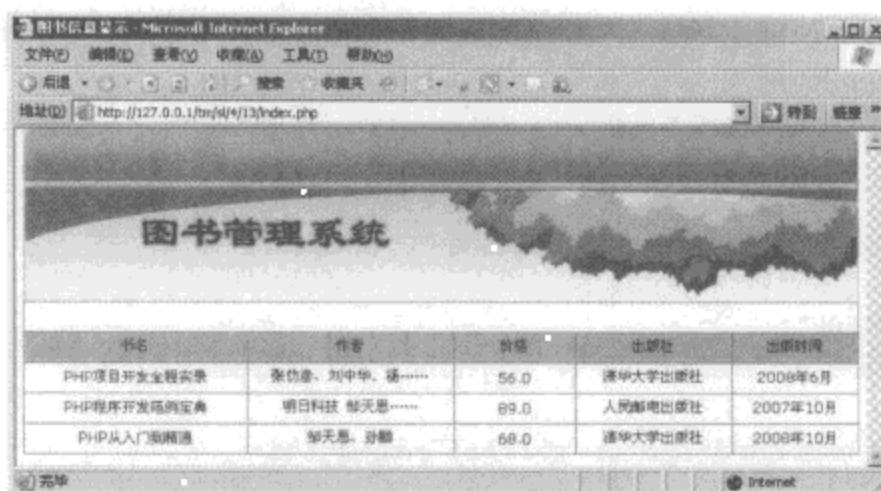


图 4.24 图书信息显示

**注意：**本实例涉及从 MySQL 数据库中读取数据的操作，数据库方面的相关知识将在第 19 章进行详细介绍。

### 4.5.3 应用 switch 语句设计网站主页

**视频讲解：**光盘\TM\Video\4\应用 switch 语句设计网站主页.exe

**例 4.14** 在本实例中，应用 switch 语句设计网站主页的布局，将网站头、尾文件设置为固定不变的版块，导航条也作为固定版块；而在主显示区中，应用 switch 语句，根据超链接中传递的值不同，显示不同的内容，代码如下。

```

<table width="180" height="523" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#F1F1F1">
    <tr>
        <td height="25" align="center" bgcolor="#FFFFFF"><span class="STYLE4">&ampnbsp&ampnbsp<a href="index.php?lmbs=最新商品">&ampnbsp最新商品</a></span></td>
    </tr>
    <tr>
        <td height="25" align="center" bgcolor="#FFFFFF"><span class="STYLE4">&ampnbsp&ampnbsp<a href="index.php?lmbs=热门商品">&ampnbsp热门商品</a></span></td>
    </tr>
    <tr>

```

```

<td height="25" align="center" bgcolor="#FFFFFF"><span class="STYLE4">&nbsp;&nbsp;&nbsp;<a href="index.php?lmbs=<?php echo urlencode("推荐商品");?>">推荐商品</a></span></td>
</tr>
<tr>
    <td height="25" align="center" bgcolor="#FFFFFF"><span class="STYLE4">&nbsp;&nbsp;&nbsp;<a href="index.php?lmbs=<?php echo urlencode("特殊商品");?>">特殊商品</a></span></td>
</tr>
</table>

<?php
switch($_GET[lmbs]){
    case "最新商品": //获取超链接传递的变量
        //如果变量的值等于“最新商品”
        include "index1.php";
        break; //否则跳出循环
    case "热门商品":
        include "index2.php";
        break;
    case "推荐商品":
        include "index3.php";
        break;
    case "特殊商品":
        include "index4.php";
        break;
    case "":
        //当该值为空时，执行下面的语句
        include "index4.php";
        break;
}
?>

```

运行本实例，单击栏目导航条中的超链接，将在主显示区中输出对应的项目内容，如图 4.25 所示。



图 4.25 应用 switch 语句创建主页框架

## 4.6 本章小结

本章主要讲述的是流程控制语句的知识，并且对算法和程序的控制结构进行了介绍。重点掌握 3 种流程控制语句——条件控制语句、循环控制语句和跳转控制语句；在掌握流程控制语句的基础上，对程序的算法和控制结构应该有所了解，读者通过对本章的学习能够从宏观的角度去认识 PHP 语言，从整体上形成一个开发的思路，逐渐形成一种属于自己的编程思想和编程方法。

## 4.7 实战练习

1. 应用 switch…case 分支控制语句输出本年度当前月份的活动安排。（答案位置：光盘\TM\Instances\4\15）
2. 在开发动态网站时应用 do…while 循环语句输出企业公告信息。（答案位置：光盘\TM\Instances\4\16）

# 第 5 章

## 函数

( 视频讲解：75分钟)

在日常开发中，如果有一个功能或者一段代码要经常使用，则可以把它写成自定义函数，在需要的时候进行调用。函数分为系统内部函数和用户自定义函数两种。除了自定义函数外，PHP 还提供了庞大的函数库，足有几千种的内置函数，可以直接使用来轻松实现相应功能。在程序中调用函数的目的，是为了简化编程的负担，减少代码量和提高效率，达到增加代码重用性、避免重复开发的目的。

通过阅读本章，可以学习到：

- 自定义函数和调用函数的方法
- 函数的嵌套调用和递归调用
- PHP 的输出语句及各语句的区别
- PHP 文件包含语句及各语句的区别
- 在程序中编写并调用自定义函数



## 5.1 自定义函数

在开发过程中，经常要多次重复某种操作或处理，如数据查询、字符操作等。如果每个模块的操作都要重新输入一次代码，不仅加大了程序员的工作量和开发时间，而且对于代码的后期维护及运行效果也有着较大的影响。使用自定义函数是解决该问题最有效的方法。可以把一段可以实现指定功能的代码封装在函数内。函数将 PHP 程序中繁琐的代码模块化，使程序员不需再频繁地编写相同的代码，只要直接调用函数即可实现指定的功能。函数的应用不但可以提高代码的可靠性，而且也提高了代码的可读性，更提高了程序员的工作效率并节省了开发时间。

### 5.1.1 函数的命名规则

PHP 函数的命名规则如下：

- ☒ 函数的命名是程序规划的核心。名字就是事物在它所处的生态环境中一个长久而深远的结果。变量及函数的命名是以能表达变量或函数的动作意义为原则的，一般是由动词开头，采用大小写混合的方式，第一个单词的首字母小写，其后单词的首字母大写。

```
function run();
function runFast();
function getBackground();
```

- ☒ 函数名称不区分大小写。例如，name()和 NAME()指向的是同一个函数，这一点读者一定要注意。如果读者误定义了两个不同大小写的重名函数，程序将中止运行。
- ☒ 函数的参数没有限制，可以定义任意需要的参数数量，也可以无参数值。
- ☒ 名称的开头不能使用数字及特殊符号。
- ☒ “.” 及类型声明等专用语不能作为名称。
- ☒ 变量或程序名的长度必须在 255 个字符以内。
- ☒ 和保留字相同的名称不能使用。
- ☒ 另外，还有一些函数命名的通用规则。例如，取数，则用 Get 开头，然后跟上要取的对象的名字；设置数，则用 Set 开头，然后跟上要设的对象的名字，如 GetXxx 或 SetXxx。

### 5.1.2 自定义函数格式

在程序开发的过程中，最高效的方法就是将某些特定的功能定义成一个函数写在一个独立的代码块中，在需要的时候单独调用。创建函数的基本语法格式为：

```
function function_name ([$arg_1],[ $arg_2], ... ,[ $arg_n]){
    fun_body;
    [return arg_n];
}
```

参数说明如下。

- ☒ **function:** 声明自定义函数时必须使用到的关键字。
- ☒ **function\_name:** 创建函数的名称，是有效的 PHP 标识符。函数名称是唯一的，其命名遵守与变量命名相同的规则，只是不能以“\$”开头。
- ☒ **arg\_1…arg\_n:** 是外界传递给函数的值，可有可无，可以有多个参数，数量根据需要而定，各参数用逗号“,”分隔，参数的类型不必指定，在调用函数时只要是 PHP 支持的类型都可以使用。
- ☒ **fun\_body:** 为自定义函数的主体，是功能实现部分。
- ☒ **return:** 将调用的代码需要的值返回，并结束函数的运行。

**注意：** 函数名称是不区分大小写的，而常量和变量的名称区分大小写。

### 5.1.3 调用自定义函数

当函数定义完成后，所要做的就是调用这个函数。调用函数的操作十分简单，只需要引用函数名并赋予正确的参数即可完成函数的调用。

**例 5.1** 在本例中定义了一个函数 example()，计算传入的参数的平方，连同表达式和结果输出到浏览器，代码如下：（实例位置：光盘\TM\Instances\5\1）

```
<?php
    /* 声明自定义函数 */
    function example($num){
        return "$num * $num = " . $num * $num; // 返回计算后的结果
    }
    echo example(10); // 调用函数
?>
```

结果为：10 \* 10 = 100

### 5.1.4 函数的参数

在调用函数时，需要向函数传递参数，被传入的参数称为实参，而函数定义的参数为形参。参数传递的方式有按值传递、按引用传递和默认参数 3 种。

#### 1. 按值传递方式

将实参的值复制到对应的形参中，在函数内部的操作针对形参进行，操作的结果不会影响到实参，即函数返回后实参的值不会改变。

**例 5.2** 在本例中，首先定义一个函数 example()，功能是将传入的参数值做一些运算后再输出。接着在函数外部定义一个变量 \$m，也就是要传入的参数。最后调用函数 example(\$m)，输出函数的返回值 \$m 和变量 \$m 的值。实例代码如下：（实例位置：光盘\TM\Instances\5\2）

```
<?php
function example( $m ){ // 定义一个函数
    $m = $m * 5 + 10;
}
```

```

echo "在函数内: \$m = \"$m\";           //输出形参的值
}
$m = 1;
example( $m );
echo "<p>在函数外: \$m = $m </p>";    //传值: 将 $m 的值传递给形参 $m
                                         //实参的值没有发生变化, 输出 $m=1
?>

```

结果为,

在函数内: \$m = 15  
在函数外: \$m = 1

## 2. 按引用传递方式

按引用传递就是将实参的内存地址传递到形参中。这时, 在函数内部的所有操作都会影响到实参的值, 返回后, 实参的值会发生变化。引用传递方式就是在传值时在原基础上加&符号。

**例 5.3** 仍然使用上例中代码, 唯一不同的就是多了一个&符号。实例代码如下: (实例位置: 光盘\TM\Instances\5\3)

```

<?php
function example( &$m ){           //定义一个函数, 同时传递参数 $m 的变量
    $m = $m * 5 + 10;
    echo "在函数内: \$m = \"$m\";           //输出形参的值
}
$m = 1;
example( $m );                   //传值: 将 $m 的值传递给形参 $m
echo "<p>在函数外: \$m = $m </p>"; //实参的值发生变化, 输出 m=15
?>

```

结果为,

在函数内: \$m = 15  
在函数外: \$m = 15

## 3. 默认参数(可选参数)

还有一种设置参数的方式, 即可选参数。可以指定某个参数为可选参数, 将可选参数放在参数列表末尾, 并且指定其默认值为空。

**例 5.4** 应用可选参数实现一个简单的价格计算功能, 设置自定义函数 values 的参数 \$tax 为可选参数, 其默认值为空。第一次调用该函数, 并且给参数 \$tax 赋值 0.25, 输出价格; 第二次调用该函数, 不给参数 \$tax 赋值, 输出价格。实例代码如下: (实例位置: 光盘\TM\Instances\5\4)

```

<?php
function values($price,$tax=""){
    $price=$price+($price*$tax); //声明一个变量 $price, 等于两个参数的运算结果
    echo "价格:$price<br>"; //输出价格
}
values(100,0.25); //为可选参数赋值 0.25
values(100); //没有给可选参数赋值
?>

```

结果为，

价格：125  价格：100

**► 注意：**当使用默认参数时，默认参数必须放在非默认参数的右侧；否则，函数将可能出错。

**► 说明：**从 PHP 5.0 开始，默认值也可以通过引用传递。

### 5.1.5 函数返回值

通常函数将返回值传递给调用者的方式是使用关键字 `return` 语句，`return` 语句是可选的。`return` 语句将函数的值返回给函数的调用者，即将程序控制权返回到调用者的作用域。注意，如果在全局作用域内使用 `return` 语句，那么将终止脚本的执行。`return` 后紧跟要返回的值，可以是变量、常量、数组或表达式等。

**例 5.5 应用 `return()` 函数返回一个操作数。**先定义函数 `values`，函数的作用是输入物品的单价、重量，然后计算总金额，最后输出商品的价格。实例代码如下：（实例位置：光盘\TM\Instances\5\5）

```
<?php
    function values($price,$tax=0.65){          // 定义一个函数，函数中的一个参数有默认值
        $price=$price+($price*$tax);           // 计算物品金额
        return $price;                         // 返回金额
    }
    echo values(100);                         // 调用函数
?>
```

结果为：165

`return` 语句只能返回一个参数，就是说只能返回一个值，不能一次返回多个。当要返回多个值时，可以在函数中定义一个数组，将多个值存储在数组中，然后用 `return` 语句将数组返回。

### 5.1.6 嵌套调用

所谓嵌套函数就是在函数中定义并调用其他函数。嵌套调用可以将一个复杂的功能分解成多个子函数，再通过调用的方式结合起来，有利于提高函数的可读性。

**例 5.6 应用嵌套函数，将获取的价格数由美元转换为人民币，程序代码如下：**（实例位置：光盘\TM\Instances\5\6）

```
<?php
function example($price,$tax){
    function examples($yuan,$taxs=7.5){
        return $yuan*$taxs;
    }
    $total=$price+($price*$tax);
    echo "价格是:$total 美元<br>";
    echo "价格是:".examples($total)."元<br>";
}
```

```
example(15.00,0.75);
?>
```

结果为，

价格是：26.25 美元 价格是：196.875 元

### 5.1.7 递归调用

PHP 的函数对自身的调用称为递归调用。在使用递归调用时，需要注意结束递归的条件，如果设置得不合适，很可能会造成死循环（无限循环）。

**例 5.7 应用递归函数将一个字符串颠倒输出和按照每个数是前面两个数的和的规律输出数据。程序代码如下：**（实例位置：光盘\TM\Instances\5\7）

```
<?php
echo "颠倒输出字符串:";
function example($str){
    if (strlen($str)>0)
        example (substr($str,1));
    echo substr($str,0,1);
    return;
}
example('pankaihua');
echo "<br>按照指定的规律输出数据:";
function num($i){
    if ($i==0)
        return 0;
    if ($i==1)
        return 1;
    return num($i-1)+num($i-2);
}
for($i=0;$i<20;$i++){
    echo num($i)."*";
}
?>
```

结果为，

颠倒输出字符串：auhiaknap

按照指定的规律输出数据：0\*1\*1\*2\*3\*5\*8\*13\*21\*34\*55\*89\*144\*233\*377\*610\*987\*1597\*2584\*4181\*

递归函数就是调用自身的函数。递归函数与循环实现的是相同的功能，相对而言递归函数比循环慢而且要占用更多的内存，所以不推荐使用递归函数。

## 5.2 内建函数

PHP 提供了很多内建函数，这些函数可以在程序中直接使用，按照其实现的功能划分为很多个函数库。另外，还可以通过扩展模块的方式引入更多的其他函数。内建函数库非常庞大，函数非常多，

足有上千种，在后面的章节中会陆续介绍常用的函数。熟练掌握自定义函数和内建函数，可以极大地提高编程人员的开发效率，迅速提高编程者的水平。

## 5.3 输出语句

### ▶ 视频讲解：光盘\TM\Video\5\输出语句.exe

程序设计中经常需要将字符串或者字符串变量输出到浏览器。本节将介绍几种输出语句的方法。由于 PHP 中数据类型是可以自动转换的，所以这些方法也可用于输出其他类型的数据。

### 5.3.1 应用 print 语句输出字符

print 输出语句的作用是将字符串输出到浏览器或打印机等输出设备，通常输出到浏览器。print 语句只可以同时输出一个字符串，需要圆括号。echo 语句可以同时输出多个字符串，并不需要圆括号。print 语句和 echo 语句的功能是相同的，但 print 语句有返回值，而 echo 语句没有返回值，所以 echo 语句相对可能快些，但这种速度差异并不明显。

语法：

```
int print ( string arg)
```

该语句执行成功则返回 1，失败则返回 0。

◀ 注意：使用 print 语句时，括号可以省略。

**例 5.8 应用 print 语句输出字符串“PHP 编程词典，让您编程无忧”，代码如下：（实例位置：光盘\TM\Instances\5\8）**

```
<?php  
print ("PHP 编程词典，让您编程无忧");  
?>
```

运行结果如图 5.1 所示。



图 5.1 应用 print 语句输出字符串

**例 5.9 上面讲解了应用 print 语句输出字符串的方法，接下来应用 print 语句输出变量中的字符串，**

代码如下：（实例位置：光盘\TM\Instances\5\9）

```
<?php
$str="打造明日科技品牌图书";
print $str;
?>
```

运行结果如图 5.2 所示。



图 5.2 应用 print 语句输出变量中的字符串

例 5.10 应用 print 语句输出新型字符串中的值，代码如下：（实例位置：光盘\TM\Instances\5\10）

```
<?php
print <<<END
明日科技 PHP 图书开发团队
END;
?>
```

运行结果如图 5.3 所示。

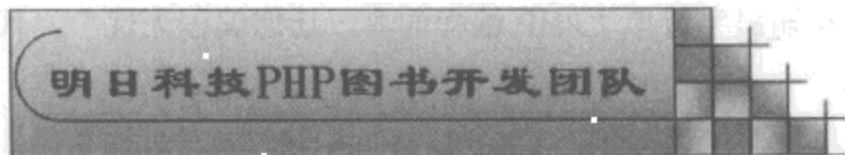


图 5.3 应用 print 语句输出新型字符串中的值

注意：由于这是一个语言结构而非函数，因此它无法被变量函数调用。

### 5.3.2 应用 echo 语句输出字符

echo 语句用于在脚本当前位置输出字符串，该语句在前面章节所举的实例中多次使用过，是程序设计中最常使用的一种输出方法。echo 使用方式非常简单，直接将要输出的字符串或者变量跟在 echo 后面，字符串或变量和 echo 之间以一个空格分隔。

语法：

```
echo "string arg1, string [argn]..."
```

echo 语句会将传入的字符串参数（arg1）进行输出。由于它本身并不是一个真正的函数，因此没有返回值。

**注意：**在使用 echo 语句时，应该时刻牢记语句结束处的分号，在 PHP 中，分号是用来分隔语句的。丢失这个“;”是编程人员经常犯的语法错误。同样也是读者在进行程序的错误处理时，首先应该注意的地方。

使用 echo 语句执行的结果是，将传递给其自身的字符串输出到浏览器。下面应用 echo 语句输出字符串“明日科技图书网：www.mingribook.com”到浏览器，代码如下。

```
<?php
echo "明日科技图书网：www.mingribook.com";
?>
```

结果为，

明日科技图书网：www.mingribook.com

接下来应用 echo 语句输出变量，代码如下。

```
<?php
$string = "明日编程词典网：www.mrbccd.com";
echo $string;
?>
```

结果为，

明日编程词典网：www.mrbccd.com

echo 语句后面可以连续输出多个字符串，它们之间以逗号分隔。例如：

```
<?php
$mr="明日科技";
$str="是一家以计算机软件技术为核心的高科技型企业";
echo $mr,$str;
?>
```

结果为：明日科技是一家以计算机软件技术为核心的高科技型企业

连续输出多个字符串，可以使用字符串运算符“.”将多个字符串连接成一个新型字符串输出。例如：

```
<?php
$str1="一个人的快乐";
$str2="不是因为他拥有的多";
$str3="而是因为他计较的少。";
echo $str1." ".$str2." ".$str3." ";
?>
```

结果为：一个人的快乐，不是因为他拥有的多，而是因为他计较的少。

echo 语句后紧跟的参数应该是字符串类型，如果不是，PHP 将自动将其进行转换，所以 echo 语句同样可以输出其他类型的数据。例如：

```
<?php
$name="纯净水";
$age="26";
$payfor="2398.65";
```

```
<?php
echo $name." 年龄: ".$age." 工资: ".$payfor;
?>
```

结果为，

纯净水 年龄: 26 工资: 2398.65

**例 5.11 应用 echo 语句直接输出 phpinfo() 函数调用的内容，代码如下：（实例位置：光盘\TM\Instances\5\11）**

```
<?php
echo phpinfo();
?>
```

运行结果如图 5.4 所示。

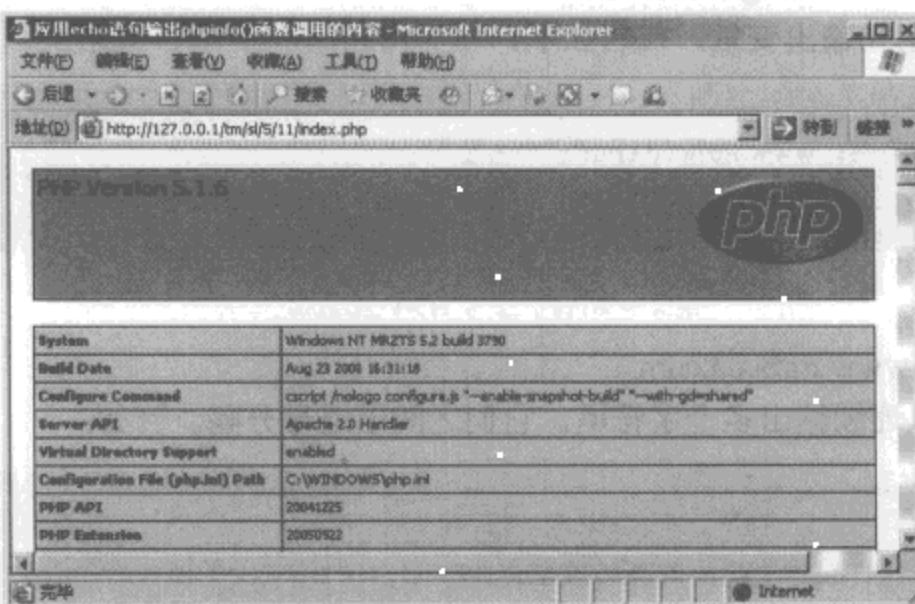


图 5.4 应用 echo 语句直接输出函数中的内容

### 5.3.3 应用 printf 语句格式化输出字符

printf 语句将字符串按照某种格式进行输出。

语法：

```
int printf( string format [, mixed args [, mixed ...]] )
```

printf 语句按照参数 format 指定的内容格式对字符串进行格式化，具体参数 format 的转换格式是以百分比符号“%”开始到转换字符为止。参数 format 的格式转换类型如表 5.1 所示。

表 5.1 参数 format 的格式转换类型

参数	说明
b	整数转换成二进制
c	整数转换成对应的 ASCII 字符
d	整数转换成十进制
e	将整数转换成以科学计数法显示

续表

参数	说 明
f	倍精确度数字转换成浮点数
o	整数转换成八进制
s	整数转换成字符串
u	整数转换成无符号整数
x	整数转换成小写十六进制
X	整数转换成大写十六进制

**例 5.12 应用 printf 语句将整数转换成不同的类型，并以格式化的方式输出字符串，代码如下：**  
**(实例位置：光盘\TM\Instances\5\12)**

```
<?php
$string = 17;
printf("%c",$string); //将整数转换成 ASCII 字符
$string = "明日科技";
printf($string); //直接输出字符
$string = 1234.00;
printf("%0.2f",$string); //将整数转换成小数点后有两位小数的浮点数
printf("%s",$string); //将整数转换成字符串
printf("%X",$string); //将整数转换成大写十六进制
?>
```

运行结果为：◆明日科技 1234.0012344D2

### 5.3.4 应用 sprintf 语句格式化输出字符串

格式化字符串用于按一定的格式输出含有许多变量的文本，是最常用的一种操作，可以使用 PHP 的 sprintf 语句来完成格式化字符串的功能。

语法：

```
string sprintf(string format, mixed args...)
```

参数 format 为转换后的格式，各个变量都以“%”后的字符规定其格式，格式转换类型如表 5.1 所示。

**例 5.13 应用 sprintf 语句格式化字符串，代码如下：**  
**(实例位置：光盘\TM\Instances\5\13)**

```
<?php
$name= "纯净水";
$pay= 2300;
$expend= 154.5;
$balance= $pay-$expend;
echo sprintf("%s:您的本月工资为￥%0.1f 元",$name,$balance)
?>
```

运行结果为：

纯净水：您的本月工资为¥2145.5 元

## 5.4 引用文件

### ■ 视频讲解：光盘\TM\Video\5\引用文件.exe

引用文件是指将另一个源文件的全部内容包含到当前源文件中进行使用。引用外部文件可以减少代码的重用性，是 PHP 编程的重要技巧。PHP 提供了语句 `include` 语句、`require` 语句、`include_once` 语句和 `require_once` 语句用于实现引用文件。这 4 种语句在使用上有一定的区别。下面分别进行详细讲解。

### 5.4.1 应用 `include` 语句引用文件

使用 `include` 语句引用外部文件时，只有代码执行到 `include` 语句时才将外部文件引用进来并读取文件的内容，当所引用的外部文件发生错误时，系统只给出一个警告，而整个 php 文件则继续向下执行。下面介绍 `include` 语句的使用方法。

语法：

```
void include(string filename);
```

参数 `filename` 是指定的完整路径文件名。

■ 注意：`include` 语句必须放到 PHP 标记中，否则代码会被视为文本而不会被执行。

**例 5.14** 如果 Web 页面具有一致的外观，可以在 PHP 中使用 `include` 语句将模板和标准元素载入到页面中。下面讲解应用 `include` 语句引用外部文件的方法。（实例位置：光盘\TM\Instances\5\14）

- (1) 首先创建一个简单的 php 动态页，命名为 `index.php`。
- (2) 然后应用 `include` 语句嵌入 3 个外部文件，分别是 `top.php` 页、`main.php` 页和 `bottom.php` 页，代码如下。

```
<table width="975" border="0" cellpadding="0" cellspacing="0">
<tr>
<td><?php include("top.php");?></td>
</tr>
<tr>
<td><?php include("main.php");?></td>
</tr>
<tr>
<td><?php include("bottom.php");?></td>
</tr>
</table>
```

- (3) 最后在以上 3 个外部文件中分别调用图片资源及数据信息。运行结果如图 5.5 所示。



图 5.5 应用 include 语句引用文件

## 5.4.2 应用 require 语句引用文件

require 语句的使用方法与 include 语句类似，都是实现对外部文件的引用。在 PHP 文件被执行之前，PHP 解析器会用被引用的文件的全部内容替换 require 语句，然后与 require 语句之外的其他语句组成新的 PHP 文件，最后再按新 PHP 文件执行程序代码。

**◆ 注意：**因为 require 语句相当于将另一个源文件的内容完全复制到本文件中，所以一般将其放在源文件的起始位置，用于引用需要使用的公共函数文件和公共类文件等。

PHP 可以使用任何扩展名来命名引用文件，如.inc 文件、.html 文件或其他非标准的扩展名文件等，但 PHP 通常用来解析扩展名被定义为.php 文件。建议读者使用标准的文件扩展名。

语法：

```
void require(string filename);
```

参数 filename 是指定的完整路径文件名。

**例 5.15 应用 require 语句引用并运行指定的外部文件 top.php，代码如下：（实例位置：光盘\TM\Instances\5\15）**

```
<?php  
require("top.php");  
?>
```

top.php 文件的代码如下。

```
<table width="750" height="131" border="0" cellpadding="0" cellspacing="0">  
  <tr>  
    <td background="images/banner.jpg">&nbsp;</td>  
  </tr>  
</table>
```

运行 index.php 页，输出显示了 require 语句引用的 Web 页。运行结果如图 5.6 所示。

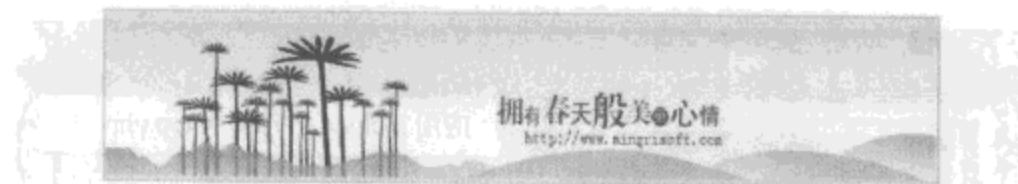


图 5.6 应用 require 语句引用外部文件

### 5.4.3 应用 include\_once 语句引用文件

在使用 `include_once` 语句时，应该明确它与 `include` 语句的区别，应用 `include_once` 语句会在导入文件前先检测该文件是否在该页面的其他部分被引用过，如果有，则不会重复引用该文件，程序只能引用一次。例如，要导入的文件中存在一些自定义函数，那么如果在同一个程序中重复导入这个文件，在第二次导入时便会发生错误，因为 PHP 不允许相同名称的函数被重复声明。

语法：

```
void include_once (string filename);
```

参数 `filename` 是指定的完整路径文件名。

**例 5.16 应用 include\_once 语句引用并运行指定的外部文件 top.php，代码如下：(实例位置：光盘\TM\Instances\5\16)**

```
<?php
include_once("top.php"); //嵌入外部文件 top.php 页
?>
```

top.php 文件的代码如下。

```
<table width="779" height="130" border="0" cellpadding="0" cellspacing="0">
<tr>
<td background="bg.jpg">&ampnbsp</td>
</tr>
</table>
```

运行结果如图 5.7 所示。



图 5.7 应用 include\_once 语句引用文件

### 5.4.4 应用 require\_once 语句引用文件

`require_once` 语句是 `require` 语句的延伸，它的功能与 `require` 语句基本类似，不同的是，在应用 `require_once` 语句时会先检查要引用的文件是不是已经在该程序中的其他地方被引用过，如果有，则不会再重复调用该文件。例如，同时应用 `require_once` 语句在同一页面中引用了两个相同的文件，那么

在输出时只有第一个文件被执行，第二次引用的文件不会被执行。

语法：

```
void require_once (string filename);
```

参数 `filename` 是指定的完整路径文件名。

**例 5.17** 下面讲解应用 `require_once` 语句调用外部文件的实现方法。（实例位置：光盘\TM\Instances\5\17）

(1) 首先，创建一个简单 php 页，命名为 `index.php`，然后应用 `require_once` 语句嵌入 3 个外部文件，代码如下。

```
<table width="1004" border="0" cellpadding="0" cellspacing="0">
<tr>
    <td><?php require_once("top.php");?></td>      // 嵌入头文件
</tr>
<tr>
    <td><?php require_once("main.php");?></td>      // 嵌入主文件
</tr>
<tr>
    <td><?php require_once("bottom.php");?></td>     // 嵌入尾文件
</tr>
</table>
```

(2) 然后在外部文件 `main.php` 页中动态显示相关的数据信息。运行结果如图 5.8 所示。



图 5.8 应用 `require_once` 语句引用文件

### 5.4.5 include 语句和 require 语句的使用区别

应用 require 语句来调用文件，其应用方法和 include 语句是类似的，但存在如下区别。

- ☒ 在使用 require 语句调用文件时，如果调用的文件没找到，require 语句会输出错误信息，并且立即终止脚本的处理。而 include 语句在没有找到文件时则会输出警告，不会终止脚本的处理。
- ☒ 使用 require 语句调用文件时，只要程序一执行，就会立刻调用外部文件；而通过 include 语句调用外部文件时，只有程序执行到该语句时，才会调用外部文件。

### 5.4.6 include\_once 语句和 require\_once 语句的使用区别

include\_once 语句和 require\_once 语句的用途是确保一个被包含文件只能被包含一次。使用这两个语句可以防止意外地多次包含相同的函数库，从而导致函数的重复定义并产生错误。

但两者之间是有区别的：include\_once 语句在脚本执行期间调用外部文件发生错误时，产生一个警告，而 require\_once 语句则导致一个致命错误。

## 5.5 实战

### 5.5.1 通过自定义函数防止新闻主题信息出现中文乱码

视频讲解：光盘\TM\Video\5\通过自定义函数防止新闻主题信息出现中文乱码.exe

例 5.18 创建一个自定义函数，用于实现屏蔽中文乱码的输出，将该函数封装在一个 function.php 文件中，然后应用 include\_once 语句引用这个文件，再通过 echo 语句输出新闻主题信息，截取前 18 个字符，并应用自定义函数屏蔽中文乱码。（实例位置：光盘\TM\Instances\5\18）

(1) 为了保持整个页面的合理布局，经常需要对一些较长的字符串进行部分输出，但由于汉字占有两个字符，如果截取位置选取不当就可能导致截取的字符串末尾出现乱码，本实例将解决上述问题。

运行本实例，如图 5.9 所示，无论在何处对字符串进行截取都不会出现乱码。

(2) PHP 对字符串进行截取可以采用 PHP 的预定义函数 substr() 实现，该函数的使用方法如下。

```
string substr ( string string, int start [, int length])
```

说明：该函数对 string 进行截取操作，截取从字符串的 start 位置开始，到 length 长度结束。

应用 substr() 函数对普通字符进行截取操作会非常方便，但对全角字符进行截取时可能会造成乱码出现。为了解决该问题可以通过自定义函数 chinesesubstr() 解决上述问题，该函数代码如下。



图 5.9 新闻主题信息显示

```
<?php
function chinesesubstr($str,$start,$len) {
    $strlen=$start+$len;
    for($i=0;$i<$strlen;$i++) {
        if(ord(substr($str,$i,1))>0xa0) {
            $tmpstr.=substr($str,$i,2);
            $i++;
        } else
            $tmpstr.=substr($str,$i,1);
    }
    return $tmpstr;
}
?>
```

该函数中的参数\$str 表示要截取的字符串，\$start 表示从何位置开始截取，\$len 表示截取字符串的长度，考虑到编程人员的习惯，自定义函数 chinesesubstr()与 PHP 预定义函数 substr()的参数设计一致，这样程序员就可以像使用函数 substr()一样使用 chinesesubstr()，这里建议将 chinesesubstr()函数单独放入某文件，使用时用 include()函数包含该文件即可。

(3) 引用自定义函数 chinesesubstr()所在的 function.php 文件，并通过循环显示公告主题的前 16 个字符。

```
<table width="160" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
    <td width="165" height="32"></td>
</tr>
<tr>
    <td height="52" background="images/tell_center.gif">
        <?php
            include_once("function.php");
            $news="明日编程词典上市了！";
            $i=1;
            do{
                ?>
            <table width="148" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
                <tr>
                    <td width="17" height="20"></td>
                    <td width="333">
                        <?php
                            echo chinesesubstr($news,0,18);
                            if(strlen($news)>18){
                                echo "...";
                            }
                            ?>
                        </td>
                    </tr>
                    <tr>
                        <td height="5"></td>
                        <td height="5" background="images/back_point_write.gif"></td>
                    </tr>
            </table>
        </?php
    
```

```

</table>
<?php
$i++;
}while($i<=5);
?>
</td>
</tr>
<tr>
<td width="165" height="12"></td>
</tr>
</table>

```

**◆ 注意：**在 PHP 动态网页开发的过程中，从数据库读取数据最为常用，由于数据库的内容在后面的章节中进行讲解，因此，本例使用了字符串变量替代数据库数据，在数据库章节中会讲解如何读取数据库中的内容。

## 5.5.2 应用 include 语句构建在线音乐网站主页

**◆ 视频讲解：**光盘\TM\Video\5\应用 include 语句构建在线音乐网站主页.exe

**例 5.19** 本实例应用 include 语句引用外部文件来构建网站主页。要使网站主页的代码简洁化，而且维护起来很方便，只需要在主页应用表格制作一个简练、大气、个性鲜明的网页布局，然后将每个布局区域内的内容封装成一个独立的文件，最后应用 include 语句引用这些独立的文件即可轻松构建主页。下面讲解应用 include 语句引用外部文件的方法。（实例位置：光盘\TM\Instances\5\19）

对于一个网站而言，主页作为直接的信息展台，所承载信息量将是巨大的。从网站程序编写来说，编写首页的代码需要几百行，甚至上千行，这对于后期维护来说是很麻烦的，尤其是当其他管理员进行维护时，由于对代码不熟悉，经常需要在几千行代码中来回查找。那么，如何来解决这种问题呢？应用引用文件是解决该问题最有效的方法。将拥有指定功能的代码封装成一个单独的文件，当用到该文件时，只需要使用 include 等引用语句对该文件进行引用即可。这样做的目的在于使得对页面的管理和维护都变得更简洁方便，并且减少了代码的重用性。

本实例的实现方法如下：

- (1) 首先创建一个简单 php 动态页，命名为 index.php。
- (2) 设计一个 3 行 1 列的表格，并将第二行拆分成两个单元格。然后应用 include 语句引用 4 个外部文件，分别是 top.html 页、left.html 页、main.html 页和 bottom.html 页，构建网站主页布局的效果如图 5.10 所示。

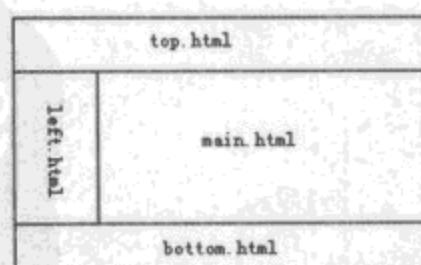


图 5.10 构建网站主页布局

在 index.php 文件中应用 include 语句引用 4 个外部文件的代码如下。

```
<table width="768" border="0" align="center" cell padding="0" cellspacing="0">
<tr>
<td colspan="2"><?php include("top.html");?></td>
</tr>
<tr>
<td width="180"><?php include("left.html"); ?></td>
<td width="588"><?php include("main.html"); ?></td>
</tr>
<tr>
<td colspan="2"><?php include("bottom.html");?></td>
</tr>
</table>
```

(3) 在引用的其他子页中应用 HTML 标记设计子页内容。运行结果如图 5.11 所示。

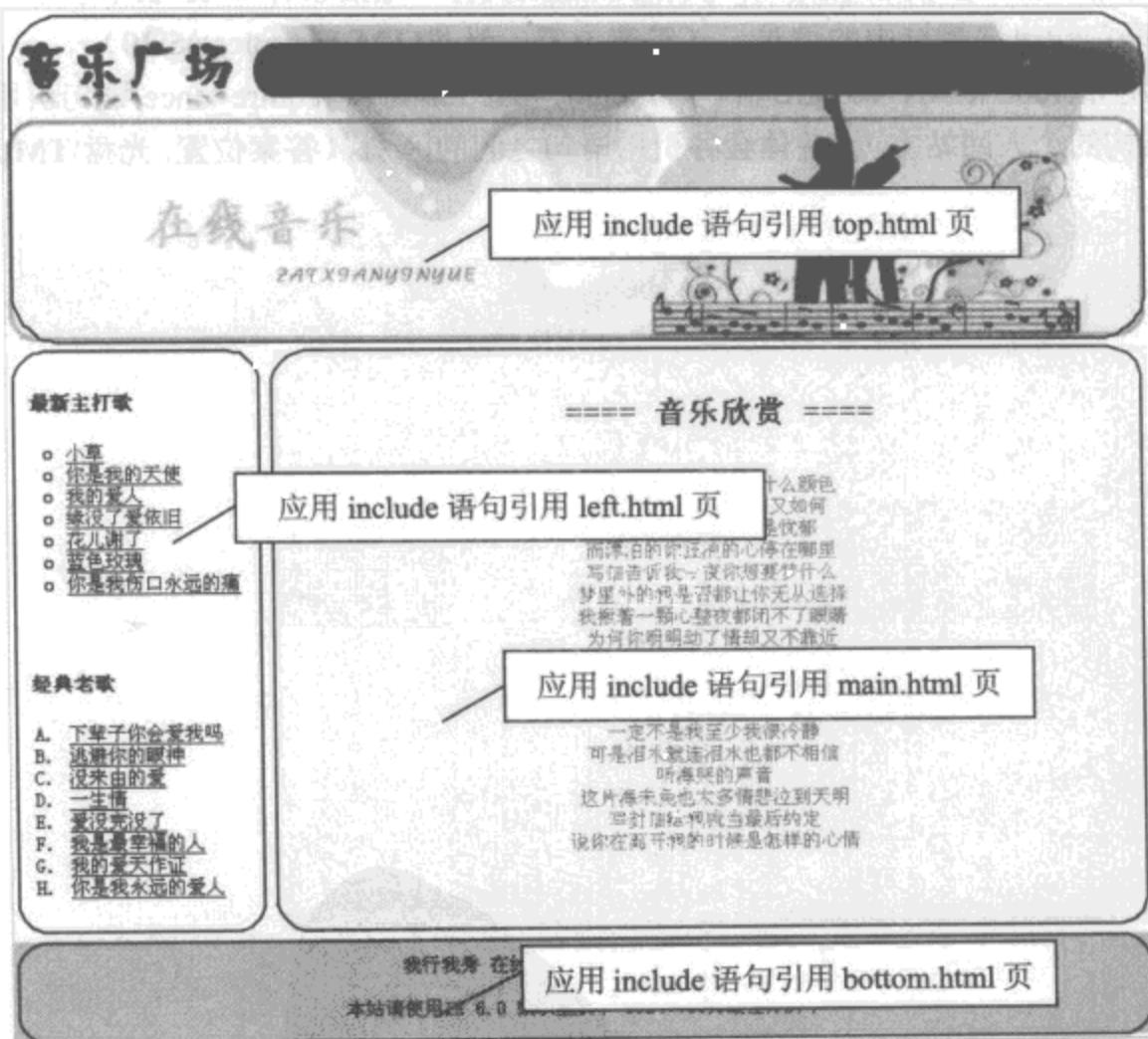


图 5.11 应用 include 语句构建“在线音乐”网站主页

说明：在 PHP 动态网页开发的过程中，在功能上使用最频繁的就是对数据库连接文件的引用。将数据库连接文件封装在一个独立的文件中，然后应用 include 语句或者 require 语句等对封装的文件进行引用，由于数据库内容将在后面的章节中进行介绍，因此，引用数据库文件的方法将在后面进行详细介绍。

## 5.6 本章小结

本章主要介绍了 PHP 语言的自定义函数、输出语句和引用文件，这是在开发过程中应用性极强的操作。熟练掌握自定义函数和引用文件不但可以简化程序流程，而且可以增强代码的重用性，降低开发成本，提高工作效率。

## 5.7 实战练习

1. 自定义一个特殊字符转换函数，可以实现对文本提交的内容进行原格式输出。将函数存储在 function.php 文件中，然后应用 require 语句引用该函数文件，最后设计 Web 表单，单击“提交”按钮后，应用 echo 语句输出编辑框中的信息。（答案位置：光盘\TM\Instances\5\20）
2. 分别应用 include 语句、require 语句、include\_once 语句和 require\_once 语句引用外部文件，实现一个二分栏结构的个人网站主页，并体会各引用语句之间的区别。（答案位置：光盘\TM\Instances\5\21）

# 第 6 章

## 字符串

( 视频讲解: 76分钟)

在许多 Web 编程中，字符串总是会被大量地生成和处理。正确地使用和处理字符串，对于 PHP 程序员来说越来越重要。本章从最简单的字符串定义一直引导读者到高层字符串处理技巧，希望广大读者能够通过本章的学习，了解和掌握 PHP 字符串，达到举一反三的目的，为了解和学习其他的字符串处理技术提供强有力的帮助。

通过阅读本章，可以学习到：

- 字符串基本概念
- 单引号和双引号的使用方法及区别
- 字符串的连接方法
- 去除字符串中的空格
- 字符串的转义及还原
- 获取字符串长度的方法
- 连接和分割字符串的方法
- 字符串的定位技术
- 添加、删除和截取字符串技术
- 查找和替换字符串技术
- 字符串的格式化技术

## 6.1 字符串简介

字符串是由零个或多个字符组成的有限序列。字符包含以下几种类型：

- 数字类型，如 1、10、100 等。
- 字母类型，如 a、b、c、d 等。
- 特殊字符，如\$、%、#、^、&等。
- 不可见字符，如\t（Tab 字符）、\n（换行符）、\r（回车符）等。

其中，不可见字符是比较特殊的一组字符，是用来控制字符串格式化输出的，在浏览器上不可见，只能看到字符串输出的结果。

例如：

```
<?php
    echo "PHP\rASP\nJSP\tASP.net";           //输出字符串
?>
```

应用本实例在 IE 浏览器上看不到不可见字符的输出结果，需要通过选择 IE 浏览器菜单中的“查看”/“源文件”命令来查看执行不可见字符串的实际输出结果。

结果为：

```
PHP
ASP
JSP  ASP.net
```

## 6.2 单引号与双引号

### ■ 视频讲解：光盘\TM\Video\6\单引号与双引号.exe

字符串通常以串的整体作为操作对象，一般用双引号或单引号标识一个字符串。单引号串和双引号串在 PHP 中的处理是不相同的。双引号串中的内容可以被解释而且替换，而单引号串中的内容则被作为普通字符处理。例如：

```
<?php
$str = 6;
echo "str is $str";           //输出结果： str is 6
echo 'str is $str';          //输出结果： str is $str
echo "str is $str\n";
echo 'str is $str\n';         //输出结果： str is 6 (同时换行)
?>
```

从以上代码中可以看出，双引号中的内容是经过 PHP 的语法分析器解析过的，任何变量在双引号中都会被转换为它的值进行输出显示；而单引号的内容是“所见即所得”的，无论有无变量，都被当作普通字符串进行原样输出。

然而，对于定义的普通字符串却显示不出两者之间的区别。例如：

```
<?php
$str1 = "I Like PHP";           //应用双引号定义一个字符串
$str2 = 'I Like PHP';          //应用单引号定义一个字符串
echo $str1;                     //输出双引号中的字符串
echo $str2;                     //输出单引号中的字符串
?>
```

结果为：

```
I Like PHP
I Like PHP
```

 技巧：单引号串和双引号串在 PHP 中的处理是不相同的。双引号串中的内容可以被解释而且替换，而单引号串中的内容被作为普通字符进行处理。

### 6.3 定界符

定界符 (<<<) 是从 PHP 4.0 开始支持的。定界符用于定义格式化的大文本，格式化是指文本中的格式将被保留，所以文本中不需要使用转义字符。在使用时后接一个标识符，然后是格式化文本（即字符串），最后是同样的标识符结束字符串。

定界符格式如下：

```
<<<str
    格式化文本
str
```

其中，符号“<<<”是关键字，必须使用；str 为用户自定义的标识符，用于定义文本的起始标识符和结束标识符，前后的标识符名称必须完全相同。

结束标识符必须从行的第一列开始。而且也必须遵循 PHP 中其他任何标签的命名规则：只能包含字母、数字、下划线，而且必须以下划线或非数字字符开始。

**例 6.1 应用定界符输出变量中的值，可以看到，它和双引号没什么区别，包含的变量也被替换成实际数值，代码如下：（实例位置：光盘\TM\Instances\6\1）**

```
<?php
$str="明日科技编程词典";
echo <<<strmark
<font color="#FF0099"> $str 上市了,详情请关注编程词典网: www.mrbcccd.com </font>
strmark;
?>
```

在上面的代码中，值得注意的是，在定界符内不允许添加注释，否则程序将运行出错。结束标识符所在的行不能包含任何其他字符，而且不能被缩进，在标识符分号前后不能有任何空白字符或制表符。如果破坏了这条规则，则程序不会被视为结束标识符，PHP 将继续寻找下去。如果在这种情况下

找不到合适的结束标识符，将会导致一个在脚本最后一行出现的语法错误。

运行结果如图 6.1 所示。

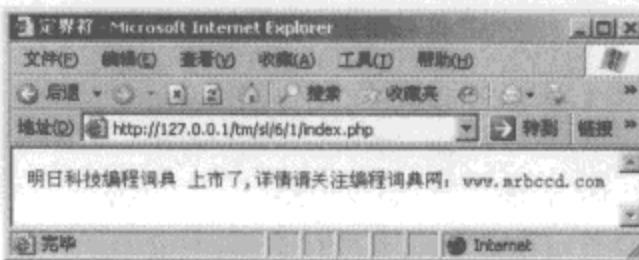


图 6.1 使用定界符定义字符串

技巧：定界符中的字符串支持单引号、双引号，无须转义，并支持字符变量替换。

例 6.2 应用定界符输出一个用户登录的表单，代码如下：(实例位置：光盘\TM\Instances\6\2)

```
<?php
$str=<<<user
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>用户登录</title>
<link rel="stylesheet" type="text/css" href="style.css" />
<body>
<table width="250" height="100" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td bgcolor="#666666"><table width="250" height="100" border="0" cellpadding="0" cellspacing="1">
<form name="form1" method="post" action="index.php">
<tr>
<td height="25" colspan="2" bgcolor="#DDAACC"><div align="center" >用户登录</div></td>
</tr>
<tr>
<td width="60" height="25" bgcolor="#FFFFFF"><div align="right">用户名： </div></td>
<td width="187" bgcolor="#FFFFFF">&ampnbsp<input type="text" name="ip" size="18" class="inputcss"></td>
</tr>
<tr>
<td height="25" bgcolor="#FFFFFF"><div align="right">密码： </div></td>
<td height="25" bgcolor="#FFFFFF">&ampnbsp<input type="text" name="ipNum" size="18" class="inputcss">
</td>
</tr>
<tr>
<td height="25" colspan="2" bgcolor="#FFFFFF"><div align="center"><input name="submit" type="submit" class="buttoncss" value="登录" />
</div></td>
</tr>
</form>
</table></td>
</tr>
```

```
</table>
</body>
</html>
user;
echo $str;
?>
```

**◆ 注意：**在上面的代码中，结束标识符 user 必须单独另起一行，并且不允许有空白字符。如果在标识符前后有其他符号、字符或注释，则会发生错误。

运行结果如图 6.2 所示。



图 6.2 使用定界符输出用户登录表单

## 6.4 连接字符串

### ◆ 视频讲解：光盘\TM\Video\6\连接字符串.exe

字符串可以用“.”（点）字符串连接符连接，它可以把两个或两个以上的字符串连接成一个新的字符串。注意这里不能用“+”（加）运算符。

字符串连接有两种形式，第一种是连接运算符“.”；第二种是连接赋值运算符“.=”。

连接运算符“.”返回其左右参数连接后的字符串。例如：

```
<?php
$name="明日编程词典：";
$url = $name. "www.mrbccd.com";
echo $url;
?>
```

结果为，

明日编程词典： www.mrbccd.com

连接赋值运算符“.=”将右边参数附加到左边的参数后连接成一个新的字符串。例如：

```
<?php
$url="明日编程词典：";
$url .= "www.mrbccd.com ";
echo $url;
?>
```

结果为，

明日编程词典: www.mrbccd.com

## 6.5 转义、还原字符串

在 PHP 编程的过程中，经常会遇到这样的问题，将数据插入到数据库中时可能引起一些问题，出现错误或者乱码等，因为数据库将传入的数据中的字符解释成控制符。针对这种问题，就需要使用一种标记或者是转义这些特殊的字符。

因此，在 PHP 语言中提供了专门处理这些问题的技术，转义和还原字符串。方法有两种：一种是手动转义、还原字符串数据，另一种是自动转义、还原字符串数据。下面分别对这两种方法进行详细讲解。

### 6.5.1 手动转义、还原字符串

字符串可以用单引号 “'”、双引号 “""”、定界符 “{}” 3 种方式定义。而指定一个简单字符串的最简单的方法是用单引号 “'” 括起来。当使用字符串时，很可能在该串中存在这几种符号与 PHP 脚本混淆的字符，因此必须要做转义语句，即在它的前面使用转义符号 “\”。

“\” 是一个转义符，紧跟在 “\” 后面的第一个字符将变为没有意义或特殊意义。例如，“'” 是字符串的定界符，“\” 这样时就使它失去了定界符的意义了，变为普通的单引号 “'”。读者可以通过 “echo \"\";” 输出一个单引号 “'”，同时转义字符 “\” 也不会显示。

 技巧：如果要在字符串中表示单引号，则需要用反斜线 “\” 进行转义。例如，要表示字符串 “I'm”，则需要写成 “I\\'m”。

例 6.3 使用转义字符 “\” 对字符串进行转义，代码如下：（实例位置：光盘\TM\Instances\6\3）

```
<?php
    echo ' select * from tb_book where bookname = \'PHP 项目开发全程实录\' ';
?>
```

结果为： select \* from tb\_book where bookname = 'PHP 项目开发全程实录'

 技巧：对于简单的字符串，建议采用手动方法进行字符串转义，而对于数据量较大的字符串，建议采用自动转义函数实现字符串的转义。

### 6.5.2 自动转义、还原字符串

自动转义、还原字符串可以应用 PHP 提供的 addslashes() 函数和 stripslashes() 函数实现。

#### addslashes() 函数

addslashes() 函数用来给字符串 str 加入斜线 “\”，对指定字符串中的字符进行转义，该函数可以转义的字符包括单引号 “'”、双引号 “""”、反斜杠 “\”、NULL 字符 “0” 。该函数比较常用的地方就是在生成 SQL 语句时，对 SQL 语句中的部分字符进行转义。

语法：

**string addslashes( string str)**

参数 str 为将要被操作的字符串。

#### ☒ stripslashes()

stripslashes() 函数用来将应用 addslashes() 函数转义后的字符串 str 返回原样。

语法：

**string stripslashes(string str);**

参数 str 为将要被操作的字符串。

**例 6.4 使用自动转义字符 addslashes() 函数对字符串进行转义，然后应用 stripslashes() 函数进行还原，代码如下：（实例位置：光盘\TM\Instances\6\4）**

```
<?php
$str = "select * from tb_book where bookname = 'PHP 项目开发全程实录'";
$a = addslashes($str);
echo $a."<br>";
$b = stripslashes($a);
echo $b."<br>";
?>
```

运行结果如图 6.3 所示。



图 6.3 自动转义和还原字符串

**技巧：**所有数据在插入数据库之前，有必要应用 addslashes() 函数进行字符串转义，以免特殊字符未经转义在插入数据库时出现错误。另外，对于应用 addslashes() 函数实现的自动转义字符串可以应用 stripslashes() 函数进行还原，但数据在插入数据库之前必须再次进行转义。

以上两个函数实现了对指定字符串进行自动转义和还原。除了上面介绍的方法外，还可以对要转义、还原的字符串进行一定范围的限制，通过应用 addcslashes() 函数和 stripcslashes() 函数实现对指定范围内的字符串进行自动转义、还原。

#### ☒ addcslashes()

实现对指定字符串中的字符进行转义，即在指定的字符 charlist 前加上反斜线。通过该函数可以将要添加到数据库中的字符串进行转义，从而避免出现乱码等问题。

语法：

**string addcslashes ( string str, string charlist)**

参数 str 为将要被操作的字符串；参数 charlist 指定在字符串中哪些字符前加上反斜线 “\”，如果参数 charlist 中包含有 “\n”、“\r” 等字符，将以 C 语言风格转换，而其他非字母数字且 ASCII 码低

于 32 以及高于 126 的字符均转换成以八进制表示。

**◆ 注意：**在定义参数 charlist 的范围时，需要明确在开始和结束的范围内的字符。

#### ☒ stripcslashes()函数

stripcslashes()函数用来将应用 addslashes() 函数转义的字符串 str 返回原样。

语法：

```
string stripcslashes ( string str)
```

参数 str 为将要被操作的字符串。

**例 6.5 应用 addslashes() 函数对字符串“编程体验网”进行转义，应用 stripcslashes() 函数对转义的字符串进行还原，代码如下：（实例位置：光盘\TM\Instances\6\5）**

```
<?php
$Sa="编程体验网"; //对指定范围内的字符进行转义
$b=addcslashes($a,"编程体验网"); //转义指定的字符串
echo "转义字符串：" . $b; //输出转义后的字符串
echo "<br>"; //执行换行
$c=stripcslashes($b); //对转义的字符串进行还原
echo "还原字符串：" . $c; //输出还原后的转义字符串
?>
```

运行结果如图 6.4 所示。

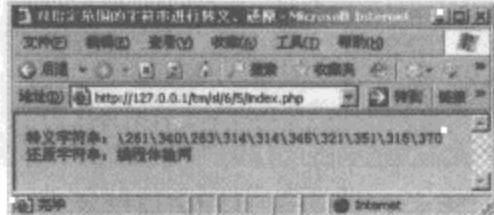


图 6.4 对指定范围的字符串进行转义和还原

**技巧：**缓存文件中，一般对缓存数据的值采用 addslashes() 函数进行指定范围的转义。

## 6.6 获取字符串长度

获取字符串长度主要通过 strlen() 函数实现，下面重点讲解 strlen() 函数的语法及其应用。

strlen() 函数主要用于获取指定字符串 str 的长度。

语法：

```
int strlen(string str)
```

**例 6.6 应用 strlen() 函数来获取指定字符串长度，代码如下：（实例位置：光盘\TM\Instances\6\6）**

```
<?php
echo strlen("编程词典网:www.mrbcccd.com"); //输出指定字符串长度
?>
```

结果为：25

说明：汉字占两个字符，数字、英文、小数点、下划线和空格各占一个字符。

该函数在获取字符串长度的同时，也可以用来检测字符串的长度。

例 6.7 应用 strlen() 函数对提交的用户密码的长度进行检测，如果其长度小于 6 位，则弹出提示信息。（实例位置：光盘\TM\Instances\6\7）

- (1) 利用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php。
- (2) 添加一个表单，将表单的 action 属性设置为“index\_ok.php”。

```
<form name="form1" method="post" action="index_ok.php">
<tr>
<td height="33"></td>
<td align="center"><span class="style1">用户名</span></td>
<td>
<input name="user" type="text" id="user" size="15">
</td>
<td align="center">
<input name="imageField" type="image" src="images/btn_dl.jpg" width="50" height="20" border="0">
</td>
<td>&nbsp;</td>
</tr>
<tr>
<td height="27">&nbsp;</td>
<td align="center"><span class="style1">密码</span></td>
<td>
<input name="pwd" type="password" id="pass" size="15">
</td>
<td colspan="2" align="left"><span class="STYLE3">*&nbsp;密码长度不能少于 6 位</span></td>
</tr>
</form>
```

(3) 应用 HTML 标记设计页面，添加一个“用户名”文本框，命名为 user；添加一个“密码”文本框，命名为 pwd；添加一个图像域，指定源文件位置为 images/btn\_dl.jpg。

(4) 再新建一个 PHP 动态页，存储为 index\_ok.php。通过 POST 方法（关于 POST 方法将在后面章节中进行详细的讲解）接收用户输入的用户密码的值。应用 strlen() 函数获取用户提交密码的长度，应用 if 条件控制语句判断密码长度是否小于 6，并给出相应的提示信息，代码如下。

```
<?php
if(strlen($_POST["pwd"])<6){ //检测用户密码的长度是否小于 6，弹出警告信息
    echo "<script>alert('用户密码的长度不得少于 6 位!请重新输入'); history.back();</script>";
}
else{ //用户密码大于等于 6 位，则弹出该提示信息
    echo "用户信息输入合法！";
}
?>
```

(5) 在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 6.5 所示。



图 6.5 应用 strlen() 函数检测字符串的长度

## 6.7 截取字符串

在 PHP 中，有一项非常重要的技术，就是截取指定字符串中指定长度的字符。PHP 对字符串截取可以通过 PHP 的预定义函数 substr() 实现。本节重点介绍字符串的截取技术。

substr() 函数从字符串中按照指定位置截取一定长度的字符。通过该函数可以获取某个固定格式字符串中的一部分，如果使用一个正数作为子串起点来调用这个函数，将得到从起点到字符串结束的这个字符串；如果使用一个负数作为子串起点来调用，将得到一个原字符串尾部的一个子串，字符个数等于给定负数的绝对值。

语法：

```
string substr ( string str, int start [, int length])
```

参数 str 用来指定字符串对象。参数 start 用来指定开始截取字符串的位置，如果参数 start 为负数，则从字符串的末尾开始截取。参数 length 为可选项，指定截取字符的个数，如果 length 为负数，则表示截取到倒数第 length 个字符。

**◆ 注意：** 本函数中参数 start 的指定位置是从 0 开始计算的，即字符串中的第一个字符表示为 0。

**例 6.8 应用 substr() 函数截取字符串中指定长度的字符，代码如下：(实例位置：光盘\TM\Instances\6\8)**

```
<?php
echo substr("www.mrbccd.com",0);           //从第 0 个字符开始截取
echo "<br>";
echo substr("www.mrbccd.com",4,10);          //从第 4 个字符开始连续截取 10 个字符
echo "<br>";
echo substr("www.mrbccd.com",-4,4);         //从倒数第 4 个字符开始截取 4 个字符
echo "<br>";
echo substr("www.mrbccd.com",0,-4);          //从第一个字符开始截取，截取到倒数第 4 个字符
?>
```

结果为：

```
www.mrbccd.com
mrbccd.com
.com
www.mrbccd
```

**例 6.9** 通常，在开发 Web 程序时，为了保持整个页面的合理布局，经常需要对一些超长文本进行部分显示。下面应用 substr() 函数截取超长文本的部分字符串，剩余的部分用“...”代替，代码如下：  
**(实例位置：光盘\TM\Instances\6\9)**

```
<?php
$str="为进一步丰富编程词典的内容和观赏性，公司决定组织“冬季恋影杯”摄影大赛，本次参赛作品要求全部为
冬季拍摄，旨在展示我国北方地区的优美景色。";
if(strlen($str)>70){                                //如果文本的字符串长度大于 70
    echo substr($str,0,70)."...";                    //输出文本的前 70 个字符串，然后输出省略号
}
else{                                                 //如果文本的字符串长度小于 70
    echo $str;                                       //直接输出文本
}
?>
```

结果为：为进一步丰富编程词典的内容和观赏性，公司决定组织“冬季恋影杯”摄影大赛…

## 6.8 比较字符串

在 PHP 中，对字符串之间进行比较的方法有 3 种：第一种应用 strcmp() 函数按照字节进行比较；第二种应用 strnatcmp() 函数按照自然排序法进行比较；第三种应用 strncmp() 函数指定从源字符串的位置开始比较。下面分别对这 3 种方法进行讲解。

### 6.8.1 按字节比较

按字节进行字符串是比较字符串最常用的方法。其中 strcmp() 函数和 strcasecmp() 函数都可以实现对字符串进行按字节的比较，这两种方法的区别是 strcmp() 函数区分字符的大小写，而 strcasecmp() 函数不区分字符的大小写。这两个函数的实现方法基本相同，这里只介绍 strcmp() 函数。

strcmp() 函数用来对两个字符串进行比较。

语法：

```
int strcmp ( string str1, string str2)
```

参数 str1 和参数 str2 指定要比较的两个字符串。如果相等则返回 0；如果参数 str1 大于参数 str2 则返回 1；如果参数 str1 小于参数 str2 则返回 -1。

 注意：该函数区分字母大小写。

**例 6.10** 应用 strcmp() 和 strcasecmp() 函数分别对两个字符串按字节进行比较，代码如下：  
**(实例位置：光盘\TM\Instances\6\10)**

```
<?php
$str1="明日编程词典!";
//定义字符串常量
$str2="明日编程词典!";
//定义字符串常量
```

```

$str3="mrsoft";
$str4="MRSOFT";
echo strcmp($str1,$str2);          //定义字符串常量
echo strcmp($str3,$str4);          //定义字符串常量
echo strcasecmp($str3,$str4);      //这两个字符串相等
echo strcasecmp($str3,$str4);      //注意该函数区分大小写
echo strcasecmp($str3,$str4);      //该函数不区分字母大小写
?>

```

结果为： 0 1 0

 **技巧：**在 PHP 中，对字符串之间进行比较的应用也是非常广泛的。例如，应用 `strcmp()` 函数来比较在用户登录系统中输入的用户名和密码是否正确。如果在验证用户名和密码时不应用此函数，那么输入的用户名和密码无论是大写还是小写，只要正确就可以登录。应用了 `strcmp()` 函数之后就避免了这种情况，即使正确，必须大小写匹配才可以登录，从而提高了网站的安全性。

## 6.8.2 按自然排序法比较

在 PHP 中，按照自然排序法进行字符串比较是通过 `strnatcmp()` 函数来实现的。自然排序法比较的是字符串中的数字部分，将字符串中的数字按照大小进行排序。

`strnatcmp()` 函数通过自然排序法比较字符串。

语法：

```
int strnatcmp ( string str1, string str2)
```

如果字符串相等则返回 0，如果参数 `str1` 大于参数 `str2` 则返回 1；如果参数 `str1` 小于参数 `str2` 则返回 -1。本函数区分字母大小写。

 **注意：**在自然排序法中，2 比 10 小。在计算机序列当中，10 比 2 小，因为“10”中的第一个数字是“1”，它小于 2。

**例 6.11 应用 `strnatcmp()` 函数按自然排序法进行字符串的比较，代码如下：**（实例位置：光盘\TM\Instances\6\11）

```

<?php
$str1="str2.jpg";           //定义字符串常量
$str2="str10.jpg";          //定义字符串常量
$str3="mrsoft1";            //定义字符串常量
$str4="MRSOFT2";            //定义字符串常量
echo strcmp($str1,$str2);    //按字节进行比较，返回 1
echo strcmp($str3,$str4);    //按字节进行比较，返回 1
echo strnatcmp($str1,$str2); //按自然排序法进行比较，返回-1
echo strnatcmp($str3,$str4); //按自然排序法进行比较，返回 1
?>

```

结果为： 1 1 -1 1

 **说明：**针对按照自然排序法进行比较，还有一个与 `strnatcmp()` 函数相同，但是不区分大小写的 `strnatcasecmp()` 函数，它的作用和 `strnatcmp()` 函数相同。

### 6.8.3 指定从源字符串的位置比较

`strcmp()`函数用来比较字符串中的前 n 个字符。

语法：

```
int strcmp(string str1,string str2,int len)
```

参数 `str1` 用来指定参与比较的第一个字符串对象。参数 `str2` 用来指定参与比较的第二个字符串对象。参数 `len` 为必选参数，用来指定每个字符串中参与比较字符的数量。

如果字符串相等则返回 0，如果参数 `str1` 大于参数 `str2` 则返回 1；如果参数 `str1` 小于参数 `str2` 则返回 -1。该函数区分字母大小写。

**例 6.12 应用 `strcmp()` 函数比较字符串的前 6 个字符是否与源字符串相等，代码如下：（实例位置：光盘\TM\Instances\6\12）**

```
<?php
$str1="I love PHP !";
//定义字符串常量
$str2="i love my student !";
//定义字符串常量
echo strcmp($str1,$str2,6);
//比较前 6 个字符
?>
```

结果为： -1

从上面的代码中可以看出，由于变量 `$str2` 中的字符串的首字母为小写，与变量 `$str1` 中的字符串不匹配，因此比较后的字符串返回值为 -1。

## 6.9 检索字符串

在 PHP 中，提供了很多用于字符串查找的函数，PHP 也可以像 Word 那样实现对字符串的查找功能。下面讲解常用的字符串检索技术。

### 6.9.1 应用 `strstr()` 函数检索指定的关键字

获取一个指定字符串在另一个字符串中首次出现的位置到后者末尾的子字符串。如果执行成功，则返回剩余字符串（存在相匹配的字符）；如果没有找到相匹配的字符，则返回 `false`。

语法：

```
string strstr ( string haystack, string needle)
```

参数 `haystack` 为必选参数，用来指定从哪个字符串中进行搜索。参数 `needle` 为必选参数，用来指定搜索的对象，如果该参数是一个数值，那么将搜索与这个数值的 ASCII 值相匹配的字符。

◆ 注意：本函数区分字母的大小写。

**例 6.13 应用 strstr() 函数获取指定字符串在字符串中首次出现的位置后所有字符，代码如下：**  
**(实例位置：光盘\TM\Instances\6\13)**

```
<?php
echo strstr("明日编程词典网","编");           //输出查询的字符串
echo "<br>";                                //执行换行
echo strstr("http://www.mrbccd.com","w");      //输出查询的字符串
echo "<br>";                                //执行换行
echo strstr("0431-84978981","8");            //输出查询的字符串
?>
```

结果为：

编程词典网

www.mrbccd.com

84978981

通过上面的代码可以看出，应用 strstr() 函数自定义检索字符串非常方便，另外， strrchr() 函数与其正好相反，该函数是从字符串后序的位置开始检索子串的。

## 6.9.2 应用 substr\_count() 函数检索子串出现的次数

在 PHP 中，有一种技术可以获取字符串中字符和单词数量，通过该技术可以查看到指定字符或者单词在字符串中出现的次数，而且还可以应用到论坛、博客或者聊天室的信息发布模块中，判断提交的信息中是否含有非法关键字。

substr\_count() 函数获取指定字符在字符串中出现的次数。

语法：

```
int substr_count(string haystack,string needle)
```

参数 haystack 是指定的字符串，参数 needle 为指定的字符。

下面应用 substr\_count() 函数获取子串在字符串中出现的次数，代码如下。

```
<?php
$str="明日编程词典";
echo substr_count($str,"词");           //输出查询的字符串
?>
```

结果为：1

从表面上看，该函数的功能就是获取指定字符在字符串中出现的次数，输出的只是一个数字，但在实际的运用中，只要对输出的数字加以判断后，就能够实现不同功能。

**例 6.14 应用 substr\_count() 函数判断在聊天室的发布信息中是否含有指定的关键字。本实例中将“女人、男人”设置为指定的关键字。如果 substr\_count() 函数的返回值大于 0，则信息中有指定的关键字，弹出提示对话框；否则信息中没有指定的关键字，输出“数据添加成功!!”。代码如下：**  
**(实例位置：光盘\TM\Instances\6\14)**

```

<?php
if($Submit==true){
$user=$_POST[user];
$text=$_POST[text];
if(substr_count($user,'女人')>0 or substr_count($text,'女人')>0 or substr_count($user,'男人')>0 or substr_
count($text,'男人')>0){
    echo "<script>alert('您输入的内容中有指定的关键字'); history.back();</script>";
}
else{
    echo "数据添加成功!!";
}
}
?>
<table width="720" border="0" cellspacing="0" cellpadding="0">
<tr>
<td></td>
</tr>
</table>
<table width="720" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="121"></td>
<td width="584" height="454"><table width="584" height="454" border="0" cellspacing="0" cellpadding="0">
<tr>
<td align="center" valign="top" bgcolor="#F2F7FB">
<form name="form1" method="post" action="index.php">
<table width="580" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="89" height="30">&nbsp;</td>
<td width="491">&nbsp;</td>
</tr>
<tr>
<td align="right"><span class="STYLE1">名称:&nbsp;</span></td>
<td height="24"><input name="user" type="text" id="user" size="50" /></td>
</tr>
<tr>
<td align="right"><span class="STYLE1">内容:&nbsp;</span></td>
<td><textarea name="text" cols="60" rows="10" id="text"></textarea></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="Submit" value="提交" /></td>
</tr>
</table>
</form>
</td>
</tr>
<tr>
<td height="48"></td>
</tr>
</table></td>
<td width="15"></td>

```

```
</tr>
</table>
```

运行结果如图 6.6 所示。

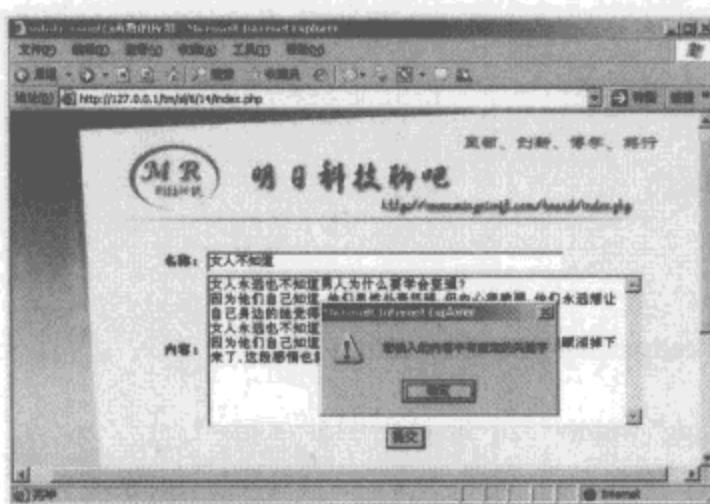


图 6.6 判断聊天室的发布信息中是否含有指定的关键字

技巧：检索子串出现的次数一般常用于搜索引擎中，针对子串在字符串中出现的次数进行统计，便于用户第一时间掌握子串在字符串中出现的次数。

## 6.10 替换子字符串

子字符串的替换可以实现对指定字符串中的指定字符进行替换。此项功能最常见的用处就是在搜索引擎的关键字处理中，可以使用子字符串替换技术将搜索到的字符串中的关键字替换颜色，使搜索到的结果便于查看。子字符串的替换技术可以通过以下两个函数实现：`str_replace()`函数和`str_ireplace()`函数。

### 1. `str_replace()`函数

使用新的子字符串替换原始字符串中被指定要替换的字符串。该函数区分字母大小写。

语法：

```
mixed str_replace ( mixed search, mixed replace, mixed subject [, int &count] )
```

将所有在参数 `subject` 中出现的 `search` 用参数 `replace` 替换，参数`&count` 表示替换字符串执行的次数。`str_replace()`函数的参数说明如表 6.1 所示。

表 6.1 `str_replace()`函数的参数说明

参    数	说    明
<code>search</code>	必要参数，指定需要查找的字符串
<code>replace</code>	必要参数，指定替换的值
<code>subject</code>	必要参数，指定查找的范围
<code>count</code>	可选参数，获取执行替换的数量

**◆ 注意：**该函数可以以数组的方式传递参数。函数返回的是一个字符串还是数组，取决于被操作的对象是字符串还是数组。如果原始字符串 subject 是一个数组，则该函数会依次用 replace 替换 subject 数组中每个元素中的 search 子字符串，同时该函数的返回值为一个数组。

**例 6.15** 下面将文本中的指定字符串“予人快乐，自己也将更快乐”替换为“予人关怀，自己也会得到关怀”，并且输出替换后的结果，程序代码如下：（实例位置：光盘\TM\Instances\6\15）

```
<?php
$str="予人温暖，自己也更温暖。予人快乐，自己也将更快乐。这是一种境界，也是心灵上的一种拥有。在温暖里，人才不会迷失，在相互扶持中，婚姻才会更长久。就算偶有寒流来袭，只要两颗心相互取暖，又怎么会给寂寞留下可乘之机呢。";
if($_POST["Submit"]=="提交"){
    $str2=$_POST[str2];
    $str1=<font color="#FF0000">$_POST[str1]</font>;
    $stres = str_replace($str2,$str1,$str);
}
?>
<table width="760" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="415" height="490" align="center" valign="top" background="images/bg.jpg">
<table width="400" height="462" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="161" height="165">&ampnbsp</td>
<td width="239">&ampnbsp</td>
</tr>
<tr>
<td height="112" colspan="2" align="center"><table width="360" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="360" align="left"><span class="STYLE4">原字符串：<br>
&ampnbsp&ampnbsp<?php echo $str;?></span></td>
</tr>
</table></td>
</tr>
<form name="form2" method="post" action="index.php">
<tr>
<td height="32" align="right"><span class="STYLE3">被替换的字符</span></td>
<td align="left"><input name="str2" type="text" id="str2" size="24"></td>
</tr>
<tr>
<td height="30" align="right"><span class="STYLE3">替换的字符</span></td>
<td height="30">
<input name="str1" type="text" id="str1" size="24">
</td>
</tr>
<tr>
<td>&ampnbsp</td>
<td align="center" valign="top">
<input name="Submit" type="submit" id="Submit" value="提交">
</td>
```

```

        </tr>
    </form>
</table></td>
<td width="345" align="center" valign="middle" background="images/bg1.jpg"><table width="300" border="0"
cellspacing="0" cellpadding="0">
<tr>
    <td class="STYLE4"><?php echo $stres;?></td>
</tr>
</table>
</td>
</tr>
</table>

```

运行结果如图 6.7 所示。

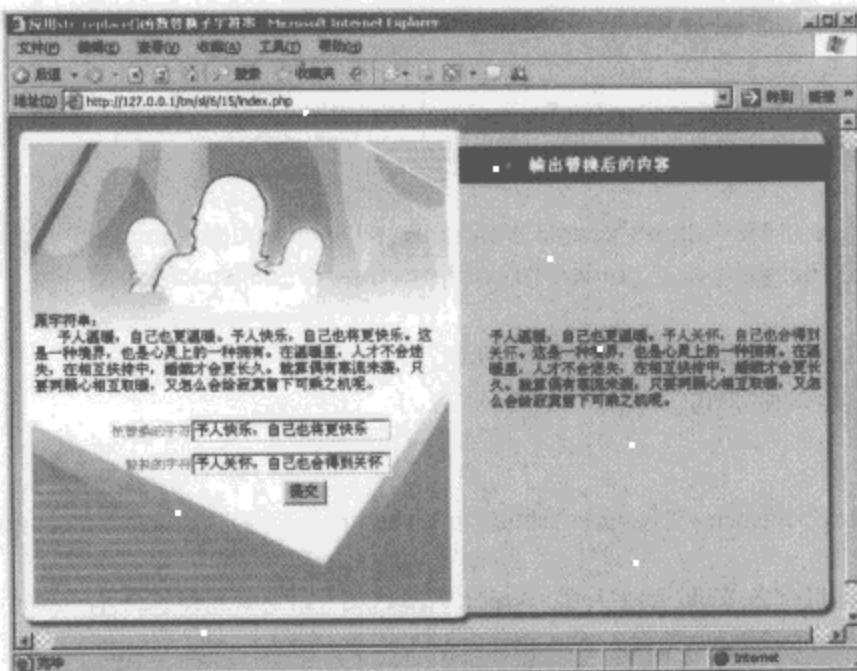


图 6.7 应用 str\_replace()函数替换子字符串

## 2. str\_ireplace()函数

将指定的字符串替换为另外一个指定的字符串。本函数不区分大小写。该函数不仅可以对指定的子字符串进行替换，而且可以控制替换的次数。该函数最适合应用到搜索引擎的关键字描红中。

使用新的子字符串替换原始字符串中被指定要替换的字符串。

语法：

```
mixed str_ireplace ( mixed search, mixed replace, mixed subject [, int &count])
```

该函数用来将所有在参数 `subject` 中出现的参数 `search` 用参数 `replace` 取代，参数 `&count` 为可选参数，表示取代字符串执行的次数。该函数在执行替换的操作时，是不区分大小写的。`str_ireplace()` 函数的参数说明可以参考表 6.1。

下面将指定字符串中的“忘 me”替换为“追求忘我”，并且输出替换后的结果，代码如下。

```
<?php
$str2="忘 Me"; //定义字符串常量
```

```

$str1="追求忘我"; //定义字符串常量
$str="    忘 me 是走向成功的一条捷径，只有在这种环境中，人才会超越自身的束缚，释放出最大的能量。";
echo str_ireplace($str2,$str1,$str); //输出替换后的字符串
?>

```

结果为：追求忘我是走向成功的一条捷径，只有在这种环境中，人才会超越自身的束缚，释放出最大的能量。

**◆ 注意：**如果在上面的例子中需要对大小写加以区分，可以应用 str\_replace()函数，不过该实例并不能实现替换功能，因为\$str 中的“me”是小写的，而查找的字符串\$str2 中的“Me”是大写的。

字符串替换技术最常见的用处就是在搜索引擎的关键字处理中，可以使用字符串替换技术将搜索到的字符串中的关键字替换颜色，如查询关键字描红功能，使搜索到的结果更便于用户查看。

**例 6.16** 下面实现一个模糊查询的功能，并应用 str\_ireplace()函数对查询的关键字进行描红。首先创建一个搜索的页面，并创建提交查询关键字的表单，然后连接数据库，根据表单提交的关键字对数据库中的数据进行检索，输出与之相匹配的内容，并且将检索出的内容中的关键字以红色字体显示。程序代码如下：（实例位置：光盘\TM\Instances\6\16）

```





```

```

</td>
</tr>
<?php }while($info=MySQL_fetch_array($sql));
}else{
?>
<tr>
<td><span class="STYLE2">对不起，您检索的信息不存在！</span></td>
</tr>
<?php } ?>
</table>

```

运行结果如图 6.8 所示。



图 6.8 查询关键字描红

**注意：**查询关键字描红是指将查询关键字以特殊的颜色、字号或字体进行标识。这样可以使浏览者快速检索到所需的关键字，方便浏览者从搜索结果中查找所需内容。查询关键字描红适用于模糊查询。

## 6.11 去除字符串中空白字符和特殊字符

用户在输入数据时，经常会无意地输入多余的空白字符，在有些情况下，字符串中不允许出现空白字符和特殊字符，这时就需要去除字符串中的空白字符和特殊字符。在 PHP 中提供了 trim() 函数去除字符串左右两边的空白字符和特殊字符，ltrim() 函数去除字符串左边的空白字符和特殊字符，rtrim() 函数去除字符串中右边的空白字符和特殊字符。下面分别对这 3 个函数进行详细介绍。

### 6.11.1 去除字符串左端空白字符和特殊字符

ltrim() 函数用于去除字符串左边的空白字符或者指定字符串。

语法：

```
string ltrim( string str [,string charlist]);
```

**例 6.17 使用 ltrim() 函数去除字符串左边的空白字符及特殊字符 “(@\_@)” , 代码如下:** (实例位置: 光盘\TM\Instances\6\17)

```
<?php
$str=" (@_@ 有一条路走过了总会想起! @_@:) ";
echo ltrim($str); //去除字符串左边的空白字符
echo "<br>"; //执行换行
echo ltrim($str," @_@:"); //去除字符串左边的特殊字符 @_@:
?>
```

结果为:

```
( @_@ 有一条路走过了总会想起! @_@:)
有一条路走过了总会想起! @_@:)
```

### 6.11.2 去除字符串右端空白字符和特殊字符

rtrim()函数用于去除字符串右边的空白字符。

语法:

```
string rtrim(string str [,string charlist]);
```

**例 6.18 使用 rtrim() 函数去除字符串右边的空白字符及特殊字符 “ @\_@:)” , 代码如下:** (实例位置: 光盘\TM\Instances\6\18)

```
<?php
$str=" (@_@ 有一段感情经历了就再难忘记! @_@:) ";
echo rtrim($str); //去除字符串右边的空白字符
echo "<br>"; //执行换行
echo rtrim($str," @_@:"); //去除字符串右边的特殊字符 @_@:
?>
```

结果为:

```
( @_@ 有一段感情经历了就再难忘记! @_@:)
( @_@ 有一段感情经历了就再难忘记!
```

### 6.11.3 去除字符串两端空白字符和特殊字符

trim()函数用于去除字符串开始位置以及结束位置的空白字符, 并返回去掉空白字符后的字符串。

语法:

```
string trim(string str [,string charlist]);
```

trim()函数的参数 str 是要操作的字符串对象, 参数 charlist 为可选参数, 指定需要从指定的字符串中删除哪些字符, 如果不设置该参数, 则所有的可选字符都将被删除。参数 charlist 的可选值如表 6.2 所示。

表 6.2 trim()函数的参数 charlist 的可选值

参 数 值	说 明
\0	NULL, 空值
\t	tab, 制表符
\n	换行符
\x0B	垂直制表符
\r	回车符
""	空白字符

◆ 注意：除了以上默认的过滤字符列表外，也可以在 charlist 参数中提供要过滤的特殊字符。

例 6.19 使用 trim() 函数去除字符串左右两边的空白字符及特殊字符 “\r\n(\:@:)” ，代码如下：（实例位置：光盘\TM\Instances\6\19）

```
<?php
$str = "\r\n(\:@_@有一条路走过了总会想起！有一段感情经历了就再难忘记！ @_@:) ";
echo trim($str); //去除字符串左右两边的空白字符
echo "<br>"; //执行换行
echo trim($str, "\r\n(\:@_@ @_@:)"); //去除字符串左右两边的空白字符和特殊字符\r\n(\:@:)
?>
```

结果为：

```
(:@_@有一条路走过了总会想起！有一段感情经历了就再难忘记！ @_@:)
有一条路走过了总会想起！有一段感情经历了就再难忘记！
```

## 6.12 格式化字符串

通过字符串格式化技术可以实现对指定字符进行个性化输出，以不同的类型进行显示。例如，在输出数字字符串时，可以应用格式化技术指定数字输出的格式，保留几位小数或者不留小数。

number\_format() 函数用来将数字字符串格式化。

语法：

```
string number_format(float number,[int num_decimal_places],[string dec_seperator],string thousands_seperator)
```

参数 number 为格式化后的字符串，该函数可以有一个、两个或是 4 个参数，但不能是 3 个参数。如果只有一个参数 number，number 格式化后会舍去小数点后的值，且每 1000 就会以逗号“,”来隔开；如果有两个参数，number 格式化后会到小数点第 num\_decimal\_places 位，且每 1000 就会以逗号来隔开；如果有 4 个参数，number 格式化后会到小数点第 num\_decimal\_places 位，dec\_seperator 用来替代小数点“.”，thousands\_seperator 用来替代每一千隔开的逗号“,”。

例 6.20 应用 number\_format() 函数对指定的数字字符串进行格式化处理，代码如下：（实例位置：光盘\TM\Instances\6\20）

```
<?php
$number = 2668.746; //定义数字字符串常量
```

```

echo number_format($number); //输出一个参数格式化后的数字字符串
echo "<br>";
echo number_format($number, 2); //输出两个参数格式化后的数字字符串
echo "<br>";
$number2 = 123456.7832; //定义数字字符串常量
echo number_format($number2, 2, ',', '.'); //输出 4 个参数格式化后的数字字符串
?>

```

结果为：

```

2,669
2,668.75
123.456.78

```

## 6.13 分割字符串

字符串的分割是通过 `explode()` 函数实现的。使用该函数可以将指定字符串中的内容按照某个规则进行分类存储，进而实现更多的功能。例如，在实现电子商务网站的购物车功能时，可以通过特殊标识符“@”将购买的多种商品组合成一个字符串存储在数据表中，在显示购物车中的商品时，就可以通过以“@”作为分割的标识符进行拆分，分割成 N 个数组元素，最后循环语句输出各个数组元素，即购买的商品。

`explode()` 函数按照指定的规则对一个字符串进行分割，返回值为数组。

语法：

```
array explode(string separator,string str,[int limit])
```

`explode()` 函数的参数说明如表 6.3 所示。

表 6.3 `explode()` 函数的参数说明

参 数	说 明
separator	必要参数，指定的分割符。如果 <code>separator</code> 为空字符串 “””， <code>explode()</code> 将返回 <code>false</code> 。如果 <code>separator</code> 所包含的值在 <code>str</code> 中找不到，那么 <code>explode()</code> 函数将返回包含 <code>str</code> 单个元素的数组
str	必要参数，指定将要被进行分割的字符串
limit	可选参数，如果设置了 <code>limit</code> 参数，则返回的数组包含最多 <code>limit</code> 个元素，而最后那个元素将包含 <code>string</code> 的剩余部分；如果 <code>limit</code> 参数是负数，则返回除了最后的 <code>-limit</code> 个元素外的所有元素

例 6.21 应用 `explode()` 函数实现对已知的字符串“品牌电脑@品牌手机@高档男士衬衫@高档女士挎包”进行分割，代码如下：（实例位置：光盘\TM\Instances\6\21）

```

<?php
$str="品牌电脑@品牌手机@高档男士衬衫@高档女士挎包"; //定义字符串常量
$str_arr=explode("@",$str); //应用标识“@”分割字符串
print_r($str_arr); //输出字符串分割后的结果
?>

```

从上面的代码中可以看出，在分割分字符串\$str时，以“@”作为分割的标识符进行拆分，分割成4个数组元素，最后应用print\_r()输出函数输出数组中的元素。

运行结果如图6.9所示。

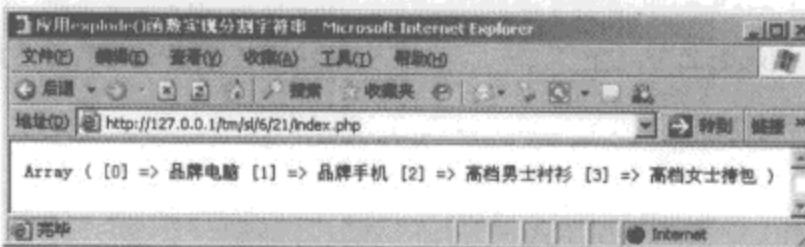


图6.9 应用explode()函数实现分割字符串

注意：默认情况下，数组的第一个元素的索引为0。

输出数组元素除了应用print\_r()函数外，还可以应用echo语句进行输出，两者的区别是，print\_r()函数输出的是一个数组列，而应用echo语句输出的是数组中的元素。将“print\_r(\$str\_arr);”应用如下代码替换即可输出数组中的元素。

```
echo $str_arr[0]; //输出数组中的第一个元素
echo $str_arr[1]; //输出数组中的第二个元素
echo $str_arr[2]; //输出数组中的第3个元素
echo $str_arr[3]; //输出数组中的第4个元素
```

结果为：品牌电脑 品牌手机 高档男士衬衫 高档女士挎包

## 6.14 字符串与HTML相互转换

字符串与HTML之间的相互转换主要应用htmlentities()来完成。htmlentities()函数将所有的字符都转换成HTML字符串。

语法：

```
string htmlentities(string string,[int quote_style],[string charset])
```

htmlentities()函数的参数说明如表6.4所示。

表6.4 htmlentities()函数的参数说明

参 数	说 明
string	必要参数，指定要转换的字符串
quote_style	可选参数，选择如何处理字符串中的引号，有3个可选值：(1) ENT_COMPAT，转换双引号，忽略单引号，它是默认值；(2) ENT_NOQUOTES，忽略双引号和单引号；(3) ENT_QUOTES，转换双引号和单引号
charset	可选参数，确定转换所使用的字符集，默认字符集是“ISO-8859-1”，指定字符集后就能够避免转换中文字符出现乱码的问题

htmlentities()函数支持的字符集如表6.5所示。

表 6.5 htmlentities()函数支持的字符集

字 符 集	说 明
BIG5	繁体中文
BIG5-HKSCS	香港扩展的 BIG5，繁体中文
cp866	DOS 特有的西里尔 (Cyrillic) 字符集
cp1251	Windows 特有的西里尔字符集
cp1252	Windows 特有的西欧字符集
EUC-JP	日文
GB2312	简体中文
ISO-8859-1	西欧，Latin-1
ISO-8859-15	西欧，Latin-9
KOI8-R	俄语
Shift-JIS	日文
UTF-8	ASCII 兼容的多字节 8 编码

例 6.22 在本实例中应用 htmlentities() 函数将字符串转换成 HTML 字符串，并设置转换的字符集为 GB2312，避免中文输出乱码的问题，程序代码如下：（实例位置：光盘\TM\Instances\6\22）

```
<table width="550" height="218" border="1" cellpadding="2" cellspacing="1" bordercolor="#33CC33" bgcolor="#CCFF33">
<tr>
    <td height="38" colspan="2" align="center"><span class="STYLE4">字符串转换成 HTML</span></td>
</tr>
<tr>
    <td width="120" align="right" class="STYLE5">转换的结果为:</td>
    <td width="413" height="100" align="left"><span class="STYLE3">
        <?php
            $str='<table width="300" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#993300">
                <tr>
                    <td align="center" bgcolor="#FFFFFF" class="STYLE3">明日科技</td>
                </tr>
                <tr>
                    <td align="center" bgcolor="#FFFFFF" class="STYLE3">认认真真做事，坦坦荡荡做人！</td>
                </tr>
            </table>';
            echo htmlentities($str,ENT_QUOTES,"GB2312")."<br>";
        ?>
        </span>
    </td>
</tr>
<tr>
    <td align="right" class="STYLE5">不转换输出结果:</td>
    <td height="80" align="center"><span class="STYLE3"><?php echo $str; ?></span></td>
</tr>
</table>
```

运行结果如图 6.10 所示。

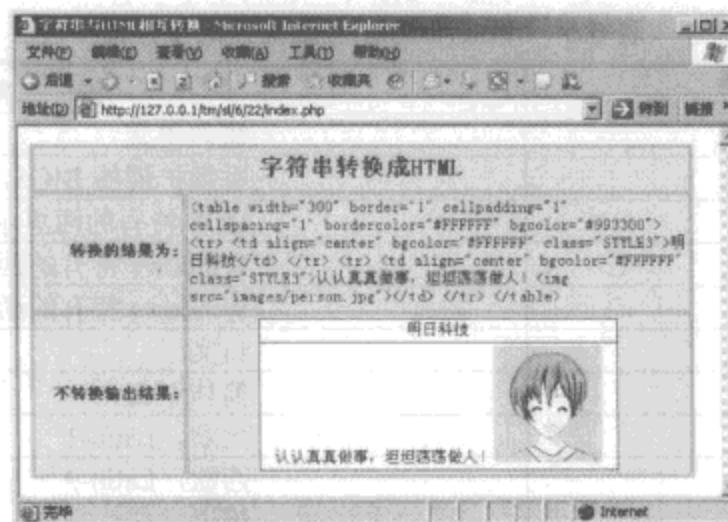


图 6.10 htmlentities()函数的应用

技巧：在应用 htmlentities() 函数对字符串转换成 HTML 字符串的过程中，设置参数 charset 的字符集为 GB2312，即可避免在输出中文字符时出现乱码。

## 6.15 实战

### 6.15.1 超长文本的分页显示

视频讲解：光盘\TM\Video\6\超长文本的分页显示.exe

例 6.23 本实例实现超长文本的分页输出，其中通过 strlen() 函数获取字符串的长度，应用 ceil() 函数将文本文件中的数据分页；通过自定义函数 msubstr() 控制数据的输出；通过 substr() 函数获取到当前页面输出的内容。（实例位置：光盘\TM\Instances\6\23）

在程序开发的过程中，经常会遇到输出超长文本文件的情况，为了使页面的布局更加合理，文本文件的输出更加流畅，采用分页输出文本文件是一个很好的方法。运行本实例，如图 6.11 所示，通过分页来输出超长文本文件。



图 6.11 新闻主题信息显示

(1) 编写 function.php 文件, 创建自定义函数 unhtml()对超长文本中的特殊字符进行替换; 创建自定义函数 msubstr()控制超长文本中数据的输出, 语法如下。

```
<?php
    function unhtml($content){ //定义自定义函数的名称
        $content=htmlspecialchars($content); //转换文本中的特殊字符
        $content=str_replace(chr(13),"<br>",$content); //替换文本中的换行符
        $content=str_replace(chr(32),"&ampnbsp",$content); //替换文本中的&ampnbsp
        $content=str_replace("[","<",$content); //替换文本中的大于号
        $content=str_replace(")_",">",$content); //替换文本中的小于号
        $content=str_replace("|\"," ",$content); //替换文本中的空格
        return trim($content); //删除文本中首尾的空格
    }
    function msubstr($str,$start,$len){ //str 指的是字符串, start 指的是字符串的起始位置, len 指的是长度
        $strlen=$start+$len; //用$strlen 存储字符串的总长度 (从字符串的起始位置到字符串的总长度)
        for($i=0;$i<$strlen;$i++){ //通过 for 循环语句, 循环读取字符串
            if(ord(substr($str,$i,1))>0xa0){ //如果字符串中首个字节的 ASCII 序数值大于 0xa0, 则表示为汉字
                $tmpstr.=substr($str,$i,2); //每次取出两位字符赋给变量$tmpstr, 等于一个汉字
                $i++; //变量自加 1
            }else{ //如果不是汉字, 则每次取出一位字符赋给变量$tmpstr
                $tmpstr.=substr($str,$i,1);
            }
        }
        return $tmpstr; //输出字符串
    }
?>
```

(2) 创建 index.php 页面, 首先定义分页变量 page, 通过 include 包含语句调用 function.php 文件。然后通过 file\_get\_contents() 函数读取根目录下 file 文件夹中 file.txt 文件中的数据, 实现数据的分页输出。其关键代码如下。

```
<?php if ($page=="") {$page=1;} include("function.php");?>
<?php
//读取超长文本中的数据, 实现超长文本中数据的分页显示
if($page){
    $counter=file_get_contents("file/file.txt"); //读取指定文件夹下的文件
    $length=strlen(unhtml($counter)); //获取数据的长度, 应用 unhtml() 函数去除特殊字符
    $page_count=ceil($length/950); //计算共有多少页
    $c=msubstr($counter,0,($page-1)*950); //获取上一页的数据
    $c1=msubstr($counter,0,$page*950); //获取当前页的数据
    echo substr($c1,strlen($c),strlen($c1)-strlen($c)); //输出当前页的数据
}
?>
```

(3) 设置分页使用的超链接, 定义链接的标识, 其代码如下。

```
<tr>
    <td width="202" height="22"><span class="STYLE3">页次: <?php echo $page;?> / <?php echo $page_count;?>
    页 </span></td>
    <td width="278"><span class="STYLE3">分页:
```

```

<?php
if($page!=1){
    echo "<a href=index.php?page=1>首页</a>&nbsp;";
    echo "<a href=index.php?page=".($page-1).">上一页</a>&nbsp;";
}
if($page<$page_count){
    echo "<a href=index.php?page=".($page+1).">下一页</a>&nbsp;";
    echo "<a href=index.php?page=$page_count.>尾页</a>";
}
?>
</span></td>
</tr>

```

## 6.15.2 控制页面中输出字符串的长度

### ■ 视频讲解：光盘\TM\Video\6\控制页面中输出字符串的长度.exe

**例 6.24** 本实例应用 `strlen()` 函数获取字符串的长度，在输出字符串时进行判断，如果字符串超出指定的长度，则使用指定的字符进行替换，并在输出时间字符串时，应用 `str_replace()` 函数将“-”替换为“/”。（实例位置：光盘\TM\Instances\6\24）

在论坛或者电子商务等网站中，经常会输出一些公告信息、最新动态等内容，这些内容都是以标题的形式进行输出，为标题设置超链接，链接到相关内容的详细信息页面。

在输出标题信息时，由于要考虑页面规范化、设计合理，所以要对输出的标题长度进行限制，如果标题的总长度超出指定范围，就需要使用省略号进行替换。运行结果如图 6.12 所示。



图 6.12 控制页面中输出字符串的长度

(1) 创建 `index.php` 页面，首先通过 `include_once` 语句调用数据库连接文件和字符串处理文件，然后执行查询语句，查询出数据表中的记录，最后，输出最新动态的标题和发布时间，并判断如果标题的内容超过 24 个字节，则输出省略号，截取发布时间。其关键代码如下。

```

<?php
    include_once("conn/conn.php"); //调用连接数据库的文件
    include_once("function.php");

```

```

?>
<table width="365" height="22" border="0" align="center" cellpadding="0" cellspacing="0">
<?php
    $sql=mysql_query("select * from tb_new_dynamic order by id desc limit 0,6",$id);
    while($myrow=mysql_fetch_array($sql)){
?>
<tr>
    <td width="20" height="22"><div align="center"></div></td>
    <td width="258" height="22"><a href="new_dynamic.php?id=<?php echo $myrow["id"];?>">
        <?php
            echo unhtml(msubstr($myrow["dynamic_title"],0,24));
            if(strlen($myrow["dynamic_title"])>24){
                echo "...";
            }
        ?>
        </a> </td>
    <td width="87">
        <?php
            echo "<fontcolor=red>[".substr(str_replace("-","/", $myrow[createtime]),0,10)."]</font>";
        ?>
    </td>
</tr>
<tr><td colspan="3"></td></tr>
<?php } ?>
</table>

```

(2) 创建 new\_dynamic.php 页面, 根据超链接中传递的 ID 值, 从数据表中查询出指定的记录, 并输出记录的详细内容。其关键代码如下。

```

<?php
    include_once("conn/conn.php"); //调用连接数据库的文件
    include_once("function.php");
?>
<?php
    $sql=mysql_query("select * from tb_new_dynamic where id='".$_GET[id]."'", $id);
    while($myrow=mysql_fetch_array($sql)){
?>
<td align="left" valign="top" class="STYLE1"><span class="STYLE1">
    <?php echo $myrow[dynamic_title];?> <br><br>
    <?php echo $myrow[dynamic_content];?></span></td>
<?php } ?>

```

## 6.16 本章小结

本章主要讲解了字符串的操作技术, 包括通过单引号、双引号和定界符标识字符串, 字符串的连接、转义和还原, 字符串的截取、比较、检索、替换和分割, 以及获取字符串的长度、格式化字符串、

去除字符串中的空白字符和字符串与 HTML 的相互转换。字符串操作技术对程序的开发虽然不起决定性的作用，但是在对一些细节的处理上却是必不可少的，例如，在搜索引擎模块的查询关键字描红功能的实现中，就必须应用到字符串的替换技术。

## 6.17 实战练习

1. 尝试开发一个页面，验证用户输入的 15 位或 18 位身份证号长度是否正确。（答案位置：光盘\TM\Instances\6\25）

2. 尝试开发一个搜索引擎，对检索到与用户输入查询关键字相匹配的结果进行加粗描红，其效果如图 6.13 所示。（答案位置：光盘\TM\Instances\6\26）

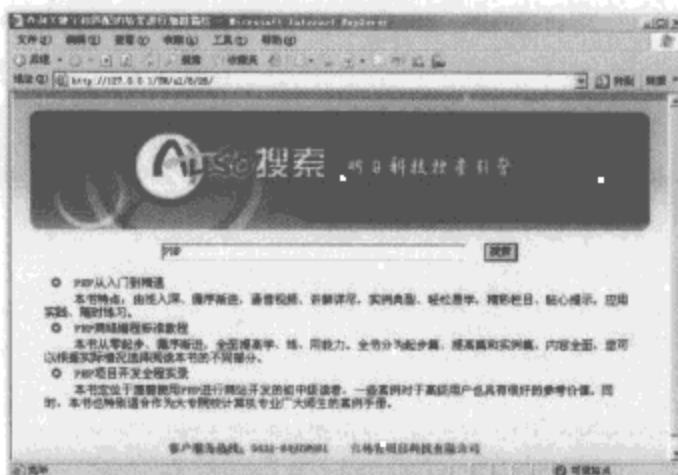


图 6.13 查询关键字描红

3. 尝试开发一个购物车，应用函数 explode() 对拆分购买的各个商品，效果如图 6.14 所示。（答案位置：光盘\TM\Instances\6\27）

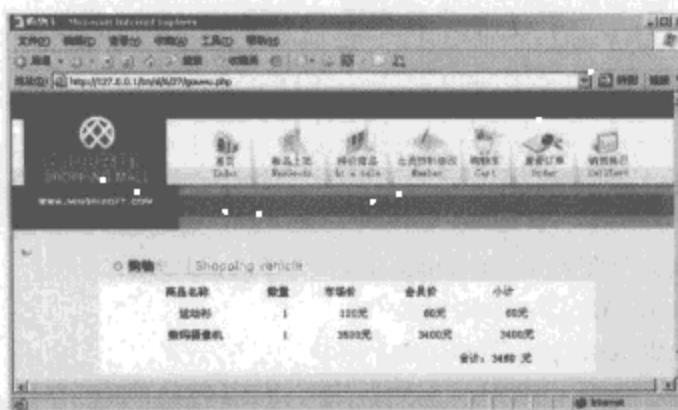


图 6.14 购物车

**提示：**当用户登录到该网站时，系统将为每个用户分配两个 session 变量 \$producelist 和 \$quantity，分别用来存储用户放入购物车中的商品 id 和这些商品的数量，当然一个变量同一时刻只能有一个值，那么如何将多个 id 值同时保存在一个 \$producelist 变量中呢？首先将 id 转换成字符型变量，并且这些变量用字符“@”进行连接，如用户分别将 id 为 1、2、3 的商品放入购物车中，这时 session 变量 \$producelist 的值应该为“1@2@3@”，然后应用 explode() 函数就可以完成 id 的提取，从而实现购物车的开发。

# 第7章

## 正则表达式

### 正则表达式

( 视频讲解: 82 分钟)

在新技术层出不穷的今天，能称得上是伟大的编程技术却寥寥无几，但其中一定会有正则表达式。然而，最容易被人忽略和遗忘的也是正则表达式。一方面，几乎所有的编程语言和文本编辑工具都支持正则表达式；而另一方面，关于正则表达式的书籍、资料却少之又少。本章将重温这个伟大的技术，希望对广大读者的学习能够有所帮助。

通过阅读本章，可以学习到：

- 正则表达式的概念及常用术语
- 正则表达式的语法规则
- PHP 中的 POSIX 函数
- PHP 中的 Perl 函数
- 正则表达式的综合应用

## 7.1 正则表达式是什么

正则表达式是一种描述字符串结构的语法规则，是一个特定的格式化模式。它可以匹配、替换、截取匹配的字串。对于用户来说，可能以前接触过 DOS，如果想匹配当前文件夹下所有的文本文件，可以输入“dir \*.txt”命令，按 Enter 键后，所有“.txt”文件将会被列出来。这里的“\*.txt”就可以理解为一个简单的正则表达式。

## 7.2 正则表达式常用术语

- grep：最初是 ED 编辑器中的一条命令，用来显示文件中特定的内容。后来成为一个独立的工具 grep。
- egrep：grep 虽然不断地更新升级，但仍然无法跟上技术的脚步。为此，贝尔实验室写出了 egrep，意为“扩展的 grep”。这大大增强了正则表达式的能力。
- POSIX（Portable Operating System Interface of UNIX）：可移植操作系统接口。在 grep 发展的同时，其他一些开发人员也根据自己的喜好开发出了具有独特风格的版本。但问题也随之而来。有的程序支持某个元字符，而有的程序则不支持。因此，就有了 POSIX。POSIX 是一系列标准，确保了操作系统之间的移植性。不过 POSIX 和 SQL 一样，没有成为最终的标准而只能作为一个参考。
- Perl（Practical Extraction and Reporting Language）：实际抽取与汇报语言。1987 年，Larry Wall 发布了 Perl。在随后的 7 年时间里，从 Perl1 到现在的 Perl5，最终成为了 POSIX 之后的另一个标准。
- PCRE：Perl 的成功，让其他的开发人员在某种程度上要兼容“Perl”，包括 C/C++、Java、Python 等都有自己的正则表达式。1997 年，Philip Hazel 开发了 PCRE 库，这是兼容 Perl 正则表达式的一套正则引擎，其他开发人员可以将 PCRE 整合到自己的语言中，为用户提供丰富的正则功能。许多软件都使用 PCRE，PHP 正是其中的一员。

## 7.3 正则表达式语法规则

### ■ 视频讲解：光盘\TM\Video\7\正则表达式语法规则.exe

正则表达式由两部分构成：元字符和文本字符。元字符就是具有特殊含义的字符，如前面提到的“\*”和“？”。“？”文本字符就是普通的文本，如字母和数字等。PCRE 风格的正则表达式一般都放置在定界符“/”中间，如“^w+([-.\']w+)\*@\w+([-.\']w+)\*\.\w+([-.\']w+)\*”、“/^http://(www\.)?.+?\\$/”。为了便于读者理解，除个别实例外，本节中的表达式不给出定界符“/”。

### 7.3.1 行定位符 (^和\$)

行定位符用于描述字串的边界。“^”表示行的开始；“\$”表示行的结尾。例如：

`^mr`

该表达式表示要匹配字串 mr 的开始位置是行头，如“mrsoft”、“Mr book”都可以匹配，而“Tomorrow mr”则不能匹配。如果要匹配“Tomorrow mr”，则可以使用下面的行结尾定位符“\$”。

`mr$`

这可以匹配以 mr 结尾的字符，而不能匹配以 mr 开头的字符，如果要匹配的字串可以出现在字符串的任意部分，那么可以直接写成：

`mr`

这样两个字符串就都可以匹配。

### 7.3.2 字符类 ([])

正则表达式是区分大小写的，如果想要忽略大小写，可以通过方括号表达式“[]”来完成。只要匹配的字符出现在方括号内，即表示匹配成功。但需要注意的是，一个方括号只能匹配一个字符。例如，要匹配字串“mr”不区分大小写，其表达式的格式如下。

`[Mm][Rr]`

这样，就可以匹配字串“mr”的所有写法。POSIX 和 PCRE 都使用了一些预定义字符类，但表示方法略有不同。POSIX 风格的预定义字符类如表 7.1 所示。

表 7.1 POSIX 预定义字符类

预定义字符类	说 明
<code>[:digit:]</code>	十进制数字集合。等同于[0-9]
<code>[:alnum:]</code>	字母和数字的集合。等同于[a-zA-Z0-9]
<code>[:alpha:]</code>	字母集合。等同于[a-zA-Z]
<code>[:blank:]</code>	空格和制表符
<code>[:xdigit:]</code>	十六进制数字
<code>[:punct:]</code>	特殊字符集合。包括键盘上的所有特殊字符，如!@#\$等
<code>[:print:]</code>	所有的可打印字符（包括空白字符）
<code>[:space:]</code>	空白字符（空格、换行符、换页符、回车符、水平制表符）
<code>[:graph:]</code>	所有的可打印字符（不包括空白字符）
<code>[:upper:]</code>	所有大写字母。[A-Z]
<code>[:lower:]</code>	所有小写字母。[a-z]
<code>[:cntrl:]</code>	控制字符

而 PCRE 的预定义字符类则是使用反斜线来表示的，详见 7.3.8 节。

### 7.3.3 选择字符 (|)

(234) 正则表达式 1.5.1

要忽略字串的大小写，还可以通过选择字符“|”来完成。该字符可以理解为“或”，如上例也可以写成：

**(M|m)(R|r)**

该表达式的意思是以字母 M 或 m 开头，后面接一个字母 R 或 r。

**说明：**使用“[]”和使用“|”的区别在于“[]”只能匹配单个字符，而“|”可以匹配任意长度的字串。

### 7.3.4 连字符 (-)

变量的命名规则是只能以字母和下划线开头。如果要使用正则表达式来匹配变量名的第一个字母，难道要写成如下格式：

**[a,b,c,d…A,B,C,D…\_]**

不用担心，正则表达式提供了连字符“-”来解决这个问题。连字符可以表示字符的范围。如上例可以写成：

**[a-zA-Z\_-]**

### 7.3.5 排除字符 (^)

上面的例子是匹配符合命名规则的变量。反过来，匹配不符合命名规则的变量。这时该怎么做呢？正则表达式提供了“^”字符。这个元字符在 7.3.1 节中曾经出现过，表示行的开始。这里将其放到方括号中，则表示排除的意思。例如：

**[^a-zA-Z\_-]**

该表达式匹配的是不以字母和下划线开头的变量名。

### 7.3.6 限定符 (? \* + {n,m})

经常使用 google 的用户可能会发现，在搜索结果页的下方，“google”中间字母“o”的个数会随着搜索页的改变而变化。那么要匹配该字串的正则表达式该如何实现呢？

对于这类重复出现的字母或字串，可以使用限定符来实现匹配。限定符主要有 6 种，如表 7.2 所示。

表 7.2 限定符的说明和举例

限 定 符	说 明	举 例
?	匹配前面的字符零次或一次	colou?r，该表达式可以匹配 colour 和 color
+	匹配前面的字符一次或多次	go+gle，该表达式可以匹配的范围从 gogle 到 goo…gle

续表

限 定 符	说 明	举 例
*	匹配前面的字符零次或多次	go*gle, 该表达式可以匹配的范围从 ggle 到 goo...gle
{n}	匹配前面的字符 n 次	go{2}gle, 该表达式只匹配 google
{n,}	匹配前面的字符最少 n 次	go{2,}gle, 该表达式可以匹配的范围从 google 到 goo...gle
{n,m}	匹配前面的字符最少 n 次, 最多 m 次	employe{0,2}, 该表达式可以匹配 employ、employe 和 employee 3 种情况

通过对 google 应用的观察发现，当搜索结果只有一页时，不显示 google 标志；只有大于等于 2 时，才显示 google。这说明字母“o”最少为两个，而最多可以看到 20 个“o”，那么其正则表达式为：

go{2,20}gle

### 7.3.7 点字符(.)

在论坛中，需要对用户发布的帖子进行审核，查看其内容中是否包含一些非法的字符。而过滤其中的非法字符就可以通过正则表达式的点字符“.”来完成。

在正则表达式中点字符“.”可以匹配除换行符外的任意一个字符。注意，是除换行符外的任意的一个字符。

**例 7.1** 在本示例中应用正则表达中的点字符“.”匹配变量中是否包含指定的字符串，程序代码如下：（实例位置：光盘\TM\Instances\7\1）

```
<?php
    $str="吉林省明日科技有限公司";
    if(preg_match("/^明日.科技$/",$str,$counts)){ //判断变量中是否包含明日、科技字符串
        echo "<script>alert('没有指定的关键字!'); history.back();</script>";
    }else{
        echo "<script>alert('存在指定的关键字!'); history.back();</script>";
    }
?>
```

结果为：存在指定的关键字！

### 7.3.8 转义字符 (\)

正则表达式中的转义字符“\”与 PHP 中的大同小异，都是将特殊字符（如“.”、“?”、“\”等）转义为普通字符。

例 7.2 应用正则表达中的转义字符“\”验证 IP 地址的格式是否正确，关键代码如下：（实例位置：光盘\TM\Instances\7\2）

```
<?php  
    $checkip = "/^(25[0-5]|2[0-4][0-9]|1[0-9]{2})|(1-9){1}[0-9]{1}|(1-9)(\.)|(25[0-5]|2[0-4][0-9]|1[0-9]{2})|(1-9){1}[0-9]{1}|(1-9){1}[0-9]{1}(\.)|(25[0-5]|2[0-4][0-9]|1[0-9]{2})|(1-9){1}[0-9]{1}|(1-9){1}[0-9]{1}(\.)|(25[0-5]|2[0-4][0-9]|1[0-9]{2})|(1-9){1}[0-9]{1}|(1-9){1}[0-9]{1}(\.)$/";  
    //编写验证 IP 地址格式的正则表达式
```

```

if($_POST[check_ip]!="" and $_POST[Submit]=="提交"){
    if(preg_match($checkip,$_POST[check_ip],$counts)){
        echo "<script>alert('IP 地址格式正确!');window.location.href='index.php';</script>";
    }else{
        echo "<script>alert('IP 地址格式不正确!');window.location.href='index.php';</script>";
    }
}
?>

```

结果为：验证 127.0.0.1 的格式，输出 IP 地址格式正确！

说明：括号在正则表达式中也算是一个元字符，关于括号的作用参见 7.3.10 节。

### 7.3.9 反斜线 (\)

除了可以做转义字符外，反斜线还有其他一些功能。

反斜线可以将一些不可打印字符显示出来，如表 7.3 所示。

表 7.3 反斜线显示的不可打印字符

字 符	说 明
\a	警报，即 ASCII 中的<BEL>字符 (0x07)
\b	退格，即 ASCII 中的<BS>字符 (0x08)。注意，在 PHP 中只有在中括号 ([ ]) 里使用才表示退格
\e	escape，即 ASCII 中的<ESC>字符 (0x1B)
\f	换页符，即 ASCII 中的<FF>字符 (0x0C)
\n	换行符，即 ASCII 中的<LF>字符 (0x0A)
\r	回车符，即 ASCII 中的<CR>字符 (0x0D)
\t	水平制表符，即 ASCII 中的<HT>字符 (0x09)
\xhh	十六进制代码
\ddd	八进制代码
\cx	即 control-x 的缩写，匹配由 x 指明的控制字符，其中 x 是任意字符

还可以指定预定义字符集，如表 7.4 所示。

表 7.4 反斜线指定的预定义字符集

限 定 符	说 明
\d	任意一个十进制数字，相当于[0-9]
\D	任意一个非十进制数字
\s	任意一个空白字符（空格、换行符、换页符、回车符、水平制表符），相当于[ \f\n\r\t]
\S	任意一个非空白字符
\w	任意一个单词字符，相当于[a-zA-Z0-9_]
\W	任意一个非单词字符

反斜线还有一种功能，就是定义断言，其中已经接触过了\b、\B，其他如表 7.5 所示。

表 7.5 反斜线定义断言的限定符

限 定 符	说 明
\b	单词分界符，用来匹配字符串中的某些位置，\b 是以统一的分界符来匹配
\B	非单词分界符序列
\A	总是能够匹配待搜索文本的起始位置
\Z	表示在未指定任何模式下匹配的字符，通常是字符串的末尾位置，或者是在字符串末尾的换行符之前的位置
\z	只匹配字符串的末尾，而不考虑任何换行符
\G	当前匹配的起始位置

### 7.3.10 括号字符 (( ))

在 7.3.8 节的实例中，读者已经领略了小括号的作用。这里，再通过几个实例来巩固对小括号字符的印象。

小括号字符的第一个作用就是可以改变限定符的作用范围，如“|”、“\*”、“^”等。来看一个表达式：

```
(thir|four)th
```

这个表达式的意思是匹配单词 thirth 或 fourth，如果不使用小括号，那么就变成匹配单词 thir 和 fourth。

小括号的第二个作用是分组，也就是子表达式，如(25[0-5]2[0-4][0-9]1[0-9]{2}][1-9]{1}[0-9]{1}[1-9])就是分组进行重复操作。

### 7.3.11 反向引用

反向引用，就是依靠子表达式的“记忆”功能来匹配连续出现的字串或字母。例如，匹配连续两个“it”，首先将单词“it”作为分组，然后在后面加上“\1”即可。格式为：

```
(it)\1
```

这就是反向引用最简单的格式。如果要匹配的字串不固定，那么就将括号内的字串写成一个正则表达式。如果使用了多个分组，那么可以用“\1”、“\2”来表示每个分组（顺序是从左到右）。例如：

```
([a-z])([A-Z])\1\2
```

除了可以使用数字来表示分组外，还可以自己指定分组名称。格式为：

```
(?P<subname>...)
```

如果想要反向引用该分组，使用如下语法。

```
(?P=subname)
```

下面重写一下表达式([a-z])([A-Z])\1\2，为这两个分组分别命名，并反向引用它们。正则表达式如下。

```
(?P<fir>[a-z])(?P<sec>[A-Z])(?P=fir)(?P=sec)
```

## 7.4 POSIX 扩展正则表达式函数

视频讲解：光盘\TM\Video\7\POSIX 正则表达式.exe

PHP 中实现 POSIX 正则表达式的函数有 7 个。下面就来了解其中几个常用函数。

### 7.4.1 查找字符串

函数格式：

```
bool ereg/eregi ( string pattern, string string [, array regs] )
```

函数功能：在字符串 string 中匹配表达式 pattern，如果匹配成功则返回 true，否则返回 false。如果有第 3 个参数 regs，则将成功匹配的字串按子串（子表达式）划分，并存储到 regs 数组中。ereg() 函数区分大小写，而 eregi() 函数则不区分大小写。

例 7.3 使用 ereg() 函数验证变量是否合法，实例代码如下：（实例位置：光盘\TM\Instances\7\3）

```
<?php
$ereg = '^[$][[:alpha:]]_[[[:alnum:]]]*';
//要匹配的字串
ereg($ereg,'$_string',$register);
//使用 ereg() 函数匹配变量名
var_dump($register);
//显示数组结构
?>
```

结果为：array(1) { [0]=> string(8) "\$\_string" }

### 7.4.2 替换字符串

函数格式：

```
string ereg_replace/eregi_replace ( string pattern, string replacement, string string )
```

函数功能：在字符串 string 中匹配表达式 pattern。如果匹配成功，则使用 replacement 来替换匹配字串，并返回替换后的 string。eregi\_replace() 函数不区分大小写。

例 7.4 将字符串中所有非大写的“mr”都换成大写“MR”，实例代码如下：（实例位置：光盘\TM\Instances\7\4）

```
<?php
$ereg = 'mr';
//要匹配的字串
$str = 'hello,Mr,mr,mR.';
//要查找的文本
$rep_str = eregi_replace($ereg,'MR',$str);
//使用 eregi_replace() 函数进行替换
echo $rep_str;
//输出替换后的文本
?>
```

结果为：hello,MR,MR,MR.

### 7.4.3 分割字符串

函数格式：

```
array split/spli( string pattern, string string [, int limit] )
```

函数功能：使用表达式 pattern 来分割字符串 string。如果有参数 limit，那么数组最多有 limit 个元素，剩余部分都写到最后一个数组元素中。如果函数错误，则返回 false。split() 函数区分大小写，而 spli() 函数不区分大小写。

例 7.5 通过 spli() 函数，使用字串“、”来分割字符串\$str，代码如下：（实例位置：光盘\TM\Instances\7\5）。

```
<?php
    $ereg = ',';
    //分割字符串使用的表达式
    $str = 'JAVA 编程词典、VB 编程词典、VC 编程词典、PHP 编程词典';
    //要被分割的字符串
    $arr_str = spli($ereg,$str);
    //使用 spli() 函数分割字符串
    var_dump($arr_str);
    //显示分割后的数组结构
?>
```

结果为：array(4) { [0]=> string(12) "JAVA 编程词典" [1]=> string(10) "VB 编程词典" [2]=> string(10) "VC 编程词典" [3]=> string(11) "PHP 编程词典" }

## 7.5 PCRE 兼容正则表达式函数

 视频讲解：光盘\TM\Video\7\PCRE 正则表达式.exe

实现 PCRE 风格的正则表达式的函数也有 7 个。但无论从执行效率，还是从语法支持上，PCRE 函数都要略优于 POSIX 函数。下面对其中几个常用的 PCRE 函数进行讲解。

### 7.5.1 查找字符串

函数格式：

```
int preg_match/preg_match_all( string pattern, string subject [, array matches] )
```

函数功能：在字符串 subject 中匹配表达式 pattern。函数返回匹配的次数。如果有数组 matches，那么每次匹配的结果将被存储到数组 matches 中。

函数 preg\_match() 的返回值是 0 或 1。因为该函数在匹配成功后就停止继续查找了。而 preg\_match\_all() 函数则会一直匹配到最后才停止。参数 array matches 对于 preg\_match\_all() 函数是必需的，而对前者则可以省略。

例 7.6 使用 preg\_match() 函数和 preg\_match\_all() 函数验证手机和座机号码的格式是否正确，并返回各自的匹配次数。实例代码如下：（实例位置：光盘\TM\Instances\7\6）

```

<?php
$checktel="/^(\d{3}-)\d{8})$|^\d{4}-(\d{7})$|^(\d{4}-)(\d{8})$/"; //定义验证座机号码的正则表达式
$checkphone="/^13(\d{9})$|^15(\d{9})$/"; //定义验证手机号码的正则表达式
if($_POST[check_tel]!="" and $_POST[Submit]=="提交"){
    if(preg_match($checktel,$_POST[check_tel],$counts)){
        echo "匹配次数: ".var_dump($counts);
        echo "<script>alert('电话号码格式正确!');window.location.href='index.php';</script>";
    }else{
        echo "匹配次数: ".var_dump($counts);
        echo "<script>alert('电话号码格式不正确!');window.location.href='index.php';</script>";
    }
}
if($_POST[check_phone]!="" and $_POST[Submites]=="提交"){
    if(preg_match_all($checkphone,$_POST[check_phone],$counts)){
        echo "匹配次数: ".var_dump($counts);
        echo "<script>alert('手机号码格式正确!');window.location.href='index.php';</script>";
    }else{
        echo "匹配次数: ".var_dump($counts);
        echo "<script>alert('手机号码格式不正确!');window.location.href='index.php';</script>";
    }
}
?>

```

运行结果如图 7.1 所示。



图 7.1 preg\_match()函数和 preg\_match\_all()函数

## 7.5.2 替换字符串

函数语法:

```
mixed preg_replace ( mixed pattern, mixed replacement, mixed subject [, int limit] )
```

函数功能: 该函数在字符串 subject 中匹配表达式 pattern, 并将匹配项替换成字串 replacement。如果有参数 limit, 则替换 limit 次。

**例 7.7** 实现一个过滤留言板中非法字符的功能。这里将“明日科技”作为关键字, 如果在留言板中存在“明日科技”, 则将其替换为“\*\*\*\*”, 实例代码如下: (实例位置: 光盘\TM\Instances\7\7)

```

<?php
    $title=$_POST[title];           //获取提交的标题
    $content=$_POST[content];       //获取提交的内容
    $str="*****";                  //定义替换字符
    $titles = preg_replace("/(明日科技)|(明日)|(科技)/",$str,$title); //验证标题并执行替换
    $contents = preg_replace("/(明日科技)|(明日)|(科技)/",$str,$content); //验证内容并执行替换
?>

```

运行结果如图 7.2 和图 7.3 所示。

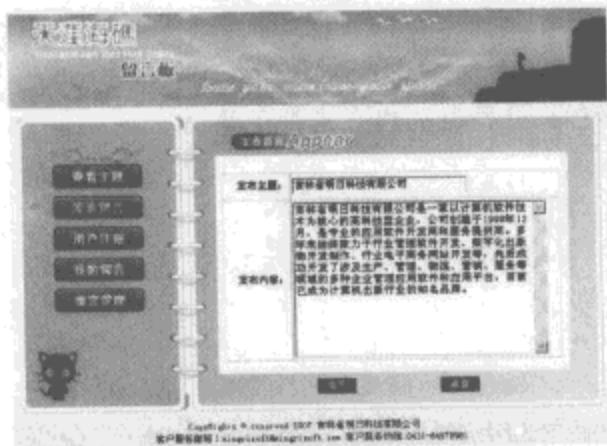


图 7.2 过滤留言板中的非法字符

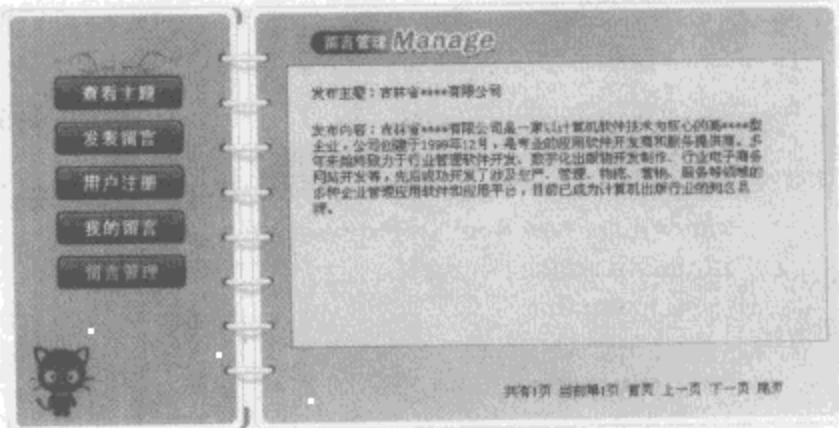


图 7.3 替换留言板中的非法字符

- 说明：**preg\_replace()函数中的字串“\$1”是在正则表达式外调用分组，按照\$1、\$2排列，依次表示从左到右的分组顺序，也就是括号顺序。另外，\$0表示的是整个正则表达式的匹配值。
- 说明：**这里有关 POSIX 扩展正则表达式函数和 PCRE 兼容正则表达式函数的讲解，只是其中的几个，并没有全部给出，读者如果想了解其他函数，可以参考 PHP 手册。

## 7.6 实 战

### 视频讲解：光盘\TM\Video\7\通过正则表达式无刷新验证用户注册信息是否合理.exe

**例 7.8 通过正则表达式无刷新验证用户注册信息是否合理。**正则表达式经常被应用于对用户注册信息的合理性进行判断中，可以对用户输入的邮编、电话号码、邮箱地址和网址的格式进行判断。本实例中应用正则表达式和 JavaScript 脚本，实现无刷新判断用户输入信息的格式是否正确。（实例位置：光盘\TM\Instances\7\8）

(1) 本实例中将正则表达式定义到自定义函数中，然后在 JavaScript 脚本中调用自定义函数，执行对用户输入信息的判断。JavaScript 脚本文件 check.js 的代码如下。

```

function checkregtel(regtel){                                //通过正则表达式验证手机号码的格式是否正确
    var str=regtel;
    var Expression=/^13(\d{9})\$|^15(\d{9})\$/;      //定义正则表达式
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
        return true;
    }else{
        return false;
    }
}

```

```

    }

function checkregtels(regtels){          //自定义函数，验证座机号码的格式是否正确
    var str=regtels;
    var Expression=/^(\d{3}-)(\d{8})$|^(\d{4}-)(\d{7})$|^(\d{4}-)(\d{8})$/; //定义正则表达式
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
        return true;
    }else{
        return false;
    }
}

function checkregemail(emails){          //自定义函数，验证 E-mail 地址的格式是否正确
    var str=emails;
    var Expression=/\w+([-.\w+]*)@\w+([-.\w+]*)\.\w+([-.\w+]*)/;           //定义正则表达式
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
        return true;
    }else{
        return false;
    }
}

function chkreginfo(form,mark){          //自定义函数，对表单中用户提交的数据进行判断
//这里省略了部分代码
    if(mark==4 || mark=="all"){          //验证 E-mail 地址的格式是否正确
        if(form.email.value==""){        //如果输入值为空
            chknew_email.innerHTML="请输入 E-mail 地址！"; //则返回“请输入 E-mail 地址！”
            form.email.style.backgroundColor="#FF0000"; //定义返回提示信息的样式
            return false;
        }else if(!checkregemail(form.email.value)){ //调用自定义函数验证 E-mail 地址格式
            chknew_email.innerHTML="邮箱地址的格式不正确！";
            form.email.style.backgroundColor="#FF0000";
            return false;
        }else{
            chknew_email.innerHTML="";
            form.email.style.backgroundColor="#FFFFFF"; //当返回值是空时
            //定义样式为白色
        }
    }
    if(mark==5 || mark=="all"){          //验证手机号码的格式
        if(form.mtel.value==""){
            chknew_mtel.innerHTML="请输入电话号码！";
            form.mtel.style.backgroundColor="#FF0000";
            return false;
        }else if(!checkregtel(form.mtel.value)){ //调用自定义函数验证手机号码格式是否正确
            chknew_mtel.innerHTML="电话号码的格式不正确！";
            form.mtel.style.backgroundColor="#FF0000";
            return false;
        }else if(isNaN(form.mtel.value)){
            chknew_mtel.innerHTML="电话号由数字组成！";
            form.mtel.style.backgroundColor="#FF0000";
            return false;
        }else{
    }
}

```

```

        chknew_mtel.innerHTML="";
        form.mtel.style.backgroundColor="#FFFFFF";
    }
}

//省略了部分代码
}

```

(2) 正则表达式和 JavaScript 脚本文件创建完成后，接下来开始创建 index.php 文件，添加 form 表单，调用 JavaScript 脚本文件，提交用户注册信息。其关键代码如下。

```

<link rel="stylesheet" type="text/css" href="css/style.css">
<script src="js/check.js"></script>
<form name="form_reg" method="post" action="index_ok.php" onSubmit="return chkreginfo(form_reg,'all')">
<table width="620" height="262" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr><td width="120" height="30"><div align="right">用户名: </div></td>
        <td colspan="3">&ampnbsp
            <input type="text" name="recuser" size="20" class="inputcss" onBlur="chkreginfo(form_reg,0)">
            <div id="chknew_recuser" style="color:#FF0000"></div>
        </td>
    </tr>
    <!--省略了部分代码-->
    <tr><td height="30"><div align="right">E-mail: </div></td>
        <td height="30" colspan="3">&ampnbsp
            <input type="text" name="email" size="20" class="inputcss" onBlur="chkreginfo(form_reg,4)">
            <div id="chknew_email" style="color:#FF0000"></div>
        </td>
    </tr>
    <!--省略了部分代码-->
    <tr><td height="30">&ampnbsp</td>
        <td height="30" colspan="2">
            <input type="image" src="images/form (2).jpg">
            
        </td>
    </tr>
</table>
</form>

```

(3) 创建 index\_ok.php 文件，输出用户提交的数据。运行结果如图 7.4 所示。

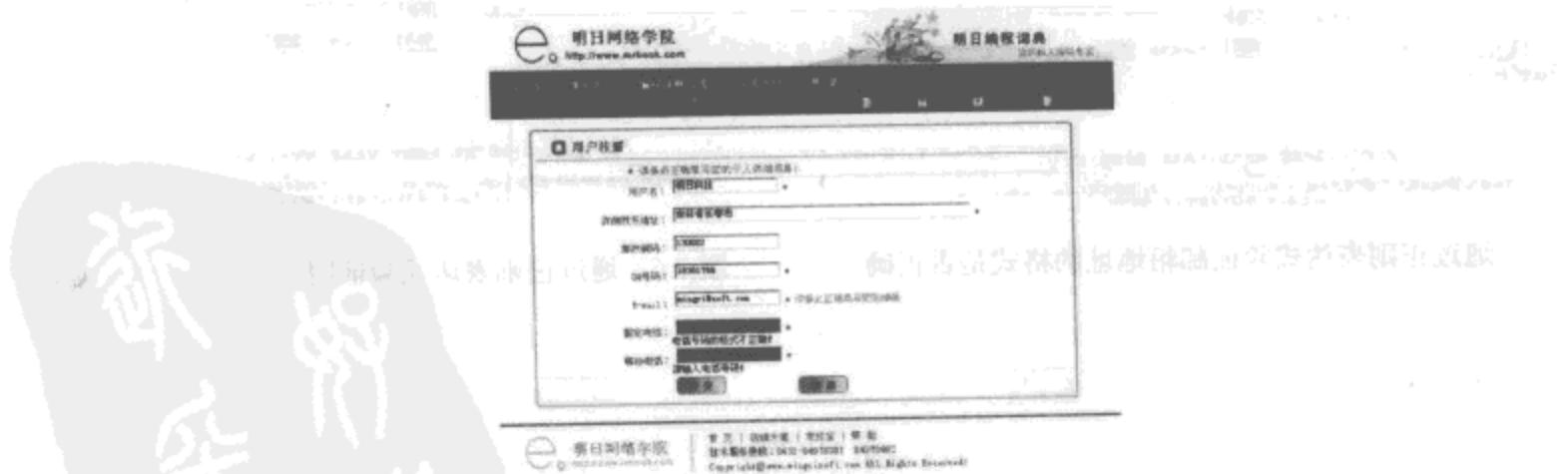


图 7.4 通过正则表达式无刷新验证用户注册信息

## 7.7 本章小结

本章介绍了什么是正则表达式，正则表达式的常用术语，以及正则表达式的常用语法规则，并且介绍了正则表达式的两种主要风格和相关的 PHP 函数；另外，通过示例、实战和实战练习对正则表达式 的实际应用进行了讲解。

如今网络发展迅速，信息的获取变得十分容易，而我们的学习似乎在走向一个误区：当遇到难以解决的问题时，不去认真思考解决的方案，而是直接借助网络寻求解决的办法，这虽然是一个非常不错的方法，但同时也造成了我们思想上的懒惰，不利于对知识的掌握。希望初学者不要走入这个误区，一定要认真地学习基础知识，积极地思考，不要一味依赖于网络。

## 7.8 实战练习

1. 通过正则表达式验证邮箱地址的格式是否正确。运行结果如图 7.5 所示。（答案位置：光盘\TM\Instances\7\9）
2. 通过正则表达式验证网址的格式是否正确。运行结果如图 7.6 所示。（答案位置：光盘\TM\Instances\7\10）

您当前的位置：会员注册 >

用户名： 用户名长度限制为0-30字符  
密码：  
密码确认：  
性别： 女  男  
出生日期：  
真实姓名：  
联系电话：  
身份证号码：  
个人空间：  
E-mail： \* 请输入正确的邮箱地址  
喜欢编码：  
联系地址：  
 (\*为必填项目)

联系方式 | 退换货原则 | 配送范围 | 交易条款 | 退货协议 | 后台入口  
吉林省明日科技有限公司 客户服务热线：0431-84972266 客户服务邮箱：mingri@mingrisoft.com  
Copyright © 明日科技有限公司 网站开发

图 7.5 通过正则表达式验证邮箱地址的格式是否正确

您当前的位置：会员注册 >

用户名： 用户名长度限制为0-30字符  
密码：  
密码确认：  
性别： 女  男  
出生日期：  
真实姓名：  
联系电话：  
身份证号码：  
个人空间： \* 请输入正确的空间地址  
E-mail：  
喜欢编码：  
联系地址：  
 (\*为必填项目)

联系方式 | 退换货原则 | 配送范围 | 交易条款 | 退货协议 | 后台入口  
吉林省明日科技有限公司 客户服务热线：0431-84972266 客户服务邮箱：mingri@mingrisoft.com  
Copyright © 明日科技有限公司 网站开发

图 7.6 通过正则表达式验证网址的格式是否正确

# 第 8 章

第六步：数组

## PHP 数组

( 视频讲解：90分钟)

数组提供了一种快速、方便地管理一组相关数据的方法，是 PHP 程序设计中的重要内容。通过数组可以对大量性质相同的数据进行存储、排序、插入及删除等操作，从而可以有效地提高程序开发效率及改善程序的编写方式。PHP 作为市面上最为流行的 Web 开发语言之一，凭借其代码开源、升级速度快等特点，对数组的操作能力更为强大，尤其为程序开发人员提供了大量方便、易懂的数组操作函数，更使 PHP 深受广大 Web 开发人员的青睐。

通过阅读本章，可以学习到：

- ■ 什么是数组
- ■ 如何声明一维数组和二维数组
- ■ 如何输出数组
- ■ 如何遍历数组
- ■ 如何合并数组
- ■ 如何进行字符串与数组之间的转换
- ■ 如何统计数组元素个数
- ■ 如何对数组进行排序

## 8.1 数组是什么

数组是一组数据的集合，将数据按照一定规则组织起来，形成一个可操作的整体。在这个可操作的整体中包含多个变量。变量中保存单个数据，而数组中则保存的是多个变量的集合，如图 8.1 所示。

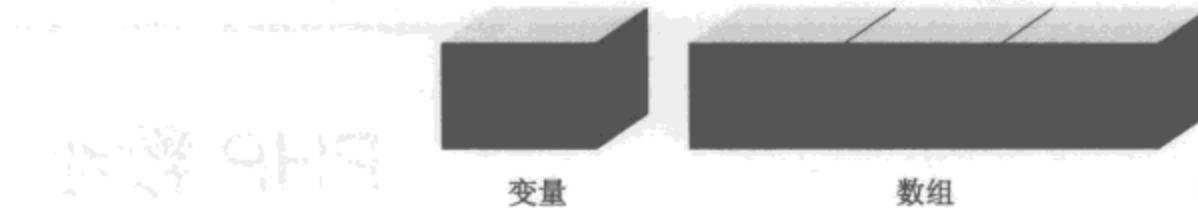


图 8.1 变量中保存单个数据，数组中保存多个变量

数组中的每个实体都包含两项：键和值。其中键可以是数字、字符串或者数字和字符串的组合，用于标识数组中相应的值；而值被称为数组中的元素，可以定义为任意数据类型，甚至是混合类型。最终通过键来获取相应的值。例如，一个足球队通常会有几十个人，但是认识他们的时候首先会把他们看作是某队的成员，然后再通过他们的号码来区分每一名队员，这时候，球队就是一个数组，而号码就是数组的下标（键），当指明是几号队员的时候就找到了这名队员（值）。

PHP 数组比许多其他高级语言中的数组更加灵活，不但支持数字索引数组，而且支持以字符串或字符串、数字混合为键名的关联数组。而在其他高级语言中，如 Java 或者 C++ 等语言的数组，只支持数字索引数组。PHP 数组结构与其他语言数组结构的比较如图 8.2 和图 8.3 所示。

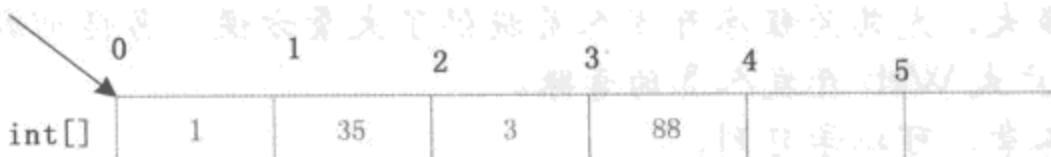


图 8.2 C++ 或者 Java 的数组结构

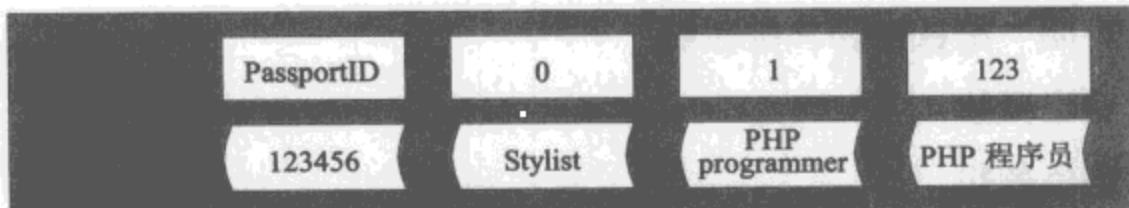


图 8.3 PHP 的数组结构

## 8.2 声明数组

**视频讲解：**光盘\TM\Video\8\声明数组.exe

PHP 中声明数组的方式主要有两种：一种是应用数组函数声明数组；另一种是通过数组标识符[] 声明数组。

## 8.2.1 数组命名规则

PHP 中声明数组的规则：数组的名称由一个“\$”（美元符号）开始，第一个字符是字母或下划线，其后是任意数量的字母、数字或下划线。例如：

```
$array_name=array('PHP'=>'php','JAVA'=>'java','JSP'=>'jsp'); //以字符串作为数组索引，指定关键字
$_aa=array('PHP','Java','Jsp','Vb'); //以数字作为数组索引，从 0 开始，没有指定关键字
$array[]='value1';
```

上述都是符合命名规则的数组，而下面的这两个数组则不符合命名规则。

```
//$1="array(1=>'a',2=>'b',3=>'c',4=>'d')"; //不可以以数字开头
$@=array('PHP'=>'php','JAVA'=>'java','JSP'=>'jsp'); //不可以使用特殊字符
```

在同一个程序中，标量变量和数组变量都不能重名。例如，如果已经存在一个名称为\$string 的变量，而又创建一个名称为\$string 的数组，那么前一个变量就会被覆盖。

数组的名称是区分大小写的，如\$String 与\$string 是不同的。

## 8.2.2 通过 PHP 函数创建数组

在 PHP 中，可以创建数组的函数如表 8.1 所示。这里主要讲解通过 array() 函数创建数组。

表 8.1 创建数组的函数

函 数	功 能
array()	创建一个数组（创建数组最常用的函数）
array_combine()	以一个数组为关键字、另一个数组为值建立新数组
array_fill()	用值填充一个数组
array_pad()	用一个值把数组填充到指定长度
compact()	创建包含变量及其值的数组
range()	创建一个数组包含一定范围内的元素

应用 array() 函数声明数组的方式如下：

```
array array ( [mixed ...])
```

参数 mixed 的语法为 “key => value”，多个参数 mixed 用逗号分开，分别定义了索引和值。索引可以是字符串或数字。如果省略了索引，会自动产生从 0 开始的整数索引。如果索引是整数，则下一个产生的索引将是目前最大的整数索引+1。如果定义了两个完全一样的索引，则后面一个会覆盖前一个。数组中的各数据元素的数据类型可以不同，也可以是数组类型。当 mixed 是数组类型时，就是二维数组。

应用 array() 函数声明数组时，数组下标既可以是数值索引也可以是关联索引。下标与数组元素值之间用 “=>” 进行连接，不同数组元素之间用逗号进行分隔。

应用 array() 函数定义数组比较灵活，可以在函数体中只给出数组元素值，而不必给出键值。

**例 8.1** 通过 array() 函数声明数组，并输出数组中的值，代码如下：（实例位置：光盘\TM\Instances\8\1）

```
<?php
$array_name=array('PHP'=>'php','JAVA'=>'java','JSP'=>'jsp'); //以字符串作为数组索引，指定关键字
print_r($array_name); //输出数组
echo "<br>"; //换行
echo $array_name[JAVA]; //输出数组中的索引为 JAVA 的元素
echo "<br>";
$_aa=array('PHP','Java','Jsp','Vb'); //以数字作为数组索引，从 0 开始，没有指定关键字
print_r($_aa); //输出整个数组
echo "<br>";
echo $_aa[0]; //输出数组中的第一个元素
?>
```

运行结果如图 8.4 所示。

**◆ 注意：**可以通过给变量赋予一个没有参数的 array() 来创建空数组，然后通过使用方括号 “[ ]” 语法来添加值。

PHP 提供创建数组的 array() 语言结构。在使用其中的数据时，可以直接应用它们在数组中的排列顺序取值，这个顺序称为数组的下标。例如：

```
<?php
echo $_aa[ 0 ]; //输出数组元素的第 0 个下标值
?>
```

结果为：PHP

**◆ 注意：**应用这种方式定义数组时，下标默认是从 0 开始的，而不是 1，然后依次增 1。所以为 0 的元素是指数组的第一个元素。

### 8.2.3 通过数组标识符 “[ ]” 创建数组

PHP 中另一种比较灵活的数组声明方式是通过数组标识符 “[ ]” 直接为数组元素赋值。如果在创建数组时不知所创建数组的大小，或在实际编写程序时数组的大小可能发生改变，采用这种数组创建的方法较好。

**例 8.2** 为了加深这种数组声明方式的理解，下面通过具体实例对这种数组声明方式进行讲解，代码如下：（实例位置：光盘\TM\Instances\8\2）

```
<?php
$array[0] = "PHP";
$array[1] = "编";
$array[2] = "程";
$array[3] = "词";
$array[4] = "典";
```

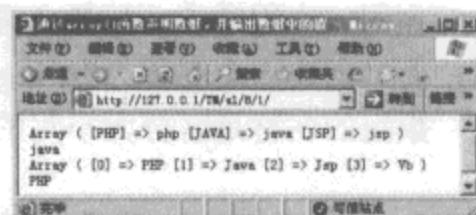


图 8.4 通过 array() 函数声明数组

```
print_r($array); //输出所创建数组的结构
?>
```

结果为：Array ([0] => PHP [1] => 编程 [2] => 程序 [3] => 词典 )

 注意：通过直接为数组元素赋值的方式声明数组时，要求同一数组元素中的数组名相同。

## 8.3 数组的类型

 视频讲解：光盘\TM\Video\8\数组的类型.exe

PHP 支持两种数组：数字索引数组（indexed array）和关联数组（associative array），数字索引数组使用数字作为键（如图 8.5 所示），关联数组使用字符串作为键（如图 8.6 所示）。

### 8.3.1 数字索引数组

PHP 数字索引由数字组成，下标从 0 开始，数值索引一般表示数组元素在数组中的位置。数字索引数组默认索引值从数字 0 开始，不需要特别指定，PHP 会自动为索引数组的键名赋一个整数值，然后从这个值开始自动增量，当然，也可以指定从某个位置开始保存数据。

数组可以构造成一系列键-值(key-value)对，其中每一对都是那个数组的一个项目或元素(element)。对于列表中的每个项目，都有一个与之关联的键(key)（或索引(index)），如图 8.5 所示。



图 8.5 数字索引数组

### 8.3.2 关联数组

关联数组的键名可以是数值和字符串混合的形式，而不像数字索引数组的键名只能为数字，在一个数组中，只要键名中有一个不是数字，那么这个数组就叫做关联数组。

关联数组（associative array）使用字符串索引（或键）来访问存储在数组中的值，如图 8.6 所示。关联索引的数组对于数据库层交互非常有用。



图 8.6 关联数组

**例 8.3 创建一个关联数组，代码如下：**（实例位置：光盘\TM\Instances\8\3）

```
<?php
$array = array("first"=>"PHP","second"=>"JAVA","third"=>"JSP");
```

```

echo $array["second"];           //输出索引为 second 的元素的值
$array["third"] = "WEB";        //为索引为 third 的元素重新赋值
echo $array["third"];           //输出索引为 third 的元素的值
?>

```

结果为： JAVA WEB

**技巧：**关联数组的键名可以是任何一个整数或字符串。如果键名是一个字符串，不要忘记给这个键名或索引加上个定界修饰符：单引号（'）或双引号（"）。对于数字索引数组，为了避免不必要的麻烦，最好也加上定界符。

## 8.4 输出数组

### 视频讲解：光盘\TM\Video\8\输出数组.exe

PHP 中对数组元素进行输出可以通过输出语句来实现，如 echo 语句、print 语句等，但应用这种输出方式只能对某数组中某一元素进行输出。而通过 print\_r() 和 var\_dump() 函数可以将数组结构进行输出。

print\_r() 函数的语法如下：

```
bool print_r( mixed expression )
```

如果该函数的参数 expression 为普通的整型、字符型或实型变量，则输出该变量本身；如果该参数为数组，则按一定键值和元素的顺序显示出该数组中的所有元素。

例 8.4 通过 print\_r() 函数输出数组的结构，代码如下：（实例位置：光盘\TM\Instances\8\4）

```

<?php
$array=array(1=>"PHP",2=>"网络编程",3=>"标准教程");
print_r($array);
?>

```

结果为： Array ( [1] => PHP [2] => 网络编程 [3] => 标准教程 )

var\_dump() 函数可以输出数组（或对象）、元素数量以及每个字符串的长度。还能够以缩进方式输出数组或对象的结构。

语法：

```
void var_dump( mixed expression [,mixed expression [...]] )
```

例 8.5 通过 var\_dump() 函数输出数组的结构，代码如下：（实例位置：光盘\TM\Instances\8\5）

```

<?php
$array=array("PHP 典型模块","PHP 从入门到精通","PHP 网络编程标准教程");
var_dump($array);
$arrayss=array('first'=>"PHP 典型模块",'second'=>"PHP 从入门到精通",'third'=>"PHP 网络编程标准教程");
var_dump($arrayss);
?>

```

运行结果如图 8.7 所示。

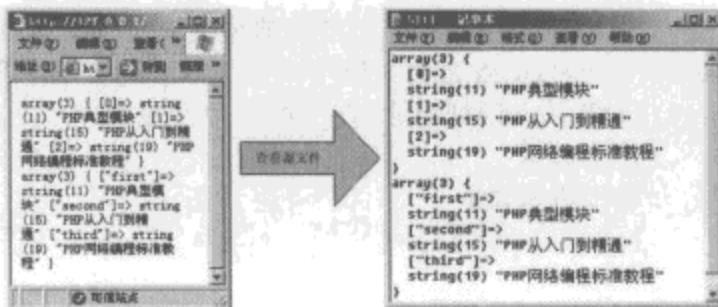


图 8.7 通过 var\_dump() 函数输出数组的结构

## 8.5 数组的构造

视频讲解：光盘\TM\Video\8\数组的构造.exe

### 8.5.1 创建一维数组

当一个数组的元素是变量时，称这个数组为一维数组。一维数组是最普通的数组，它只保存一列内容。

**例 8.6** 创建两个一维数组，一个数字索引数组，另一个是关联数组，代码如下：（实例位置：光盘\TM\Instances\8\6）

```

<?php
$array=array("PHP 典型模块","PHP 从入门到精通","PHP 网络编程标准教程");
print_r($array);
$arrays=array('first'=>"PHP 典型模块",'second'=>"PHP 从入门到精通",'third'=>"PHP 标准教程");
print_r($arrays);
?>

```

运行结果如图 8.8 所示。

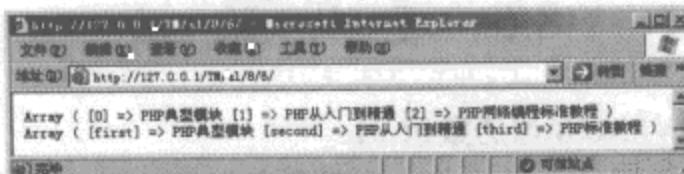


图 8.8 创建并输出一维数组

### 8.5.2 创建二维数组

数组元素本身仍是数组的数组被称为二维数组。二维数组的定义和使用与一维数组相同，唯一的区别是二维数组的元素仍然是数组。

**例 8.7** 声明一个二维数组，代码如下：（实例位置：光盘\TM\Instances\8\7）

```

<?php
$str = array (

```

```

"PHP 类图书"=>array ("PHP 从入门到精通","PHP 典型模块","PHP 标准教程"),
"JAVA 类图书"=>array ("a"=>"JAVA 范例手册","b"=>"JAVA WEB 范例宝典"),
"ASP 类图书"=>array ("ASP 从入门到精通",2=>"ASP 范例宝典","ASP 典型模块")
);
//声明数组
print_r ($str);
//输出数组元素
?>

```

结果为：

Array(

    [PHP 类图书] => Array(

        [0] => PHP 从入门到精通

        [1] => PHP 典型模块

        [2] => PHP 标准教程

    )

    [JAVA 类图书] => Array(

        [a] => JAVA 范例手册

        [b] => JAVA WEB 范例宝典

    )

    [ASP 类图书] => Array(

        [0] => ASP 从入门到精通

        [2] => ASP 范例宝典

        [3] => ASP 典型模块

    )

)

上面的代码实现一个二维数组的声明，按照同样的思路，可以创建更高维数的数组，如三维数组或四维数组等。

## 8.6 遍历数组

### 视频讲解：光盘\TM\Video\8\遍历数组.exe

遍历数组中的所有元素是常用的一种操作，在遍历的过程中可以完成查询或其他功能。如去商场购物，如果想要买一台电脑，就需要在商场中逛一遍，看是否有满意的机型，而逛商场的过程就相当于遍历数组的操作。在 PHP 中遍历数组的方法有多种，下面介绍最常用的 4 种方法。

#### 8.6.1 foreach 结构遍历数组

遍历数组元素最常用的方法是使用 `foreach` 结构。`foreach` 结构并非操作数组本身，而是操作数组的一个备份。

 说明：有关 `foreach` 循环的详细讲解可参考第 4 章流程控制语句的 4.3.4 节。

例 8.8 通过 foreach 结构遍历数组中的数据，代码如下：（实例位置：光盘\TM\Instances\8\8）

```
<?php
    $name=array('编程词典网','编程体验网','编程资源网'); //声明数组
    $url = array('0'=>'www.mrbcccd.com','1'=>'www.bcty365.com','2'=>'www.bc110.com'); //声明数组
    foreach ($name as $key=>$value) { //遍历数组
        echo $value."——".$url[$key].'  
<br>';
    }
?>
```

运行结果如图 8.9 所示。



图 8.9 通过 foreach 结构遍历数组中的数据

### 8.6.2 each()函数遍历数组

返回数组中当前指针位置的键名和对应的值，并向前移动数组指针。键值对被返回为 4 个单元的数组，键名为 0、1、key 和 value。单元 0 和 key 包含数组单元的键名，1 和 value 包含数据，如果内部指针越过了数组的末端，则函数返回 false。

语法：

```
array each ( array array)
```

参数 array 为输入的数组。

### 8.6.3 list()函数遍历数组

把数组中的值赋给一些变量。与 array() 函数类似，这不是真正的函数，而是语言结构。list() 函数仅能用于数字索引的数组，且数字索引从 0 开始。

语法：

```
void list ( mixed ...)
```

参数 mixed 为被赋值的变量名称。

例 8.9 通过实例讲解应用 each() 函数和 list() 函数的结合，输出存储在数组中的用户登录信息。（实例位置：光盘\TM\Instances\8\9）

具体开发步骤如下：

(1) 应用开发工具(如 Dreamweaver), 新建一个 PHP 动态页, 命名为 index.php。

(2) 应用 HTML 标记设计页面, 首先建立用户登录表单, 用于提交用户登录信息, 然后应用 each() 函数提取全局数组\$\_POST 中的内容, 并最终应用 while 循环输出用户所提交的注册信息, 代码如下。

```
<!-- ----- 定义用户登录表单 ----- -->
<form name="form1" method="post">
    <table width="233" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td width="62" height="24" align="right" bgcolor="#FFFFFF"><span class="STYLE3">用户名: </span></td>
            <td width="189" height="24" bgcolor="#FFFFFF"><input name="username" type="text" id="user" size="18"></td>
        </tr>
        <tr>
            <td height="24" align="right" bgcolor="#FFFFFF"><span class="STYLE3">密 &nbsp;&nbsp;码: </span></td>
            <td height="24" bgcolor="#FFFFFF"><input name="pwd" type="password" id="pwd" size="18"></td>
        </tr>
        <tr align="center" bgcolor="#CCFF33">
            <td height="24" colspan="2" bgcolor="#FFFFFF">
                <input type="image" name="imageField" src="images/bg1.JPG">&nbsp;&nbsp;
                <input type="image" name="imageField2" src="images/bg2.JPG" onClick="form.reset();return false;"></td>
            </tr>
        </table>
    </form>
<?php
    while(list($name,$value)=each($_POST)){ //输出用户登录信息
        if($name!="imageField_x" and $name!="imageField_y"){
            echo "$name=$value<br>";
        }
    }
?>
```

在 IE 浏览器中输入地址, 按 Enter 键, 输入用户名及密码, 单击“提交”按钮, 运行结果如图 8.10 所示。



图 8.10 应用 list() 函数获取用户登录信息

说明：each()函数用于返回当前指针位置的数组值，并将指针推进一个位置。返回的数组包含4个键，键0和key包含键名，而键1和value包含相应的数据。如果程序在执行each()函数时，指针已经位于数组末尾，则返回false。

#### 8.6.4 for语句遍历数组

想要通过for循环遍历数组，首先必须应用count()函数，获取到数组中单元数目，然后将数组中单元数目作为for循环的条件，执行循环输出。

count()函数语法：

```
int count ( mixed array [, int mode])
```

参数array为指定输入的数组；参数mode为可选参数，COUNT\_RECURSIVE（或1），如选中此参数，本函数将递归地对数组计数。这对计算多维数组的所有单元尤其有用。此参数的默认值为0。

有关for循环语句的详细讲解可参考第4章流程控制语句的4.3.3节，这里不再赘述。

例8.10 本实例应用count()函数和for语句循环输出数组中的数据，代码如下：（实例位置：光盘\TM\Instances\8\10）

```
<?php
$name=array('编程词典网','编程体验网','编程资源网'); //声明数组
$url = array('0'=>'www.mrbccd.com','1'=>'www.bcty365.com','2'=>'www.bc110.com'); //声明数组
for($i=0; $i<count($name); $i++) { //根据数组中元素个数，循环遍历数组
    echo $name[0]."——".$url[0].'  
<br><br>';
}
?>
```

运行结果如图8.11所示。



图8.11 应用count()函数和for语句循环输出数组中的数据

### 8.7 合并数组

合并数组是把一个数组追加到另一个数组中。主要应用array\_merge()函数实现。

语法：

```
array array_merge (array array1,array array2 [,array...]);
```

在合并时，如果输入的数组中有相同的字符串键名，则后面的值将覆盖前面的值；如果数组包含数字键名，后面的值不会覆盖原来的值，而是附加到后面。

**例 8.11 应用 array\_merge() 函数合并两个数组，代码如下：**（实例位置：光盘\TM\Instances\8\11）

```
<?php
$str1 = array ("图书"=>"PHP 标准教程");                                //声明数组
$str2 = array ("网络类","定价","图书"=>"PHP 从入门到精通","PHP"=>"95","元");    //声明数组
$result = array_merge ($str1,$str2);                                         //合并数组
print_r ($result);                                                       //输出数组
?>
```

在上述实例中，\$str1 中字符串键“图书”的值在合并时被\$str2 中相同键值的“PHP 从入门到精通”覆盖；而在数字键中，\$str2 中的“网络类”、“定价”和“元”，则会直接合并到\$str1 中，其下标会自动增长。

结果为：

Array ([图书] => PHP 标准教程 [0] => 网络类 [1] => 定价 [PHP] => 95 [2] => 元 )

 **说明：**array\_merge\_recursive() 函数的功能与 array\_merge() 函数相似，也用来实现合并数组，但是它可以保留同时出现在两个数组中相同字符键值上的元素。如上例中的“PHP 标准教程”和“PHP 从入门到精通”。

## 8.8 字符串与数组的转换

 **视频讲解：**光盘\TM\Video\8\字符串与数组的转换.exe

字符串与数组的转换在程序开发过程中经常使用，主要应用 explode() 函数和 implode() 函数来实现。下面分别进行详细讲解。

### 8.8.1 应用 explode() 函数将字符串转换成数组

explode() 函数将字符串依指定的字符串或字符 separator 分隔。

语法：

```
array explode(string separator, string string, [int limit])
```

返回由字符串组成的数组，每个元素都是 string 的一个子串，它们被字符串 separator 作为边界点分隔出来。如果设置 limit 参数，则返回的数组包含最多 limit 个元素，而最后那个元素将包含 string 的剩余部分；如果 separator 为空字符串（""），explode() 函数将返回 false；如果 separator 所包含的值在 string 中找不到，那么 explode() 函数将返回包含 string 单个元素的数组；如果参数 limit 是负数，则返回除了最后的-limit 个元素外的所有元素。

**例 8.12 应用 explode() 函数以“、”为分隔符对指定的字符串进行拆分，代码如下：**（实例位置：光盘\TM\Instances\8\12）

```
<?php
$str = "PHP 从入门到精通、PHP 项目开发全程实录、PHP 典型模块"; //定义一个字符串
$strs = explode("、", $str); //应用 explode()函数将字符串转换成数组
print_r($strs); //输出数组元素
?>
```

结果为：Array ([0] => PHP 从入门到精通、[1] => PHP 项目开发全程实录、[2] => PHP 典型模块)

## 8.8.2 应用 implode()函数将数组转换成一个新字符串

将数组的内容组合成一个字符串。

语法：

```
string implode(string glue, array pieces)
```

参数 glue 是字符串类型，指定分隔符。参数 pieces 是数组类型，指定要被合并的数组。

**例 8.13 应用 implode()函数将数组中的内容以空格做分隔符进行连接，从而组合成一个新的字符串，代码如下：**（实例位置：光盘\TM\Instances\8\13）

```
<?php
$array=array("PHP 从入门到精通","PHP 项目开发全程实录","PHP 典型模块");
$string=implode(" ",$array); //以空格作为分隔符将数组中的元素组合成一个新字符串
echo $string
?>
```

结果为：PHP 从入门到精通 PHP 项目开发全程实录 PHP 典型模块

## 8.9 统计数组元素个数

视频讲解：光盘\TM\Video\8\统计数组元素的个数.exe

在 PHP 中，应用 count() 函数对数组中的元素个数进行统计。

语法：

```
int count ( mixed array [, int mode])
```

count() 函数的参数说明如表 8.2 所示。

表 8.2 count() 函数的参数说明

参    数	说    明
array	必要参数。输入的数组
mode	可选参数。COUNT_RECURSIVE (或 1)，如选中此参数，本函数将递归地对数组计数。对计算多维数组的所有单元尤其有用。此参数的默认值为 0

**例 8.14 应用 count() 函数统计存储在数组中的 PHP 类图书的数量，并输出统计结果，代码如下：**（实例位置：光盘\TM\Instances\8\14）

```
<?php
$array = array("PHP 从入门到精通","PHP 标准教程","PHP 典型模块","PHP 项目开发全程实录","PHP 网络编程自学手册");
$count = count($array); //统计数组元素的个数
echo "PHP 类图书数量: ".$count; //输出统计结果
?>
```

结果为，

PHP 类图书数量: 5

## 8.10 数组排序

PHP 提供了以多种不同方式对数组进行排序的函数，这些函数允许用户在数组内部对元素进行排列。通过排序可以对数据进行合理的管理，提高程序的执行效率。下面对常用的数组排序函数进行讲解。

### 8.10.1 数字数组排序

#### 1. 应用 sort() 函数对数字数组按字母顺序进行升序排序

sort() 函数用于对数字数组按字母顺序进行升序排序。在指定特殊参数时，也可以按数值排序。字母顺序，也被称为词典顺序，是使用 ASCII 比较序列根据传统的“字典顺序”进行排序。大写字母在小写字母之前，数字和标点符号位于它们中间。

sort() 函数根据数组元素值以升序进行排序。当数组被排序后，索引会被重置。该函数用于以数字为索引的数组，而 asort() 函数用于关联数组。排序成功后返回 true，否则返回 false。

语法：

```
bool sort( array &array [, int sort_flags] )
```

参数 array 为必选参数，用于指定数组；参数 sort\_flags 为可选参数，用于修改该函数的默认行为，具体说明如表 8.3 所示。

表 8.3 sort() 函数中 sort\_flags 参数说明

sort_flags 参数取值	说 明
SORT_NUMERIC	把元素值作为数值进行比较
SORT_REGULAR	按照普通方式比较元素值（不改变其类型）
SORT_STRING	把元素值当作字符串进行比较
SORT_LOCALE_STRING	基于当前区域把元素值当作字符串进行比较

注意：本函数将为 &array 中的单元赋予新的键名，并删除原有的键名进行重新排序。

例 8.15 应用 sort() 函数对数组进行升序排序（最低到最高）。首先应用 array() 函数创建数组\$array，然后使用 sort() 函数对\$array 数组进行升序排列，最后应用 for 循环输出排序后的数组元素，代码如下：

（实例位置：光盘\TM\Instances\8\15）

```
<?php
$array=array(1,4,8,25,0,42,88,66);
$array1=sort($array);           //用 sort()函数对$array 数组进行升序排序，并返回新数组$array1
for($i=0;$i<count($array);$i++){
    //应用 for 循环语句输出数组元素
    echo $array[$i]."\n";
}
?>
```

结果为： 0 1 4 8 25 42 66 88

 注意： sort() 函数区分大小写，所有大写字母都在小写字母的前面。这个函数还有第二个参数，可以传递 SORT\_NUMERIC、SORT\_STRING 或者 SORT\_REGULAR（默认值）。指定排序功能是非常有用的，例如，当要比较数字 12 和 2 时，从字符串来看，12 要小于 2，而从数字角度看正好相反。

**例 8.16** 应用 sort() 函数，设置第二个参数 SORT\_STRING，把元素值当作字符串进行比较。首先通过 array() 函数定义数组\$array，然后设置 sort() 函数的参数 SORT\_STRING，对\$array 数组按字符串进行排列，最后通过 for 循环输出排序后的数组元素，代码如下：（实例位置：光盘\TM\Instances\8\16）

```
<?php
$array=array(1,4,8,25,0,42,88,66);
$array1=sort($array,SORT_STRING); //用 sort()函数对$array 数组按字符串进行排序
for($i=0;$i<count($array);$i++){
    //应用 for 循环语句输出数组元素
    echo $array[$i]."\n";
}
?>
```

这里定义的数组与例 8.15 中的相同，但排序的结果却发生了变化，结果为：

0 1 25 4 42 66 8 88

## 2. 应用 rsort() 函数对数组进行降序排序

与 sort() 函数相反， rsort() 函数对数组进行逆向排序（最高到最低），并重建数组的索引。

语法：

**void rsort ( array array [, int sort\_flags] )**

rsort() 函数的参数说明如表 8.4 所示。

表 8.4 rsort() 函数的参数说明

参    数	说    明
array	必要参数。输入的数组
sort_flags	可选参数。可改变排序的行为，排序类型标记： SORT_REGULAR（按照普通方式比较元素值） SORT_NUMERIC（把元素当作数值进行比较） SORT_STRING（把元素当作字符串进行比较）

**例 8.17** 应用 rsort() 函数对输入的数组进行逆向排序，代码如下：（实例位置：光盘\TM\Instances\8\17）

```
<?php
    $array = array ("p"=>"php","a"=>"asp","j"=>"jsp");
    //声明数组
    $arrays=rsort($array);
    //对数组元素进行降序排列
    for($i=0;$i<count($array);$i++){
        echo $array[$i]."&nbsp;&nbsp;";
        //应用 for 循环语句输出数组元素
    }
?>
```

## 8.10.2 关联数组排序

如果使用的是关联数组，想要对其进行排序，并且还要保持关键字与值之间的对应关系，那么就需要使用 asort()、arsort()或者 ksort()函数。这 3 个函数都可以对关联数组进行排序，并且保持关键字与值之间的对应关系。只是它们的侧重点有所不同。

### 1. 应用 ksort()函数对关联数组的关键字进行排序

ksort()函数根据关键字对数组进行排序，并且保持关键字与值之间的对应关系。操作成功后返回 true，失败后返回 false。

语法：

```
int ksort ( array array [, int sort_flags])
```

ksort()函数的参数说明可以参考表 8.4。示例如下：

```
<?php
$array = array('项目开发全程实录'=>1,'典型模块'=>2,'范例宝典'=>3);
ksort($array);
print_r($array);
?>
```

结果为：Array ([典型模块] => 2 [范例宝典] => 3 [项目开发全程实录] => 1 )

### 2. 应用 asort()函数对关联数组的元素值进行排序

asort()函数对关联数组进行排序，并且保持关键字与值之间的对应关系。操作成功时返回 true，否则返回 false。asort()函数主要用于更看重元素值顺序的关联数组。

语法：

```
void asort ( array array [, int sort_flags])
```

该函数的参数说明可以参考表 8.4。示例如下：

```
<?php
$array = array('项目开发全程实录'=>1,'典型模块'=>2,'范例宝典'=>3);
asort($array);
print_r($array);
?>
```

结果为：Array ([项目开发全程实录] => 1 [典型模块] => 2 [范例宝典] => 3 )

### 3. 应用 arsort()函数对关联数组的元素值进行逆向排序

arsort()函数根据元素值对数组进行逆向排序，并且保持关键字与数值之间的对应关系。该函数主要用于重视对应关系的关联数组。操作成功时返回 true，失败返回 false。

语法：

```
void arsort ( array array [, int sort_flags])
```

该函数的参数说明可以参考表 8.4。

## 8.11 实战

**例 8.18** 在开发一个投票管理系统时，经常需要在后台添加投票选项到投票系统，作为投票的内容。下面应用 explode()函数对添加的投票选项通过“\*”进行区分，然后通过 while 循环语句输出添加的投票选项。（实例位置：光盘\TM\Instances\8\18）

具体开发步骤如下：

- (1) 应用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php。
- (2) 应用 HTML 标记设计页面，首先创建投票选项提交的表单，然后应用 each()函数提取全局数组\$\_POST 中的内容，最后应用 while 循环输出投票选项内容。其关键代码如下。

```
<?php
if($_POST[Submit]!=""){
    //判断提交的值不为空
    $content=$_POST[content];
    //获取提交的内容
    $data=explode("*",$content);
    //通过 explode()函数以“*”分隔提交的内容
    while(list($name,$value)=each($data)){
        //循环输出数组中的数据
        echo '<input type="checkbox" name="checkbox" value="checkbox">';
        echo $value."\n";
    }
}
?>
```

在 IE 浏览器中输入地址，按 Enter 键，输入投票选项的内容，各选项间用“\*”进行分隔，单击“提交”按钮，运行结果如图 8.12 所示。

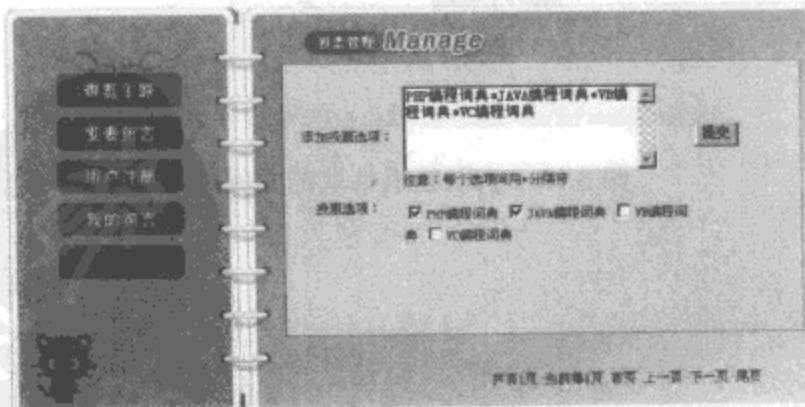


图 8.12 在投票管理中应用 explode()函数

若想向数组元素中添加键值对使用`array[]`或`array[键名]`

## 8.12 本章小结

本章中对数组的讲解，结合了笔者的实践经验，主要讲解了数组中常用的操作方法，包括遍历数组、字符串与数组的转换和统计数组元素个数，并且配合典型的实例，便于加深对函数的理解。还对数组的定义、类型和构造进行了介绍。注意，这里介绍的只是 PHP 数组中的部分函数，有关 PHP 数组中其他函数的介绍，读者可以参考 PHP 中文手册。

## 8.13 实战练习

- 尝试声明一个一维数组和一个二维数组，并对数组元素进行输出。（答案位置：光盘\TM\Instances\8\19）
- 尝试开发一个页面，应用 `list()` 函数和 `each()` 函数将存储在数组中的图书重新排序。排序前后的结果比较如图 8.13 和图 8.14 所示。（答案位置：光盘\TM\Instances\8\20）

php标准教程	68人
php从入门到精通	23人
php网络编程自学手册	56人
php项目开发全程实录	55人

图 8.13 排序前的结果

php标准教程	68人
php网络编程自学手册	56人
php项目开发全程实录	55人
php从入门到精通	23人

图 8.14 排序后的结果

- 尝试开发一个页面，应用 `explode()` 函数以“,”为分隔符，将字符串转换成数组，然后合成一个新数组，并从新数组中随机抽取 3 个元素值，最后输出随机抽取的元素，运行结果如图 8.15 所示。（答案位置：光盘\TM\Instances\8\21）

MingRi	
图书列表	
<a href="#">php网络编程自学手册</a>	
php项目开发全程实录	55
php从入门到精通	23
<a href="#">随机抽取三本书</a>	

图 8.15 获取随机抽取的元素值

# 第 9 章

## PHP 与 Web 页面交互

( 视频讲解: 124 分钟)

PHP 与 Web 页面交互是学习 PHP 语言编程的基础。在 PHP 中提供了两种与 Web 页面交互的方法，一种是通过 Web 表单提交数据，另一种是通过 URL 参数传递。

本章将详细讲解 PHP 与 Web 页面交互的相关知识，为以后学习 PHP 语言编程做好铺垫。

通过阅读本章，可以学习到：

- 表单及表单元素
- 在 Web 页中插入表单的过程
- 获取表单数据的两种方法
- PHP 传参的常用方法
- 对 URL 传递参数编码和解码技术
- 如何在 Web 页中插入 PHP 脚本
- 各种获取表单数据的操作
- PHP 与 Web 表单的综合应用

## 9.1 表 单

### 视频讲解：光盘\TM\Video\9\表单.exe

Web 表单的功能是让浏览者和网站有一个互动的平台。Web 表单主要用来在网页中发送数据到服务器，如提交注册信息时需要使用表单。当用户填写完信息后执行提交（submit）操作，于是将表单中的数据从客户端的浏览器传送到服务器端，经过服务器端 PHP 程序进行处理后，再将用户所需要的信息传递回客户端的浏览器上，从而获得用户信息，使 PHP 与 Web 表单实现交互。

#### 9.1.1 创建表单

使用<form>元素，在 HTML 标记间插入表单元素，即可创建一个表单。

表单结构：

```
<form name="form_name" method="method" action="url" enctype="value" target="target_win" id="id">
    .....
    //省略插入的表单元素
</form>
```

<form>标记的属性如表 9.1 所示。

表 9.1 <form>标记的属性

<form>标记属性	说 明
name	表单的名称
method	设置表单的提交方式，GET 或者 POST 方法
action	指向处理该表单页面的 URL（相对位置或者绝对位置）
enctype	设置表单内容的编码方式
target	设置返回信息的显示方式，target 的属性值如表 9.2 所示
id	表单的 ID 号

表 9.2 target 属性值

属性 值	说 明
blank	将返回信息显示在新的窗口中
parent	将返回信息显示在父级窗口中
self	将返回信息显示在当前窗口中
top	将返回信息显示在顶级窗口中

例如，创建一个表单，以 POST 方法提交到数据处理页 check\_ok.php，代码如下。

```
<form name="form1" method="post" action="check_ok.php">
    .....
    //省略插入的表单元素
</form>
```

在使用 form 表单时，必须要指定其行为属性 action，它指定表单提交数据的处理页。GET 方法是

将表单内容附加在 URL 地址后面；POST 方法是将表单中的信息作为一个数据块发送到服务器上的处理程序中，在浏览器的地址栏不显示提交的信息。method 属性默认为 GET 方法。

## 9.1.2 认识表单元素

表单（form）由表单元素组成。常用的表单元素有以下几种标记：输入域标记<input>、选择域标记<select>和<option>、文本域标记<textarea>等。下面分别进行介绍。

### 1. 输入域标记<input>

输入域标记<input>是表单中最常用的标记之一。常用的文本域、按钮、单选按钮及复选框等构成了一个完整的表单。

语法：

```
<form>
  <input name="filed_name" type="type_name">
</form>
```

参数 name 是指输入域的名称，参数 type 是指输入域的类型。在<input type="">标记中一共提供了 10 种类型的输入区域，用户所选择使用的类型由 type 属性决定。type 属性取值及举例如表 9.3 所示。

表 9.3 type 属性取值及举例

值	举 例	说 明	运 行 结 果
text	<input name="user" type="text" value="纯净水" size="12" maxlength="1000">	name 为文本框的名称，value 是文本框的默认值，size 指文本框的宽度（以字符为单位），maxlength 指文本框的最大输入字符数	添加一个文本框： <input type="text" value="纯净水"/>
hidden	<input type="hidden" name="ddh">	隐藏域，用于在表单中以隐含方式提交变量值。隐藏域在页面中对于用户而言是不可见的，添加隐藏域的目的在于通过隐藏的方式收集或者发送信息。浏览者单击“发送”按钮发送表单时，隐藏域的信息也被一起发送到 action 指定的处理页	添加一个隐藏域： <input name="ddh" type="hidden"/>
password	<input name="pwd" type="password" value="666666" size="12" maxlength="20">	密码域，用户在该文本框中输入字符时将被替换显示为*号，起到保密作用	添加一个密码域： <input type="password" value="*****"/>
file	<input name="file" type="file" enctype="multipart/form-data" size="16" maxlength="200">	文件域，当文件上传时，可用来打开一个模式窗口以选择文件。然后将文件通过表单上传到服务器，如上传 Word 文档等各种类型的文件。但是需要注意的是，上传文件时需要指明表单的属性 enctype="multipart/form-data" 才可以实现上传功能	添加一个文件域： <input name="file" type="file"/>
image	<input name="imageField" type="image" src="images/banner.gif" width="120" height="24" border="0">	图像域是指可以用在提交按钮位置上的图片，这幅图片具有按钮的功能	添加一个图像域： 

续表

值	举 例	说 明	运 行 结 果
radio	<input name="sex" type="radio" value="1" checked>男 <input name="sex" type="radio" value="0">女	单选按钮，用于设置一组选择项，用户只能选择一项。checked 属性用来设置单选按钮默认值	添加一组单选按钮（如“您的性别为：”） 男 女
checkbox	<input name="checkbox" type="checkbox" value="1" checked>封面 <input name="checkbox" type="checkbox" value="1" checked>正文内容 <input name="checkbox" type="checkbox" value="0">价格	复选框，允许用户选择多个选择项。checked 属性用来设置该复选框默认值。例如，收集个人信息时，要求在个人爱好的选项中进行多项选择等	添加一组复选框（如“影响您购买本书的因素：”） 封面 正文内容 价格
submit	<input type="submit" name="Submit" value="提交">	将表单的内容提交到服务器端	添加一个“提交”按钮： 提交
reset	<input type="reset" name="Submit" value="重置">	清除与重置表单内容，用于清除表单中所有文本框的内容，而且使选择菜单项恢复到初始值	添加一个“重置”按钮： 重置
button	<input type="button" name="Submit" value="按钮">	按钮可以激发提交表单的动作，可以在用户需要修改表单的时候，将表单恢复到初始的状态，还可以依照程序的需要，发挥其他作用。普通按钮一般是配合 JavaScript 脚本来进行表单的处理	添加一个普通按钮： 按钮

## 2. 选择域标记<select>和<option>

通过选择域标记<select>和<option>可以建立一个列表或者菜单。菜单节省空间，正常状态下只能看到一个选项，单击按钮打开菜单后才能看到全部的选项。列表可以显示一定数量的选项，如果超出了这个数量，会自动出现滚动条，浏览者可以通过拖动滚动条来查看各选项。

语法：

```
<select name="name" size="value" multiple>
    <option value="value" selected>选项 1</option>
    <option value="value">选项 2</option>
    <option value="value">选项 3</option>
    ...
</select>
```

参数 name 表示选择域的名称；参数 size 表示列表的行数；参数 value 表示菜单选项值；参数 multiple 表示以菜单方式显示数据，省略则以列表方式显示数据。

选择域标记<select>和<option>的显示方式及举例如表 9.4 所示。

表9.4 选择域标记&lt;select&gt;和&lt;option&gt;的显示方式及举例

显示方式	举例	说明	运行结果
列表方式	<pre>&lt;select name="spec" id="spec"&gt;     &lt;option value="0" selected&gt;网络编程&lt;/option&gt;     &lt;option value="1"&gt;办公自动化&lt;/option&gt;     &lt;option value="2"&gt;网页设计&lt;/option&gt;     &lt;option value="3"&gt;网页美工&lt;/option&gt; &lt;/select&gt;</pre>	下拉列表框，通过选择域标记<select>和<option>建立一个列表，列表可以显示一定数量的选项，如果超出了这个数量，会自动出现滚动条，浏览者可以通过拖动滚动条来查看各选项。selected属性用来设置该菜单项时默认被选中	<p>请选择所学专业：</p>
菜单方式	<pre>&lt;select name="spec" id="spec" multiple&gt;     &lt;option value="0" selected&gt;网络编程&lt;/option&gt;     &lt;option value="1"&gt;办公自动化&lt;/option&gt;     &lt;option value="2"&gt;网页设计&lt;/option&gt;     &lt;option value="3"&gt;网页美工&lt;/option&gt; &lt;/select&gt;</pre>	multiple 属性用于下拉列表框<select>标记中，指定该选项时用户可以使用 Ctrl 和 Shift 键进行多选	<p>请选择所学专业：</p>

说明：上面的表格中给出了静态菜单项的添加方法，而在 Web 程序开发过程中，也可以通过循环语句动态添加菜单项。

### 3. 文本域标记<textarea>

文本域标记<textarea>用来制作多行的文本域，可以在其中输入更多的文本。

语法：

```
<textarea name="name" rows="value" cols="value" value="value" wrap="value">
    ...文本内容
</textarea>
```

参数 name 表示文本域的名称；rows 表示文本域的行数；cols 表示文本域的列数（这里的 rows 和 cols 以字符为单位）；value 表示文本域的默认值。wrap 用于设定显示和送出时的换行方式，值为 off 表示不自动换行；值为 hard 表示自动按 Enter 键换行，换行标记一同被发送到服务器，输出时也会换行；值为 soft 表示自动按 Enter 键换行，换行标记不会被发送到服务器，输出时仍然为一列。

文本域标记<textarea>的值及举例如表 9.5 所示。

表9.5 文本域标记&lt;textarea&gt;的值及举例

值	举例	说明	运行结果
textarea	<pre>&lt;textarea name="remark" cols="20" rows="4" id="remark"&gt; 请输入您的建议! &lt;/textarea&gt;</pre>	文本域，也称多行文本框，用于多行文本的编辑 默认 wrap 属性时，默认为自动换行方式	

## 9.2 在 Web 页中创建表单

视频讲解：光盘\TM\Video\9\在 Web 页中创建表单.exe

例 9.1 以“用户注册”模块为例，详细讲解在 Web 页中创建用户注册表单的方法。（实例位置：光盘\TM\Instances\9\1）

具体操作步骤如下：

- (1) 创建 index.php 文件，设计用户注册的页面。
- (2) 在表格中添加一个名称为 form1 的表单。在表单中添加表单元素，代码如下。

```
<form action="index.php" method="post" name="form1" enctype="multipart/form-data">
<table width="405" height="24" bordercolor="#FFFFFF" bgcolor="#990066">
    <tr>
        <td width="93" height="25" align="right" bgcolor="#FF99FF">用户名： </td>
        <td width="299" height="25" align="left" bgcolor="#FF99FF">
            <input name="user" type="text" id="user" size="20" maxlength="100">
        </td>
    </tr>
    <tr>
        <td height="25" align="right" bgcolor="#FF99FF">性别： </td>
        <td height="25" colspan="2" align="left" bgcolor="#FF99FF">
            <input name="sex" type="radio" value="男" checked>男
            <input type="radio" name="sex" value="女">女
        </td>
    </tr>
    <tr>
        <td width="93" height="25" align="right" bgcolor="#FF99FF">密码： </td>
        <td width="299" height="25" colspan="2" align="left" bgcolor="#FF99FF">
            <input name="pwd" type="password" id="pwd" size="20" maxlength="100">
        </td>
    </tr>
    <tr>
        <td height="25" align="right" bgcolor="#FF99FF">学历： </td>
        <td height="25" colspan="2" align="left" bgcolor="#FF99FF">
            <select name="select">
                <option value="高中">高中</option>
                <option value="专科">专科</option>
                <option value="本科" selected>本科</option>
            </select>
        </td>
    </tr>
    <tr>
        <td height="25" align="right" bgcolor="#FF99FF">爱好： </td>
        <td height="25" colspan="2" align="left" bgcolor="#FF99FF">
            <input name="fond[]" type="checkbox" id="fond[]"/>写作
            <input name="fond[]" type="checkbox" id="fond[]"/>音乐
        </td>
    </tr>
    <tr>
        <td height="25" align="right" bgcolor="#FF99FF">个人头像： </td>
        <td height="25" colspan="2" align="left" bgcolor="#FF99FF">
            <input name="photo" type="file" size="28" maxlength="1000" id="photo">
        </td>
    </tr>
    <tr>
        <td height="25" align="right" bgcolor="#FF99FF">个人简介： </td>
    </tr>
</table>
</form>
```

```

<td height="25" colspan="2" align="left" bgcolor="#FF99FF">
    <textarea name="intro" cols="35" rows="4" id="intro"></textarea>
</td>
</tr>
<tr align="center">
    <td height="35" colspan="3" bgcolor="#FF99FF">
        <input type="submit" name="submit" value="注册">
        <input type="reset" name="submit2" value="重写">
    </td>
</tr>
</table>
</form>

```

**注意：**在一个 Web 页面中，允许有多个表单，在编写代码时，由表单的 name 和 ID 属性值对各个表单进行区分。

**注意：**由于该页未使用 PHP 脚本，因此该文件属于静态页，可以将其存储为.html 格式，然后直接应用浏览器打开该文件查看运行结果即可。

(3) 在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 9.1 所示。

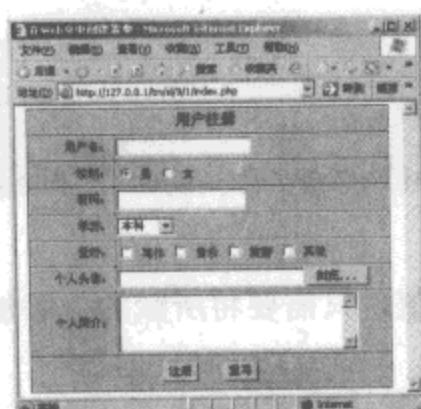


图 9.1 在 Web 页中创建表单

### 9.3 在 Web 页中嵌入 PHP 脚本

#### ■ 视频讲解：光盘\TM\Video\9\在 Web 页中嵌入 PHP 脚本.exe

在 Web 页中嵌入 PHP 脚本的方法有两种：一种是直接在 HTML 标记中添加“<?php?>” PHP 标记符，写入 PHP 脚本；另二种是对表单元素的 value 属性进行赋值。下面分别对这两种方法进行讲解。

#### 9.3.1 在 HTML 标记中嵌入 PHP 脚本

在 Web 编码过程中，通过在 HTML 标记中添加 PHP 脚本标记“<?php?>”来嵌入 PHP 脚本，两个标记之间的所有文本都会被解释为 PHP 语言，而标记之外的任何文本都会被认为是普通的 HTML。例如，在<body>标记中添加 PHP 标识符，应用 echo 语句输出字符串到浏览器，代码如下。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>在 HTML 标记中嵌入 PHP 脚本</title>
</head>
<body>
<?php
    echo "欢迎您学习 PHP 语言！";
?>
</body>
</html>
```

### 9.3.2 为表单元素赋值

在 Web 程序开发过程中，为了使表单元素在运行时有默认值，通常需要对表单元素的 value 属性进行赋值。下面通过具体的实例讲解赋值的方法。

例如，对表单元素文本框进行赋值，只需要将所赋的值添加到文本框所对应的 value 属性中即可，代码如下。

```
<?php
$name="明日科技";                                //为变量$name 赋值
?>
用户名: <input name="name" type="text" id="name"  value="<?php echo $name;?>" />
```

运行结果如图 9.2 所示。

例如，对表单元素隐藏域进行赋值，只需要将所赋的值添加到 value 属性后即可，代码如下。



图 9.2 为文本框赋值

```
<?php
$hidden="yg0025";                                //为变量$hidden 赋值
?>
隐藏域的值 :<input type="hidden" name="HID"  value="<?php echo $hidden;?>" />
```

从上面代码中可以看出，首先为变量\$hidden 赋初始值，然后将变量\$hidden 的值赋给隐藏域。在程序开发过程中，经常要用到隐藏域来存储一些不必要显示的信息或需要传送的参数，隐藏域的值在程序运行过程中并不可见，为了看到效果，可以通过 echo 语句将隐藏域的 name 属性值进行输出，如“echo \$\_POST["HID"];”。

## 9.4 提交表单数据

### ■ 视频讲解：光盘\TM\Video\9\提交表单数据.exe

提交表单数据有两种方法：POST 方法和 GET 方法。采用哪种方法提交表单数据由<form>表单的 method 属性值决定。下面详细讲解这两种提交表单数据的方法。

## 9.4.1 POST方法提交表单数据

应用 POST 方法提交表单数据的方法非常简单，只需要将<form>表单中的 method 属性值设置成“POST”即可。POST 方法不依赖于 URL，不会将传递的参数值显示在地址栏中。另外，POST 方法可以没有限制地传递数据到服务器，所有提交的信息在后台传输，用户在浏览器端是看不到这一过程的，安全性高。所以，POST 方法比较适合用于发送一个保密的（如信用卡号）或者比较大量的数据到服务器。

**例 9.2 应用 POST 方法提交表单信息到服务器，代码如下：**（实例位置：光盘\TM\Instances\9\2）

```
<form name="form1" method="post" action="index.php">
<table width="300" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td height="30">&nbsp;&nbsp;订单号:</td>
        <td><input type="text" name="user" size="20" >
            <input type="submit" name="submit" value="提交">
        </td>
    </tr>
</table>
</form>
```

在上面的代码中，form 表单的 method 属性指定 POST 方法为传递方式，并通过 action 属性指定了数据处理页为 index.php，因此，当单击“提交”按钮后，提交文本框的信息到服务器。运行结果如图 9.3 所示。



图 9.3 使用 POST 方法提交表单数据

## 9.4.2 GET方法提交表单数据

GET 方法是<form>表单中 method 属性的默认方法。使用 GET 方法提交的表单数据被附加到 URL 上，并作为 URL 的一部分发送到服务器端。在程序的开发过程中，由于 GET 方法提交的数据是附加到 URL 上发送的，因此，在 URL 的地址栏中将会显示“URL+用户传递的参数”。

GET 方法的传参格式如下：

http://url?name1=value1&name2=value2.....  
 URL      参数 1      参数 2，也称查询字符串

其中，url 为表单响应地址（如 127.0.0.1/index.php）。name1 为表单元素的名称，value1 为表单元

素的值。url 和表单元素之间用“?”隔开，而多个表单元素之间用“&”隔开，每个表单元素的格式都是“name=value”固定不变。

**◆ 注意：**若要使用 GET 方法发送表单，URL 的长度应限制在 1MB 字符以内。如果发送的数据量太大，数据将被截断，从而导致意外或失败的处理结果。

**例 9.3** 创建一个名称为 form1 的表单，指定 method 的属性值为“GET”。添加一个文本框，命名为“user”，添加一个密码域，命名为“pwd”，将表单中的数据提交到 index.php 文件，代码如下：  
(实例位置：光盘\TM\Instances\9\3)

```
<form name="form1" method="get" action="index.php">
<table width="352" border="0" cellpadding="0" cellspacing="0">
<tr>
    <td width="352" height="30">&nbsp;&nbsp;用户名:<br/>
        <input name="user" type="text" size="12" > 密码:<br/>
        <input name="pwd" type="password" id="pwd" size="12">
        <input type="submit" name="submit" value="提交">
    </td>
</tr>
</table>
</form>
```

运行本实例，在文本框中输入用户名和密码，单击“提交”按钮，文本框中的信息就会显示在 URL 地址栏中，如图 9.4 所示。

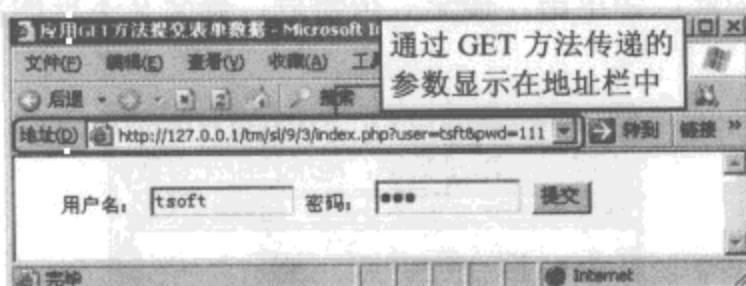


图 9.4 使用 GET 方法提交表单数据

显而易见，这种方法会将参数暴露无疑。如果用户传递的参数是非保密性的参数（如 id=8），那么采用 GET 方法传递数据是可行的；如果用户传递的是保密性的参数（如用户登录的密码，或者信用卡号等），这种方法就会不安全。解决该问题是将表单的 method 指定的 GET 方法改为 POST 方法。

### 9.4.3 POST 方法与 GET 方法的使用区别

在浏览器中向服务器发送表单数据的方法有两种，即 POST 方法和 GET 方法。这两种方法在 Web 页面的应用上有着本质的不同。

- POST 方法发送变量数据时，对于用户而言是保密性质的。从 HTTP 协议来看，数据附加于 header 的头信息中，用户不能随意修改，这对于 Web 应用程序而言，安全性要好得多，而且使用 POST 方法向 Web 服务器发送数据的大小不受限制。
- GET 方法是在访问 URL 时使用浏览器地址栏传递值。GET 方法方便直观，但缺点是访问该

网站的用户也可以修改 URL 串后发送给服务器, GET 传递的字符串长度有一定的限制, 不能超过 250 个字符; 如果超长, 浏览器会自动截取, 导致数据丢失或程序运行出错。另外, GET 方法不支持 ASCII 字符之外的任何字符, 如果包含有汉字或其他非 ASCII 字符, 需要应用 PHP 的内置函数将参数值转换成其他编码格式进行传递。

因此, 在网站开发过程中, 程序员应根据实际需要灵活地选择 POST 方法和 GET 方法来提交表单数据。

## 9.5 PHP 全局变量

### 视频讲解: 光盘\TM\Video\9\PHP全局变量.exe

PHP 的全局变量有 3 种: `$_POST[]`、`$_GET[]` 和 `$_SESSION[]`, 分别用于获取表单、URL、Session 变量的值。这 3 种方法在使用上有很大的区别, 下面分别进行详细介绍。

### 9.5.1 `$_POST[]`全局变量

使用 PHP 的 `$_POST[]` 全局变量可以获取表单元素的值。在实际程序开发过程中, 使用哪种方法获取数据资源, 是由 `<form>` 表单元素的 `method` 属性决定的。如果表单中 `method` 属性指定的是用 POST 方法进行数据传递, 那么在处理数据时就应该使用 `$_POST[]` 全局变量获取表单数据。

格式为:

```
$_POST[name]
```

例如, 建立一个表单, 设置 `method` 属性值为 POST, 添加一文本框, 命名为 `user`, 获取表单元素的代码如下。

```
<?php  
$user=$_POST["user"]; //应用$_POST[]全局变量获取表单元素中文本框的值  
?>
```

 注意: 有的 PHP 版本中直接写 `$user` 即可调用表单元素的值, 这和 `php.ini` 的配置有关系。在 `php.ini` 文件中检索到 “`register_globals=ON/OFF`” 时, 如果为 ON, 则可以直接写成 `$user`, 反之则不可以。虽然直接应用表单名称十分方便, 但也存在一定的安全隐患。推荐关闭 `register_globals`。

通过 `$_POST[]` 全局变量获取表单数据, 实际上就是获取不同的表单元素的数据。`<form>` 标签中的 `name` 是所有表单元素都具备的属性, 即为这个表单元素的名称, 在使用时需要使用 `name` 属性来获取相应的 `value` 属性值。所以添加的所有控件必须定义其 `name` 属性值, 另外, 为了避免获取的数据出现错误, 表单元素在命名上尽可能不要重复, 尽量使用具有一定意义的英文缩写或拼音命名。

在程序开发过程中, 获取文本框、密码域、隐藏域、按钮以及文本域的值的方法是相同的, 都是一个表单元素对应一个值, 通过 `name` 属性来获取相应的 `value` 属性值即可。

因此本节仅以获取文本框中的数据信息为例，讲解获取表单数据的方法。希望读者能够举一反三，自行完成获取其他控件值。

#### 例 9.4 通过实例讲解如何获取文本框的信息。（实例位置：光盘\TM\Instances\9\4）

具体操作步骤如下：

- (1) 应用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php。
- (2) 创建一个表单，添加两个文本框和一个“登录”按钮，代码如下。

```
<form name="form1" method="post" action="">
    用户名: <input type="text" name="user" size="20" >
    密码: <input name="pwd" type="password" id="pwd" size="20" >
    <input name="submit" type="submit" id="submit" value="登录" />
</form>
```

(3) 在<form>表单元素外的任意位置添加 PHP 标记符，应用 if 条件语句判断用户是否提交了表单，如果条件成立，则通过\$\_POST[]全局变量获取用户名“\$\_POST[user]”和密码“\$\_POST[pwd]”，代码如下。

```
<?php
if($_POST["submit"]=="登录"){ //判断提交的按钮名称是否等于“登录”
    //应用 echo 语句输出使用$_POST[]全局变量获取的用户名和密码
    echo "您输入的用户名为：" . $_POST[user]. "&nbsp;&nbsp;密码为：" . $_POST[pwd];
}
?>
```

**注意：**在应用文本框传递值时，一定要正确书写文本框的名称，在表单元素的命名上不应该有空格存在；在获取文本框提交的值时，书写的文本框名称一定要与提交文本框页中设置的名称相同，否则将不能获取到文本框的值。

- (4) 在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 9.5 所示。

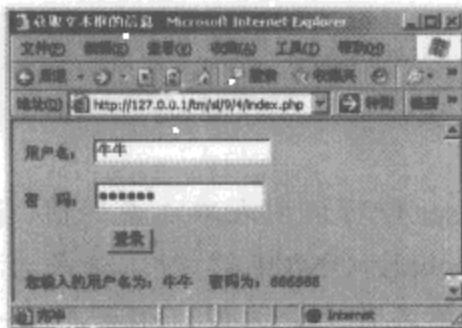


图 9.5 获取文本框的信息

### 9.5.2 \$\_GET[]全局变量

PHP 使用\$\_GET[]全局变量获取通过 GET 方法传递的值，使用格式为：

`$_GET[name]`

例如，创建一个表单，设置 method 属性值为 GET，添加一个文本框，name 为 user，获取表单元

素值的代码如下。

```
<?php
$user=$_GET["user"]; //应用$_GET[]全局变量获取表单元素中文本框的值
?>
```

**注意：**PHP 可以应用\$\_POST[]或\$\_GET[]全局变量来获取表单元素的值。但值得注意的是，获取的表单元素名称区分字母大小写。如果在编写 Web 程序时忽略字母大小写，那么在程序运行时将获取不到表单元素的值或弹出错误提示信息。

### 9.5.3 \$\_SESSION[]全局变量

使用\$\_SESSION[]变量可以跨页获取变量的值，格式为：

`$_SESSION[name]`

在 PHP 动态页中，可以将变量或者表单元素的值赋值给\$\_SESSION[]全局变量，进而实现变量值或者表单元素值的跨页传递。

例如，建立一个表单，添加一个文本框，命名为 user，应用\$\_SESSION[]全局变量获取表单元素的代码如下。

`$user=$_SESSION["user"]`

应用\$\_SESSION[]传参的方法获取的变量值，保存之后任何页面都可以使用。但是这种方法很耗费系统资源，建议读者慎重使用。Session 全局变量的具体应用将在第 12 章进行详细讲解。

## 9.6 对 URL 传递的参数进行编/解码

**视频讲解：**光盘\TM\Video\9\对 URL 传递的参数进行编/解码.exe

### 9.6.1 对 URL 传递的参数进行编码

使用 URL 参数传递数据，就是在 URL 地址后面加上适当的参数。URL 实体对这些参数进行处理。使用方法如下：

`http://url?name1=value1&name2=value2.....`

URL 传递的参数，也称为查询字符串

显而易见，这种方法会将参数暴露无疑，因此，本节针对该问题讲述一种 URL 编码方式，对 URL 传递的参数进行编码。

URL 编码是一种浏览器用来打包表单输入数据的格式。是对用地址栏传递参数进行的一种编码规则。例如，在参数中带有空格，则传递参数时就会发生错误，而用 URL 编码过以后，空格转换成了“%20”，

这样错误就不会发生，对中文进行编码也是同样的情况，最主要的一点就是它可以对 URL 传递的参数进行编码。

PHP 中对字符串进行 URL 编码使用的是 `urlencode()` 函数，该函数的语法如下：

```
string urlencode( string str)
```

该函数可以实现将字符串 str 进行 URL 编码。

**例 9.5** 本实例中，单击图片，通过 URL 传递图片名称到指定文件页，应用 `urlencode()` 函数对图片的名称进行 URL 编码，显示在 IE 地址栏中的字符串是 URL 编码后的字符串，代码如下：（实例位置：光盘\TM\Instances\9\5）

```
<a href="index.php?picname=<?php echo urlencode("青年歌手大赛");?>">

</a>
```

运行结果如图 9.6 所示。

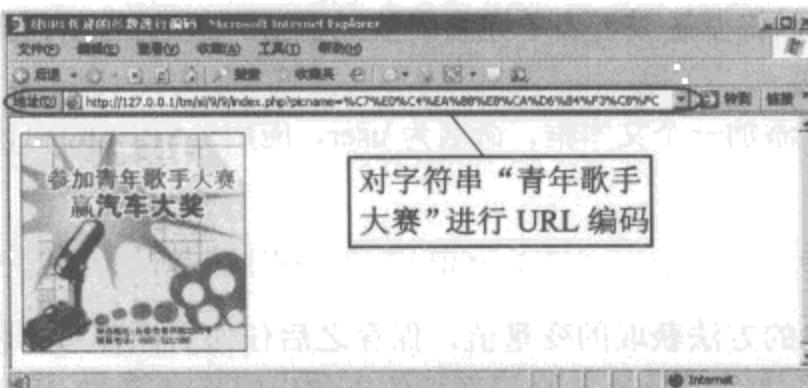


图 9.6 对字符串“青年歌手大赛”进行 URL 编码

**说明：**对于服务器而言，编码前后的字符串并没有什么区别，服务器能够自动识别。这里是为了讲解 URL 编码的使用方法，而在实际应用中，对一些非保密性的参数不需要进行编码，读者可根据实际情况有选择地使用。

## 9.6.2 对 URL 编码的字符串进行解码

对于 URL 传递的参数直接应用 `$_GET[]` 方法获取即可。而对于进行 URL 加密的查询字符串，需要通过 `urldecode()` 函数对获取后的字符串进行解码。该函数的语法如下：

```
string urldecode( string str)
```

该函数可以实现将 URL 编码 str 查询字符串进行解码。

**例 9.6** 在例 9.5 中应用 `urlencode()` 函数实现了对字符串“青年歌手大赛”进行编码，将编码后的字符串传给变量 `picname`。在本实例中，将应用 `urldecode()` 函数对获取的变量 `picname` 进行解码，将解码后的结果输出到浏览器，代码如下：（实例位置：光盘\TM\Instances\9\6）

```
<a href="index.php?picname=<?php echo urlencode("青年歌手大赛");?>">

```

```
</a>
<?php echo "您单击的图片名称是:".urldecode($_GET[picname]);?>
```

运行结果如图 9.7 所示。

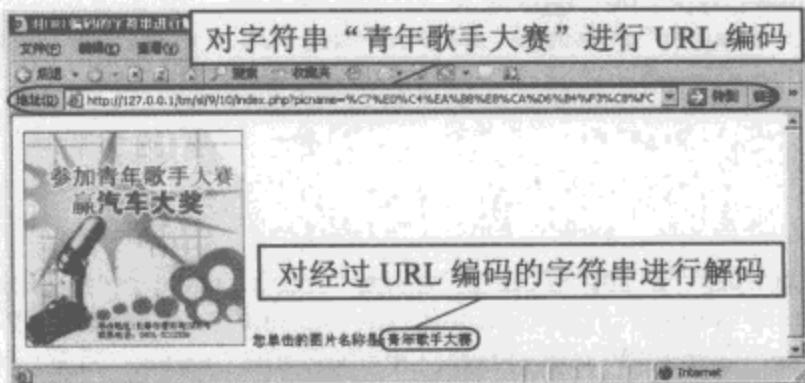


图 9.7 对 URL 编码的字符串进行解码

## 9.7 实战

表单是实现动态网页的一种主要的外在形式，使用表单可以收集客户端提交的有关信息，表单是实现网站互动功能的重要组成部分。下面通过两个实例来讲解 PHP 与 Web 表单元素之间的应用。

### 9.7.1 发布和查看公告信息

#### ■ 视频讲解：光盘\TM\Video\9\发布和查看公告信息.exe

**例 9.7** 在本实例中，综合了前面各实例中介绍的有关表单元素的知识，实现表单元素的综合应用。本实例主要应用表单元素实现一个发布公告信息的功能，通过将表单信息提交到数据处理页获取表单元素的值，实现查看公告信息的功能。（实例位置：光盘\TM\Instances\9\7）

(1) 应用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php。

(2) 创建 form 表单（设置表单的数据处理页为 show\_message.php），添加文本框、编辑框、下拉列表框、提交按钮和重置按钮，并应用表格对表单元素进行合理的布局，程序代码如下。

```
<form action="show_message.php" method="post" name="addmess" id="addmess">
<table width="560" height="180" bordercolor="#ACD2DB" bgcolor="#ACD2DB" class="big_td">
<tr>
    <td width="100" height="25" bgcolor="#DEEBEF" scope="col">标题:</td>
    <td height="25" align="left" valign="middle" bgcolor="#DEEBEF" scope="col">
        <input type="text" name="title" id="title" />
        &nbsp;</td>
    </tr>
    <tr>
        <td align="right" valign="middle" bgcolor="#DEEBEF">内容:</td>
        <td align="left" valign="middle" bgcolor="#DEEBEF">
            <textarea name="content" id="content" cols="56" rows="10"></textarea>
        </td>
    </tr>
</table>
<input type="submit" value="发布" />
<input type="reset" value="重置" />
```

```

</tr>
<tr>
    <td height="30" align="right" valign="middle" bgcolor="#DEEBEF">类别: </td>
    <td height="30" align="left" valign="middle" bgcolor="#DEEBEF">
        <select name="type" id="type">
            <option value="企业公告" selected="selected">企业公告</option>
            <option value="活动安排">活动安排</option>
        </select>
    </td>
</tr>
<tr>
    <td height="30" colspan="2" align="center" valign="middle" bgcolor="#DEEBEF">
        <input name="submit" type="submit" id="submit" value="发布" />
        <input name="submit2" type="reset" id="submit2" value="重置" />
    </td>
</tr>
</table>
</form>

```

(3) 对表单提交的数据进行处理，应用 echo 语句输出提交的各表单元素值。代码如下。

```





```

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 9.8 所示。添加相应的公告信息后，单击“发布”按钮，查看公告信息页面的运行结果如图 9.9 所示。

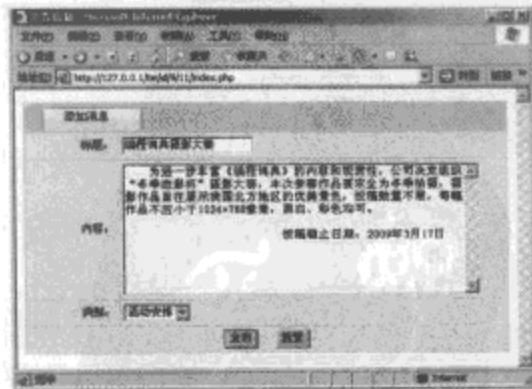


图 9.8 发布公告信息

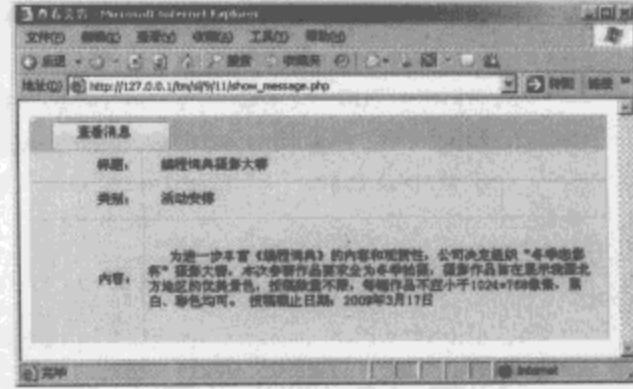


图 9.9 查看公告信息

## 9.7.2 图书借阅

### 视频讲解：光盘\TM\Video\9\图书借阅.exe

例9.8 在本实例中，应用表单和表单元素设计一个图书借阅模块，通过单击“完成借阅”提交按钮，提交图书借阅的信息到数据处理页。通过\$\_POST[]全局变量来获取图书借阅信息的数据。其完整代码如下：（实例位置：光盘\TM\Instances\9\8）

```
<form name="form1" method="post" action="index.php">
<table width="738" bordercolor="#FFFFFF" bgcolor="#9ECFEE" class="tableBorder_grey">
<tr>
    <td height="33" valign="top" background="Images/bookborr.gif">
        <table width="96%" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td height="27">姓 &nbsp;&nbsp;&nbsp;名:<br/>
                    <input name="readername" type="text" id="readername">
                </td>
                <td>图书名称:<br/>
                    <input name="bookname" type="text" id="bookname">
                </td>
                <td>书架:<br/>
                    <input name="bookcase" type="text" id="bookcase" size="17">
                </td>
            </tr>
            <tr>
                <td height="27">出 &nbsp;版 &nbsp;社:<br/>
                    <input name="publisher" type="text" id="publisher">
                </td>
                <td>定 &nbsp;&nbsp;&nbsp;价:<br/>
                    <input name="price" type="text" id="price">
                </td>
                <td>可借数量:<br/>
                    <input name="number" type="text" id="number" size="17">
                </td>
            </tr>
        </table>
    </td>
</tr>
<tr>
    <td height="32" ><input name="Button2" type="submit" id="Button2" value="完成借阅"></td>
</tr>
<tr>
    <td valign="top" bgcolor="#D2E5F1" style="padding:5px">
        <table width="100%" bordercolor="dark"="#9ECFEE" bgcolor="#FFFFFF">
            <tr align="center" bgcolor="#E2F4F6">
                <td width="20%" height="25">图书名称</td>
                <td width="15%">借阅时间</td>
                <td width="15%">应还时间</td>
                <td width="7%">数量</td>
                <td width="15%">出版社</td>
                <td width="8%">书架</td>
                <td width="8%">定价(元)</td>
                <td colspan="2" style="font-size:1.5em;">读者姓名</td>
            </tr>
        </table>
    </td>
</tr>
```

```

<tr>
    <td height="25" style="padding:5px;"><?php echo $_POST["bookname"];?></td>
    <td style="padding:5px;"> <?php echo date("Y-m-d");?></td>
    <td style="padding:5px;"> <?php echo date("Y-m-d", (time() + 3600 * 24 * 30)); //归还图
书日期为当前期日期+30 天期限?></td>
    <td align="center" style="padding:5px;"><?php echo $_POST["number"];?></td>
    <td align="center"><?php echo $_POST["publisher"];?></td>
    <td align="center"><?php echo $_POST["bookcase"];?></td>
    <td align="center"><?php echo $_POST["price"];?></td>
    <td width="13%" align="center"><?php echo $_POST["readernumber"];?></td>
</tr>
</table>
</td>
</tr>
</table>
</form>

```

在上面的代码中，应用 `date("Y-m-d")` 获取系统的当前日期作为图书借阅的日期。而“归还图书日期=当前期日期+30 天期限”，根据这个公式，可以获取图书的归还日期，即当前日期时间戳应用 `time()` 函数获取，30 天期限的时间戳等于  $3600 \text{ 秒} \times 24 \text{ 小时} \times 30 \text{ 天}$ ，并通过 `date()` 函数格式化为指定日期格式，得出的应还时间表达式为 `date("Y-m-d", (time() + 3600 * 24 * 30))`。

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 9.10 所示。

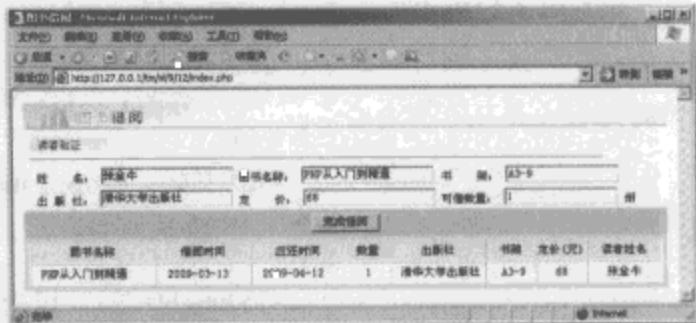


图 9.10 图书借阅

## 9.8 本章小结

本章主要介绍了创建表单及表单元素、POST 方法和 GET 方法的应用、PHP 全局变量以及对 URL 传参的编码和解码。相信通过本章的学习，读者可以掌握 PHP 与 Web 页的交互，为深入学习 PHP 打下扎实的基础。

## 9.9 实战练习

- 尝试创建一个签写留言页面，要求在页面中添加常用的表单元素，文本框、编辑框、单选按钮组、复选框和按钮等，并对各表单元素合理命名。（答案位置：光盘\TM\Instances\9\9）
- 开发一个用户注册页面，尝试添加各种表单元素，并应用 `$_POST[]` 全局变量输出提交的用户注册信息。（答案位置：光盘\TM\Instances\9\10）
- 添加一个“明日编程词典网”超链接，单击该超链接，跳转到指定页，并实现对 GET 方法传递的参数进行编码，然后对编码的字符串进行解码并输出，测试程序，并查看 IE 浏览器中地址栏的变化。（答案位置：光盘\TM\Instances\9\11）

# 第10章

PHP 与 JavaScript 交互

## PHP 与 JavaScript 交互

( 视频讲解: 81分钟)

JavaScript 是一种可以嵌入 HTML 代码中由客户端浏览器运行的脚本语言。在网页中使用 JavaScript 脚本代码，不仅可以实现网页特效，还可以响应用户请求实现动态交互的功能。在 PHP 动态网页中灵活运用 JavaScript，可以实现更强大的功能。本章将介绍 JavaScript 脚本语言的基础知识，使读者在掌握 JavaScript 基础知识的同时，更好地实现 PHP 与 JavaScript 的交互。

通过阅读本章，可以学习到：

- » **JavaScript 是什么**
- » **JavaScript 能做什么**
- » **JavaScript 脚本语言的基础**
- » **在网页中执行 JavaScript、调用自定义函数和引用 JS 文件的方法**
- » **解决浏览器不支持 JavaScript 语言的方法**
- » **运用 JavaScript 实现自定义函数**
- » **在 PHP 动态页中调用 JavaScript 脚本**

## 10.1 了解 JavaScript

JavaScript 是个脚本编程语言，支持 Web 应用程序的客户端和服务器端构件的开发，在 Web 系统中得到了非常广泛的应用。下面对 JavaScript 进行简单的介绍。

### 10.1.1 JavaScript 是什么

JavaScript 是由 Netscape Communication Corporation（网景公司）开发的，是一种基于对象和事件驱动并具有安全性能的解释型脚本语言。它不但可用于编写客户端的脚本程序，由 Web 浏览器解释执行；而且还可以编写服务器端执行的脚本程序，在服务器端处理用户提交的信息并动态地向浏览器返回处理结果。

### 10.1.2 JavaScript 能做什么

JavaScript 是比较流行的一种制作网页特效的脚本语言，它由客户端浏览器解释执行，可以应用在 PHP、ASP、JSP 和 ASP.NET 网站中，同时目前比较热门的 Ajax 就是以 JavaScript 为基础。由此可见，熟练掌握并应用 JavaScript 对于网站开发人员非常重要。

JavaScript 主要应用于以下几个方面：

- (1) 在网页中加入 JavaScript 脚本代码，可以使网页具有动态交互的功能，便于网站与用户间的沟通，及时响应用户的操作，对提交的表单做即时的检查，如验证表单元素是否为空，验证表单元素是否是数值型，检测表单元素是否输入错误等。
- (2) 应用 JavaScript 脚本制作网页特效，如动态的菜单、浮动的广告等，为页面增添绚丽的动态效果，使网页内容更丰富、更活泼。
- (3) 应用 JavaScript 脚本建立复杂的网页内容，如打开新窗口载入网页。
- (4) 应用 JavaScript 脚本可以对用户的不同事件产生不同的响应。
- (5) 应用 JavaScript 制作各种各样的图片、文字、鼠标、动画和页面的效果。
- (6) 应用 JavaScript 制作一些小游戏。

## 10.2 JavaScript 语言基础

 视频讲解：光盘\TM\Video\10\JavaScript 语言基础.exe

JavaScript 脚本语言与其他语言一样，有其自身的基本数据类型、表达式和运算符以及程序的基本框架结构。通过本节的学习读者可以掌握更多 JavaScript 脚本语言的基础知识。

### 10.2.1 JavaScript 数据类型

JavaScript 主要有 6 种数据类型，如表 10.1 所示。

表 10.1 JavaScript 数据类型

数据类型	说 明	举 例
字符串型	使用单引号或双引号括起来的一个或多个字符	如 "PHP"、"I like study PHP" 等
数值型	包括整数或浮点数（包含小数点的数或科学记数法的数）	如 -128、12.9、6.98e6 等
布尔型	布尔型常量只有两种状态：true 或 false	如 event.returnValue=false
对象型	用于指定 JavaScript 程序中用到的对象	如网页表单元素
null 值	可以通过给一个变量赋 null 值来清除变量的内容	如 a=null
undefined	表示该变量尚未被赋值	如 var a

## 10.2.2 JavaScript 变量

变量是指程序中一个已经命名的存储单元，它的主要作用就是为数据操作提供存放信息的容器。对于变量的使用首先必须明确变量的命名规则、变量的声明方法及其变量的作用域。

### 1. 变量的命名规则

JavaScript 变量的命名规则如下：

- 必须以字母或下划线开头，中间可以是数字、字母或下划线。
- 变量名不能包含空格或加号、减号等符号。
- JavaScript 的变量名是严格区分大小写的。例如，User 与 user 代表两个不同的变量。
- 不能使用 JavaScript 中的关键字。JavaScript 的关键字如表 10.2 所示。

表 10.2 JavaScript 的关键字

abstract	continue	finally	instanceof	private	this
boolean	default	float	int	public	throw
break	do	for	interface	return	typeof
byte	double	function	long	short	true
case	else	goto	native	static	var
catch	extends	implements	new	super	void
char	false	import	null	switch	while
class	final	in	package	synchronized	with

说明：虽然 JavaScript 的变量可以任意命名，但为了在编程时使代码更加规范，最好使用便于记忆且有意义的变量名称，以增加程序的可读性。

### 2. 变量的声明与赋值

在 JavaScript 中，一般在使用变量前需要先声明变量，不过有时变量可以不先声明，使用时再根据变量的实际作用来确定其所属的数据类型。所有的 JavaScript 变量都由关键字 var 声明。

语法：

```
var variable;
```

在声明变量的同时也可以对变量进行赋值：

```
var variable=08;
```

 技巧：建议在使用变量前就对其声明，因为声明变量的最大好处就是能及时发现代码中的错误。

由于 JavaScript 是采用动态编译的，而动态编译不易于发现代码中的错误，特别是变量命名方面的错误。

声明变量时所遵循的规则如下。

- 可以使用一个关键字 var 同时声明多个变量，例如：

```
var m,n;
```

- 可以在声明变量的同时对其进行赋值，即为初始化，例如：

```
var m=1;n=100;
```

- 如果只是声明了变量，并未对其进行赋值，则默认值为 undefined。

 注意：在 JavaScript 中，可以使用分号代表一个语句的结束，如果每个语句都在不同的行中，那么分号可以省略；如果多个语句在同一行中，分号就不能省略。建议读者不要省略分号，以养成良好的编程习惯。

例如，声明 3 个不同数据类型的变量，代码如下。

```
var i=100; //定义变量 i 为数值类型
var str="明日科技，明天会更好！"; //定义变量 str 为字符串类型
var content=true; //定义变量 content 为布尔类型
```

 注意：在程序开发过程中，可以使用 var 语句多次声明同一个变量，如果重复声明的变量已经有一个初始值，那么此时的声明变量就相当于对变量重新赋值。

### 10.2.3 JavaScript 注释

在 JavaScript 中，采用的注释方法有以下两种。

#### 1. 单行注释

单行注释使用 “//” 进行标识。“//” 符号后面的文字都不被程序解释执行。例如：

```
//这里是程序代码的注释
```

#### 2. 多行注释

多行注释使用 “/\*...\*/” 进行标识。“/\*...\*/” 符号后面的文字不被程序解释执行。例如：

```
/*
这里是多行程序注释
*/
```

**◆ 注意：**多行注释 “/\*...\*/” 中可以嵌套单行注释 “//”，但是不可以嵌套多行注释 “/\*...\*/”。因为第一个 “/\*” 会与其后面第一个 “\*/” 相匹配，从而使后面的注释不起作用，甚至导致程序出错。

另外，JavaScript 还能识别 HTML 注释的开始部分 “”。

这种现象存在的主要原因是：在 JavaScript 中，如果第一行以 “” 结束，那么其间的程序就包含在一个完整的 HTML 注释中，会被不支持 JavaScript 的浏览器忽略掉，不能被显示。如果第一行以 “” 结束，JavaScript 会将两行都忽略掉，而不会忽略这两行之间的部分。用这种方式可以针对那些无法理解 JavaScript 的浏览器而隐藏代码，而对那些可以理解 JavaScript 的浏览器则不必隐藏。

### 10.3 自定义函数

#### ◆ 视频讲解：光盘\TM\Video\10\自定义函数.exe

自定义函数就是由用户自己命名并且书写的能实现特定功能的程序单元。用户使用自定义函数必须事先声明，不能直接使用未声明的自定义函数。

JavaScript 用 function 来定义函数，语法格式如下：

```
function 函数名([参数]){
    return var;
}
```

} 函数体

自定义函数的调用方法是：

```
函数名();
```

其中的括号绝对不能省略。

**例 10.1** 自定义一个 check\_tel() 函数，实现对用户输入的电话号码的格式进行判断，然后在函数外调用自定义函数 check\_tel()，向自定义函数中传递一个电话号码，最后应用 document.write() 对象执行自定义函数，并输出结果，语法如下：（实例位置：光盘\TM\Instances\10\1）

```
<script language="javascript">
function check_tel(regtels){
    var str=regtels;
    var Expression=/^(\d{3}-)(\d{8})$|^( \d{4}-)(\d{7})$|^( \d{4}-)(\d{8})$/; //编写验证电话号码格式的正则表达式
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){ //格式正确时返回 true
        return true;
    }else{ //格式不正确则返回 false
        return false;
    }
}
document.write(check_tel("0431-84972266")); //调用 check_tel() 函数并传递参数，输出结果
</script>
```

结果为： true

**◆ 注意：**在同一个页面中不能定义名称相同的函数。另外，当用户自定义函数后，需要对该函数进行引用，否则自定义的函数将失去意义。

## 10.4 JavaScript 脚本嵌入方式

### 10.4.1 在 HTML 中嵌入 JavaScript 脚本

**◆ 视频讲解：**光盘\TM\Video\10\在 HTML 中嵌入 JavaScript 脚本.exe

JavaScript 作为一种脚本语言，可以嵌入到 HTML 文件中。在 HTML 中嵌入 JavaScript 脚本的方法是使用<script>标记。

语法：

```
<script language="javascript">
...
</script>
```

应用<script>标记是直接执行 JavaScript 脚本最常用的方法，大部分含有 JavaScript 的网页都采用这种方法，其中，通过 language 属性可以设置脚本语言名称和版本。

**◆ 注意：**如果在<script>标记中未设置 language 属性，Internet Explorer 浏览器和 Netscape 浏览器将默认使用 JavaScript 脚本语言。

**例 10.2** 实现在 HTML 中嵌入 JavaScript 脚本，直接在<script>和</script>标识符中间写入 JavaScript 代码，用于弹出一个提示对话框，代码如下：（实例位置：光盘\TM\Instances\10\2）

```
<html>
<head>
<title>在 HTML 中嵌入 JavaScript 脚本</title>
</head>
<body>
<script language="javascript">
    alert("PHP 编程词典，正在酝酿中!");
</script>
</body>
</html>
```

在上面的代码中，<script>与</script>标记之间调用 JavaScript 脚本语言 Window 对象的 alert 方法向客户端浏览器弹出一个提示对话框。在 IE 浏览器中打开 HTML 文件，运行结果如图 10.1 所示。

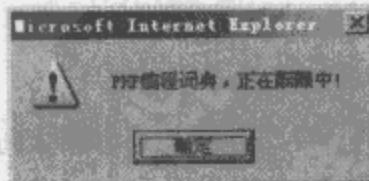


图 10.1 在 HMTL 中嵌入 JavaScript 脚本

**◆ 注意：**JavaScript 脚本通常写在<head>…</head>标记和<body>…</body>标记之间。写在<head>标记中间的一般是函数和事件处理函数；写在<body>标记中间的是网页内容或调用函数的程序块。

在 HTML 中通过“javascript:”可以调用 JavaScript 的方法。例如，在页面中插入一个按钮，在该按钮的 onClick 事件中应用“javascript:”调用 Window 对象的 alert 方法弹出一个警告提示框，代码如下。

```
<input type="submit" name="Submit" value="单击这里" onClick="javascript:alert('您单击了这个按钮！')">
```

## 10.4.2 应用 JavaScript 事件调用自定义函数

**◆ 视频讲解：**光盘\TM\Video\10\应用 JavaScript 事件调用自定义函数.exe

在 Web 程序开发过程中，经常需要在表单元素的相应事件下调用自定义函数。例如，在按钮的单击事件下调用自定义函数 check()，验证表单元素是否为空，代码如下。

```
<input type="submit" name="Submit" value="检测" onclick="check();">
```

然后在该 Web 表单的当前页写一个 check() 自定义函数即可。

**例 10.3** 创建用户登录页面，在用户登录按钮中通过 onclick 事件调用自定义函数，验证表单中提交的元素是否为空。自定义函数的代码如下：（实例位置：光盘\TM\Instances\10\3）

```
<script>
function check_form(){
    if(form1.username.value==""){
        alert("请输入用户名！");
        form1.username.select();
        return(false);
    }
    if(form1.password.value==""){
        alert("请输入密码！");
        form1.password.select();
        return(false);
    }
}
</script>
```

创建 form 表单元素的代码如下。

```
<form id="form1" name="form1" method="post" action="index.php">
    <input name="username" type="text" id="username" size="18" />
    <input name="password" type="password" id="password" size="18" />
    <input type="submit" name="Submit" value="登录" onclick="return check_form();"/>
    <input type="reset" name="Submit2" value="重置" />
</form>
```

如果在登录页面中只输入用户名，将弹出如图 10.2 所示的页面。



图 10.2 应用 JavaScript 事件调用自定义函数

### 10.4.3 在 PHP 动态网页中引用 JS 文件

#### ■ 视频讲解：光盘\TM\Video\10\在 PHP 动态网页中引用 JS 文件.exe

在网页中，除了在<script>与</script>标记之间编写 JavaScript 脚本代码以外，还可以通过<script>标记中的 src 属性指定外部的 JavaScript 文件（即 JS 文件，以.js 为扩展名）的路径，从而引用对应的 JS 文件。

语法：

```
<script src=url language="Javascript"></script>
```

其中，url 是 JS 文件的路径，其中“language="Javascript"”可以省略，因为<script>标记默认使用的就是 JavaScript 脚本语言。

JavaScript 脚本不仅可以与 HTML 结合在一起使用，同时也可以与 PHP 动态网页结合使用，其引用的方法是一样的。使用外部 JS 文件的优点如下：

- ☑ 使用 JS 文件可以将 JavaScript 脚本代码从网页中独立出来，便于代码的阅读。
- ☑ 一个外部 JS 文件，可以同时被多个页面调用。当公共的 JavaScript 脚本代码需要修改时，只需要修改一个 JS 文件中的代码即可，便于代码的维护。
- ☑ 通过<script>标记中的 src 属性不但可以调用同一个服务器上的 JS 文件，还可以通过指定路径来调用其他服务器上的 JS 文件。

**例 10.4** 在网页中，通过<script>标记的 src 属性引用外部的 JS 文件，用于弹出一个提示对话框。index.php 文件中的代码如下：（实例位置：光盘\TM\Instances\10\4）

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>通过 script 标记的 src 属性引用外部的 JS 文件</title>
</head>
<script language="javascript" src="js/script.js"></script>
<body>
<table width="1149" height="342" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td width="1149"></td>
  </tr>
</table>
</body>
</html>
```

将 script.js 文件存储在根目录下的 js 文件夹中，代码如下。

```
var newformW=300;
var newformH=180;
function pp0{
    var T=window.screen.width-newformW;
    var L=window.screen.height+newformH;
    var name=window.open("advertise.htm","", "width='"+newformW+"',height='"+newformH+"',top='"+T+"',left='"+L+"');
}
pp0;
```

从上面的代码中可以看出，在 index.php 文件中通过设定<script>标记中的 src 属性，引用了 js 文件夹下的 script.js 文件。在 script.js 文件中调用 JavaScript 脚本语言 Window 对象的 open 方法打开一个指定的 html 文件，并设置该页面的大小，以及在主页面中的位置。运行结果如图 10.3 所示。



图 10.3 在 PHP 动态网页中引用 JS 文件

在网页中引用 JS 文件和注意事项如下。

- 在 JS 文件中，只能包含 JavaScript 脚本代码，不能包含<script>标记和 HTML 代码。
- 在引用 JS 文件的<script>与</script>标记之间不应存在其他的 JavaScript 代码，即使存在 JavaScript 代码，浏览器也会忽略此脚本代码，而只执行 JS 文件中的 JavaScript 脚本代码。

#### 10.4.4 解决浏览器不支持 JavaScript 问题

##### ■ 视频讲解：光盘\TM\Video\10\解决浏览器不支持 JavaScript 问题.exe

虽然大多数浏览器都支持 JavaScript 脚本，但是由于浏览器设置的不同，导致很多浏览器不支持 JavaScript 脚本。如果遇到不支持 JavaScript 脚本的浏览器，网页会达不到预期效果或出现错误。解决这个问题可以使用以下 3 种方法。

##### 1. 开启 IE 浏览器对 JavaScript 的支持

目前有些支持 JavaScript 的浏览器为了安全关闭了对 JavaScript 的支持。这时，浏览器可以启用对 JavaScript 脚本的支持来解决这一问题。具体操作步骤如下：

- (1) 选择 IE 浏览器菜单中的“工具” / “Internet 选项”命令，打开“Internet 选项”对话框，选择“安全”选项卡，选择 Internet 安全设置项，单击“自定义级别”按钮，打开如图 10.4 所示的对话框。

(2) 将“Java 小程序脚本”和“活动脚本”两个选项设置为启用状态。单击“确定”按钮，即可开启 IE 浏览器支持 JavaScript 脚本的功能。

## 2. 开启 IE 浏览器对本地 JavaScript 的支持

IE 浏览器将网页分为 Internet、本地 Intranet、受信任的站点和受限制的站点 4 个区域，但不包括本地网页。通常在 Windows XP 操作系统中，打开 IE 浏览器中包含 JavaScript 的网页时，弹出如图 10.5 所示的安全提示对话框。

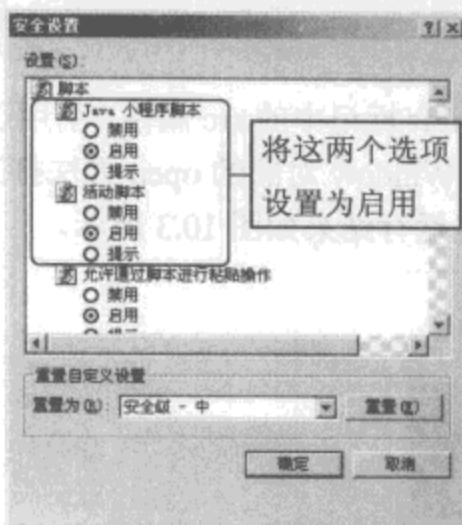


图 10.4 启用 JavaScript 脚本功能

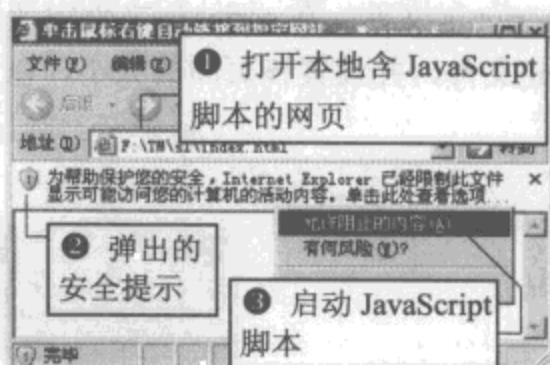


图 10.5 安全提示对话框

如果用户要继续执行网页中包含的 JavaScript 脚本，可以用鼠标右键单击安全提示区域，在弹出的快捷菜单中选择“允许阻止的内容”命令，如图 10.5 所示，即可成功运行本网站。但此选项仅针对当前网页。若要永久消除 IE 浏览器的这种安全提示，需要对 IE 浏览器进行如下设置。

选择 IE 浏览器菜单中的“工具” / “Internet 选项”命令，打开“Internet 选项”对话框，选择“高级”选项卡，在安全设置区选中“允许活动内容在我的计算机上的文件中运行”和“允许来自 CD 的活动内容在我的计算机上运行”复选框（此选项仅适用于 Windows XP 操作系统），单击“确定”按钮，即可成功解决上述问题。

## 3. 应用注释符号验证浏览器是否支持 JavaScript 脚本功能

如果用户不能确定浏览器是否支持 JavaScript 脚本，那么可以应用 HTML 提供的注释符号进行验证。HTML 注释符号是以“”结束的。如果在此注释符号内编写 JavaScript 脚本，对于不支持 JavaScript 的浏览器，将会把编写的 JavaScript 脚本作为注释处理。

**例 10.5** 使用 JavaScript 脚本在页面中输出一个字符串，将 JavaScript 脚本编写在 HTML 注释中，如果浏览器支持 JavaScript 将输出此字符串；如果不支持将不输出此字符串，代码如下：（实例位置：光盘\TM\Instances\10\5）

```
<html>
<head>
<title>应用注释符号验证浏览器是否支持 JavaScript 脚本功能</title>
</head>
<body>
<script type="text/javascript" >
```

```
<!--
  document.write("您的浏览器支持 JavaScript 脚本！");
-->
</script>
</body>
</html>
```

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 10.6 所示。

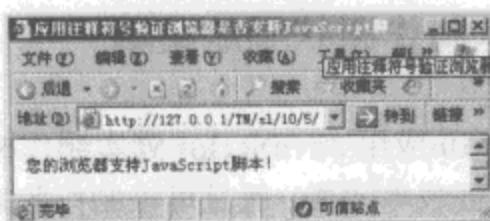


图 10.6 应用注释符号验证浏览器是否支持 JavaScript 脚本功能

#### 4. 应用<noscript>标记验证浏览器是否支持 JavaScript 脚本

除了可以通过上述的应用注释符号验证浏览器是否支持 JavaScript 脚本外，还可以应用<noscript>标记进行验证。

如果当前浏览器支持 JavaScript 脚本，那么该浏览器将会忽略<noscript>…</noscript>标记之间的任何内容。如果浏览器不支持 JavaScript 脚本，那么浏览器将会把<noscript>…</noscript>这两个标记之间的内容显示出来。此标记可以提醒浏览器当前使用的浏览器是否支持 JavaScript 脚本。

**例 10.6** 使用 JavaScript 脚本在页面中输出一个字符串，并使用<noscript>标记提醒浏览器当前浏览器是否支持 JavaScript 脚本。程序代码如下：（实例位置：光盘\TM\Instances\10\6）

```
<html>
<head>
<title>应用<noscript>标记验证浏览器是否支持 JavaScript 脚本</title>
</head>
<body>
<script language="javascript">
  document.write("您的浏览器支持 JavaScript 脚本");
</script>
<noscript>
  您的浏览器不支持 JavaScript 脚本
</noscript>
</body>
</html>
```

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 10.7 所示。

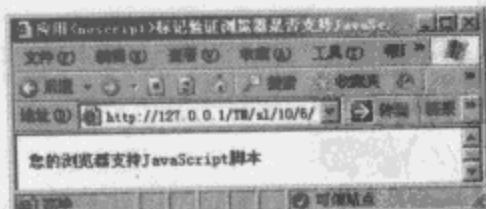


图 10.7 应用<noscript>标记验证浏览器是否支持 JavaScript 脚本

 技巧：当解释程序遇到</SCRIPT>标记时会终止当前脚本。要显示“</SCRIPT>”本身，将“<”改写为“&lt;”，将“>”改写为“&gt;”即可。若要使用 document.write 输出<script></script>标签时，需要将闭合标签通过反斜杠进行转义，如<script>…<\script>。

## 10.5 实战

### 10.5.1 应用 JavaScript 脚本制作树状导航菜单

 视频讲解：光盘\TM\Video\10\应用 JavaScript 脚本制作树状导航菜单.exe

应用 JavaScript 脚本不仅可以验证表单元素，而且可以制作各式各样的网站导航菜单。本节以网站开发中最常应用的树状导航菜单为例，讲解其实现方法。

例 10.7 应用 JavaScript 脚本实现一个树状导航菜单的功能。(实例位置：光盘\TM\Instances\10\7)

具体开发步骤如下：

(1) 在根目录下的 js 文件夹中创建 script.js 文件。在 script.js 中通过 function 创建自定义函数，控制单元格样式（style）display 的值。判断当 display 的值为空时隐藏单元格；当 display 的值为 none 时显示单元格。自定义函数的关键代码如下。

```
function open1(){
    if(qygl.style.display=="")          //判断当 display 的值为空时，为 display 赋值为 none
        qygl.style.display='none';
    else if(qygl.style.display=='none')   //判断当 display 的值为 none 时，为 display 赋值为空
        qygl.style.display="";
}
//省略了部分代码
function open2_1(){
    if(open2_1.style.display=="")
        open2_1.style.display='none';
    else if(open2_1.style.display=='none')
        open2_1.style.display="";
}
```

(2) 创建 index.php 页面，通过<script>标记调用 JS 文件。

(3) 创建表格，通过 onClick()事件调用自定义函数 open1(); 获取超链接中传递的变量，根据变量值定义单元格的显示和隐藏。关键代码如下。

```
<table width="85%" border="0" cellspacing="0" cellpadding="0">
<tr>
    <td width="84%" height="30">
        
    </td>
</tr>
</table>
<table id="qygl" width="85%" border="0" cellspacing="0" cellpadding="0">
<?php
if(!($GET[lmbs]=="公司简介" || $GET[lmbs]=="组织机构")){
    <tr>
        <td width="10%" height="30" style="text-align: center; vertical-align: middle;">
            <a href="index.php?lmbs=公司简介" style="color: black; text-decoration: none; font-size: 12px; font-weight: bold;">公司简介
        </td>
        <td width="90%" style="padding-left: 10px; padding-top: 5px; padding-bottom: 5px;">
            公司简介是公司对外宣传、介绍公司情况的主要途径，是公司形象的重要体现。公司简介的内容应包括公司的历史沿革、组织结构、主营业务、企业文化、发展战略等方面。
        </td>
    </tr>
    <tr>
        <td width="10%" height="30" style="text-align: center; vertical-align: middle;">
            <a href="index.php?lmbs=组织机构" style="color: black; text-decoration: none; font-size: 12px; font-weight: bold;">组织机构
        </td>
        <td width="90%" style="padding-left: 10px; padding-top: 5px; padding-bottom: 5px;">
            组织机构是指公司的内部管理体系，包括公司总部、各部门、各分公司等。组织机构图通常以树状结构形式呈现，展示了公司内部的层级关系和部门职能。
        </td>
    </tr>
}</?php
</table>
```

```

?>
    style="display:none"
<?php } ?>      >
<tr>
    <td width="35%" height="23" align="right">-|</td>
    <td width="65%" height="23"><a href="index.php?lmbs=公司简介"></a></td>
</tr>
<tr>
    <td width="35%" height="23" align="right">-|</td>
    <td width="65%" height="23"><a href="index.php?lmbs=组织机构"></a></td>
</tr>
<tr><td height="2" colspan="2" align="right"></td></tr>
</table>

```

运行本实例，选择页面左侧的“企业管理”导航菜单，将输出“公司简介”和“组织机构”两个导航条，选择“公司简介”导航条，将在右侧的页面中输出公司简介的完整内容，运行结果如图 10.8 所示。

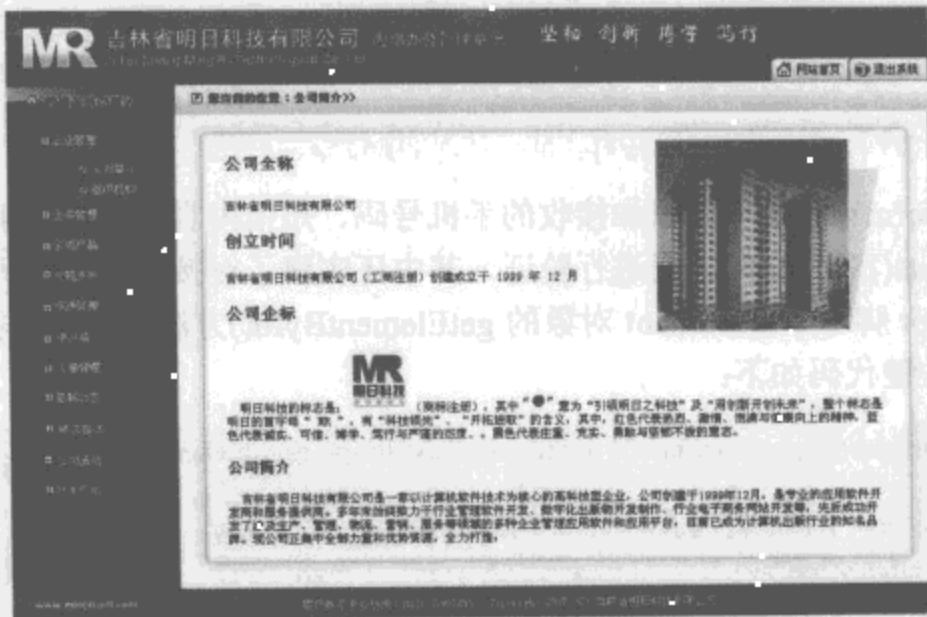


图 10.8 树状导航菜单的应用

## 10.5.2 PHP 与 JavaScript 脚本的交互应用

### 视频讲解：光盘\TM\Video\10\PHP 与 JavaScript 脚本的交互应用.exe

在动态网站的开发过程中，经常会应用 JavaScript 脚本实现一些特殊的功能。在本实例中有 4 个功能的实现都应用到了 JavaScript 脚本，分别是：（1）判断短语信息添加时提交的内容和类别是否为空；（2）通过 JavaScript 脚本无刷新验证手机号码的格式是否正确；（3）通过 JavaScript 脚本实现复选框中数据的提交、全选和反选的功能；（4）通过 JavaScript 脚本删除文本框中的短语和号码。

**例 10.8 PHP 与 JavaScript 脚本的交互应用。（实例位置：光盘\TM\Instances\10\8）**

具体开发步骤如下：

（1）首先创建 index.php 文件，添加表格，定义网站的框架，通过 include 包含语句调用指定的文

件。这里主要调用了 short\_note.php、short\_gl.php 和 select.php 3 个包含文件。

- short\_note.php：是发送短信主页面，在该页面中添加手机号码，编写短信内容。
- short\_gl.php：常用短语管理页面，实现常用短语浏览、添加和删除操作。
- select.php：输出常用短语、客户和同事的手机号码，实现对常用短语和手机号码的提交。

(2) 创建发送短信主页面 short\_note.php 文件。首先通过<script>标记中的 src 属性指定外部的 JavaScript 文件 check\_tel.js。代码如下：

```
<script src="js/check_tel.js"></script>
```

然后创建名称为 form\_reg 的表单，用于提交手机号码，并通过自定义函数 chkreginfo() 对表单中提交的手机号码进行无刷新验证。其关键代码如下：

```
<form name="form_reg" method="post">
<td width="100" align="right"><div align="right" class="STYLE1">添加手机号: </div></td>
<td width="335" align="left">
    <input type="text" id="new_tel" name="new_tel" size="40" class="inputcss" onBlur="chkreginfo(form_reg,0)"><div id="chknew_tel" style="color:#FF0000"></div></td>
<td width="120"><div align="left">
    <input name="button" type="button" class="buttoncss" value="添加">
</div></td>
</form>
```

最后创建名称为 form1 的表单，添加接收的手机号码、短信内容以及注册手机的号码和密码，并通过自定义函数 checkits() 对提交的数据进行验证。其中还实现了对接收手机号码和短信内容删除的功能，应用的是 JavaScript 脚本中 document 对象的 getElementById() 方法，通过为指定的 ID 赋值为空，完成删除的操作。其关键代码如下：

```
<form name="form1" method="post" action="" onSubmit="javascript: return checkits()">
<tr>
    <td height="50" class="STYLE1" align="right">&ampnbsp&ampnbsp接收对象: </td>
    <td colspan="2" align="left">
        <table width="386" height="50" bordercolor="#FFFFFF" bgcolor="#FFFFBF">
            <tr>
                <td width="260" rowspan="2">
                    <textarea name="touser" cols="45" rows="3" class="inputcss" id="touser" style="background-color:#e8f4ff" onMouseOver ="this.style.backgroundColor='#ffffff'" onMouseOut ="this.style. backgroundColor='#e8f4ff'"></textarea>
                </td>
                <td width="86" height="33" class="STYLE2"><!--删除电话号码-->
                    <a href="#" onClick="javascript:document.getElementById('touser').value=''; return false;">删除号码</a>
                </td>
            </tr>
            <tr>
                </td>
            </tr>
        </table>
    </td>
</tr>
<!--省略了部分代码-->
</form>
```

(3) 自定义函数 chkreginfo() 和 checkits() 都存储在 js 文件夹下的 check\_tel.js 文件中。check\_tel.js 的代码如下：

```

function checkregtel(regtel){ //通过自定义函数验证手机号码的格式是否正确
    var str=regtel;
    var Expression=/^13(\d{9})$|^15(\d{9})$/;
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
        return true;
    }else{
        return false;
    }
}

function chkreginfo(form,mark,edit){ //通过自定义函数实现对手机号码的无刷新验证
    if(mark==0 || mark=="all"){
        if(form.new_tel.value==""){
            chknew_tel.innerHTML="请输入电话号码！";
            form.new_tel.style.backgroundColor="#FF0000";
            return false;
        }else if(!checkregtel(form.new_tel.value)){ //调用自定义函数 checkregtel()验证手机号码的格式
            chknew_tel.innerHTML="电话号码的格式不正确！";
            form.new_tel.style.backgroundColor="#FF0000";
            return false;
        }else if(isNaN(form.new_tel.value)){
            chknew_tel.innerHTML="电话号由数字组成！";
            form.new_tel.style.backgroundColor="#FF0000";
            return false;
        }else{
            chknew_tel.innerHTML="";
            form.new_tel.style.backgroundColor="#FFFFFF";
        }
    }
    return true;
}

//省略了创建自定义函数 checkits() 的代码
function mail_address(mail_address){ //获取提交的数据
    var id;
    id=document.getElementById('touser').value; //将提交的电话号赋给 ID 是 touser 的文本框
    if(id==""){ //判断在文本框中是否存在重复的值
        document.getElementById('touser').value=mail_address;
    }else{
        if(id.match(mail_address)!=null){
            alert('该电话号码已经被添加！');
            return false;
        }
        //如果文本框中有多个电话号码，则以“*”作为分隔符
        document.getElementById('touser').value=id+'*'+mail_address;
    }
    return false;
}

```

(4) 创建 select.php 文件。该文件中引用了 4 个 JavaScript 脚本文件。这 4 个 JavaScript 脚本文件可以分为两类：其中一类只有一个文件 check\_list.js，实现树状导航菜单的功能；另一类包括 3 个文件 check\_submit.js、check\_submit1.js 和 check\_submit2.js，它们的功能都是相同的，实现复选框中数据的提交、全选和反选功能。select.php 文件的关键代码如下。

```

<?php session_start();include_once("conn/conn.php"); ?>           <!--连接数据库-->
<script src="js/check_submit.js"></script>                   <!--调用自定义函数，操作复选框的值 -->
<script src="js/check_submit1.js"></script>
<script src="js/check_submit2.js"></script>
<script src="js/check_list.js"></script>                   <!--实现树状导航菜单的功能 -->





```

```

a[<?php echo $m;?>]" value=<?php echo $myrow0[tb_customer_tel];?>">&nbsp;&nbsp;
</td>
</tr>
<?php
    $m++;
}
?>
<tr><!--定义提交、全选和反选按钮，并通过 onClick 事件调用指定的自定义函数-->
<td height="24" align="center">
    <input type="submit" name="Submit0" class="buttoncss" value="提交" onClick="return
del(<?php echo $m;?>)">
    <input name="button" type=button class="buttoncss" onClick="alldel(<?php echo $m;?>)"
value="全选">
    <input type=button value="反选" class="buttoncss" onClick="overdel(<?php echo $m;?>)">
</td>
</tr>
</form>
</table>
<!--省略了部分代码-->

```

这里给出的是 select.php 文件的部分代码，有关自定义函数 check\_list.js、check\_submit.js、check\_submit1.js 和 check\_submit2.js 的代码可参考光盘中的内容。

(5) 最后是创建 short\_gl.php 文件，输出常用短语。通过 insert\_note.php 文件提交常用短语，在该文件中应用自定义函数 checkit() 判断表单中提交的数据是否为空；通过 insert\_note\_ok.php 文件对表单中提交的数据进行处理；通过 delete\_colleague\_customer.php 文件删除指定的常用短语。有关这些文件的代码可参考光盘中的内容。

运行本实例，可以在添加手机号码的文本框中直接添加手机号码，指定短信的接收对象；也可以在右侧的树状导航菜单中，从客户和同事中通过复选框来选择短信的接收对象，从常用短语中选择短信祝福语；还可以通过删除短语和删除号码的按钮删除短信和接收对象的号码；最终添加注册手机号码和密码，单击“发送”按钮完成短信发送，运行结果如图 10.9 所示。

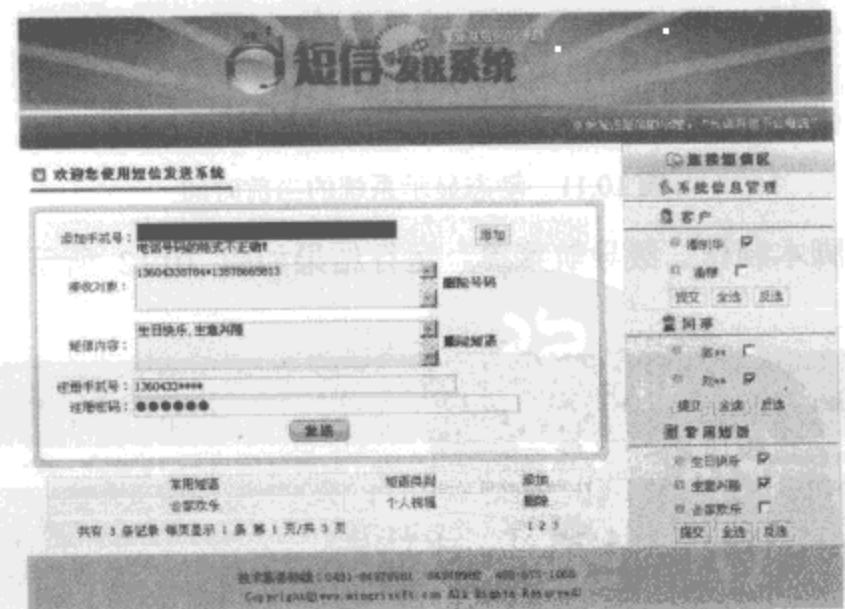


图 10.9 应用 JavaScript 脚本控制输入字符串的长度

说明：这里在处理短信群发时利用“\*”来分隔手机号码；本实例创建短信发送的操作界面，只是为了讲解 PHP 与 JavaScript 脚本的交互应用，并没有真正实现短信发送的功能。

## 10.6 本章小结

通过本章的学习，读者可以了解 JavaScript 是什么、JavaScript 能做什么以及 JavaScript 的语言基础。本章重点介绍了在 HTML 静态页和 PHP 动态页中调用 JavaScript 脚本的不同方法，以及如何自定义函数和灵活运用 JavaScript 流程控制语句。在熟悉和掌握了各个知识点后，相信读者能够举一反三，在 PHP 与 JavaScript 脚本语言的交互下开发更适用的网络程序。

## 10.7 实战练习

1. 通过 PHP 与 JavaScript 脚本交互应用，实现一个级联菜单的功能。运行结果如图 10.10 所示。  
(答案位置：光盘\TM\Instances\10\9)



图 10.10 级联菜单

2. 在 PHP 动态页中引用 JS 文件来动态显示系统的当前时间。运行结果如图 10.11 所示。  
(答案位置：光盘\TM\Instances\10\10)



图 10.11 动态显示系统的当前时间

3. 应用 JavaScript 脚本制作二级导航菜单。运行结果如图 10.12 所示。  
(答案位置：光盘\TM\Instances\10\11)



图 10.12 动态显示系统的当前时间

# 第11章 置好及

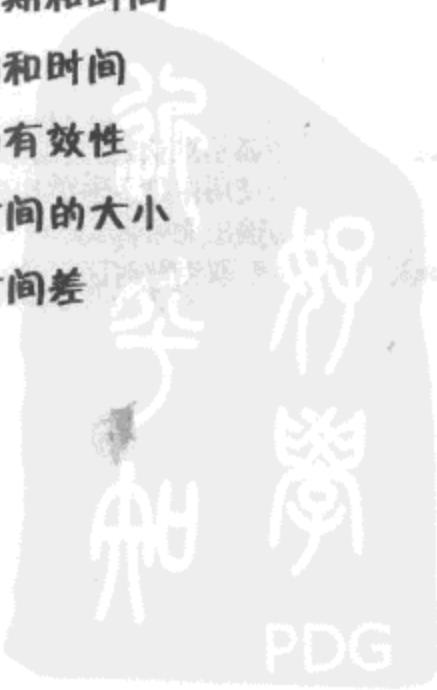
## 日期和时间

( 视频讲解：93分钟)

对日期和时间进行操作的程序无处不在，特别是在基于 Web 的应用程序中，如表单提交的时间、用户登录的时间、数据库中数据的更新和删除的时间等。想要记录这些操作执行的时刻，就需要通过日期和时间来完成。日期和时间在我们的生活中必不可少，同样，在互联网中，日期和时间也是非常重要的。本章就来讲解在 PHP 中是如何操作日期和时间的。

通过阅读本章，可以学习到：

- » **什么是时区和 UNIX 时间戳**
  - » **如何设置系统的时区**
  - » **如何获取时间戳**
  - » **如何将指定时间解析成时间戳**
  - » **如何获取当前日期和时间**
  - » **如何格式化日期和时间**
  - » **如何验证日期的有效性**
  - » **如何比较两个时间的大小**
  - » **如何灵活运用时间差**



## 11.1 系统时区设置

### 11.1.1 时区的划分

全球分为 24 个时区，每个时区都有自己的本地时间，同一时间内各时区的本地时间相差 1~23 小时，如英国伦敦本地时间与北京本地时间相差 8 个小时。在国际无线电通信领域，使用一个统一的时间，称为通用协调时间（Universal Time Coordinated, UTC），UTC 与格林威治标准时间（Greenwich Mean Time, GMT）相同。

### 11.1.2 PHP 中的时区设置

#### ■ 视频讲解：光盘\TM\Video\11\PHP 中的时区设置.exe

由于 PHP 5.0 对 date() 函数进行了重写，因此，目前的日期时间函数比系统时间少 8 个小时。PHP 语言默认设置的是标准的格林威治时间（即采用的是零时区），所以要获取本地当前的时间必须更改 PHP 语言中的时区设置。

更改 PHP 语言中的时区设置有如下两种方法。

(1) 修改 php.ini 文件中的设置，找到 [date] 下的 “;date.timezone =” 选项，将该项修改为 “date.timezone =Asia/Hong\_Kong”，然后重新启动 Apache 服务器。

(2) 在应用程序中，在使用时间日期函数之前添加如下函数：

```
date_default_timezone_set(timezone);
```

参数 timezone 为 PHP 可识别的时区名称，如果时区名称 PHP 无法识别，则系统采用 UTC 时区。在 PHP 手册中提供了各时区名称列表，其中，设置我国北京时间可以使用的时区包括 PRC（中华人民共和国）、Asia/Chongqing（重庆）、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐），这几个时区名称是等效的。

设置完成后，date() 函数便可正常使用，不会再出现时差问题。

**例 11.1 通过 date() 函数格式化输出当前时刻的 UTC 时间和北京时间，代码如下：**（实例位置：光盘\TM\Instances\11\1）

```
<?php
    echo "UTC 时间: ".date("Y-m-d H:i:s");           //显示默认的 UTC 时间
    date_default_timezone_set("PRC");
    echo "北京时间: ".date("Y-m-d H:i:s");           //使用中华人民共和国的时区
    echo "当前时区: ".date_default_timezone_get();     //输出北京时间
?>
```

运行结果如图 11.1 所示。



图 11.1 以不同的时区输出当前时间

## 11.2 UNIX 时间戳

### 11.2.1 什么是时间戳

时间戳是文件属性中的创建、修改和访问时间。数字时间戳服务(Digital Time Stamp Service, DTS)是 Web 网站安全服务项目之一，能提供电子文件的日期和时间信息的安全保护。

时间戳是一个经加密后形成的凭证文档，它包括以下 3 个部分：

- (1) 需要添加时间戳的文件用 Hash 编码加密形成摘要。
- (2) DTS 接收文件的日期和时间信息。
- (3) 对接收的 DTS 文件加密。

数字时间是由认证单位 DTS 来添加的，以 DTS 接收到文件的时间为依据。

时间戳的作用原理是通过其他加密法将时间的数值转换为加密的数值，时间变化后加密的数值也随之变化。

时间戳的优点是：变化的加密数值来防止数值被窃取后非法重复利用，也就起到了加密的作用。时间戳主要依赖于时间，在约定的一段时间内产生唯一的一个数值。

### 11.2.2 UNIX 时间戳

在 UNIX 系统中，日期与时间表示为自 1970 年 1 月 1 日零点起到当前时刻的秒数，这种时间称为 UNIX 时间戳，以 32 位二进制数表示。其中，1970 年 1 月 1 日零点称为 UNIX 世纪元。UNIX 时间戳提供了一种统一、简洁的时间表示方式，在不同的操作系统中均支持这种时间表示方式，同一时间在 UNIX 和 Windows 中均以相同的 UNIX 时间戳表示，所以不需要在不同的系统中进行转换。同时，UNIX 时间戳是一个时间差，与时区没有关系，无论当前 PHP 中使用的是何种时区，其 UNIX 时间戳是唯一的。

PHP 为 UNIX 时间戳的处理提供了各种函数。到目前的 PHP 版本为止，由于任何已知的 Windows 版本以及其他一些系统均不支持负的时间戳，因此在 Windows 中无法表示 1970 年 1 月 1 日之前的时间。目前 UNIX 时间戳是以 32 位二进制数表示的，32 位二进制数值范围为 (-2147483648 ~ +2147483647)，因此，目前 UNIX 时间戳可表示的最大时间为 2038 年 1 月 19 日 3 点 14 分 7 秒，该时刻时间戳为 2147483647，对于该时刻之后的时间，需要扩展表示 UNIX 时间戳的二进制位数。

### 11.2.3 获取指定日期的时间戳

 视频讲解：光盘\TM\Video\11\获取指定日期的时间戳.exe

PHP 中应用 `mktime()` 函数将一个时间转换成 UNIX 的时间戳值。

`mktime()` 函数根据给出的参数返回 UNIX 时间戳。时间戳是一个长整数，包含了从 UNIX 纪元（1970 年 1 月 1 日）到给定时间的秒数。其参数可以从右向左省略，任何省略的参数都会被设置成本地日期和时间的当前值。即，如果不设置任何参数，那么 `mktime()` 函数获取的将是本地的当前日期和时间。

语法：

```
int mktime(int hour, int minute, int second, int month, int day, int year, int [is_dst])
```

`mktime()` 函数的参数说明如表 11.1 所示。

表 11.1 `mktime()` 函数的参数说明

参 数	说 明
hour	小时数
minute	分钟数
second	秒数（一分钟之内）
month	月份数
day	天数
year	年份数，可以是两位或四位数字，0~69 对应于 2000—2069，70~100 对应于 1970—2000
is_dst	参数 <code>is_dst</code> 在夏令时可以被设为 1，如果不是则设为 0；如果不确定是否为夏令时则设为 -1（默认值）

 注意：有效的时间戳典型范围是格林威治时间 1901 年 12 月 13 日 20:45:54 到 2038 年 1 月 19 日 03:14:07（此范围符合 32 位有符号整数的最小值和最大值）。在 Windows 系统中此范围限制为从 1970 年 1 月 1 日到 2038 年 1 月 19 日。

例 11.2 应用 `mktime()` 函数获取系统的当前日期时间，由于返回的是时间戳，所以还要通过 `date()` 函数对其进行格式化后才能够输出日期和时间，代码如下：（实例位置：光盘\TM\Instances\11\2）

```
<?php
echo "时间戳: ".mktime()."";
//返回当前的时间戳
echo "日期: ".date("Y-m-d",mktime())."<br>";
//使用 date() 函数输出格式化后的日期
echo "时间: ".date("H:i:s",mktime());
//使用 date() 函数输出格式化后的时间
?>
```

运行结果如图 11.2 所示。

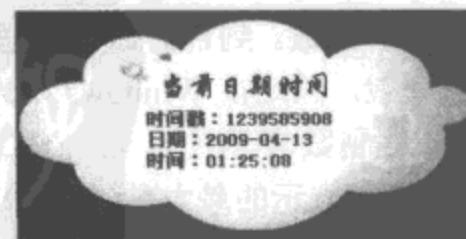


图 11.2 使用 `mktime()` 函数获取当前日期时间

### 11.2.4 获取当前时间戳

**视频讲解：**光盘\TM\Video\11\获取当前时间戳.exe

PHP 通过 time() 函数获取当前的 UNIX 时间戳，返回值为从 UNIX 纪元（格林威治时间 1970 年 1 月 1 日 00:00:00）到当前时间的秒数。语法如下：

```
int time ( void )
```

该函数没有参数，返回值为 UNIX 时间戳的整数值。

**例 11.3 应用 time() 函数获取当前时间戳，并将时间戳格式化输出，代码如下：**（实例位置：光盘\TM\Instances\11\3）

```
<?php
    $nextWeek = time() + (7 * 24 * 60 * 60);           //7 days; 24 hours; 60 mins; 60secs
    echo time()."<br>";                                //当前时间戳
    echo "Now: ".date("Y-m-d")."<br>";                //输出当前日期
    echo "Next Week: ".date("Y-m-d",$nextWeek);        //输出变量 nextweek 的日期
?>
```

运行结果如图 11.3 所示。



图 11.3 使用 time() 函数获取当前时间戳

### 11.2.5 将英文文本的日期时间描述解析为 UNIX 时间戳

PHP 中应用 strtotime() 函数将任何英文文本的日期时间解析为 UNIX 时间戳，其值为相对于 now 参数给出的时间，如果没有提供此参数则用系统当前时间。

语法：

```
int strtotime ( string time [, int now] )
```

该函数有两个参数。如果参数 time 的格式是绝对时间，则 now 参数不起作用；如果参数 time 的格式是相对时间，其对应的时间就是参数 now 来提供的，当没有提供参数 now 时，对应的时间就为当前时间。如果解析失败，则返回 false。在 PHP 5.1.0 之前的版本中，本函数在失败时返回 -1。

**例 11.4 应用 strtotime() 函数获取英文格式日期时间字符串的 UNIX 时间戳，并将部分时间输出，代码如下：**（实例位置：光盘\TM\Instances\11\4）

```
<?php
echo strtotime("now"), "\n"; //当前时间的时间戳
echo "输出时间:".date("Y-m-d H:i:s",strtotime ("now")),"<br>"; //输出当前时间
echo strtotime ("21 May 2009"), "\n"; //输出指定日期的时间戳
echo "输出时间:".date("Y-m-d H:i:s",strtotime ("21 May 2009")),"<br>"; //输出指定日期的时间
echo strtotime ("+3 day"), "\n";
echo "输出时间:".date("Y-m-d",strtotime ("+3 day")),"<br>";
echo strtotime ("+1 week")."<br>";
echo strtotime ("+1 week 2 days 3 hours 4 seconds")."<br>";
echo strtotime ("next Thursday")."<br>";
echo strtotime ("last Monday"), "\n";
?>
```

运行结果如图 11.4 所示。

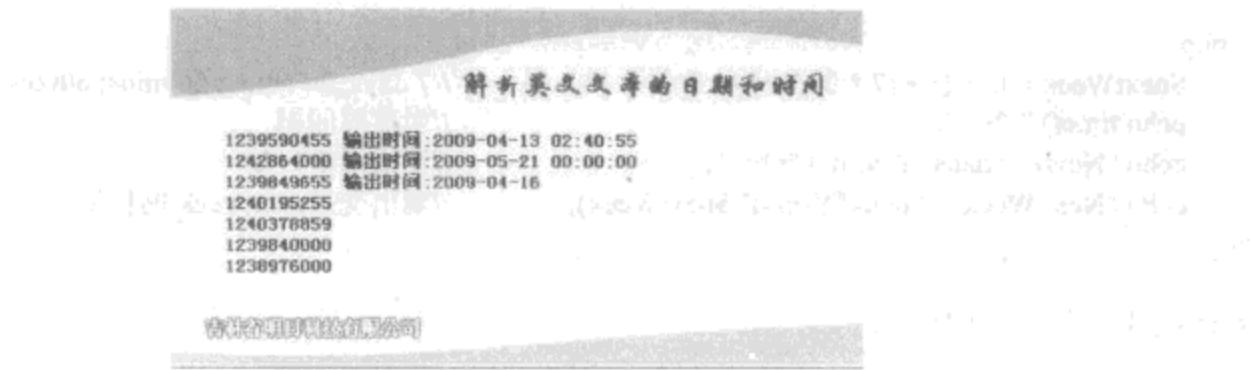


图 11.4 使用 strtotime() 函数获取当前时间戳

## 11.3 PHP 日期和时间的处理

### ■ 视频讲解：光盘\TM\Video\11\PHP 日期和时间的处理.exe

日期和时间的处理可以分为格式化日期和时间、获取日期和时间信息、获取本地化的日期和时间及检验日期和时间的有效性等。

#### 11.3.1 格式化日期和时间

PHP 中通过 date() 函数对本地日期和时间进行格式化。语法如下：

```
date(string format,int timestamp)
```

参数 format 指定日期和时间输出的格式。有关参数 format 指定的格式如表 11.2 所示。

表 11.2 参数 format 的格式化选项

参 数	说 明
a	小写的上午和下午值，返回值 am 或 pm
A	大写的上午和下午值，返回值 AM 或 PM
B	Swatch Internet 标准时间，返回值 000~999

续表

参数	说 明
d	月份中的第几天，有前导零的两位数字，返回值 01~31
D	星期中的第几天，文本格式，3 个字母，返回值 Mon 到 Sun
F	月份，完整的文本格式，返回值 January 到 December
G	小时，12 小时格式，没有前导零，返回值 1~12
H	小时，24 小时格式，没有前导零，返回值 0~23
i	有前导零的分钟数，返回值 00~59
I	判断是否为夏令时，返回值如果是夏令时为 1，否则为 0
J	月份中的第几天，没有前导零，返回值 1~31
l	星期数，完整的文本格式，返回值 Sunday 到 Saturday
L	判断是否为闰年，返回值如果是闰年为 1，否则为 0
m	数字表示的月份，有前导零，返回值 01~12
M	3 个字母缩写表示的月份，返回值 Jan 到 Dec
n	数字表示的月份，没有前导零，返回值 1~12
o	与格林威治时间相差的小时数，如 0200
r	RFC 822 格式的日期，如 Thu, 21 Dec 2000 16:01:07 +0200
s	秒数，有前导零，返回值 00~59
S	每月天数后面的英文后缀，两个字符，如 st、nd、rd 或者 th。可以和 j 一起使用
t	指定月份所应有的天数
T	本机所在的时区
U	从 UNIX 纪元 (January 1 1970 00:00:00 GMT) 开始至今的秒数
w	星期中的第几天，数字表示，返回值为 0~6
W	ISO-8601 格式年份中的第几周，每周从星期一开始
y	两位数字表示的年份，返回值如 88 或 08
Y	4 位数字完整表示的年份，返回值如 1998、2008
z	年份中的第几天，返回值 0~366
Z	时差偏移量的秒数。UTC 西边的时区偏移量总是负的，UTC 东边的时区偏移量总是正的，返回值为 -43200~43200

参数 timestamp 是可选的，用于指定时间戳，如果没有给出时间戳则使用本地当前时间 time()。

有关通过 date() 函数获取系统当前时间的方法在前面的实例中已经应用过，这里不再赘述。

**例 11.5 应用 date() 函数对日期进行格式化，设置不同的参数，进而输出不同格式的日期，代码如下：**（实例位置：光盘\TM\Instances\11\5）

```
<?php
    echo date("Y-m-d")."<br>";                                //2009-04-13
    echo date("m.d.y")."<br>";                                //04.13.09
    echo date("j, n, Y")."<br>";                                //13, 4, 2009
    echo date("F j, Y, g:i a")."<br>";                          //April 13, 2009, 3:19 am
    echo date("D M j G:i:s T Y")."<br>";                      //Mon Apr 13 3:19:26 UTC 2009
    echo date("\t \s \t\h\le jS \d\al\y")."<br>";                //It is the 13th day
```

```

echo date("H:i:s 这是当前时间")."<br>"; //03:19:26 这是当前时间
echo date('h-i-s, j-m-y,这是我的一天')."<br>"; //03-19-26, 13-04-09,这是我的一天
?>

```

运行结果如图 11.5 所示。

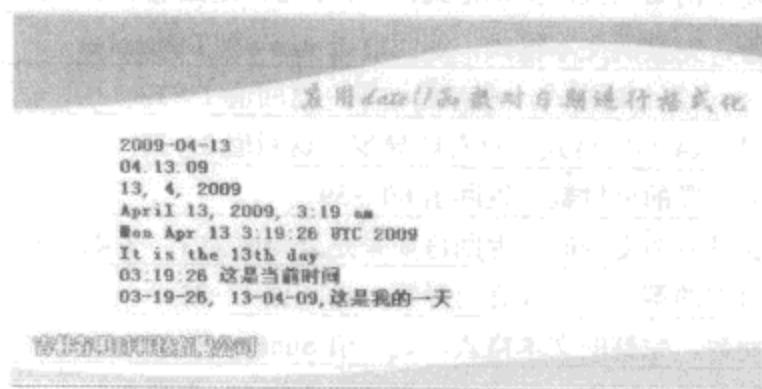


图 11.5 应用 date() 函数对日期进行格式化

**◆ 注意：**在运行本章的实例时，也许有的读者得到的时间和系统时间并不相等，这不是程序的问题。因为在 PHP 语言中默认设置的是标准的格林威治时间，而不是北京时间。如果出现了时间不相符的情况，读者可参考 11.1.2 节中的内容。

### 11.3.2 获取日期和时间信息

PHP 中通过 getdate() 函数获取日期和时间指定部分的相关信息。语法如下：

```
array getdate(int timestamp)
```

该函数返回数组形式的日期、时间信息，如果没有时间戳，则以当前时间为基准。该函数返回的关联数组元素的说明如表 11.3 所示。

表 11.3 getdate() 函数返回的关联数组元素说明

函 数	说 明
seconds	秒，返回值为 0~59
minutes	分钟，返回值为 0~59
hours	小时，返回值为 0~23
mday	月份中第几天，返回值为 1~31
wday	星期中第几天，返回值为 0（表示星期日）~6（表示星期六）
mon	月份数字，返回值为 1~12
year	4 位数字表示的完整年份，返回的值如 2000 或 2008
yday	一年中第几天，返回值为 0~365
weekday	星期几的完整文本表示，返回值为 Sunday 到 Saturday
month	月份的完整文本表示，返回值为 January 到 December
0	返回从 UNIX 纪元开始的秒数

例 11.6 应用 getdate() 函数获取系统当前的日期信息，并输出返回值，代码如下：（实例位置：

光盘\TM\Instances\11\6)

```
<?php
    $arr = getdate();
    echo $arr[year]."-".$arr[mon]."-".$arr[mday]." ";
    //使用 getdate()函数将当前信息保存
    echo $arr[hours].":".$arr[minutes].":".$arr[seconds]." ".$arr[weekday];
    //返回当前的时间信息
    echo "Today is the $arr[yday]th of year";
    //输出今天是一年中的第几天
?>
```

运行结果如图 11.6 所示。

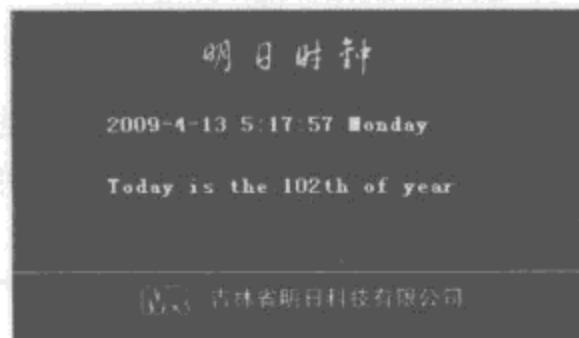


图 11.6 用 getdate() 函数获取时间信息

### 11.3.3 获取本地化的日期和时间

不同的国家和地区，使用不同的时间、日期、货币的表示法，以及不同的字符集。例如，在大多数西方国家，都使用 Friday，但在以汉语为主的国家中，都使用星期五，虽然都是同一个含义，但表示的方式却不尽相同。这时，就需要设置本地化环境。

在 PHP 中应用 setlocale() 函数设置本地化环境，应用 strftime() 函数根据区域设置格式化本地日期和时间。

#### 1. setlocale() 函数

setlocale() 函数可以改变 PHP 默认的本地化环境。语法如下：

```
string setlocale(string category, string locale)
```

参数 category 的选项如表 11.4 所示。

参数 locale 如果为空，就会应用系统环境变量的 `locale` 或 `LANG` 的值；否则，就会应用 `locale` 参数所指定的本地化环境。如“`en_US`”为美国本地化环境，“`chs`”则指简体中文，“`cht`”为繁体中文。

**说明：**对于 Windows 平台的用户，可以登录 <http://msdn.microsoft.com> 来获取语言和国家（地区）的编码列表。如果是 UNIX/Linux 系统，则可以使用 `locale -a` 命令来确定所支持的本地化环境。

表 11.4 category 参数选项及说明

参 数	说 明
<code>LC_ALL</code>	包含了下面所有的设置本地化规则
<code>LC_COLLATE</code>	字符串比较
<code>LC_CTYPE</code>	字符串分类和转换，如转换大小写

(待续表)

参 数	说 明
LC_MONETARY	本地化环境的货币形式
LC_NUMERIC	本地化环境的数值形式
LC_TIME	本地化环境的时间形式

## 2. strftime()函数

strftime()函数根据区域设置来格式化输出日期和时间。语法如下：

```
string strftime(string format, int timestamp)
```

参数 format 指定日期和时间输出的格式。有关参数 format 识别的转换标记如表 11.5 所示。

参数 timestamp 指定时间戳，如果没有指定时间戳则应用本地时间。

表 11.5 参数 format 识别的转换标记

参 数	说 明
%a	星期的简写
%A	星期的全称
%b	月份的简写
%B	月份的全称
%c	当前区域首选的日期时间表达
%C	世纪值（年份除以 100 后取整，范围从 00~99）
%d	月份中的第几天，十进制数字（范围从 01~31）
%D	和%m/%d/%y 相同
%e	月份中的第几天，十进制数字，一位的数字前会加上一个空格（范围从'1'~'31'）
%g	和%G 相同，但是没有世纪值
%G	4 位数的年份，符合 ISO 星期数（参见%V）。与%V 的格式和值相同，但如果 ISO 星期数属于前一年或者后一年，则使用那一年
%h	和%b 相同
%H	24 小时制的十进制小时数（范围从 00~23）
%I	12 小时制的十进制小时数（范围从 00~12）
%j	年份中的第几天，十进制数（范围从 001~366）
%m	十进制月份（范围从 01~12）
%M	十进制分钟数
%n	换行符
%p	根据给定的时间值为 am 或 pm，或者当前区域设置中的相应字符串
%r	用 a.m 和 p.m 符号表示的时间
%R	24 小时符号的时间
\$S	十进制秒数
%t	制表符
%T	当前时间，和%H:%M:%S 相同
%u	星期几的十进制数表达[1,7]，1 表示星期一
%U	本年的第几周，从第一周的第一个星期天作为第一天开始

续表

参数	说明
%V	本年第几周的 ISO 8601:1988 格式，范围从 01~53，第一周是本年第一个至少还有 4 天的星期，星期一作为每周的第一天（用%G 或者%g 作为指定时间戳相应周数的年份组成）
%W	本年的第几周数，从第一周的第一个星期一作为第一天的开始
%w	星期中的第几天，星期天为 0
%x	当前区域首选的时间表示法，不包括时间
%X	当前区域首选的时间表示法，不包括日期
%y	没有世纪数的十进制年份（范围从 00~99）
%Y	包括世纪数的十进制年份
%Z	时区名或缩写
%%	文字上的%字符

**例 11.7 应用 setlocale() 函数进行区域化设置，然后通过 strftime() 函数对本地时间进行格式化输出，代码如下：**（实例位置：光盘\TM\Instances\11\7）

```
<?php
    setlocale(LC_ALL,"en_US");
    echo "美国格式: ".strftime("Today is %A");
    echo "<p>";
    setlocale(LC_ALL,"chs");
    echo "中文简体格式: ".strftime("今天是%A");
    echo "<p>";
    setlocale(LC_ALL,"cht");
    echo "<p>";
    echo "繁体中文格式: ".strftime("今天是%A");
?
>
```

运行结果如图 11.7 所示。



图 11.7 本地化日期

**说明：**因为本页面中的编码格式为 GB2312，所以最后繁体中文显示的日期为乱码，如果将编码格式改为 big5，繁体中文星期将会正确显示出来，但其他文字则变为乱码。读者可以选择“查看”/“编码”命令，在弹出的菜单中选择“繁体中文(big5)”命令来查看效果。

#### 11.3.4 检验日期和时间的有效性

一年有 12 个月，一个月有 31 天（或 30 天、28 天、29 天），一星期有 7 天……这些常识都是人

尽皆知的事。但计算机并不能自己分辨数据的对与错，只能依靠开发者提供的功能去执行或检查。PHP 中通过 `checkdate()` 函数检验日期和时间的有效性。语法如下：

```
bool checkdate(int month,int day,int year)
```

其中，`month` 的有效值为 1~12。`day` 的有效值为当月的最大天数，如 1 月为 31 天，2 月为 29 天（闰年）。`year` 的有效值为 1~32767。

**例 11.8 应用 `checkdate()` 函数验证数据录入系统中输入的日期是否合理，如果合理则返回“数据录入成功”，否则返回“您输入的日期不合法！！”，操作步骤如下：(实例位置：光盘\TM\Instances\11\8)**

(1) 创建 `index.php` 页面，添加 `form` 表单，设置表单元素，用于提交商品的录入信息，将表单中的数据提交到 `index_ok.php` 文件中。

(2) 创建 `index_ok.php` 文件，通过 `$_POST` 全局变量获取表单中提交的数据，应用 `checkdate()` 函数对表单中提交的日期进行判断，代码如下。

```
<?php
if($_POST[Submit]==true and $_POST[number]==true and $_POST[name]==true and $_POST[day]==true){
    //判断表单中提交的数据是否为空
    if(checkdate($month,$day,$year)==true){ //验证表单中提交的日期是否合理
        echo "数据录入成功!!";
    }else{
        echo "<script> alert('您输入的日期不合法!!'); history.back();</script>";
    }
}else{
    echo "<script> alert('请详细填写录入数据!!'); history.back();</script>";
}
?>
```

运行结果如图 11.8 所示。



图 11.8 使用 `checkdate()` 函数验证日期

## 11.4 实战

### 11.4.1 实现倒计时的功能

■ 视频讲解：光盘\TM\Video\11\实现倒计时的功能.exe

**例 11.9 应用 `strtotime()` 函数精确计算两个时间的差值，实现一个倒计时的功能。实例代码如下：**

(实例位置: 光盘\TM\Instances\11\9)

```
<?php
    $time1 = strtotime(date("Y-m-d H:i:s"));
    $time2 = strtotime("2009-10-01 00:00:00"); //2009 年国庆节
    $time3 = strtotime("2010-01-01 "); //2009 年元旦
    $sub1 = ceil(($time2 - $time1) / 3600); //60 * 60
    $sub2 = ceil(($time3 - $time1) / 86400); //60 * 60 * 24
    echo "距离 2009 年国庆节还有<font color=red> $sub1 </font>小时!!!";
    echo "<p>";
    echo "距离 2009 年元旦还有<font color=red>$sub2 </font>天!!!";
?

```

说明: ceil() 函数的格式为 float ceil(float value), 该函数为取整函数, 返回不小于参数 value 值的最小整数。如果有小数部分, 则进一位。这里要注意, 该函数的返回类型为 float 型, 而不是整型。

运行结果如图 11.9 所示。

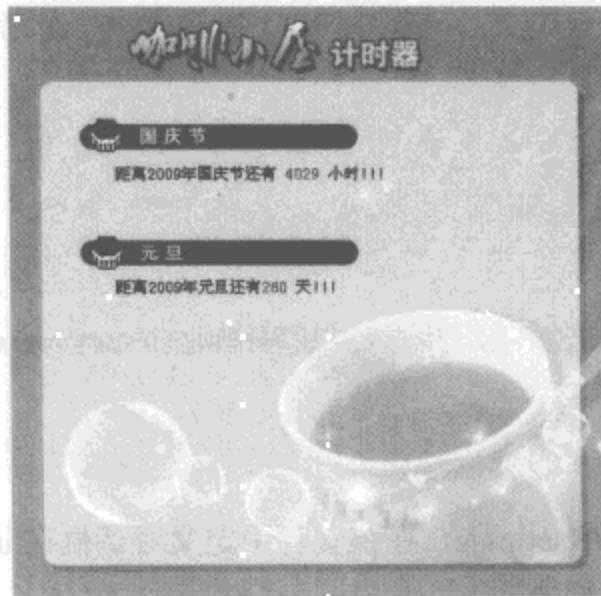


图 11.9 实现倒计时的功能

#### 11.4.2 实现生日提示功能

视频讲解: 光盘\TM\Video\11\实现生日提示功能.exe

例 11.10 在公司内部, 每当有员工过生日的时候, 公司都会送给员工一个精致的礼物。为了能够准确记录每位员工的生日, 在公司内部管理系统中实现这样一个功能: 在每天登录公司内部管理系统时, 系统中会自动提示在 5 天之内哪位员工将要过生日, 以及当天有哪位员工过生日。(实例位置: 光盘\TM\Instances\11\10)

具体操作步骤如下:

(1) 创建公司内部管理系统的登录页面 index.php, 添加 form 表单, 将用户名和密码提交到 index\_ok.php 页面。

(2) 创建 index\_ok.php 页面，获取登录提交的用户名和密码，判断用户名和密码是否正确，如果用户名和密码正确，则进入到公司内部管理系统的主页面，并通过<script>标记调用 JS 文件，弹出一个提示对话框，在对话框中输出生日提示信息；如果用户名和密码不正确，则弹出提示信息“对不起，用户名或密码错误！”，并跳转到登录页面，其关键代码如下。

```
<?php
    session_start(); //初始化 Session 变量
    include_once("conn/conn.php"); //连接数据库
    if($_POST[username]==true and $_POST[userpwd]==true){
        $result=mysql_query("select * from tb_manage where username='".$_POST[username]."' and password='".$_POST[userpwd]."'");
        if($result){
            $_SESSION[username]=$_POST[username]; //定义 session 变量
        }
    }
    else{
        echo "<script>alert('对不起，用户名或密码错误!');history.back();</script>";
    }
?
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>实现生日提示功能</title>
</head>
<script language='javascript' src='js/script.js'></script>
<body>
<!--省略了主页中的内容-->
</body>
</html>
<?php
```

(3) 创建 JavaScript 脚本文件 script.js，在该文件中定义对话框在 index\_ok.php 页面中输出的位置，并定义调用的文件 cue.php，代码如下。

```
var newformW=300; //设置弹出对话框页面的宽
var newformH=180; //设置弹出对话框页面的高
function pp(){
    var T>window.screen.width-newformW; //定义在页面中的位置
    var L>window.screen.height+newformH; //定义在页面中的位置
    var name>window.open("cue.php","","width="+newformW+",height="+newformH+",top="+T+",left="+L);
}
pp(); //执行自定义函数
```

(4) 创建 cue.php 文件，获取当日生日提示信息。

解析一：判断是否是以管理员身份登录，如果不是则给出提示信息；如果是则继续执行，代码如下。

```
<?php
    session_start(); //初始化 Session 变量
    if($_SESSION[username]!=""){
        //判断是否为正确登录
```

```
//省略了执行代码
<?php
    }else{
        echo "您没有正确登录!";
    }
?>
```

解析二：连接数据库，获取当前时间，分别定义存储在 5 天之内过生日和当日过生日的员工姓名的数组，代码如下。

```
include_once("conn/conn.php"); //连接数据库
$dates = date("Y-m-d"); //获取当前时间
$array_birth=array(); //定义存储在 5 天之内过生日的人姓名的数组
$array_births=array(); //定义存储当天过生日的人姓名的数组
```

解析三：执行查询语句，查询出数据库中存储的员工生日数据，并通过 while 循环输出生日数据，应用 substr() 函数截取生日数据中的月和日，截取当前时间数据中的年份，最终定义一个新的生日字符串，代码如下。

```
$sql=mysql_query("select * from tb_employee"); //查询数据库中的数据
while($myrow=mysql_fetch_array($sql,$con)){
    $birth=substr($myrow[birth],5,10); //截取用户生日数据的月和日
    $dates1=substr($dates,0,5); //截取当前日期中的年份
    $dates2=$dates1.$birth; //定义新的生日字符串
```

解析四：通过 strtotime() 函数将生日日期和当前日期解析为 UNIX 时间戳，应用生日日期时间戳减去当日日期时间戳，其返回结果以天为单位，代码如下。

```
$time1=strtotime($dates2); //将生日日期转换为时间戳
$time2=strtotime($dates); //将当前日期转换为时间戳
$date_birth=ceil(($time1-$time2)/86400); //用生日日期减去当前日期，结果以天为单位
```

解析五：通过 if 语句进行判断，如果返回结果大于 0 并且小于等于 5，则说明该员工在 5 天之内过生日，则将该员工的姓名存储到一个数组中；如果返回的结果等于 0，则说明该员工当日过生日，则将该员工的名称存储到另一个数组中，代码如下。

```
if($date_birth<=5 and $date_birth>0){ //判断如果结果小于等于 5，大于 0
    array_push($array_birth,$myrow[name]);
} else if($date_birth==0){ //如果结果等于 0，则将该用户添加到另一数组中
    array_push($array_births,$myrow[name]);
} else{
    echo "广告内容!";
}
?>
```

解析六：判断数组中的元素是否为空，如果不为空，则通过 for 循环输出数组中的元素，即输出在 5 天之内过生日的员工和当日过生日的员工，代码如下。

```

<?php
    if(count($array_birth)>0){
?>
<table width="350" border="0" cellpadding="0" cellspacing="0">
    <tr><td width="350" height="30" align="center">近期将有下面几位过生日</td></tr>
    <tr>
        <td height="22" align="center">
<?php
    for($m=0;$m<count($array_birth);$m++){
        echo $array_birth[$m]."&nbsp;&nbsp;";
    }
?>
        </td>
    </tr>
    <tr><td height="22" align="center">不要忘记哦！</td></tr>
</table>
<?php } ?>
//省略了输出当日过生日员工的代码

```

运行结果如图 11.10 所示。



图 11.10 实现生日提示功能

### 11.4.3 计算在线考试用时和剩余时间

视频讲解：光盘\TM\Video\11\计算在线考试用时和剩余时间.exe

**例 11.11** 在线考试系统中，多数都是在规定的时间内完成考试，如果超过规定时间，或者自动提交答案，或者自动放弃考试。本实例中将要讲解的就是如何计算考试的用时和剩余时间。（实例位置：光盘\TM\Instances\11\11）

具体操作步骤如下：

(1) 创建 index.php 页面，首先介绍考试规则；然后添加 form 表单，设置“开始考试”按钮，提交到 exam.php 页面；最后，应用 mtime() 函数获取当前日期的时间戳，并把时间戳的值赋给 session 变量，代码如下。

```
<?php
    session_start(); // 初始化 Session 变量
?>
<form name="form2" method="post" action="exam.php">
    <input type="submit" name="Submit" value="开始考试" />
<?php
    $dates=mtime(); // 获取当前时间的时间戳
    session_register("dates")==$dates;// 将时间戳的值赋给 session 变量
?>
</form>
```

 注意：在线考试时间的计算起点就是从单击“开始考试”按钮开始。

(2) 创建 exam.php 文件。首先通过<script>标记中的 src 属性，调用屏蔽键盘事件和计算考试时间的 JS 文件；然后在 body 标记中，通过 onkeydown() 事件调用 JavaScript 自定义函数，实现对键盘中事件的屏蔽；接着，通过 div 标签输出考试用时和剩余时间；最后，添加 form 表单，输出考试试题，设置提交按钮。代码如下。

```
<?php session_start(); ?>
<!-- 应用 script 标记中的 src 属性，调用屏蔽键盘事件和计算考试时间的 JS 文件--&gt;
&lt;script language=javascript src="js/screen_event.js"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="js/calculation_time.js"&gt;&lt;/script&gt;
&lt;body onkeydown="keydown()"&gt;
&lt;table width="750" bordercolor="#FFFFFF" bgcolor="#666666"&gt;
    &lt;tr&gt;
        &lt;td width="165" height=23&gt;&lt;span class="STYLE1"&gt;考试时间：&lt;/span&gt;&lt;/td&gt;
        &lt;td width="42" &gt;&lt;span class="STYLE1"&gt;&lt;font color="#FF0000"&gt;20&lt;/font&gt;分钟&lt;/span&gt;&lt;/td&gt;
        &lt;td width="58" &gt;&lt;span class="STYLE1"&gt;计时&lt;/span&gt;&lt;/td&gt;
        &lt;td width="193" nowrap bgcolor="#EEEEEE"&gt;&lt;div class="STYLE1" id="show_time"&gt;&lt;/div&gt;&lt;/td&gt;
        &lt;td width="77" &gt;&lt;span class="STYLE1"&gt;剩余时间：&lt;/span&gt;&lt;/td&gt;
        &lt;td width="182" &gt;&lt;div class="STYLE1" id="sparetime"&gt;&lt;/div&gt;&lt;/td&gt;
    &lt;/tr&gt;
&lt;/table&gt;
<!-- 省略了添加的 form 表单中的内容 --&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>

```

(3) 创建 calculation\_time.js 文件，通过 Ajax 无刷新技术计算和输出考试用时和剩余时间，关键代码如下。

```
var xmlhttp=false; // 创建一个布尔型变量，用来检测是否为合法的 IE 实例
try {
    xmlhttp=new ActiveXObject("Msxml2.XMLHTTP"); // 检测是否使用的是 IE
    if (xmlhttp==null) // 如果 JavaScript 的版本大于 5
        xmlhttp=new XMLHttpRequest();
```

```

} catch (e) {                                //如果不是，则使用老版本的 ActiveX 对象
    try {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e2) {}
}
/* *****如果使用的是非 IE 浏览器，则创建一个该对象的 JavaScript 实例***** */
if (!xmlhttp && typeof XMLHttpRequest != "undefined") {
    try{
        xmlhttp = new XMLHttpRequest();
    }catch(e3){ xmlhttp = false;}
}

timer = window.setInterval("ShowTime()",1000);           //每隔一秒钟调用一次 ShowTime()函数
//定义 ShowTime()函数以通过 XMLHttpRequest 对象读取 ShowTime.php 文件中的数据
function ShowTime(){
    xmlhttp.open("post","showtime.php", true);           //以 POST 方法发送一个新请求
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){                      //如果服务器响应发出的请求，则执行以下操作
            tet = xmlhttp.responseText;                  //获取返回的响应信息
            /* *****将获取到的信息赋予指定的 DIV 标记***** */
            document.getElementById("show_time").innerHTML = tet;
        }
    }
    xmlhttp.send(null);                                //发送请求
}

time = window.setInterval("sparetime()",1000);           //每隔一秒钟调用一次 sparetime()函数
/* *****定义 sparetime()函数以通过 XMLHttpRequest 对象读取 sparetime.php 文件中的数据***** */
function sparetime(){
    xmlhttp.open("post","sparetime.php", true);           //以 POST 方法发送一个新请求
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){                      //如果服务器响应发出的请求，则执行以下操作
            tet = xmlhttp.responseText;                  //获取返回的响应信息
            /* *****将获取到的信息赋予指定的 DIV 标记***** */
            document.getElementById("sparetime").innerHTML = tet;
            if(tet=="00:00"){                            //当剩余时间为 00:00 时
                form1.submit();                          //提交 form1 表单中的数据
            }
        }
    }
    xmlhttp.send(null);                                //发送请求
}

```

- (4) 在 calculation\_time.js 文件中，自定义函数 ShowTime() 调用 showtime.php 文件，获取考试用时；自定义函数 sparetime() 调用 sparetime.php 文件，获取考试的剩余时间。
- showtime.php 文件中计算的是考试用时，代码如下。

```
<?php
    session_start(); //初始化 Session 变量
    $dates=$_SESSION[dates]; //获取 session 变量中存储的考试开始时间
    $dates2=mktime(); //获取当前时间的时间戳
    $dates3=$dates2-$dates; //用当前时间的时间戳减去考试开始时的时间戳
    echo date("i:s",$dates3); //输出考试当前用时
?>
```

sparetime.php 文件中计算的是考试剩余时间，代码如下。

```
<?php
    session_start(); //初始化 Session 变量
    $dates=$_SESSION[dates]; //获取 session 中存储的考试开始的时间戳
    $dates1=$dates+20*60; //计算出考试规定的用时，时间为 20 分钟
    $dates2=mktime(); //获取当前时间的时间戳
    $dates3=$dates1-$dates2; //用考试规定的时间减去当前时间的时间戳
    echo date("i:s",$dates3); //输出考试的剩余时间
?>
```

(5) 创建 screen\_event.js 文件，编写自定义函数，屏蔽键盘和鼠标事件。

(6) 创建 exam\_result.php 文件，输出考试的结果。

在线考试的主页面运行结果如图 11.11 所示。

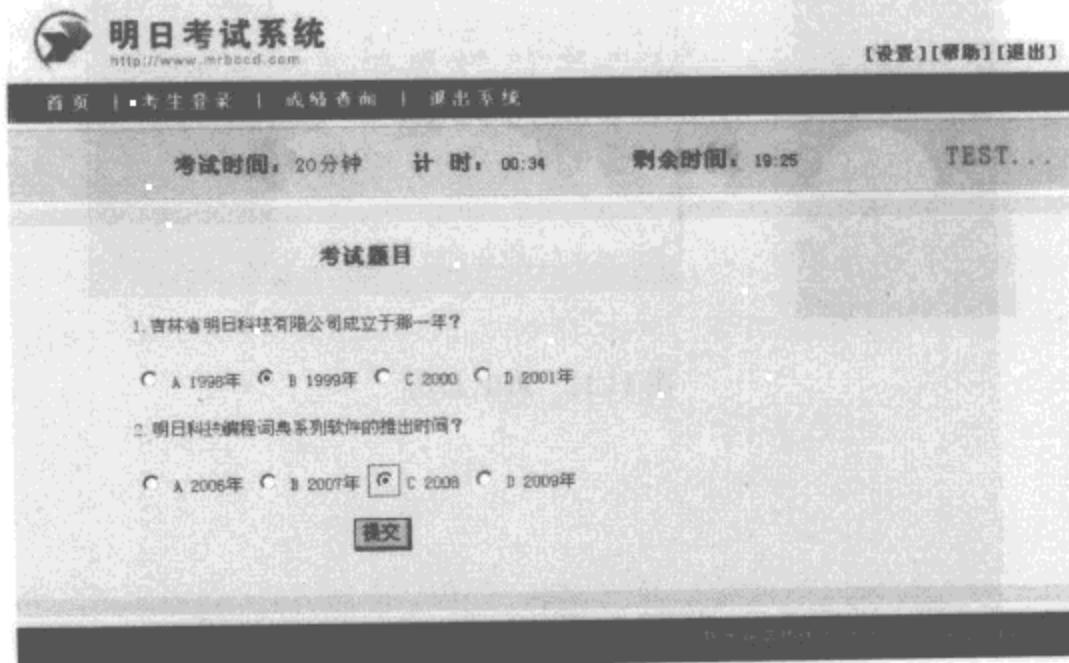


图 11.11 计算在线考试用时和剩余时间

## 11.5 本章小结

本章讲解了 PHP 中常用的日期和时间处理函数，并且通过实例讲解了这些函数的典型应用。首先，介绍时区的划分和系统时区的设置；然后，介绍 UNIX 时间戳，包括什么是时间戳、如何获取时间戳，

以及如何将英文文本格式的日期时间解析为时间戳；接着，讲解 PHP 中日期和时间的处理方法，包括日期和时间的格式化、获取和验证有效性等；最后，通过实战和实战练习讲解 PHP 中日期和时间函数的应用。希望通过本章的学习，读者可以掌握 PHP 中常用的日期和时间函数，熟悉常用功能的实现方法。

## 11.6 实战练习

1. 将时间戳作为随机数的种子，生成验证码。（答案位置：光盘\TM\Instances\11\12）
2. 计算页面的运行时间。（答案位置：光盘\TM\Instances\11\13）
3. 在用户注册模块中应用 PHP 日历。运行结果如图 11.12 所示。（答案位置：光盘\TM\Instances\11\14）

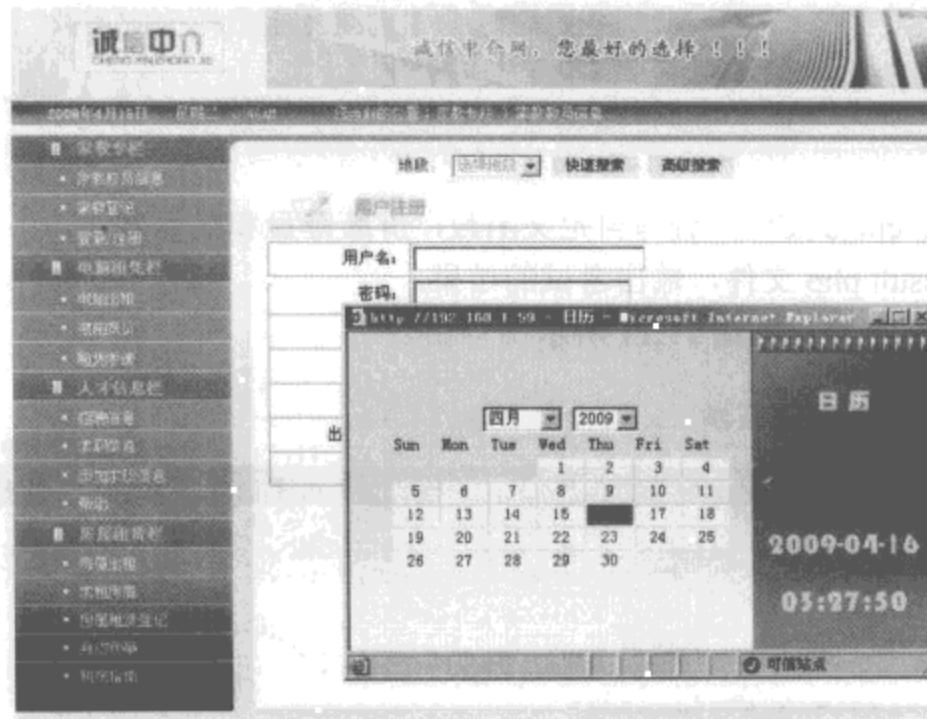


图 11.12 PHP 日历

## 第2部分

### 核心技术

- 第12章 会话管理
- 第13章 图形图像
- 第14章 文件系统
- 第15章 面向对象
- 第16章 PHP 加密技术
- 第17章 MySQL 数据库基本操作
- 第18章 phpMyAdmin 图形化管理工具
- 第19章 使用 MySQL 函数操作数据库
- 第20章 ADODB 类库
- 第21章 程序调试与异常处理



# 木鼓心经

本草纲目 卷之三十一

甘草 甘草根 甘草茎 甘草皮

大枣 大枣肉 大枣核 大枣壳

蜜炙大枣 蜜炙大枣肉 蜜炙大枣核

本草纲目 卷之三十二

甘草子 甘草子根 甘草子茎 甘草子皮

甘草子根甘草子茎甘草子皮甘草子壳

甘草子根甘草子茎甘草子皮甘草子壳

甘草子根甘草子茎甘草子皮甘草子壳

甘草子根甘草子茎甘草子皮甘草子壳

甘草子  
甘草子根  
甘草子茎  
甘草子皮  
甘草子壳  
PDG

# 第12章

## 会话管理

( 视频讲解: 117分钟)

Cookie 和会话完美地解决了 HTTP Web 协议的无状态，被广泛应用在网站开发领域，用于保持用户信息，以及追踪个人和商业交易等。Cookie 适用于持久数据的存储，例如，在论坛中用户都希望登录一次后，下次登录就不再需要再输入登录信息了；而 Session 则适用于单次访问的情况，如网上银行，如果用户异常断线或者登录超时，Session 就会根据时间自动中断连接。可以说，掌握 Cookie 和 Session 技术，对于 Web 网站的开发是至关重要的。

通过阅读本章，可以学习到：

- » Cookie 是什么以及 Cookie 能做什么
- » 如何创建 Cookie
- » 读取 Cookie 的方法
- » 删除 Cookie 的两种方法
- » Cookie 的生命周期
- » 运行 Cookie 技术开发 Web 网站
- » Session 是什么以及 Session 能做什么
- » 启动 Session、应用 Session 和删除 Session 的方法
- » 运用 Session 技术开发 Web 网站
- » Cookie 与 Session 之间的区别

## 12.1 Cookie 管理

Cookie 是在 HTTP 协议下，将服务器传递给浏览器的少量数据保存到用户浏览器的一种方式。通过这种方式，即使在浏览器被关闭和连接中断的情况下，用户仍然可以维护状态数据。

Cookie 的使用很普遍，许多提供个性化服务的网站，都是利用 Cookie 来辨认使用者，以便于送出使用者量身定做的内容，如 Web 接口的免费 E-mail 网站都要用到 Cookie。有效地使用 Cookie 可以轻松完成很多复杂的任务。下面对 Cookie 的相关知识进行详细介绍。

### 12.1.1 了解 Cookie

#### ■ 视频讲解：光盘\TM\Video\12\了解 Cookie.exe

本节将简单介绍 Cookie 是什么、Cookie 能做什么以及如何删除和阻止 Cookie。希望读者通过本节的学习对 Cookie 有一个明确的认识。

Cookie 的概念最早是由 Netscape 在 1994 年提出来的，它是保存在浏览器中的小信息包。更确切地说，Cookie 是保存在用户硬盘上的由字符串组成的小文本文件。

文本文件的命令格式如下：

用户名@网站地址[数字].txt

举个简单的例子，如果用户的系统盘为 C 盘，操作系统为 Windows 2000/XP/2003，当使用 IE 浏览器访问 Web 网站时，Web 服务器会自动以上述的命令格式生成相应的 Cookies 文本文件，并存储在用户硬盘的指定位置，如图 12.1 所示。

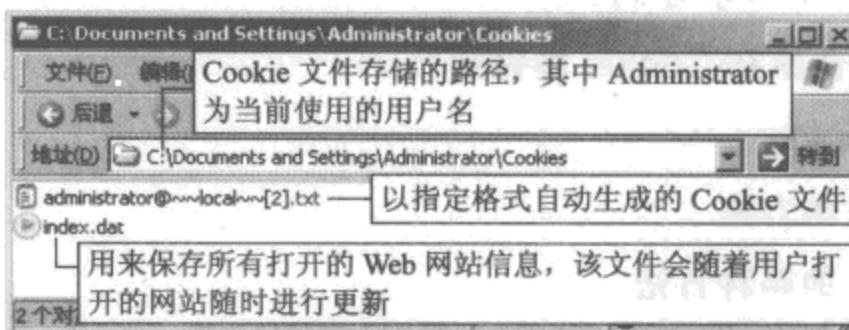


图 12.1 Cookie 文件的存储路径

**注意：**在 Cookies 文件夹下，每个 Cookie 文件都是一个简单而又普通的文本文件，而不是程序。Cookies 中的内容大多经过了加密处理，因此，表面看来只是一些字母和数字组合，而只有服务器的 CGI 处理程序才知道它们真正的含义。

Cookie 可以让 Web 页面更有针对性、更加友好，保存关于用户的重要信息，包括使用的语言、阅读和音乐偏好，访问站点的次数，购物车中的物品记录。Cookie 常用于以下 3 个方面：

- 记录访客的某些信息。例如，可以利用 Cookie 记录用户访问网页的次数，或者记录访客曾经输入过的信息，另外，某些网站可以应用 Cookie 自动记录访客上次登录的用户名。

- 在页面之间传递变量。浏览器并不会保存当前页面上的任何变量信息，当页面被关闭后，页面上的任何变量信息将随之消失。如果用户声明一个变量 `id=8`，要把这个变量传递到另一个页面，可以把变量 `id` 以 Cookie 的形式保存下来，然后在下一页通过读取该 Cookie 来获取变量的值。
- 将所查看的 Internet 页存储在 Cookies 临时文件夹中，这样可以提高以后浏览的速度。

**注意：**一般不要用 Cookie 来保存数据集或其他大量数据。并非所有的浏览器都支持 Cookie，并且数据信息是以明文文本的形式保存在客户端计算机中，因此最好不要保存敏感的、未加密的数据，否则会影响网络的安全性。

Cookie 也有一定的负作用，有用户质疑由于在自己硬盘上保存不明数据会带来安全隐患。所以用户可以关闭 Cookie 功能，并且从硬盘上删除 Cookie。这里以 IE 浏览器为例，可以从“工具”菜单中选择“Internet 选项”命令，直接删除 Cookie 文件，如图 12.2 所示；也可以选择“隐私”选项卡，阻止所有 Cookie，如图 12.3 所示。

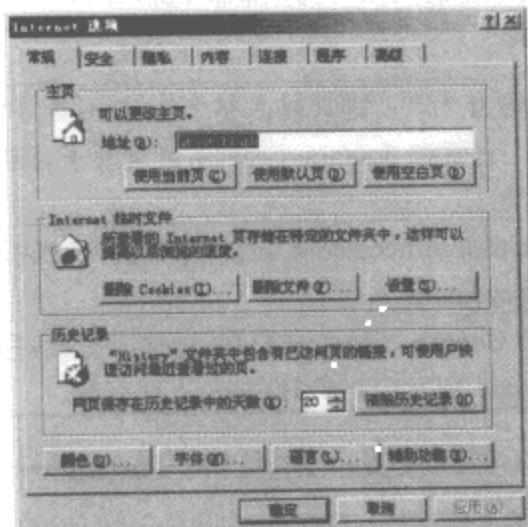


图 12.2 删除 Cookie

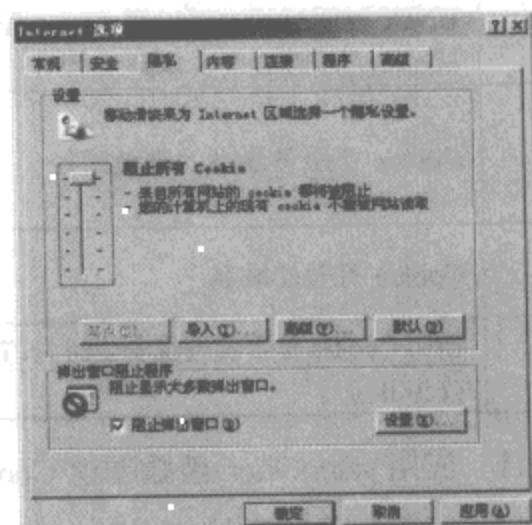


图 12.3 阻止所有 Cookie

### 12.1.2 创建 Cookie

Cookie 是经由服务器端的程序通过 HTTP 请求和响应发送到浏览器的，是 HTTP 头标的组成部分，而头标必须在页面其他内容之前发送。即使是一个 HTML 标记或者 echo 语句，甚至一个空行都会导致程序出错（如图 12.4 所示）。

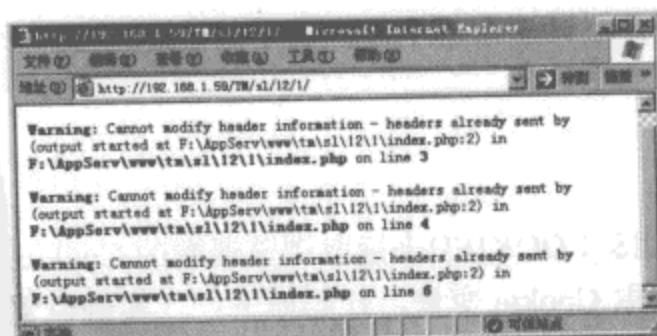


图 12.4 在头标之前发送输出结果引发的错误

在 PHP 中通过 `setcookie()` 函数创建 Cookie，至少接受一个参数，也就是 Cookie 的名称（如果只设置了名称参数，那么在远程客户端上的同名 Cookie 会被删除）。

语法：

```
bool setcookie ( string name [, string value [, int expire [, string path [, string domain [, int secure ]]]]])
```

该函数的参数说明如表 12.1 所示。

表 12.1 `setcookie()` 函数的参数说明

参 数	说 明	举 例
name	Cookie 的变量名	可以通过 <code>\$_COOKIE["cookiename"]</code> 调用变量名为 cookiename 的 Cookie
value	Cookie 变量的值，该值保存在客户端，不能用来保存敏感数据	可以通过 <code>\$_COOKIE["values"]</code> 获取名为 values 的值
expire	Cookie 的失效时间， <code>exkpire</code> 是标准的 UNIX 时间标记，可以用 <code>time()</code> 函数或 <code>mktime()</code> 函数获取，单位为秒	如果不设置 Cookie 的失效时间，那么 Cookie 将永远有效，除非手动将其删除
path	Cookie 在服务器端的有效路径	如果该参数设置为 “/”，则它就在整个 domain 内有效，如果设置为 “/11”，它就在 domain 下的 /11 目录及子目录内有效。默认是当前目录
domain	Cookie 有效的域名	如果要使 Cookie 在 <code>mrbccd.com</code> 域名下的所有子域都有效，应该设置为 “ <code>mrbccd.com</code> ”
secure	指明 Cookie 是否仅通过安全的 HTTPS，值为 0 或 1	如果值为 1，则 Cookie 只能在 HTTPS 连接上有效；如果为默认值 0，则 Cookie 在 HTTP 和 HTTPS 上都有效

例 12.1 应用 `setcookie()` 函数创建 Cookie，代码如下：（实例位置：光盘\TM\Instances\12\1）

```
<?php
setcookie("TMCookie",'PHP 从基础到项目实战');
setcookie("TMCookie", 'PHP 从基础到项目实战', time() + 60); //设置 Cookie 有效时间为 60 秒
//设置有效时间为 60 秒，有效目录为 "/tm/"，有效域名为 "mrbccd.com" 及其所有子域名
setcookie("TMCookie", $value, time() + 3600, "/tm/", ".mrbccd.com", 1);
?>
```

运行本实例，在 Temporary Internet Files 系统临时文件夹下会自动生成一个 Cookie 文件，名为 `administrator@1[1].txt`，Cookie 的有效期为 60 秒，在 Cookie 失效后，Cookie 文件会自动删除。

 注意：HTTP 协议中规定，每个站点向单个用户最多只能发送 20 个 Cookie。

### 12.1.3 读取 Cookie

在 PHP 中通过超级全局数组 `$_COOKIE[]` 来读取浏览器端的 Cookie 值。

例 12.2 应用 `isset()` 函数读取 Cookie 变量，代码如下：（实例位置：光盘\TM\Instances\12\2）

```
<?php
if(!isset($_COOKIE["visit_time"])){
    //检测 Cookie 文件是否存在，如果不存在
```

```

setcookie("visit_time",date("Y-m-d H:i:s"));
//设置一个Cookie变量
echo "欢迎您第一次访问网站!";
//输出字符串

else{
    setcookie("visit_time",date("Y-m-d H:i:s"),time()+60);
    //如果Cookie存在
    //设置带Cookie失效时间的变量
    echo "您上次访问网站的时间为: ".$_COOKIE["visit_time"];
    //输出上次访问网站的时间
    echo "<br>";
    //输出回车符
}

echo "您本次访问网站的时间为: ".date("Y-m-d H:i:s");
//输出当前的访问时间
?>

```

在上面的代码中，首先应用 `isset()` 函数检测 Cookie 是否存在，如果不存在，则应用 `setcookie()` 函数创建一个 Cookie，并输出相应的字符串；如果 Cookie 存在，则应用 `setcookie()` 函数设置 Cookie 失效时间，并输出用户上次访问网站的时间。最后在页面中输出当前时间。

首次运行本实例，由于没有检测到 Cookie 文件，运行结果如图 12.5 所示。如果用户在 Cookie 设置到期时间（本例为 60 秒）前刷新或再次访问该实例，运行结果如图 12.6 所示。

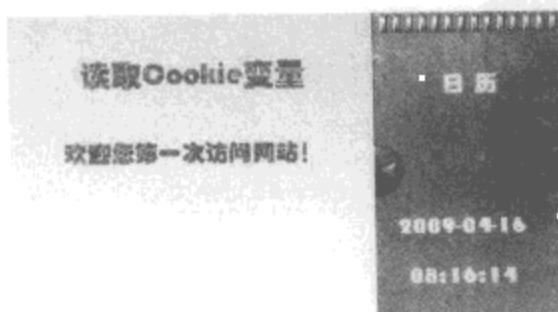


图12.5 第一次访问网页

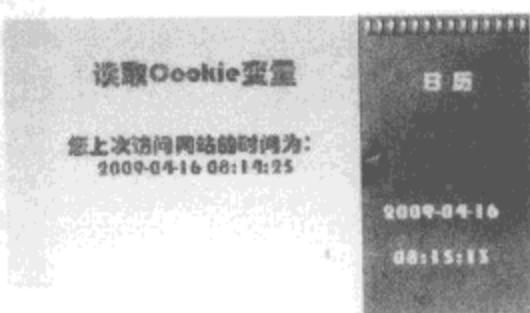


图12.6 刷新或再次访问本网页

**◆ 注意：**如果未设置 Cookie 的到期时间，将在关闭浏览器时自动删除 Cookie 数据。如果为 Cookie 设置了到期时间，浏览器将会记住 Cookie 数据，即使用户重新启动计算机，只要没到期，再访问网站时也会获得如图 12.6 所示的数据信息。

#### 12.1.4 删除Cookie

当 Cookie 被创建后，如果没有设置它的失效时间，其 Cookie 文件会在关闭浏览器时被自动删除。那么如何在关闭浏览器之前删除 Cookie 文件呢？方法有两种：一种是应用 `setcookie()` 函数删除；另一种是应用浏览器手动删除 Cookie。下面分别进行介绍。

##### 1. 应用 `setcookie()` 函数删除Cookie

删除 Cookie 和创建 Cookie 的方式基本类似，删除 Cookie 也使用 `setcookie()` 函数。删除 Cookie 只需要将 `setcookie()` 函数中的第二个参数设置为空值，第三个参数将 Cookie 的过期时间设置为小于系统的当前时间即可。

例如，将 Cookie 的过期时间设置为当前时间减 1 秒，代码如下：

```
setcookie("name","",time()-1);
```

上面的代码中，`time()` 函数返回以秒表示的当前时间戳，把过期时间减 1 秒就会得到过去的时间，从而删除 Cookie。

**► 注意：**把过期时间设置为 0，可以直接删除 Cookie。

## 2. 应用浏览器手动删除 Cookie

在使用 Cookie 时，Cookie 自动生成一个文本文件存储在 IE 浏览器的 Cookies 临时文件夹中。应用浏览器删除 Cookie 文件的具体操作步骤如下：

(1) 选择 IE 浏览器中的“工具” / “Internet 选项”命令，打开“Internet 选项”对话框，如图 12.7 所示。

(2) 在“常规”选项卡中单击“删除 Cookies”按钮，在弹出的对话框中单击“确定”按钮，即可成功删除全部 Cookie 文件。

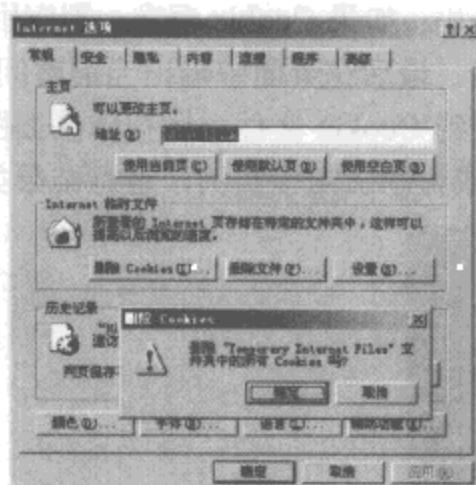


图 12.7 应用浏览器手动删除 Cookie

### 12.1.5 Cookie 应用的注意事项

(1) 在应用 `setcookie()` 函数创建 Cookie 之前，不能有任何 HTML 输出，如果用户犯了这个错误，那么结果可想而知，随之而来的肯定是一堆错误代码。切记，即使是空格、空行也不可以。

(2) 通过 `setcookie()` 函数创建 Cookie 后，在当前页应用 `echo $_COOKIE["name"]` 不会有任何输出。必须是在刷新后或者到达下一个页面时才可以看到 Cookie 值。因为 `setcookie()` 函数执行后，会向客户端发送一个 Cookie，如果不刷新或者浏览下一个页面，客户端就不能将 Cookie 送回。

浏览器创建一个 Cookie 后，对于每一个针对该网站的请求，都会在 Header 中带着这个 Cookie；不过，对于其他网站的请求 Cookie 是绝对不会跟着发送的。而且浏览器会这样一直发送，直到 Cookie 过期为止。

(3) 使用 Cookie 的限制。如果 Cookie 不设定过期时间，那么它的生命周期为浏览器会话的期间，只要关闭 IE 浏览器，Cookie 就会消失。这种 Cookie 被称为会话 Cookie，一般不保存在硬盘上，而是保存在内存中。

如果设置了过期时间，那么浏览器会把 Cookie 保存到硬盘中，再次打开 IE 时依然有效，直到它的有效期超时。

一个浏览器能创建的 Cookie 数量最多为 30 个，并且每个 Cookie 的容量不能超过 4KB，每个 Web 站点能设置的 Cookie 总数不能超过 20 个。

(4) Cookie 是保存在客户端的，如果用户禁止使用 Cookie，那么 Cookie 也就失去了作用。因为

浏览器会拦截 Cookie，并且询问用户是否允许 Cookie 进入。所以在应用 Cookie 实现某些功能时，还是要考虑清楚，搞不好哪个用户会将它当作病毒处理掉，功能也就不能实现了。

### 12.1.6 Cookie 的综合应用——自动登录

#### ■ 视频讲解：光盘\TM\Video\12\自动登录.exe

所谓自动登录，就是用户在第一次成功登录某个网站后，一段时间内，再次登录这个网站时不再需要填写用户名和密码，而是可以直接进入。

**例 12.3 应用 Cookie 实现自动登录的功能，其有效时间是 1 小时。（实例位置：光盘\TM\Instances\12\3）**

具体开发步骤如下：

(1) 创建用户登录页面，通过<script>标记的 src 属性调用 JS 文件，对用户提交数据进行验证；添加 form 表单，提交用户名和密码，关键代码如下。

```
<script language="javascript" src="js/login.js"></script>
<script language="javascript" src="js/xmlhttprequest.js"></script>
<form id="form1" name="form1" method="post" action="">
<div id="bgdiv">
    <div id="txtdiv1"><input id="lgname" name="lgname" type="text" class="tx"/></div>
    <div id="txtdiv2"><input id="lgpwd" name="lgpwd" type="password" class="txt"/></div>
    <div id="txtdiv3"><input id="lgchk" type="text" maxlength="4" style="width:35px;"><img id='chkid' src="" /><a id="changea">看不清</a></div>
    <div id="btndiv">
        <button id="lbtn">&ampnbsp</button>
        <button id="fdbtn" onclick="form.reset();return false;">&ampnbsp</button>
        <input id="chknm" name="chknm" type="hidden" value="" />
    </div>
    <div id="regimg" style=" visibility: hidden;">&ampnbsp</div>
</div>
</form>
```

(2) 编写 login\_chk.php 文件，获取表单中提交的用户名和密码，对其进行判断。如果用户名和密码不正确则给出提示信息；如果用户名和密码正确，则将用户名存储到 Cookie 中，并设置过期时间为 1 小时，最后跳转到 main.php 页面，代码如下。

```
<?php
    include_once 'conn/conn.php'; //执行连接数据库的操作
    $name = addslashes($_GET['name']); //获取用户名
    $pwd = $_GET['pwd']; //获取密码
    if(!empty($name) and !empty($pwd)){
        $sql = "select * from tb_member where name = '".$name."'";
        $active = $conn->getFields($sql,2); //执行查询语句，查询数据库用户名
        $conn->close_rst(); //关闭数据库
        if($active == ""){
            $reback = 4; //判断用户名是否正确
        }else{ //如果不正确则返回该值
    }
```

```

$sql .= " and password = '". $pwd."'";
$num = $conn->getRowsNum($sql);
if($num == 0 or $num == ""){
    $reback = 1;
} else{
    setcookie("name", $name, time() + 3600); //如果正确，则将登录用户名数据存储到 Cookie 中
    $reback = '-1';
}
echo $reback; //返回判断的结果
?>

```

(3) 创建 main.php 文件，登录成功后跳转到主页面。通过\$\_COOKIE[]全局数组获取Cookie中的值，判断如果Cookie的值不为空则输出主页面中的内容，否则给出提示信息，并跳转到登录页面，代码如下。

```

<?php
if($_COOKIE[name]==""){ //根据 Cookie 的值，判断浏览器是否具有访问该页面的权限
    echo "<script>alert('您不具有访问该页面的权限！'); window.location.href='index.php';</script>";
} else{ //如果正确，则输出主页内容
?>
<!--省略了主页面中的内容-->
<?php
}
?>

```

(4) 创建 valcode.php 文件，应用 GD2 函数生成验证码，代码如下。

```

<?php
//header("content-type:image/png"); //生成验证码
$num = $_GET['num']; //获取传递变量值
$imageWidth=60; //定义图像宽度
$imageHeight=18; //定义图像高度
$numImage = imagecreate($imageWidth,$imageHeight); //创建图像
imagecolorallocate($numImage,240,240,240); //定义图像背景颜色
for($i=0;$i<strlen($num);$i++){
    $x = mt_rand(1,8)+$imageWidth*$i/4;
    $y = mt_rand(1,$imageHeight/4);
    $color=imagecolorallocate($numImage,mt_rand(0,150),mt_rand(0,150),mt_rand(0,150));
    imagestring($numImage,5,$x,$y,$num[$i],$color); //随机输出验证码
}
for($i=0;$i<200;$i++){
    $randColor=imagecolorallocate($numImage,rand(200,255),rand(200,255),rand(200,255));
    imagesetpixel($numImage,rand()%70,rand()%20,$randColor); //设置背景效果
}
imagepng($numImage); //生成 png 格式图像
imagedestroy($numImage); //销毁图像
?>

```

(5) 在根目录的 js 文件夹下创建 login.js 文件，在该文件中通过自定义函数对表单中提交的用户名和密码进行验证；通过自定义函数调用 valcode.php 文件生成验证码。

(6) 在根目录的 conn 文件夹下创建 conn.php 文件，实现连接数据库的操作。

运行本实例，如果是第一次登录，则需要填写用户名、密码和验证码，如图 12.8 所示。如果在 Cookie 有效时间内不是第一次登录，则可以在 IE 中输入“<http://127.0.0.1/TM/sl/12/3/main.php>”，直接进入到网站主页，而无须登录，运行结果如图 12.9 所示。



图 12.8 用户登录页面



图 12.9 进入网站主页

## 12.2 Session 管理

会话是指在用户访问 Web 站点上的时期内存储的用户操作的一些动作。PHP 提供了管理会话的机制，可以追踪到用户在这一时期的动作，可以显示出用户喜欢什么、需要什么等。与 Cookie 类似，会话的理念也在于保存状态。不过与 Cookie 相比，会话似乎显得更加强大一些，不但能够管理大量数据，而且可以将信息保存在服务器端，相对更安全，并且也没有存储长度的限制。

### 12.2.1 了解 Session

Session 从用户访问页面开始，到断开与网站连接为止，形成 Sessions 的生命周期。用户每一次连接时，PHP 都会自动生成一个唯一的 Session ID 以标识当前用户，与其他用户进行区分。

在会话期间，PHP 默认生成名为 PHPSESSID 的标识（该名称可以在 php.ini 文件中进行更改），它伴随每一个页面被送往浏览器，然后随下一次页面请求返回给 Web 服务器。

Session ID 可以作为会话信息保存到数据库中，作为主键（Primary Key）来区分不同的用户，或作为服务器端的会话文件名称中的唯一字符串，这样可以跟踪每个用户的登录次数、在线与否，或计算用户的在线时间。

Session 会话时，Session ID 会分别保存在客户端和服务器端两个位置。在客户端，使用临时的 Cookie

保存在浏览器指定目录中（称为 Session Cookie）；在服务器端，以文本文件形式保存在指定的 Session 目录中。

Session 在 Web 技术中占有非常重要的地位。由于网页是一种无状态的连接程序，因此无法记录用户的浏览状态。因此必须通过 Session 记录用户的有关信息，以供用户再次以此身份对 Web 服务器提供要求时作确认。例如，在电子商务网站中，通过 Session 记录用户登录的信息，以及用户所购买的商品，如果没有 Session，用户就会每进入一个页面都需要输入用户名和密码。

## 12.2.2 启动 Session

视频讲解：光盘\TM\Video\12\启动 Session.exe

启动 PHP 会话的方式有两种：一种是应用 session\_start() 函数；另一种是应用 session\_register() 函数为会话登录一个变量来隐含地启动会话。

### 1. 通过 session\_start() 函数创建会话

session\_start() 函数用于创建一个会话。

语法：

```
bool session_start(void);
```

说明：使用 session\_start() 之前浏览器不能有任何输出，否则会发生类似于如图 12.10 所示的错误。

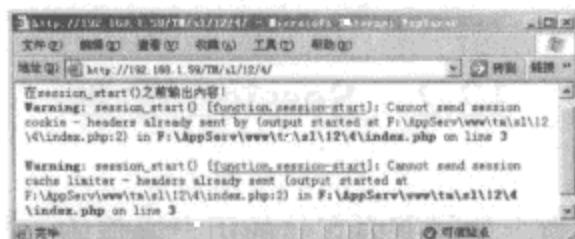


图 12.10 在使用 session\_start() 前输出字符串产生的错误

### 2. 通过 session\_register() 函数创建会话

session\_register() 函数用来为会话登录一个变量来隐含地启动会话，但要求设置 php.ini 文件的选项，将 register\_globals 指令设置为“on”，然后重新启动 Apache 服务器。

注意：使用 session\_register() 函数时，不需要调用 session\_start() 函数，PHP 会在注册变量之后隐含地调用 session\_start() 函数。

会话变量启动后，全部被保存在全局数组\$\_SESSION 中。通过全局数组\$\_SESSION 创建一个会话变量很容易，只需直接给该数组添加一个元素即可。

例如，启动会话，创建一个 Session 变量并赋予空值，代码如下：

```
<?php
    session_start();
    $_SESSION["name"] = null;
?>
```

### 12.2.3 应用 Session

PHP 中的 Session 有一个非常强大的功能：可以保存当前用户的特定数据和相关信息。可以保存的数据类型包括字符串、数组和对象等。将各种类型的数据添加到 Session 中，必须应用全局数组\$\_SESSION[]。

例如，将一个字符串值存储到 Session 中。首先判断会话变量是否有一个会话 ID 存在，如果不存在，则通过全局数组\$\_SESSION 创建一个会话变量，然后将字符串值赋给这个会话变量，最后通过全局数组\$\_SESSION 输出字符串的值，代码如下。

```
<?php
    session_start(); //初始化 Session 变量
    $string="PHP 从基础到项目实战";
    if (empty($_SESSION[name])){
        $_SESSION[name]=$string; //定义字符串
        echo $_SESSION[name]; //判断 Session 会话变量是否为空
        //将字符串值赋给会话变量
        //输出会话变量
    }else{
        echo $_SESSION[name];
    }
?>
```

下面应用全局数组\$\_SESSION[]将数组中的数据保存到 Session 中，并且输出 Session 中保存的数据。

**例 12.4** 首先初始化一个 Session 变量，然后创建一个数组，并通过全局数组\$\_SESSION[]将数组中的数据保存到 Session 中，最后遍历 Session 数组中的数据，代码如下：(实例位置：光盘\TM\Instances\12\4)

```
<?php
    session_start(); //初始化 Session 变量
    $array=array('PHP 从入门到精通','PHP 网络编程自学手册','PHP 函数参考大全','PHP 开发典型模块大全','PHP 网络编程标准教程','PHP 程序开发范例宝典');
    $_SESSION[mr_book]=$array; //将数组中的数据写入到 Session 中
?>
<?php
    foreach($_SESSION[mr_book] as $key=>$value){ //读取 Session 数组中存储的数据
        if($value=="PHP 开发典型模块大全"){ //当$value 的值等于 PHP 开发典型模块大全时换行
            $br=<br><br>;
        }else{
            $br=&nbsp;&nbsp;;
        }
        echo $value.$br; //输出 Session 数组中的内容
    }
?>
```

运行结果如图 12.11 所示。

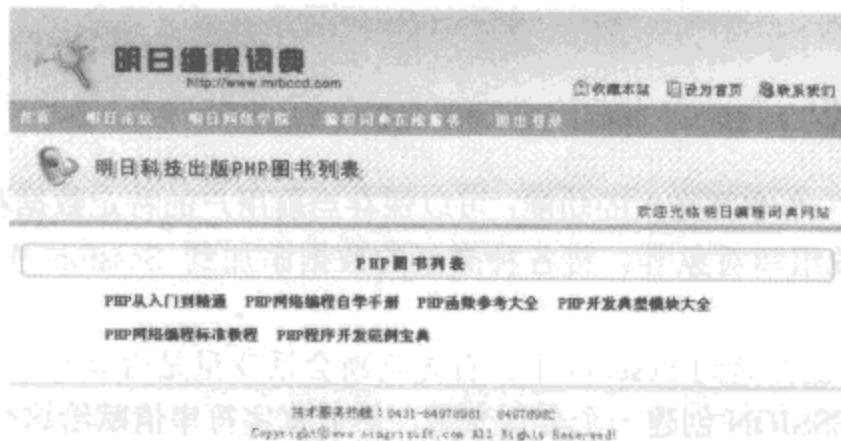


图 12.11 将数组中的数据保存到 Session 中

## 12.2.4 删除 Session

删除会话的方法主要有删除单个会话、删除多个会话和结束当前会话 3 种。下面分别进行介绍。

### 1. 删除单个会话

删除会话变量，同数组的操作一样，直接注销\$\_SESSION 数组的某个元素即可。

例如，注销\$\_SESSION['name'] 变量，可以使用 unset() 函数，代码如下：

```
unset($_SESSION['name']);
```

**► 注意：**应用 unset() 函数时，要注意\$\_SESSION 数组中的某元素不能省略，即不可以一次注销整个数组，这样会禁止整个会话的功能，如使用 unset(\$\_SESSION) 函数会将全局变量\$\_SESSION 销毁，而且没有办法将其恢复，用户也不能再注册\$\_SESSION 变量。如果读者要删除多个或全部会话，可采用下面的两种方法。

### 2. 删除多个会话

如果要一次注销所有的会话变量，可以将一个空的数组赋值给\$\_SESSION，代码如下：

```
$_SESSION = array();
```

### 3. 结束当前会话

如果整个会话已经结束，首先应该注销所有的会话变量，然后使用 session\_destroy() 函数清除结束当前的会话，并清空会话中的所有资源，彻底销毁 Session，代码如下：

```
session_destroy();
```

## 12.2.5 Session 的综合应用——通过 Session 控制页面访问权限

### ► 视频讲解：光盘\TM\Video\12\通过 Session 判断用户的操作权限.exe

在互联网中，Web 网站中的有些页面必须是用户登录以后才能够访问的，普通浏览者不具备访问特定页面的权限。而在 HTTP Web 协议的无状态中判断浏览者是否是登录用户，最有效的方法就是使

用 Cookie 和会话。本实例中应用会话来控制页面的访问权限，将登录用户的用户名保存到 Session 中，用于判断当前用户是否是登录用户，从而避免了在每个页面中都需要用户登录的麻烦。

#### 例 12.5 通过 Session 控制页面的访问权限。（实例位置：光盘\TM\Instances\12\5）

具体操作步骤如下：

(1) 创建用户登录页面 index.php，添加一个表单 form1，应用 POST 方法进行传参，action 指向的数据处理页为 index\_ok.php，添加文本框属性，提交用户名（name）和密码（pwd）。

(2) 创建 index\_ok.php 文件。首先初始化 Session 变量，连接数据库。然后获取表单中提交的用户名和密码，执行 SQL 查询语句，判断用户名和密码是否正确。如果用户名和密码不正确则给出提示信息并返回到登录页面；如果正确则跳转到特定页面 default.php，并且将登录用户名写入到 Session 中，代码如下。

```
<?php
    session_start();
    include_once 'conn/conn.php';
    $name = addslashes($_POST['name']);
    $pwd = $_POST['pwd'];
    if(!empty($name) and !empty($pwd)){
        $sql = "select * from tb_member where name = '".$name."' and password = '".$pwd."'";
        $num = $conn->getRowsNum($sql);
        $conn->close_rst();
        if($num == 0 or $num == ""){
            echo "<script>alert('用户名和密码不正确！'); window.location.href='index.php';</script>";
        }else{
            $_SESSION[name]=$name;
            echo "<script>alert('登录成功！'); window.location.href='default.php';</script>";
        }
    }
?>
```

(3) 创建特定页面 default.php，输出只有登录用户才可以访问的内容。在页面中，首先初始化 Session 变量，然后判断 Session 变量\$\_SESSION[name]的值是否为空，如果该值为空，则说明当前用户不是登录用户，弹出提示信息并跳转到登录页面；如果该值不为空，则可以输出当前页面中的内容。关键代码如下。

```
<?php
    session_start();
    if($_SESSION[name]== ""){
        echo "<script language='javascript'>alert('请您正确登录本页面！');window.location.href='index.php';</script>";
    }
?>
<!--省略了当前页面中的内容-->
```

(4) 创建 safe.php 文件，注销 Session 变量，代码如下。

```
<?php
    session_start(); //初始化 Session
```

```

unset($_SESSION['name']);           //注销 Session 变量 name
session_destroy();                 //销毁 Session 变量
header("location:index.php");      //跳转到 index.php 页面
?>

```

(5) 在根目录下的 conn 文件夹下创建 conn.php 文件, 实现连接数据库的操作, 代码可参考光盘中的内容。

运行本实例, 在博客用户登录页面输入用户名和密码, 则可以直接进入到 default.php 页面中, 如图 12.12 所示; 如果不登录直接浏览 default.php 页面, 则弹出如图 12.13 所示的对话框, 单击“确定”按钮后将跳转到登录页面。

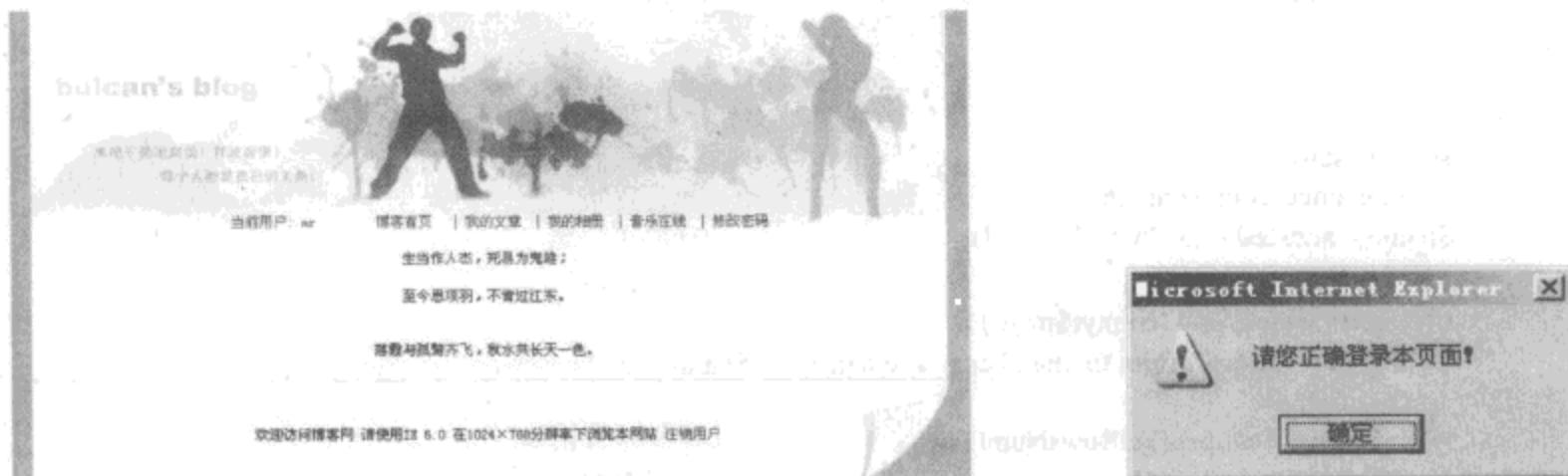


图 12.12 以用户身份登录访问

图 12.13 直接访问页面的结果

### 12.3 Cookie 与 Session 的比较

Session 与 Cookie 最大的区别是: Session 是将信息保存在服务器上, 并通过一个 Session ID 来传递客户端的信息, 服务器在接收到 Session ID 后根据这个 ID 来提供相关的 Session 信息资源; Cookie 是将所有的信息以文本文件的形式保存在客户端, 并由浏览器进行管理和维护。

由于 Session 为服务器存储, 远程用户没办法修改 Session 文件的内容。而 Cookie 为客户端存储。所以 Session 要比 Cookie 安全得多。当然使用 Session 还有很多优点, 如控制容易, 可以按照用户自定义存储(存储于数据库)等。

## 12.4 实战

### 12.4.1 通过 Cookie 实现单点登录

视频讲解: 光盘\TM\Video\12\通过Cookie实现单点登录.exe

现在越来越多的企业、个人喜欢用互联网来进行沟通和交流, 如利用留言板、聊天室、论坛和博客等进行交流和展示自己的方法层出不穷。

上述每种交流方式都具有自己的身份验证机制，这样势必造成：如果某位用户要以会员的身份访问网站，就需要在每个网站上注册，并且通过身份验证后才能以会员的身份访问网站；即使用户以同样的用户名与密码在每个网站上注册时，虽然可以在避免忘记和混淆用户名与密码的方面有一定的作用，但是当用户在某一时间段内访问网站的多个页面或者在不同网站间跳转时，仍然需要登录后才能以会员的身份访问网站。这样不仅给用户带来了不便，而且用户在登录时也浪费了服务器的空间。

如果所有的网站都能够实现单点登录，不仅在用户体验方面有所提高，而且真正体现了集团多个网站间的兄弟性。这种有机结合不但便于对其进行管理，而且能更好地体现公司大平台、大渠道的理念。同时，这也有利于网站的相互促进与相互宣传。

**例 12.6** 开发一个博客网站和一个论坛网站，通过 Cookie 实现单点登录的功能，即无论在博客或者论坛中，只要登录后，这个用户名在一定时间内，在这两个网站中是通用的。以博客网站为例，操作步骤如下：（实例位置：光盘\TM\Instances\12\6）

(1) 创建 index.php 页面，添加用户登录的表单，将登录信息提交到 index\_ok.php 页面中。在 index.php 页面中首先连接数据库，然后判断 Cookie 的值是否为空，如果 Cookie 的值不为空，并且在指定的数据表中存在该记录，那么将直接进入网站的主页，不必进行登录操作；如果 Cookie 的值为空则输出网站登录页面，并且删除指定数据表中已经过期的用户登录信息。代码如下。

```
<?php
    include_once 'conn/conn.php'; //执行连接数据库的操作
    if($_COOKIE[name]!="" and $_COOKIE[id]!=""){
        $sql = "select * from tb_login where name = '".$_COOKIE[name]."' and login_id = '".$_COOKIE[id]."'";
        $num = $conne->getRowsNum($sql); //返回查询结果
        $conne->close_rst(); //关闭数据库
        if($num != 0 or $num != ""){
            header("Location: default.php"); //如果在数据表存在该记录，则直接跳转到登录页面
        }else{ //否则输出登录页面的内容
    }
    >
<!--省略了部分代码-->
<?php
}
}else{
//清除数据库中 1 小时以前存储的用户登录数据（即 Session 过期以后的数据）
$datetimes=mktime()-3600; //获取 1 小时前的时间戳
$sqls="delete from tb_login where datetime<'".$datetimes."'";
$nums = $conne->uidRst($sqls); //返回查询结果
?
<!--省略了部分代码-->
<?php } ?>
```

(2) 创建 index\_ok.php 文件，获取表单中提交的登录信息，完成登录操作。单点登录的第一个关键点：将登录成功的用户名和 Session ID 值保存到 Cookie 中，设置 Cookie 过期时间为 1 小时，有效范围是服务器的根目录。单点登录的第二个关键点：将登录的用户名和 Session ID 值存储到一个数据表中。这个 Cookie 值和数据表中的数据是验证当前用户是否具有单点登录权限的唯一依据，代码如下。

```
<?php
    session_start();
```

```

include_once 'conn/conn.php'; //执行连接数据库的操作
$name = addslashes($_POST['name']); //获取用户名
$pwd = $_POST['pwd']; //获取密码
if(!empty($name) and !empty($pwd)){
    $sql = "select * from tb_member where name = '".$name."' and password = '".$pwd."'";
    $num = $conne->getRowsNum($sql); //返回查询结果
    $conne->close_rst(); //释放查询结果
    if($num == 0 or $num == ""){
        echo "<script>alert('用户名和密码不正确！'); window.location.href='index.php';</script>";
    }else{ //如果正确，则将登录用户名数据存储到 Cookie 中
        $session_id=session_id(); //获取 Session ID
        setcookie("name", $name, time()+3600, "/"); //创建 Cookie
        setcookie("id", md5($session_id), time()+3600, "/"); //创建 Cookie
        $sqls="insert into tb_login(name,login_id,datetime)values('".$name."','".$_SESSION['id']."'".mktime()."')";
        $nums = $conne->uidRst($sqls); //返回查询结果
        echo "<script>alert('登录成功！'); window.location.href='default.php';</script>";
    }
}
?>

```

(3) 创建 default.php 网站主页面。其关键就是根据 Cookie 值和指定数据表中存储的数据判断当前用户是否具有访问主页的权限，如果有则输出网站主页的内容；否则将跳转到登录页面。代码如下。

```

<?php
if($_COOKIE[name]!="" and $_COOKIE[id]!=""){
    include_once 'conn/conn.php'; //执行连接数据库的操作
    $sql = "select * from tb_login where name = '".$_COOKIE[name]."' and login_id = '".$_COOKIE[id]."'";
    $num = $conne->getRowsNum($sql); //返回查询结果
    $conne->close_rst(); //关闭数据库
    if($num == 0 or $num == ""){
        header("Location: index.php");
    }else{
?>
//省略了主页面代码！
<?php
    }
}else{
    header("Location: index.php");
}
?>

```

(4) 在 conn 文件夹下创建 conn.php 文件，实现连接数据库的操作，代码可参考本章光盘。

运行本实例，如果是第一次登录，那么将进入到登录页面，登录成功后将进入到博客主页，如图 12.14 所示。还可以单击博客主页中的论坛超链接，以此用户的身份直接进入到论坛系统主页中，如图 12.15 所示。同样也可以单击论坛系统中的“青春般的小雨滴的博客”超链接跳转到博客的页面中。



图 12.14 进入博客主页面



图 12.15 进入论坛系统主页

## 12.4.2 Session 购物车

### ■ 视频讲解：光盘\TM\Video\12\Session 购物车.exe

购物车是在网上购物时使用的一个临时存储商品的“车辆”。购物车能为我们在网上购物提供很大的方便，不必担心一次购买多个商品时要进行多次提交结算的操作，而可以将其放入购物车中，等选购完所有商品之后，一起进行结算。下面将开发一个 Session 购物车，并对其开发的过程进行详细的讲解。

**例 12.7 应用 Session 技术开发一个购物车，开发步骤如下：**（实例位置：光盘\TM\Instances\12\7）

(1) 创建 index.php 页面，实现商品展示功能。设计原理是：从数据库中读取出商品的信息，将商品的信息在页面中进行分栏、分页显示，并为商品设置购买和查看购物车的超链接。运行结果如图 12.16 所示。

图 12.16 购物车商品展示页面

关键代码如下：

```

<?php require_once("conn/conn.php");?> //连接数据库
<table width="510" height="220" align="center">
    <tr><td height="3" colspan="4" bgcolor="#FFFFFF"></td>
<?php
    $number=0;
    $sql=mysql_query("select count(*) as total from tb_commodity ",$conn);
    $info=mysql_fetch_array($sql);
    $total1=$info[total]; //统计数据库中数据总数
    if(empty($_GET[pages])||is_numeric($_GET[pages])==false){ //判断变量 pages 是否为空
        $page1=1; //如果变量为空，则赋值为 1
    }else{ //如果不为空，则获取变量的值
        $page1=intval($_GET[pages]);
    }
    if($total1==0){ //判断如果数据库中没有数据，则输出“暂无商品”
        echo "<div align=center>暂无商品</div>";
    }else{
        $pagesize1=3; //定义每页显示 10 条记录
        if($total1<$pagesize1){ //如果数据库中数据小于每页显示的记录数
            $pagecount1=1; //则定义 pagecount1 变量的值为 1
        }else{
            if($total1%$pagesize1==0){
                $pagecount1=intval($total1/$pagesize1); //用总的记录数除以每页记录数，获取共有几页
            }else{
                $pagecount1=intval($total1/$pagesize1)+1;
            }
        }
        $query=mysql_query("select * from tb_commodity order by tb_commodity_id desc limit ".($page1-1)*$pagesize1.", $pagesize1",$conn ); //定义查询语句
        while($myrow=mysql_fetch_array($query)){
            if(($number % 3) == 0){
?>
                </tr>
                <tr> <?php } ?>
                    <td height="185" colspan="4" bgcolor="#FFFFFF">
                        <table width="170" height="186" border="0"><tr>
                            <td height="125" align="center" bgcolor="#FFFFFF">
                                &gt;">
                            </td></tr>
                            <!--省略了部分代码-->
                        <tr>
                            <td height="23" align="center" bgcolor="#FFFFFF">
                                <a href="buy_commodity.php?id=<?php echo $myrow[tb_commodity_id];?>">
                                    
                                </a>
                                <a href="shopping_car.php">
                                    
                                </a>
                            </td>
                        </tr>
                    </table>
                </td>
            </tr>
        }
    }
}

```

```

        </a>
    </td>
</tr>
</table>
</td>

<?php
    if(($number %3)!= 0){}
        $number++;
    }
?
</tr>
</table>

```

(2) 创建 by\_commodity.php 文件, 实现购买功能, 即向购物车中添加商品。首先创建一个购物车, 然后判断购物车中是否为空, 如果为空, 则将商品展示页中“购买”超链接传递的商品 ID (`$_GET[id]`) 和数量添加到购物车中; 如果不为空, 则判断添加商品的 ID 是否在购物车中已经存在, 如果存在则不能重复添加, 否则将商品 ID 添加到购物车中, 最后跳转到 shopping.php 购物车页面。代码如下。

```

<?php
    session_start(); //初始化 Session 变量
    session_register("goodsid"); //定义 session 变量用于存储商品 ID
    session_register("goodsnum"); //定义 session 变量用于存储购买商品的数量
    if($_SESSION["goodsid"]=="") && $_SESSION["goodsnum"]==""){ //判断 session 变量是否为空
        $_SESSION["goodsid"]=$_GET["id"]."@"; //如果 session 为空, 则赋值为商品的 ID, 并以@分隔
        $_SESSION["goodsnum"]="1@"; //如果 session 为空, 则为其赋值为 1, 并以@分隔
    }else{
        $array=explode("@",$_SESSION["goodsid"]); //则以@为分隔符, 将 session 中的数据写入到数组中
        if(in_array($_GET["id"],$array)){ //判断数组中是否存在指定的 ID
            echo "<script>alert('该商品已经被放入购物车！');history.back();</script>";
            exit;
        }
        //如果数组中不存在指定的 ID, 则说明该商品还没有放入购物车中
        $_SESSION["goodsid"].=$_GET["id"]."@"; //将该商品添加到购物车中
        $_SESSION["goodsnum"]="1@"; //更改商品数量
    }
    echo "<script>window.location.href='shopping_car.php';</script>";
?>

```

(3) 创建 shopping.php 购物车页面, 对购买商品进行统计, 包括购买数量、每种商品的金额和所有商品的总金额, 并且实现删除购物车中商品、更改购物车中商品数量和清空购物车的功能。同时还为返回商品展示页面设置了超链接, 代码如下。

```

<?php
    session_start();
    require_once("conn/conn.php");
?>
<table width="516" height="47">
<?php

```

```

$array=explode("@", $_SESSION["goodsid"]);
$arraynum=explode("@", $_SESSION["goodsnum"]);
$totalprice=0; //定义价格变量
$arrayinfo=array(); //创建数组
for($i=0;$i<count($array);$i++){
    if($array[$i]!=""){
        $sqlcart=mysql_query("select * from tb_commodity where tb_commodity_id='".$array[$i]."'", $conn);
        $infocart=mysql_fetch_array($sqlcart); //读取数据库中的数据
        array_push($infocart,$arraynum["$i"]); //将购买数量的数据写入到数据库中返回的数组中
        array_push($arrayinfo,$infocart); //将整理的数据压入到创建的新数组中
        $totalprice+=$infocart["tb_commodity_price"]*$arraynum["$i"]; //计算总的价格
    }
}
if(count($arrayinfo)==0){
?>
<tr><td height="22" colspan="5" ><div align="center">对不起，您的购物车中暂无商品信息！</div></td></tr>
<?php
} else{
    for($m=0;$m<count($arrayinfo);$m++){
?>
<tr>
<form name="form<?php echo $arrayinfo[$m][0];?>" method="post" action="change_commodity_counts.php">
    <td height="22" align="center" bgcolor="#FFFFFF"><?php echo $arrayinfo[$m][1]; ?></td>
    <td height="22" bgcolor="#FFFFFF"><div align="center"><?php echo $arrayinfo[$m][2]; ?></div></td>
    <td height="22" bgcolor="#FFFFFF"><div align="center">
        <input type="text" name="goodsnum" value="<?php echo $arrayinfo[$m][7]; ?>" >
        <input type="hidden" name="id" value="<?php echo $arrayinfo[$m][0]; ?>"></div></td>
    <td bgcolor="#FFFFFF"><?php echo $arrayinfo[$m][7]*$arrayinfo[$m][2];?></td>
    <td height="22" bgcolor="#FFFFFF">
        <a href="javascript:form<?php echo $arrayinfo[$m][0];?>.submit();" class="a1">更改数量</a>
        <a href="delete_commodity.php?id=<?php echo $arrayinfo[$m][0];?>" class="a1">删除</a>
    </td>
</form>
</tr>
<?php
    }
?>
</table>
<table width="513" height="22" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td><<
            <a href='index.php' class='a1'>继续购买</a>
            <a href='clear_shopping_car.php' class='a1'>清空购物车</a>&nbsp;>>
        </td>
        <td>&nbsp;商品金额总计： &nbsp;<?php echo $totalprice; ?>&nbsp;元</td>
    </tr>
</table>

```

购物车页面的运行结果如图 12.17 所示。

商品名称	单价(元)	数量(个)	小计	操作
相机	2800	1	2800	<a href="#">更改数量</a>   <a href="#">删除该项</a>
手机	3500	2	7000	<a href="#">更改数量</a>   <a href="#">删除该项</a>
LCD电视	3500	3	10500	<a href="#">更改数量</a>   <a href="#">删除该项</a>

图 12.17 购物车页面

(4) 创建 change\_commodity\_counts.php 文件，更改购物车中商品的数量。首先获取 shopping.php 页面表单中提交的商品数量和商品 ID，然后通过正则表达式验证商品数量是否为正整数，最后根据商品 ID 找到指定的商品，更改商品数量，并返回到购物车页面。代码如下。

```
<?php session_start();
$id=$_POST["id"]; //获取商品ID
$num=$_POST["goodsnum"]; //获取商品数量
$preg="/^[-9]*[0-9]$|[0-9]*[0-9]$/"; //编写正则表达式
if($num==""){ //判断提交的值是否为空
    echo "<script>alert('数量不能为空!');history.back();</script>";
    exit;
} else if(!preg_match($preg,$num,$str)){ //判断提交的数据是否是正整数
    echo "<script>alert('数量只能为正整数!');history.back();</script>";
    exit;
}
$arrayid=explode("@",$_SESSION["goodsid"]);
$arraynum=explode("@",$_SESSION["goodsnum"]);
$key=array_search($id,$arrayid); //在数组中搜索给定的值，如果成功则返回相应的键名
$arraynum[$key]=$num; //更改商品数量
$_SESSION["goodsnum"]=implode("@",$arraynum); //更改商品数量
echo "<script>window.location.href='shopping_car.php';</script>";
?>
```

(5) 创建 delete\_commodity.php 文件，删除购物车中指定的商品。首先获取到超链接中传递的商品 ID，然后将 Session 中的字符串数据转换成数组，并根据商品 ID 获取到指定要删除的商品的键名，最后根据键名为指定的数组赋空值，并重新将数组添加到购物车中。代码如下。

```
<?php
session_start(); //初始化 Session 变量
```

```

$Id=$_GET["id"];
//获取要删除商品的 ID
$arrayid=explode("@",$_SESSION["goodsid"]);
//将购物车中存储的商品 ID 的字符串转换到数组中
$arraynum=explode("@",$_SESSION["goodsnum"]);
//将购物车中存储的商品数量的字符串转换到数组中
$key=array_search($Id,$arrayid);
//获取到数组中指定的数据，并返回键名
$arrayid[$key]="";
//根据返回的键名，将数组中的数据赋值为空
$arraynum[$key]="";
//根据返回的键名，将数组中的数据赋值为空
$_SESSION["goodsid"]=implode("@",$arrayid);
//重新将数组中的数据添加到购物车中
$_SESSION["goodsnum"]=implode("@",$arraynum);
//重新将数组中的数据添加到购物车中
echo "<script>window.location.href='shopping_car.php';</script>";
?>

```

(6) 创建 clear\_shopping\_car.php 文件，清空购物车，即将 SESSION 变量 \$\_SESSION["goodsid"] 和 \$\_SESSION["goodsnum"] 赋值为空。代码如下。

```

<?php
session_start();
//初始化 Session 变量
$_SESSION["goodsid"]=""; //为购物车的 ID 赋值为空
$_SESSION["goodsnum"]=""; //为购物车数量赋值为空
echo "<script>window.location.href='shopping_car.php';</script>"; //完成清空操作，跳转到购物页面
?>

```

## 12.5 本章小结

本章介绍了 Cookie 及 Session 的概念和功能，重点放在对 Cookie 及 Session 的技术讲解上，并且通过具体的实例对它们在实际 Web 开发中的应用进行了剖析。Cookie 与 Session 是进行 Web 开发不可或缺的一部分。希望读者通过本章的学习，能够熟练地掌握这两项技术。

## 12.6 实战练习

1. 使用 Cookie 防止重复投票。（答案位置：光盘\TM\Instances\12\8）
2. 应用 Session 技术实现聊天室换肤功能。运行结果如图 12.18 所示。（答案位置：光盘\TM\Instances\12\9）

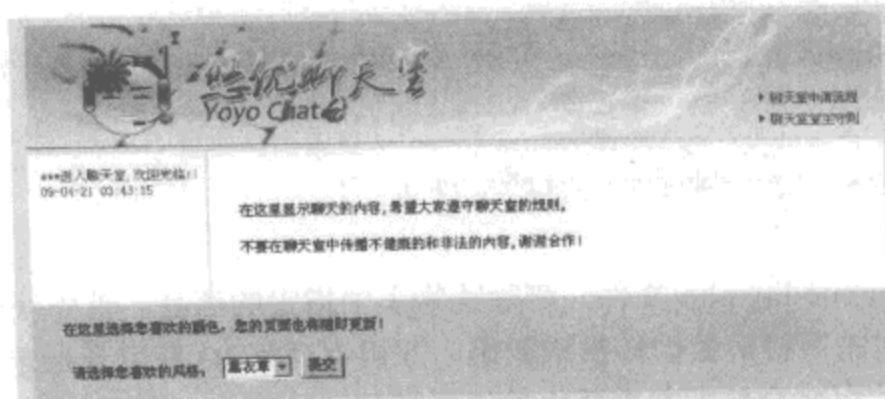


图 12.18 用 Session 技术实现聊天室换肤功能

# 第 13 章

## 图形图像

( 视频讲解: 106 分钟)

**GD2** 函数库是一个源代码公开的用于创建图形图像的函数库，该函数库由 C 语言编写，可以在 Perl、PHP 等多种语言中调试运用。PHP 使用 GD2 函数库可以制作出各类丰富的图形图像效果，如统计图、为图片添加水印以及动态图表等，在企业 OA 管理系统、电子商务网站等大型项目中得以广泛的应用。很多开源项目都对 GD2 函数库提供了很好的技术支持，如 Jpgraph 类库就是基于 GD2 函数库开发的用于制作复杂统计图的类库。

绘制饼形图、柱形图和折线图，这是对数据进行图形化分析的最佳方案，运用 GD2 函数库直接绘制相对灵活，而运用 Jpgraph 绘制则方便易用。本章将分别对 GD2 函数及 Jpgraph 类库进行详细讲解。

通过阅读本章，可以学习到：

- **GD2 库的基本概念**
- **GD2 函数库安装及配置方法**
- **Jpgraph 类库的安装与配置**
- **应用 GD2 函数创建图像**
- **应用 Jpgraph 类库创建各种统计图**



## 13.1 了解 GD2 函数库

PHP 目前在 Web 开发领域已经被广泛地应用，互联网上已经有近半数的站点采用 PHP 作为核心语言。PHP 不仅可以生成 HTML 页面，而且可以创建和操作二进制形式的数据，如图像、文件等。其中使用 PHP 操纵图形可以通过 GD2 函数库来实现，利用 GD2 函数库可以在页面中绘制各种图像、统计图，如果与 Ajax 技术相结合还可以制作出各种强大的动态图表。GD2 库是一个开放的、动态创建图像的源代码公开的函数库，可以从官方网站 <http://www.boutell.com/gd> 下载最新版本的 GD2 库。目前，GD2 库支持 gif、png、jpeg、wbmp 和 xbm 等多种图像格式。

## 13.2 参数设置

### 视频讲解：光盘\TM\Video\13\参数设置.exe

PHP 5 中 GD2 函数库已经作为扩展被默认安装，但目前有些版本中，还需要对 php.ini 文件进行设置来激活 GD2 函数库。用文本编辑工具，如记事本等程序打开 php.ini 文件，将该文件中的 “`;extension=php_gd2.dll`” 选项中的分号 “;” 删除，如图 13.1 所示，保存修改后的文件，并重新启动 Apache 服务器即可激活 GD2 函数库。

在成功加载 GD2 函数库后，可以通过 `phpinfo()` 函数来获取 GD2 函数库的安装信息，验证 GD 库是否安装成功。在 Apache 的默认站点目录中编写 `phpinfo.php` 文件，并在该文件中编写如下代码：

```
<?php
    phpinfo(); //输出 PHP 配置信息
?>
```

在 IE 浏览器的地址栏中输入 “<http://127.0.0.1/phpinfo.php>”，按 Enter 键后，如果在打开的页面中检索到如图 13.2 所示的 GD 库的安装信息，说明 GD 库安装成功，这样开发人员就可以在程序中使用 GD2 函数库编写图形图像。

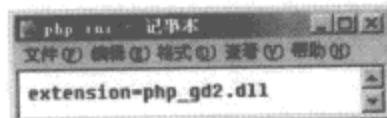


图 13.1 激活 GD2 函数库

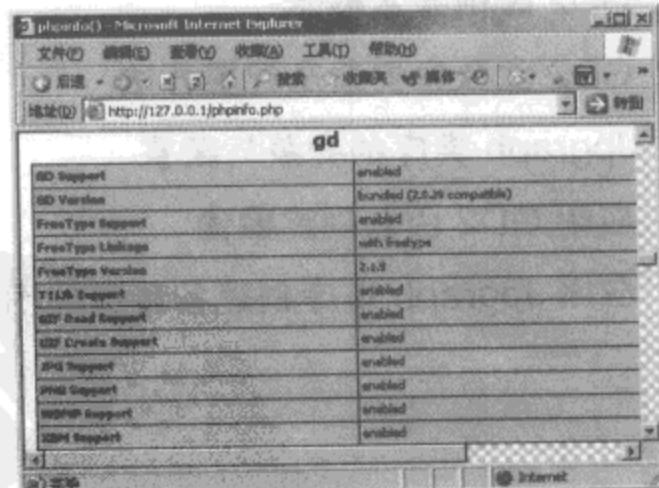


图 13.2 GD2 函数库的安装信息

### 13.3 Jpgraph类库

Jpgraph是基于GD2函数库编写的主要用于创建统计图的类库。它在绘制统计图方面不仅功能非常强大，而且代码编写方便，只需简单的几行代码就可以绘制出非常复杂的统计图效果，从而在很大程度上提高了编程人员的开发效率。

#### 13.3.1 Jpgraph类库简介

Jpgraph类库是一个可以应用在PHP4.3.1以上版本的用于图形图像绘制的类库，该类库完全基于GD2函数库编写。Jpgraph类库提供了多种方法用于创建各类统计图，包括坐标图、柱状图、饼形图等。使用Jpgraph类库使复杂的统计图编写工作变得简单，大大提高了开发者的开发效率，在现今的PHP项目中得以广泛应用。

#### 13.3.2 Jpgraph的安装

##### ■ 视频讲解：光盘\TM\Video\13\Jpgraph的安装.exe

安装Jpgraph前，首先需要下载该类库的压缩包。Jpgraph类库的压缩包主要有两种形式：zip格式和tar格式，如果是Linux/UNIX平台可以选择tar格式的压缩包，如果是微软的win32平台可选择上述两种格式的压缩包中的任一种都可以。Jpgraph类库可以从其官方网站<http://www.aditus.nu/jpgraph/>下载，目前最新的版本是2.3。

如果已经下载了Jpgraph的安装包，可以按照如下步骤进行安装：

- (1) 将压缩包下的全部文件解压到一个文件夹中，如“F:\AppServ\www\jpgraph”。
- (2) 打开PHP的安装目录，编辑php.ini文件并修改其中的include\_path参数，在其后增加文件夹名，如include\_path = ".;F:\AppServ\www\jpgraph"。
- (3) 重新启动Apache服务器即可生效。

**◆ 注意：**Jpgraph需要GD库的支持。如果用户希望Jpgraph类库仅对当前站点有效，只需将Jpgraph压缩包下的src文件夹中的全部文件复制到网站所在的目录下，使用时调用src文件夹下的指定文件即可，这些内容在后面的典型实例中将具体讲解。

#### 13.3.3 Jpgraph的配置

##### ■ 视频讲解：光盘\TM\Video\13\Jpgraph的配置.exe

Jpgraph提供了一个专门用于配置Jpgraph类库的文件jpg-config.inc.php。在使用Jpgraph前，可以通过修改该配置文件实现Jpgraph的配置。

配置jpg-config.inc.php文件包含以下两项：

支持中文的配置

Jpgraph 支持的中文标准字体可以通过修改 CHINESE\_TTF\_FONT 的设置来实现，如下所示。

```
DEFINE('CHINESE_TTF_FONT','bkai00mp.ttf');
```

默认图片格式的配置

根据当前 PHP 环境中支持的图片格式来设定默认的生成图片的格式。Jpgraph 默认图片格式的配置通过修改 DEFAULT\_GFORMAT 的设置来实现。默认值“auto”表示 Jpgraph 将依次按照 PNG、GIF 和 JPEG 的顺序来检索系统支持的图片格式，如下所示。

```
DEFINE("DEFAULT_GFORMAT","auto");
```

 注意：如果用户使用的是 Jpgraph 2.3 版本，则不需要重新进行配置。

## 13.4 常用图像处理

在开发 Web 项目时，图片是前台页面不可缺少的组成部分，传统方式在页面中加入图片一般通过 HTML 语言中的标记实现；在 PHP 语言中为了提升页面传输速率、便于对数据统计，有时采用 GD2 函数库或 Jpgraph 类库由程序员手工绘制。

### 13.4.1 创建画布

 视频讲解：光盘\TM\Video\13\创建画布.exe

GD2 函数库在图形图像绘制方面功能非常强大，开发人员既可以在已有图片的基础上进行绘制，也可以在没有任何素材的基础上绘制，在这种情况下首先要创建画布，之后所有操作都将依据所创建的画布进行。在 GD2 函数库中创建画布通过 imagecreate() 函数实现。

语法：

```
resource imagecreate ( int x_size, int y_size )
```

该函数用于返回一个图像标识符，参数 x\_size、y\_size 为图像的尺寸，单位为像素（pixel）。

例 13.1 应用 imagecreate() 函数创建一个宽 200 像素，高 100 像素的画布，并为画布设置背景颜色 RGB 值为（211 126 29），最后输出一个 gif 格式的图像，代码如下：（实例位置：光盘\TM\Instances\13\1）

```
<?php
$im = imagecreate(200,100); //创建一个 200×100 像素的画布
$white = imagecolorallocate($im, 211, 126, 29); //设置画布的背景颜色为粉色
imagegif($im); //输出图像
?>
```

在上面的代码中，主要应用 imagecreate() 函数来创建一个基于普通调色板的画布，通常支持 256 色，其中该函数中的参数 200 和 100 分别为图像的宽度和高度，单位为像素（pixel）。运行结果如图 13.3 所示。

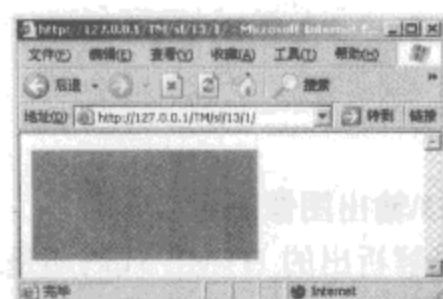


图 13.3 创建画布

### 13.4.2 颜色处理

#### 视频讲解：光盘\TM\Video\13\颜色处理.exe

应用 GD2 函数绘制图形需要为图形中的背景、边框和文字等元素指定颜色，GD2 中使用 `imagecolorallocate()` 函数设置颜色，该函数返回一个标识符，代表了由给定的 RGB 成分组成颜色。

语法：

```
int imagecolorallocate ( resource image, int red, int green, int blue)
```

`image` 参数是 `imagecolorallocate()` 函数的返回值。`red`、`green` 和 `blue` 分别是所需颜色的红、绿、蓝成分，这些参数是 0~255 的整数或者十六进制的 0x00~0xFF。

注意：第一次对 `imagecolorallocate()` 函数的调用会填充背景色。

**例 13.2** 首先应用 RGB 值为 (12 135 20) 的色值作为背景色创建一幅画布，然后应用色值为 (255 255 255) 的色值在已经创建好的画布上画一个正圆，代码如下：（实例位置：光盘\TM\Instances\13\2）

```
<?php
header("Content-type: image/png");
$img = imagecreate(200, 200); //将图像输出到浏览器
$bg = imagecolorallocate($img, 12,135, 20); //创建一个 200×200 像素的图像
$white = imagecolorallocate($img, 255, 255, 255); //设置图像背景色
imagearc($img, 100, 100, 150, 150, 0, 360, $white); //设置绘制图像的颜色为白色
imagepng($img); //绘制一个白色的圆
imagedestroy($img); //输出图像
//释放内存
?>
```

运行本实例将在页面中出现如图 13.4 所示的图形。在上述代码中，`header()` 函数用于向浏览器发送 HTTP 头信息，“Content-type: image/png” 表示通知浏览器输出的内容为 png 格式的图片，最后应用 `imagedestroy()` 函数释放资源。



图 13.4 颜色处理

### 13.4.3 输出图像

#### ■ 视频讲解：光盘\TM\Video\13\输出图像.exe

PHP 作为一种 Web 语言，无论是解析出的 HTML 代码还是二进制的图片最终都要通过浏览器显示。应用 GD2 函数绘制图像首先需要用 `header()` 函数发送 HTTP 头信息给浏览器，告知所要输出图像的类型，然后应用 `imagegif()` 函数、`imagejpeg()` 函数或者 `imagepng()` 函数等分别输出 gif、jpg 和 png 格式的图片。下面将介绍这些函数的使用方法。

语法：

```
void header( string string )
```

参数 `string` 为要指定的 HTTP 头信息。

语法：

```
bool imagegif( resource image [, string filename] )
```

`imagegif()` 函数从 `image` 图像以 `filename` 为文件名创建一个 gif 图像。`image` 参数是 `imagecreate()` 或 `imagecreatefromgif()` 等创建图像函数的返回值，图像格式为 gif。如果应用 `imagecolortransparent()` 函数，则图像设置为透明，格式即为 gif。`filename` 参数为可选，如果省略，则原始图像流将被直接输出。

`imagejpeg()` 和 `imagepng()` 函数的使用方法与 `imagegif()` 函数的使用方法类似，这里不再赘述。

**例 13.3** 首先应用 `header()` 函数发送 HTTP 头信息给浏览器告知输出的媒体类型，在这里输出 gif 格式的图片，然后创建一幅 192×36 像素的画布，并在画布上绘制 “I like PHP” 的文字，最后应用函数 `imagegif()` 输出所创建的图像。（实例位置：光盘\TM\Instances\13\3）

```
<?
    header("Content-type: image/gif");
    $image = imagecreatetruecolor(192,36);           //新建图像
    $trans_color = imagecolorallocate($image, 255, 255, 0);
                                                //设置图像的颜色
    $color = imagecolorallocate($image, 255, 0, 255);
                                                //设置图像的颜色
    imagettftext($image, 18, 0, 50, 26, $color, "impact.ttf", "I like PHP"); //输出文字
    imagegif($image);                            //输出图像
    imagedestroy($image);                      //释放内存空间
?>
```

运行上述代码，将在浏览器中输出如图 13.5 所示的图像，在图像上单击鼠标右键查看图片属性即显示为 gif 格式。上述代码中的 `imagettftext()` 函数用于在画布中输出文字。

图 13.5 输出图像

### 13.4.4 绘制基本图形

#### ▶ 视频讲解：光盘\TM\Video\13\绘制基本图形.exe

应用 GD2 函数可以绘制的图形有多种，最基本的图形包括线条、圆、方形等，无论开发人员绘制多么复杂的图形都是在这些最基本的图形基础上进行深化的，只有掌握了最基本图形的绘制方法，才能够绘制出各种具有独特风格的图形。

在 GD2 函数库中可以分别应用 `imageline()` 函数、`imagearc()` 函数和 `imagerectangle()` 函数绘制直线、圆形和方形。下面将详细介绍这些函数的使用方法。

语法：

```
bool imageline ( resource image, int x1, int y1, int x2, int y2, int color )
```

`imageline()` 函数用 `color` 颜色在图像 `image` 中从坐标  $(x_1, y_1)$  到  $(x_2, y_2)$ （图像左上角为  $(0, 0)$ ）绘制一条线段。

语法：

```
bool imagearc ( resource image, int cx, int cy, int w, int h, int s, int e, int color )
```

`imagearc()` 函数的参数说明如表 13.1 所示。

表 13.1 `imagearc()` 函数的参数说明

参 数	说 明
<code>image</code>	表示图像的 handle
<code>cx, cy</code>	原点坐标 $(0, 0)$ 为图片的左上角，参数 <code>cx</code> 、 <code>cy</code> 为椭圆圆心坐标
<code>w, h</code>	参数 <code>w</code> 、 <code>h</code> 为水平轴长和垂直轴长
<code>s, e</code>	分别为起始角与结束角
<code>color</code>	弧线的颜色

语法：

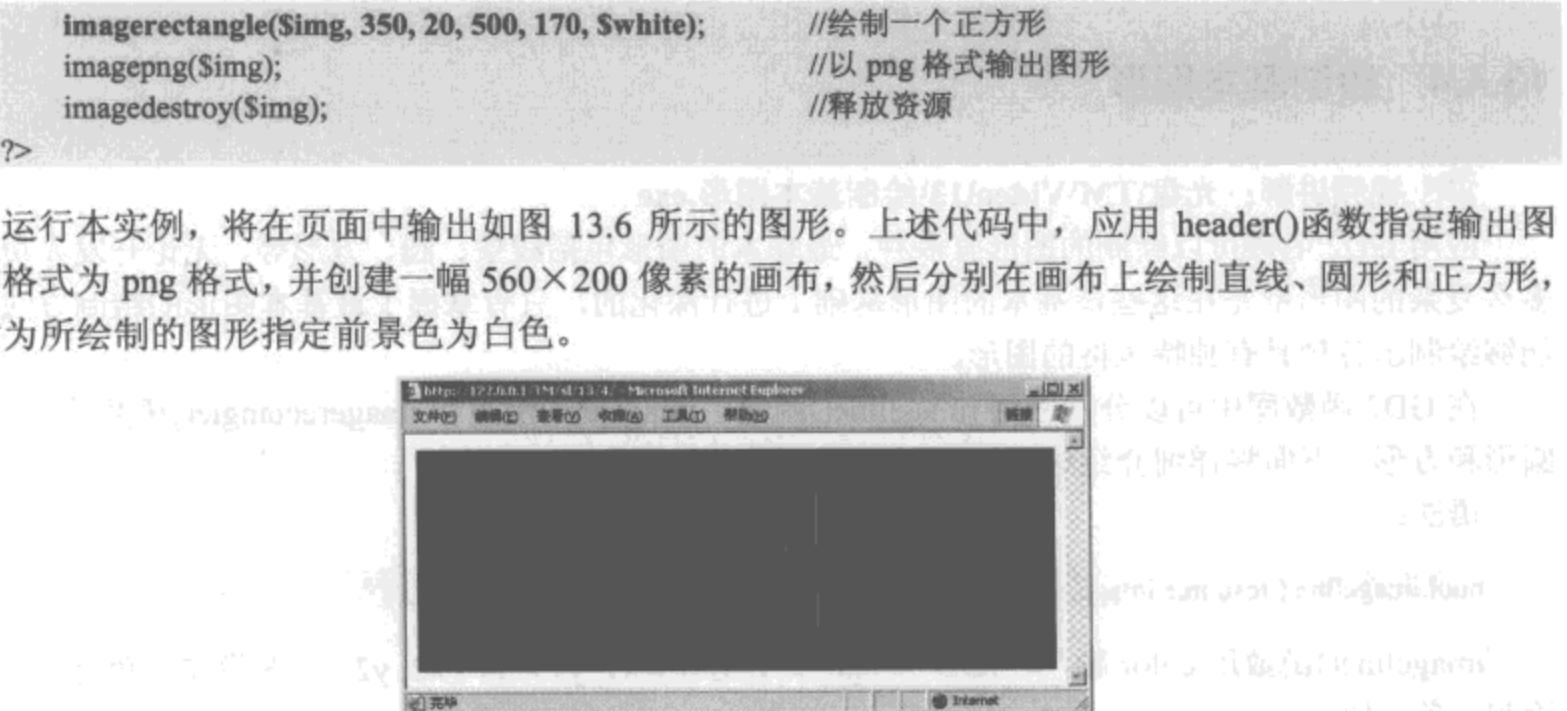
```
bool imagerectangle ( resource image, int x1, int y1, int x2, int y2, int col )
```

利用 `imagerectangle()` 函数以 `col` 颜色在 `image` 图像中绘制一个矩形，其左上角坐标为  $(x_1, y_1)$ ，右下角坐标为  $(x_2, y_2)$ 。图像的左上角坐标为  $(0, 0)$ 。

**例 13.4 应用 GD2 函数最基本的图形绘制功能分别绘制直线、正圆和正方形，并用白色作为线条基色，代码如下：**（实例位置：光盘\TM\Instances\13\4）

```
<?php
    header("Content-type: image/png");
    $img = imagecreate(560, 200); // 将图像输出到浏览器
    $bg = imagecolorallocate($img, 0, 0, 255); // 创建一个 560×200 像素的图像
    $white = imagecolorallocate($img, 255, 255, 255); // 设置图像背景色
    imageline($img, 20, 20, 150, 180, $white); // 设置绘制图像的颜色为白色
    imagearc($img, 250, 100, 150, 150, 0, 360, $white); // 绘制一条直线
    // 绘制一个圆
```

```


    imagerectangle($img, 350, 20, 500, 170, $white); //绘制一个正方形
    imagepng($img); //以 png 格式输出图形
    imagedestroy($img); //释放资源
?>

```

运行本实例，将在页面中输出如图 13.6 所示的图形。上述代码中，应用 header() 函数指定输出图形的格式为 png 格式，并创建一幅 560×200 像素的画布，然后分别在画布上绘制直线、圆形和正方形，同时为所绘制的图形指定前景色为白色。

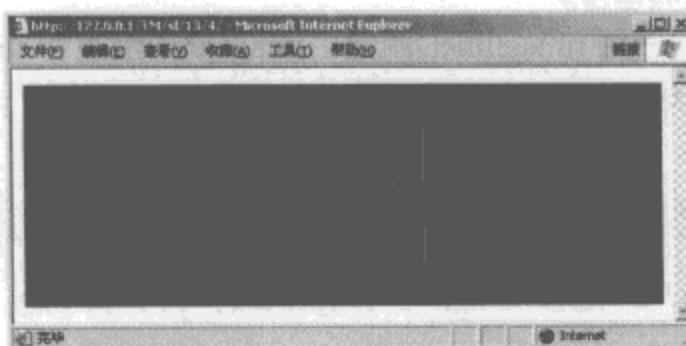


图 13.6 GD2 绘制基本图形

#### 13.4.5 填充几何图形

##### 视频讲解：光盘\TM\Video\13\填充几何图形.exe

使用 GD2 函数不仅可以绘制线条图形，而且可以绘制填充图形，如填充圆形、填充矩形等。下面对 GD2 中常用填充图形的绘制方法进行介绍。

语法：

```
bool imagefill ( resource image, int x, int y, int color)
```

imagefill() 函数在 image 图像的坐标 (x, y)（图像左上角为 (0, 0)）处用 color 颜色执行区域填充（即与 (x, y) 点颜色相同且相邻的点都会被填充）。

语法：

```
bool imagefilledarc ( resource image, int cx, int cy, int w, int h, int s, int e, int color, int style )
```

imagefilledarc() 函数在 image 所代表的图像中以 (cx, cy)（图像左上角为 (0, 0)）为坐标点绘制一椭圆弧。如果成功则返回 true，否则返回 false。w 和 h 参数分别指定了椭圆的宽和高，s 和 e 参数以角度指定了起始和结束点。style 可以是下列值按位取或 (OR) 后的值：

- IMG\_ARC\_PIE。
- IMG\_ARC\_CHORD。
- IMG\_ARC\_NOFILL。
- IMG\_ARC\_EDGED。

IMG\_ARC\_PIE 和 IMG\_ARC\_CHORD 是互斥的。IMG\_ARC\_CHORD 只是用直线连接了起始和结束点，IMG\_ARC\_PIE 则产生圆形边界（如果两个都用，IMG\_ARC\_CHORD 生效）。IMG\_ARC\_NOFILL 指明弧或弦只有轮廓，不填充。IMG\_ARC\_EDGED 指明用直线将起始点和结束点与中心点相连，和

IMG\_ARC\_NOFILL 一起使用是绘制饼状图轮廓的好方法（而不用填充）。  
语法：

```
bool imagefilledellipse ( resource image, int cx, int cy, int w, int h, int color )
```

该函数在 image 所代表的图像中以 (cx, cy) (图像左上角为 (0, 0)) 为中心绘制一个椭圆。w 和 h 参数分别指定了椭圆的宽和高，使用 color 颜色填充。如果成功则返回 true，否则返回 false。

语法：

```
bool imagefilledrectangle ( resource image, int x1, int y1, int x2, int y2, int color )
```

该函数在 image 图像中绘制一个用 color 颜色填充了的矩形，其左上角坐标为 (x1, y1)，右下角坐标为 (x2, y2)。(0, 0) 为图像的最左上角。

**例 13.5 应用 GD2 函数绘制填充圆形和填充正方形，详细实现代码如下：(实例位置：光盘\TM\Instances\13\5)**

```
<?php
header("Content-type: image/png");
//将图像输出到浏览器
$img = imagecreate(400, 200);
//创建一个 400×200 像素的图像
$bg = imagecolorallocate($img, 0, 0, 255);
$white = imagecolorallocate($img, 255, 0, 255);
imagefilledellipse($img, 100, 100, 150, 150, $white); //绘制圆形
imagefilledrectangle($img, 200, 50, 300, 150, $white); //绘制正方形
imagepng($img);
imagedestroy($img);
?>
```

运行上述代码将在页面中输出如图 13.7 所示的图形。上述代码中首先创建一幅 400×200 像素的画布，然后分别在画布上绘制填充圆形和填充正方形。



图 13.7 GD2 绘制填充图形

#### 13.4.6 输出文字

视频讲解：光盘\TM\Video\13\输出文字.exe

PHP 中的 GD 库支持中文，但是必须要以 UTF-8 格式的参数来进行传递，如果直接绘制中文字串就会显示乱码，这是因为 GD2 对中文只能接收 UTF-8 编码格式，并且默认使用了英文的字体，所以只需将显示的中文字串进行转码，并设置绘制中文字符使用的字体，即可绘制中文字符。在 GD2 函数

数据库中应用 `imagestring()` 函数在画布上输出字符串。语法：

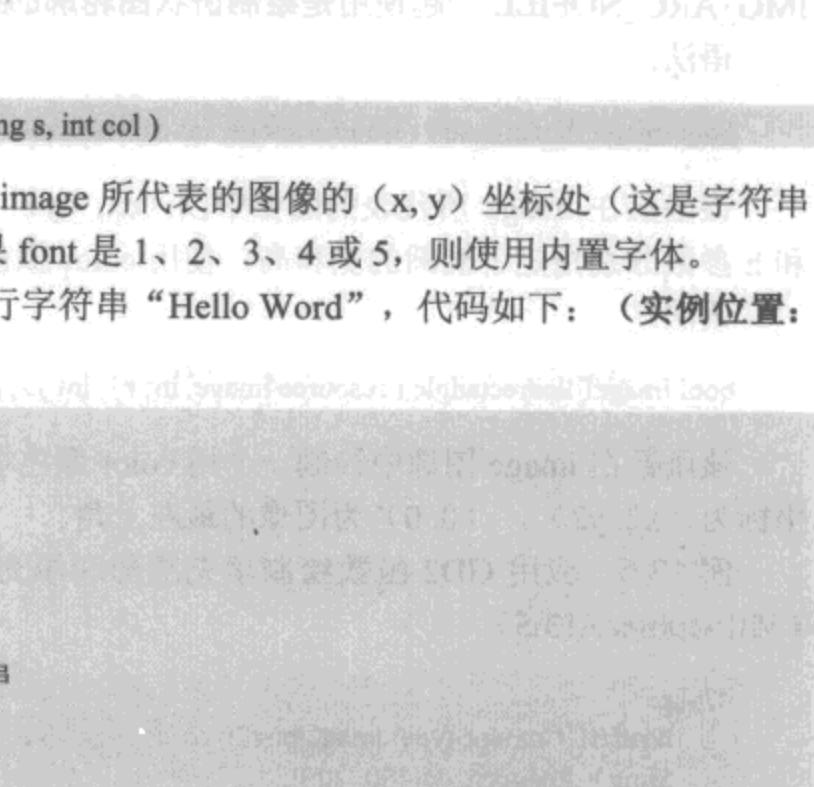
```
bool imagestring( resource image, int font, int x, int y, string s, int col )
```

`imagestring()` 函数用 `col` 颜色将字符串 `s` 绘制到 `image` 所代表的图像的 `(x, y)` 坐标处（这是字符串左上角坐标，整幅图像的左上角为 `(0, 0)`）。如果 `font` 是 1、2、3、4 或 5，则使用内置字体。

**例 13.6 应用 `imagestring()` 函数水平地绘制一行字符串“Hello Word”，代码如下：（实例位置：光盘\TM\Instances\13\6）**

```
<?php
    header("Content-type: image/png");
    $im = imagecreate(100, 50);
    $bg = imagecolorallocate($im, 255, 255, 255);
    $red = imagecolorallocate($im, 255, 0, 0);
    imagestring($im, 5, 0, 0, "Hello Word", $red);
    //在指定的坐标处，水平地绘制一行红色的字符串
    header("Content-type: image/png");
    imagepng($im); //输出图像
?>
```

运行本实例，如图 13.8 所示，在页面中输出文字“Hello Word”的字样，上述代码中首先创建一幅  $100 \times 50$  像素的画布，然后应用 `imagestring()` 函数在页面中绘制文字。



Hello Word

图 13.8 颜色处理

## 13.5 实战

图形是网页中重要的组成元素，网页中如果没有丰富多彩的图形图像就会缺少生气。漂亮的图形图像能让整个网页看起来更富有吸引力，使文字难以表达的思想让人一目了然，并且可以清晰地表达出数据之间的关系。

### 13.5.1 创建一个简单的图像

#### 视频讲解：光盘\TM\Video\13\创建一个简单的图像.exe

应用 GD2 函数绘制图像，首先需要创建画布。创建画布的方法主要有两种，一种是在没有任何图片背景的情况下创建空白画布，另一种是在已有图片的基础上创建画布。下面将通过具体实例讲解如何创建一幅有图片背景的图像。

**例 13.7 应用 GD2 函数创建一幅简单的图像。（实例位置：光盘\TM\Instances\13\7）**

具体实现步骤如下：

- (1) 使用函数 `header()` 发送 HTTP 头信息，告知浏览器所输出媒体的类型，代码如下。

```
header("Content-type: image/png");
```

(2) 以 bg.gif 作为背景创建一幅图片，代码如下。

```
$im = imagecreatefromgif("bg.gif");
```

(3) 使用 imagecolorallocate() 函数指定前景色为红色，代码如下。

```
$red = imagecolorallocate($im, 255, 0, 0);
```

(4) 在图像上绘制文字 “http://www.\*\*.com”，并输出图片释放内存空间，代码如下。

```
imagestring($im, 5, 0, 0, "http://www.**.com", $red); //绘制文字  
imagepng($im); //输出图像  
imagedestroy($im); //释放内存空间
```

运行本实例，如图 13.9 所示，可以在打开的图片左上角看到 “http://www.\*\*.com” 字样。在实际开发中，可以应用本实例为图片添加文字水印。



图 13.9 创建简单的图像

### 13.5.2 应用 GD2 函数在照片上添加文字

**视频讲解：**光盘\TM\Video\13\应用 GD2 函数在照片上添加文字.exe

为保证网站中所上传的图片不被他人盗用，经常需要在所上传的图片中添加水印。在 PHP 中使用 imageTTFTText() 函数即可将文字动态地添加到图片中。

**例 13.8 应用 imageTTFTText() 函数将文字“丹东断桥”以 TTF (TrueType Fonts) 字体输出到图像中。（实例位置：光盘\TM\Instances\13\8）**

程序开发步骤如下：

- (1) 使用 header() 函数发送 HTTP 头指定发送媒体类型。
- (2) 使用 iconv() 函数将“丹东断桥”由 GB2312 格式转换为 UTF-8 格式。
- (3) 载入一张 jpg 格式的背景图片，并用 imageTTFTText() 函数在图像中以 simhei.ttf 字体输出一串蓝色的文字“丹东断桥”，代码如下。

```
<?php  
header("content-type:image/gif"); //定义输出为图像类型  
$im=imagecreatefromjpeg("bg.jpg"); //载入照片  
$textcolor=imagecolorallocate($im,56,73,136); //设置字体颜色为蓝色，值为 RGB 颜色值  
$fnt="c:/windows/fonts/simhei.ttf"; //定义字体  
$text =iconv("gb2312", "utf-8", "丹东断桥"); //将中文转换为 UTF-8 格式  
imageTTFTText($im,80,0,150,120,$textcolor,$fnt,$text); //写 TTF 文字到图中  
imagegif($im); //建立 gif 图形  
imagedestroy($im); //结束图形，释放内存空间  
?>
```

在上面的代码中，主要应用 `imageTTFText()` 函数输出文字到照片中。其中，`$im` 是指照片，80 是字体的大小，0 是文字的水平方向，(150, 120) 是文字的坐标值，`$textcolor` 是文字的颜色，`$fnt` 是字体，`$text` 是照片文字。

运行前后的效果如图 13.10 和图 13.11 所示。



图 13.10 照片原图



图 13.11 添加文字后的照片

### 13.5.3 应用图像处理技术生成验证码

#### 视频讲解：光盘\TM\Video\13\应用图像处理技术生成验证码.exe

用户注册主要就是为了保存用户的一些基本信息，包括用户名、真实姓名、密码等相关信息，以及跟踪和定位不同用户，从而更好地服务于用户。同时为了提高站点的安全性，避免由于网速慢造成用户注册信息的重复提交，往往会在用户注册表单中增加验证码功能。

**例 13.9 应用 GD2 函数绘制图形验证码，并通过 JavaScript 调用所绘制的验证码显示在用户注册表单中。（实例位置：光盘\TM\Instances\13\9）**

程序开发步骤如下：

(1) 编写用户注册信息录入表单，用于录入用户输入的注册信息，其中的表单及表单元素如表 13.2 所示。

表 13.2 用户注册页面所涉及的表单及表单元素

元素类型	元素名称	属性设置	说明
表单	form1	<code>name="form1" method="post" action="success.php"</code> <code>onSubmit="return chkinput(this)"</code>	表单
文本域	username	<code>name="username" size="30" class="input"</code>	用户名
密码域	password	<code>type="password" name="password" size="30" class="input"</code>	密码
密码域	defPassword	<code>name="defPassword" size="30" class="input"</code>	确认密码
文本域	email	<code>name="email" size="30" class="input"</code>	E-mail 地址
文本域	tel	<code>name="tel" size="30" class="input"</code>	联系电话
文本域	address	<code>name="address" size="30" class="input"</code>	联系地址
文本域	xym	<code>type="text" name="xym" size="6" class="input"</code>	输入验证码
隐藏域	xym1	<code>type="hidden" name="xym1"</code>	临时保存验证码的值
提交按钮	submit	<code>name="submit" type="submit" value="登录"</code>	提交按钮
重置按钮	reset	<code>name="reset" type="reset" value="重置"</code>	重置按钮

(2) 对用户在注册表单中录入的个人信息进行合法性检查是必不可少的,这样不仅可以过滤掉一些非法字符,而且可以有效地提高网站的安全性;既能方便和指导用户填写个人信息,而且可以方便站点人员管理。本实例中主要应用 JavaScript 对用户的注册信息在前台进行验证,从而有效降低服务器负载,关键代码如下。

```
<script language="javascript">
    function chkinput(form)
    {
        <!--省略了程序代码 -->
        return true; //通过上边所有验证则允许表单提交
    }
</script>
```

(3) 应用 JavaScript 在 Web 页面动态生成 4 位随机数,当用户单击“看不清”按钮时,不刷新页面而重新生成验证码,并将验证码的值赋给隐藏域。关键代码如下。

```
<script language="javascript">
    var num1=Math.round(Math.random()*10000000);
    var num=num1.toString().substr(0,4); //生成 4 位随机数
    document.write("<img name=codeimg src='ValidatorCode.php?code='"+num+"'>");
    form1.defValidatorCode.value=num; //将随机数赋给隐藏域
    function reCode(){
        var num1=Math.round(Math.random()*10000000);
        var num=num1.toString().substr(0,4);
        document.codeimg.src="ValidatorCode.php?code="+num;
        form1.defValidatorCode.value=num;
    }
</script>
```

(4) 通过\$\_GET[]全局数组接收传递的 4 位参数,然后分别解析该参数各个位置上的值,并用imagestring()函数将解析后的值按顺序绘制在验证码图片上,同时为验证码图片加干扰点。实现该过程的完整代码如下。

```
<?php
header('content-type:image/png'); //指定验证码图片的格式
srand((double) microtime() * 1000000);
$im = imagecreate(60, 20);
imagefill($im, 0, 0, imagecolorallocate($im, 200, 200, 200));
$validatorCode = $_GET['code']; //接收传递参数的值
imagestring($im, rand(2, 5), 10, 3, substr($validatorCode, 0, 1), imagecolorallocate($im, 0, rand(0, 255), rand(0, 255)));
imagestring($im, rand(2, 5), 20, 3, substr($validatorCode, 1, 1), imagecolorallocate($im, rand(0, 255), 0, rand(0, 255)));
imagestring($im, rand(2, 5), 30, 3, substr($validatorCode, 2, 1), imagecolorallocate($im, rand(0, 255), rand(0, 255), 0));
imagestring($im, rand(2, 5), 40, 3, substr($validatorCode, 3, 1), imagecolorallocate($im, 0, rand(0, 255), rand(0, 255)));
for ($i = 0; $i < 200; $i++) { //绘制 200 个干扰点
    imagesetpixel($im, rand() % 70, rand() % 30, imagecolorallocate($im, rand(0, 255), rand(0, 255), rand(0, 255)));
}
imagepng($im);
imagedestroy();
```

**► 注意：**在 PHP 页面中，如果页面代码不仅包含 PHP 代码，还包含其他语言代码，如 JavaScript、CSS 或 HTML 等，需要写完整的 PHP 代码块标记，即“<?php”与“?>”都要写；如果页面代码完全由 PHP 构成，则建议只写开始标记，不写结束标记，这样可以防止恶意用户通过非法注入等手段在页面代码后注入恶意代码。

运行本实例，在用户注册页面中添加用户的基本信息，填写完成后单击“提交”按钮，如果用户录入信息合法并且验证码录入正确，则会显示用户注册成功信息。运行效果如图 13.12 所示。

填写用户注册信息

用户名：

密 码：

确认密码：

E-mail：

电话：

联系地址：

验证码： 9957 看不清

用户在使用\*\*\*网站网络服务过程中，必须遵循以下原则：

- 遵守中国有关的法律和法规。不得利用本网站危害国家安全、泄露国家秘密，不得侵犯国家社会集体的和公民的合法权益。不得利用\*\*\*网站网络服务系统进行任何不利于\*\*\*网站的行为。
- 遵守所有与网络服务有关的网络协议、规定和程序。
- 不干扰或混乱网络服务。
- 不得为任何非法目的而使用本网络服务系统。
- 不得传播任何虚假的、骚扰性的、中伤他人的、辱骂性的、恐吓性的、庸俗淫秽的或其它任何非法的信息资料。

版权所有 © 吉林省\*\*\*科技有限公司  
公司地址：吉林省长春市二道区\*\*\*路29号

图 13.12 带有验证码的用户注册表单

### 13.5.4 Jpgraph 创建柱状图展示年度收支情况

#### ► 视频讲解：光盘\TM\Video\13\Jpgraph 创建柱状图展示年度收支情况.exe

应用 GD2 函数库可以创建各式各样的图形，但是制作复杂的统计图形，仅通过 GD2 函数来实现则会显得非常繁琐。PHP 从诞生走向成熟经过多次变革，在这期间，许多组织和机构都开发了基于 PHP 的框架和类库。Jpgraph 图形类库就是其中非常实用、流行的代表，不仅功能强大，而且操作简便。

例 13.10 应用 Jpgraph 创建柱状图展示年度收支情况。该实例的实现过程如下：（实例位置：光盘\TM\Instances\13\10）

(1) 将 Jpgraph 类库导入到程序中。

使用 Jpgraph 类库，首先应该从 <http://www.aditus.nu/jpgraph/> 网站下载该类库的压缩包，下载完毕后解压该压缩包，将会呈现如图 13.13 所示的目录结构。

如果希望服务器中所有站点均有效，可以按如下步骤进行配置。

首先，解压下载的压缩包，并将解压后的文件夹重命名为 jpgraph，该文件夹子目录的结构如图 13.13 所示。可以只保留 src 目录，并将该目录保存到服务器磁盘中，如 c:\jpgraph。

然后，编辑 php.ini 文件，修改 include\_path 配置项，在该项后增加 Jpgraph 类库的保存目录，如 include\_path = “.;c:\jpgraph”。

最后，重新启动 Apache 服务器，则配置生效。

如果只希望在本站点使用 Jpgraph，则直接解压下载的压缩包到工程目录下即可。

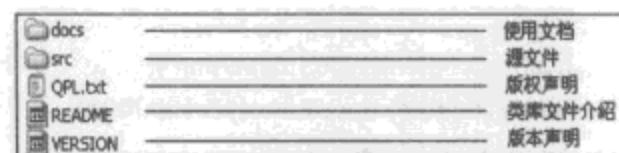


图 13.13 Jpgraph 压缩包解压后的目录

通过以上两种方式之一安装完成 Jpgraph 后，在程序最开始处使用如下语句即可将 Jpgraph 导入到项目中。

```
require_once 'src/jpgraph.php';
```

(2) 建立 index.php 页面，该页面包含实现柱状图的详细代码，如下所示。

```
require_once 'src/jpgraph.php'; //导入 Jpgraph 类库
require_once 'src/jpgraph_bar.php'; //导入 Jpgraph 类库的柱状图功能
$data = array(80, 73, 89, 85, 92, 75, 85, 65, 84, 79, 85, 98); //设置统计数据
$graph = new Graph(600, 300); //设置画布大小
$graph->SetScale('textlin'); //设置坐标刻度类型
$graph->SetShadow(); //设置画布阴影
$graph->img->SetMargin(40, 30, 20, 40); //设置统计图边距
$barplot = new BarPlot($data); //实例化 BarPlot 对象
$barplot->SetFillColor('blue'); //设置柱形图前景色
$barplot->value->Show(); //输出图像
$graph->Add($barplot);
$graph->title->Set('吉林省***科技有限公司年度收支'); //设置统计图标题
$graph->xaxis->title->Set('月份'); //设置 X 轴名称
$graph->yaxis->title->Set('总金额(万元)'); //设置 Y 轴名称
$graph->title->SetFont(FF_SIMSUN, FS_BOLD); //设置标题字体
$graph->xaxis->title->SetFont(FF_SIMSUN, FS_BOLD); //设置 X 轴字体
$graph->yaxis->title->SetFont(FF_SIMSUN, FS_BOLD); //设置 Y 轴字体
$graph->Stroke(); //输出图像
```

运行结果如图 13.14 所示，以柱状图的形式展示了公司全年的收支情况。

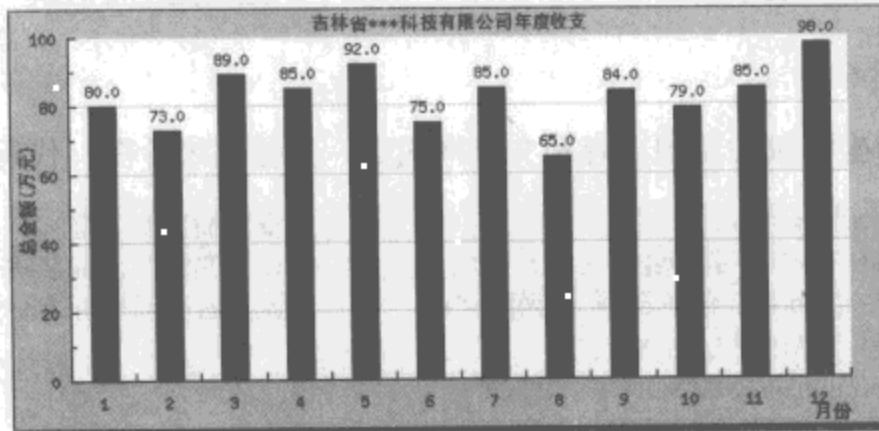


图 13.14 Jpgraph 创建柱状图展示年度收支情况

### 13.5.5 Jpgraph 创建折线图统计图书销售走势

#### ■ 视频讲解：光盘\TM\Video\13\Jpgraph 创建折线图统计图书销售走势.exe

在企业运营中，经常需要对各种数据进行统计，利用图表动态分析不同的数据表中的数据。这可以帮助企业领导者更好地制定决策，而且可以使数据显示更加直观。

例 13.11 应用 Jpgraph 技术绘制包含两种图书销售走势的折线图，这样不仅可以直观地展示出图书的销售走势，而且可以对这两种图书销售情况进行动态比较。为了能够使用 Jpgraph 的功能，首先在

程序中导入 Jpgraph 类库，然后创建两个数组分别表示两种图书的年度销售情况，创建 Graph 类的对象并用创建的数组作为参数创建两个 LinePlot 类的对象。设置统计图的标题、刻度、背景色和折线颜色等参数，并将两个 LinePlot 对象添加到统计图对象中。（实例位置：光盘\TM\Instances\13\11）

该实例的实现过程如下：

(1) 在程序中导入 Jpgraph 类库及直线绘制功能，代码如下。

```
require_once 'src/jpgraph.php'; //导入 Jpgraph 类库
require_once 'src/jpgraph_line.php'; //导入 Jpgraph 类库的柱状图功能
```

(2) 创建统计数据，并将数据保存到一维数组中，代码如下。

```
$data1 = array(89, 78, 99, 65, 92, 85, 85, 55, 64, 79, 85, 98); //设置统计数据
$data2 = array(68, 70, 69, 80, 50, 60, 75, 65, 64, 80, 75, 65); //设置统计数据
```

(3) 创建统计图对象，并设置坐标轴的刻度样式，代码如下。

```
$graph = new Graph(600, 300); //创建统计图对象
$graph->SetScale("textlin"); //设置刻度样式
$graph->setY2Scale("lin");
$graph->SetShadow(); //设置背景带阴影
$graph->img->SetMargin(40, 50, 20, 70); //设置统计图边距范围
$graph->title->Set("图书销售走势表"); //设置统计图标题
```

(4) 创建 LinePlot 对象，并将创建后的对象添加到统计图对象中，代码如下。

```
$lineplot1 = new LinePlot($data1); //建立 LinePlot 对象
$lineplot2 = new LinePlot($data2); //建立 LinePlot 对象
$graph->Add($lineplot1);
$graph->AddY2($lineplot2);
```

(5) 设置统计图和坐标轴的标题内容和文字样式等，并输出统计图，代码如下。

```
$graph->xaxis->title->Set("月份"); //设置横坐标名称
$graph->yaxis->title->Set("book A 销售金额（万元）"); //设置左侧纵坐标名称
$graph->y2axis->title->Set("book B 销售金额（万元）"); //设置右侧纵坐标名称
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);
$graph->yaxis->title->SetFont(FF_SIMSUN, FS_BOLD);
$graph->y2axis->title->SetFont(FF_SIMSUN, FS_BOLD);
$graph->xaxis->title->SetFont(FF_SIMSUN, FS_BOLD); //设置字体
$lineplot1->SetColor('red');
$lineplot2->SetColor('blue');
$lineplot1->SetLegend('book A');
$lineplot2->SetLegend('book B');
$graph->legend->SetLayout(LEGEND_HOR);
$graph->legend->Pos(0.4, 0.95, 'center', 'bottom');
$graph->Stroke(); //输出统计图
```

运行结果如图 13.15 所示，图中以折线的形式显示出图书 A 和图书 B 的年度销售走势。

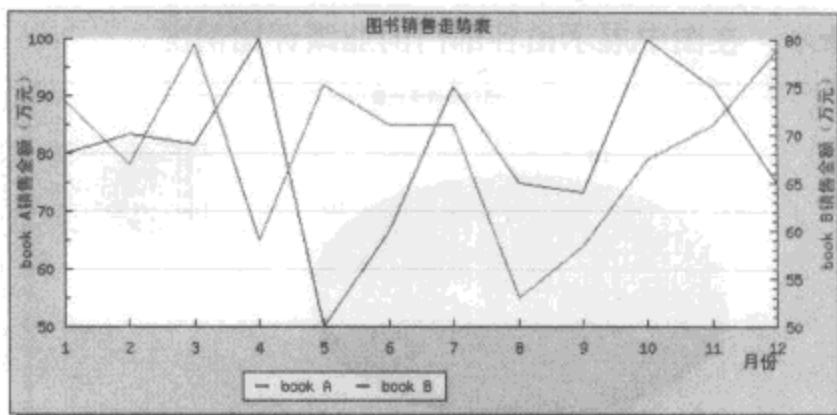


图 13.15 Jpgraph 创建折线图统计图书销售走势

### 13.5.6 Jpgraph 创建 3D 饼形图展示部门业绩比较

#### ■ 视频讲解：光盘\TM\Video\13\Jpgraph 创建 3D 饼形图展示部门业绩比较.exe

用 Jpgraph 类库制作统计图功能极其强大，不仅可以绘制平面图形，而且可以绘制具有 3D 效果的图形。直接使用 GD2 函数库可以绘制出各种图形，当然也包括 3D 饼形图，但使用 GD2 函数绘制 3D 图形需要花费大量的时间，而且相对复杂，而采用 Jpgraph 类库绘制 3D 饼图却十分方便、快捷。

**例 13.12** 使用 Jpgraph 创建 3D 饼形图进行部门业绩比较。程序最开始用文件包含函数导入 Jpgraph 类库的饼形图绘制功能，然后即可使用相关方法进行绘制。（实例位置：光盘\TM\Instances\13\12）

该实例的实现过程如下：

(1) 在程序中导入 Jpgraph 类库及饼图绘制功能，代码如下。

```
require_once 'src/jpgraph.php'; //导入 Jpgraph 类库
require_once 'src/jpgraph_pie.php'; //导入 Jpgraph 类库的饼图功能
require_once 'src/jpgraph_pie3d.php'; //导入 Jpgraph 类库的 3D 饼形图功能
```

(2) 创建数值型数组作为统计数据，代码如下。

```
$data = array(89, 78, 99, 65, 92); //设置统计数据
```

(3) 创建统计图对象，并对统计图的标题内容、字体进行设置，代码如下。

```
$graph = new PieGraph(600, 300);
$graph->SetShadow();
$graph->title->Set("部门业绩评比表");
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);
```

(4) 创建 3D 饼图对象并输入统计图，代码如下。

```
$pieplot = new PiePlot3D($data); //创建 3D 饼形图对象
$pieplot->SetCenter(0.4);
$month = array('ASP 部门', 'PHP 部门', 'JSP 部门', '.net 部门', 'C++ 部门');
$pieplot->SetLegends($month);
$graph->legend->SetFont(FF_SIMSUN, FS_BOLD); //设置字体
$graph->Add($pieplot); //将 3D 饼形图添加到统计图对象中
$graph->Stroke(); //输出图形
```

运行结果如图 13.16 所示，在图中展示出各部门的业绩评比情况。

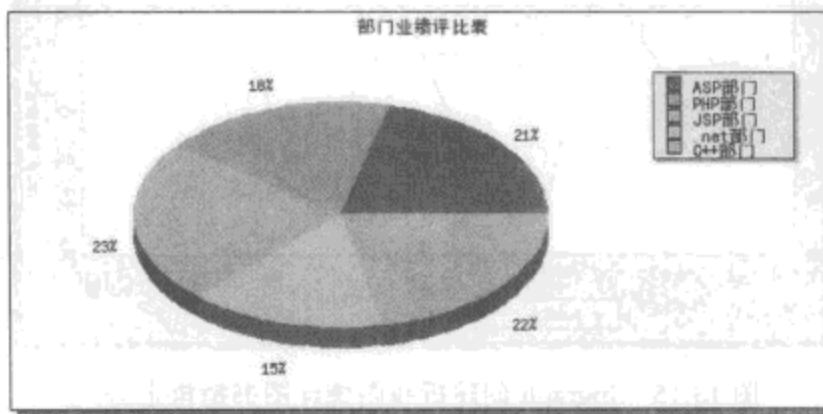


图 13.16 Jpgraph 创建 3D 饼形图进行部门业绩比较

## 13.6 本章小结

通过本章的学习，读者可以了解 PHP 中 GD2 函数库以及 Jpgraph 类库的配置和使用方法，掌握常用 GD2 函数的语法格式和最基本的 GD2 图形的绘制方法，并能够灵活运用 Jpgraph 类库绘制具有独特风格的统计图。读者如果能够在此基础上进一步研究 PHP 中的 GD2 函数库和 Jpgraph 类库的用法，定能在 PHP 图表和图形制作方面有新的突破，而且能够使自身的编程水平得到明显的提高。

## 13.7 实战练习

1. 应用 GD2 函数绘制 3D 饼形图分析软件图书市场占有率，要求运行效果与图 13.17 类似。（答案位置：光盘\TM\Instances\13\13）

2. 使用 Jpgraph 创建柱状图统计年度营业额，要求效果如图 13.18 所示。（答案位置：光盘\TM\Instances\13\14）

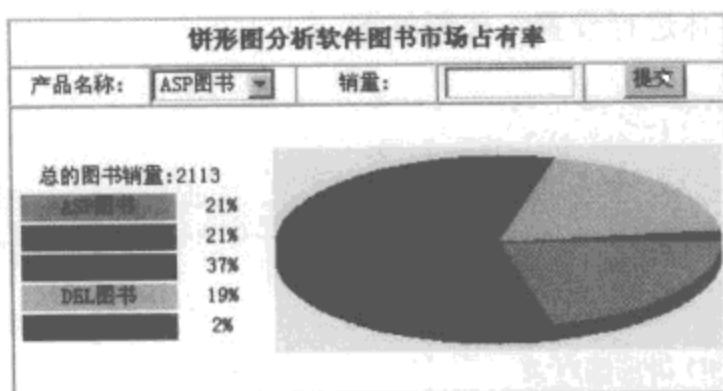


图 13.17 应用饼形图分析软件图书市场占有率

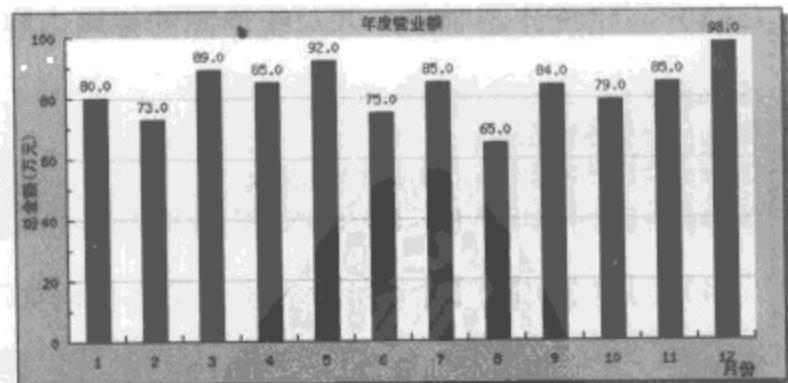


图 13.18 应用柱状图统计年度营业额

# 第 14 章

## 文件系统

( 视频讲解：74分钟)

文件是用来存取数据的方式之一。相对于数据库来说，文件在使用上要方便、直接得多。如果数据较少、较简单，使用文件无疑是最合适的方法。PHP 提供了一系列的文件函数可以对文件进行操作，同时也支持文件上传功能，并且可通过修改 php.ini 文件的配置控制上传文件。

通过阅读本章，可以学习到：

- ■ 如何打开和关闭文件
- ■ 如何读写文件
- ■ 如何操作文件
- ■ 文件上传技术
- ■ 如何控制上传文件
- ■ 如何访问远程文件
- ■ 文件指针的应用
- ■ 如何锁定文件
- ■ 目录处理技术

## 14.1 文件处理

文件处理技术是 PHP 的基本处理能力之一，通过该技术可以有效地管理文件和目录。学习文件处理技术的关键是思路要清晰，并且掌握文件处理的步骤和常用函数，就可以运用自如。

例如，访问一个文件，只需 3 步就可以：打开文件、读写文件和关闭文件。其他的操作要么是包含在读写文件中（如显示内容、写入内容等），要么与文件自身的属性有关系（如文件遍历、文件改名等）。在本节中，将对 PHP 的常用文件处理函数进行详细讲解。

### 14.1.1 打开文件

有打开就有关闭，所以 `fopen()` 函数和 `fclose()` 函数是成对出现的，否则就会出现错误。打开文件可不像平时使用 Note、Word 程序那么简单。一不小心就有可能将文件内容全部删掉。

在 PHP 中使用 `fopen()` 函数打开一个文件，语法如下：

```
resource fopen ( string filename, string mode [, bool use_include_path [,resource context]] )
```

参数 `filename` 是要打开的文件名。参数 `filename` 可以是包含文件路径的文件名，如 `C:\windows\php.ini` 或者 `\php\php.ini`。在参数 `filename` 指定的文件名中，为了避免不同系统之间切换可能带来的麻烦，采用 “\” 作为路径分隔符。

参数 `filename` 也可以是由某种协议给出的 URL，如 `http://mingrisoft.com` 或者 `ftp://www.mingrisoft.com/`。指定 URL 地址可以打开远程文件。详细内容可参考 14.3.1 节。

参数 `mode` 是打开文件的方式，可取的值如表 14.1 所示。

表 14.1 `fopen()` 函数中参数 `mode` 的取值列表

mode	模式名称	说 明
r	只读	读模式——进行读取，文件指针位于文件的开头
r+	只读	读写模式——进行读写，文件指针位于文件的开头。在现有文件内容的末尾之前进行写入就会覆盖原有内容
w	只写	写模式——进行写入，文件指针指向头文件。如果该文件存在，则所有文件内容被删除；如果文件不存在，则函数将创建这个文件
w+	只写	写模式——进行读写，文件指针指向头文件。如果该文件存在，则所有文件的内容被删除；如果该文件不存在，则函数将创建这个文件
x	谨慎写	写模式打开文件，从文件头开始写。如果文件已经存在，则该文件将不会被打开，函数返回 <code>false</code> ，PHP 将产生一个警告
x+	谨慎写	读/写模式打开文件，从文件头开始写。如果文件已经存在，则该文件将不会被打开，函数返回 <code>false</code> ，PHP 将产生一个警告
a	追加	追加模式打开文件，文件指针指向尾文件。如果该文件已有内容，则将从文件末尾开始追加；如果该文件不存在，则函数将创建这个文件
a+	追加	追加模式打开文件，文件指针指向头文件。如果该文件已有内容，则从文件末尾开始追加或者读取；如果该文件不存在，则函数将创建这个文件

续表

mode	模式名称	说 明
b	二进制	二进制模式——用于与其他模式进行连接。如果文件系统能够区分二进制文件和文本文件，可能会使用它。Windows 可以区分；UNIX 则不区分。推荐使用这个选项，便于获得最大程度的可移植性。它是默认模式
t	文本	用于与其他模式的结合。这个模式只是 Windows 下的一个选项

**◆ 注意：**文本模式打开文件与二进制方式打开文件的不同点是，文本模式打开文件，操作系统会对内容进行转义，在 Windows 下，可以透明地将\n 转换为\r\n；而以二进制方式打开文件，文件不经过操作系统预处理，读取的内容与文件在磁盘上保存的内容相同。

可选布尔型参数 `use_include_path` 决定是否在 `include_path`（由配置文件 `php.ini` 中的 `include_path` 参数进行设置）定义的目录中搜索 `filename` 文件。例如，在 `php.ini` 文件中指定一个路径 `F:\AppServ\www\index.php`，如果希望服务器在这个路径下打开所指定的文件，即可将此参数设置为 1 或 `true`。

可选资源型参数 `context` 称为上下文，是设置流操作的特定选项，用于控制流的操作特性。一般情况下只需使用默认的流操作设置，不需要使用此参数。

应用 `fopen()` 函数打开指定的文件，代码如下。

```
<?php
//以只读方式打开当前执行脚本所在目录下的 count.txt 文件
$file1=fopen("./count.txt","r");
//以读写方式打开指定文件夹下的文件，并清空文件内容
$file2=fopen("C:/count.txt","r+");
//以二进制只写方式打开指定文本，并清空文件
$file3=fopen("C:/text.txt","wb");
//以只读方式打开远程文件
$file4=fopen("http://192.168.1.249/in/index.php","r");
?>
```

### 14.1.2 读取文件

相对打开文件和关闭文件来说，从文件中读取数据更复杂一些。利用 PHP 提供的文件处理函数，可以读取一个字符、一行字串或者整个文件，也可以读取任意长度的字串。

#### 1. 读取一个字符：`fgetc()`

在对某一个字符进行查找、替换时，就需要有针对性地对某个字符进行读取，在 PHP 中可以使用 `fgetc()` 函数实现此功能。函数语法如下：

```
string fgetc ( resource handle )
```

该函数返回一个字符，该字符从 `handle` 指向的文件中得到，遇到 EOF 则返回 `false`。

**例 14.1** 使用 `fgetc` 函数逐个字符读取文件 `count.txt` 的内容并输出，代码如下：(实例位置：光盘\TM\Instances\14\1)

```

<?php
    $fopen = fopen('./count.txt','rb');           //创建文件资源
    while(false !== ($chr = fgetc($fopen))){      //使用 fgetc 函数取得一个字符，判断是否为 false
        echo $chr;                                //如果不是，输出该字符
    }
    fclose($fopen);                            //关闭文件资源
?>

```

结果为：明日科技

## 2. 读取一行数据：fgets()和fgetss()

### (1) fgets()函数

fgets()函数用于一次读取一行数据。语法如下：

```
string fgets( int handle [, int length] )
```

参数 handle 是被打开的文件，参数 length 是要读取的数据长度。函数能够实现从指定文件 handle 中读取一行并返回长度最大值为 length-1 个字节的字符串。在遇到换行符、EOF 或者读取了 length-1 个字节后停止。如果忽略 length 参数，那么读取数据直到行结束。

### (2) fgetss()函数

fgetss()函数是 fgets()函数的变体，用于读取一行数据，同时 fgetss()函数会过滤掉被读取内容中的 html 和 php 标记。语法如下：

```
string fgetss ( resource handle [, int length [, string allowable_tags]] )
```

该函数能够从读取的文件中去掉任何 html 和 php 标记。它可以使用 allowable\_tags 参数来控制哪些标记不被去掉。

**例 14.2 分别使用 fgets()函数和 fgetss()函数读取 count.php 文件并输出结果，看一看有什么区别，代码如下：**（实例位置：光盘\TM\Instances\14\2）

```

<!-- 使用 fgets 函数读取.php 文件 --><!-- ----- --> </td>
<?php
    $fopen = fopen('./count.php','rb');           //使用 feof 函数测试指针是否到了文件结束的位置
    while(!feof($fopen)){
        echo fgets($fopen);                      //输出当前行
    }
    fclose($fopen);
?>
<!-- fgetss 函数读取.php 文件 -->
<?php
    $fopen = fopen('./count.php','rb');           //使用 feof()函数测试指针是否到了文件结束的位置
    while(!feof($fopen)){
        echo fgetss($fopen);                     //输出当前行
    }
    fclose($fopen);
?>

```

运行结果如图 14.1 所示。

### 3. 读取任意长度的字串: fread()

fread()函数可以从文件中读取指定长度的数据。语法如下:

```
string fread( int handle, int length )
```

参数 handle 为指向的文件资源, length 是要读取的字节数。当函数读取 length 个字节或到达 EOF 时停止执行。

**例 14.3 使用 fread() 函数读取文件 count.txt 的内容。代码如下:** (实例位置: 光盘\TM\Instances\14\3)

```
<?php
    $filename = "count.txt"; //要读取的文件
    $fp = fopen($filename,"rb"); //打开文件
    echo fread($fp,100); //使用 fread() 函数读取文件内容的前 100 个字节
    echo "<p>";
    echo fread($fp,filesize($filename)); //输出其余的文件内容
?>
```

运行结果如图 14.2 所示。

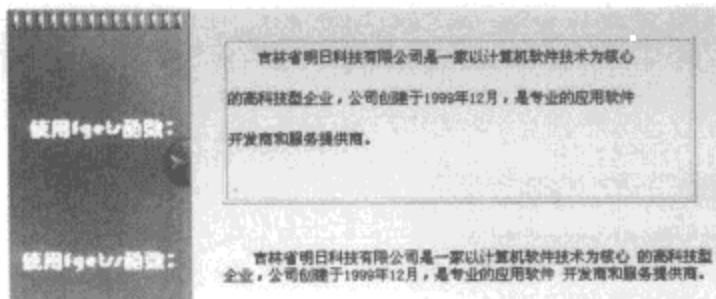


图 14.1 fgets()和 fgetss()函数的区别

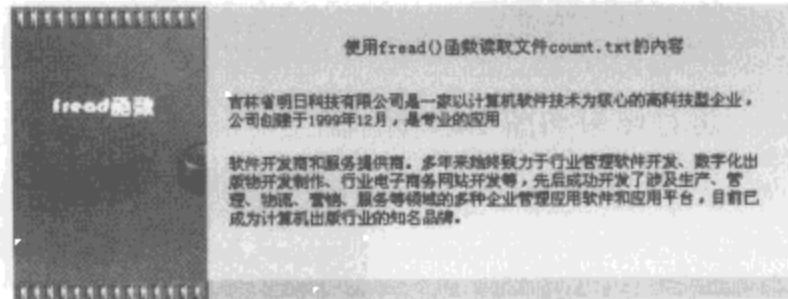


图 14.2 使用 fread()函数读取文件

### 4. 读取整个文件: readfile()、file()和 file\_get\_contents()

#### (1) readfile()函数

readfile()函数用于读入一个文件并将其写入到输出缓冲, 如果出现错误则返回 false。语法如下:

```
int readfile(string filename)
```

使用 readfile()函数, 不需要打开/关闭文件, 不需要“echo/print”等输出语句, 直接写出文件路径即可。

#### (2) file()函数

file()函数也是读取整个文件的内容, 只是 file()函数将文件内容按行存放到数组中, 包括换行符在内。如果失败则返回 false。语法如下:

```
array file(string filename)
```

#### (3) file\_get\_contents()函数

该函数将文件内容 (filename) 读入一个字符串。如果有 offset 和 maxlen 参数, 将在参数 offset 所指定的位置开始读取长度为 maxlen 的内容。如果失败, 返回 false。语法如下:

```
string file_get_contents ( string filename[, int offset [, int maxlen]] )
```

该函数适用于二进制对象，是将整个文件的内容读入到一个字符串中的首选方式。

**例 14.4 使用 readfile() 函数、file() 函数和 file\_get\_contents() 函数分别读取文件 count.txt 的内容，代码如下：**（实例位置：光盘\TM\Instances\14\4）

```
<!-- 使用 readfile() 函数读取 count.txt 文件的内容 -->
<?php
    readfile('count.txt');
?>
<!-- 使用 file() 函数读取 count.txt 文件的内容 -->
<?php
    $f_arr = file('count.txt');
    foreach($f_arr as $cont){
        echo $cont."<br>";
    }
?>
<!-- 使用 file_get_contents() 函数读取 count.txt 文件的内容 -->
<?php
    $f_chr = file_get_contents('count.txt');
    echo $f_chr;
?>
```

运行结果如图 14.3 所示。

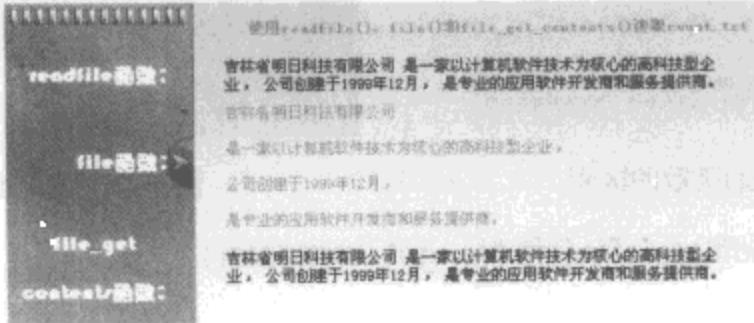


图 14.3 读取整个文件

### 14.1.3 写入文件

写入数据也是 PHP 中常用的文件操作，在 PHP 中可以使用 fwrite() 和 file\_put\_contents() 函数向文件中写入数据。fwrite() 函数还有个别名叫 fputs()，它们的用法相同。fwrite() 函数的语法如下：

```
int fwrite ( resource handle, string string [, int length] )
```

该函数把内容 string 写入文件指针 handle 处。如果指定长度 length，当写入 length 个字节时，写入就会停止。如果文件内容长度小于 length，则会输出全部文件内容。

file\_put\_contents() 函数是 PHP 5 新加的函数。语法如下：

```
int file_put_contents ( string filename, string data [, int flags] )
```

- filename 为写入数据的文件。
- data 为要写入的数据。
- flags 可以是 FILE\_USE\_INCLUDE\_PATH、FILE\_APPEND 或 LOCK\_EX，这里只要知道 LOCK\_EX 的含义即可，LOCK\_EX 意为独占锁定，在 14.3.3 节中将会介绍。

使用 file\_put\_contents() 函数，和依次调用 fopen()、fwrite()、fclose() 函数的功能是一样的。下面通过一个实例比较一下新函数的优越性。

**例 14.5** 首先应用 fwrite() 函数向 count.txt 文件写入数据，然后使用 file\_put\_contents() 函数写入数据，代码如下：（实例位置：光盘\TM\Instances\14\5）

```
<?php
    $filepath = "count.txt";
    $str = "吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业，公司创建于 1999 年 12 月，是专业的应用软件开发商和服务提供商。";
    $fopen = fopen($filepath,'wb') or die('文件不存在');
    fwrite($fopen,$str);
    fclose($fopen);
    readfile($filepath);
?>
<?php
    file_put_contents($filepath,$str);
    readfile($filepath);
?>
```

运行结果如图 14.4 所示。

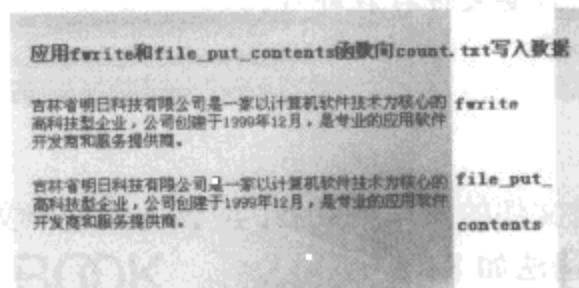


图 14.4 使用 fwrite() 和 file\_put\_contents() 函数写入数据

#### 14.1.4 操作文件

除了可以对文件内容进行读写，对文件本身同样也可以进行操作，如复制文件、重新命名、查看修改日期等。PHP 内置了大量的文件操作函数，常用的文件函数如表 14.2 所示。

表 14.2 常用的文件操作函数

函数原型	函数说明	举例
bool copy( string path1, string path2 )	将文件从 path1 复制到 path2。如果成功，则返回 true；失败返回 false	copy('tm.txt','..//tm.txt')
bool rename( string filename1, string filename2 )	把 name1 重新命名为 name2	rename('1.txt','tm.txt')

续表

函数原型	函数说明	举例
bool unlink( string filename )	删除文件，成功返回 true，失败返回 false	unlink('./tm.txt')
int fileatime( string filename )	返回文件最后一次被访问的时间，时间以 UNIX 时间戳的方式返回	fileatime('1.txt')
int filemtime( string filename )	返回文件最后一次被修改的时间，时间以 UNIX 时间戳的方式返回	date('Y-m-d H:i:s', filemtime('1.txt'))
int filesize( string filename )	取得文件 filename 的大小 (bytes)	filesize('1.txt')
array pathinfo(string name [, int options])	返回一个数组，包含文件 name 的路径信息。有 dirname、basename 和 extension。可通过 option 设置要返回的信息，有 PATHINFO_DIRNAME、PATHINFO_BASENAME 和 PATHINFO_EXTENSION。默认为返回全部	\$arr = pathinfo('/tm/sl/12/5/1.txt'); foreach(\$arr as \$method => \$value){ echo \$method.": ".\$value." };
string realpath ( string filename )	返回文件 filename 的绝对路径，如 c:\tmp\...\1.txt	realpath('1.txt')
array stat ( string filename )	返回一个数组，包括文件的相关信息，如上面提到的文件大小、最后修改时间等	\$arr = stat('1.txt'); foreach(\$arr as \$method => \$value){ echo \$method.": ".\$value." };

说明：在读写文件时，除了 file()、readfile()等少数几个函数外，其他操作必须要先使用 fopen()函数打开文件，最后用 fclose()函数关闭文件。而文件的信息函数（filesize()、filemtime()等）则都不需要打开文件，只要文件存在即可。

### 14.1.5 关闭文件

文件有打开就应该有关闭，对文件的操作结束后，应该关闭这个文件，否则可能引起错误。在 PHP 中使用 fclose()函数来关闭文件。语法如下：

```
bool fclose ( resource handle );
```

该函数将参数 handle 指向的文件关闭，如果成功则返回 true，否则返回 false。其中的文件指针必须是有效的，并且是通过 fopen()函数成功打开的文件。有关该函数的应用可以参考上面的实例，这里不再赘述。

## 14.2 文件上传

 视频讲解：光盘\TM\Video\14\文件上传.exe

文件上传可以通过 HTTP 协议来实现。要使用文件上传功能，首先要在配置文件 php.ini 中对上传做一些设置，然后通过预定义变量\$\_FILES 对上传文件做一些限制和判断；最后使用 move\_uploaded\_file()函数实现上传。

### 14.2.1 配置 php.ini 文件

实现上传功能，首先要在 `php.ini` 中开启文件上传，并对其中的一些参数作出合理的设置。找到 `File Uploads` 项。可以看到下面有 3 个属性值，表示含义如下。

- `file_uploads`: 如果值是 `on`，说明服务器支持文件上传；如果为 `off`，则不支持。
- `upload_tmp_dir`: 上传文件临时目录。在文件被成功上传之前，文件首先存放到服务器端的临时目录中。如果要指定位置，就在这里进行设置。否则使用系统默认目录即可。
- `upload_max_filesize`: 服务器允许上传文件的最大值，以 MB 为单位。系统默认为 2MB，用户可以自行设置。

除了 `File Uploads` 项，还有几个属性也会影响到上传文件的功能。

- `max_execution_time`: PHP 中一个指令所能执行的最大时间，单位是秒。
- `memory_limit`: PHP 中一个指令所分配的内存空间，单位是 MB。

`php.ini` 文件配置完成后，需要重新启动 Apache 服务器，配置才能生效。

### 14.2.2 预定义变量`$_FILES`

`$_FILES` 变量存储的是上传文件的相关信息，这些信息对于上传功能有很大的作用。该变量是一个二维数组。保存的信息如表 14.3 所示。

表 14.3 预定义变量`$_FILES` 元素

元素名	说明
<code>\$_FILES[filename][name]</code>	存储了上传文件的文件名，如 <code>exam.txt</code> 、 <code>myDream.jpg</code> 等
<code>\$_FILES[filename][size]</code>	存储了文件大小，单位为字节
<code>\$_FILES[filename][tmp_name]</code>	文件上传时，首先在临时目录中被保存成一个临时文件。该变量为临时文件名
<code>\$_FILES[filename][type]</code>	上传文件的类型
<code>\$_FILES[filename][error]</code>	存储了上传文件的结果。如果返回 0，则说明文件上传成功

例 14.6 实现一个上传文件域，通过`$_FILES` 变量输出上传文件的资料，代码如下：（实例位置：光盘\TM\Instances\14\6）

```
<!-- 上传文件的 form 表单，必须有 enctype 属性 -->
<form action="" method="post" enctype="multipart/form-data">
    <!-- 上传文件域，type 类型为 file -->
    <input type="file" name="upfile"/>
    <!-- 提交按钮 -->
    <input type="submit" name="submit" value="上传" />
</form>
<!-- 处理表单返回结果 -->
<?php
    if(!empty($_FILES)){
        foreach($_FILES['upfile'] as $name => $value)
            echo $name.'='.$value.'<br>';
    }
//判断变量$_FILE 是否为空
//使用foreach循环输出上传文件信息的名和值
```

```

    }
?>

```

运行结果如图 14.5 所示。



图 14.5 \$\_FILES 预定义变量

### 14.2.3 文件上传函数

PHP 中应用 move\_uploaded\_file() 函数实现文件上传。move\_uploaded\_file() 函数将指定文件上传到服务器中指定的位置。如果成功则返回 true，否则返回 false。语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

参数 filename 指定上传文件的临时文件名，即\$\_FILES[tmp\_name]；参数 destination 指定文件上传后保存的新路径和名称。

**例 14.7** 创建一个上传表单，允许上传 2MB 以下、格式为 jpg 的图片文件，将上传文本保存在根目录下的 upfile 文件夹下，上传成功浏览该目录，代码如下：（实例位置：光盘\TM\Instances\14\7）

```

<!-- 上传文件的 form 表单，必须有 enctype 属性 -->
<form action="" method="post" enctype="multipart/form-data">
<br>
    <td width="150" height="30" align="right" valign="middle">请选择上传文件：</td>
    <td width="250"><input type="file" name="up_file"/></td>
    <td width="100"><input type="submit" name="submit" value="上传" /></td>
</tr>
</form>
<?php
    if(!empty($_FILES[up_file][name])){
        /* 判断是否有上传文件 */
        $fileinfo = $_FILES[up_file];
        /* 将文件信息赋给变量$fileinfo */
        $type=strstr($fileinfo['name'], ".");
        //获取文件类型
        if($type!=".JPG"){
            echo "您上传文件格式不正确！";
        }else{
            if($fileinfo['size'] < 2097152 && $fileinfo['size'] > 0){
                /* 判断文件大小 */
                $path="upfile/".$_FILES["up_file"]["name"];
                //定义上传文件的路径
            }
        }
    }
?>
```

```

move_uploaded_file($fileinfo['tmp_name'],$path); //上传文件
//上传成功后，浏览该目录
if(is_dir("upfile")){
    $dir = scandir("upfile/");
    foreach($dir as $value){
        echo $value."<br>";
    }
} else{
    echo "目录路径错误！";
}
else{
    echo '文件大小不符合要求';
}
}

?>

```

运行结果如图 14.6 所示。

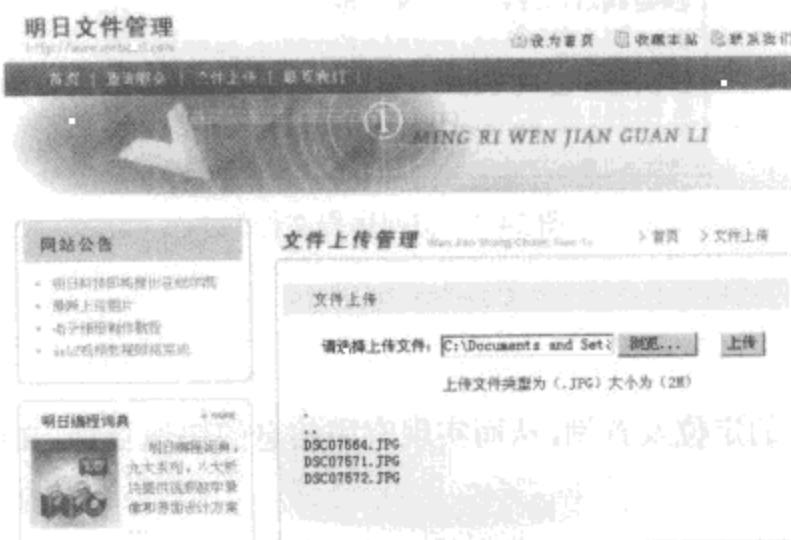


图 14.6 单文件上传

### 14.3 文件处理的高级应用

#### ■ 视频讲解：光盘\TM\Video\14\高级应用.exe

在 PHP 中除了可以对文件进行基本的读写操作外，还可以对文件指针进行查找、定位，对正在读取的文件进行锁定等，本节将进一步学习文件处理的一些高级技术。

#### 14.3.1 远程文件的访问

在 PHP 中如果要访问远程文件，就必须将配置文件 php.ini 中的参数 allow\_url\_fopen 设置为开启。allow\_url\_fopen 参数默认是开启的，允许打开 http 协议和 ftp 协议指定的远程文件；如果 allow\_url\_fopen

设置为 off，则不允许打开远程文件，函数将返回 false。

**例 14.8 应用 fopen() 函数和 readfile() 函数访问一个远程文件，代码如下：(实例位置：光盘\TM\Instances\14\8)**

```
<?php
    //访问远程文件
    $filename = "http://192.168.1.249/admin/file/20080330924147598412.txt"; //要读取的文件
    $fp = fopen($filename,"rb"); //打开文件
    echo fread($fp,50); //使用 fread() 函数读取文件内容的前 100 个字节
    echo '<p>';
    //通过 readfile 读取整个文件的内容
    readfile("http://192.168.1.249/admin/file/20080330924147598412.txt","r");
?>
```

运行结果如图 14.7 所示。

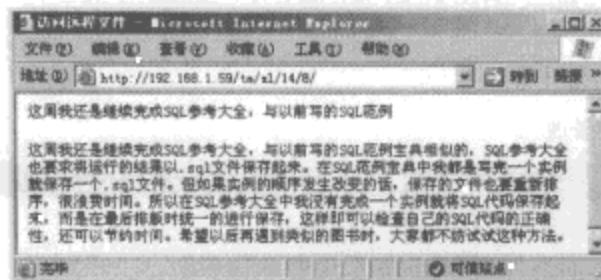


图 14.7 访问远程文件

### 14.3.2 文件指针

PHP 可以实现文件指针的定位及查询，从而实现有用信息的快速查询。文件指针函数包括 rewind()、fseek()、feof() 和 ftell()。

#### 1. rewind() 函数

该函数将文件 handle 的指针设为文件流的开头。语法如下：

```
bool rewind ( resource handle )
```

**注意：**如果将文件以附加（“a”）模式打开，写入文件的任何数据总是会被附加在后面，不论文件指针在何处。

#### 2. fseek() 函数

fseek() 函数实现文件指针的定位。语法如下：

```
int fseek ( resource handle, int offset [, int whence] )
```

- ☒ handle 参数为要打开的文件。
- ☒ offset 为指针位置或相对 whence 参数的偏移量，可以是负值。
- ☒ whence 的值包括如下 3 种：
  - SEEK\_SET，位置等于 offset 字节。

- SEEK\_CUR, 位置等于当前位置加上 offset 字节。
- SEEK\_END, 位置等于文件尾加上 offset 字节。

如果忽略 whence 参数, 系统默认为 SEEK\_SET。

### 3. feof()函数

该函数判断文件指针是否在文件尾。语法如下:

```
bool feof( resource handle )
```

如果文件指针处于文件结束的位置, 就返回 true, 否则返回 false。

### 4. ftell()函数

ftell()函数返回当前指针的位置。语法如下:

```
int ftell( resource handle )
```

**例 14.9 使用 4 个指针函数来输出文件 count.txt 中的内容, 代码如下: (实例位置: 光盘\TM\Instances\14\9)**

```
<?php
$filename = "count.txt";
if(is_file($filename)){
    echo "文件总字节数: ".filesize($filename)."<br>";
    $fopen = fopen($filename,'rb');
    echo "初始指针位置是: ".ftell($fopen)."<br>";
    fseek($fopen,50);
    echo "使用 fseek()函数后指针位置: ".ftell($fopen)."<br>";
    echo "输出当前指针后面的内容: ".fgets($fopen)."<br>";
    if(feof($fopen))
        echo "当前指针指向文件末尾: ".ftell($fopen)."<br>";
    rewind($fopen);
    echo "使用 rewind()函数后指针的位置: ".ftell($fopen)."<br>";
    echo "输出前 33 个字节的内容: ".fgets($fopen,33);
    fclose($fopen);
} else{
    echo "文件不存在";
}
?>
```

//指定文件路径及文件名  
//判断文件是否存在  
//输出总字节数  
//打开文件  
//输出指针位置  
//移动指针  
//输出移动后的指针位置  
//输出从当前指针到行尾的内容  
//判断指针是否指向文件末尾  
//如果指向了文件尾, 则输出指针位置  
//使用 rewind()函数  
//查看使用 rewind()函数后指针的位置  
//输出前 33 个字节的内容  
//关闭文件

运行结果如图 14.8 所示。

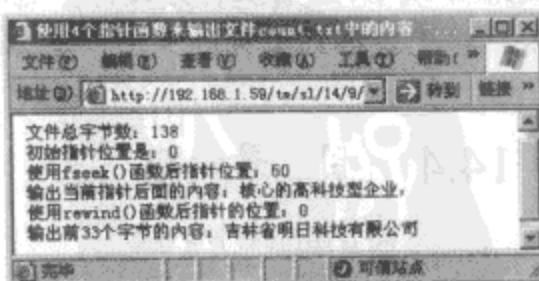


图 14.8 文件指针函数

### 14.3.3 锁定文件

在向一个文本文件写入内容时，需要先锁定该文件，以防其他用户同时也修改此文件内容。PHP 中锁定文件的函数为 flock()，语法如下：

```
bool flock ( int handle, int operation )
```

参数 handle 为一个已经打开的文件指针，operation 的参数值如表 14.4 所示。

表 14.4 operation 的参数值

参数值	说 明
LOCK_SH	取得共享锁定（读取程序）
LOCK_EX	取得独占锁定（写入程序）
LOCK_UN	释放锁定
LOCK_NB	防止 flock() 函数在锁定时堵塞

例 14.10 使用 flock 锁定文件，然后再写入数据，最后解除锁定，关闭文件。代码如下：（实例位置：光盘\TM\Instances\14\10）

```
<?php
$filename = 'count.txt';
$fd = fopen($filename,'w');
flock($fd, LOCK_EX);
fwrite($fd, "hightman1");
flock($fd, LOCK_UN);
fclose($fd);
readfile($filename);
?>
```

运行结果如图 14.9 所示。

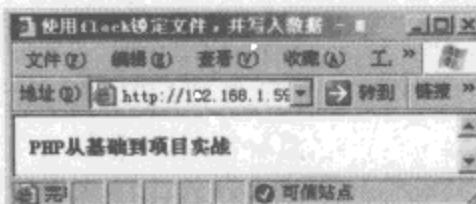


图 14.9 锁定文件并写入数据

**注意：**在向文件写入数据时，使用 w 或 w+ 选项来打开文件。这时如果使用了 LOCK\_EX，则同一时间访问此文件的其他用户将无法得到文件的大小，也不能进行写操作。

## 14.4 目录处理

**视频讲解：**光盘\TM\Video\14\目录处理.exe

目录也是文件，是一种特殊的文件。要浏览目录下的文件，首先也要打开目录，浏览完毕后，同

样要关闭目录。就这点来说，两者没有什么分别，都要经过 3 步，即打开目录、浏览目录和关闭目录。

#### 14.4.1 打开目录

打开目录和打开文件类似，但打开的文件如果不存在，则自动创建一个新文件；而打开的目录如果不正确，则一定会报错。

在 PHP 中使用 `opendir()` 函数打开目录。语法如下：

```
resource opendir( string path)
```

参数 `path` 是一个合法的目录路径，成功执行后返回目录的指针；如果 `path` 不是一个合法的目录或者因为权限、文件系统错误而不能打开目录，`opendir()` 函数将返回 `false`，并产生一个 `E_WARNING` 级别的错误信息。可以在 `opendir()` 前面加上“@”符号来抑制错误信息的输出。

#### 14.4.2 浏览目录

在 PHP 中浏览目录中的文件使用的是 `scandir()` 函数。语法如下：

```
array scandir( string directory [, int sorting_order ] )
```

该函数返回一个数组，包含 `directory` 中的所有文件和目录。参数 `sorting_order` 指定排序顺序，默认按字母升序排序，如果应用参数 `sorting_order`，则变为降序排序。

**例 14.11 浏览根目录下的所有文件，代码如下：（实例位置：光盘\TM\Instances\14\11）**

```
<?php
    $path = 'F:/AppServ/www/tm/sl/14/11';           //要浏览的目录
    if(is_dir($path)){
        $dir = scandir($path);                         //判断文件名是否为目录
        foreach($dir as $value){                       //使用 scandir() 函数取得所有文件及目录
            echo $value."<br>";                        //使用 foreach() 函数循环
        }                                               //循环输出文件及目录名称
    }else{
        echo "目录路径错误！";
    }
?>
```

运行结果如图 14.10 所示。

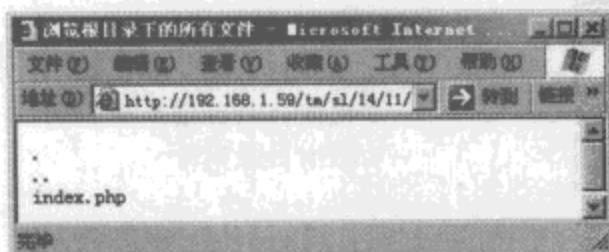


图 14.10 浏览目录

## 14.4.3 操作目录

因为目录是特殊的文件，所以对文件的操作处理函数（如重命名等）一般也适用于目录。PHP 也为目录本身提供了一些特殊操作函数。表 14.5 中列举了一些常用的目录操作函数。

表 14.5 常用的目录操作函数

函数原型	函数说明	举例
bool mkdir ( string pathname )	新建一个指定的目录	mkdir('temp');
bool rmdir ( string dirname )	删除所指定的目录，该目录必须是空的	rmdir('tmp')
string.getcwd ( void )	取得当前工作的目录	getcwd() echo getcwd() . " ";
bool.chdir ( string directory )	改变当前目录为 directory	chdir('../'); echo getcwd() . " ";
float.disk_free_space ( string directory )	返回目录中的可用空间 (bytes)。 被检查的文件必须通过服务器的文件系统访问	disk_free_space('d:\\appserv');
float.disk_total_space ( string directory )	返回目录的总空间大小 (bytes)	disk_total_space('d:\\appserv');
string.readdir ( resource handle )	返回目录中下一个文件的文件名 (使用此函数时，目录必须是使用 opendir 函数打开的)。在 PHP 5 之前，都是使用这个函数来浏览目录的	while(false==(\$path=readdir(\$handle))){ echo \$path; }
void.rewinddir ( resource handle )	将指定的目录重新指到目录的开头	rewinddir(\$handle)

## 14.4.4 关闭目录

PHP 中通过 closedir() 函数关闭目录。语法如下：

```
void.closedir ( resource handle )
```

参数 handle 为使用 opendir() 函数打开的一个目录所返回的目录指针。

例如，下面是一个打开和关闭目录的流程代码。

```
<?php  
$path = 'F:/AppServ/www/tm/sl/14/11';  
if (is_dir($path)){  
    if ($dire = opendir($path))  
        echo $dire;  
    //检测是否是一个目录  
    //判断打开目录是否成功  
    //输出目录指针  
}  
else{  
    echo '路径错误';  
    exit();  
}
```

```
...
closedir($dire);
?>
```

is\_dir()函数用于判断当前路径是否为一个合法的目录。如果合法，返回 true，否则返回 false。

## 14.5 实战

### 14.5.1 多图片上传

#### ■ 视频讲解：光盘\TM\Video\14\多图片上传.exe

在动态网页开发过程中，有时需要同时上传多幅图片到服务器中。实现同时上传多幅图片的上传原理和单图片上传是相同的，只不过在对上传的字段进行处理时增加了一个数组的应用。

**例 14.12** 在明日商城系统的后台管理中，需要上传大量的图片到服务器，这时就需要使用多图片上传功能。上传成功后使用分页技术将图片以分页的形式显示到页面中，其操作步骤如下：（实例位置：光盘\TM\Instances\14\12）

(1) 在根目录下的 conn 文件夹中创建 conn.php 文件，实现连接数据库的操作，代码如下。

```
<?php
$Id=mysql_connect("localhost","root","root") or die('连接失败:' . mysql_error());
if(mysql_select_db("db_database14",$Id))           //连接指定的数据库
    echo "";
else
    echo ('数据库选择失败:' . mysql_error());
mysql_query("set names gb2312");                  //设置编码格式
?>
```

(2) 创建 index.php 页面，添加 form 表单，设置上传图片的文件域，指定文件域的名称为数组 picture[]，同样设置图片名称文本框也为数组 files[]，并且对服务器中存储的图片进行分页输出。关键代码如下。

```
<form action="index_ok.php" method="post" enctype="multipart/form-data" name="form1">
<tr>
    <td width="88" height="25" align="center" class="STYLE1">内容 1</td>
    <td width="169" align="center"><input name="files[]" type="text" id="files[]" size="15"></td>
    <td width="321"><input name="picture[]" type="file" id="picture[]" size="30"></td>
</tr>
<!--省略了部分代码--&gt;
&lt;tr&gt;
    &lt;td&gt;&amp;nbsp;&lt;/td&gt;
    &lt;td&gt;&amp;nbsp;&lt;/td&gt;
    &lt;td&gt;&lt;input type="submit" name="Submit" value="提交"&gt;&lt;/td&gt;
&lt;/tr&gt;
&lt;/form&gt;</pre>

```



PDG

(3) 创建 index\_ok.php 文件, 获取表单中提交的数组文件, 应用\$\_FILES 变量和 move\_uploaded\_file() 函数实现图片的上传, 并且将上传图片在服务器中存储的路径、名称和时间存储到指定的数据表中, 代码如下。

```
<?php
    session_start(); // 初始化 Session 变量
    include "conn/conn.php"; // 实现与数据库连接
    if($_POST["Submit"]!=""){
        if(!is_dir("./upfile")){
            mkdir("./upfile");
        }
        $data=date("Y-m-d H:m:s"); // 获取当前时间
        $picture[]="";
        $array=array_unique($picture); // 向表单提交的数组中增加一个空元素
        array_pop($array); // 判断是否存在指定的文件夹
        for($i=0;$i<count($array);$i++){ // 如果不存在, 则创建文件夹
            $spath="upfile/".$_FILES["picture"]["name"][$i]; // 定义图片在服务器中存储的路径
            move_uploaded_file($_FILES["picture"]["tmp_name"][$i],$path); // 执行上传操作
            list($name,$value)=each($files); // 将数据添加到数据表中
            $query="insert into tb_up_file (file_test,data,file_name) values ('$path','$data','$value')";
            $result=mysql_query($query);
        }
        echo "图片上传成功, 请稍等..."; // 执行循环
        echo "<meta http-equiv=\"refresh\" content=\"3; url=index.php\">";
    }
?>
```

运行结果如图 14.11 所示。



图 14.11 多图片上传

## 14.5.2 通过文本文件统计页面访问量

### 视频讲解: 光盘\TM\Video\14\通过文本文件统计页面访问量.exe

在进行网站访问量统计的过程中, 可以将统计的数据存储到数据库中, 同样也可以将数据存储到文本文件中。如果使用文本文件来统计访问量的数据, 那么一定会应用到文件系统函数。

**例 14.13** 通过文本文件 counter.txt 来统计页面的访问量, 并通过 GD2 函数来输出访问结果。具体操作步骤如下: (实例位置: 光盘\TM\Instances\14\13)

(1) 创建 index.php 页面。将页面的每次访问结果通过文件系统函数写入到指定的文本文件中，并且通过 Session 变量防止因页面刷新而导致统计的数据不准确。并且以 png 图像的形式输出页面的访问结果，代码如下。

```
<?php session_start();
if($_SESSION[temp]== ""){ //判断$_SESSION[temp]==" 的值是否为空, 其中的 temp 为自定义的变量
    if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
    }else{
        $counter=fgets($fp,1024); //读取文件中数据
        fclose($fp); //关闭文本文件
        $counter++;
        $fp=fopen("counter.txt","w"); //以写的方式打开文本文件<!---->
        fputs($fp,$counter); //将新的统计数据增加 1
        fclose($fp);
    }
    $_SESSION[temp]=1; //登录以后, $_SESSION[temp]的值不为空, 给$_SESSION[temp]赋值 1
}
?>
//输出访问结果

```

(2) 创建 gd1.php 文件，应用 GD2 函数，将从文本文件中读取出的数据输出到 png 图像中，代码如下。

```
<?php
//以图形的形式输出数据库中的记录数
if(($fp=fopen("counter.txt","r"))==false){ //打开指定的文本文件
    echo "打开文件失败!";
} else{
    $counter=fgets($fp,1024); //读取文本文件中的数据
    fclose($fp); //关闭文件
    $im=imagecreate(240,24); //通过 GD2 函数创建画布
    $gray=imagecolorallocate($im,255,255,255); //定义背景
    $color =imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255)); //定义字体颜色
    //输出中文字符
    $text=iconv("gb2312","utf-8","网站的访问量:");
    $font = "Fonts/FZHCJW.TTF";
    imagettftext($im,14,0,20,18,$color,$font,$text); //对指定的中文字字符串进行转换
    //设置中文字字符使用的字体
    imagestring($im,5,160,5,$counter,$color); //输出中文
    imagepng($im); //输出网站的访问次数
    imagedestroy($im); //生成 png 图像
    //销毁图像
}
?>
```

运行结果如图 14.12 所示。



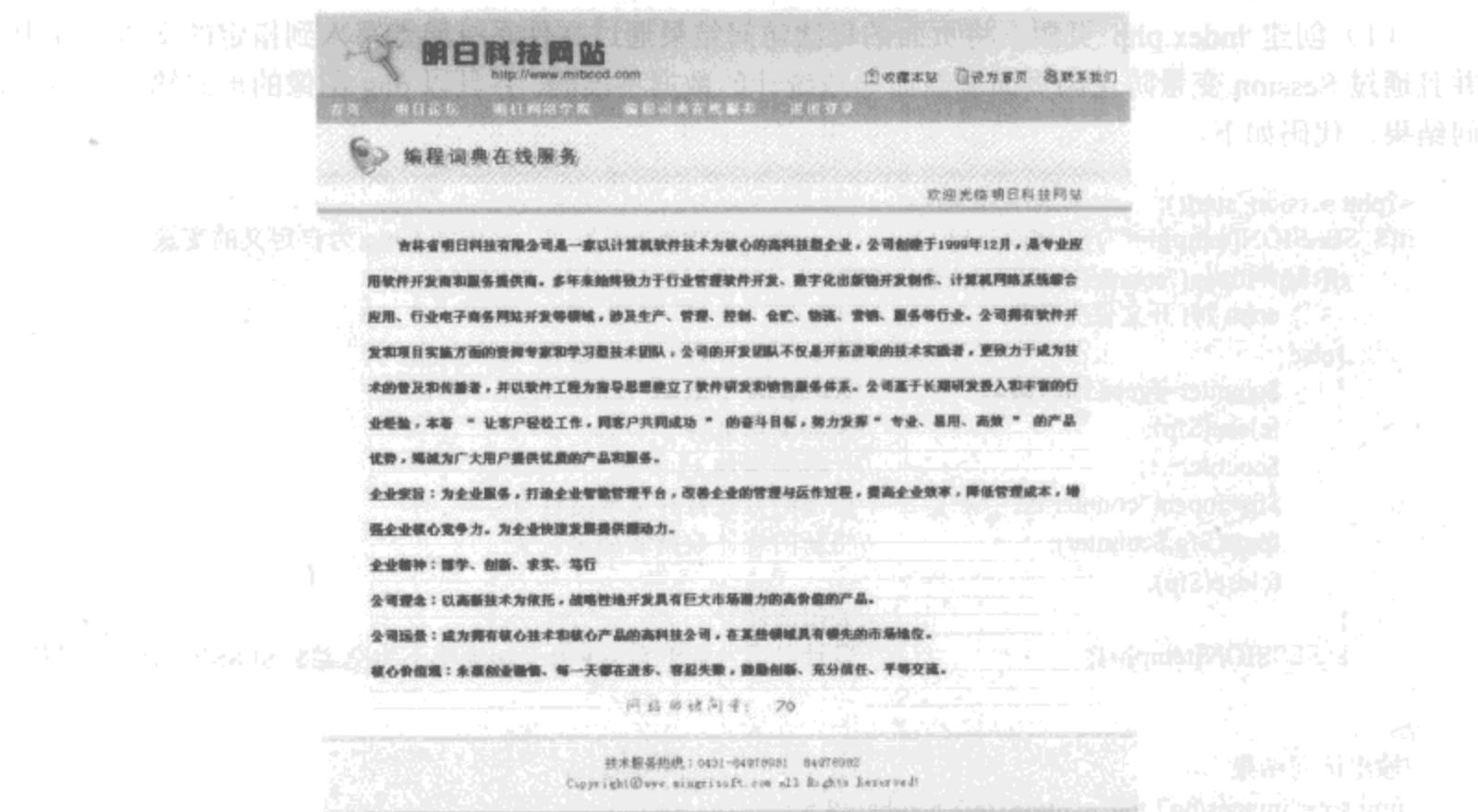


图 14.12 通过文本文件统计页面访问量

## 14.6 本 章 小 结

本章介绍了文件和目录操作的基本流程，并且对其中所使用的函数进行了详细的讲解。本章的学习重点是文件上传技术，包括对 php.ini 文件的配置、函数的讲解和实现的过程都是作为一个 PHP 程序员必须掌握的。希望读者能够在实例和实践练习的帮助下，深入理解本章的重点知识，牢固掌握常用函数。

## 14.7 实战练习

1. 文件系统函数在超长文本文件分页输出中的应用。（答案位置：光盘\TM\Instances\14\14）
2. 限制大小的文件上传。（答案位置：光盘\TM\Instances\14\15）
3. 通过目录处理函数实现文件浏览的功能。（答案位置：光盘\TM\Instances\14\16）

# 第15章

基础与进阶 1.2.1

## 面向对象

( 视频讲解: 160分钟)

在本章中，将学习面向对象编程的基本概念，包括类、对象、继承、多态等。通过学习这些概念，可以更好地理解 PHP 的面向对象编程机制，从而提高代码的可读性和可维护性。

函数是 PHP 实现结构化编程的重要因素，但函数与变量是分离的，当项目较大或者程序的变量较多时，非常不利于开发人员开发和维护。面向对象的编程方式是如今开发模式的主流，具有独立性、灵活性和可重用性等特点，非常适合大型项目的开发。PHP 能够成为 Web 开发领域的主流语言，面向对象的开发模式也是重要原因之一。

通过阅读本章，可以学习到：

- 面向对象的概念
- PHP 中的面向对象的编程方式和特点
- 类的定义及实例化
- 声明类成员
- 继承和多态的实现
- PHP 的高级应用
- 抽象类的实现
- 接口的使用
- 魔术方法



## 15.1 面向对象的基本概念

面向对象包括 3 部分：面向对象分析（Object Oriented Analysis, OOA）、面向对象设计（Object Oriented Design, OOD）以及面向对象编程（Object Oriented Programming, OOP）。面向对象编程的两个重点概念是类和对象。

### 15.1.1 类

类是变量与作用于这些变量的方法的集合，事物都具有其自身的属性和方法，通过这些属性和方法可以将不同物质区分开来。例如，人具有身高、体重、胖瘦以及肤色等属性，还可以进行吃饭、学习、走路等能动性的活动，这些活动可以说是人具有的功能。可以把人看作程序中的一个类，那么人的身高可以比作类中的属性，走路可以比作类中的方法。也就是说：类是属性和方法的集合，是面向对象编程方式的核心和基础，通过类可以将零散的用于实现某项功能的代码进行有效的管理。

### 15.1.2 对象

类是具备某项功能的抽象模型，实际应用中还需要对类进行实例化，这样就引入了对象的概念。对象是类进行实例化后的产物，是一个实体。仍然以人为例，“黄种人是人”这句话没有错误，但反过来说“人是黄种人”，这句话一定是错误的。因为除了黄种人，还有黑人、白人等。那么“黄种人”就是“人”这个类的一个实例对象。可以这样理解对象和类的关系：对象实际上就是“有血有肉的、能摸得到看得着的”一个类。

### 15.1.3 面向对象编程的三大特点

面向对象的编程方式具有封装性、继承性和多态性三大特点。封装可以有效地避免程序设计时代码耦合，继承可以有效地利用代码资源，多态性可以有效地提高代码的重用性。

#### 1. 封装性

封装性，也可以称为信息隐藏，就是将一个类的使用和实现分开，只保留有限的接口（方法）与外部联系。对于用到该类的开发人员，只要知道这个类该如何使用即可，而不用去关心这个类是如何实现的。这样做可以让开发人员更好地把精力集中起来专注于别的事情，同时也避免了程序之间的相互依赖而带来的不便。

#### 2. 继承性

继承性就是派生类（子类）自动继承一个或多个基类（父类）中的属性与方法，并可以重写或添加新的属性或方法。继承这个特性简化了对象和类的创建，增加了代码的可重性。继承分为单继承和多继承，PHP 目前只支持单继承，也就是说，一个子类有且只有一个父类。

### 3. 多态性

多态性是指同一个类的不同对象，使用同一个方法可以获得不同的结果，这种技术称为多态性。多态性增强了软件的灵活性和重用性。

## 15.2 类与对象

在面向对象的编程语言中，类是对对象的抽象，在类中可以定义对象的属性和方法的描述；对象是类的实例，类只有被实例化后才能被使用。

### 15.2.1 定义类

#### ■ 视频讲解：光盘\TM\Video\15\定义类.exe

在 PHP 中，使用关键字 class 加类名的方式定义类，然后用大括号包裹类体，在类体中定义类的属性和方法。类的格式如下：

```
<?php
class MyClass{
    private $var;           //定义属性
    public function fun()    //定义方法
    {
        ...
    }
}
?>
```

MyClass 为类的名称，两个大括号中间的部分就是类的全部内容。在 PHP 5.0 以前的版本中，使用关键字 var 声明类属性；而在 PHP 5.0（包括 5.0 版）以后的版本中，引入了 private、protected 和 public 关键字，分别用来定义私有、保护和公有成员，这就可以使程序更加安全。类中定义成员函数的方法与定义普通函数的方法相同，而且还可以用关键字 private、protected 和 public 对成员函数的访问权限进行限定。

**注意：**一个类，即一对大括号之间的全部内容都要在一段代码段中，即一个“<?php ... ?>”之间，不能分割成多块儿，例如：

```
<?php
class MyClass{
    //...
}
<?php
    //...
}
?>
```

这种格式是不允许的。

## 15.2.2 实例化类

### 视频讲解：光盘\TM\Video\15\实例化类.exe

对类定义完成后并不能直接使用，还需要对类进行实例化，即声明对象。PHP 中使用关键字 new 来声明一个对象。实现过程如下：

**对象名 = new 类名(var0, var1 ..., varn)"**

在对类进行实例化时，需要为类的构造方法指定参数。

**例 15.1 定义学生类（Student 类），在类体中定义的成员变量\$name 和 \$age 分别用来表示学生姓名和年龄；\_\_construct() 为类的构造方法，用于为类的成员变量初始化；最后定义方法 getNameAndAge() 来输出学生信息，代码如下：**（实例位置：光盘\TM\Instances\15\1）

```
<?php
class Student
{
    private $name; // 定义类的属性
    private $age;

    public function __construct($name, $age) // 定义构造方法
    {
        $this->name = $name;
        $this->age = $age;
    }

    public function getNameAndAge() // 定义类的成员方法
    {
        return "学生" . $this->name . "今年" . $this->age . "周岁";
    }
}

$student = new Student("小明", 15); // 类的实例化
echo $student->getNameAndAge(); // 调用类的成员方法
?>
```

运行上述代码，将在页面中输出：学生小明今年 15 周岁

## 15.2.3 使用对象

### 视频讲解：光盘\TM\Video\15\使用对象.exe

对类定义完成后，需要将其进行实例化后才能使用，对象可以调用类中的公有属性和公有方法，即被关键字 public 修饰的属性和方法。调用方法如下：

对象名-> 属性；	// 调用类的属性
对象名-> 方法；	// 调用类的方法

从上述代码中可以发现，PHP 调用类的属性和方法使用符号“->”而非“.”，其他一些面向对象

的编程语言，如 Java 和 C# 等一般采用点，初学者应该注意，不要混淆。

**例 15.2** 定义类 Book，并在类中定义公有成员 bookName 代表书名，公有方法 getBookName() 用来获取图书名称。对 Book 类进行实例化后分别通过 bookName 属性和 getBookName() 方法在页面中输出图书名称。（实例位置：光盘\TM\Instances\15\2）

```
<?php
class Book
{
    public $bookName; //定义属性

    public function __construct($bookName) //定义构造方法
    {
        $this->bookName = $bookName;
    }

    public function getBookName() //定义 getBookName()方法获取图书名称
    {
        return $this->bookName;
    }
}

$book = new Book("《PHP 从基础到项目实战》");
echo $book->bookName; //对类进行实例化
echo "<br>"; //通过属性输出图书名称
echo $book->getBookName(); //通过 getBookName()方法输出图书的名称
?>
```

运行上述代码将在页面中输出如下内容：

《PHP 从基础到项目实战》

《PHP 从基础到项目实战》

如果将类体中的属性或方法声明为私有或保护的成员，再次运行本实例将提示错误信息。所以从上面的实例可知，PHP 的属性只能直接调用类中的公有属性和公有方法。

#### 15.2.4 成员变量

##### 视频讲解：光盘\TM\Video\15\成员变量.exe

在类中所定义的变量，称为成员变量（也有称为属性或字段的）。成员变量用来保存信息数据，或与成员方法进行互交来实现某项功能。

定义成员变量的格式一般为：

变量声明符 成员变量名

**说明：** 变量声明符可以使用 public、private、protected、static 和 final 中的任意一个。

访问成员变量和访问成员方法是一样的，只要把成员方法换成成员变量即可，格式为：

对象名 -> 成员变量

例 15.3 在 MyClass 类中声明一个成员变量\$class\_name，接着为这个变量声明两个成员方法 \$setClassName() 和 \$getClassName()。实例化 MyClass 类并为类的构造函数传递参数“Hello PHP”，最后输出结果。实例代码如下：（实例位置：光盘\TM\Instances\15\3）

```
<?php
class MyClass{
    public $class_name;
    function setClassName($name){
        $this->class_name = $name;
    }
    function getClassName(){
        return $this->class_name;
    }
}
$c_book = new MyClass();
$c_book->setClassName("Hello PHP");
echo $c_book->class_name."<br>";
echo $c_book->getClassName();
?>
```

运行上述代码，将在页面中输出如下内容：

Hello PHP

Hello PHP

## 15.2.5 成员函数

### 视频讲解：光盘\TM\Video\15\成员方法.exe

在类体中所定义的函数称为成员函数或成员方法，主要是为了实现某项功能，在 PHP 4.x 中定义成员函数的方法与定义普通函数的方法相同，而 PHP 5.x 中可以为成员方法设置访问权限修饰符，从而可以有效提高代码的逻辑性和安全性。PHP 5.x 中定义类的成员函数的语法格式如下：

```
访问权限修饰符 function 成员函数名(var0, var1...,varn)
{
    //函数体
}
```

例 15.4 定义类 MyClass 并在类体中定义属性\$name，定义公有方法 setName() 和 getName() 分别用来为属性\$name 赋值和获取\$name 的值，实例代码如下：（实例位置：光盘\TM\Instances\15\4）

```
<?php
class MyClass
{
    private $name;
    public function setName($name)
    {
        $this->name = $name;
    }
    public function getName()
    {
        return $this->name;
    }
}
```

```

    }
    public function getName() //获取$name 属性值
    {
        return $this->name;
    }
}
$myClass = new MyClass(); //对类进行实例化
$myClass->setName("计算机");
echo $myClass->getName(); //输出$name 属性的值
?>

```

运行本实例，将在页面中输出如下内容：

计算机

### 15.3 类的权限修饰符

PHP 5.0 以后的版本中，可以对类中属性和方法的访问权限进行限定，将类中的成员分为私有成员、保护成员和公有成员。这样使得 PHP 的面向对象的编程方式更加人性化，开发的程序安全性也有明显提高。可以说，PHP 5.0 的面向对象的新特性是 PHP 编程方式的又一个新起点，是 PHP 进一步成熟的体现。

#### 15.3.1 public（公有成员）

##### ■ 视频讲解：光盘\TM\Video\15\公有成员.exe

顾名思义，公有成员即可以被公开使用、不必隐藏的成员。公有成员使用关键字 `public` 作为修饰符，不仅可以在类体内被调用，而且可以被子类和类的对象所调用。

**例 15.5** 定义类 `Book`，并在类体内定义公有变量 `$name` 以及公有方法 `setName()` 和 `getName()`，这两个方法分别用来为成员变量赋值和提取成员变量的值，然后定义类 `LBook`，使 `LBook` 继承类 `Book`，再通过子类 `LBook` 的对象访问父类 `Book` 的对象。代码如下：（实例位置：光盘\TM\Instances\15\5）

```

<?php
class Book{
    public $name = 'computer'; //声明公有成员变量$name
    public function setName($name){ //设置公有变量方法
        $this -> name = $name;
    }
    public function getName(){ //读取公有变量方法
        return $this -> name;
    }
}
class LBook extends Book{ //Book 类的子类
}
$lbook = new LBook(); //实例化对象
$lbook -> setName("PHP 从基础到项目实战");

```

```
echo $lbook->getName();
?>
```

运行上述实例，将在页面中输出如下内容：

### PHP 从基础到项目实战

从运行结果可以判断，子类可以继承父类的公有成员，在类体外也可以通过对象访问类中的公有成员。

## 15.3.2 private（私有成员）

### ■ 视频讲解：光盘\TM\Video\15\私有成员.exe

被 `private` 关键字修饰的变量和方法，只能在所属类的内部被调用和修改，不可以在类体外和类的子类中所访问。

**例 15.6 对私有变量\$name 的修改与访问，只能通过调用成员方法来实现。如果直接调用私有变量，将会发生错误。实例代码如下：**（实例位置：光盘\TM\Instances\15\6）

```
<?php
class Book{
    private $name = 'computer'; //声明私有变量$name
    public function setName($name){ //设置私有变量方法
        $this->name = $name;
    }
    public function getName(){ //读取私有变量方法
        return $this->name;
    }
}
class LBook extends Book{ //Book 类的子类
}
$lbook = new LBook(); //实例化对象
echo '正确操作私有变量的方法：'; //正确操作私有变量
$lbook->setName("PHP 从入门到精通");
echo $lbook->getName(); //直接操作私有变量的结果： //错误操作私有变量
echo Book::$name;
?>
```

运行结果如图 15.1 所示。

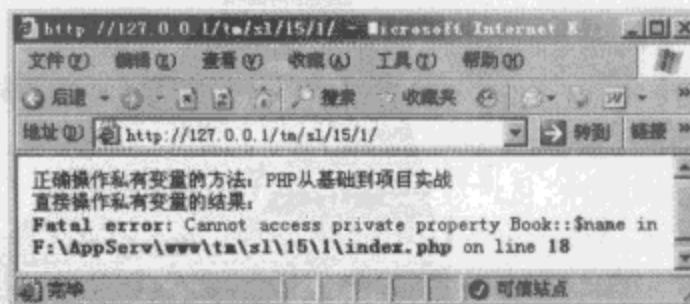


图 15.1 `private` 关键字的使用

### 15.3.3 protected (保护成员)

观看视频讲解：光盘\TM\Video\15\保护成员.exe

`private` 关键字可以将数据完全隐藏起来，除了在本类外，其他地方都不可以调用，当然子类也不可以。但对于有些变量希望子类能够调用，但对其他类来说，还要做到封装。这时就可以使用关键字 `protected`。被 `protected` 修饰的类成员，可以在本类和子类中被调用，其他地方则不可以被调用。

**例 15.7** 首先声明一个 `protected` 变量，然后使用子类中的方法调用一次，最后在类外直接调用一次，查看运行结果。实例代码如下：（实例位置：光盘\TM\Instances\15\7）

```
<?php
class Book{
    protected $name = 'computer'; //声明保护变量$name
}
class LBook extends Book{ //Book 类的子类
    public function showMe(){
        echo '对于 protected 修饰的变量，在子类中是可以直接调用的，如$name = '.$this->name;
    }
}
$lbook = new LBook(); //实例化对象
$lbook->showMe();
echo '<p>但在其他的地方是不可以调用的，否则：</p>'; //对私有变量进行操作
$lbook->name = 'history';
?>
```

运行结果如图 15.2 所示。

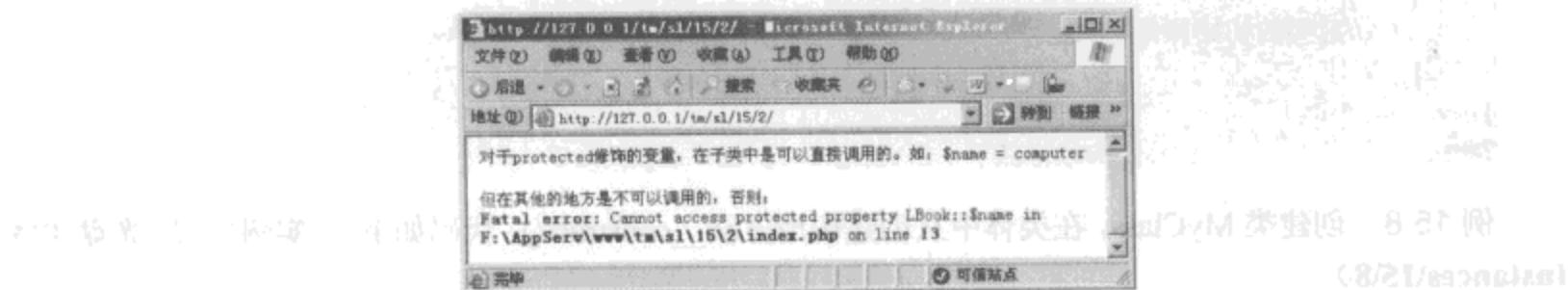


图 15.2 `protected` 关键字

**说明：**虽然 PHP 中没有对修饰变量的关键字做强制性的规定和要求，但从面向对象的特征和设计方面来考虑，一般使用 `private` 或 `protected` 关键字来修饰变量，以防止变量在类外被直接修改和调用。

## 15.4 构造函数

PHP 中对类进行实例化的同时会调用类的构造函数进行一些初始操作，如为类中的成员变量赋初值。可以说构造函数是类的入口。

### 15.4.1 创建构造函数

视频讲解：光盘\TM\Video\15\创建构造函数.exe

在 PHP 4.x 中构造函数是与类名同名的函数，而在 PHP 5.x 以后的版本中，构造函数统一使用名为 `_construct()` 的函数。由于 PHP 5.x 向下兼容 PHP 4.x，所以在 PHP 5.x 中使用与类名同名的构造函数也是可以的，不过开发人员如果使用 5.0 以上版本的 PHP，建议使用 `_construct()` 作为构造函数，这样可以保证编码风格的统一。

在 PHP 4.x 中，构造函数的定义格式如下：

```
<?php
class MyClass
{
    function MyClass(var0, var1...,varn)
    {
        //对类的初始操作
    }
}
?>
```

在 PHP 5.0 以上版本中，使用 `_construct()` 方法作为类的构造方法。

```
<?php
class MyClass
{
    public function __construct(var0, var1...,varn)
    {
        //对类的初始操作
    }
}
?>
```

例 15.8 创建类 `MyClass`，在类体中只创建构造方法 `_construct()`，代码如下：（实例位置：光盘\TM\Instances\15\8）

```
<?php
class MyClass
{
    public function __construct() //构造函数
    {
        echo "对类的初始操作";
    }
}

$myClass = new MyClass(); //类的实例化
?>
```

运行上述代码，将在页面中输出“对类的初始操作”字样。可以发现，使用 `new` 关键字对类进行

实例化操作后，并未进行其他操作就执行了构造方法中的代码，这说明构造方法是在对类进行实例化操作过程中自动调用的。

### 15.4.2 调用父类构造函数

#### 视频讲解：光盘\TM\Video\15\调用父类构造函数.exe

利用子类不仅可以调用父类的属性和普通方法，而且也可以通过调用父类的构造方法为父类初始化，如下所示。

```
parent::__construct();
```

与调用其他属性和方法一样，子类也是通过 `parent` 关键字调用父类的构造方法的。

**例 15.9** 创建 `MyClass` 作为父类，在父类中定义构造方法 `__construct()`，定义子类 `ChildClass`，在子类的构造方法中，使用 `parent` 关键字调用父类的构造方法。（实例位置：光盘\TM\Instances\15\9）

```
<?php
class MyClass //父类
{
    public function __construct() //父类构造函数
    {
        echo "父类的构造函数";
    }
}

class ChildClass extends MyClass //子类
{
    public function __construct() //子类构造函数
    {
        parent::__construct(); //调用父类构造函数
    }
}

$childClass = new ChildClass(); //对类的实例化
?>
```

运行上述实例，将在页面中输出“父类的构造函数”字样。

## 15.5 析构函数

#### 视频讲解：光盘\TM\Video\15\析构函数.exe

析构函数与构造函数的作用是相反的，析构函数是在对象被注销时被调用的。尽管 PHP 在请求结束后会确保资源被释放，但析构函数还是有着非常重要的作用，如释放特定资源、记录日志等。PHP 中使用 `__destruct()` 方法作为析构方法。

```
访问权限修饰符 function __destruct()
{
    //函数体
}
```

**例 15.10** 定义类 MyClass，在类中分别定义构造函数\_\_construct()、普通成员函数 fun()和析构函数\_\_destruct()，并通过该类的对象调用类中的 fun()方法，如下所示。（实例位置：光盘\TM\Instances\15\10）

```
<?php
class MyClass
{
    public function __construct()          //构造函数
    {
        echo "调用构造函数<br>";
    }

    public function fun()                 //普通成员函数
    {
        echo "调用普通成员函数<br>";
    }

    public function __destruct()          //析构函数
    {
        echo "调用析构函数<br>";
    }
}

$myClass = new MyClass();                //对类进行实例化
$myClass->fun();                      //调用 fun()方法
?>
```

运行上述代码，结果如图 15.3 所示。从运行结构中可以看出，在类的整个周期中，最先调用构造函数，然后是普通成员函数，最后为析构函数。

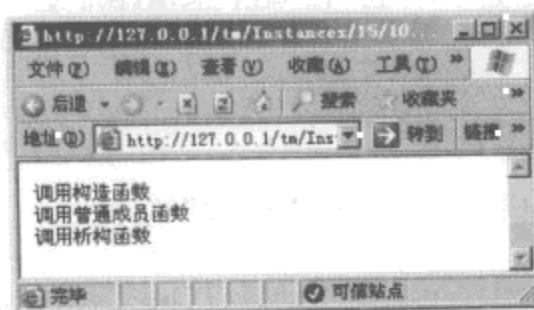


图 15.3 类中函数执行顺序

## 15.6 使用\$this 变量访问方法和属性

■**视频讲解：**光盘\TM\Video\15\使用\$this 变量访问方法和属性.exe

PHP 面向对象的编程方式中，在对象中的方法执行时会自动定义一个\$this 变量，这个变量表示对

对象本身的引用。使用 \$this 变量可以引用该对象的其他方法和属性，并使用 “->” 作为连接符，如下所示。

```
$this->属性; //注意属性名前没有“$”
$this->方法;
```

在使用 \$this 引用对象自身的方法时，直接加方法名并为方法指定参数即可，如果引用的是类的属性，一定注意不要加“\$”。

**例 15.11** 定义类 MyClass，并在类中定义\$name 属性，使用该类的构造函数为\$name 属性初始化，定义 getName()方法获取\$name 属性，最后定义 printName()方法，在该方法中通过 \$this 关键字调用 getName()方法输出\$name 属性的值。（实例位置：光盘\TM\Instances\15\11）

```
<?php
class MyClass
{
    private $name; //定义$name 属性

    public function __construct($name) //定义构造函数
    {
        $this->name = $name;
    }
    public function getName() //获取$name 属性
    {
        return $this->name;
    }
    public function printName() //打印$name 属性
    {
        echo $this->getName();
    }
}
$myClass = new MyClass("I Like PHP"); //实例化 MyClass 类
$myClass->printName(); //调用类的 printName()方法
?>
```

运行上述实例，将在页面中输出 “I Like PHP” 字样。

**◆ 注意：**初学者应该明确 self、parent 和 \$this 关键字的区别，self 关键字用来指向当前的类，而且该关键字通常用来访问类的静态成员、方法和常量。parent 关键字用于指向父类，所以可以使用该关键字调用父类的属性和方法。\$this 变量用来在类体内调用自身的属性和方法。

## 15.7 static 关键字

**◆ 视频讲解：**光盘\TM\Video\15\static 关键字.exe

和其他一些面向对象的编程语言一样，不是所有的变量（方法）都要通过创建对象来调用，还可以通过给变量（方法）加上 static 关键字来直接调用。调用静态成员的格式为：

## 关键字::静态成员

关键字可以是如下两种。

- self: 在类内部调用静态成员时所使用。
- 静态成员所在的类名: 在类外调用类内部的静态成员时所使用。

**◆ 注意:** 在静态方法中, 只能调用静态变量, 而不能调用普通变量。而普通方法则可以调用静态变量。

使用静态成员, 除了可以不需要实例化对象, 另一个作用就是在对象被销毁后, 仍然保存被修改的静态数据, 以便下次继续使用。这个概念比较抽象, 下面来看一个实例。

**例 15.12** 首先声明一个静态变量\$num, 声明一个方法, 在方法的内部调用静态变量, 然后给变量加 1。依次实例化这个类的两个对象, 并输出方法。可以发现两个对象中的方法返回的结果有了一点联系。直接使用类名输出静态变量, 看看有什么效果。实例代码如下: (实例位置: 光盘\TM\Instances\15\12)

```
<?php
class Book{                                //Book 类
    static $num = 0;                         //声明一个静态变量$num, 初值为 0
    public function showMe(){                  //声明一个方法
        echo '您是第'.self::$num.'位访客';   //输出静态变量
        self::$num++;                          //将静态变量加 1
    }
}
$book1 = new Book();                        //实例化对象$book1
$book1 -> showMe();                      //调用对象$book1 的 showMe()方法
echo "<br>";
$book2 = new Book();                        //实例化对象$book2
$book2 -> showMe();                      //调用对象$book2 的 showMe()方法
echo "<br>";
echo '您是第'.Book::$num.'位访客';       //直接使用类名调用静态变量
?>
```

运行结果如图 15.4 所示。

如果将程序代码中的静态变量改为普通变量, 如 “private \$num = 0;”, 结果就不一样了。读者可以自己尝试一下。

**说明:** 静态成员不需实例化对象, 当类第一次被加载时就已经分配了内存空间, 所以直接调用静态成员的速度要快一些。但如果静态成员声明过多, 空间一直被占用, 反而会影响系统的功能。

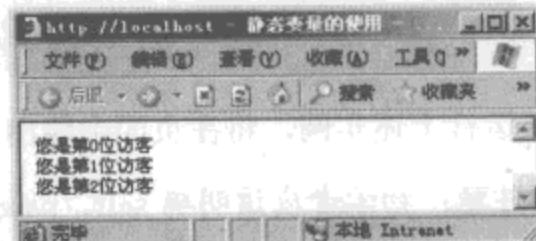


图 15.4 静态变量的使用

## 15.8 final 关键字

**◆ 视频讲解:** 光盘\TM\Video\15\final 关键字.exe

“final”的中文含义是最终的、最后的。被关键字 final 修饰过的类和方法就是“最终的类和方法”。

也就是说，用 final 关键字修饰的类不能被继承，用 final 关键字修饰的方法不能被重写。

如果有一个类的格式为：

```
final class MyClass{
//...
}
```

则说明其他类不能再使用 extends 关键字继承该类。

如果有一个方法使用 final 关键字修饰，如下所示：

```
final function method_name()
```

说明该方法在所在类的子类中不可以被重写。

**例 15.13** 为 final 类 MyClass 生成一个子类 ChildClass。可以看到程序报错，无法执行。实例代码如下：（实例位置：光盘\TM\Instances\15\13）

```
<?php
final class MyClass{
    function __construct(){
        //...
    }
}
class ChildClass extends MyClass{
    static function exam(){
        //...
    }
}
$childClass = new ChildClass();
?>
```

结果为：**Fatal error: Class ChildClass may not inherit from final class (MyClass) in E:\AppServ\www\TM\sl\15\13\index.php on line 11**

## 15.9 抽象类

 视频讲解：光盘\TM\Video\15\抽象类.exe

抽象类是一种不能被实例化的类，只能作为其他类的父类来使用。抽象类使用 abstract 关键字来声明，格式为：

```
abstract class AbstractName{
...
}
```

抽象类和普通类相似，都包含成员变量、成员方法。两者的区别在于：抽象类至少要包含一个抽象方法。抽象方法没有方法体，其功能的实现只能在子类中完成。抽象方法也是使用 abstract 关键字来修饰的，格式为：

```
abstract function abstractName();
```

注意：在抽象方法后面要有分号“;”。

抽象类和抽象方法主要应用于复杂的层次关系中，这种层次关系要求每一个子类都包含并重写某些特定的方法。下面举一个例子。中国的美食是多种多样的，有粤菜、鲁菜、川菜、吉菜等菜系。每种菜系使用的都是煎、炒、烹、炸等手法，只是在具体的步骤上各有不同。如果把中国美食当作一个大类 Cate，下面的各大菜系就是 Cate 的子类，而煎炒烹炸则是每个类中都有的方法。每个方法在子类中的实现都是不同的，在父类中无法规定。为了统一规范，不同子类的方法要有一个相同的方法名：decoc (煎)、stir\_fry (炒)、cook (烹)、fry (炸)。

**例 15.14** 实现一个商品抽象类 MyObject，该抽象类包含一个抽象方法 service()。为抽象类生成两个子类 MyBook 和 MyComputer，分别在两个子类中实现抽象方法。最后实例化两个对象，调用实现后的抽象方法，输出结果。（实例位置：光盘\TM\Instances\15\14）

```
<?php
/* 抽象类 MyObject */
abstract class MyObject{
/* 抽象方法 service */
    abstract function service($getName,$price,$num);
}
/* 声明抽象类的子类 MyBook */
class MyBook extends MyObject{
/* 实现抽象方法 service */
    function service($getName,$price,$num){
        echo '您购买的商品是'.$getName.', 该商品的价格是: '.$price.' 元。';
        echo '您购买的数量为: '.$num.' 本。';
        echo '如发现缺页、损坏请在 3 日内更换。';
    }
}
/* 声明另一个抽象类的子类 MyComputer */
class MyComputer extends MyObject{
/* 实现抽象方法 service */
    function service($getName,$price,$num){
        echo '您购买的商品是'.$getName.', 该商品的价格是: '.$price.' 元。';
        echo '您购买的数量为: '.$num.' 台。';
        echo '如发生非人为质量问题, 请在 3 个月内更换。';
    }
}
$book = new MyBook(); //实例化对象$book
$computer = new MyComputer(); //实例化对象$computer
$book -> service('《PHP 从入门到精通》',85,3); //调用$book 对象的 service 方法
echo '<p>';
$computer -> service('XX 笔记本',8500,1); //调用 computer 对象的 service 方法
?>
```

运行结果如图 15.5 所示。

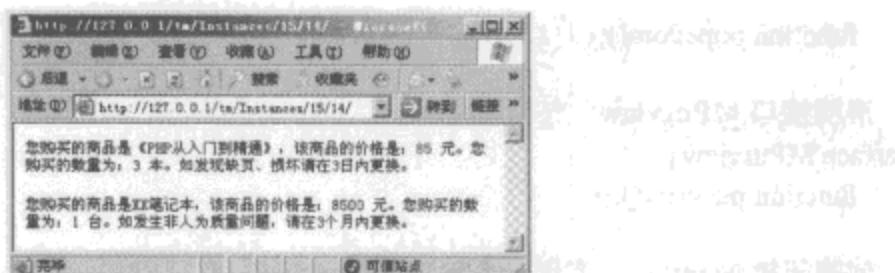


图 15.5 抽象类

## 15.10 接口的使用

### 视频讲解：光盘\TM\Video\15\接口的使用.exe

继承特性简化了对象、类的创建，增加了代码的可重性。但 PHP 只支持单继承。如果想实现多重继承，就要使用接口。PHP 可以实现多个接口。

接口类通过 `interface` 关键字来声明，并且类中只能包含未实现的方法和一些成员变量。格式如下：

```
interface InterfaceName{
    function interfaceName1();
    function interfaceName2();
    ...
}
```

**注意：**不要用 `public` 以外的关键字来修饰接口中的类成员，对于方法，不写关键字也可以。这是由接口类自身的天性决定的。

子类是通过 `implements` 关键字来实现接口的，如果要实现多个接口，那么每个接口之间使用逗号“，”连接。而且所有未实现的方法需要在子类中全部实现，否则 PHP 将会出现错误。格式如下：

```
class SubClass implements InterfaceName1, InterfaceName2{
    function interfaceName1(){
        //功能实现
    }
    function interfaceName2(){
        //功能实现
    }
    ...
}
```

**例 15.15** 首先声明两个接口 `MPopedom` 和 `MPurview`，接着声明两个类 `Member` 和 `Manager`，其中 `Member` 类继承了 `MPopedom` 接口，`Manager` 继承了 `MPopedom` 和 `MPurview` 接口。分别实现各自的成员方法后，实例化两个对象 `$member` 和 `$manager`。最后调用实现后的成员方法。实例代码如下：（实例位置：光盘\TM\Instances\15\15）

```
<?php
/* 声明接口 MPopedom */
interface MPopedom{
```

```

        function popedom();
    }
/* 声明接口 MPurview */
interface MPurview{
    function purview();
}
/* 创建子类 Member，实现一个接口 MPurview */
class Member implements MPurview{
    function purview(){
        echo '会员拥有的权限。';
    }
}
/* 创建子类 Manager，实现多个接口 MPurview 和 MPopedom */
class Manager implements MPurview,MPopedom{
    function purview(){
        echo '管理员拥有会员的全部权限。';
    }
    function popedom(){
        echo '管理员还有会员没有的权限';
    }
}
$member = new Member(); //类 Member 实例化
$manager = new Manager(); //类 Manager 实例化
$member -> purview(); //调用$member 对象的 purview 方法
echo '<p>';
$manager -> purview(); //调用$manager 对象的 purview 方法
$manager ->popedom(); //调用$manager 对象的 popedom 方法
?>

```

运行结果如图 15.6 所示。

通过上面的实例，读者可能已经发现，抽象类和接口实现的功能十分相似。抽象类的优点是可以在抽象类中实现公共的方法，而接口则可以实现多继承。什么时候使用抽象类，什么时候使用接口，就要看具体实现了。

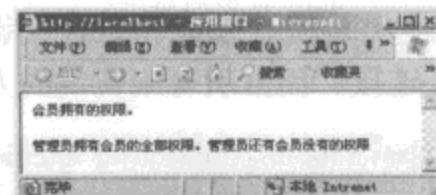


图 15.6 应用接口

## 15.11 实战

### 15.11.1 面向对象的用户登录身份验证

**视频讲解：**光盘\TM\Video\15\面向对象的用户登录身份验证.exe

利用用户身份验证类来实现登录用户的身份验证，不仅可以为以后的维护工作带来很大方便，而且可以提高程序的安全性。

**例 15.16 编写用户登录身份验证类对用户的用户名和密码进行验证。（实例位置：光盘\TM\Instances\15\16）**

本实例的开发步骤如下：

(1) 建立用户登录表单, 表单及表单元素如表15.1所示。

表15.1 用户登录表单所涉及的表单及表单元素

元素类型	元素名称	属性设置	说明
表单	form1	name="form1" method="post" action="index.php"	表单
文本域	username	name="username" size="22" class="inputcss"	用户名
密码域	password	name="userpwd" size="22" class="inputcss"	密码
提交按钮	submit	name="submit" value="登录" class="buttoncss"	提交按钮

(2) 接收用户身份录入表单提交的用户名和密码, 代码如下。

```
$username=$_POST[username]; //接收提交的用户名
$userpwd=$_POST[userpwd]; //接收提交的密码
if(trim($username)==""||trim($userpwd)==(""))
{
    echo "<script>alert('请输入用户名或用户密码!');history.back();</script>";
    exit;
}
```

(3) 建立密码验证类, 对用户登录身份进行验证, 代码如下。

```
class chk //定义密码验证类
{
    private $name; //定义用户名属性
    private $pwd; //定义密码属性
    public function __construct($x,$y) //构造函数, 对类的属性初始化
    {
        $this->name=$x;
        $this->pwd=$y;
    }
    public function chkuser() //验证用户身份
    {
        include_once("conn.php");
        $rs=new com("adodb.recordset"); //创建记录集对象
        $rs->open("select * from tb_user where username='".$this->name."' and userpwd='".$this->pwd."'", $conn,3,1);
        if($rs->eof || $rs->bos)
        {
            echo "<script>alert('对不起, 密码或用户名错误!');history.back();</script>";
            exit;
        }
        else
        {
            echo "<script>alert('恭喜您登录成功!');history.back();</script>";
            exit;
        }
    }
}
```

(4) 对用户身份验证类进行实例化, 并调用类的验证方法进行验证, 代码如下。

```
$chk1=new chk($username,$userpwd);           //对密码验证类进行实例化
$chk1->chkuser();                          //调用 chkuser()方法验证用户身份
```

运行本实例，如图 15.7 所示。如果登录用户输入正确的用户名和密码，则系统给出登录成功信息，反之系统给出登录失败信息。



图 15.7 用户登录

### 15.11.2 面向对象的数据分页显示

#### 视频讲解：光盘\TM\Video\15\面向对象的数据分页显示.exe

**例 15.17 面向对象的数据分页显示。**虽然 PHP 采用面向过程的方式实现的数据分页功能显示相对容易，但是维护却很复杂，同时也不利于重复应用。本实例介绍一种通过面向对象的方式实现的数据分页功能，将分页的方法写入到一个类中，直接调用这个分页类来实现分页的功能。（实例位置：光盘\TM\Instances\15\17）

分页类的代码如下：

```
<?php
class page{
    private $pagesize;          //保存每页最多显示的记录数
    private $page;              //保存当前的页数
    private $pagecount;         //保存总页数
    private $total;             //保存总记录数
    private $conn;               //保存数据库连接句柄
    public function __construct($pagesize,$page){ //构造函数，实现对类体中部分属性的初始化
        $this->pagesize=$pagesize;
        $this->page=$page;
    }
    public function listInfo(){           //定义 listInfo()方法显示记录内容
        if($this->page=="") || !is_numeric($this->page)){
            $this->page=1;
        }
        $this->conn=MySQL_connect("localhost","root","root");           //连接数据库服务器
        MySQL_select_db("db_database15",$this->conn);                  //连接数据库
        MySQL_query("set names gb2312");                                //设置编码格式
        $sql=MySQL_query("select count(*) as total from tb_student",$this->conn); //执行查询语句
        $info=MySQL_fetch_array($sql);                                     //获取查询结果
        $this->total=$info[total];
        if($this->total==0) {
            echo "<table width=704 height=20 border=0 align=center cellpadding=0 cellspacing=0>"+
                "<tr>"+
                "<td><div align=center>暂无学生信息！</div></td>"+
                "</tr>"+
        }
    }
}
```

```

        "</table>";
    }else{
        if($this->total % $this->pagesize)==0) {
            $this->pagecount=intval($this->total/$this->pagesize);
        }else{
            if($this->total<=$this->pagesize){
                $this->pagecount=1;
            }else{
                $this->pagecount=ceil($this->total/$this->pagesize);
            }
        }
        $sql=MySQL_query("select * from tb_student order by sno desc limit ".$this->pagesize*($this->page-1).",
        $this->pagesize,$this->conn);
    ?>
    <table width="703" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#4C6404">
    <!--省略部分代码-->
    <?php
        while($info=MySQL_fetch_array($sql)){
    ?>
    <!--省略部分代码-->
    <?php
    }
    echo "</table>";
}
}

public function toPage(){
    //定义 toPage()方法用于显示分页链接
?>
    <table width="704" height="20" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td width="342">&ampnbsp公有学生信息 &ampnbsp<?php echo $this->total;?>&ampnbsp条 &ampnbsp<?php echo $this->pagesize;?>&ampnbsp条&ampnbsp<?php echo $this->page;?>&ampnbsp页/共&ampnbsp<?php echo $this->pagecount;?>&ampnbsp页</td>
            <td width="362"><div align="right">
                <a href=<?php echo $_SERVER["PHP_SELF"]?>?page=1>首页</a>
                <a href=<?php echo $_SERVER["PHP_SELF"]?>?page=<?php
                    if($this->page>1)
                        echo $this->page-1;
                    else
                        echo 1;
                ?>">上一页</a>
                <a href=<?php echo $_SERVER["PHP_SELF"]?>?page=<?php
                    if($this->page<$this->pagecount-1)
                        echo $this->page+1;
                    else
                        echo $this->pagecount;
                ?>">下一页</a>
                <a href=<?php echo $_SERVER["PHP_SELF"]?>?page=<?php echo $this->pagecount;?>">尾页</a>
            </div></td>
        </tr>
    </table>

```

```

<?php
}

public function __destruct(){
    MySQL_close($this->conn);
}

$obj=new page(4,$_GET["page"]);           //实例化对象，使每页显示 4 条记录
?>

```

对分页类进行实例化后即可调用类中的 listInfo()方法和 toPage()方法，分别显示数据和显示分页链接，代码如下。

```

<?php
    $obj->listInfo();          //调用 listInfo()方法显示记录内容
    $obj->toPage();            //调用 toPage()方法显示分页链接
?>

```

运行结果如图 15.8 所示。

编号	姓名	性别	年级	地址
010603300746	罗一	16	三年三班	吉林省辽源市
010603300745	柳一	15	三年三班	吉林省四平市
010603300743	张一	15	三年三班	吉林省四平市
010603300742	钱一	16	三年三班	吉林省长春市

图 15.8 面向对象的数据分页

## 15.12 本 章 小 结

本章主要对面向对象的基本概念和类的特性进行了阐述，并详细介绍了类的创建方式及常用关键字的使用方法。通过本章的学习，读者能够具有 PHP 面向对象的编程思想，并具备编写程序中的基本类的能力。如果读者能够在此基础上进一步学习，定能使自己的编程能力有新的突破和提高。

## 15.13 实战练习

1. PHP 向页面中输出内容时，需要对 HTML 标记进行转义，如果每个输出模块都编写转义代码，则不仅会使程序的开发效率降低，而且不利于代码重用，所以可以将转义代码封装成类，在需要转义 HTML 标记的模块调用该类。（答案位置：光盘\TM\Instances\15\18）
2. 在开发数据库管理程序时，建立一个专门的数据库连接类连接数据库。（答案位置：光盘\TM\Instances\15\19）

# 第16章

小站系列教材

## PHP 加密技术

( 视频讲解: 8分钟)

数据加密在人们生活中的地位越来越重要，尤其是对于在网络上发生的大量交易和传输的大量数据。

所谓数据加密就是将易懂的资料转换为不易懂资料的过程，解密则是将不易懂的资料转换为原来易懂资料的过程。不易懂的资料被称作密码，易懂的资料被称作明码。

数据的加/解密都需要一定的算法，这些算法可以非常的简单，如著名的凯撒码；但当前的加密算法要相对复杂得多，其中一些利用现有的方法甚至是无法破译的。而对于普通的开发者来说，可以使用一些已有的、比较有名气的加密算法，如MD5、SHA等，也可以自己创建加密函数。

通过阅读本章，可以学习到：

- 加密的概念和简单分类
- PHP 内置加密函数
- 单项加密函数
- PHP 加密扩展库
- 双向加密函数
- MD5 校验码

## 16.1 加密技术简介

数据加密的基本原理就是对明文的文件或数据按某种算法进行处理，使其成为一段不可读的代码（通常称为“密文”），通过这样的途径来达到保护数据不被非法窃取、阅读的目的。在 PHP 中能对数据进行加密的函数主要有 crypt()、md5() 和 sha1()，还有加密扩展库 Mcrypt 和 Mhash。

### 16.1.1 MD5 算法

#### 1. MD5 算法的概念

MD5 的全称是 Message-Digest Algorithm 5(信息-摘要算法)，在 20 世纪 90 年代初由 MIT Laboratory for Computer Science 和 RSA Data Security Inc 的 Ronald L. Rivest 开发，经由 MD2、MD3 和 MD4 发展而来。它的作用是让大容量信息在用数字签名软件签署私人密钥前被“压缩”成一种保密的格式（就是把一个任意长度的字节串转换成一定长度的大整数）。注意这里所说的是“字节串”而不是“字符串”，因为这种变换只与字节的值有关，而与字符集或编码方式无关。

#### 2. MD5 算法的发展历程

Rivest 在 1989 年开发出 MD2 算法。在这个算法中，首先对信息进行数据补位，使信息的字节长度是 16 的倍数。然后，以一个 16 位的检验和追加到信息末尾，并且根据这个新产生的信息计算出散列值。后来，Rogier 和 Chauvaud 发现如果忽略检验和将产生 MD2 冲突。因为 MD2 算法的加密后结果是唯一的。

为了加强算法的安全性，Rivest 在 1990 年又开发出 MD4 算法。MD4 算法同样需要填补信息以确保信息的字节长度加上 448 后能被 512 整除（信息字节长度  $\text{mod } 512 = 448$ ）。然后，一个以 64 位二进制表示的信息的最初长度被添加进来。信息被处理成 512 位迭代结构的区块，而且每个区块要通过 3 个不同步骤的处理。Den Boer 和 Bosselaers 以及其他很快发现了攻击 MD4 版本中第一步和第三步的漏洞。Dobbertin 向大家演示了如何利用一部普通的个人电脑在几分钟内找到 MD4 完整版本中的冲突（这个冲突实际上是一种漏洞，它将导致对不同的内容进行加密却可能得到相同的加密后结果）。毫无疑问，MD4 就此被淘汰掉。

虽然 MD4 算法在安全上存在很大的漏洞，但是它对后来一些信息安全加密算法的开发有着不可忽视的引导作用。除了 MD5 以外，其中比较有名的还有 sha1、ripe-md 以及 haval 等。

1991 年，MD5 算法诞生，它在技术上更趋近成熟，安全性也更高。MD5 在 MD4 的基础上增加了“安全-带子”（safety-belts）的概念，其中信息-摘要的大小和填充的必要条件与 MD4 完全相同。Den Boer 和 Bosselaers 曾发现 MD5 算法中的假冲突（pseudo-collisions），但除此之外就没有其他被发现的加密后结果。

虽然在 MD5 中仍存在假冲突（pseudo-collisions），但是单从 1991 年到 2001 年这 10 年间，竟没有出现替代 MD5 算法的 MD6 或被叫做其他什么名字的新算法这一点，就可以看出这个瑕疵并没有过多影响 MD5 的安全性。因此上面所有这些都不足以成为 MD5 在实际应用中的问题。并且，由于使用 MD5 算法不需要支付任何版权费用，所以在一般的情况下（非绝密应用领域。但即便是应用在绝密领

域内，MD5 也不失为一种非常优秀的中间技术），MD5 都称得上是非常安全的。

### 3. MD5 算法的应用

(1) MD5 的典型应用是对一段信息 (message) 产生信息-摘要 (message-digest)，以防止被篡改。例如，在 UNIX 下有很多软件在下载时都有一个文件名相同，即文件扩展名为.md5 的文件，在这个文件中通常只有一行文本，大致结构如下：

```
md5 (tanajiya.tar.gz) = 0ca175b9c0f726a831d895e269332461
```

这就是 tanajiya.tar.gz 文件的数字签名。MD5 将整个文件看作一个大文本信息，通过其不可逆的字符串变换算法，产生了这个唯一的 MD5 信息摘要。如果在以后传播这个文件的过程中，无论文件的内容发生何种形式的改变（包括人为修改或者下载过程中线路不稳定引起的传输错误等），只要对这个文件重新计算 MD5 时就会发现信息摘要不相同，由此可以确定得到的是一个不正确的文件。如果有另一个第三方的认证机构，用 MD5 还可以防止文件作者的“抵赖”，这就是所谓的数字签名应用。

(2) MD5 还广泛用于加密和解密技术上。例如，在 UNIX 系统中，用户的密码就是以 MD5（或其他类似的算法）经加密后存储在文件系统中。当用户登录时，系统把用户输入的密码计算成 MD5 值，然后再去和保存在文件系统中的 MD5 值进行比较，进而确定输入的密码是否正确。通过这样的步骤，系统在并不知道用户密码明码的情况下就可以确定用户登录系统的合法性。这不但可以避免用户密码被其他具有系统管理员权限的用户知道，而且还在一定程度上增加了密码被破解的难度。

#### 16.1.2 SHA1 算法

SHA (Secure Hash Algorithm, 安全哈希算法) 家族是由美国国家安全局 (NSA) 设计，美国国家标准与技术研究院 (NIST) 发布的一系列密码杂凑函数。正式名称为 SHA 的家族第一个成员发布于 1993 年，人们为它取了一个非正式的名称 SHA-0 以避免与它的后继者混淆。1995 年，SHA-1 算法正式发布。

#### 16.1.3 DES 算法

DES 加密算法是由美国国家标准局 1973 年开始研究的除国防部外其他部门计算机系统的数据加密标准。DES 算法的入口参数有 3 个：Key、Data 和 Mode。其中 Key 为 8 个字节共 64 位，是 DES 算法的工作密钥；Data 是要被加密或解密的数据；Mode 为 DES 的工作模式，有加密和解密两种。

DES 算法的工作原理：如果 Mode 为加密模式，则用 Key 和数据 Data 进行加密，生成 Data 的密文形式作为 DES 的输出结果；如果 Mode 为解密模式，则用 Key 把密文形式的数据 Data 解密，还原为 Data 的明文形式作为 DES 的输出结果。

DES 经常用作银行支付接口、移动通信、SP 的网间通信常用加密算法。在通信网络的两端，双方约定一致的 Key，在通信的源点用 Key 对核心数据进行 DES 加密，然后以密文形式在公共通信网（如电话网）中传输到通信网络的终点，数据到达目的地后，用同样的 Key 对密文数据进行解密，就可以再现明文形式的核心数据。这样就保证了核心数据（如 PIN、MAC 等）在公共通信网中传输的安全性和可靠性。

如果想在 PHP 中使用 DES 算法加密，需要加载 Mcrypt 扩展库。

### 16.1.4 MIME/BASE64 的算法

MIME 是多部分 (multi-part)、多媒体电子邮件和 WWW 超文本的一种编码标准，用于传送如图形、声音和传真等非文本数据。MIME 定义于 RFC1341，可以用 MIMENCODE 方法将二进制数据转换为一种被称为 BASE64 的 ASCII 子集的字符的组合。

MIME/BASE64 的算法很简单，它将字符流顺序放入一个 24 位的缓冲区，缺字符的地方补零。然后将缓冲区截断为 4 个部分，高位在先，每个部分 6 位，用下面的 64 个字符重新表示：“ABCDEFHJKL  
LMNOPQRSTUWVXYZabcdefghijklmnopqrstuvwxyz0123456789+/-”。如果输入只有一个或两个字节，那么输出将用“=” 补足，这样可以隔断附加信息造成编码的混乱。这就是 BASE64。

## 16.2 PHP 内置加密函数

PHP 内置了多种加密函数，这里主要介绍其中的 3 种：crypt() 函数、md5() 函数和 sha1() 函数。

### 16.2.1 使用 crypt() 函数进行加密

crypt() 函数可以完成单向加密功能。语法如下：

```
string crypt(string str[, string salt]);
```

其中，str 参数是需要加密的字符串，salt 参数为加密时使用的干扰串。如果省略第二个参数，则会随机生成一个干扰串。crypt() 函数支持 4 种算法和长度，如表 16.1 所示。

表 16.1 crypt() 函数支持的 4 种算法和 salt 参数的长度

算 法	salt 长度
CRYPT_STD_DES	2-character (默认)
CRYPT_EXT_DES	9-character
CRYPT_MD5	12-character (以\$1\$开头)
CRYPT_BLOWFISH	16-character (以\$2\$开头)

说明：默认情况下，PHP 使用两个字符的 DES 干扰串，如果系统使用的是 MD5，则 PHP 会使用 12 个字符的干扰串。可以通过 CRYPT\_SALT\_LENGTH 变量查看当前所使用的干扰串的长度。

**例 16.1** 首先声明一个字符串变量\$str，赋值为“应用 crypt() 函数进行单向加密！”，然后使用 crypt() 函数进行加密并输出。实例代码如下：（实例位置：光盘\TM\Instances\16\1）

```
<?php
$str = '应用 crypt() 函数进行单向加密!';
// 声明字符串变量$str
echo '加密前$str 的值为：' . $str;
$crypttostr = crypt($str);
// 对变量$str 加密
```

```
echo '<p>加密后$str 的值为: '.$crypttostr; //输出加密后的变量
?>
```

运行结果如图 16.1 所示。

按 F5 键刷新，会发现通过 crypt() 函数对数据进行加密时，每次生成的加密数据都不相同，并且 crypt() 函数是单项加密的，密文不可还原成明码。如果要对加密后的数据进行判断，就需要应用 crypt() 函数的 salt 参数对明文进行加密（即在判断时，对输出的信息再次使用相同的 salt 参数进行加密，对比两次加密后的结果）。

**例 16.2** 实现用户登录的功能，如果登录成功则输出“用户登录成功。”，否则输出“用户登录失败！”，并通过 crypt() 函数对用户名和密码进行加密处理，代码如下：（实例位置：光盘\TM\Instances\16\2）

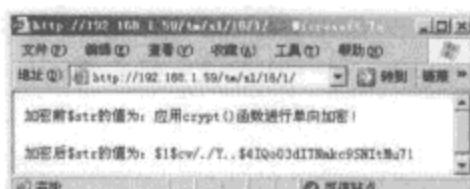


图 16.1 使用 crypt() 函数进行加密

```
<?php
    <!-- 连接数据库 -->
    $conn = mysql_connect("localhost","root","root") or die("数据库连接错误".mysql_error());
    mysql_select_db("db_database16",$conn) or die("数据库访问错误".mysql_error());
    mysql_query("set names gb2312");
?><!-- 提交表单 -->
<form id="form1" name="form1" method="post" action="index.php">
    <input name="username" type="text" id="username" size="24" />
    <input name="password" type="password" id="password" size="25" />
    <input type="image" name="imageField" src="images/bg2.JPG" />
    <input type="image" name="imageField2" src="images/bg1.JPG" onclick="form.reset();return false;" />
</form>
<!--验证表单中提交用户名和密码是否正确-->
<?php
    if(trim($_POST[username])!= "" and trim($_POST[password])!= ""){
        $usr = crypt(trim($_POST[username]),$_POST[username]);
        $pwd = crypt(trim($_POST[password]),$_POST[password]);
        $sql = "select * from tb_user where username = '".$usr."' and password='".$pwd."'";
        $rst = mysql_query($sql,$conn);
        $result=mysql_num_rows($rst);
        if($result>0){
            echo "<font color='red'>用户登录成功。</font>";
        }else{
            echo "<font color='green'>用户登录失败!</font>";
        }
    }else{
        echo "请认真填写用户名和密码！";
    }
?>
```

运行结果如图 16.2 所示。

**◆ 注意：** 在使用 crypt() 函数保护重要的数据时，在默认状态下使用的 crypt() 函数并不是最安全的，该函数只能用在对安全性要求较低的系统中，如果需要较高的安全性能，建议使用下面介绍的加密技术。



图 16.2 使用 crypt() 函数进行数据验证

## 16.2.2 使用 md5() 函数进行加密

md5() 函数使用 MD5 算法，可以把不同长度的数据信息经过一系列的算法计算成一个 128 位的数值。语法如下：

```
string md5 ( string str [, bool raw_output] );
```

其中，字符串 str 为要加密的明文；raw\_output 参数如果设为真，则函数返回一个二进制形式的密文，默认为 false。

## 16.2.3 使用 sha1() 函数进行加密

SHA 算法与 MD5 类似。PHP 提供的 sha1() 函数返回的是一个 160 位的哈希(40 位十六进制字符)，在加密安全性方面要高于 MD5。语法如下：

```
string sha1 ( string str [, bool raw_output] )
```

函数返回一个 40 位的十六进制数，如果参数 raw\_output 为 true，则返回一个 20 位的二进制数。默认 raw\_output 为 false。

**◆ 注意：**sha 后面的 1 是阿拉伯数字（1、2、3）里的 1，不是字母 L，读者一定要注意。

**例 16.3 对一个指定的字符串分别进行 MD5 加密运算和 SHA1 加密运算，代码如下：（实例位置：光盘\TM\Instances\16\3）**

```
<?php
    echo md5('PHP 从基础到项目实战'); //使用 md5()函数加密字符串“PHP 从基础到项目实战”
    echo sha1('PHP 从基础到项目实战'); //使用 sha1()函数加密字符串“PHP 从基础到项目实战”
?>
```

MD5 加密运算和 SHA1 加密运算字符串的对比效果如图 16.3 所示。

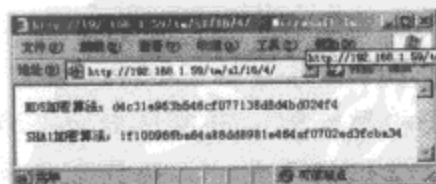


图 16.3 使用 md5() 和 sha1() 函数加密的对比效果

### 16.2.4 使用 base64\_encode() 函数进行编码

base64\_encode() 函数使用 MIME/BASE64 对数据进行编码。语法如下：

```
string base64_encode ( string data )
```

base64\_encode() 函数返回使用 BASE64 对 data 进行编码的数据。设计此种编码是为了使二进制数据可以通过非纯 8-bit 的传输层传输，如电子邮件的主体。

通过 base64\_encode() 函数编码的数据要比原始数据多占用 33% 左右的空间。

**例 16.4 应用 base64\_encode() 函数对一个字符串进行编码，并输出编码后的字符串，代码如下：**  
**(实例位置：光盘\TM\Instances\16\4)**

```
<?php
$str = '吉林省明日科技有限公司';
echo base64_encode($str);
?>
```

结果为：vKrB1sqhw/fI1b/GvLzT0M/euavLvg==

### 16.2.5 使用 base64\_decode() 函数进行解码

base64\_decode() 函数对使用 MIME/BASE64 编码的数据进行解码。语法如下：

```
string base64_decode ( string encoded_data )
```

base64\_decode() 函数对 encoded\_data 进行解码，返回原始数据，失败则返回 false。返回的数据可能为二进制。

**例 16.5 应用 base64\_decode() 函数对例 16.4 中返回的编码数据进行解码，并输出解码后的数据，代码如下：**  
**(实例位置：光盘\TM\Instances\16\5)**

```
<?php
$str = '吉林省明日科技有限公司';
echo base64_decode($str);
?>
```

结果为：吉林省明日科技有限公司

## 16.3 PHP 加密扩展库

PHP 除了自带的几种加密函数外，还有功能更全面的加密扩展库 Mcrypt 和 Mhash。其中，Mcrypt 扩展库可以实现加密解密功能，就是说既能将明文加密，也可以将密文还原；Mhash 扩展库则包含了 MD5 在内的多种 Hash 算法实现的混编函数。

### 16.3.1 Mcrypt 扩展库

#### 1. 安装 Mcrypt 库

单项加密的优势是密文无法还原为明文，即使数据被截获也不会造成资料外泄。但有些情况下还是需要将密文还原成明文的，这时就需要使用双向加密技术。Mcrypt 是一个功能十分强大的加密算法扩展库。在标准的 PHP 安装过程中并没有安装 Mcrypt，但 PHP 的主目录下包含了 libmcrypt.dll 和 libmhash.dll 文件（libmhash.dll 是 Mhash 扩展库，这里一起安装）。

首先将 libmcrypt.dll 和 libmhash.dll 文件复制到本机系统目录 Windows\下，然后在 php.ini 文件中找到`:extension=php_mcrypt.dll`  
`:extension=php_mhash.dll`这两行，将前面的分号“;”去掉，最后重新启动服务器，即可使用这两个扩展库。

#### 2. Mcrypt 库常量

例 16.6 Mcrypt 库支持 20 多种加密算法和 8 种加密模式。读者可以通过 `mcrypt_list_algorithms()` 和 `mcrypt_list_modes()` 函数来查看，实例代码如下：（实例位置：光盘\TM\Instances\16\6）

```
<?php
    $en_dir = mcrypt_list_algorithms(); //函数返回 Mcrypt 支持的加密算法数组
    echo "Mcrypt 支持的算法有：" ;
    foreach($en_dir as $en_value){
        echo $en_value." ";
    }
?>

_____

<?php
    $mo_dir = mcrypt_list_modes(); //函数返回 Mcrypt 支持的算法模式数组
    echo "<p>Mcrypt 支持的加密模式有：" ;
    foreach($mo_dir as $mo_value){
        echo $mo_value." ";
    }
?>
```

运行结果如图 16.4 所示。

这些算法和模式在实际应用中要用常量来表示。写的时候要分别加上前缀 `MCRYPT_` 和 `MCRYPT_MODE_`，例如：

- twofish 算法表示为 `MCRYPT_TWOFISH`。
- cbc 加密模式表示为 `MCRYPT_MODE_CBC`。

#### 3. Mcrypt 应用

例 16.7 使用 Mcrypt 进行加密和解密与使用 `md5()`、`sha1()` 等函数不同，可以直接调用。为了让读者能清楚地了解 Mcrypt 的工作流程，下面通过一个实例来讲解 Mcrypt 是如何工作的，代码如下：（实例位置：光盘\TM\Instances\16\7）

```
<?php
    $str = "PHP 从基础到项目实战 PHP 从入门到精通"; //加密文本
    $key = "key:111"; //密钥
```

```

$cipher = MCRYPT_DES; //密码类型
$modes = MCRYPT_MODE_ECB; //密码模式
$iv = mcrypt_create_iv(mcrypt_get_iv_size($cipher,$modes),MCRYPT_RAND); //初始化向量
echo "加密前: ".$str."<p>";
$str_encrypt = mcrypt_encrypt($cipher,$key,$str,$modes,$iv); //加密函数
echo "加密后: ".$str_encrypt."<p>";
$str_decrypt = mcrypt_decrypt($cipher,$key,$str_encrypt,$modes,$iv); //解密函数
echo "还原: ".$str_decrypt."<p>";
?>

```

运行结果如图 16.5 所示。

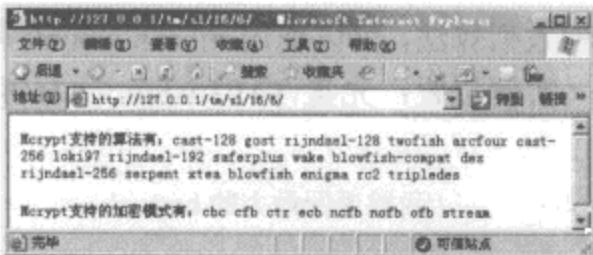


图 16.4 Mcrypt 库支持的加密算法与加密模式

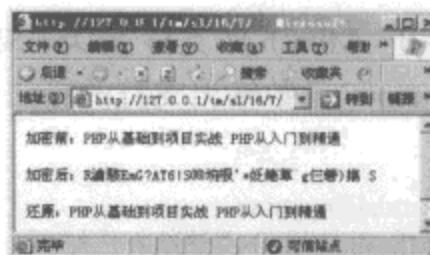


图 16.5 使用 Mcrypt 库加密解密

下面对代码中加粗的函数进行讲解。

**string mcrypt\_create\_iv ( int size [, int source] )**

使用 Mcrypt 进行数据加密、解密之前，首先要创建一个初始化向量（简称 iv）。创建初始化向量需要两个参数：size 指定了 iv 的大小；source 为 iv 的源。source 可以取如下值。

- MCRYPT\_RAND: 系统随机数。
- MCRYPT\_DEV\_RANDOM: 读取目录/dev/random 中的数据（UNIX 系统）。
- MCRYPT\_DEV\_URANDOM: 读取目录/dev/urandom 中的数据（UNIX 系统）。

**int mcrypt\_get\_iv\_size ( string cipher, string mode )**

该函数返回初始化向量（iv）的大小。函数中的两个参数是前面刚介绍过的加密算法（cipher）和算法模式（mode）。

**string mcrypt\_encrypt ( string cipher, string key, string data, string mode [, string iv] )**

初始化向量后，使用 `mcrypt_encrypt` 加密函数对数据进行加密。`mcrypt_encrypt()` 函数的 5 个参数如下：

- cipher: 加密算法。例 16.7 中的变量 `$cipher`，这里的加密算法可以和初始化向量中的加密算法不同。
- key: 密钥。例 16.7 中的变量 `$key`。
- data: 需要加密的数据。例 16.7 中的变量 `$str`。
- mode: 算法模式。例 16.7 中的变量 `$modes`，可以和初始化向量中的模式不一样。
- iv: 初始化向量。例 16.7 中的变量 `$iv`。

**string mcrypt\_decrypt ( string cipher, string key, string data, string mode [, string iv] )**

解密函数 `mcrypt_decrypt` 和加密函数 `mcrypt_encrypt` 的参数几乎是相同的。唯一不同的是参数 `data`，解密函数 `mcrypt_decrypt()` 中的 `data` 为需要解密的数据，而不是原始数据。在例 16.7 中为需要解密的变量 `$str_encrypt`。

 注意：加密函数和解密函数中的 cipher、key 和 mode 参数必须要一致，否则数据不会被还原。

Mcrypt 扩展库包含 30 多个函数，对加密技术感兴趣的读者可以参考 PHP 手册中的详细介绍。

### 16.3.2 Mhash 扩展库

#### 1. 安装 Mhash 库

Mhash 库的安装与 Mcrypt 库的安装类似，读者可以参见 16.3.1 节中 Mcrypt 库的安装过程。

#### 2. Mhash 库常量

**例 16.8** Mhash 库支持 MD5、SHA1、CRC32 等多种散列算法，可以使用 mhash\_count() 和 mhash\_get\_hash\_name() 函数输出支持的算法名称，代码如下：（实例位置：光盘\TM\Instances\16\8）

```
<?php
$num = mhash_count(); //函数返回最大的 hash id
echo "Mhash 库支持的算法有：" ;
for($i = 0; $i <= $num; $i++){
    echo $i."=>".mhash_get_hash_name($i)." " ; //输出每一个 hash id 的名称
}
?>
```

运行结果如图 16.6 所示。

如果在实际应用中使用上面的常量，需要在算法名称前面加上 MHASH\_ 前缀，如 CRC32 表示为 MHASH\_CRC32。

#### 3. Mhash 应用

相比 Mcrypt 扩展库的 30 多个函数，Mhash 库中只有 5 个函数，除了上面使用到的两个函数外，下面介绍另外 3 个函数。

**mhash\_get\_block\_size()** 函数

语法：

```
int mhash_get_block_size ( int hash )
```

该函数返回指定的 hash 所占区块大小，如 mhash\_get\_block\_size(MHASH\_CRC32)。

**mhash()** 函数

语法：

```
string mhash ( int hash, string data [, string key] )
```

该函数返回一个哈希值。参数 hash 为要使用的算法，参数 data 是要加密的数据，参数 key 是加密使用的密钥。

**mhash\_keygen\_s2k()** 函数

语法：

```
string mhash_keygen_s2k ( int hash, string password, string salt, int bytes )
```

该函数将根据参数 password 和 salt 返回一个单位为字节的 key 值，参数 hash 为要使用的算法。其中 salt 为一个固定 8 字节的值，如果用户给出的数值小于 8 字节，将用 0 补齐。

**例 16.9 应用 mhash() 函数和 mhash\_keygen\_s2k() 函数生成一个校验码，并使用 bin2hex() 函数将二进制结果转换为十六进制，代码如下：（实例位置：光盘\TM\Instances\16\9）**

```
<?php
    $filename = 'count.txt';
    $str = file_get_contents($filename);
    $hash = 2;
    $password = '111';
    $salt = '1234';
    $key = mhash_keygen_s2k(1,$password,$salt,10);
    $str_mhash = bin2hex(mhash($hash,$str,$key));
    echo "count.txt 的校验码是: ".$str_mhash;
?>
```

运行结果如图 16.7 所示。

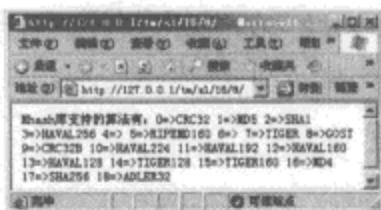


图 16.6 Mhash 库支持的加密算法

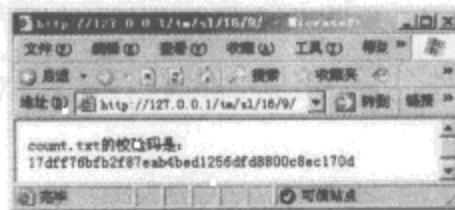


图 16.7 使用 Mhash 库生成校验码

## 16.4 实 战

### ■ 视频讲解：光盘\TM\Video\16\用户注册密码的 MD5 加密.exe

很多网站注册用户的密码都是先使用 MD5 加密，然后再保存到数据库中的。在用户登录时，程序将用户输入的密码计算成 MD5 值，然后再去和数据库中存储的 MD5 值进行比较。在这个过程中，程序自身都不会“知道”用户的真实密码，从而保证了注册用户的个人隐私，提高了安全性。

**例 16.10 实现用户注册的功能，将用户注册的密码利用 MD5 进行加密，然后保存到数据库中。其操作步骤如下：（实例位置：光盘\TM\Instances\16\10）**

(1) 首先连接数据库。创建 conn.php 文件，实现与数据库的连接，代码如下。

```
<?php
    $conn = mysql_connect("localhost","root","root") or die("数据库连接错误".mysql_error());
    mysql_select_db("db_database16",$conn) or die("数据库访问错误".mysql_error());
    mysql_query("set names gb2312");
?>
```

(2) 创建 index.php 页面，添加用户登录的表单，提交到 index\_ok.php 页面进行处理。form 表单的代码如下。

```
<form id="form1" name="form1" method="post" action="index_ok.php">
    <input name="username" type="text" id="username" size="24" />
```

```

<input name="password" type="password" id="password" size="25" />
<input name="address" type="text" id="address" size="35" />
<input name="tel" type="text" id="tel" size="20" />
<input type="image" name="imageField" src="images/bg2.JPG" />
<input type="image" name="imageField2" src="images/bg1.JPG" onclick="form.reset();return false;" />
</form>

```

(3) 创建 index\_ok.php 文件, 对表单中提交的数据进行处理, 对用户密码利用 MD5 进行加密, 最后将数据添加到数据库中, 代码如下。

```

<?php
    include("conn/conn.php");
    if(trim($_POST[username])!= "" and trim($_POST[password])!= "" and trim($_POST[tel])!=""){
        $pwd = md5($_POST[password]);
        $sql = "insert into tb_user(username,password,address,explains)values('".$_POST[username]."', '".$pwd."', '".$_POST[address]."', '".$_POST[tel]."')";
        $result = mysql_query($sql,$conn);
        if($result){
            echo "<font color='red'>注册成功。</font>";
        }else{
            echo "<font color='green'>注册失败!</font>";
        }
    }else{
        echo "请认真填写用户名和密码!";
    }
?>

```

本实例中注册一个名称为 mr 的用户, 密码为 mrsoft。数据库中存储的密码如图 16.8 所示。

服务器: localhost > 数据库: db_database16 > 表 : tb_user				
ID	username	password	address	explains
1	贝ryh2JdaiHqE	128rzbWQ3c	长春市	网页设计师
2	Administrator	123456	长春市	网络营销、电子商务
6	mr	f0b380e945559a74475ed8c8bb048ca5	长春市	1380437****

图 16.8 经过 MD5 加密的用户密码

## 16.5 本 章 小 结

本章主要讲解了 PHP 中的加密函数 crypt()、md5() 和 sha1() 的功能和使用方法, 并且对 PHP 扩展库 Mcrypt 和 Mhash 进行了介绍, 其中 crypt()、md5()、sha1() 和 Mhash 扩展库属于单向加密, 而 Mcrypt 扩展库属于双向加密。相信通过本章的学习及实践, 读者可以熟练地使用各种加密手段对敏感数据进行保护。

## 16.6 实 战 练 习

1. 使用 OR、XOR 等运算符, 自定义一个加密函数。(答案位置: 光盘\TM\Instances\16\11)
2. 应用 md5() 函数制作一个用户登录验证页, 以验证用户登录所使用的用户名和密码是否正确。(答案位置: 光盘\TM\Instances\16\12)

# 第 17 章

## MySQL 数据库基本操作

( 视频讲解: 70 分钟)

数据库作为程序中数据的主要载体，在整个项目中扮演着重要的角色。PHP 自身可以与大多数数据库进行连接，但 MySQL 数据库是开源界所公认的与 PHP 兼容性最好的数据库。MySQL 具有安全性、跨平台性、体积小和高效等特点，可谓 PHP 的“黄金搭档”。MySQL 操作简便、易学易用，受到很多业界人士的青睐。

通过阅读本章，可以学习到：

- MySQL 数据库的特点
- MySQL 5 的新特性
- 如何连接与断开 MySQL 数据库
- 如何创建、删除数据库
- 如何创建、修改和删除数据表
- 如何管理数据表

## 17.1 MySQL 简介

PHP 在开发 Web 站点或一些管理系统时，需要对大量的数据进行保存。XML 文件和文本文件虽然可以作为数据的载体，但不易进行管理和对大量数据的存储，所以在项目开发时，数据库就显得非常重要。PHP 可以连接的数据库种类较多，其中 MySQL 数据库与其兼容较好，在 PHP 数据库开发中被广泛地应用。

### 17.1.1 MySQL 是什么

MySQL 是一款安全、跨平台、高效的，并与 PHP、Java 等主流编程语言紧密结合的数据库系统。该数据库系统是由瑞典的 MySQL AB 公司开发、发布并支持，由 MySQL 的初始开发人员 David Axmark 和 Michael Monty Widenius 于 1995 年建立的。MySQL 的象征符号是一只名为 Sakila 的海豚，代表着 MySQL 数据库的速度、能力、精确和优秀本质。

目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，使得很多公司都采用 MySQL 数据库以降低成本。

MySQL 数据库可以称得上是目前运行速度最快的 SQL 语言数据库之一。除了具有许多其他数据库所不具备的功能外，MySQL 数据库还是一种完全免费的产品，用户可以直接通过网络下载 MySQL 数据库，而不必支付任何费用。

### 17.1.2 MySQL 特点

MySQL 具有以下主要特点。

- ☒ 功能强大：MySQL 中提供了多种数据库存储引擎，各引擎各有所长，适用于不同的应用场景，用户可以选择最合适的引擎以得到最高性能，可以处理每天访问量超过数亿的高强度的搜索 Web 站点。MySQL 5 支持事务、视图、存储过程、触发器等。
- ☒ 支持跨平台：MySQL 支持至少 20 种以上的开发平台，包括 Linux、Windows、FreeBSD、IBMAIX、AIX、FreeBSD 等。这使得在任何平台下编写的程序都可以进行移植，而不需要对程序做任何的修改。
- ☒ 运行速度快：高速是 MySQL 的显著特性。在 MySQL 中，使用了极快的 B 树磁盘表（MyISAM）和索引压缩；通过使用优化的单扫描多连接，能够极快地实现连接；SQL 函数使用高度优化的类库实现，运行速度极快。
- ☒ 支持面向对象：PHP 支持混合编程方式。编程方式可分为纯粹面向对象、纯粹面向过程、面向对象与面向过程混合 3 种方式。
- ☒ 安全性高：灵活和安全的权限与密码系统，允许基本主机的验证。连接到服务器时，所有的密码传输均采用加密形式，从而保证了密码的安全。
- ☒ 成本低：MySQL 数据库是一种完全免费的产品，用户可以直接通过网络下载。
- ☒ 支持各种开发语言：MySQL 为各种流行的程序设计语言提供支持，为它们提供了很多的 API

- 函数，包括 PHP、ASP.NET、Java、Eiffel、Python、Ruby、Tcl、C、C++、Perl 语言等。
- 数据库存储容量大：MySQL 数据库的最大有效表尺寸通常是由操作系统对文件大小的限制决定的，而不是由 MySQL 内部限制决定的。InnoDB 存储引擎将 InnoDB 表保存在一个表空间内，该表空间可由数个文件创建，表空间的最大容量为 64TB，可以轻松处理拥有上千万条记录的大型数据库。
- 支持强大的内置函数：PHP 中提供了大量内置函数，几乎涵盖了 Web 应用开发中的所有功能。它内置了数据库连接、文件上传等功能，MySQL 支持大量的扩展库，如 MySQLi 等，可以为快速开发 Web 应用提供便利。

### 17.1.3 MySQL 5 支持的特性

MySQL 5 已经是一个非常成熟的企业级应用的数据库管理系统，在许多大型的开源项目中被广泛地应用。MySQL 5 支持许多基本和高级的特性，例如：

- 支持各种数据类型。
- 支持事物、主键外键、行级锁定等特性。
- select 查询语句和 where 字句中，提供完整的操作符和函数支持。
- 支持子查询。
- 支持 group by 和 order by 子句。
- 支持各种聚合函数。
- 支持 left outer join 和 right outer join 多表连接查询。
- 支持表别名、字段别名。
- 支持跨库多表连接查询。
- 支持查询缓存，能够极大地提升查询性能。
- 支持存储过程、视图和触发器等特性。
- 支持多平台、多 CPU 等特性。
- 支持嵌入式，可以将 MySQL 集成到嵌入式程序中。

## 17.2 启动、连接、断开和停止 MySQL 服务器

MySQL 数据库服务器的安装和维护相对简便，在 Linux/UNIX 操作系统中，使用终端命令即可实现对 MySQL 数据库服务器的管理，而在 Windows 操作系统中，不仅可以在命令提示符窗口中通过命令的方式管理 MySQL 服务器，而且可以通过图形界面进行管理。下面将以 Windows 操作系统为例介绍如何对其进行启动、关闭等操作。

### 17.2.1 启动 MySQL 服务器

 视频讲解：光盘\TM\Video\17\启动 MySQL 服务器.exe

在 Windows 操作系统中启动 MySQL 服务器的方法主要有两种：通过系统服务方式和在命令提示

符下通过命令行方式。

### 1. 通过系统服务启动 MySQL 服务器

如果 MySQL 设置为 Windows 服务，则可以通过选择“开始”/“管理工具”/“服务”命令，打开 Windows 服务管理器，在服务器的列表中找到 mysql 服务，单击鼠标右键，在弹出的快捷菜单中选择“启动”命令，启动 MySQL 服务，如图 17.1 所示。

### 2. 在命令提示符下启动 MySQL 服务器

选择“开始”/“运行”命令，在弹出的“运行”窗口中输入“cmd”命令，按 Enter 键进入命令提示符窗口。在命令提示符下输入：

```
> net start mysql
```

按 Enter 键后，启用 MySQL 服务器，如图 17.2 所示。

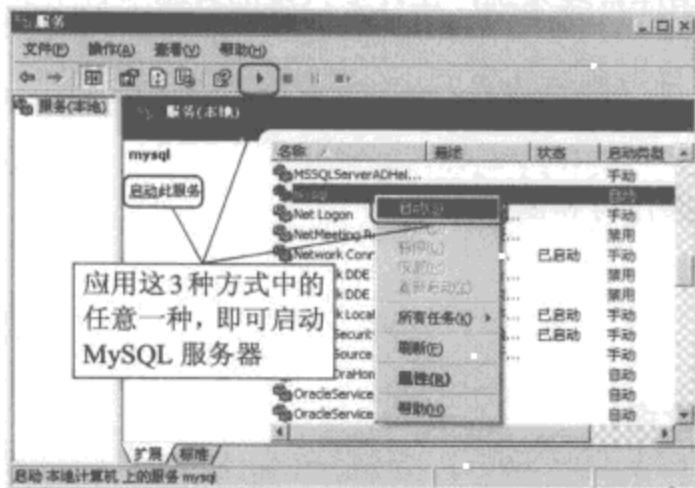


图 17.1 通过系统服务启动 MySQL 服务器

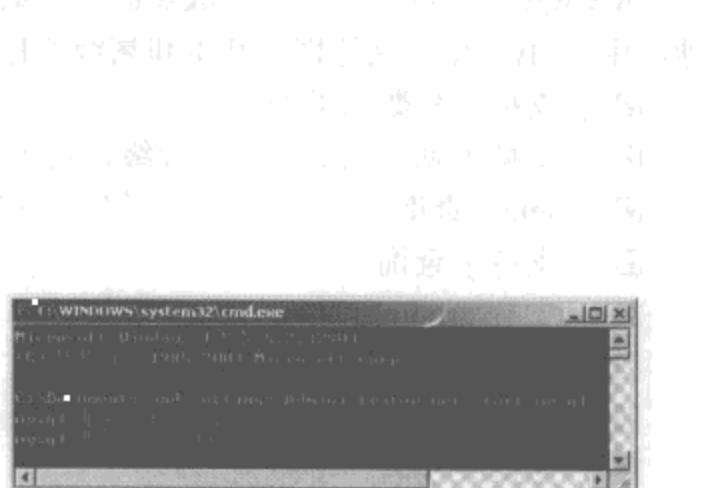


图 17.2 在命令提示符下启动 MySQL 服务器

## 17.2.2 连接和断开 MySQL 服务器

### 视频讲解：光盘\TM\Video\17\连接、断开 MySQL 服务器.exe

对 MySQL 数据库进行管理，首先要建立与 MySQL 服务器的连接。下面分别介绍连接和断开 MySQL 服务器的方法。

#### 1. 连接 MySQL 服务器

连接 MySQL 服务器通过 mysql 命令实现。在 MySQL 服务器启动后，选择“开始”/“运行”命令，在弹出的“运行”窗口中输入“cmd”命令，按 Enter 键后打开命令提示符窗口。在命令提示符下输入：

```
> mysql -h 主机名 -u 用户名 -p 密码
```

输入完命令语句后，按 Enter 键即可连接 MySQL 服务器，如图 17.3 所示。

#### 2. 断开 MySQL 服务器

连接到 MySQL 服务器后，可以通过在 MySQL 提示符下输入“exit”或者“quit”命令断开 MySQL

连接，如图 17.4 所示。

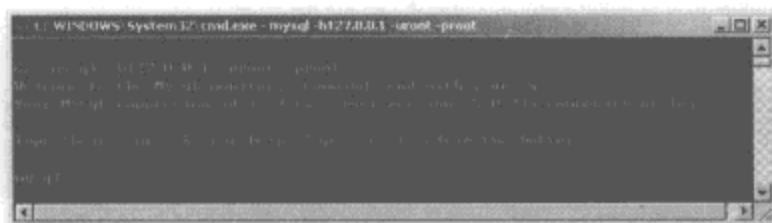


图 17.3 连接 MySQL 服务器

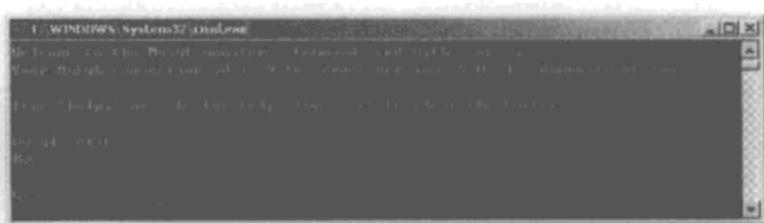


图 17.4 断开 MySQL 服务器

### 17.2.3 停止 MySQL 服务器

#### ■ 视频讲解：光盘\TM\Video\17\停止 MySQL 服务器.exe

为有效节省系统资源，在使用完 MySQL 服务后需要将其及时关闭。与启动 MySQL 服务类似，在 Windows 操作系统中，停止 MySQL 服务器主要有两种方法：通过系统服务中关闭 MySQL 服务和命令提示符下通过命令行的方式关闭 MySQL 服务。下面具体介绍这两种实现方式。

#### 1. 通过系统服务停止 MySQL 服务器

如果使用的是安装版的 MySQL 数据库，则可以通过选择“开始”/“管理工具”/“服务”命令，打开 Windows 服务管理器，在服务器的列表中找到 mysql 服务并右击，在弹出的快捷菜单中选择“停止”命令，停止 MySQL 服务器，如图 17.5 所示。

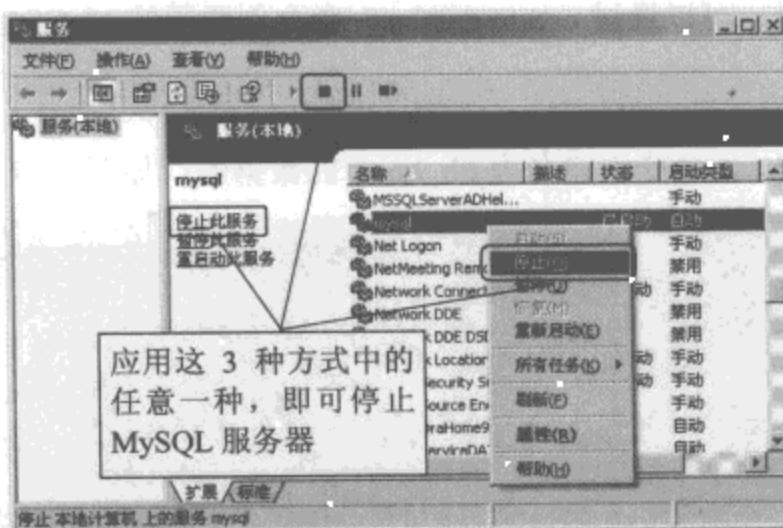


图 17.5 停止 MySQL 服务器

#### 2. 在命令提示符下停止 MySQL 服务器

选择“开始”/“运行”命令，输入“cmd”命令，进入命令提示符窗口，在命令提示符下输入：

```
> net stop mysql
```

按 Enter 键即可停止 MySQL 服务器，如图 17.6 所示。

#### 3. 使用 mysqladmin 命令停止 MySQL 服务器

选择“开始”/“运行”命令，输入“cmd”命令，进入命令提示符窗口，在命令提示符下输入：

```
\> mysqladmin -uroot shutdown -proot
```

按 Enter 键即可停止 MySQL 服务, 如图 17.7 所示。



图 17.6 在命令提示符中停止 MySQL 服务器

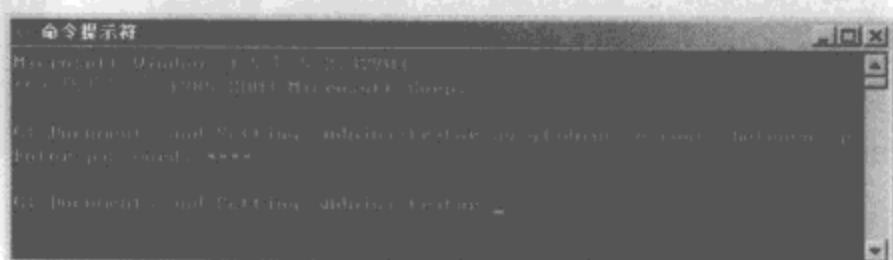


图 17.7 使用 mysqladmin 命令停止 MySQL 服务器

## 17.3 使用 MySQL 命令行操作数据库

### ■ 视频讲解：光盘\TM\Video\17\使用 MySQL 命令行操作数据库.exe

成功登录 MySQL 数据库服务器后, 即可在命令提示符窗口中对 MySQL 数据库进行管理, 如创建、查看、更改和删除 MySQL 数据库等操作。

### 17.3.1 创建数据库

登录 MySQL 数据库后, 可以通过 create database 命令创建新的 MySQL 数据库。语法如下:

```
create database 数据库名
```

例如, 使用 create database 命令创建名为 db\_book 的数据库, 如图 17.8 所示。



图 17.8 使用 create database 命令创建数据库

在创建数据库时, 数据库命名要遵循以下几项规则:

- 不能与其他数据库重名, 否则将发生错误。
- 名称可以由任意英文字母、阿拉伯数字, 下划线“\_”或者“\$”组成, 可以使用上述的任意字符开头, 但不能使用单独的数字, 否则会造成数据库名与数值相混淆。
- 名称最长可为 64 个字符, 而别名可长达 256 个字符。
- 不能使用 MySQL 关键字作为数据库、表名。
- 默认情况下, Windows 下数据库名、表名的字母大小写是不敏感的, 而在 Linux 下数据库名、表名的字母大小写是敏感的。为了便于数据库在平台间进行移植, 建议读者采用小写字母来

定义数据库名和表名。

### 17.3.2 查看数据库

数据库创建完成后，可以使用 `show databases` 命令查看 MySQL 数据库中所有已经存在的数据库。语法如下：

```
show databases;
```

例如，使用 `show databases` 命令显示本地 MySQL 数据库中所有存在的数据库名，如图 17.9 所示。



图 17.9 显示所有数据库名

### 17.3.3 选择数据库

上面的讲解中，虽然成功创建了 `db_book` 数据库，但并不表示已经选择了 `db_book` 数据库。MySQL 中可以应用 `use` 命令选择一个数据库，使其成为当前默认数据库。语法如下：

```
use 数据库名;
```

例如，选择名称为 `db_book` 的数据库，使其成为当前默认的数据库，如图 17.10 所示。

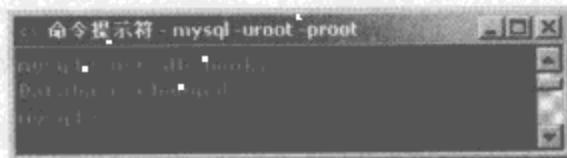


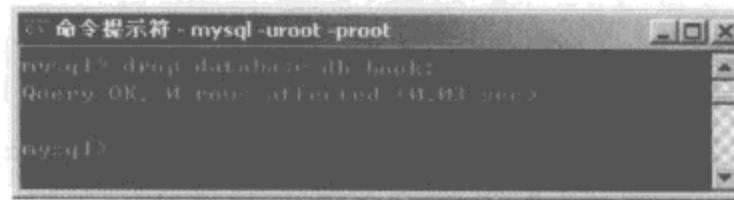
图 17.10 选择数据库

### 17.3.4 删除数据库

MySQL 数据库中，可以使用 `drop database` 命令删除指定的数据库。语法如下：

```
drop database 数据库名;
```

例如，通过 `drop database` 语句删除名称为 `db_book` 的数据库，如图 17.11 所示。



命令提示符  
数据库删除成功  
数据库不存在

图 17.11 删除数据库

## 17.4 使用 MySQL 命令行操作数据表

### ■ 视频讲解：光盘\TM\Video\17\使用 MySQL 命令行操作数据表.exe

创建完成数据库后，即可在命令提示符下对数据库进行操作，如创建数据表、更改数据表结构以及删除数据表等。

### 17.4.1 创建数据表

MySQL 数据库中，可以使用 create table 命令创建数据表。语法如下：

```
create[TEMPORARY] table [IF NOT EXISTS] 数据表名
[(create_definition,...)][table_options] [select_statement]
```

create table 语句的参数说明如表 17.1 所示。

表 17.1 create table 语句的参数说明

关键字	说 明
TEMPORARY	如果使用该关键字，表示创建一个临时表
IF NOT EXISTS	该关键字用于避免表存在时 MySQL 报告的错误
create definition	这是表的列属性部分。MySQL 要求在创建表时，表要至少包含一列
table options	表的一些特性参数
select statement	SELECT 语句描述部分，用它可以快速地创建表

下面介绍列属性 create\_definition 的使用方法，每一列具体的定义格式如下：

```
col_name type [NOT NULL | NULL] [DEFAULT default_value] [AUTO_INCREMENT]
[PRIMARY KEY] [reference_definition]
```

属性 create\_definition 的参数说明如表 17.2 所示。

表 17.2 属性 create\_definition 的参数说明

参 数	说 明
col_name	字段名
type	字段类型
NOT NULL   NULL	指出该列是否允许是空值，但是数据“0”和空格都不是空值，系统一般默认允许为空值，所以当不允许为空值时，必须使用 NOT NULL

续表

参 数	说 明
DEFAULT default_value	表示默认值
AUTO_INCREMENT	表示是否是自动编号，每个表只能有一个 AUTO_INCREMENT 列，并且必须被索引
PRIMARY KEY	表示是否为主键。一个表只能有一个 PRIMARY KEY。如表中没有一个 PRIMARY KEY，而某些应用程序要求 PRIMARY KEY，MySQL 将返回第一个没有任何 NULL 列的 UNIQUE 键，作为 PRIMARY KEY
reference definition	为字段添加注释

在实际应用中，使用 create table 命令创建 MySQL 数据库只需指定最基本的属性即可，格式如下：

```
create table table_name (列名1 属性, 列名2 属性 ...);
```

例如，在命令提示符下应用 create table 命令，在数据库 db\_admin 中创建一个名为 tb\_admin 的数据表，表包括 id、user、password 和 createtime 等字段，实现过程如图 17.12 所示。

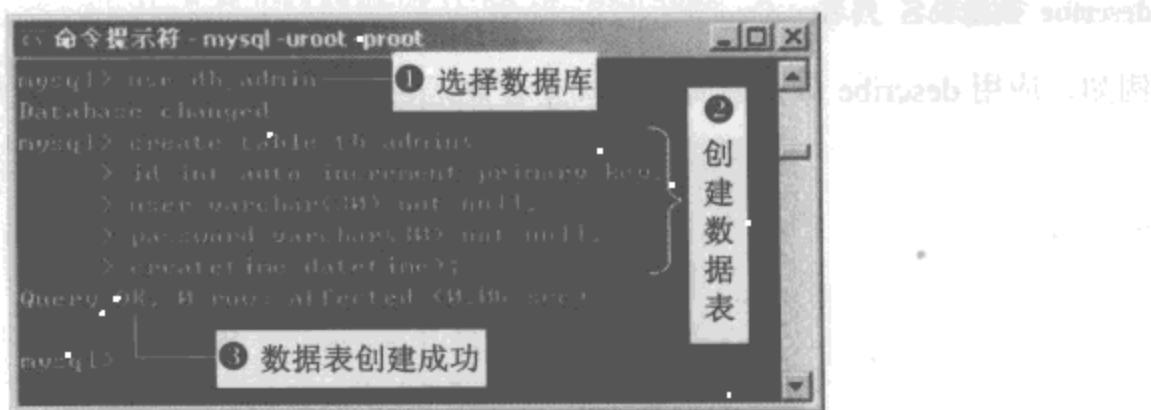


图 17.12 创建 MySQL 数据表

#### 17.4.2 查看表结构

成功创建数据表后，可以使用 show columns 命令或 describe 命令查看指定数据表的表结构。下面分别对这两个语句进行介绍。

##### 1. show columns 命令

show columns 命令的语法格式如下：

```
show [full] columns from 数据表名 [from 数据库名];
```

或写成：

```
show [full] columns FROM 数据表名.数据库名;
```

例如，应用 show columns 命令查看数据表 tb\_admin 表结构，如图 17.13 所示。



图 17.13 查看表结构

## 2. describe 命令

describe 命令的语法格式如下：

```
describe 数据表名;
```

其中，describe 可以简写为 desc。在查看表结构时，也可以只列出某一列的信息，语法格式如下：

```
describe 数据表名 列名;
```

例如，应用 describe 命令的简写形式查看数据表 tb\_admin 表的某一列信息，如图 17.14 所示。



图 17.14 查看表的某一列信息

### 17.4.3 修改表结构

修改表结构采用 alter table 命令。修改表结构指增加或者删除字段、修改字段名称或者字段类型、设置取消主键外键、设置取消索引以及修改表的注释等。

语法：

```
alter [IGNORE] table 数据表名 alter_spec[,alter_spec]...
```

注意，当指定 IGNORE 时，如果出现重复关键的行，则只执行一行，其他重复的行被删除。其中，alter\_spec 子句用于定义要修改的内容，语法如下：

<b>alter_specification:</b>	
<b>ADD</b> [COLUMN] <b>create_definition</b> [FIRST   AFTER <b>column_name</b> ]	-添加新字段
<b>ADD INDEX</b> [ <b>index_name</b> ] ( <b>index_col_name</b> ,...)	-添加索引名称
<b>ADD PRIMARY KEY</b> ( <b>index_col_name</b> ,...)	-添加主键名称
<b>ADD UNIQUE</b> [ <b>index_name</b> ] ( <b>index_col_name</b> ,...)	-添加唯一索引
<b>ALTER</b> [COLUMN] <b>col_name</b> {SET DEFAULT <b>literal</b>   DROP DEFAULT}	-修改字段名称
<b>CHANGE</b> [COLUMN] <b>old_col_name</b> <b>create_definition</b>	-修改字段类型

MODIFY [COLUMN] create_definition	--修改子句定义字段
DROP [COLUMN] col_name	--删除字段名称
DROP PRIMARY KEY	--删除主键名称
DROP INDEX index_name	--删除索引名称
RENAME [AS] new_tbl_name	--更改表名
table_options	

alter table 语句允许指定多个动作，动作间使用逗号分隔，每个动作表示对表的一个修改。

例如，添加一个新的字段 email，类型为 varchar(50)，not null，将字段 user 的类型由 varchar(30) 改为 varchar(40)，如图 17.15 所示。



图 17.15 修改表结构

图 17.15 中只给出了修改 user 字段类型的结果，可以通过如下语句查看整个表的结构，以确认 email 字段是否添加成功。

```
mysql> show tb_admin;
```

#### 17.4.4 重命名表

重命名数据表采用 rename table 命令，语法格式如下：

```
rename table 数据表名 1 To 数据表名 2
```

说明：该语句可以同时对多个数据表进行重命名，多个表之间以逗号“,”分隔。

例如，对数据表 tb\_admin 进行重命名，更名后的数据表为 tb\_user，如图 17.16 所示。



图 17.16 对数据表进行更名

### 17.4.5 删除表

删除数据表的操作很简单，与删除数据库的操作类似，使用 drop table 命令即可实现。格式如下：

```
drop table 数据表名;
```

例如，删除数据表 tb\_user，如图 17.17 所示。

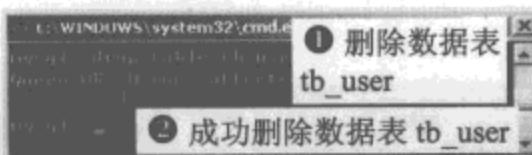


图 17.17 删除数据表

在删除数据表的过程中，如果删除一个不存在的表将会产生错误，这时在删除语句中加入 if exists 关键字就可避免出错。格式如下：

```
drop table if exists 数据表名;
```

## 17.5 MySQL 语句操作

### ■ 视频讲解：光盘\TM\Video\17\MySQL 语句操作.exe

向数据表中插入、浏览、修改和删除记录可以在 MySQL 命令行中使用 SQL 语句完成。下面介绍如何在 MySQL 命令行中执行基本的 SQL 语句。

### 17.5.1 插入记录

建立一个空的数据库和数据表时，首先要想到的就是如何向数据表中添加数据。这项操作可以通过 insert 命令来实现。

语法：

```
insert into 数据表名(column_name,column_name2, ... ) values (value1,value2, ... )
```

在 MySQL 中，一次可以同时插入多行记录，各行记录的值清单在 values 关键字后以逗号“,” 分隔，而标准的 SQL 语句一次只能插入一行。

例如，向管理员信息表 tb\_admin 中插入一条数据信息，如图 17.18 所示。

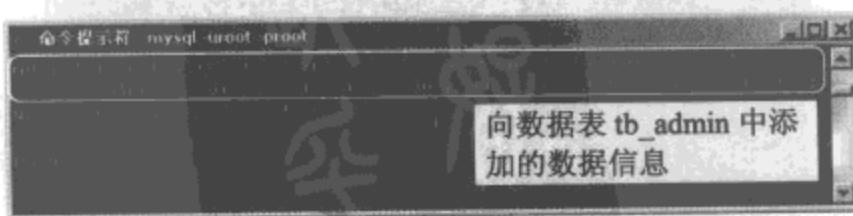


图 17.18 插入记录

## 17.5.2 浏览数据库记录

录屏 17.18

要从数据库中把数据查询出来，就要用到数据查询命令 select。select 命令是最常用的查询命令。语法：

```
select selection_list
from 数据表名
where primary_constraint
group by grouping_columns
order by sorting_columns
having secondary_constraint
limit count
```

--要查询的内容，选择哪些列  
--指定数据表  
--查询时需要满足的条件，行必须满足的条件  
--如何对结果进行分组  
--如何对结果进行排序  
--查询时满足的第二条件  
--限定输出的查询结果

该语句所使用的子句将在后面逐一进行介绍。下面先介绍 select 命令的简单应用。

### 1. 使用 select 命令查询一个数据表

使用 select 命令时，首先要确定所要查询的列。“\*”代表所有的列。

例如，查询管理员信息表 tb\_admin 中的所有数据，如图 17.19 所示。

这是查询整个表中所有列的操作，还可以针对表中的某一列或多列进行查询。

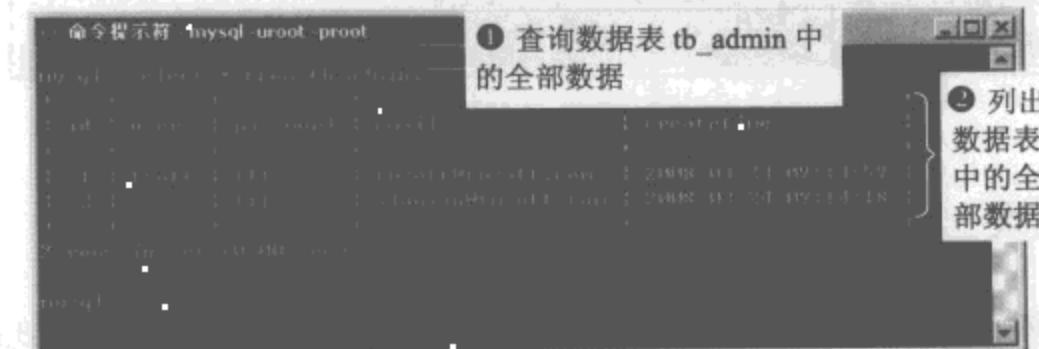


图 17.19 查询数据表的全部数据

### 2. 查询表中的一列或多列

针对表中的多列进行查询，只要在 select 后面指定要查询的列名即可，多列之间用“,”分隔。

例如，查询管理员信息表 tb\_admin 中的 id (ID 编号)、user (用户名)、password (用户密码)、email (用户邮箱) 字段，并指定查询条件是用户 ID 编号为 1，如图 17.20 所示。



图 17.20 查看数据表中指定字段的数据

### 17.5.3 修改记录

命令提示符 - mysql -uroot -proot

要执行修改的操作可以使用 update 命令，该语句的格式如下：

```
update 数据表名 set column_name = new_value1, column_name2 = new_value2, ... where condition
```

其中，set 子句指出要修改的列及其给定的值；where 子句是可选的，如果给出该子句将指定记录中哪行应该被更新，否则，所有的记录行都将被更新。

例如，下面将管理员信息表 tb\_admin 中用户名为 tsoft 的管理员密码 “111” 修改为 “896552”，如图 17.21 所示。

```

命令提示符 - mysql -uroot -proot
mysql> update tb_admin set password='896552' where user='tsoft';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from tb_admin where user='tsoft';
+----+-----+-----+-----+
| id | user | password | email |
+----+-----+-----+-----+
| 1  | tsoft | 111       | tsoft@ttssoft.com | 2008-03-24 09:14:59 |
+----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

图 17.21 修改指定条件的记录

### 17.5.4 删除记录

在数据库中有些数据已经失去意义或者是错误的，这时就需要将它们删除，此时可以使用 delete 命令。该命令的格式如下：

```
delete from 数据表名 where condition
```

**注意：**该语句在执行过程中，如果没有指定 where 条件，将删除所有的记录；如果指定了 where 条件，将按照指定的条件进行删除。

使用 delete 命令删除整个表的效率并不高，还可以使用 truncate 命令，利用它可以快速删除表中所有的内容。

例如，删除管理员数据表 tb\_admin 中用户名为“小欣”的记录信息，如图 17.22 所示。

```

命令提示符 - mysql -uroot -proot
mysql> delete from tb_admin where user='xixi';
Query OK, 1 row affected (0.00 sec)

mysql>

```

图 17.22 删除数据表中指定的记录

## 17.6 实 战

视频讲解：光盘\TM\Video\17\管理 MySQL 数据库及数据表.exe

作为一名 PHP 开发人员，除具备扎实的 PHP 基础外，还需要掌握 MySQL 数据库方面的相关知识，因为在开发 PHP 项目时，用数据库作为数据的载体是必不可少的，而 MySQL 数据库是业界所公认的 PHP 的最佳搭档。

**例 17.1** 创建数据库 db\_test01，并在该数据库中创建数据表 tb\_test01，最后使用 insert 命令向 tb\_test01 表中插入一条数据。

开发步骤如下：

(1) 选择“开始”/“运行”命令，在弹出的“运行”窗口中输入“cmd”命令来打开命令提示符窗口，在命令提示符下输入如下命令建立与 MySQL 数据库的连接（以 MySQL 用户名为 root，密码为 root，并且将本机作为 MySQL 数据库服务器为例）。

```
mysql -h127.0.0.1 -uroot -proot
```

(2) 在命令提示符下输入如下命令来创建数据库 db\_test01。

```
create database db_test01;
```

(3) 选择 db\_test01 数据库，输入如下命令。

```
use db_test01;
```

(4) 使用 create table 命令在数据库 db\_test01 下创建数据表 tb\_test01，并要求 tb\_test01 表具有 id、username 和 password 这 3 个字段，如下所示。

```
create table tb_test01(
    id int auto_increment primary key,
    username varchar(50) not null,
    password varchar(50) not null
);
```

(5) 使用 insert 命令向 tb\_test01 表中插入一条记录，代码如下。

```
insert into tb_test01(username, password) values(' 小刘', ' 12345678');
```

(6) 使用 select 命令查询 tb\_test01 表中的所有记录，如图 17.23 所示。



图 17.23 显示 tb\_test01 表中的所有记录

## 17.7 本章小结

本章对 MySQL 数据库的基本概念、MySQL 5 的新特性进行了介绍，并详细介绍了 Windows 系统下，使用“命令提示符”窗口创建和维护 MySQL 数据库和数据表的方法，在讲解过程中注重实践和常用命令的讲解，一切以能够帮助读者打好基础为出发点。通过本章的学习，读者能够了解 MySQL 数据库的基本操作和维护方法，掌握 MySQL 数据库中最基本和最常用命令的语法格式，并能够具备基本管理和维护 MySQL 数据库的能力。

## 17.8 实战练习

1. 创建数据库 db\_shop，并在 db\_shop 中创建数据表 tb\_goods。完成表的创建后，向数据表中插入两条记录，然后删除第一条记录，最后显示表结构。（答案位置：光盘\TM\Instances\17\1）
2. 创建数据库 db\_shop，然后将 db\_shop 更名为 db\_shopping，并确保更名成功。（答案位置：光盘\TM\Instances\17\2）

# 18章

## phpMyAdmin 图形化管理工具

( 视频讲解：57分钟)

phpMyAdmin 是由 PHP 开发的一个类似于 SQL Server 的可视化图形管理工具。有了该工具，PHP 开发者就不必通过命令来操作 MySQL 数据库了，可以像 SQL Server 那样通过可视化的图形来操作数据库。phpMyAdmin 可以运行在各种版本的 PHP 及 MySQL 下，对数据库进行操作，如创建、修改和删除数据库、数据表以及数据等。

通过阅读本章，可以学习到：

- » 在 Windows 系统下 phpMyAdmin 工具的安装和配置
- » 在 Linux 系统下 phpMyAdmin 工具的安装和配置
- » 创建、修改和删除数据库的方法
- » 创建、修改和删除数据表的方法
- » 运用 SQL 语句操作数据表
- » 使用 phpMyAdmin 工具向数据表中插入数据
- » 使用 phpMyAdmin 工具搜索数据表中的数据
- » 数据导入导出的方法



## 18.1 phpMyAdmin 简介

phpMyAdmin 是众多 MySQL 图形化管理工具中应用最广泛的一种，是一款使用 PHP 开发的 B/S 模式的 MySQL 客户端软件，该工具是基于 Web 跨平台的管理程序，并且支持简体中文。用户可以在其官方网站 [www.phpmyadmin.net](http://www.phpmyadmin.net) 上免费下载到最新的版本。phpMyAdmin 为 Web 开发人员提供了类似于 Access、SQL Server 的图形化数据库操作界面，通过该管理工具完全可以对 MySQL 进行操作，如创建数据库、数据表和生成 MySQL 数据库脚本文件等。

## 18.2 phpMyAdmin 的安装和配置

用户在配置好 Apache 服务器对 PHP 的支持后（在第 2 章详细讲解了 PHP 开发环境），通过安装方便快捷的 phpMyAdmin 图形化管理工具来操作 MySQL 数据库。下面将分别介绍如何在 Windows 操作系统下和 Linux 操作系统下安装和配置 phpMyAdmin 工具。

### 18.2.1 Windows 系统下 phpMyAdmin 的安装配置

#### ■ 视频讲解：光盘\TM\Video\18\Windows 系统下 phpMyAdmin 的安装配置.exe

安装 phpMyAdmin 前，首先应到其官方网站首页 [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php) 下载安装包，本节以 phpMyAdmin-2.11.5-all-languages.rar 为例进行讲解。

#### 1. 安装 phpMyAdmin 图形化管理工具

将下载的 phpMyAdmin-2.11.5-all-languages.rar 文件解压缩到 Apache 主目录下，解压后的名称是 phpMyAdmin-2.11.5-all-languages，其中 2.11.5 是 phpMyAdmin 的版本号，为了便于使用可以将解压缩后的文件夹重新命名为 phpMyAdmin。

**◆ 注意：**如果读者使用的是 AppServ 安装的 PHP 开发环境，那么无须再安装 phpMyAdmin 图形化管理工具，因为 AppServ 已经自带该工具。该工具的文件存储在 www 主目录下。

#### 2. 配置 phpMyAdmin 图形化管理工具

将 phpMyAdmin 安装目录下 libraries 目录下面的 config.default.php 复制到 phpMyAdmin 根目录下，改名为 config.inc.php，然后用 Dreamweaver 开发工具打开 config.inc.php 文件，将光标定位到指定的标识行位置进行配置。config.inc.php 文件初始值设置如图 18.1 所示，配置后的文件如图 18.2 所示。

```
$cfg['PmaAbsoluteUri'] = '';
$cfg['blowfish_secret'] = '';
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
```

图 18.1 配置 config.inc.php 文件前



```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin';
$cfg['blowfish_secret'] = 'cookie';
$cfg['Servers'][$i]['auth_type'] = 'http';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'root';
```

图 18.2 配置 config.inc.php 文件后

第 39 行用来设置 phpMyAdmin 网址。

第 64 行设置访问方式为 cookie，本机则不需要设置，但如果是网络，就需要设置成 cookie。

第 151 行用来设置 phpMyAdmin 的登录方式。可以设置的登录方式有 3 种：config、http 和 cookie。其中 config 是默认的登录方式，该方式表示用户正确输入用户名和密码后可直接登录；如果采用 http 方式登录，会弹出一个对话框，提示输入用户名和密码；如果使用 cookie 方式登录，在访问 phpMyAdmin 时会打开一个登录页面，提示用户输入用户名和密码，并将用户名和密码保存在系统的 cookie 中，下次访问时用户名和密码会自动显示在文本框和密码输入框中，通常用于互联网。

第 158 行用来设置 MySQL 数据库的用户名。

第 165 行用来设置 MySQL 数据库的密码。

**说明：**如果读者未使用 Dreamweaver 工具打开 config.inc.php 文件，那么对应的行号会有所差异，读者可以搜索相关的关键字（如['auth\_type']）进行搜索定位，然后更改对应的配置文件。

设置完毕后，保存 config.php 文件，在浏览器中输入“<http://localhost/phpMyAdmin>”，将弹出用户身份验证窗口，如图 18.3 所示。

在登录 phpMyAdmin 工具时，输入正确的用户名和密码（这里在安装 MySQL 时，设置 MySQL 数据库的用户名为 root，密码为 root），即可进入 phpMyAdmin 的主界面，如图 18.4 所示。

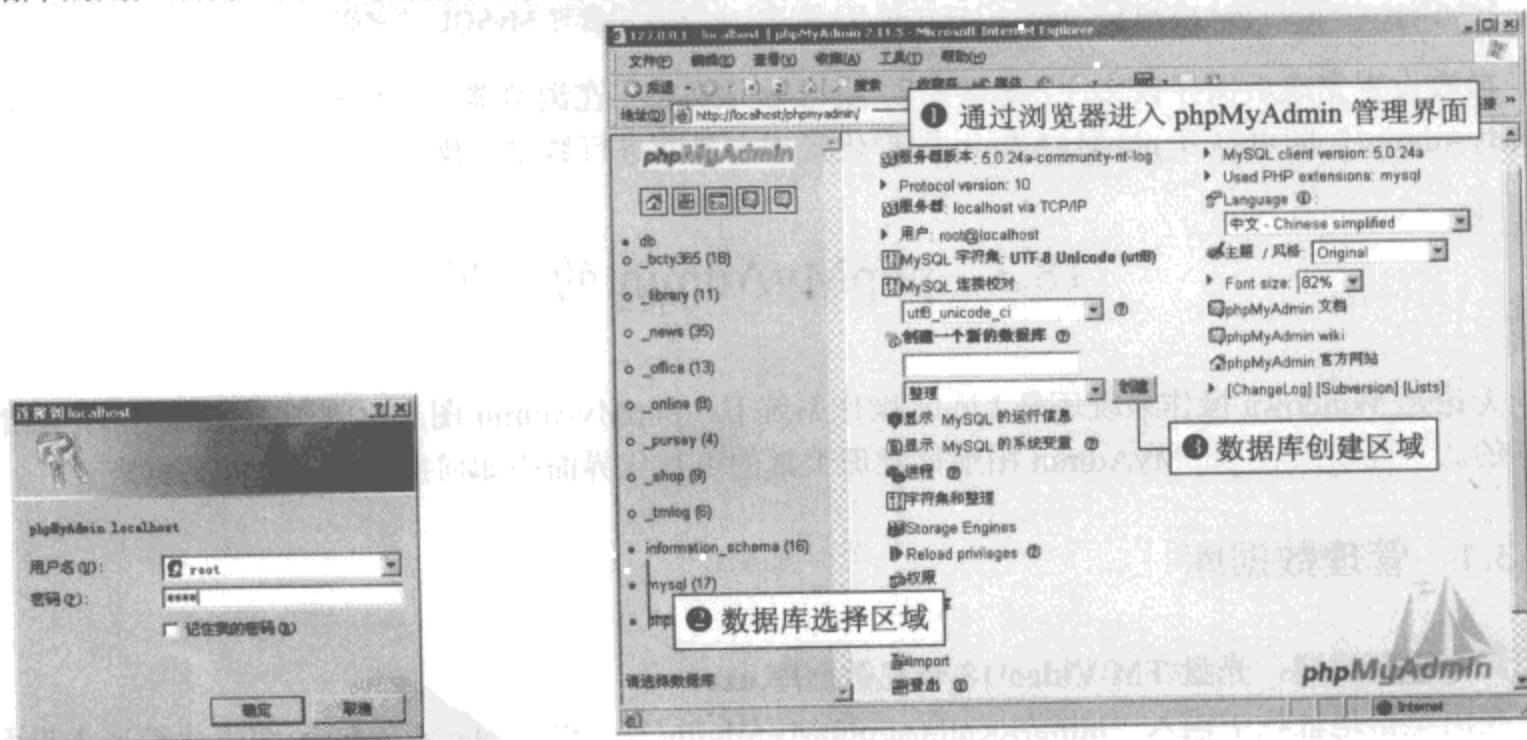


图 18.3 用户登录对话框

图 18.4 phpMyAdmin 的管理界面

## 18.2.2 Linux 系统下 phpMyAdmin 的安装配置

phpMyAdmin 是跨平台的，可以在 Linux、Windows 以及其他平台上运行。下面介绍如何在 Linux 操作系统下安装和配置 phpMyAdmin 工具。

### 1. 安装 phpMyAdmin 图形化管理工具

在 Linux 操作系统下安装 phpMyAdmin 的方法与在 Windows 操作系统下相似，同样将下载的软件

包 phpMyAdmin-2.11.5-all-languages.rar 解压缩到配置环境的根目录下，为了便于使用可以将解压缩后的文件夹重新命名为 phpMyAdmin。

命令模式下的操作如下：

```
# mv phpMyAdmin-2.11.5-all-languages.rar/var/www/html
# tar zxvf phpMyAdmin-2.11.5-all-languages.rar
# mv phpMyAdmin-2.11.5-all-languages  phpMyAdmin
```

通过上述 3 个命令将下载的软件包解压缩到 /var/www/html 目录下，完成 phpMyAdmin 图形化管理工具的安装。

## 2. 配置 phpMyAdmin 图形化管理工具

配置依然在 phpMyAdmin 目录下的 config.inc.php 文件中进行，更改如下参数的配置。

```
//将参数 PmaAbsoluteUri 的值设定为 phpMyAdmin 所处的位置
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';
$cfg['blowfish_secret'] = 'cookie'; //设定访问方式为 cookie
$cfg['Servers'][$i]['host'] = 'localhost'; //设定 MySQL 所在的主机名或者 IP 地址
$cfg['Servers'][$i]['auth_type'] = 'http'; //设定进入 phpMyAdmin 管理 MySQL 的方式
$cfg['Servers'][$i]['user'] = 'root'; //设定管理 MySQL 的账号
$cfg['Servers'][$i]['password'] = 'root'; //设定管理 MySQL 的密码
```

参数设置完成后保存该文件，重新启动 Apache 服务器。在浏览器的地址栏中输入“<http://localhost/phpMyAdmin/>”即可打开 phpMyAdmin 图形化管理主界面进行数据库操作。

## 18.3 phpMyAdmin 的使用

无论是 Windows 操作系统还是 Linux 操作系统下，phpMyAdmin 图形化管理工具的使用方法都是相同的。下面讲解在 phpMyAdmin 图形化管理工具的可视化界面中如何操纵数据库及数据表。

### 18.3.1 管理数据库

#### ■ 视频讲解：光盘\TM\Video\18\管理数据库.exe

在浏览器地址栏中输入“<http://localhost/phpMyAdmin/>”，进入 phpMyAdmin 图形化管理主界面，接下来即可进行 MySQL 数据库的操作。下面将分别介绍如何创建、修改和删除数据库。

#### 1. 创建数据库

在 phpMyAdmin 的主界面，首先在文本框中输入数据库的名称“db\_study”，然后在下拉列表框中选择所要使用的编码，一般选择 gb2312\_chinese\_ci 简体中文编码格式，单击“创建”按钮，创建数据库，如图 18.5 所示。成功创建数据库后，将显示如图 18.6 所示的界面。

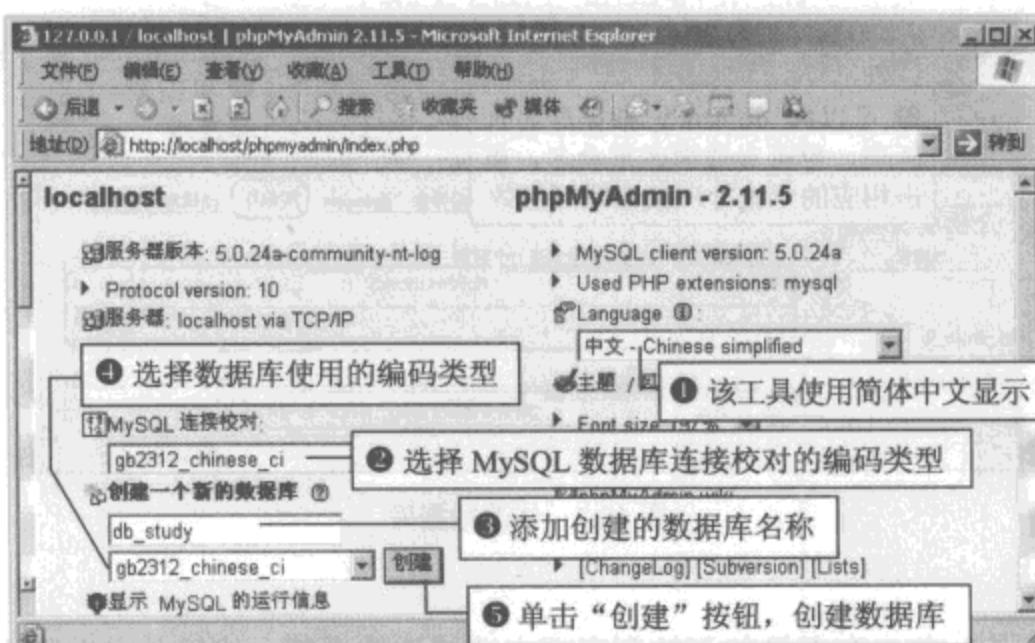


图 18.5 phpMyAdmin 管理主界面

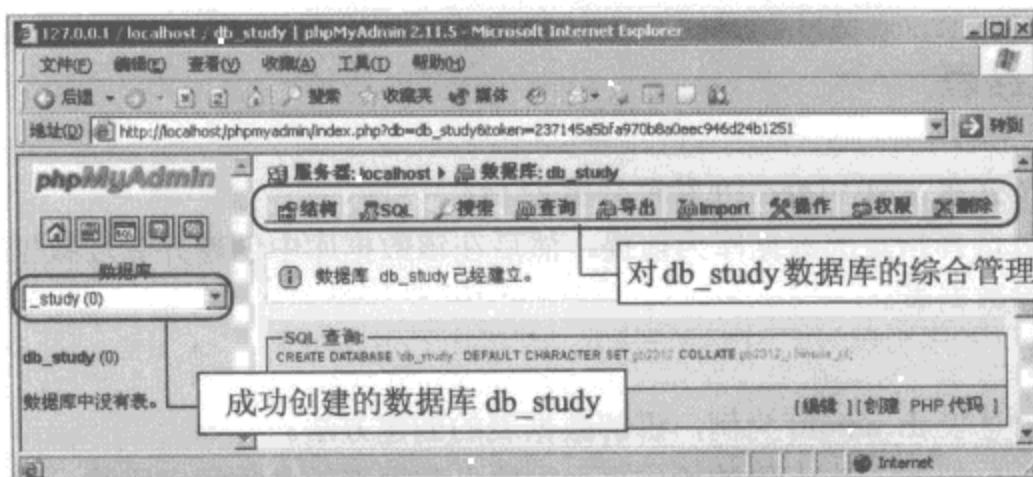


图 18.6 成功创建数据库

**说明：**在图 18.6 右侧界面中可以对该数据库进行相关操作，如结构、SQL、导出、搜索、查询、删除等，单击相应的超链接进入相应的操作界面。但在创建数据库后还没有创建数据表的情况下，只能够执行结构、SQL、Import、操作、权限和删除 6 项操作，其他 3 项操作 **搜索**、**查询**、**导出** 不能执行，当指向其超链接时，显示不可用标记 **X**。

## 2. 修改数据库

在如图 18.6 所示的界面中，在右侧界面还可以对当前数据库进行修改。单击界面中的 **操作** 超链接，进入修改操作页面。

(1) 可以对当前数据库执行创建数据表的操作，只要在创建数据表的提示信息下面的两个文本框中分别输入要创建的数据表的名称和字段总数，然后单击“执行”按钮即可进入到创建数据表结构页面，具体创建方法将在 18.3.2 节中详细讲解。

(2) 也可以对当前的数据库重命名，在“重新命名数据库为”文本框中输入新的数据库名称，单击“执行”按钮，即可成功修改数据库名称。

修改数据库的效果如图 18.7 所示。

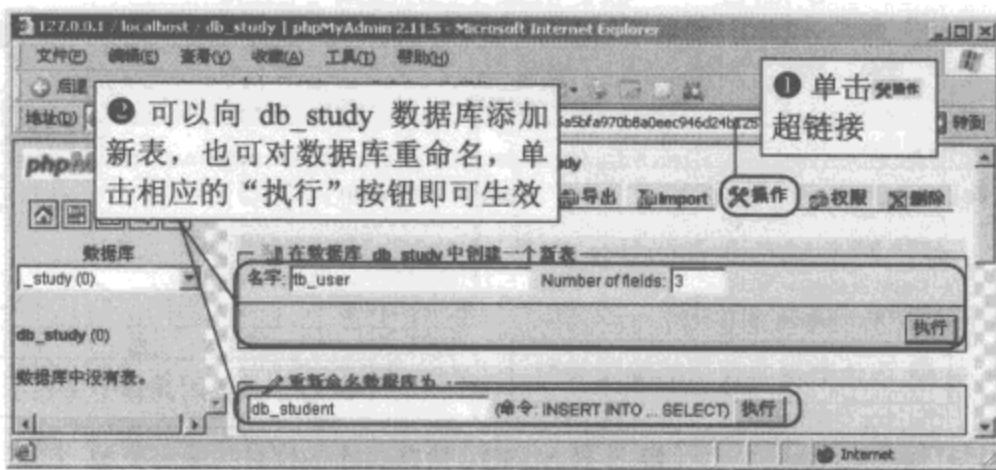


图 18.7 修改数据库

### 3. 删除数据库

要删除某个数据库，首先在左侧的下拉列表框中选择该数据库，然后单击右侧界面中的删除超链接（如图 18.7 所示）即可成功删除指定的数据库。

#### 18.3.2 管理数据表

##### 视频讲解：光盘\TM\Video\18\管理数据表.exe

管理数据表是以选择指定的数据库为前提，然后在该数据库中创建并管理数据表。下面将介绍如何创建、修改、删除数据表。

##### 1. 创建数据表

下面以管理员信息表 tb\_admin 为例，讲解数据表的创建方法。

创建数据库 db\_study 后，在右侧的操作页面中输入数据表的名称和字段数，然后单击“执行”按钮，即可创建数据表，如图 18.8 所示。

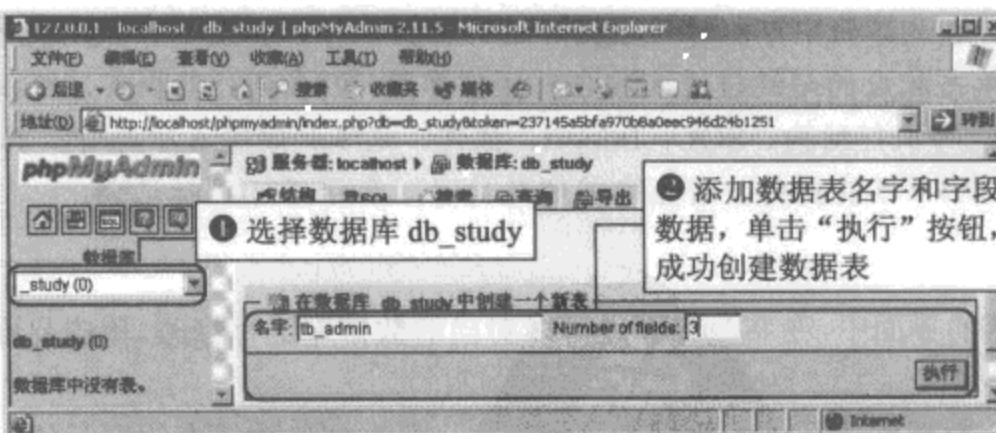


图 18.8 创建数据表

成功创建数据表 tb\_admin 后，将显示数据表结构界面。在表单中对各个字段的详细信息进行录入，包括字段名、数据类型、长度/值、编码格式、是否为空、主键等，完成对表结构的详细设置。所有的信息都输入完成以后，单击“保存”按钮，创建数据表结构，如图 18.9 所示。成功创建数据表结构后，将显示如图 18.10 所示的界面。

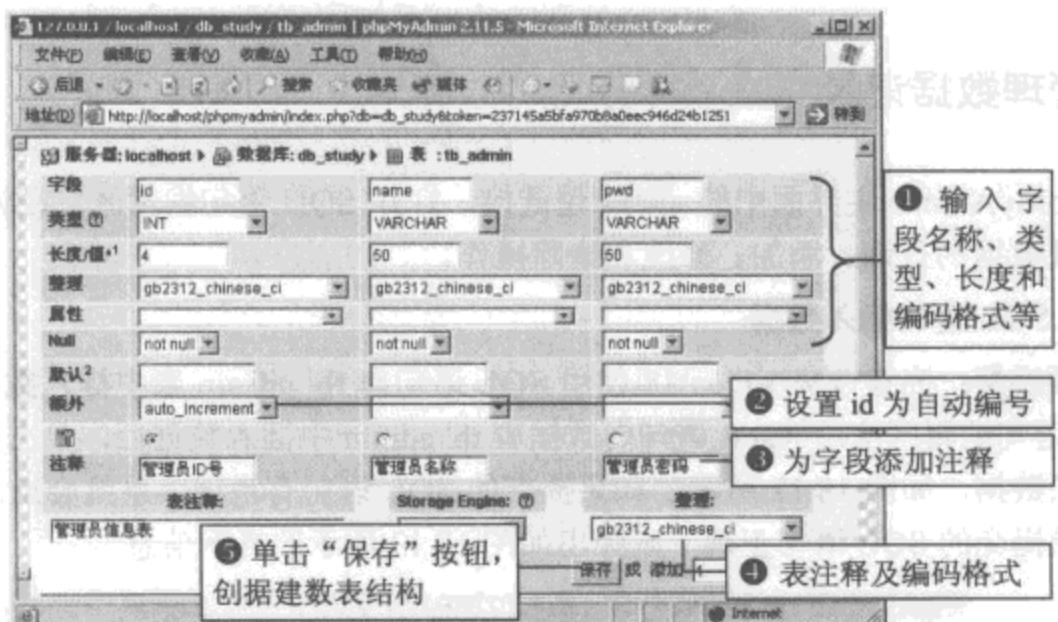


图 18.9 创建数据表结构

技巧：单击“执行”按钮，可以对数据表结构以横版形式进行表结构编辑。

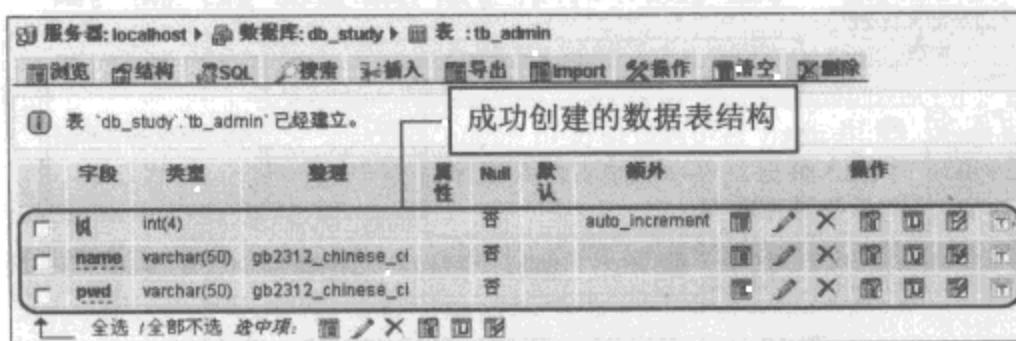


图 18.10 成功创建数据表

## 2. 修改数据表

一个新的数据表被创建后，进入数据表页面中，在这里可以通过改变表的结构来修改表，可以执行添加新的列、删除列、索引列、修改列的数据类型或者字段的长度/值等操作，如图 18.11 所示。

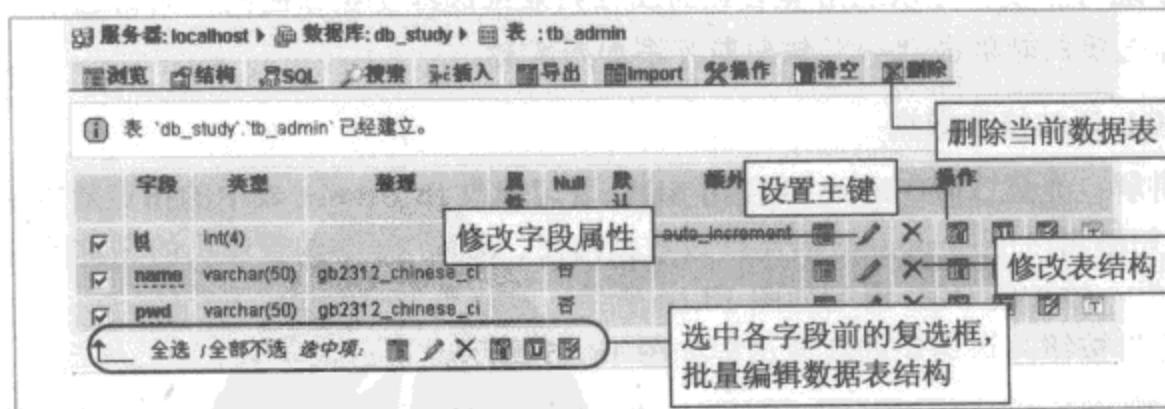


图 18.11 修改数据表结构

## 3. 删除数据表

要删除某个数据表，首先在左侧的下拉列表框中选择该数据库，在指定的数据库中选择要删除的数据表，然后单击右侧界面中的 超链接（如图 18.11 所示）即可成功删除指定的数据表。

### 18.3.3 管理数据记录

单击 phpMyAdmin 主界面中的 **SQL** 超链接，打开 SQL 语句编辑区。在编辑区输入完整的 SQL 语句，来实现数据的查询、添加、修改和删除操作。

#### 1. 使用 SQL 语句插入数据

##### 视频讲解：光盘\TM\Video\18\使用 SQL 语句向 tb\_admin 表中插入数据.exe

在 SQL 语句编辑区应用 insert 语句向数据表 tb\_admin 中插入数据后，单击“执行”按钮，向数据表中插入一条数据，如图 18.12 所示。如果提交的 SQL 语句有错误，系统会给出一个警告，提示用户修改它；如果提交的 SQL 语句正确，则弹出如图 18.13 所示的提示信息。

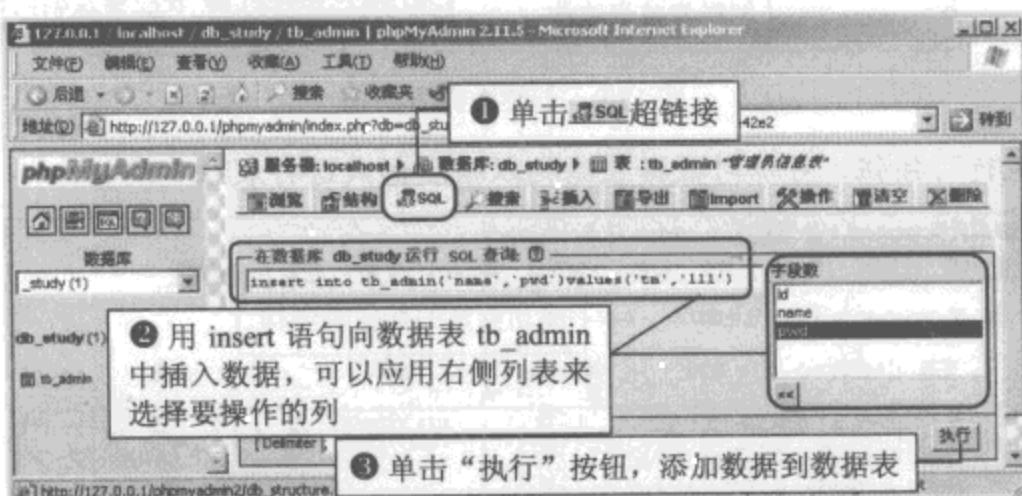


图 18.12 使用 SQL 语句向数据表中插入数据

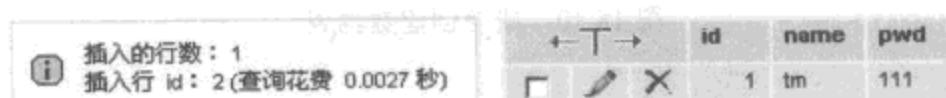


图 18.13 成功添加数据信息

技巧：为了编写方便，可以利用其右侧的属性列表来选择要操作的列，只需选中要添加的列，双击其选项或者单击“<<”按钮即可添加列名称。

#### 2. 使用 SQL 语句修改数据

##### 视频讲解：光盘\TM\Video\18\使用 SQL 语句修改 tb\_admin 表中的指定数据.exe

在 SQL 语句编辑区应用 update 语句修改数据信息，将 id 为 1 的管理员的名称改为“纯净水”，密码改为“111”，添加的 SQL 语句如图 18.14 所示。

单击“执行”按钮，该语句的实现过程如图 18.15 所示。

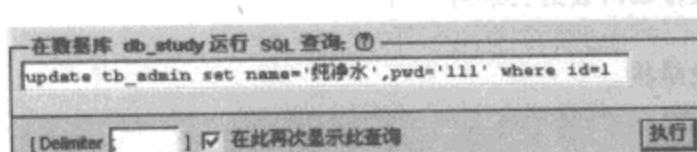


图 18.14 添加修改数据信息的 SQL 语句

←↑→ id name psw			修改 id=1 的数据 → ←↑→ id name psw		
1	tm	111	< 数据更新前 >		
< 数据更新后 >			1 纯净水 111		

图 18.15 修改单条数据的实现过程

### 3. 使用 SQL 语句查询数据

**视频讲解：**光盘\TM\Video\18\使用 SQL 语句查询 tb\_admin 表中指定条件的数据.exe

在 SQL 语句编辑区应用 select 语句检索指定条件的数据信息，将 id 小于 4 的管理员全部显示出来，添加的 SQL 语句如图 18.16 所示。

单击“执行”按钮，该语句的实现过程如图 18.17 所示。

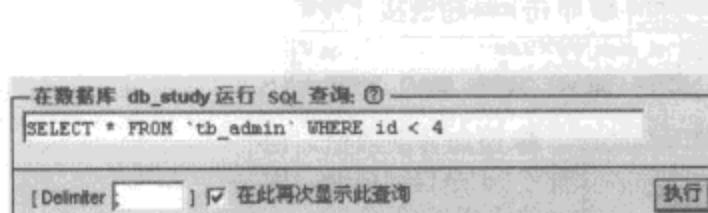


图 18.16 添加查询数据信息的 SQL 语句



图 18.17 查询指定条件的数据信息的实现过程

除了对整个表的简单查询外，还可以执行复杂的条件查询（使用 where 子句提交 LIKE、ORDER BY、GROUP BY 等条件查询语句）及多表查询，读者可通过上机进行实践，练习灵活运用 SQL 语句功能。

### 4. 使用 SQL 语句删除数据

**视频讲解：**光盘\TM\Video\18\使用 SQL 语句删除 tb\_admin 表中指定条件的数据.exe

在 SQL 语句编辑区应用 delete 语句检索指定条件的数据或全部数据信息，删除名称为 tm 的管理员信息，添加的 SQL 语句如图 18.18 所示。

**注意：**如果 delete 语句后面没有 where 条件值，那么将删除指定数据表中的全部数据。

单击“执行”按钮，弹出确认删除操作对话框，单击“确定”按钮，执行数据表中指定条件的删除操作。该语句的实现过程如图 18.19 所示。

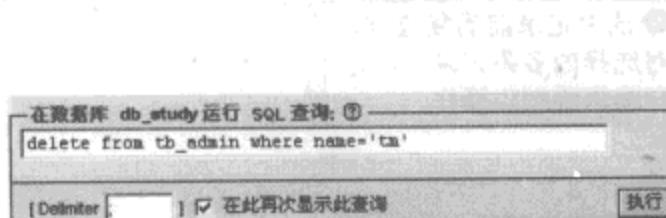


图 18.18 添加删除指定数据信息的 SQL 语句

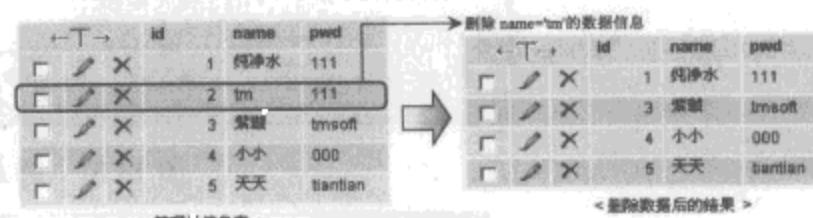


图 18.19 删除指定条件的数据信息的实现过程

### 5. 通过 form 表单插入数据

选择某个数据表后，单击 超链接，进入插入数据界面，如图 18.20 所示。在界面中输入各字段值，单击“执行”按钮即可插入记录。默认情况下，一次可以插入两条记录。

### 6. 浏览数据

选择某个数据表后，单击 超链接，进入浏览界面，如图 18.21 所示。单击每行记录中的 按钮，可以对该记录进行编辑；单击每行记录中的 按钮，可以删除该条记录。

### 7. 搜索数据

选择某个数据表后，单击 超链接，进入搜索页面，如图 18.22 所示。在这个页面中，可以在选择字段的列表框中选择一个或多个列，如果要选择多个列，先按 Ctrl 键并单击要选择的字段名，查询结果将按照选择的字段名进行输出。

在该界面中可以对记录按条件进行查询。查询方式有两种：第一种方式是选择构建 where 语句查询，直接在“where 语句的主体”文本框中输入查询语句，然后单击其后的“执行”按钮；第二种方式是使用按例查询，选择查询的条件，并在文本框中输入要查询的值，单击“执行”按钮。

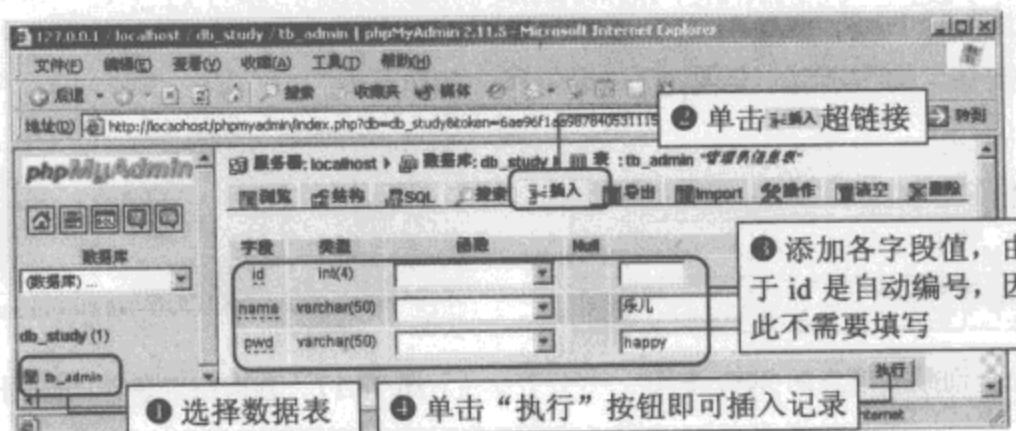


图 18.20 插入数据

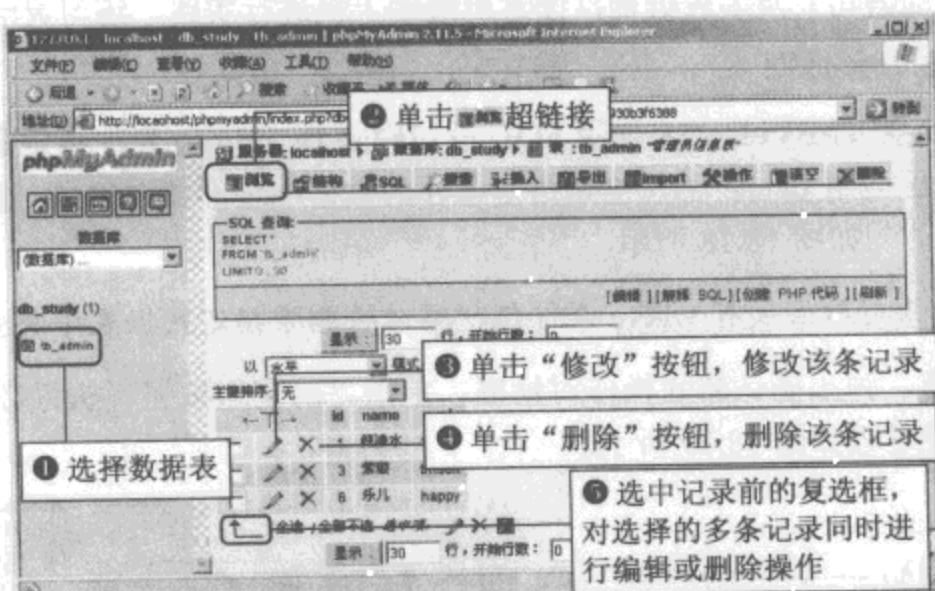


图 18.21 浏览数据

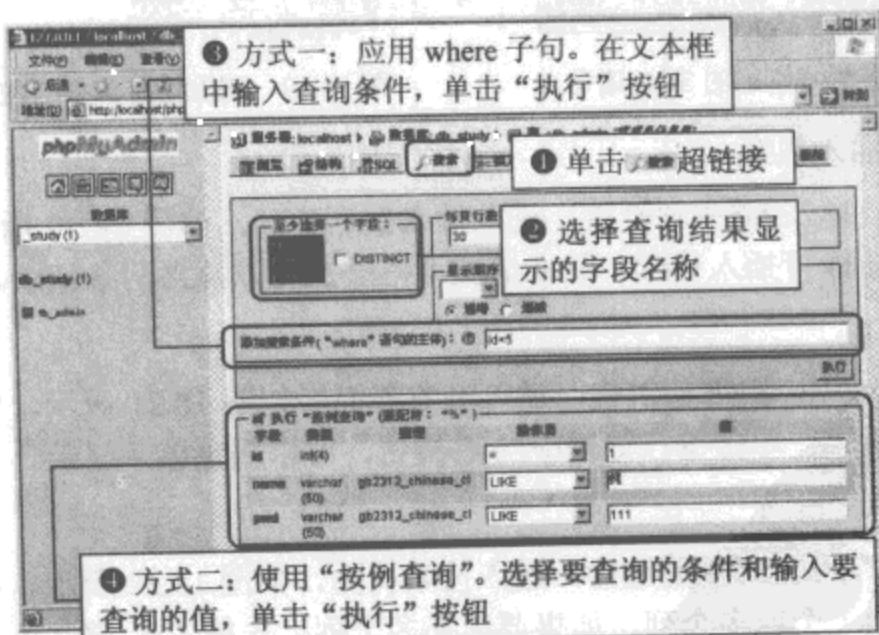


图 18.22 搜索查询

### 18.3.4 导入导出数据

#### ■ 视频讲解：光盘\TM\Video\18\导入导出数据.exe

导入和导出 MySQL 数据库脚本是互逆的两个操作。导入是执行.sql 文件，将数据导入到数据库中；导出是将数据表结构、表记录存储为.sql 脚本文件。通过导入和导出的操作可以实现数据库的备份和还原。

#### 1. 导出 MySQL 数据库脚本

单击 phpMyAdmin 主界面中的“导出”超链接，打开导出编辑区，如图 18.23 所示。选择导出文件的格式，这里默认使用选项 SQL，选中“另存为文件”复选框，单击“执行”按钮，弹出如图 18.24 所示的“文件下载”对话框，单击“保存”按钮，将脚本文件以.sql 格式存储在指定位置。

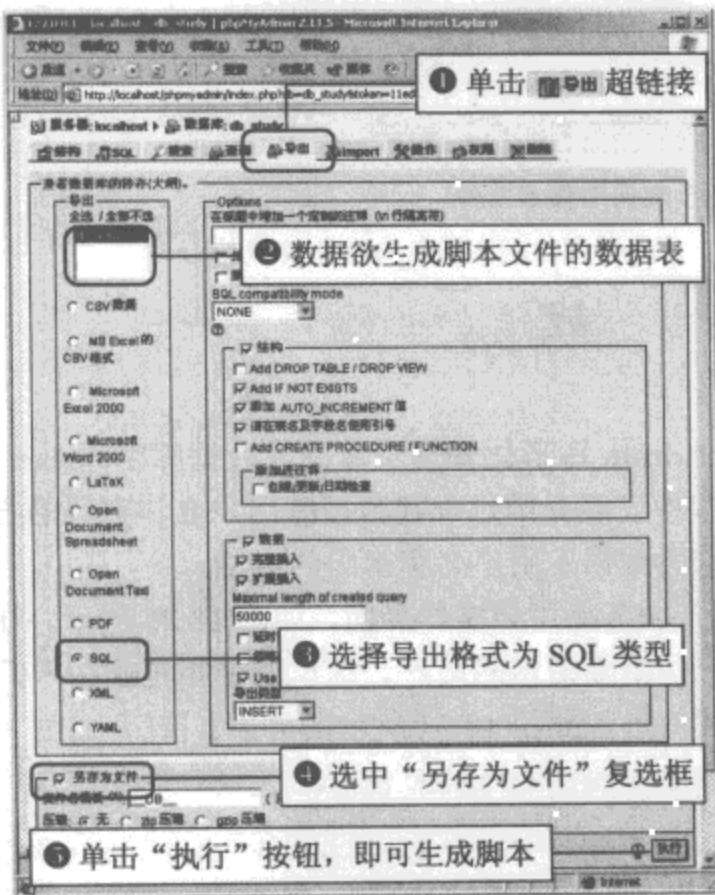


图 18.23 生成 MySQL 脚本文件设置界面

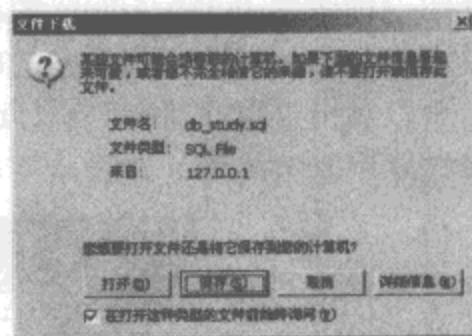


图 18.24 存储 MySQL 脚本对话框

#### 2. 导入 MySQL 数据库脚本

单击“Import”超链接，进入执行 MySQL 数据库脚本界面，单击“浏览”按钮查找脚本文件（如 db\_study.sql）所在位置，如图 18.25 所示，单击“执行”按钮，即可执行 MySQL 数据库脚本文件。

**◆ 注意：**在执行 MySQL 脚本文件前，首先检测是否有与所导入数据库同名的数据库，如果没有同名的数据库，则首先要在数据库中创建一个名称与数据文件中的数据库名相同的数据库，然后再执行 MySQL 数据库脚本文件。另外，在当前数据库中，不能有与将要导入数据库中的数据表重名的数据表存在，如果有重名的表存在，导入文件就会失败，并提示错误信息。

**说明：**读者也可通过单击 phpMyAdmin 图形化工具左侧的 按钮，在打开的对话框中单击“导入文件”超链接，然后选择脚本文件所在的位置，从而执行脚本文件。

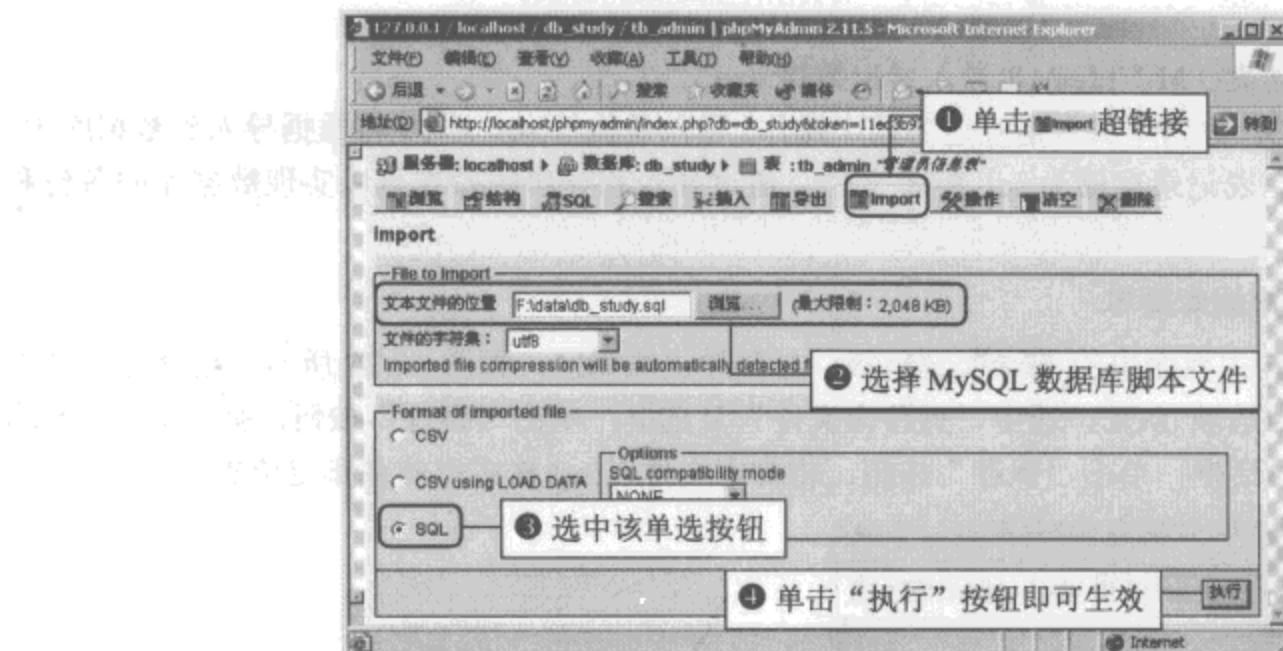


图 18.25 执行 MySQL 数据库脚本文件

## 18.4 实战

### 视频讲解：光盘\TM\Video\18\通过 phpMyAdmin 图形化管理工具修改数据库密码.exe

前文在 phpMyAdmin 图形化管理工具安装的过程中已经介绍过设置数据库用户名和密码的方法，下面介绍一种通过 phpMyAdmin 图形化管理工具来修改数据库用户名和密码的方法。

单击 phpMyAdmin 主界面中的 超链接，打开服务器用户操作界面，如图 18.26 所示。在该界面中，可以对指定用户的权限进行编辑，添加新用户和删除指定的用户。这里选择指定的用户，单击“编辑权限”超链接，对指定用户的权限进行设置，进入如图 18.27 所示的界面。

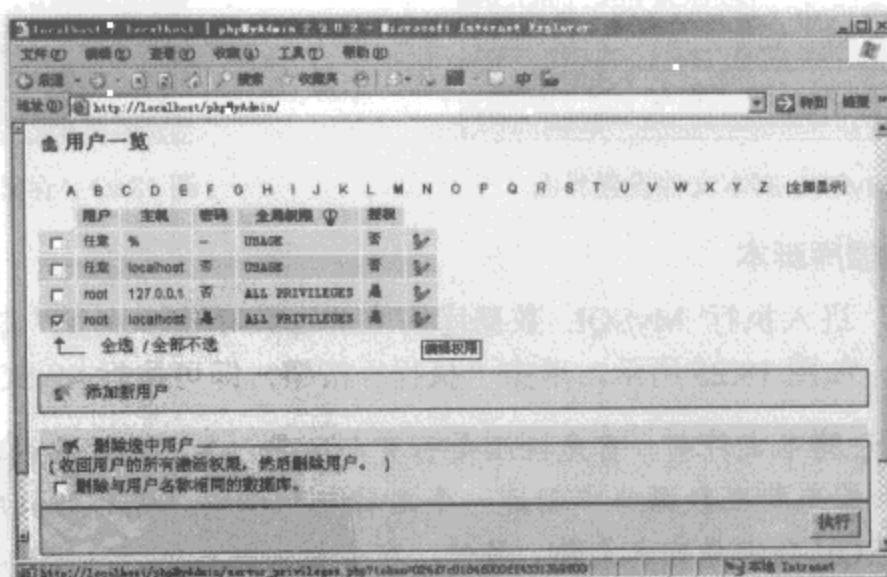


图 18.26 服务器用户一览表

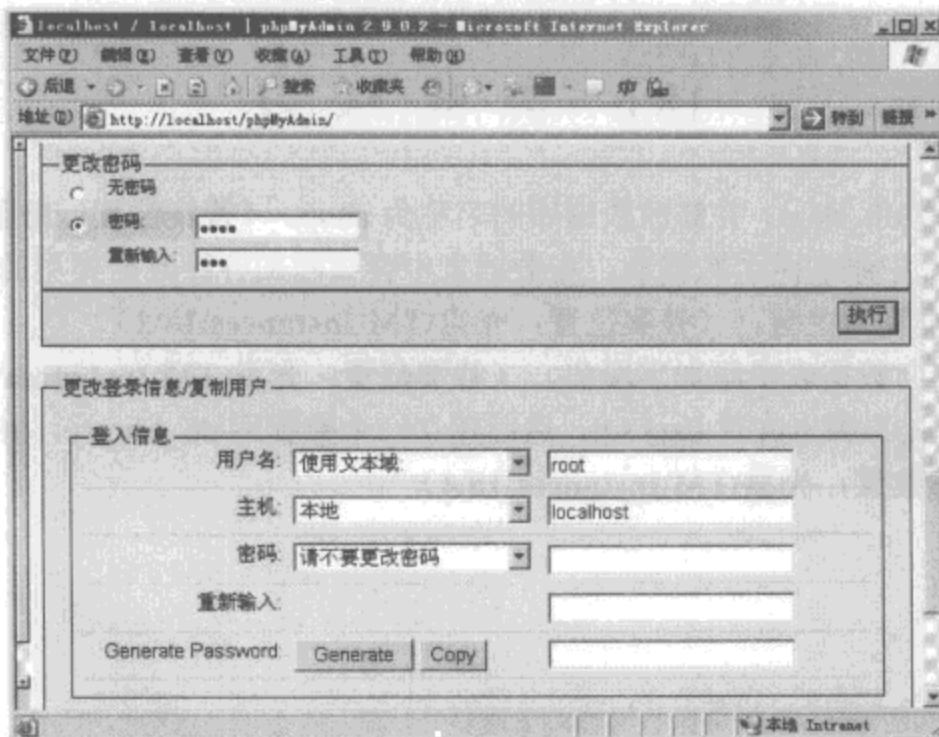


图 18.27 编辑用户权限

在图 18.27 所示的界面中，可以设置用户的权限、修改密码、更改登录信息/复制用户。在输入原密码和新密码之后，单击“执行”按钮即可完成对用户密码的修改操作，返回主页面，如图 18.28 所示。

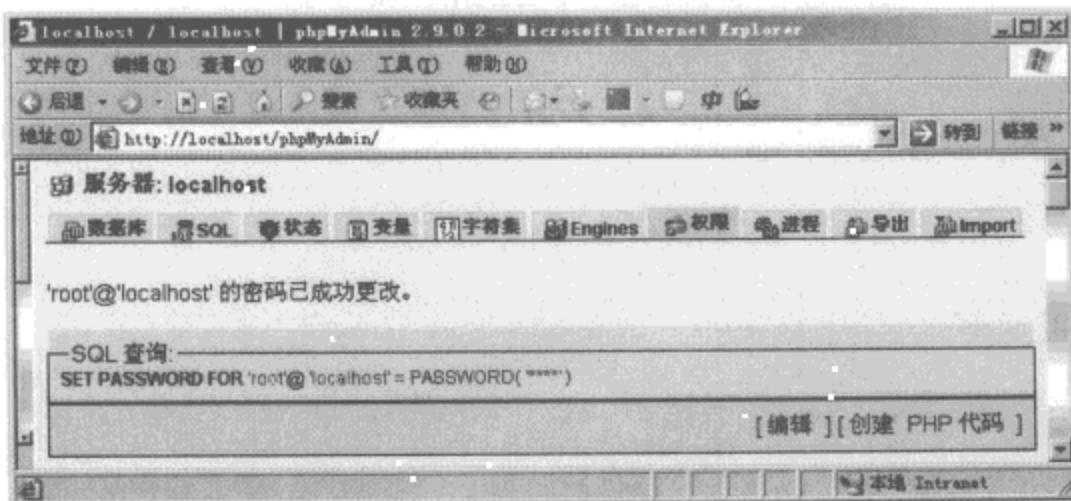


图 18.28 用户密码修改成功

## 18.5 本章小结

phpMyAdmin 是 MySQL 数据库管理和操作的可视化工具，可以方便地对 MySQL 数据库进行管理。通过本章的学习，读者可以独立安装和配置 phpMyAdmin V2.11.5 版本，并且可以突破在命令提示符下创建数据库和数据表的瓶颈，应用可视化的工具 phpMyAdmin 轻松进行管理数据库和数据表等相关操作，还可以对大型的网站通过导入和导出 MySQL 数据库脚本来维护网站数据库。

## 18.6 实战练习

1. 创建一个数据库 db\_shop，并修改数据库的名称为 shop。（答案位置：光盘\TM\Instances\18\1）
2. 在数据库 shop 中添加两个数据表，在数据表中尝试添加各种数据类型的字段，设置每个表中的 id 为自动编号，并设置为主键。（答案位置：光盘\TM\Instances\18\2）
3. 应用 SQL 语句向数据表中添加字段值。（答案位置：光盘\TM\Instances\18\3）
4. 将数据库生成 SQL 脚本文件 data.sql，然后建立一个数据库 db\_library，将生成的脚本文件导入到该数据表中。（答案位置：光盘\TM\Instances\18\4）

# 第19章

## 使用 MySQL 函数操作数据库 ( 视频讲解: 182 分钟)

PHP 所支持的数据库类型较多，在这些数据库中，MySQL 数据库与 PHP 的兼容最好，与 Linux 系统、Apache 服务器和 PHP 语言构成了当今主流的 LAMP 网站架构模式，并且 PHP 提供了多种操作 MySQL 数据库的方式，可以适合不同需求和不同类型项目的需要。本章将介绍如何通过 MySQL 函数操作数据库，这也是中小型项目常用的方式之一。

通过阅读本章，可以学习到：

- PHP 访问 MySQL 数据库的步骤
- 常用 MySQL 数据库函数的使用方法
- PHP 操作 MySQL 数据库进行增、删、改、查操作
- 数据分页显示的实现方法
- 开发数据库项目基本流程

## 19.1 PHP 访问 MySQL 数据库的一般步骤

和其他语言类似，PHP 操作 MySQL 数据库的过程一般分为 5 步，分别为连接 MySQL 数据库服务器、选择数据库、执行 SQL 语句、关闭结果集以及断开与 MySQL 服务器的连接，如图 19.1 所示。下面将具体介绍其实现过程。

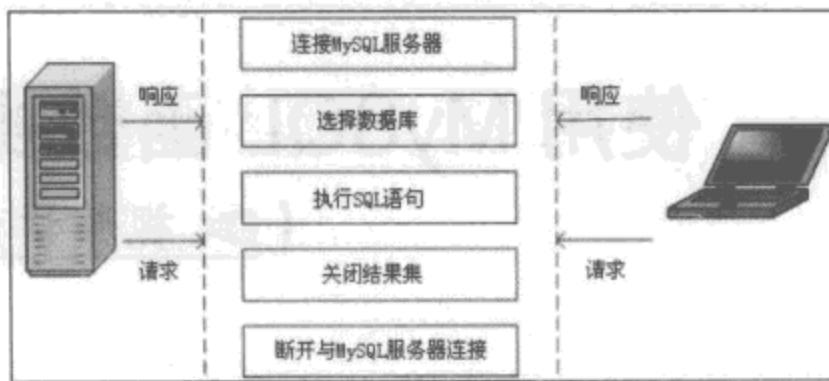


图 19.1 PHP 访问 MySQL 数据库的一般步骤

### 1. 连接 MySQL 服务器

应用 `mysql_connect()` 函数建立与 MySQL 服务器的连接，并返回一个连接标识，在以后对 MySQL 服务器进行操作时，可以根据这个连接标识定位不同的连接。

### 2. 选择数据库

应用 `mysql_select_db()` 函数选择 MySQL 数据库服务器上的数据库，并与该数据库建立连接。

### 3. 执行 SQL 语句

在选择的数据库中应用 `mysql_query()` 函数执行 SQL 语句。对数据的操作主要包括以下 5 种方式。

- 查询数据：应用 `select` 语句实现数据的查询功能。
- 显示数据：应用 `select` 语句显示数据的查询结果。
- 插入数据：应用 `insert` 语句向数据库中插入数据。
- 更新数据：应用 `update` 语句修改数据库中的记录。
- 删除数据：应用 `delete` 语句删除数据库中的记录。

### 4. 关闭结果集

数据库操作完成后，需要关闭结果集，以释放系统资源。

```
mysql_free_result($result);
```

### 5. 断开与 MySQL 服务器的连接

每使用一次 `mysql_connect()` 或 `mysql_query()` 函数，都会消耗系统资源。这在少量用户访问 Web 网站时影响不明显，但如果用户连接超过一定数量，就会造成系统性能的下降，甚至死机。为了避免这种现象的发生，在完成数据库的操作后，可以应用 `mysql_close()` 函数关闭与 MySQL 服务器的连接，以节省系统资源。

```
mysql_close($link);
```

## 19.2 使用 MySQL 函数库

19.1 节简要介绍了 PHP 访问和操作 MySQL 数据库的一般步骤。PHP 提供了大量操作 MySQL 数据库的方式，其中函数方式相对简便并且应用较广泛。本节将详细介绍 PHP 中 MySQL 相关函数的使用方法。

### 19.2.1 连接 MySQL 服务器

视频讲解：光盘\TM\Video\19\连接 MySQL 服务器.exe

PHP 操作 MySQL 数据库，首先要建立与 MySQL 数据库的连接，PHP 实现与数据库连接相对简便，只需使用 `mysql_connect()` 函数即可，函数语法如下：

```
resource mysql_connect ([string server [, string username [, string password [, bool new_link [, int client_flags]]]]])
```

`mysql_connect()` 函数用于打开一个到 MySQL 服务器的连接，如果成功则返回一个 MySQL 连接标识，失败则返回 `false`。该函数的参数如表 19.1 所示。

表 19.1 `mysql_connect()` 函数的参数说明

参 数	说 明
server	MySQL 服务器。可以包括端口号，如"hostname:port"；或者到本地套接字的路径，如对于 <code>localhost</code> 的":/path/to/socket"。如果 PHP 指令 <code>mysql.default_host</code> 未定义（默认情况），则默认值是'localhost:3306'
username	用户名。默认值是服务器进程所有者的用户名
password	密码。默认值是空密码
new_link	如果用同样的参数再次调用 <code>mysql_connect()</code> 函数，将不会建立新连接，而将返回已经打开的连接标识。参数 <code>new_link</code> 改变此行为并使 <code>mysql_connect()</code> 函数总是打开新的连接，即使 <code>mysql_connect()</code> 函数曾在前面被用同样的参数调用过
client_flags	<code>client_flags</code> 参数可以是以下常量的组合： <code>MYSQL_CLIENT_SSL</code> , <code>MYSQL_CLIENT_COMPRESS</code> , <code>MYSQL_CLIENT_IGNORE_SPACE</code> 或 <code>MYSQL_CLIENT_INTERACTIVE</code>

例 19.1 应用 `mysql_connect()` 函数创建与 MySQL 服务器的连接，MySQL 数据库服务器地址为 127.0.0.1，用户名为 root，密码为 root，代码如下：（实例位置：光盘\TM\Instances\19\1）

```
<?php
$host = "127.0.0.1"; //MySQL 服务器地址
$userName = "root"; //用户名
$password = "root"; //密码
if ($connID = mysql_connect($host, $userName, $password)){
    //建立与 MySQL 数据库的连接，并弹出提示对话框
    echo "<script language='javascript'>alert('数据库连接成功！');</script>";
} else{
    echo "<script language='javascript'>alert('数据库连接失败！');</script>";
}
?>
```

运行上述代码，如果在本地计算机中安装了 MySQL 数据库，并且 root 用户的用户名也为 root，则会弹出如图 19.2 所示的对话框。



图 19.2 数据库连接成功

**技巧：**在项目完成后，为了屏蔽由于数据库连接失败而显示的不友好的错误信息，可以在 mysql\_connect() 函数前加 “@”，该符号用来屏蔽错误提示。

## 19.2.2 选择数据库

**视频讲解：**光盘\TM\Video\19\选择数据库.exe

成功与 MySQL 数据库建立连接后，需要选择 MySQL 数据库服务器中指定的数据库。PHP 中使用 mysql\_select\_db() 函数实现数据库的选择功能，该函数的语法格式如下：

```
bool mysql_select_db ( string database_name [, resource link_identifier] )
```

mysql\_select\_db() 函数用于设定与指定的连接标识符所关联的服务器上的当前激活数据库。如果没有指定连接标识符，则使用上一个打开的连接。如果没有打开的连接，本函数将无参数调用 mysql\_connect() 函数来尝试打开一个并使用。其后的每个 mysql\_query() 函数调用都会作用于当前激活数据库。该函数的参数说明如表 19.2 所示。

表 19.2 mysql\_select\_db() 函数的参数说明

参 数	说 明
database name	必要参数，用户指定要选择的数据库名称
link identifier	可选参数，数据库连接 ID，如果省略该参数，则默认为最近一次与数据库建立的连接

**例 19.2** 首先使用 mysql\_connect() 函数建立与 MySQL 数据库的连接并返回数据库连接 ID，然后使用 mysql\_select\_db() 函数选择 MySQL 数据库服务器中名为 db\_database19 的数据库，实现代码如下：  
(实例位置：光盘\TM\Instances\19\2)

```
<?php
$host = "127.0.0.1"; //MySQL 服务器地址
$userName = "root"; //用户名
$password = "root"; //密码
$dbName = "db_database19"; //数据库
$connID = mysql_connect($host, $userName, $password); //建立与 MySQL 数据库服务器的连接
if(mysql_select_db($dbName, $connID)){ //选择数据库
    echo "数据库选择成功!";
} else{
    echo "数据库选择失败!";
}
```

```

}
?>

```

运行上述代码，如果本地 MySQL 数据库服务器中存在名为 db\_database19 的数据库，将在页面中显示如图 19.3 所示的提示信息。

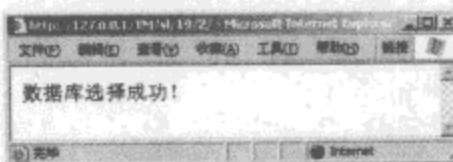


图 19.3 数据库选择成功

### 19.2.3 执行 SQL 语句

#### 视频讲解：光盘\TM\Video\19\执行 SQL 语句.exe

成功选择 MySQL 数据库服务器中的数据库后，即可对所选数据库中的数据表进行查询、更改以及删除等操作，PHP 使用 mysql\_query() 函数就可以实现上述所有操作，操作极其简便，说明在 PHP 底层进行了复杂的封装，而提供给上层开发人员一种简便的编程模式，这也是 PHP 操作简便的体现和应用广泛的原因。mysql\_query() 函数的语法格式如下：

```
resource mysql_query ( string query [, resource link_identifier] )
```

mysql\_query() 函数用于执行一条查询语句，该函数的参数说明如表 19.3 所示。

表 19.3 mysql\_query() 函数的参数说明

参 数	说 明
query	字符串类型，传入的是 SQL 的指令
link_identifier	资源类型，传入的是由 mysql_connect() 函数或 mysql_pconnect() 函数返回的连接号。如果省略该参数，则会使用最后一个打开的 MySQL 数据库连接

例 19.3 查询学生信息表中学生的成绩信息，代码如下：（实例位置：光盘\TM\Instances\19\3）

```

<?php
$host = "127.0.0.1";                                //MySQL 数据库服务器
$userName = "root";                                    //用户名
$password = "root";                                    //密码
$dbName = "db_database19";                            //数据库名
$connID = mysql_connect($host, $userName, $password); //连接 MySQL 数据库
mysql_select_db($dbName, $connID);                     //选择 MySQL 数据库
mysql_query("set names gbk");                         //设置字符集
echo "<table border=\"1px\" align=\"center\">
      <tr>
        <td>学号</td>
        <td>姓名</td>
        <td>班级</td>
        <td>语文</td>
        <td>数学</td>
      </tr>
    <tr>
      <td>1001</td>
      <td>李华</td>
      <td>100班</td>
      <td>95</td>
      <td>98</td>
    </tr>
  </table>";

```

```

<td>英语</td>
</tr>";
$query = mysql_query("select sno, sname, class, chinese, math ,english from tb_student", $connID);
//执行查询
while($result = mysql_fetch_array($query))
//获取结果集并输出查询结果
{
    echo "<tr>
        <td>".$result["sno"]."</td>
        <td>".$result["sname"]."</td>
        <td>".$result["class"]."</td>
        <td>".$result["chinese"]."</td>
        <td>".$result["math"]."</td>
        <td>".$result["english"]."</td>
    </tr>";
}
echo "</table>";
?>

```

运行上述实例，结果如图 19.4 所示。

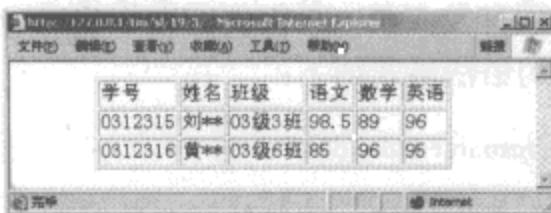


图 19.4 查询学生成绩

#### 19.2.4 应用 mysql\_fetch\_array()函数从数组结果集中获取信息

视频讲解：光盘\TM\Video\19\应用 mysql\_fetch\_array()函数获取结果集.exe

使用 mysql\_query()函数执行查询后，并不能返回查询结果，那么如何才能获取查询结果呢？PHP 提供了 mysql\_fetch\_array()函数实现获取查询结果集的功能，该函数的语法格式如下。

```
array mysql_fetch_array ( resource result [, int result_type] )
```

mysql\_fetch\_array()函数的参数说明如表 19.4 所示。

表 19.4 mysql\_fetch\_array()函数的参数说明

参 数	说 明
result	资源类型的参数，要传入的是由 mysql_query()函数返回的“数据指针”
result_type	可选项，整数型参数，要传入的是 MYSQL_ASSOC、MYSQL_NUM、MYSQL_BOTH 3 种由 PHP 定义好的常数之一，默认值是 MYSQL_BOTH 用 MYSQL_ASSOC 只得到关联索引（相当于 mysql_fetch_assoc()函数） 用 MYSQL_NUM 只得到数字索引（相当于 mysql_fetch_row()函数） 用 MYSQL_BOTH 将得到一个同时包含关联和数字索引的数组

例 19.4 按员工编号以模糊查询的方式查询员工信息，并显示全部查询结果。（实例位置：光盘\TM\

**Instances\19\4)**

具体实现步骤如下：

(1) 建立与 MySQL 数据库的连接，并返回数据库连接 ID，代码如下。

```
<?php
$connID=mysql_connect("localhost","root","root");           //建立与数据库的连接
mysql_select_db("db_database19", $connID);                  //选择数据库
mysql_query("set names gbk");                                //设置字符集
?>
```

(2) 建立查询信息录入表单，表单及表单元素如表 19.5 所示。

表 19.5 员工信息录入表单

元素类型	元素名称	属性设置	说明
表单	form1	name="form1" method="post" action="<?php echo \$_SERVER['PHP_SELF']?>"	表单
文本域	number	name="number" type="text" id="number"	录入员工编号
隐藏域	flag	type="hidden" name="flag" value="1"	判断表单是否提交
提交按钮	submit	name="submit" type="submit" value="提交"	提交按钮

(3) 使用\$\_POST 全局数组接收表单提交的 flag 元素的值，并使用 iset() 函数判断是否已经设置了该元素的值，如果已设置则说明已经提交了表单，然后采用模糊查询的方式查询所有与查询关键字相匹配的员工信息，并使用 while 循环将查询结果显示出来，代码如下。

```
<?php
if(isset($_POST["flag"])){
    $query=mysql_query("select * from tb_employee where number like '%".$_POST["number"]."%'");
    if($query){
        while($myrow=mysql_fetch_array($query)){
?>
            <tr>
                <td align="center" bgcolor="#FFFFFF" class="STYLE4"><span class="STYLE2"><?php echo $myrow[number];?>
            </span></td>
                <td align="center" bgcolor="#FFFFFF" class="STYLE4"><span class="STYLE2"><?php echo $myrow[name];?>
            </span></td>
                <td height="23" align="center" bgcolor="#FFFFFF" class="STYLE4"><span class="STYLE2"><?php echo $myrow
[tel];?></span></td>
                <td height="23" align="center" bgcolor="#FFFFFF" class="STYLE4"><span class="STYLE2"><?php echo $myrow
[address];?></span></td>
            </tr>
        <?php
    }
}
?>
```

运行该实例，在员工查询信息录入表单中输入员工编号，然后单击“提交”按钮，即可以模糊查询的方式查询出所有与查询关键字相匹配的员工信息，如图 19.5 所示。

The screenshot shows a search interface titled "查询企业员工的详细信息". It has a search input field with value "100" and a submit button. Below it is a table with columns: 编号 (ID), 姓名 (Name), 电话 (Phone), and 地址 (Address). The table contains 7 rows of data.

编号	姓名	电话	地址
1004	潘**	12345678	长春
1005	刘**	87456123	四平
1001	王**	12345645	吉林
1002	李**	1231233	沈阳
1006	欧阳*	12345678	通化
1007	独孤**	45612312	大理

图 19.5 查询员工信息

### 19.2.5 应用 mysql\_fetch\_object()函数从结果集中获取一行作为对象

视频讲解：光盘\TM\Video\19\应用 mysql\_fetch\_object()函数获取结果集.exe

19.2.4 节中讲解了应用 mysql\_fetch\_array() 函数来获取结果集中的数据。除了这个方法以外，应用 mysql\_fetch\_object() 函数也可以轻松实现这一功能，下面通过同一个实例的不同方法来体验一下这两个函数在使用上的区别。首先介绍 mysql\_fetch\_object() 函数。

语法：

```
object mysql_fetch_object ( resource result )
```

mysql\_fetch\_object() 函数和 mysql\_fetch\_array() 函数类似，只有一点区别：返回一个对象而不是数组，即该函数只能通过字段名来访问数组。访问结果集中行的元素的语法结构如下：

```
$row->col_name //col_name 为列名, $row 代表结果集
```

例如，如果从某数据表中检索 id 和 name 值，可以用 \$row->id 和 \$row->name 访问行中的元素值。

注意：本函数返回的字段名是区分大小写的，这是初学者学习时最容易忽视的问题。

例 19.5 使用 mysql\_fetch\_object() 函数获取查询到图书的信息。（实例位置：光盘\TM\Instances\19\5）  
具体开发步骤如下：

(1) 建立图书查询表单，表单及表单元素说明如表 19.6 所示。

表 19.6 图书信息查询表单及表单说明

元素类型	元素名称	属性设置	说明
表单	myform	name="myform" method="post" action=""	表单
文本域	txt_book	name="txt_book" type="text" id="txt_book" size="25"	查询关键字
提交按钮	submit	type="submit" name="Submit" value="查询"	查询按钮

(2) 建立与 MySQL 数据库的连接、设置字符集，并返回数据库连接 ID，代码如下。

```
$link=mysql_connect("localhost","root","root") or die("数据库连接失败".mysql_error()); //建立与数据库的连接
mysql_select_db("db_database19",$link); //选择数据库
mysql_query("set names gb2312"); //设置字符集
```

(3) 应用 mysql\_query() 函数执行 SQL 查询语句，并使用 mysql\_fetch\_object() 函数获取查询语句

的结果集，代码如下。

```
if($_POST[Submit]=="查询"){
    $txt_book=$_POST[txt_book]; //接收查询关键字
    $sql=mysql_query("select * from tb_book where bookname like '%".trim($txt_book)."%'");
    //如果选择的条件为“like”，则进行模糊查询
    $info=mysql_fetch_object($sql);
}
```

(4) 应用 do...while 循环语句以对象的方式输出结果集中的图书信息到浏览器中，代码如下。

```
do{
?>
<tr align="left" bgcolor="#FFFFFF">
    <td height="20" align="center"><?php echo $info->id; ?></td>
    <td>&ampnbsp<?php echo $info->bookname; ?></td>
    <td align="center"><?php echo $info->issuDate; ?></td>
    <td align="center"><?php echo $info->price; ?></td>
    <td align="center">&ampnbsp<?php echo $info->maker; ?></td>
    <td>&ampnbsp<?php echo $info->publisher; ?></td>
</tr>
<?php
}while($info=mysql_fetch_object($sql));
```

保存 index.php 动态页，在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 19.6 所示。

请输入图书名称 平装					
编号	图书名称	出版时间	图书定价	作者	出版社
9	PHP 网络编程自学手册	2008-03-01	52	明日科技	人民邮电出版社
6	PHP 数据库系统开发完全手册	2001-06-01	52	明日科技	人民邮电出版社

图 19.6 查询图书信息

## 19.2.6 应用 mysql\_fetch\_row()函数逐行获取结果集中的每条记录

视频讲解：光盘\TM\Video\19\应用 mysql\_fetch\_row()函数获取结果集.exe

前面介绍了应用 mysql\_fetch\_array()函数和 mysql\_fetch\_object()函数来获取结果集中的数据。本节向读者介绍第 3 种方法，应用 mysql\_fetch\_row()函数逐行获取结果集中的每条记录。首先来了解 mysql\_fetch\_row()函数。

语法：

```
array mysql_fetch_row ( resource result )
```

mysql\_query()函数将查询语句发送到服务器中执行。查询语句不使用分号终止。如果查询非法或由于某些原因不能执行，则 mysql\_query()函数返回 false，否则返回一个结果集标识符。mysql\_fetch\_row()函数从和指定的结果标识关联的结果集中获取一行数据并作为数组返回，将此行赋给变量\$row，每个结果的列储存在一个数组的单元中，偏移量从 0 开始，即以\$row[0]的形式访问第一个元素（只有一个

元素时也是如此)。依次调用 `mysql_fetch_row()` 函数将返回结果集中的下一行，直到没有更多行则返回 `false`。

**例 19.6** 查询图书信息，并使用 `mysql_fetch_row()` 函数获取结果集显示图书信息。(实例位置：光盘\TM\Instances\19\6)

具体开发步骤如下：

(1) 创建项目、添加表单、连接 MySQL 服务器以及设置默认数据库的实现过程与例 19.5 开发步骤中的(1)~(2)相同，这里不再赘述。

(2) 在应用 `mysql_query()` 函数执行 SQL 查询语句后，与实例 19.5 不同的是，本实例使用 `mysql_fetch_row()` 函数获取查询语句的结果集。代码如下。

```
<?php
$sql=mysql_query("select * from tb_book");
$row=mysql_fetch_row($sql);
if($_POST[Submit]=="查询"){
$txt_book=$_POST[txt_book];
//如果选择的条件为“like”，则进行模糊查询
$sql=mysql_query("select * from tb_book where bookname like '%".trim($txt_book)."%'");
$row=mysql_fetch_row($sql);
}
?>
```

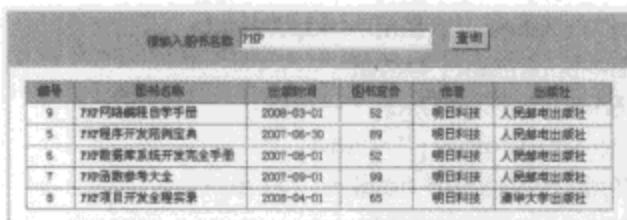
(3) 应用 `if` 条件语句对结果集变量 `$info` 进行判断，如果该值为假，则检索的图书信息不存在，应用 `echo` 语句输出提示信息，代码如下。

```
<?php
if($row==false){          //如果检索的信息不存在，则输出相应的提示信息
    echo "<div align='center' style='color:#FF0000; font-size:12px>对不起，您检索的图书信息不存在!</div>";
}
?>
```

(4) 应用 `do…while` 循环语句以对象的方式输出结果集中的图书信息到浏览器中，代码如下。

```
<?php
do{
?>
<tr align="left" bgcolor="#FFFFFF">
<td height="20" align="center"><?php echo $row[0]; ?></td>
<td>&ampnbsp<?php echo $row[1]; ?></td>
<td align="center"><?php echo $row[2]; ?></td>
<td align="center"><?php echo $row[3]; ?></td>
<td align="center">&ampnbsp<?php echo $row[4]; ?></td>
<td>&ampnbsp<?php echo $row[5]; ?></td>
</tr>
<?php
}while($row=mysql_fetch_row($sql));
?>
```

保存 `index.php` 动态页，在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 19.7 所示。



The screenshot shows a MySQL query results window. At the top, there is a search bar with the text "请输入图书名称: PHP" and a "查询" button. Below the search bar is a table with the following data:

编号	图书名称	出版时间	图书定价	作者	出版社
9	PHP 项目实践教学手册	2008-03-01	52	明日科技	人民邮电出版社
5	PHP 程序开发指南宝典	2007-06-30	89	明日科技	人民邮电出版社
6	PHP 数据库系统开发完全手册	2007-06-01	52	明日科技	人民邮电出版社
7	PHP 函数参考大全	2007-09-01	99	明日科技	人民邮电出版社
8	PHP 项目开发全理解读	2008-04-01	65	明日科技	清华大学出版社

图 19.7 使用 mysql\_fetch\_row() 函数获取结果集查询图书信息

### 19.2.7 应用 mysql\_num\_rows() 函数获取查询结果集中的记录数

▶ 视频讲解：光盘\TM\Video\19\应用 mysql\_num\_rows() 函数获取记录数.exe

要获取由 select 语句查询到的结果集中行的数目，则必须使用 mysql\_num\_rows() 函数实现。首先来看一下该函数的语法结构。

语法：

```
int mysql_num_rows ( resource result )
```

▶ 注意：使用 mysql\_unbuffered\_query() 函数查询到的数据结果，无法使用 mysql\_num\_rows() 函数来获取查询结果的记录数。

**例 19.7 使用 mysql\_num\_rows() 函数获取查询结果的记录数。(实例位置：光盘\TM\Instances\19\7)**  
具体开发步骤如下：

(1) 建立与 MySQL 数据库的连接，设置字符集为 GB2312，并返回数据库连接 ID，代码如下。

```
$link=mysql_connect("localhost","root","root") or die("数据库连接失败".mysql_error()); //建立连接
mysql_select_db("db_database19",$link); //选择数据库
mysql_query("set names gb2312"); //设置字符集
```

(2) 默认情况下显示所有图书信息，代码如下。

```
$sql=mysql_query("select * from tb_book"); //查询所有图书信息
$info=mysql_fetch_object($sql); //获取结果集
```

(3) 如果用户输入了查询关键字，并单击了“查询”按钮，则采用模糊查询的方式显示出所有符合条件的记录，代码如下。

```
if($_POST[Submit]=="查询"){
    $txt_book=$_POST[txt_book];
    $sql=mysql_query("select * from tb_book where bookname like '%".trim($txt_book)."%'");
    //如果选择的条件为“like”，则进行模糊查询
    $info=mysql_fetch_object($sql);
}
if($info==false){ //如果检索的信息不存在，则输出相应的提示信息
    echo "<div align='center' style='color:#FF0000; font-size:12px>对不起，您检索的图书信息不存在!</div>";
}
do{
?>
<tr align="left" bgcolor="#FFFFFF">
```

```

<td height="20" align="center"><?php echo $info->id; ?></td>
<td>&ampnbsp<?php echo $info->bookname; ?></td>
<td align="center"><?php echo $info->issuDate; ?></td>
<td align="center"><?php echo $info->price; ?></td>
<td align="center">&ampnbsp<?php echo $info->maker; ?></td>
<td>&ampnbsp<?php echo $info->publisher; ?></td>
</tr>
<?php
    }while($info=mysql_fetch_object($sql));
?>

```

在 IE 浏览器中输入地址，按 Enter 键，程序默认输出图书信息表中的全部图书信息，并自动汇总记录条数，如图 19.8 所示。在文本框中输入要检索的图书名称，如“开发”（支持模糊查询，程序自动去除查询关键字左右空格），单击“查询”按钮，即可按条件检索指定的图书信息到浏览器，并自动汇总检索到的记录条数，运行结果如图 19.9 所示。

请输入图书名称						查询
编号	图书名称	出版时间	图书定价	作者	出版社	
9	PHP 网络编程自学手册	2006-03-01	52	明日科技	人民邮电出版社	
5	PHP 程序开发范例宝典	2007-06-30	89	明日科技	人民邮电出版社	
6	PHP 数据库系统开发完全手册	2007-06-01	52	明日科技	人民邮电出版社	
7	PHP 语言参考大全	2007-09-01	99	明日科技	人民邮电出版社	
8	PHP 项目开发全程实录	2008-04-01	62	明日科技	清华大学出版社	

在默认页应用 mysql\_num\_rows() 函数获取数据表的总记录数

图 19.8 默认统计数据表中所有记录

请输入图书名称						查询
编号	图书名称	出版时间	图书定价	作者	出版社	
5	PHP 程序开发范例宝典	2007-06-30	89	明日科技	人民邮电出版社	
6	PHP 数据库系统开发完全手册	2007-06-01	52	明日科技	人民邮电出版社	
8	PHP 项目开发全程实录	2008-04-01	62	明日科技	清华大学出版社	

应用 mysql\_num\_rows() 函数获取查询结果集中的总记录数

图 19.9 应用 mysql\_num\_rows() 函数获取查询结果集中的总记录数

**技巧：**如果要获取由 insert、update、delete 语句所影响到的数据行数，则必须应用 mysql\_affected\_rows() 函数来实现。

## 19.2.8 关闭连接

### 视频讲解：光盘\TM\Video\19\关闭连接.exe

完成对数据库的操作后，需要及时断开与数据库的连接并释放内存，否则会浪费大量的内存空间，在访问量较大的 Web 项目中，很可能导致服务器崩溃。在 MySQL 函数库中，使用 mysql\_close() 函数断开与 MySQL 服务器的连接，该函数的语法格式如下：

```
bool mysql_close ([resource link_identifier])
```

mysql\_close() 函数用于关闭指定的连接标识所关联的 MySQL 服务器的非持久连接。如果没有指定 link\_identifier 参数，则关闭最后一个打开的连接。

**例 19.8** 使用 mysql\_connect() 函数建立与数据库的连接，然后使用 mysql\_select\_db() 函数选择数据库并使用 mysql\_close() 函数断开与 MySQL 数据库的连接，在断开与 MySQL 数据库连接后，再次使用 mysql\_select\_db() 函数选择数据库，从两次选择数据库的情况来判断 mysql\_close() 函数是否能起到断开数据库连接的作用。实例代码如下：（实例位置：光盘\TM\Instances\19\8）

```

<?php
$host = "127.0.0.1"; //MySQL 数据库服务器
$userName = "root"; //用户名
$password = "root"; //密码
$dbName = "db_database19"; //数据库名
$connID = mysql_connect($host, $userName, $password); //连接 MySQL 数据库
mysql_select_db($dbName, $connID); //选择数据库
mysql_close($connID); //断开与数据库连接
mysql_select_db($dbName, $connID); //再次选择数据库
?>

```

运行本实例，将在页面中输出如图 19.10 所示的错误信息，从错误信息中可以判断，第一次对数据库选择操作是成功的，而第二次操作是失败的，即可证明 `mysql_close()` 函数成功关闭了与数据库的连接。

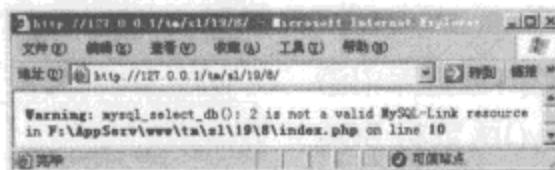


图 19.10 错误提示

### 19.3 PHP 操作 MySQL 数据库

PHP 操作 MySQL 数据库是 PHP 中基础且较重要的技术，开发人员只有熟练地掌握这部分知识才能够独立开发出基于 PHP 的数据库应用。

本节通过一个公告栏进一步讲解 PHP 通过使用 MySQL 操作函数实现对 MySQL 数据库进行增、删、改、查操作的具体实现过程。

#### 19.3.1 应用 insert 命令动态添加公告信息

##### 视频讲解：光盘\TM\Video\19\应用 insert 命令动态添加公告信息.exe

在编写公告栏实例前，首先要建立数据库及数据表结构。创建数据表的方法有两种，分别为在命令提示符下创建和使用 phpMyAdmin 图形化管理工具创建。下面介绍这两种创建数据库的具体步骤和实现过程。

在命令提示符下创建数据表结构。

选择“开始”/“运行”命令，打开“运行”窗口。在“运行”窗口中输入“cmd”命令将弹出命令提示符窗口。在命令提示符下使用如下命令登录 MySQL 数据库服务器：

```
mysql -uroot -proot
```

成功登录 MySQL 数据库服务器后，使用 `use` 命令选择数据库 `db_database19`，如下所示：

```
use db_database19
```

选择数据库 `db_database19` 后，输入如下命令创建数据表 `tb_affiche`：

```
CREATE TABLE 'tb_affiche' (
    'id' INT( 4 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
    'title' VARCHAR( 200 ) CHARACTER SET gb2312 COLLATE gb2312_chinese_ci NOT NULL ,
    'content' TEXT CHARACTER SET gb2312 COLLATE gb2312_chinese_ci NOT NULL ,
    'createtime' DATETIME NOT NULL
) ENGINE = MYISAM ;
```

利用 phpMyAdmin 图形化管理工具创建数据表结构。

在 phpMyAdmin 图形化管理工具中，选择数据库 db\_database19，然后按图 19.11 所示的表结构创建 tb\_affiche 表。

下面按照 PHP 访问 MySQL 数据库的一般步骤讲解使用 PHP 中的 MySQL 操作函数向数据库表添加数据的方法。

**例 19.9 应用 insert 命令动态地向数据库中添加公告信息，然后应用 mysql\_query() 函数将 SQL 语句发送到 MySQL 服务器，从而完成将数据动态添加到数据库的操作。（实例位置：光盘\TM\Instances\19\9）**

程序开发步骤如下：

(1) 在左侧导航区的添加公告信息的图片上添加热区，代码如下。

```


    <area shape="rect" coords="30,45,112,63" href="add_affiche.php">


```

(2) 创建 add\_affiche.php 页面，在该页面中建立公告信息录入表单，并指定表单的提交地址为 check\_add\_affiche.php 页面，代码如下。

```
<form name="form1" method="post" action="check_add_affiche.php">
<table width="520" height="212" border="0" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
    <tr>
        <td width="87" align="center">公告主题:</td>
        <td width="433" height="31">
            <input name="txt_title" type="text" id="txt_title" size="40">
        * </td>
    </tr>
    <tr>
        <td height="124" align="center">公告内容:</td>
        <td><textarea name="txt_content" cols="50" rows="8" id="txt_content"></textarea></td>
    </tr>
    <tr>
        <td height="40" colspan="2" align="center">
            <input name="Submit" type="submit" class="btn_grey" value="保存" onClick="return check(form1)">
            <input type="reset" name="Submit2" value="重置">
        </td>
    </tr>
</table>
```



图 19.11 公告信息数据表 tb\_affiche 的字段说明

```

</tr>
</table>
</form>

```

提交表单前，要对用户录入数据的合法性进行验证，以便提高系统的安全性。公告录入表单中所要录入的数据只需验证是否为空即可。该部分是采用 JavaScript 脚本在客户端进行验证的，这可以有效降低服务器的负载，实现代码如下。

```

<script language="javascript">
function check(form){
    if(form.txt_title.value==""){          //验证表单信息是否为空
        alert("请输入公告标题!");form.txt_title.focus();return false;
    }
    if(form.txt_content.value==""){          //公告内容不能为空，否则弹出提示信息
        alert("请输入公告内容!");form.txt_content.focus();return false;
    }
    form.submit();                         //提交表单信息到数据处理页
}
</script>

```

(3) 提交表单信息到数据处理页 `check_add_affiche.php` 页面，连接 MySQL 数据库服务器，并指定数据库的编码格式为 GB2312 编码。通过 POST 方法获取表单传递过来的值，然后应用 `insert` 命令将表单信息添加到数据表，并由 `mysql_query()` 函数发送到服务器，从而完成公告信息的添加。添加成功则弹出提示信息，并重新定位到首页 `index.php`，代码如下。

```

<?php
$conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
*****这里必须指定数据库的编码方式为 GB2312，否则存储到数据表中的公告信息为乱码*****
*****/
mysql_query("set names gb2312");           //选择编码格式为 GB2312
$title=$_POST[txt_title];                     //获取公告标题信息
$content=$_POST[txt_content];                 //获取公告内容
$createTime=date("Y-m-d H:i:s");             //获取系统的当前时间，“H”表示 24 小时制必须大写
*****应用 mysql_query() 函数执行 insert...into 语句发送到服务器*****
$sql=mysql_query("insert into tb_affiche(title,content,createTime)values('$title','$content','$createTime')");
echo "<script>alert('公告信息添加成功!');window.location.href='add_affiche.php';</script>";
mysql_free_result($sql);                      //关闭记录集
mysql_close($conn);                          //关闭 MySQL 数据库服务器
?>

```

在上面的代码中，`date()` 函数用来获取系统的当前时间，在公告信息添加成功后，应用 JavaScript 弹出提示对话框，并在 JavaScript 脚本中应用 “`window.location.href='index.php'`” 重定位网页，这里将网页重新定位到首页。

(4) 在 IE 浏览器中输入地址，按 Enter 键，单击“添加公告信息”超链接，在页面中添加公告主题和公告内容，单击“保存”按钮，弹出“公告信息添加成功”提示信息，运行结果如图 19.12 所示。单击“确定”按钮，重新定位到首页。

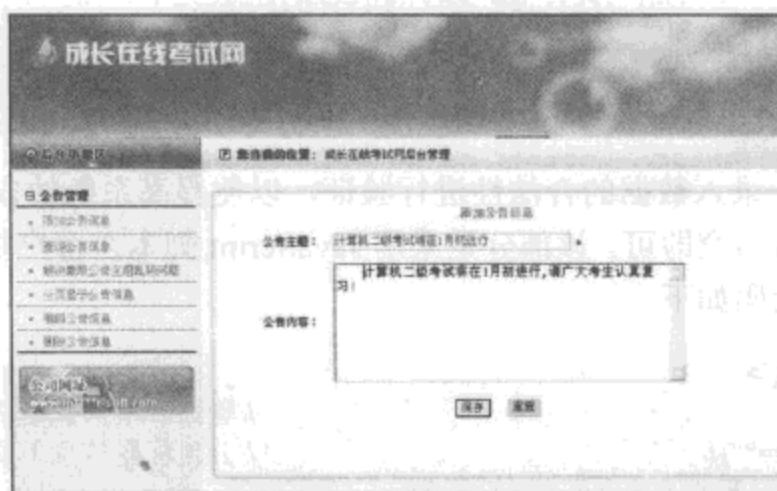


图 19.12 添加公告页面的运行结果

### 19.3.2 应用 select 命令查询公告信息

**视频讲解：**光盘\TM\Video\19\应用 select 命令查询公告信息.exe

添加公告信息后，即可对公告信息执行查询操作。

**例 19.10** 应用 select 命令动态检索数据库中的公告信息，并应用 mysql\_query() 函数将 SQL 语句发送到 MySQL 服务器，从而完成数据的检索操作。（实例位置：光盘\TM\Instances\19\10）

程序开发步骤如下：

(1) 建立导航页面 menu.php，然后应用 include 语句将菜单导航页面 menu.php 嵌入 index.php 页面中。

(2) 建立 search\_affiche.php 页面，在 search\_affiche.php 页的右侧信息显示区域添加一个表单、一个文本框、一个提交按钮，代码如下。

```
<form name="form1" method="post" action="">
    查询关键字 &nbsp;
    <input name="txt_keyword" type="text" id="txt_keyword" size="40">
    <input type="submit" name="Submit" value="搜索" onClick="return check(form)">
</form>
```

为防止用户搜索空信息，本程序在保存按钮的 onclick 事件中，调用一个由 JavaScript 脚本定义的 check() 函数，用来限制文本框信息不能为空。当用户单击“保存”按钮时，自动调用 check() 函数，该函数的代码如下。

```
<script language="javascript">
function check(form){
    if(form.txt_keyword.value==""){
        alert("请输入查询关键字!");form.txt_keyword.focus();return false;
    }
    form.submit();
}
</script>
```

//验证表单信息是否为空  
//查询关键字不能为空，否则弹出提示信息  
//提交表单信息

(3) 提交表单信息到数据处理页 check\_add\_affiche.php，连接 MySQL 数据库服务器，并指定数

数据库的编码格式为 GB2312 编码。通过 POST 方法获取表单传递过来的值，然后应用 insert 命令将表单信息添加到数据表，从而完成公告信息的添加。添加成功则弹出提示信息，并重新定位到首页 index.php，代码如下。

```
<?php
$conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
mysql_query("set names gb2312"); //选择编码格式为 GB2312
$keyword=$_POST[txt_keyword]; //获取查询关键字内容
$sql=mysql_query("select * from tb_affiche where title like '%".$keyword."%' or content like '%".$keyword."%'");
$row=mysql_fetch_object($sql); //获取查询结果集
if(!$row){ //如果未检索到信息资源，则弹出提示信息
echo "<font color='red'>您搜索的信息不存在，请使用类似的关键字进行检索!</font>";
}
do{ //应用 do...while 循环语句输出查询结果
?
<tr bgcolor="#FFFFFF">
<td><?php echo $row->title;?></td>
<td><?php echo $row->content;?></td>
</tr>
<?php
}while($row=mysql_fetch_object($sql)); //do...while 循环语句结束
mysql_free_result($sql); //关闭记录集
mysql_close($conn); //关闭 MySQL 数据库服务器
?>
```

(4) 在 IE 浏览器中输入地址，按 Enter 键，单击“查询公告信息”超链接，将弹出如图 19.13 所示的查询公告信息页面，在页面的文本框中输入查询关键字，然后单击“搜索”按钮即可查询到所有与查询关键字匹配的公告信息。

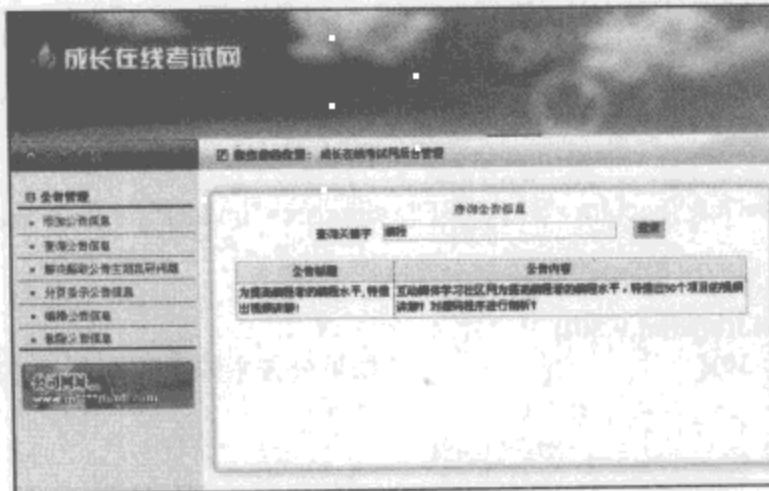


图 19.13 查询公告页面的运行结果

### 19.3.3 解决截取公告主题乱码问题

**视频讲解：**光盘\TM\Video\19\解决截取公告主题乱码问题.exe

在开发 Web 程序时，为了保持整个页面的合理布局，经常需要对一些较长的主题进行截取输出，

但由于汉字占有两个字符，如果截取位置选取不当，将会导致截取的字符串末尾出现乱码。例如，应用 substr() 函数对普通字符进行截取操作会非常方便，但对全角字符进行截取可能会出现乱码。可以通过自定义函数 chinesesubstr() 解决上述问题。解决截取字符串乱码前后的对比如图 19.14 和图 19.15 所示。

```
<?php
echo substr($info[title], 0, 30);
if(strlen($info[title])>30){
    echo "...";
}
?>
```

图 19.14 应用 substr() 函数截取字符串出现乱码

```
<?php
echo chinesesubstr($info[title], 0, 30);
if(strlen($info[title])>30){
    echo "...";
}
?>
```

图 19.15 应用自定义函数 chinesesubstr() 解决截取字符串乱码

### 例 19.11 应用自定义函数 chinesesubstr() 解决截取公告主题出现乱码的问题。(实例位置: 光盘\TM\Instances\19\11)

程序开发步骤如下:

- (1) 在菜单导航页 menu.php 中插入一张图片 image\_09.gif，在图片上添加热区，链接文件为 intercept.php。然后应用 include 语句将菜单导航页 menu.php 嵌入 index.php 页面中。
- (2) 建立 intercept.php 页，在 intercept.php 页的右侧信息显示区域动态输出公告主题信息，并截取公告主题的前 30 个字符，代码如下。

```
<?php
$conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
mysql_query("set names gb2312"); //选择编码格式为 GB2312
$sql=mysql_query("select * from tb_affiche order by createtime desc limit 0,6");
$info=mysql_fetch_array($sql); //获取查询结果集
if($info==false){ //如果未检索到信息资源，则弹出提示信息
    echo "本站暂无公告信息!";
} else{
    do{
        ?>
        <tr bgcolor="#E3E3E3">
            <td height="24" align="left" bgcolor="#FFFFFF">
            <?php
                //调用自定义函数 chinesesubstr(), 屏蔽乱码
                echo chinesesubstr($info[title],0,30); //如果公告主题的长度大于 30 个字符，则输出“...”
                if(strlen($info[title])>30){
                    echo "...";
                }
            ?>
            </td>
        </tr>
    <?php
        }while($info=mysql_fetch_array($sql)); //do...while 循环语句结束
    }
    mysql_free_result($sql); //关闭记录集
    mysql_close($conn); //关闭 MySQL 数据库服务器
?>
```

在上面的代码中，应用了自定义函数 chinesesubstr()，为了管理和维护方便，这里将该函数封装在一个独立的 function.php 页中。因此，调用时需要应用 include 语句进行引用，代码如下。

```
<?php include("function.php"); ?>
```

其中，function.php 页中定义一个用于截取一段字符串的函数 chinesesubstr()，代码如下。

```
<?php
function chinesesubstr($str,$start,$len) { // $str 指字符串，$start 指字符串的起始位置，$len 指字符串长度
    $strlen=$start+$len; // 用 $strlen 存储字符串的总长度，即从字符串的起始位置到末尾的总长度
    for($i=0;$i<$strlen;$i++) {
        if(ord(substr($str,$i,1))>0xa0) { // 如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示汉字
            $tmpstr.=substr($str,$i,2); // 每次取出两位字符赋给变量$tmpstr，即等于一个汉字
            $i++; // 变量自加 1
        } else
            $tmpstr.=substr($str,$i,1); // 如果不是汉字，则每次取出一位字符赋给变量$tmpstr
    }
    return $tmpstr; // 返回字符串
}
?>
```

(3) 在 IE 浏览器中输入地址，按 Enter 键，系统自动按添加公告的时间降序排列，应用 limit 语句取出最新的 6 条公告主题信息，每条公告主题仅截取前 30 个字符进行输出，公告主题大于 30 个字符的字符串部分用省略号“...”代替，运行结果如图 19.16 所示。如果未检索到相匹配的公告信息，则输出提示信息。

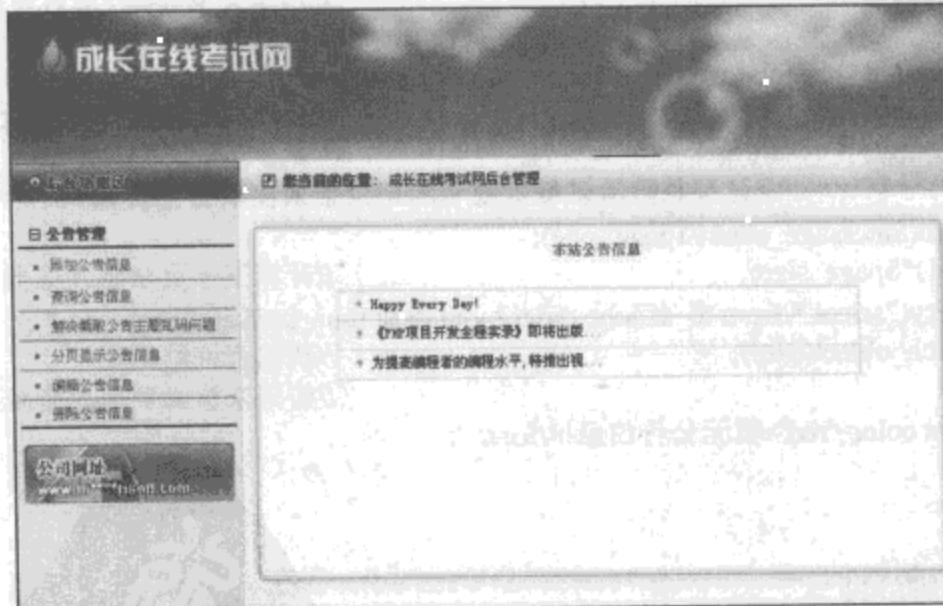


图 19.16 解决截取公告主题乱码问题

#### 19.3.4 分页显示公告信息

视频讲解：光盘\TM\Video\19\分页显示公告信息.exe

在实现了添加公告信息后，便可以对公告信息执行查询操作。下面在实例 19.9 的基础上实现查询

公告信息。

**例 19.12 应用 select 命令动态检索数据库中的公告信息，并应用 mysql\_query() 函数将 SQL 语句发送到 MySQL 服务器，从而完成数据的检索操作。（实例位置：光盘\TM\Instances\19\12）**

程序开发步骤如下：

(1) 将菜单导航部分封装成一个独立的页 menu.php，在该页中插入一张图片 image\_09.gif，在图片上添加热区，链接 page\_affiche.php 文件。

(2) 在 page\_affiche.php 页的右侧信息显示区域，连接 MySQL 数据库服务器，连接数据源，并指定数据库的编码格式为 GB2312 编码。应用 select 命令将检索公告信息表中的数据，并由 mysql\_query() 函数发送到服务器，从而完成公告信息的分页显示，代码如下。

```
<table width="550" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#999999">
<tr align="center" bgcolor="#f0f0f0">
    <td width="221">公告标题</td>
    <td width="329">公告内容</td>
</tr>
<?php
    $conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
    mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
    /******这里必须指定数据库的编码方式为 GB2312，否则输出到浏览器中的公告信息为乱码******/
    *****/
    mysql_query("set names gb2312");
    /******$page 为当前页，如果$page 为空，则初始化为 1******/
    if($page==""){
        $page=1;
        if(is_numeric($page)){
            //判断变量$page 是否为数字，如果是则返回 true
            $page_size=4;
            //每页显示 4 条记录
            $query="select count(*) as total from tb_affiche order by id desc";
            $result=mysql_query($query);
            //查询符合条件的记录总条数
            $message_count=mysql_result($result,0,'total');
            //要显示的总记录数
            /******由记录总数除以每页显示的记录数求出所分的页数******/
            $page_count=ceil($message_count/$page_size);
            $offset=($page-1)*$page_size;
            //计算下一页从第几条数据开始循环
            $sql=mysql_query("select * from tb_affiche order by id desc limit $offset, $page_size");
            $row=mysql_fetch_object($sql);
            if(!$row){
                //如果未检索到信息资源，则输出提示信息
                echo "<font color='red'>暂无公告信息!</font>";
            }
        }
        do{
        ?>
            <tr bgcolor="#FFFFFF">
                <td><?php echo $row->title;?></td>
                <td><?php echo $row->content;?></td>
            </tr>
        <?php
            }while($row=mysql_fetch_object($sql));
        ?
    }
</table>
```

**◆ 注意：** do…while()循环和 while()循环的区别。do…while()循环是先执行{}中的代码段，然后判断 while 中的条件表达式是否成立，如果返回 true，则重复输出{}中的内容，否则结束循环，执行 while 下面的语句；while 循环是先判断 while 中的表达式，当返回 true 时，再执行{}中的代码。两者的主要区别在于，do…while()循环比 while()循环多输出一次结果。

(3) 添加一个一行一列的表格，添加如下代码，实现翻页功能。

```
<table width="550" border="0" cellspacing="0" cellpadding="0">
<tr>
<!-- 翻页条 -->
<td width="37%">&ampnbsp&ampnbsp页次: <?php echo $page;?>/<?php echo $page_count;?>页&ampnbsp记录: <?php
echo $message_count;?> 条&ampnbsp</td>
<td width="63%" align="right">
<?php
/* 如果当前页不是首页 */
if($page!=1){
/* 显示“首页”超链接 */
echo "<a href=page_affiche.php?page=1>首页</a>&ampnbsp";
/* 显示“上一页”超链接 */
echo "<a href=page_affiche.php?page=".($page-1).">上一页</a>&ampnbsp";
}
/* 如果当前页不是尾页 */
if($page<$page_count){
/* 显示“下一页”超链接 */
echo "<a href=page_affiche.php?page=".($page+1).">下一页</a>&ampnbsp";
/* 显示“尾页”超链接 */
echo "<a href=page_affiche.php?page=$page_count.>尾页</a>";
}
mysql_free_result($sql); //关闭记录集
mysql_close($conn); //关闭 MySQL 数据库服务器
?>
</td>
</tr>
</table>
```

(4) 在 IE 浏览器中输入地址，按 Enter 键，分页显示公告信息，运行结果如图 19.17 所示。

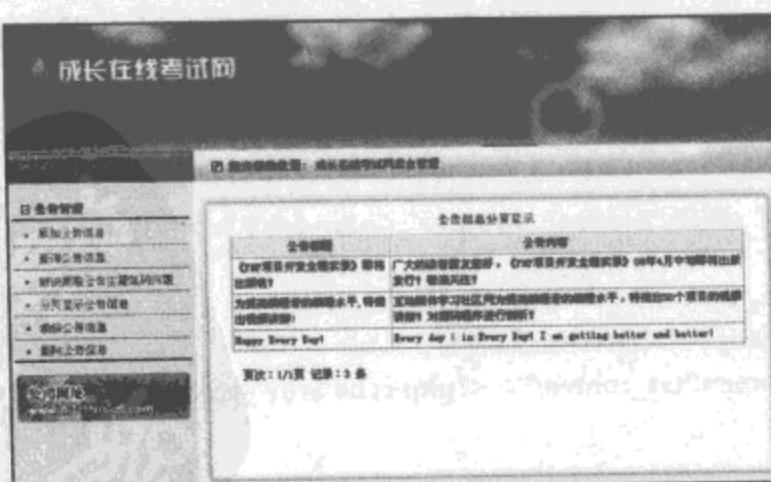


图 19.17 分页显示公告信息的运行结果

### 19.3.5 应用 update 命令动态编辑公告信息

**视频讲解：**光盘\TM\Video\19\应用 update 语句动态编辑公告信息.exe

公告信息也不是一成不变的，可以对公告主题及公告内容的动态信息进行编辑。

**例 19.13** 应用 update 命令动态编辑数据库中的公告信息，然后通过 mysql\_query() 函数将 SQL 语句发送到 MySQL 服务器，从而完成数据的编辑操作。（实例位置：光盘\TM\Instances\19\13）

程序开发步骤如下：

(1) 在菜单导航页 menu.php 的 image\_09.gif 图片上添加热区，链接 update\_affiche.php 文件。

**注意：**为了提高效率，将首页 index.php 另存为 update\_affiche.php。

(2) 在 update\_affiche.php 页的右侧信息显示区域应用 select 命令检索出全部的公告信息，添加链接文件 modify.php，并向 modify.php 页传递公告 ID 的参数值，代码如下。

```
<a href="modify.php?id=<?php echo $row->id;?>">
    
</a>
```

(3) 按照首页的风格创建一个编辑公告信息的 modify.php 页，在该页面中添加一个表单、一个文本框、一个编辑框、一个隐藏域、一个提交按钮和一个重置按钮，设置表单的 action 属性值为 check\_modify\_ok.php，代码如下。

```
<?php
$conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
mysql_query("set names gb2312");
$id=$_GET[id];                                //应用 GET 方法接收欲编辑的公告 ID
$sql=mysql_query("select * from tb_affiche where id=$id"); //检索公告 ID 所对应的公告信息
$row=mysql_fetch_object($sql);                  //获取结果集
?>
<form name="form1" method="post" action="check_modify_ok.php">
    <table width="520" height="212" border="0" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
        <tr>
            <td width="87" align="center">公告主题：</td>
            <td width="433" height="31">
                <input name="txt_title" type="text" id="txt_title" size="40" value="<?php echo $row->title;?>">
                <!-- 将公告 ID 的值赋给隐藏域 -->
                <input name="id" type="hidden" value="<?php echo $row->id;?>"></td>
            </tr>
        <tr>
            <td height="124" align="center">公告内容：</td>
            <td><textarea name="txt_content"><?php echo $row->content;?></textarea></td>
        </tr>
        <tr>
            <td height="40" colspan="2" align="center">
                <input name="Submit" type="submit" class="btn_grey" value="修改" onClick="return check(form1);">
            </td>
        </tr>
    </table>
</form>
```

```

<input type="reset" name="Submit2" value="重置"></td>
</tr>
</table>
</form>

```

(4) 提交表单信息到数据处理页 check\_modify\_ok.php。首先，连接 MySQL 数据库服务器，然后选择数据库，设置 GB2312 的编码格式。应用 POST 方法获取表单信息，接下来应用 mysql\_query() 函数向 MySQL 数据库服务器发送修改公告信息的 SQL 语句，最后应用 if…else 条件语句对修改后的信息进行判断，并弹出相应的提示信息，重新定位到公告信息编辑页，代码如下。

```

<?php
*****连接数据源，读者可将此处封装成独立的页，然后应用 include 语句调用，效率会很高*****
$conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
mysql_query("set names gb2312");
***** ****
$title=$_POST[txt_title];                                //获取更改的公告主题
$content=$_POST[txt_content];                            //获取更改的公告内容
$id=$_POST[id];                                         //获取更改的公告 ID
//应用 mysql_query()函数向 MySQL 数据库服务器发送修改公告信息的 SQL 语句
$sql=mysql_query("update tb_affiche set title='".$title',content='".$content' where id=$id");
if($sql){
    echo "<script>alert('公告信息编辑成功！');history.back();window.location.href='modify.php?id=$id';</script>";
} else{
    echo "<script>alert('公告信息编辑失败！');history.back();window.location.href='modify.php?id=$id';</script>";
}
?>

```

(5) 在 IE 浏览器中输入地址，按 Enter 键，在页面中输入查询关键字，单击“搜索”按钮，即可输出检索到的公告信息资源，单击超链接，弹出编辑公告详细信息页，对指定的公告信息进行编辑，单击“修改”按钮，即可完成指定公告信息的编辑操作，运行结果如图 19.18 所示。

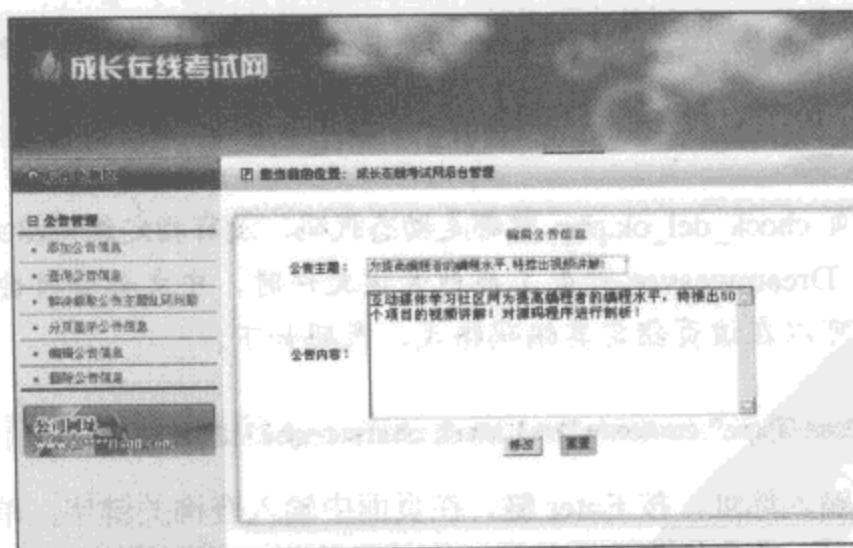


图 19.18 编辑公告页面的运行结果

### 19.3.6 应用 delete 命令动态删除公告信息

视频讲解：光盘\TM\Video\19\应用 delete 语句动态删除公告信息.exe

公告信息是对一些重要的、新鲜的事务进行管理，因此，为了节省系统资源，可以定期对公告主题及公告内容的信息进行删除。

**例 19.14** 应用 delete 命令动态编辑数据库中的公告信息，然后通过 mysql\_query() 函数将 SQL 语句发送到 MySQL 服务器，从而完成数据的删除操作。（实例位置：光盘\TM\Instances\19\14）

程序开发步骤如下：

- (1) 在菜单导航页 menu.php 的 image\_09.gif 图片上添加热区，链接 delete\_affiche.php 文件。
- (2) 在 delete\_affiche.php 页的右侧信息显示区域应用 select 命令检索出全部的公告信息。添加链接文件 check\_del\_ok.php，并向 check\_del\_ok.php 页传递公告 ID 的参数值，代码如下。

```
<a href="check_del_ok.php?id=<?php echo $row->id;?>">

</a>
```

(3) 提交公告信息的 ID 到数据处理页 check\_del\_ok.php。首先，连接 MySQL 数据库服务器，然后选择数据库，设置 GB2312 的编码格式。应用 GET 方法获取欲删除的公告 ID，接下来应用 mysql\_query() 函数向 MySQL 数据库服务器发送删除公告信息的 SQL 语句，最后应用 if…else 条件语句对修改后的信息进行判断，并弹出相应的提示信息，重新定位到公告信息编辑页，代码如下。

```
<?php
$conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误".mysql_error());
mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
mysql_query("set names gb2312");
$id=$_GET[id];
$sql=mysql_query("delete from tb_affiche where id=$id");
if($sql){
    echo "<script>alert('公告信息删除成功！');history.back();window.location.href='delete_affiche.php?id=$id';</script>";
} else{
    echo "<script>alert('公告信息删除失败！');history.back();window.location.href='delete_affiche.php?id=$id';</script>";
}
?>
```

技巧：由于数据处理页 check\_del\_ok.php 中都是动态代码，没有指定编码格式为 GB2312 编码类型，以致读者在用 Dreamweaver 开发工具打开该文件时，中文部分将会显示乱码。为了解决这一问题，读者可以在该页指定其编码格式，代码如下。

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```

(4) 在 IE 浏览器中输入地址，按 Enter 键，在页面中输入查询关键字，单击“搜索”按钮，即可输出检索到的公告信息资源。单击指定公告信息后面的按钮，弹出删除公告信息提示，然后单击“确定”按钮，即可完成对指定公告信息的删除操作，运行结果如图 19.19 所示。

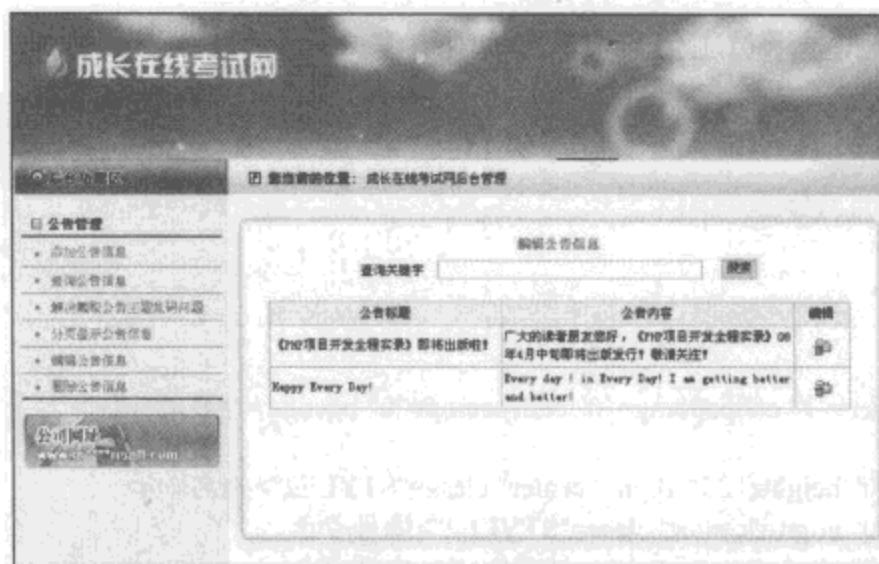


图 19.19 删除公告页面信息的运行结果

## 19.4 实战

MySQL 数据库函数的使用方法是每位 PHP 开发人员所必须掌握的，只有掌握了这些函数的使用方法才能具备最基本数据库系统开发的能力。为了巩固本章所讲的内容，下面通过具体实例进一步讲解和深化 MySQL 数据库函数的应用及 PHP 操纵 MySQL 数据库进行增、删、改、查的操作方式。

### 19.4.1 输入页码跳转到指定页

#### 视频讲解：光盘\TM\Video\19\输入页码跳转到指定页.exe

在网站的应用过程中，经常需要将数据库中很多记录显示到页面中来，如果将所有的记录在一个页面中显示，会给浏览者带来很多麻烦，最好的解决方法就是使用分页技术，将记录进行分页显示。通过分页读取记录信息，在每页中显示几条记录，这样既方便浏览者的浏览，也节省了页面的空间。

**例 19.15 以分页的方式显示员工信息，并通过输入页码跳转到指定的页。（实例位置：光盘\TM\Instances\19\15）**

程序开发步骤如下：

(1) 首先建立与 MySQL 数据库的连接，并选择数据库 db\_database19，如果发生错误则使用 mysql\_error() 函数输出错误号，最后使用 mysql\_query() 函数执行 set names gb2312 命令来设置数据库的字符集为简体中文，代码如下。

```
<?php
$Id=mysql_connect("localhost","root","root")or die('连接失败：'.mysql_error());
if(mysql_select_db("db_database19",$Id))
    echo "";
else
    echo ('连接失败：'.mysql_error());
mysql_query("set names gb2312");
?>
```

(2) 使用分页的方式显示员工信息，代码如下。

```

<?php
include("conn/conn.php");
if ($page=="") {$page=1;} if ($ljjl=="") {
    $ljjl=0;
};
?>
<body>
<table width="450" border="1" cellpadding="0" cellspacing="0" bgcolor="#4DB1FD">
    <tr>
        <td width="100" height="25" align="center" class="STYLE2">姓名</td>
        <td width="100" align="center" class="STYLE2">编号</td>
        <td width="125" align="center" class="STYLE2">电话</td>
        <td width="175" align="center" class="STYLE2">地址</td>
    </tr>
<?php
if($page){
    $page_size=2;                                //每页显示两条记录
    $query="select count(*) as total from tb_insert where id"; //从数据库中读取数据
    $result=mysql_query($query);
    $message_count=mysql_result($result,0,"total");           //获取总的记录数
    $page_count=ceil($message_count/$page_size);             //获取总的页数
    $offset=($page-1)*$page_size;
    $query="select * from tb_insert where id order by id desc limit $offset, $page_size";
    $result=mysql_query($query);
    while ($myrow=@mysql_fetch_array($result)){
?
<tr>
        <td height="25" align="center"><span class="STYLE2"><?php echo $myrow[name];?></span></td>
        <td align="center"><span class="STYLE2"><?php echo $myrow[number];?></span></td>
        <td align="center"><span class="STYLE2"><?php echo $myrow[tel];?></span></td>
        <td align="center"><span class="STYLE2"><?php echo $myrow[address];?></span></td>
    </tr>
<?php
}
?>
</table>
<table width="450" border="1" cellpadding="0" cellspacing="0" bgcolor="#4DB1FD">
    <tr>
        <td width="45%" align="center"><span class="STYLE1">&ampnbsp&ampnbsp页次:</span>
            <?php echo $page;?> / <?php echo $page_count;?> 页 记录:
            <?php echo $message_count;?> 条&ampnbsp </span>
        </td>
        <td width="55%" height="22" align="center"><span class="STYLE1"> 分页:
<?php
    if($page!=1){
        echo "<a href=index.php?page=1>首页 &ampnbsp";
    }
    if($page<$page_count){
        echo "<a href=index.php?page=$page+1> </a> ";
    }
    if($page>1){
        echo "<a href=index.php?page=$page-1> </a> ";
    }
    if($page==$page_count){
        echo "<a href=index.php?page=$page> </a> ";
    }
}
?>
</td>

```

```

echo "<a href=index.php?page=".($page-1).">上一页</a>&ampnbsp";
}
if($page<$page_count){
    echo "<a href=index.php?page=".($page+1).">下一页</a>&ampnbsp";
    echo "<a href=index.php?page=$page_count.>尾页</a>";
}
?>
</span></td>
</tr>
</table>

```

运行上述实例，如图 19.20 所示，在图中的文本框中输入要浏览的页码，然后单击“跳转”按钮即可跳转到要浏览的页面。

姓名	编号	电话	地址
李**	1002	1231233	沈阳
王**	1001	12345645	吉林

次：2 / 3 页 记录：6 条 分页：首页 上一页 下一页 尾页  跳转

图 19.20 跳转到指定的页面

#### 19.4.2 图片的分栏分页显示

##### ■ 视频讲解：光盘\TM\Video\19\图片的分栏分页显示.exe

PHP 实现 MySQL 数据库中的数据分页显示相对简单，在此基础上还可以实现如图片、代码块的分栏分页显示。

**例 19.16 图片的分栏分页显示。（实例位置：光盘\TM\Instances\19\16）**

程序开发步骤如下：

(1) 创建数据库连接，代码如下。

```

<?php
$Id=mysql_connect("localhost","root","root")or die('连接失败:'.mysql_error());
if(mysql_select_db("db_database19",$Id))
echo "";
else
echo ('连接失败:'.mysql_error());
mysql_query("set names gb2312");
?>

```

(2) 通过分页、分类、分栏方法将数据库中的数据显示到页面中，代码如下。

```

<table width="400" border="0" cellpadding="0" cellspacing="1" bgcolor="#663300">
<?php
if($page){
    $page_size=8;
    $query="select count(*) as total from tb_picture where sort='1'";
    $result=mysql_query($query);
?>

```

```

$messag_count=mysql_result($result,0,"total");
$page_count=ceil($messag_count/$page_size);
$offset=($page-1)*$page_size;
$query="select * from tb_picture where sort='1' limit $offset, $page_size";
$result=mysql_query($query);
$row=mysql_fetch_array($result);
if($row[sort]==1){
    for($i=1; $i<=2; $i++) {
        echo "<tr>";
        if($i==1) {
            $query="select * from tb_picture where sort='1' limit $offset, $page_size";
            $result=mysql_query($query); }
        $j=1;
        while($myrow=mysql_fetch_array($result)){
            $id=$myrow[id];
            if ($j<=4){
                ?>
                <td height="50" align="center" valign="top" bgcolor="#FFFFFF" class="STYLE1">
<table width="50" border="0" cellspacing="0">
<tr align="center" valign="middle">
<td width="50" height="50" align="center" valign="middle" background="images/beijing.jpg"><?php
echo $myrow[id];?></td>
</tr>
</table></td>
<?php
}
++$j;
if($j==5){
    break;
}
}
echo "</tr>";
}
}
?>
<?php
for($i=1; $i<=2; $i++) {
    echo "<tr>";
}
<?php
if($i==1){
    $query="select * from tb_picture where sort='2' limit $offset, $page_size";
    $result=mysql_query($query); }
    $j=1;
    while($myrow=mysql_fetch_array($result)){
        $id=$myrow[id];
        if($j<=4){
            ?>
            <td height="50" align="center" valign="top" bgcolor="#FFFFFF" class="STYLE1"> <table width="50"
border="0" cellspacing="0">

```

```

<tr align="center" valign="middle">
    <td width="50" height="50" align="center" valign="middle" background="images/beijing.jpg"><?php
echo $myrow[id];?></td>
</tr>
</table>
</td>
<?php
}
++$j;
if ($j==5){ break;
}
}
echo "</tr>";
}
?>
</table>
<table width="400" border="0" cellspacing="0" cellpadding="0">
<tr>
    <td width="50%" height="22" class="hongse01 STYLE1">&ampnbsp&ampnbsp页次: <font class="huise01"><?php
echo $page;?></font>
    / <font class="huise01"><?php echo $page_count;?> </font>页 记录: <font class="huise01">
<?php echo $message_count;?>
    </font>条&ampnbsp</td>
    <td width="39%" height="22" class="hongse01 STYLE1"> 分页:
<?php
    $xsoudh="id=$id";
    $next=$lijl*10;
    $n=$lijl-1;
    $m=$lijl+1;
    $prev_page=$page-10;
    if ($lijl==0){
        echo "<img src=\"images/02.jpg\" width=\"8\" height=\"9\" title=\"首页\">";
    }else{
        echo "<a href='$_PATH_INFO?page=1'><img src=\"images/02.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"首页\"></a>&ampnbsp";
        $ccc=$vv-10;
        echo "<a href='$_PATH_INFO?page=$prev_page'><img src=\"images/01.jpg\" width=\"8\" height=\"9\" title=\"上一页\"></a>";
    }
?>
<?php
for($j=1;$j<=$page_count;$j++){
    $pnex=$next+$j;
    if($mm==10){ break; }
    if($mm>$page_count){ break; }
    if($page_count-$vv<10){
        if ($mm>=$page_count-$vv){ break; }
    }
?>
<?php

```

```

echo "<a href='PATH_INFO?page=$pnext'> $pnext </a>";
$mm=$mm+1;
}
?>
</td>
<td width="11%"> <span class="STYLE1">
<?php
$vv=$vv+$mm;
if($page_count-$vv<=0){
echo "<img src='images/03.jpg' width='8' height='9' title='尾页'>";
}else{
echo "<a href='PATH_INFO?page=$pnext'><img src='images/03.jpg' width='8' height='9' title='下一页'></a>";
}
if ($message_count==0){ echo "没有记录!"; }
?
?>
</span></td></tr>
</table>

```

(3) 读取数据库中二进制文件调用的文件 image\_1.php, 代码如下。

```

<?php
$id=mysql_connect('localhost','root','root');
mysql_select_db('db_database19',$id);
$query="select * from tb_picture where id=".$recid;
$result=mysql_query($query);
if(!$result) die("error: mysql query");
$num=mysql_num_rows($result);
if($num<1) die("error: no this recorder");
$data = mysql_result($result,0,"xq");
mysql_close($id);
echo $data;
?>

```

运行本实例, 如图 19.21 所示, 通过页面中的分页链接即可以分栏分页的方式浏览所有图片。

1	2	3	4
5	6	7	8
13	14	15	16
17	18	19	20

页次：1 / 2 页 记录：12 条 分页： 1 2

图 19.21 图片的分栏分页显示

### 19.4.3 留言板

视频讲解：光盘\TM\Video\19\留言板.exe

在 PHP 数据库程序开发中，留言板程序是最基本的，但其开发思路和流程却非常有意义，如果初学者能够掌握留言板的开发流程，就可以掌握其他大部分应用，因为万变不离其宗，程序开发的思想都是一致的。

**例 19.17 制作留言板，要求具有发表留言、浏览留言、更改留言和删除留言的功能。（实例位置：光盘\TM\Instances\19\17）**

程序开发步骤如下：

(1) 建立 conn.php 文件，该文件中的代码用于实现与 MySQL 数据库的连接，在制作留言板的其他模块时，只需使用包含语句包含该文件即可，这样可以提高代码的重用率，也可以为以后的维护工作带来极大的方便，代码如下。

```
<?php
$conn=mysql_connect("localhost","root","root"); //建立与 MySQL 数据库的连接
mysql_select_db("db_database19",$conn); //选择数据库
mysql_query("set names gb2312"); //设置字符集
?>
```

(2) 制作发表留言模块，这里主要讲解如何使用 PHP 的数据库操作函数向 MySQL 数据库中添加一条新记录。本模块中，首先建立发表留言表单，表单及表单元素详细说明如表 19.7 所示。

表 19.7 发表留言表单及表单元素

元素类型	元素名称	属性设置	说明
表单	form1	name="form1" method="post" action="saveleaveword.php" onsubmit="return chkinput(this)"	表单
文本域	title	type="text" name="title" size="40" class="inputcss"	留言主题
多行文本域	content	name="content" rows="15" cols="55" class="inputcss"	留言内容
提交按钮	submit	type="submit" name="submit" value="发表" class="buttoncss"	提交按钮
重置按钮	reset	type="reset" name="reset" value="重写" class="buttoncss"	重置按钮

完成留言表单制作后，需要制作保存用户留言信息模块，该模块中使用 mysql\_query() 函数向数据库中插入一条新留言记录，代码如下。

```
<?php
session_start();
include_once("conn.php"); //包含数据库连接模块
$sql=mysql_query("select id from tb_user where usernc='". $_SESSION["unc"]. "'",$conn); //查询留言者 ID 号
$info=mysql_fetch_array($sql);
$userid=$info[id];
$createTime=date("Y-m-d H:i:s"); //获取创建时间
if(mysql_query("insert into tb_leaveword(userid,createTime,title,content)
values('$userid','$createTime','$title','$content')",$conn)){ //发表留言
echo "<script>alert('留言发表成功！');history.back();</script>";
}else{
```

```

echo "<script>alert('留言发表失败！');history.back();</script>";
}
?>

```

(3) 制作浏览留言模块，本实例采用分页的方式显示用户的留言信息。通过本模块的学习，不仅可以使读者进一步了解如何使用 `mysql_query()` 函数执行 MySQL 数据库中信息的检索，更重要的是能够掌握 PHP 操作 MySQL 数据库实现分页的方式。分页模块一般包含两部分，分别为数据显示部分和分页链接部分。

数据显示部分，用户根据页码参数（一般指用 GET 方法传递的要显示的页码值）显示当前页所要显示的数据，本实例中实现该过程的代码如下。

```

<?php
$sql=mysql_query("select count(*) as total from tb_leaveword ",$conn);           //查询总记录数
$info=mysql_fetch_array($sql);
$total=$info[total];
if($total==0){                                         //如果总记录为 0，则给出无留言提示
    echo "<div align=center>对不起，暂无留言！</div>";
} else{
    if(!isset($_GET["page"]) || !is_numeric($_GET["page"])){
        $page=1;
    } else{
        $page=intval($_GET["page"]);
    }
    $pagesize=3;                                         //指定当前要显示的页数
    if($total%$pagesize==0){
        $pagecount=intval($total/$pagesize);             //计算总页数
    } else{
        $pagecount=ceil($total/$pagesize);
    }
    $sql=mysql_query("select * from tb_leaveword order by createtime desc limit ".($page-1)*$pagesize.",$pagesize", $conn);          //通过循环显示当前页的所有记录
    while($info=mysql_fetch_array($sql)){
        //通过循环显示当前页的所有记录
    }
    <!-- 省略代码，请详见本书附带光盘 -->
    <?php
    }
}
?>

```

分页链接部分，主要通过 GET 方法向留言显示页提供页码参数，从而根据该参数判断当前页码要显示的记录，本实例实现该功能的代码如下。

```


|                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div align="left">     共有留言&ampnbsp<?php echo \$total;?>&ampnbsp条&ampnbsp     每页显示&ampnbsp<?php echo \$pagesize;?>&ampnbsp条&ampnbsp     第&ampnbsp<?php echo \$page;?>&ampnbsp页/共&ampnbsp<?php echo \$pagecount;?>&ampnbsp页     </div> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


```

```

</td>
<td width="199"><div align="right">
<a href=<?php echo $_SERVER["PHP_SELF"]?>?page=1" class="a1">首页</a>&nbsp;
<a href=<?php echo $_SERVER["PHP_SELF"]?>?page=<?php
    if($page>1)
        echo $page-1;
    else
        echo 1;
?>" class="a1">上一页</a>&nbsp;
<a href=<?php echo $_SERVER["PHP_SELF"]?>?page=<?php
    if($page<$pagecount)
        echo $page+1;
    else
        echo $pagecount;
?>" class="a1">下一页&nbsp;
<a href=<?php echo $_SERVER["PHP_SELF"]?>?page=<?php echo $pagecount;?>" class="a1">尾页
</a>
</div>
</td>
</tr>
</table>

```

(4) 制作更改留言信息模块，该模块主要介绍如何使用 MySQL 数据库操作函数更新 MySQL 数据库中的记录。首先建立留言信息更改表单，该表单结构与发表留言表单结构类似，这里不再赘述。然后编写用于更改用户留言信息的代码，如下所示。

```

<?php
include_once("conn.php");
$id=$_GET["id"];
if($_POST["submit"]!=""){
    if(mysql_query("update tb_leaveword set title='". $_POST["title"]."',content='". $_POST["content"]."' where id='". $_POST["id"]."'",,$conn)){
        echo "<script>alert('留言更改成功！');window.opener.location.reload();window.close();</script>";
    }else{
        echo "<script>alert('留言更改失败！');history.back();</script>";
    }
    exit; //不需继续执行后面的代码
}
?>

```

(5) 制作删除留言信息模块，该模块首先使用\$\_GET 全局数组接收传递过来的留言 ID 号，然后使用 mysql\_query() 函数执行 SQL 语言的 delete 命令根据留言 ID 号删除指定的留言，代码如下。

```

<?php
include_once("conn.php"); //包含数据连接模块
if(mysql_query("delete from tb_leaveword where id='". $_GET["id"]."'",,$conn)){ //执行删除操作
    echo "<script>alert('留言删除成功！');history.back();</script>";
}else{
    echo "<script>alert('留言删除失败！');history.back();</script>";
}
?>

```

运行本实例，结果如图 19.22 所示，该页面为留言板的主界面，通过页面上部的导航条即可定位到该留言板的所有功能模块。



图 19.22 留言板主界面

## 19.5 本章小结

本章首先介绍了 PHP 访问 MySQL 数据库的一般流程，然后详细介绍了该流程每一步骤的具体实现方法，并重点讲解了常用函数的使用方法，最后通过留言板实例更深层次地介绍了 PHP 如何实现对 MySQL 数据库进行增、删、改、查操作。通过本章的学习，读者能够掌握 PHP 操作 MySQL 数据库的一般流程，掌握常用 MySQL 函数的使用方法，并能够具备独立完成基本数据库程序的能力。希望本章除能够起到抛砖引玉的作用，能够帮助读者在此基础上更深层次地学习 PHP 操作 MySQL 数据库的相关技术，并进一步学习使用面向对象的方式操作 MySQL 数据库的方法。

## 19.6 实战练习

1. 查询结果分页显示，要求运行效果与图 19.23 类似。（答案位置：光盘\TM\Instances\19\18）
2. 按照员工编号查询员工的详细信息，要求运行效果与图 19.24 类似。（答案位置：光盘\TM\Instances\19\19）



图 19.23 查询结果分页显示

查询企业员工的详细信息			
根据员工工号查询			
编号	姓名	电话	地址
1001	王**	12345678	吉林

图 19.24 按编号查询员工信息

# 第 20 章

## ADODB 类库

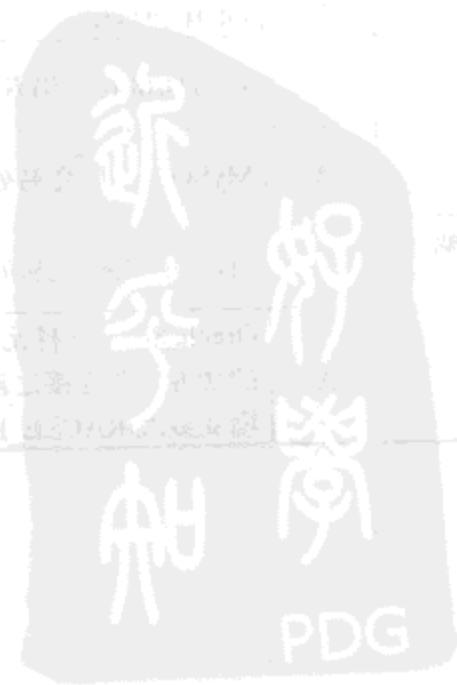
( 视频讲解: 96 分钟)

PHP 几乎为各种类型的数据库都提供了庞大的函数库支持，而且支持得很完整，但遗憾的是，在 PHP 提供的所有函数库中无论在名称或参数结构上，都有很大的差异，这使得 PHP 的系统开发在面临更换数据库时，变得非常繁琐。而 ADODB 类库的诞生，就解决了这个难题。

ADODB 提提供了一套完整的方法和属性让开发者控制数据库系统。在操作过程中无须计较使用的是什么数据库，而只要记得它的功能即可。因为在更换数据库时，只要修改一个属性值，ADODB 就会自动依据设定取用正确的 PHP 函数。此外，有可能还需要修改 SQL 语句，因为不同的数据库在 SQL 语句的使用上有一点区别。这就是 ADODB 的魅力所在。本章将对 ADODB 类库进行详细的介绍。

通过阅读本章，可以学习到：

- ADODB 类库是什么
- ADODB 类库的安装
- ADODB 类库的常用方法
- ADODB 类库的应用
- 如何灵活运行 ADODB 类库



## 20.1 ADODB 类库是什么

ADODB，全称是 Active Data Objects Data Base，它是存取数据库所使用到的一组函数。虽然 PHP 是构建 Web 系统强有力的工具，但是由于 PHP 存取函数一直没有标准化，不同数据库间的函数名称、参数差异很大，在更换数据库时，会带来大量的代码修复工作。这时，就需要一组函数库来隐藏不同数据库函数间的差异，让开发者可以很简单地去切换数据库，这就是 ADODB 类库。

ADODB 类库的优点：

- 安装简单，易学易用。提供了与微软的 ADODB 相似的语法功能。无论是初学者，还是从其他语言转过来的开发人员（主要是 ASP），都可以很快上手。
- 以标准的 SQL 语句书写的程序在更换数据库时，不需要改变源程序。
- 可以生成 Smarty 循环需要的二维数组，简化了 Smarty 开发。
- 支持缓存查询，最大可能地提高查询速度。

ADODB 类库的缺点是执行效率慢。由于 ADODB 的类库非常庞大，仅是主执行类 (adodb.inc.php) 就有 120KB，所以在使用 ADODB 类库时要充分考虑到这一点。

## 20.2 ADODB 支持的数据库

ADODB 目前支持 MySQL、Oracle、Microsoft SQL Server、Sybase、PostgreSQL、FoxPro、Access、ADO 和 ODBC 等绝大多数的数据库。ADODB 所支持的常见数据库及参数如表 20.1 所示。

表 20.1 ADODB 支持的数据库

数据库名称	测试状态	资料库	RecordCount()支持	需要安装的驱动	操作系统
access	B	Microsoft Access/Jet，需要建立一个 ODBC/DSN	Y/N	ODBC	Windows
db2	C	DB2，可以通过 ODBC 获得可以信赖的操作效果	Y/N	DB2 CLI/ODBC interface	UNIX Windows
vfp	A	Microsoft Visual FoxPro，需要建立一个 ODBC/DSN	Y/N	ODBC	Windows
mssql	A	在日期格式上仍有一些问题	Y/N	Mssql client	UNIX Windows
mysql	A	MySQL 支持交易处理	Y/N	MySQL client	UNIX Windows
mysqtl 或 maxsql	A	MySQL 支持交易处理	Y/N	MySQL client	UNIX Windows
oci8	A	Oracle 8/9，支持比 Oracle 驱动程式还多的功能。在连接之前，可能需要先配好环境变数('ORACLE_HOME=...')	Y/N	Oracle client	UNIX Windows

续表

数据库名称	测试状态	资料库	RecordCount()支持	需要安装的驱动	操作系统
oci8po	A	Oracle 8/9 可携式驱动程式	Y/N	Oracle client	UNIX Windows
odbc	A	标准 ODBC 用 PConnect('DSN','user','pwd').连接	? depends on database	ODBC	UNIX Windows
odbc_mssql	C	用 ODBC 连接 MSSQL	Y/N	ODBC	UNIX Windows
odbc_oracle	C	用 ODBC 连接 ORACLE	Y/N	ODBC	UNIX Windows
postgres	C	PostgreSQL 不支援 LIMIT 指令	Y	PostgreSQL client	UNIX Windows
sybase	C	Sybase	Y/N	Sybase client	UNIX Windows

表 20.1 说明如下：

- ☒ A=已经经过很多验证及测试，可靠度最高。
- ☒ B=已经测试并使用了，但可能仍有一些功能没有达成。
- ☒ C=使用者自行配置或试用的驱动程式，可能没有完全支持 ADODB 的功能。
- ☒ “RecordCount()支持”，指的是 RecordCount()函数是否会回传用 SELECT 指令取得的记录数（不支持传回-1）。Y/N 表示当全局变量\$ADODB\_COUNTER=true 时，会以模拟的方式取得，但如果估计到记录数会很大时，最好把这个值设为 false，也就是关掉这个模拟功能，因为这会耗掉很多内存。这个变量在每次执行时都会检查，所以可以选择性地使用。

## 20.3 ADODB 下载与安装

要使用 ADODB 来操作数据库，首先就要获取和安装 ADODB。读者只要到网上下载 ADODB 包，解压到 Web 服务器目录下即可。本书使用的 ADODB 的版本是 5.0。

 注意：要使用 ADODB，使用的 PHP 必须是 4.01 以上的版本。

在详细介绍 ADODB 类库之前，首先了解一下使用 ADODB 类库的简要步骤。

**例 20.1 测试 ADODB 是否配置成功。应用 ADODB 类库连接 MySQL 数据库，指定连接的数据**库为 db\_database20，使用的数据表是 tb\_object。将 ADODB 类库存储在根目录的上级目录中，名称为 adodb5，代码如下：（实例位置：光盘\TM\Instances\20\1）

```

<?php
include_once ('./adodb5/adodb.inc.php');
$conn = ADONewConnection('mysql');
$conn -> PConnect('localhost','root','root','db_database20');
$conn -> execute('set names gb2312');
//载入 adodb.ini.php 文件
//连接 MySQL 数据库
//连接 db_database20 库
//设置编码

```

```

$rst = $conn -> Execute("select * from tb_object") or die('执行错误');
//执行 SQL 语句
while(!$rst -> EOF){ //配合 while 语句循环输出结果
    echo $rst -> fields[1].<br>;
    $rst -> moveNext(); //指针下移
}
$rst -> close();
$conn -> close();
?>

```

结果为：图书 电器 服装

使用 ADODB 类库的一般操作步骤为：

- (1) 载入 adodb.inc.php 文件。要使用 ADODB 类库，首先要启动 ADODB，启动 ADODB 的方法就是载入 adodb.ini.php 文件。
- (2) 建立连接。使用 ADONewConnection() 函数。
- (3) 连接数据库。使用 PConnect() 函数。
- (4) 执行数据库操作。使用 Execute 方法返回结果集。
- (5) 对结果集进行操作。
- (6) 关闭连接。

## 20.4 ADODB 类库

ADODB 类库非常庞大，仅一个 adodb.inc.php 文件大小就有 120KB，所以这里只介绍一些常用函数和变量。这些函数按照功能的不同可分为连接数据库函数、操作数据库函数、公用变量、公用函数、结果集处理函数和错误处理函数等。

### 20.4.1 连接数据库函数

#### 视频讲解：光盘\TM\Video\20\连接数据库函数.exe

在介绍连接数据库的方法之前，先来了解一些在连接数据库过程中应用到的函数。

##### **ADONewConnection(\$databaseType)**

函数作用：连接数据库系统。

参数 \$databaseType 表示要连接的数据库系统的名称，如 mysql、mssql 等。

##### **PConnect(\$host,[\\$user],[\\$password],[\\$database])**

函数作用：持久化连接数据库。

参数说明如下。

- \$host：数据库系统的服务器所在地址。如果是本机操作，格式为 localhost。
- \$user：数据库用户名。
- \$password：数据库密码。
- \$database：使用到的数据库。

Connect()也是连接数据库的函数。该函数为非持久化连接数据库。持久化连接和非持久化连接的

区别在于：持久化连接不用每次都创建新连接，可以增加程序的执行速度。但有些数据库不支持，如果遇到不支持的数据库，可以使用 Connect() 函数代替 PConnect() 函数。

ADODB 支持多种数据库的连接，下面简单介绍几种常用数据库的连接方法。

### 1. MySQL

连接 MySQL 数据库有两种不同的方法。

第一种使用标准的连接字符串来连接，代码如下。

```
<?php
    include_once ('./adodb5/adodb.inc.php');           //载入 (include) adodb.inc.php 文件
    $conn = ADONewConnection('mysql');                   //建立连接
    $conn -> PConnect('localhost','root','root','db_database20'); //连接数据库
?>
```

其中 localhost、root、root 和 db\_database20 分别表示要连接数据库服务器的地址、用户名、密码和数据库名称。

第二种是通过数据源名称 (DSN) 的方式进行连接，代码如下。

```
<?php
    include_once ('./adodb5/adodb.inc.php');           //载入 (include) adodb.inc.php 文件
    $conn = ADONewConnection('mysql://root:root@localhost/db_database20'); //连接数据库
?>
```

其实现的功能是相同的。

### 2. Microsoft SQL Server

连接 Microsoft SQL Server 数据库使用 ODBC 的连接方法，代码如下。

```
<?php
    include_once ('./adodb5/adodb.inc.php');           //载入 (include) adodb.inc.php 文件
    $conn = ADONewConnection('odbc_mssql');             //连接 SQL 数据库
    $conn-> PConnect("Driver={SQL Server};Server=localhost;Database=mydb;","username","password");
?>
```

其中 localhost、username、password 和 mydb 分别表示要连接数据库服务器的地址、用户名、密码和数据库名称。

**例 20.2** 通过 ADODB 使用 ODBC 方法连接 Microsoft SQL Server 数据库，查询 Microsoft SQL Server 系统库 master 中的表 systypes，查询条件是 xtype 的值为 173，并输出查询结果，代码如下：（实例位置：光盘\TM\Instances\20\2）

```
<?php
    include_once ('./adodb5/adodb.inc.php');           //载入 (include) adodb.inc.php 文件
    $conn = ADONewConnection('odbc_mssql');             //连接 SQL 数据库
    $conn-> PConnect("Driver={SQL Server};Server=PKH;Database=master;","sa","");
    $ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;              //设置字段为结果集索引
    $sqlstr = 'select * from systypes where xtype = ?'; //创建 SQL 语句
    $rst = $conn -> execute($sqlstr,'173') or die('connect error'); //执行 SQL，查询 xtype 等于 173 的数据
?>
```

```

while(!$rst->EOF){
    echo $rst->fields['name'].'=>'.$rst->fields['xtype'].";
    $rst->moveNext();
}
$rst->close();
$conn->close();
?>

```

结果为： binary=>173

### 3. Microsoft Access

连接 Access 数据库使用的是 ODBC 方法，代码如下。

```

<?php
include_once('../adodb5/adodb.inc.php');           //载入(include) adodb.inc.php 文件
$conn = ADONewConnection('access');                 //连接 Access 数据库
$conn->PConnect("Driver={Microsoft Access Driver (*.mdb)};Dbq=d:\\mydb.mdb;Uid=Admin;Pwd=");
?>

```

其中， d:\\mydb.mdb 表示数据库所在的文件名。

## 20.4.2 操作数据库函数

### 视频讲解：光盘\TM\Video\20\操作数据库函数.exe

#### 1. Execute(\$sql[\$inputarr=false])

函数作用：执行 SQL 语句，并返回一个结果集（ADORecordSet 对象）；如果失败则返回 false。

**注意：**无论是执行 select 语句，还是 insert 或 update 语句，只要执行成功，execute() 函数都会返回一个结果集。

参数说明如下。

- \$sql：要执行的 SQL 语句。
- \$inputarr：用来作为传入的结合变量。

没有变量 \$inputarr 时，\$sql 为普通的 SQL 语句，executue() 函数的格式为：

```
$connect->Execute('select * from tb_object where id = 1');
```

当使用变量 \$inputarr 时，execute() 函数的格式为：

```
$connect->Execute('select * from tb_object where id = ?',array($v1));
```

结合变量（Binding variables），变量的结合可以加速 SQL 指令编译及读取的速度，产生较佳的效能。

**例 20.3 应用 execute() 函数的第二个参数 \$inputarr，为 where 子句补充上完整的条件，查询出数据表中 ID 等于 2 的记录，代码如下：**（实例位置：光盘\TM\Instances\20\3）

```

<?php
include_once('../adodb5/adodb.inc.php');           //载入(include) adodb.inc.php 文件
$ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;              //设置字段为结果集索引

```

```

$conn = ADONewConnection('mysql');
$conn -> PConnect('localhost','root','root','db_database20');           //建立连接
$conn -> execute('set names gb2312');                                     //连接数据库
$conn -> execute('select * from tb_object where id = ?');                //设置编码格式
$sqlstr = 'select * from tb_object where id = ?';                         //创建 SQL 语句
$rst = $conn -> execute($sqlstr,'2') or die('connect error');            //执行 SQL 语句，查询出 id 等于 2 的数据
while(!$rst -> EOF){                                                       // while 语句循环输出结果
    echo $rst -> fields['bigclass'].'=>'.$rst -> fields['d_time'].'';
    $rst -> movenext();                                                    //输出查询结果
}                                                                           //指针下移
}
$rst -> close();                                                        //关闭连接
$conn -> close();
?>

```

结果为：电器=>2008-02-23 15:23:56

## 2. CacheExecute([\$sec,\$sql[\$inputarr=false]])

函数作用：除了具有 `execute()` 函数的功能外，还可将查询结果保存到缓存中。如果以后还有相同的查询，即可直接从缓存中获取。

参数说明如下。

- `$sec`: 查询结果集在缓存中被保存的时间。
- `$sql`: 执行的 select 查询语句。这里只能是 select 语句。
- `$inputarr`: 结合变量。

## 3. SelectLimit(\$sql[\$numrows=-1][\$offset=-1][\$inputarr=false])

函数作用：返回一个指定起始位置和记录数的结果集。MySQL 数据库支持 limit 查询功能，但 Access 等数据库系统并不支持 limit。ADODB 考虑到了这一点，模拟了一个和 limit 功能类似的函数。这里参数的位置和 MySQL 数据库中 limit 的参数位置不太一样。使用时需要注意。

参数说明如下。

- `$sql`: 要执行的 select 查询语句。
- `$numrows`: 是要查询的记录数。如果该值为-1，则函数会一直查询到最后一条记录。
- `$offset`: 表示从第几条记录开始计算。
- `$inputarr`: 结合变量。

例 20.4 使用 `selectlimit()` 函数查询数据，从第 `$offset` 条记录开始查起，返回 `$numrows` 条记录，最后输出数据。代码如下：（实例位置：光盘\TM\Instances\20\4）

```

<?php
include_once 'conn/conn.php';                                         //载入数据库连接文件
$sql = 'select * from tb_object';                                     //要查询的 SQL 语句
$numrows = 2;                                                       //查找两条记录数
$offset = 1;                                                       //从第一条记录开始计算
$rst = $conn -> selectlimit($sql,$numrows,$offset) or die('execute error'); //调用 selectlimit()函数
while(!$rst -> EOF){                                                 //如果不是最后一条记录，循环
    echo $rst -> fields['id'].'=>'.$rst -> fields['bigclass']; //输出 id 和字段值
    echo '<br>';
    $rst -> movenext();                                              //指针移动一位
}
?>

```

```

    }
?>

```

结果为：电器=>2008-03-05 09:17:57 服装=>2008-02-25 14:56:21

#### 4. CacheSelectLimit([\$sec,] \$sql[, \$numrows=-1][,\$offset=-1][,\$inputarr=false])

函数作用：除了具有 selectlimit() 函数的功能外，还将查询结果保存到缓存当中。如果以后有相同的查询，将直接从缓存中获取。

参数含义同上。

#### 5. CacheFlush()

函数作用：清除所有 ADODB 数据库的缓存。

#### 6. GetUpdateSQL(&\$rs, \$arrFields, \$forceUpdate=false)

函数作用：更新记录信息。

参数说明如下。

- \$rs：要更新的结果集。
- \$arrFields：更新字段及更新内容。
- \$forceUpdate：如果 \$forceUpdate 为 true，无论 \$arrFields 和 \$rs 的值是否相等都将更新。

例 20.5 本例中首先查找 ID 等于 2 的记录，然后创建要更新字段的数组，最后将该数组内容通过 getUpdateSQL() 函数更新到数据库中。代码如下：（实例位置：光盘\TM\Instances\20\5）

```

<?php
    include_once ('./adodb5/adodb.inc.php'); //载入 (include) adodb.inc.php 文件
    $conn = ADONewConnection('mysql'); //建立连接
    $conn -> PConnect('localhost','root','root','db_database20'); //连接数据库
    $conn -> execute('set names gb2312'); //设置编码格式
    $sqlstr = 'select * from tb_object where id = 2'; //声明 SQL 语句
    $rst = $conn -> execute($sqlstr) or die('Error: '.$conn -> errorMsg()); //执行 SQL 语句
    echo '修改前的数据，时间为： '.$rst -> fields['d_time'].'<br>'; //输出数据库中的时间
    $fields = array(); //声明一个空数组$fields
    $fields['d_time'] = date("Y-m-d H:i:s"); //将当前时间存到数组中
    $update = $conn -> getUpdateSQL($rst,$fields) or die('update error: '.$conn -> errorMsg()); //更新数据
    $conn -> execute($update); //执行更新
    echo '修改后的数据，时间为： '; //执行查询
    $rst = $conn -> execute($sqlstr); //输出更新后的信息
    echo $rst -> fields['d_time'];
    $rst -> close(); //关闭连接
    $conn -> close();
?>

```

结果为：

更新前的时间：2009-04-28 03:19:29

更新后的时间：2009-04-28 03:21:16

#### 7. GetInsertSQL(&\$rs, \$arrFields)

函数作用：添加新记录。

参数说明如下。

\$rs: 要添加记录的结果集。

\$arrFields: 新记录的字段值及对应字段名。

例 20.6 使用 getInsertSQL() 函数向表 tb\_object 中插入一条新数据。实例代码如下：（实例位置：光盘\TM\Instances\20\6）

```

<?php
    include_once ('./adodb5/adodb.inc.php'); //载入(include) adodb.inc.php 文件
    $ADODB_FETCH_MODE = ADODB_FETCH_ASSOC; //设置结果集按照字段名称为索引进行存取
    $conn = ADONewConnection('mysql'); //建立连接
    $conn -> PConnect('localhost','root','root','db_database20'); //连接数据库
    $conn -> execute('set names gb2312'); //设置编码格式
    $sqlstr = 'select * from tb_object where id = 0'; //生成 SQL 语句
    $rst = $conn -> execute($sqlstr) or die('Error: '.$conn -> errorMsg()); //执行 SQL 语句
    $fields = array(); //创建一个数组
    $fields['bigclass'] = '办公用品'; //向数组中添加数据
    $fields['d_time'] = date("Y-m-d H:i:s");
    $insert = $conn -> getInsertSQL($rst,$fields) or die('update error: '.$conn -> errorMsg()); //添加新数据
    $conn -> execute($insert); //执行添加
    echo '添加的数据为: ';
    $showsqli = "select * from tb_object where d_time = '".$fields['d_time']."'"; //创建 SQL 语句
    $rst = $conn -> execute($showsqli); //查找刚创建的记录
    foreach($rst -> fields as $nm => $vl) //使用 foreach 输出新记录
        echo $nm.'=>'.$vl.' ';
    $rst -> close(); //关闭连接
    $conn -> close();
?>

```

结果为，

添加的数据为：id=>67 bigclass=>办公用品 d\_time=>2009-04-28 03:31:34

### 8. DBDate(\$date)

函数作用：转换不同数据库之间的日期格式。例如，MySQL 使用的时间格式为 Y-m-d，Oracle 则为 D-M-Y。

参数\$date 是要存储的时间。

### 9. qstr(\$string)

函数作用：使用引号来处理字符串。例如，一个字符串包含了单引号 ‘’，在 MySQL 中可以直接使用 ‘’ 表示，但在其他的数据库中，则使用两个单引号 “” 表示。使用 qstr() 函数可以处理和解决这个问题。

参数\$string 为要被处理的字符串。

例 20.7 使用 DBDate() 函数和 qstr() 函数向表 tb\_object 中插入一条数据，代码如下：（实例位置：光盘\TM\Instances\20\7）

```

<?php
    include_once 'conn/conn.php'; //载入数据库连接文件

```

```

$bigclass = $conn -> qstr("建筑耗材"); //用 qstr() 函数处理带引号的字符串
$d_time = $conn -> DBDate(date('Y-m-d H:i:s'));//用 DBDate() 函数处理日期时间
$sql = "insert into tb_object(bigclass,d_time) value('.$bigclass.','.$d_time.')";//生成 insert 语句
$rst = $conn -> execute($sql) or die($conn -> errorMsg());//执行 insert 语句
if($rst)
    echo '添加成功'; //添加成功
?>

```

结果为：添加成功

说明：在本实例中，为了使代码整洁、便于对使用数据库的管理，将连接数据库的操作定义到 conn.php 文件中，在使用时只要通过 include\_once 语句进行调用即可。

### 10. Affected\_rows()

函数作用：返回最后更新或被删除的记录数。如果数据库不支持，则返回 false。

例 20.8 对 ID=2 的记录进行修改，将该记录的 d\_time 字段值设为当前时间，然后调用 Affected\_rows() 函数，查看返回结果，代码如下：（实例位置：光盘\TM\Instances\20\8）

```

<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    $sqlstr = 'update tb_object set d_time = now() where id = 2'; //SQL 更新语句
    $rst = $conn -> execute($sqlstr) or die('update error: '.$conn -> errorMsg()); //执行更新语句
    echo '更新记录数: '.$conn -> Affected_rows(); //查看更新的记录数
?>

```

结果为，

被更新的记录数是：1

### 11. Insert\_ID()

函数作用：返回最后插入的记录 ID 值。如果数据库不支持该函数，则返回 false。

## 20.4.3 控制结果集存取方式函数

视频讲解：光盘\TM\Video\20\控制结果集存取方式函数.exe

### 1. \$ADODB\_COUNTRECS

当该变量为 true 时，RecordCount() 函数会在数据库驱动不支持 select 指令返回的记录总数时进行自动模拟，返回记录数，默认值为 true。每次进行查询操作时都会自动检查该变量的值。由于这个功能十分消耗内存，所以不建议使用。

### 2. \$ADODB\_CACHE\_DIR

该变量用于设置缓存目录。一般在使用缓存函数时，会用到该变量，如 CacheExecute() 函数。

### 3. \$ADODB\_FETCH\_MODE

该变量决定结果集以哪种方式进行存取。该变量有 4 个值：

(1) define('ADODB\_FETCH\_DEFAULT',0)

(2) define('ADODB\_FETCH\_NUM',1)

结果集按照字段序号为索引进行存取，如\$rst->fields[0]、\$rst->fields[1]。

(3) define('ADODB\_FETCH\_ASSOC',2)

结果集按照字段名称为索引进行存取，如\$rst->fields[bigclass]、\$rst->fields[id]。

(4) define('ADODB\_FETCH\_BOTH',3)

结果集同时支持(2)、(3)两种方式。如果未设置该变量，默认值为(1)，不同类型的数据库驱动默认值也不相同。这里不建议使用变量(4)，因为有很多的驱动并不支持。

**例 20.9** 仍使用例 20.1 中的内容，只是增加了一个公共变量，将结果集以字段名作为索引进行输出，代码如下：(实例位置：光盘\TM\Instances\20\9)

```
<?php
    include_once ('./adodb5/adodb.inc.php');
    $ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;
    $conn = ADONewConnection('mysql');
    Connect($host,[ $user ],[$password],[ $database ]);
    $conn -> PConnect('localhost','root','root','db_database20');
    $conn -> execute('set names gb2312');
    $rst = $conn -> Execute('select * from tb_object') or die('执行错误');
    while(!$rst -> EOF){
        echo $rst -> fields['id'].'=>'.$rst -> fields['bigclass'].';
        $rst -> movenext();
    }
    $rst -> close();
    $conn -> close();
?>
```

结果为：1=>图书 2=>电器 3=>服装

#### 20.4.4 操作结果集函数

视频讲解：光盘\TM\Video\20\操作结果集函数.exe

当使用 execute() 函数执行 SQL 指令时，会回传一个结果集。该结果集实际上是一个 ADOResultSet 对象。通过对该对象的控制，可以对结果进行各项操作。

##### 1. fields

该变量保存的是当前指针所指向的记录。

##### 2. EOF

该变量记录当前指针是否指向最后一条记录。如果是最后一条记录，则设定为 true，否则为 false。

##### 3. GetArray([\$number\_of\_rows])

函数作用：返回从当前指针指向的记录开始，到 (\$number\_of\_rows-1) 行的全部记录的数组。

参数 \$number\_of\_rows 表示指定的记录行。如果没有给出，则一直到 EOF 才停止。

##### 4. UserDate(\$str, [\$fmt])

函数作用：将日期字符串\$str 转换为变量\$fmt 设置的日期格式。

参数说明如下。

- \$str: 日期格式的字符串。
- \$fmt: 要转换的日期格式。默认值为 Y-m-d。

#### 5. UserTimeStamp(\$str, [\$fmt])

函数作用：将时间字符串\$str 转换为变量\$fmt 设置的时间格式。

参数说明如下。

- \$str: 时间字符串。
- \$fmt: 时间格式设置字符串。默认值为 Y-m-d H:i:s。

#### 6. MoveNext()

函数作用：将 ADORecordSet（结果集）的指针下移一位。如果成功，则返回 true，否则返回 false。

#### 7. Move(\$to)

函数作用：将 ADORecordSet（结果集）的指针移动到指定位置。如果\$to 等于 0，则指针指向结果集的第一条数据；如果\$to 的值大于结果集，则指针指向最后一条数据。注意，这里变量\$to 只能是绝对定位。

#### 8. MoveFirst()

函数作用：指针移动到第一条数据。等同于 Move(0)。

#### 9. MoveLast()

函数作用：指针移动到最后一条数据。等同于 Move(RecordCount()-1)。

#### 10. FetchRow()

函数作用：返回当前指针指向的记录的数组，如果是 EOF，则回传 false。FetchRow()不要和 MoveNext()混用。

#### 11. FetchField(\$column\_number)

函数作用：返回一个对象，包含字段\$column\_number 的名称、空间长度等相关信息。

参数\$column\_number 表示要查看的字段名称。

#### 12. FetchNextObject([\$toupper=true])

函数作用：返回当前指针所指向的记录的对象形式，并且指针自动下移一行。

参数\$toupper: 如果等于 true，则字段名为大写形式。

#### 13. FieldCount()

函数作用：返回结果集中的字段数。

#### 14. RecordCount()

函数作用：返回结果集中的记录数。

#### 15. CurrentRow()

函数作用：返回当前指针所指的记录序号。第一条记录用 0 表示。

### 16. MetaType(\$nativeDBType [, \$field\_max\_length], [\$fieldobj])

函数作用：标准化不同数据库系统下的数据类型的表示法。处理不同的数据库系统有一个问题：每一个数据库系统对于相同的数据类型会有不同的表示法，如支持超长文本的类型，有的表示为 `text`，有的表示为 `long character`，还有 `clob` 等。`MetaType()` 函数可以将这几种表示法统一起来。如用大写字母 `D` 来表示所有的日期类型，用大写字母 `T` 表示所有的时间类型。

参数说明如下。

- `$nativeDBType`: 数据表中的数据类型。
- `$fields_max_length`: 字段最大长度。
- `$fieldobj`: 字段对象。

`MetaType()` 函数支持的标准化数据类型如下。

- `C`: 支持 `char`、`varchar` 等字符类型。
- `X`: 支持 `text`、`long character` 等长类型。
- `B`: `blob` 或 `binary image`。
- `D`: 日期 `date`。
- `T`: 时间 `timestamp`。
- `L`: 逻辑类型和位类型。
- `N`: 包含编号、整型、浮点型等数字类型。
- `R`: 包含序列、自动增加整数等序列类型。

例 20.10 通过 `FieldCount()` 函数获取结果集记录数，然后通过 `for` 循环输出数据，并对每个字段类型进行判断，如果类型为“T”，则使用 `DBDate()` 函数格式化输出时间，其他类型则直接输出。代码如下：（实例位置：光盘\TM\Instances\20\10）

```
<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    $sqlstr = 'select * from tb_object where id = 1'; //SQL 查询语句
    $rst = $conn -> execute($sqlstr) or die('error: ' . $conn -> errorMsg()); //执行查询语句
    if(false != $rst){ //如果有查询结果
        for($i = 0; $i < $rst -> FieldCount(); $i++){ //循环输出各个字段
            $fields = $rst -> FetchField($i); //生成字段信息对象
            $type = $fields -> type; //从对象中获取字段的类型信息
            echo '>';
            echo $rst -> metaType($type,-1,$fields); //输出字段标准类型
            if($rst -> metaType($type,-1,$fields) == "T") //如果标准类型为“T”
                echo '('. $conn -> DBDate($rst -> fields[$i]).')'; //使用 DBDate 函数格式化时间
            else //如果是其他类型，直接输出
                echo '('. $rst -> fields[$i].')';
        }
    }
?>
```

结果为：=>R(1)=>C(图书)=>T('2008-02-27 16:46:48')

## 20.4.5 处理事务函数

### 1. BeginTrans()

函数作用：开启事务处理。该函数和接下来的两个函数都是应用在 PHP 的事务处理中。

### 2. CommitTrans()

函数作用：如果成功完成数据库操作，则执行 CommitTrans() 函数。

### 3. RollbackTrans()

函数作用：结束一次操作，恢复操作前的所有改变。如果成功则返回 true，如果服务器不支持则返回 false。

## 20.4.6 生成 HTML 表格函数

### rs2html(\$rst,[ \$table\_attributes], [ \$col\_titles])

函数作用：返回一个 HTML 表格格式的结果集。要使用这个函数，需要载入 tohtml.inc.php 文件。参数说明如下。

\$rst：要返回的结果集。

\$Table\_attributes：对表格的参数及属性的设置。

\$col\_titles：对字段的重新命名。

例 20.11 应用 rs2html() 函数直接输出返回的结果集 \$rst，并设置表格的属性，代码如下：（实例位置：光盘\TM\Instances\20\11）

```
<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    include_once './adodb5/tohtml.inc.php'; //载入 tohtml.inc.php 文件
    $rst = $conn -> execute('select * from tb_object'); //返回查询结果集
    rs2html($rst, 'width="350" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#FF0000",
array('序号','类别','添加时间')); //输出结果集
?>
```

运行结果如图 20.1 所示。

序号	类别	添加时间
1	图书	Wed 27, Feb 2008, 04:46:46
2	电器	Tue 28, Apr 2009, 12:17:20
3	服装	Mon 25, Feb 2008, 02:58:21
74	建筑耗材	Tue 28, Apr 2009, 03:51:27

图 20.1 应用 rs2html() 函数直接输出返回的结果集

### 20.4.7 生成下拉列表框函数

视频讲解：光盘\TM\Video\20\生成下拉列表框函数.exe

1. GetMenu(\$name, [\$default\_str = ""], [\$blank = true], [\$multiple\_select = false], [\$size = 0], [\$moreAttr = ""])

函数作用：返回一个 HTML 下拉列表框（select）。结果集的第一列（fields[0]）被显示到下拉列表当中，第二列（fields[1]）被设置为<option>标签的 value 属性。

参数说明如下。

- ☒ \$name：下拉列表框的名称。
- ☒ \$default\_str：如果\$default\_str 的值等于其中一个 fields[0]，那么这个 fields[0]就是该下拉列表的默认选项。这相当于为这个 fields[0]添加了属性<selected>。
- ☒ \$blank：该参数的默认值为 true。在生成下拉列表框时，第一行选项默认为空，即<option></option>。如果为 false，则没有默认的空选项。
- ☒ \$multiple\_select：是否支持多选。默认值为 false，即不允许。
- ☒ \$size：列表框选取的大小。默认为 0，生成一个下拉列表；如果 \$size 的值大于 0，则生成一个列表框。
- ☒ \$moreAttr：该参数可以增加列表框的属性，如添加一个 JavaScript 脚本等。

2. GetMenu2(\$name, [\$default\_str=""], [\$blank1stItem=true], [\$multiple\_select=false], [\$size=0], [\$moreAttr=""])

函数作用和参数含义同上。该函数和 GetMenu() 函数的不同之处是：getMenu() 函数中的参数 \$default\_str 和 fields[0] 比较，如果相等，那么 fields[0] 就是这个下拉列表的默认选项；而 getMenu2() 函数中的参数 \$default\_str 则是和 fields[1] 进行比较，其他不变。

例 20.12 本例中首先使用查询语句和 execute() 函数返回一个结果集，然后在一个 form 表单中使用 getMenu2() 函数将结果集输出到下拉列表框。当单击“提交”按钮时，通过 \$\_POST[] 预定义变量获取下拉列表框提交的值，代码如下：（实例位置：光盘\TM\Instances\20\12）

```
<?php
include_once 'conn/conn.php'; //载入数据库连接文件
$sqlstr = 'select bigclass,id from tb_object';
$rst = $conn -> execute($sqlstr) or die('execute error: ' . $conn -> errorMsg()); //SQL 查询语句
//执行查询语句，返回结果集
?>
<!-- 提交表单 --&gt;
&lt;form method='post' action=""&gt;
请选择购买商品的类别：
<!-- 调用 getMenu2 函数 --&gt;
&lt;?php echo $rst -&gt; getMenu2('bigclass',$_POST['bigclass'],false); ?&gt;
&lt;input type='submit' name='submit' value='提交' /&gt;
&lt;/form&gt;
<!-- ----- --&gt;
&lt;?php</pre>

```

```

if(isset($_POST[bigclass])){
    echo '您选择的商品类别是: '.$_POST[bigclass];
}
?>

```

运行结果如图 20.2 所示。

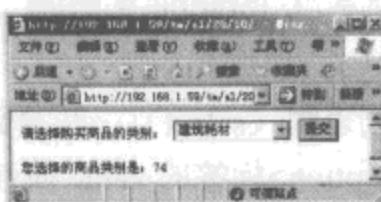


图 20.2 使用 getMenu2() 函数输出下拉列表框

#### 20.4.8 实现分页功能函数

##### 视频讲解：光盘\TM\Video\20\实现分页功能函数.exe

###### 1. PageExecute(\$sql, \$nrows, \$page)

函数作用：分页功能函数。

参数说明如下。

- \$sql: SQL 查询语句。
- \$nrows: 每页显示的记录数。
- \$page: 保存当前页数，默认为 1。

###### 2. CachePageExecute(\$sec, \$sql, \$nrows, \$page)

函数作用：分页显示函数，同时将显示后的记录放到缓存中。在\$sec 秒内，如果是重复查看，将从缓存中读取数据。

参数\$sec 表示数据在缓存中保留的时间。

其他参数见 PageExecute() 函数。

###### 3. AbsolutePage(\$page=-1)

函数作用：返回当前的页数。要和 PageExecute() 函数配合使用，用于分页。

###### 4. AtFirstPage(\$status="")

函数作用：如果当前页是第一页，返回 true。要和 PageExecute() 函数配合使用，用于分页。

###### 5. AtLastPage(\$status="")

函数作用：如果当前是最后一页，返回 true。要和 PageExecute() 函数配合使用，用于分页。

###### 6. ADODB\_Pager(\$conn, \$sql, \$id = 'adodb', \$showPageLinks = false)

函数说明：该函数是类 adodb\_pager 的构造函数，通过类中的 render() 函数可以实现一个巧妙的分页操作。如果要使用这个功能，需要载入 adodb\_paper.inc.php 页。

参数说明如下。

- \$conn: 数据库连接对象。

- \$sql: 执行的 SQL 语句。
- \$id: 每个分页的 id 号。
- \$showPageLinks: 是否显示各个页的链接, 默认为 false。

例 20.13 使用 adodb\_pager 类将表 tb\_object 中的记录分页显示出来。在本实例中对载入文件 adodb\_pager.ini.php 进行了修改, 修改代码如下: (实例位置: 光盘\TM\Instances\20\13)

```
var $first = '<code>首页</code>';
var $prev = '<code>上一页</code>';
var $next = '<code>下一页</code>';
var $last = '<code>尾页</code>';
```

上述修改是为了将翻页改成中文连接。同样, 在应用的过程中还可以修改 adodb\_pager.ini.php 文件中的其他内容, 如表格的背景、宽度, 以及边框的宽度、颜色等。

本实例的代码如下:

```
<?php
    include 'conn/conn.php'; //载入数据库连接文件
    include _once './adodb5/adodb_pager.inc.php'; //载入 adodb_pager.ini.php 文件
    /* 声明对象 */
    $pager = new ADODB_Pager($conn, "select id,bigclass as '大类', d_time as '添加时间' from tb_object");
    /* 调用$Render 函数确定每页显示多少 */
    $pager -> Render(3);
?>
```

运行结果如图 20.3 所示。

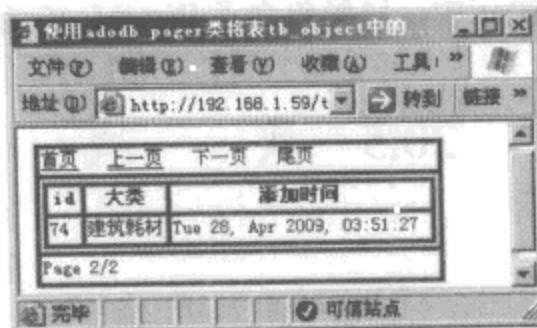


图 20.3 adodb\_pager 类用于显示记录分页

#### 20.4.9 错误处理函数

ADODB 的优势是可以让各种数据库函数统一起来, 对相同的数据进行相同的操作。但在实际操作时, 难免会出现各种问题。下面介绍几种错误处理及调试的方法。

##### 1. debug

如果变量被设定为 true, 当有输出操作时, 同时也输出调试信息。

##### 2. ErrorMsg()

函数作用: 返回最后的状态或出错信息。即使没有错误发生, 也会返回一个字串。所以, 一般情

况下，只有在发生错误时（返回 `false`）才调用该函数。在启用 `debug` 变量后，只要有错误发生就会自动调用该函数。

例 20.14 人为地造成一个错误，将 SQL 语句中的 from 写成 form，来看一下输出的错误信息，代码如下：（实例位置：光盘\TM\Instances\20\14）

```
<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    $conn -> debug = true; //开启调试
    $sql = 'select * from tb_object'; //SQL 查询语句
    $rst = $conn -> execute($sql) or die($conn -> errorMsg()); //调用查询语句
?>
```

运行结果如图 20.4 所示。

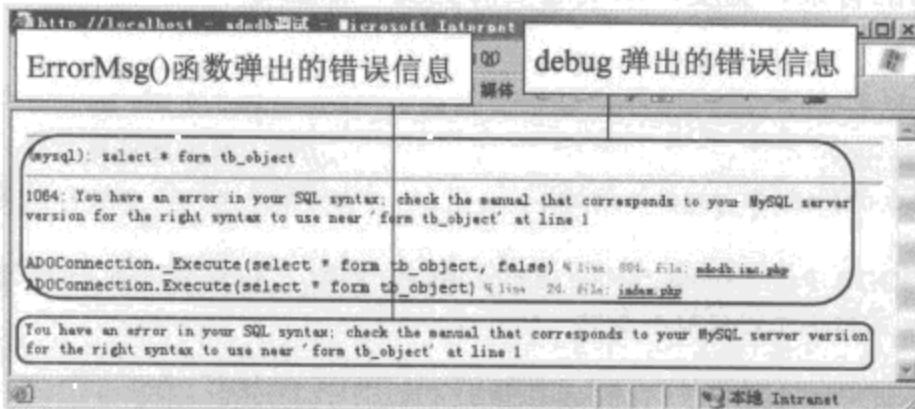


图 20.4 ADODB 调试

可以发现，debug 弹出的错误提示中，已经包含了ErrorMsg()函数的错误信息。

## 20.5 实战

### 20.5.1 实现分页



 视频讲解：光盘\TM\Video\20\实现分页.exe

**例 20.15** 除了可以使用 adodb\_pager.inc.php 中的类实现分页效果外,用 ADODB 类库中默认的几个函数同样可以实现这样的效果。分页主要使用 PageExecute()、AbsolutePage()、AtFirstPage()、AtLastPage()和 LastPageNo() 5 个函数来完成。实例代码如下: (实例位置: 光盘\TM\Instances\20\15)

```
<table width="384" border="0" cellpadding="0" cellspacing="0">
    <tr><td height="30">
<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    include('../adodb5/tohtml.inc.php'); //载入 tohtml.inc.php 文件
    $sql = 'select * from tb_object'; //查询语句
    $num = 2; //每页显示的记录数
    if(isset($_GET['n_page'])) { //判断当前页码
```

```

$c_page = $_GET[n_page]; //将$n_page 赋给变量$c_page
} else {
    $c_page = 1; //初始化变量$c_page
}
$rst = $conn -> PageExecute($sql,$num,$c_page); //执行 PageExecute()函数
if(false != $rst){
    if(!$rst -> AtfirstPage()){ //如果当前页不是首页
?>        <!-- 输出向上翻页超链接 -->
        <a href="<?php echo "?n_pge=1' ?>"> 首页 </a>
        <a href="<?php echo "?n_page='".($rst -> AbsolutePage() - 1); ?>"> 上一页 </a>
        <!-- ----- -->
<?php
    }
    if(!$rst -> AtlastPage()){ //如果当前页不是尾页
?>        <!-- 输出向下翻页超链接 -->
        <a href="<?php echo "?n_page='".($rst -> AbsolutePage() + 1); ?>"> 下一页 </a>
        <a href="<?php echo "?n_page='".($rst -> LastPageNo()); ?>"> 尾页 </a>
        <!-- ----- -->
<?php
    }
?>
        </td></tr>
        <tr><td>
<?php
    rs2html($rst,'width=350 border="1" bgcolor="#FF0000",array('ID','类型','添加时间'));
?>
        </td></tr>
<?php } ?>
        <tr><td height="30">
            当前是第<?php echo $rst -> AbsolutePage(); ?>页/一共是<?php echo $rst -> LastPageNo(); ?>页</td></tr>
</table>

```

运行结果如图 20.5 所示。

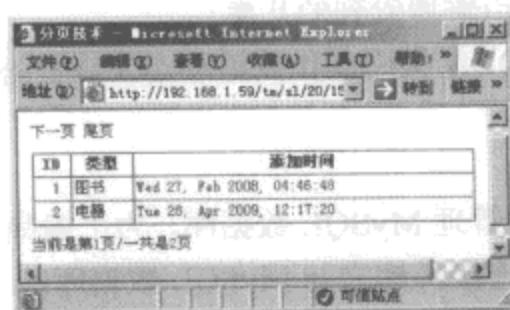


图 20.5 分页技术

## 20.5.2 处理事务

视频讲解：光盘\TM\Video\20\处理事务.exe

在对数据库进行操作时，由于发生意外，导致数据更新一半时就停止了。这时可以通过事务将数

据恢复到更新以前。下面就来实现一个通过事务实现回滚的实例。

**例 20.16 在 ADODB 类库中，通过 BeginTrans()、CommitTrans() 和 RollbackTrans() 3 个函数实现处理事务。实例代码如下：（实例位置：光盘\TM\Instances\20\16）**

```
<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    $conn -> BeginTrans(); //开始事务处理
    $sql = 'delete from tb_object where id = 2'; //SQL 删除语句
    $rst = $conn -> execute($sql) or die('execute error: ' . $conn -> ErrorMsg()); //执行删除语句
    $num = $conn -> Affected_rows(); //查看被更新的记录数
    if(false !== $rst){ //如果 $rst 不为假
        if($num != 0){ //如果 $num 不为 0，说明删除成功
            $conn -> CommitTrans(); //执行提交
            echo '删除成功！';
            exit();
        }else{ //如果 $num 为 0，说明没有删除记录
            echo '没有数据，或数据已删除';
            exit();
        }
    }else{ //如果发生意外
        $conn -> RollbackTrans(); //执行回滚操作
        echo '出现意外。';
    }
?>
```

本实例如果存在指定的数据，则执行删除后返回“删除成功！”；如果不存在则输出“没有数据，或数据已经删除”；如果出现意外则执行回滚，并输出“出现意外。”

### 20.5.3 缓存函数+ADODB 动态生成静态页

 视频讲解：光盘\TM\Video\20\缓存函数+ADODB 动态生成静态页.exe

现在很多站点都采用动态生成静态页面的方式，这样不仅可以有效地提高站点的访问效率、降低服务器负载，而且可以提高被搜索引擎搜索到的几率。

**例 20.17 使用缓存函数+ADODB 类库制作能够动态生成静态页面的学生成绩管理系统。（实例位置：光盘\TM\Instances\20\17）**

本实例的实现步骤如下：

(1) 本实例使用 ADODB 技术管理 MySQL 数据库，为了提高代码重用率，体现 PHP 面向对象编程的特点，开发实例时创建数据库连接类建立与 MySQL 数据库的连接，代码如下。

```
<?php
class ConnDB
{
    var $dbType; //数据库类型
    var $host; //MySQL 数据库服务器地址
    var $userName; //用户名
    var $password; //密码
    var $dbName; //数据库名
```

```

var $isDebug;                                //是否调试
var $connID;                                 //保存数据库连接标识

function ConnDB ($dbType = 'mysql', $host, $userName, $password, $dbName, $isDebug = false)
{
    //构造函数，用于对类中的属性进行初始化
    $this->dbType = $dbType;
    $this->host = $host;
    $this->userName = $userName;
    $this->password = $password;
    $this->dbName = $dbName;
    $this->isDebug = $isDebug;
}

function getConnID ()
{
    require_once 'library/adodb/adodb.inc.php';           //导入 ADODB 类库
    $this->connID = NewADOConnection($this->dbType);      //建立连接
    if ($this->dbType == 'mysql' || $this->dbType == 'mssql') { //MySQL 或 SQL Server 数据库
        $this->connID->Connect($this->host, $this->userName, $this->password, $this->dbName);
        if ($this->dbType == 'mysql') {
            $this->connID->Execute('set names gbk');          //如果是 MySQL 数据库则设置字符集
        }
    } elseif ($this->dbType == 'access') {                  //Access 数据库
        $this->connID->Connect('Driver = {Microsoft Access Driver (*.mdb)}; Dbq =' . realpath($this->dbName) .
        ';Uid =' . $this->userName . ';Pwd =' . $this->password . ';');
    } else {
        return false;
    }
    $this->connID->debug = $this->isDebug;                //是否调试
    return $this->connID;
}

function closeConnID ()                          //关闭与数据库的连接
{
    @$this->connID->Disconnect();
}
}

```

上述代码主要应用 ConnDB 类通过 ADODB 类库建立与 MySQL 数据库的连接。分析代码可知，ConnDB 类不仅可以建立与 MySQL 数据库的连接，而且可以建立与 Access 和 SQL Server 数据库的连接。ConnDB 类中首先使用构造方法对类中的属性进行实例化，然后创建 getConnID()方法用于返回数据库连接的 ID，最后创建 closeConnID()方法释放数据库连接资源。

(2) 建立基于 ADODB 类库的分页类，用于实现学生成绩信息的分页显示，代码如下。

```

<?php
class PageDB
{
    function pageData ($sql, $connID, $pageSize, $page)
    {
        $rs = $connID->Execute($sql);
        $totalRecord = $rs->RecordCount();
        $totalPage = ceil($totalRecord / $pageSize);
        $rs->Close();
        $rs = null;
        $list = array();
        $start = ($page - 1) * $pageSize;
        $end = $start + $pageSize;
        for ($i = $start; $i < $end; $i++) {
            $list[] = $rs->GetRow($i);
        }
        return $list;
    }
}

```

```

    {
        $arrayPageInfo = array(); // 定义要返回的数组
        if (isset($page) && $page != "") // 判断是否有页码传递
            $nowPage = $_GET['page']; // 有则获取页码的值
        } else {
            $nowPage = 1; // 没有则默认显示第一页
        }
        $rs = $connID->PageExecute($sql, $pageSize, $nowPage);
        $arrayData = $rs->GetRows(); // 执行查询
        if (count($arrayData) == 0 || $rs == false) {
            return false; // 如果出错或没有查询到则返回 false
        } else {
            $arrayPageInfo['data'] = $arrayData; // 查询到的数据
            $rsAll = $connID->Execute($sql);
            $countRs = count($rsAll->GetRows()); // 总记录数
            $countPage = ceil($countRs / $pageSize); // 总页数
            $arrayPageInfo['pageSize'] = $pageSize; // 将分页信息保存到要返回的数组$arrayPageInfo 中
            $arrayPageInfo['countRs'] = $countRs;
            $arrayPageInfo['page'] = $nowPage;
            $arrayPageInfo['countPage'] = $countPage;
            $arrayPageInfo['first'] = 1;
            if ($page > 1) {
                $arrayPageInfo['previous'] = $rs->AbsolutePage() - 1;
            } else {
                $arrayPageInfo['previous'] = 1;
            }
            if ($page < $countPage) {
                $arrayPageInfo['next'] = $rs->AbsolutePage() + 1;
            } else {
                $arrayPageInfo['next'] = $countPage;
            }
            $arrayPageInfo['last'] = $countPage;
        }
        return $arrayPageInfo; // 返回包含所有分页信息的数组
    }
}

```

PageDB 类主要实现 ADODB 类库操作 MySQL 数据库的分页显示，类中所定义的 pageData()方法用于返回一个参数数组，该数组包含所有分页信息，如每页显示的记录数、总页数、总记录数及分页导航链接具体页码值等，这样只需根据返回数组即可获取所有分页信息，为进一步开发做好铺垫。

(3) 建立类 StaticPage 用于动态生成静态页面，该类主要使用 PHP 中缓存函数实现，具体实现代码如下。

```

class StaticPage
{
    function pageBegin () // 指定要生成静态页的开始部分
    {
        ob_start();
    }
}

```

```

function pageEnd () //定义要生成静态页的结束部分
{
    ob_end_flush();
}

function getStaticPageName ($url, $extendsName, $defaultPage = 'index')
{
    //根据动态页的实际地址生成静态页的名称
    $baseName = basename($url); //不包含目录的 PHP 动态页名
    $pageName = substr($baseName, 0, strpos($baseName, '.php'));//去掉扩展名
    if ($pageName == "") {
        $pageName = $defaultPage;
    }
    $searchString = substr(strchr($baseName, '?'), 1, strlen(strchr($baseName, '?'))); //获取查询字符串部分
    $arraySearchString = explode('&', $searchString); //分割查询字符串并保存到数组中
    $searchName = "";
    for ($i = 0; $i < count($arraySearchString); $i++) { //使用 “-” 连接查询字符串的值
        $searchStringTmpValue = strstr($arraySearchString[$i], '=');
        if ($searchStringTmpValue != "") {
            $searchName .= '-' . substr($searchStringTmpValue, 1, strlen($searchStringTmpValue));
        }
    }
    $fileName = $pageName . $searchName . '.' . $extendsName; //生成静态页文件名
    return $fileName; //返回静态页文件名
}

function makeStaicPage ($url, $extendsName, $saveDir)
{
    //生成静态页文件
    $content = ob_get_contents(); //获取 PHP 页面生成的 HTML 页面内容
    if (! is_dir($saveDir)) { //创建保存静态页的目录
        mkdir($saveDir);
    }
    $fileName = $this->getStaticPageName($url, $extendsName);
    $fp = fopen($saveDir . '/' . $fileName, 'w'); //将 HTML 内容写入静态页
    fwrite($fp, $content);
}

function reMakeCondition ($saveDirAndFile, $reMakeTime = -1)
{
    //设定重新生成静态页的条件
    if ($reMakeTime == -1) {
        if (file_exists($saveDirAndFile)) {
            return true;
        }
    } elseif (file_exists($saveDirAndFile) && time() - filemtime($saveDirAndFile) < $reMakeTime) {
        //如果静态页超出指定的时间范围或不存在则返回 true, 表示将重新生成静态页
        return true;
    }
}

```

```

        }
        return false;
    }
}

```

上述代码中定义的 pageBegin()方法和 pageEnd()方法分别用来定义所要生成的静态页面的开始和结束位置，定义的 getStaticPageName()方法用来根据实际 PHP 页面的请求地址生成静态页面地址，makeStaicPage()方法用于创建静态页面，最后定义 reMakeCondition()方法用于指定重新生成静态页面的条件。

分析上述代码可知，本实例所讲述的生成静态页的方法是真正意义上的生成静态页面，而非伪静态，这样的好处是不受服务器局限，而使用伪静态的方法需要开启 Apache 服务器的 Rewrite 功能，如果用户使用租用的空间，可能有些空间提供商未提供上述技术支持。

(4) 建立 index.php 页面，用于实现学生信息的分页显示，在页面最开始调用 StaticPage 类，并根据 reMakeCondition()方法判断与当前 PHP 页面所对应的静态页面是否存在或静态页面是否已经过期，如果是则重新生成，否则使用 header()函数重新定向到与该页面所对应的静态页面中，并使用 exit()函数终止该 PHP 页面剩余代码的执行，从而提高代码执行效率。

```

<?php
require_once 'StaticPage.php'; //包含 StaticPage 类
$ststicPage = new StaticPage(); //对 StaticPage 进行实例化
$ststicPage->pageBegin(); //定义生成静态页面的起始位置
$dirName = './pages'; //定义生成静态页的目录
$fileName = $ststicPage->getStaticPageName($_SERVER['REQUEST_URI'], 'shtml'); //生成与该 PHP 页面对应的静态页的名称
if ($ststicPage->reMakeCondition($dirName . '/' . $fileName, 60)){ //判断是否需要重新生成静态页
    header('location:pages/' . $fileName); //不需要则重新定向到与该 PHP 页面所对应的静态页面
    exit();
}
$baseUrl = '/TM/Instances/20/17';
require_once 'ConnDB.php'; //包含数据库连接类
require_once 'PageDB.php'; //包含分页类
$connObj = new ConnDB('mysql', 'localhost', 'root', 'root', 'db_database20', false); //实例数据库连接类
$connID = $connObj->getConnID(); //返回数据库连接 ID
?>

```

(5) 实例 PageDB 分页类，通过 for 循环显示学生信息。

```

<?php
$pageObj = new PageDB(); //实例分页类
$pageInfo = $pageObj->pageData("select id,studentno ,studentname, classname, tel ,email from tb_student order by id ", $connID, 5, $_GET['page']); //获取包含分页信息的数组
$arrayPageData = $pageInfo['data']; //分页数组下标为 data 项的元素为学生信息内容
for ($i = 0; $i < count($arrayPageData); $i ++){ //通过 for 循环显示学生信息
?>
<!--显示学生信息 -->
<?php
}
?>

```

(6) 显示分页链接，学生信息显示页面根据该链接参数 page 的值决定显示第几个页面。

(7) 关闭数据库的连接，并调用 `StsticPage` 类的 `makeStaicPage()`方法生成静态页，最后调用 `StsticPage` 类的 `pageEnd()`方法指定生成静态页面的结束位置。

```
<?php  
$connObj->closeConnID(); //关闭与数据库的连接  
$stticPage->makeStaicPage($_SERVER['REQUEST_URI'], 'shtml', './pages'); //生成静态页面  
$stticPage->pageEnd(); //指定生成静态页面的结束位置  
?>
```

运行上述实例，当第一次访问学生信息页面时所浏览的是 PHP 动态页面的内容，如果返回页面继续浏览将访问该 PHP 页面所对应的静态页，分别如图 20.6 和图 20.7 所示。

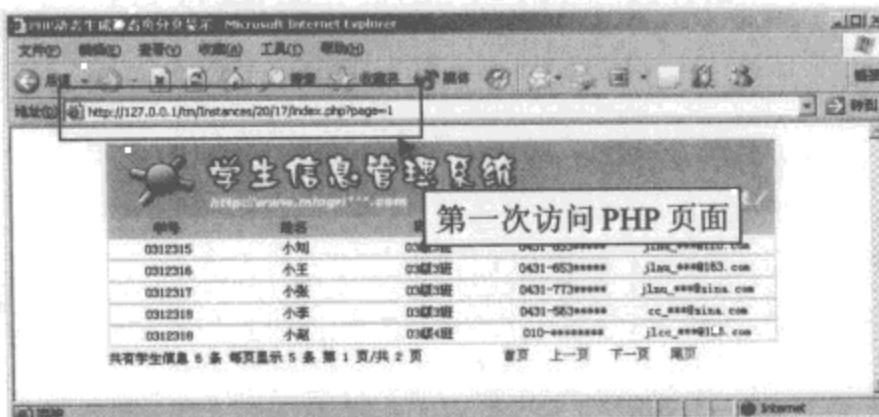


图 20.6 第一次访问 PHP 页面

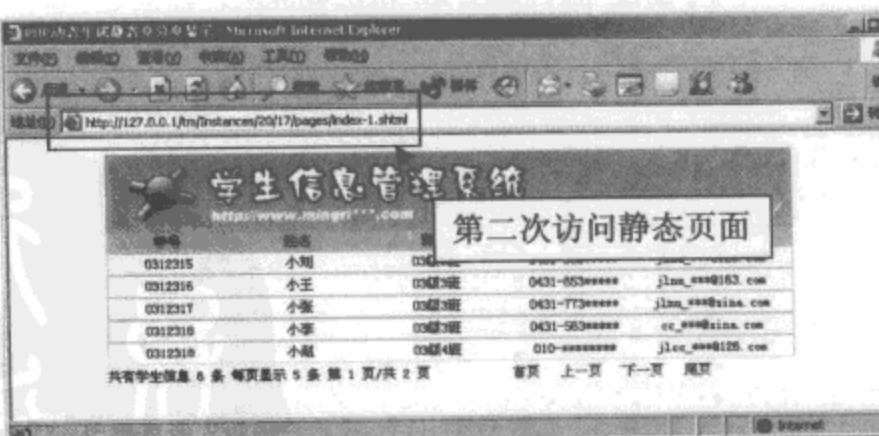


图 20.7 第二次及以后访问 PHP 页面

本书由人民邮电出版社授权新浪博客发布，未经许可，不得以任何形式进行转载或抄袭。

## 20.6 本章小结

本章介绍了 ADODB 类库的概念、支持的数据库和下载安装的方法，对类库中的常用函数作了详细的讲解，并且通过实例来帮助读者加深理解。相信读者通过本章的学习，能够熟练地应用 ADODB 类库操作各种数据库。

## 20.7 实战练习

回答下列问题，将答案写在书本背面的“学习心得”栏中。

1. 使用 ADODB 类库读取 Access 数据库中的数据。（答案位置：光盘\TM\Instances\20\18）
2. 使用 ADODB 类库同时连接多个数据库，并将 Access 数据库表中的数据存储到 MySQL 数据库中。（答案位置：光盘\TM\Instances\20\19）

# 第21章

## 程序调试与异常处理

( 视频讲解: 70分钟)

本章将介绍 PHP 和 MySQL 的常见错误类型、调试的基本方法以及如何处理 MySQL 中的中文乱码问题。

即使是具有多年开发经验的程序开发人员，在开发项目时也难免会发生代码编写错误，发生错误其实并不可怕，只要认真分析、努力思考就一定能够将问题解决。调试既是一种技术，也是一种艺术。随着经验的不断积累，我们就会更熟练地发现并且纠正错误，这其中的关键就是勇敢地去面对错误，不要气馁。本章将介绍一些调试 PHP 和 MySQL 代码的实用技巧，本章的学习定能使初学者受益匪浅、能力有所提高。

通过阅读本章，可以学习到：

### ■ 程序调试的基本流程

### ■ PHP 中常见错误类型

### ■ PHP 中常见语法错误

### ■ PHP 的基本调试策略

### ■ 如何对 MySQL 数据库进行 UTF-8 编码

### ■ 如何处理 MySQL 数据库中文的乱码问题

更多资源请访问我的新浪博客(<http://blog.sina.com.cn/ckook>)

## 21.1 程序基本调试流程

即使是一名具有多年开发经验的编程人员，在编写代码时也不可能一次通过，往往要经过多次调试处理才能实现某一功能。具备基本的调试能力和策略，不仅可以拓展开发人员的编程思路，更能有效缩短项目的整体开发周期，是开发人员所必备的素质。在实际应用中，程序的基本调试流程如图 21.1 所示。



图 21.1 程序基本调试流程

从上述调试流程中可以判断，错误调试的过程是一个往复的过程，也就是说，在对程序中的错误进行调试时，可能要进行多次修改才能成功。在实际开发中，开发人员可以根据自己的经验选择适合自己的错误调试方法。下面具体介绍错误调试的一般流程。

(1) 根据错误提示判断错误原因：程序错误一般分为开发时错误和运行时错误。开发时错误一般是由于开发人员的关键字书写错误造成的，现在很多开发工具都支持开发时错误提示，例如，在开发 PHP 程序时，可以使用 Zend Studio For Eclipse 这个开发工具捕获开发时错误。运行时错误是由于开发人员代码逻辑编写错误等原因造成的，一般可由编译工具在代码编译阶段提示错误，开发人员根据错误提示就可以定位到错误代码的大概位置。

(2) 分析出错原因：定位到出错代码的大概位置后，根据错误提示分析错误原因，如关键字拼写错误、编码逻辑错误等。

(3) 查找错误根源：分析出错的基本原因后，还需要进一步分析错误根源，不要只将错误改正就认为可以了，读者在学习编写程序时要有知其然、并知其所以然的精神，这样才能不断积累提高。

(4) 备份代码：找到出错原因后，不应该立即对代码进行更改，首先需要备份出错代码，防止在更改过程中造成更大错误。

(5) 修改可疑代码：完成代码备份后，开发人员可以对错误代码进行修改，之后再运行调试，也就是说，如果代码修改后再一次发生错误，那么再回到调试流程的第一步。

上面介绍的程序调试流程是最基本的调试过程，读者不必循规蹈矩，可以在开发中根据实际情况总结出适合自己的调试方法和流程。

## 21.2 PHP 中的错误类型

在实际项目开发中，可能会遇到各种错误，不过只要开发人员在平时学习和工作中多积累、多总结，就可以在开发过程中游刃有余，轻松地解决程序中的问题。在项目开发中常见的错误类型包括语法错误、语义错误、逻辑错误和注释错误等。

单的简单操作能让你快速掌握：通过观察代码的运行结果，理解如何解决常见的错误。

## 21.2.1 语法错误

### 视频讲解：光盘\TM\Video\21\语法错误.exe

每一种计算机语言都有自身的语法规则，开发人员必须按照这种语法规则书写代码，否则会发生语法错误。PHP 语言自身有其独特的语法规则，如变量名前要加“\$”、每行要以“;”结束、关键字大小写敏感等。不过，PHP 自身是一种弱声明语言，也就是说 PHP 中的变量不需声明即可使用，这就为开发人员带来了很大的方便。PHP 中常见的语法错误有如下几种形式。

#### 1. 缺少结束符引起的错误

编写 PHP 代码时，要求每一行以“;”结束，如果代码编写人员因疏忽未写结束符“;”，在运行或调试程序时就会发生错误。

**例 21.1 使用 for 循环输出 1~10 之间的数字，并设置缺少结束符错误。（实例位置：光盘\TM\Instances\21\1）**

```
<?php
for($i=1; $i<11; $i++)
{
    echo $i           //没写结束符
}
?>
```

运行上述代码，将在页面中输出如图 21.2 所示的错误提示。

分析页面中的错误，从语意中可以判断程序的第 5 行应该以“;”结束，而不应该以“}”结束，也就是说，程序将“}”当作第 5 行的结束符，而非 for 循环体的结束标志。

#### 2. 缺少单引号或双引号引起的语法错误

和其他语言不同，PHP 中无论使用单引号还是双引号，所引部分都当作字符串常量，区别是使用单引号效率较双引号高，而双引号字符串中可以包含变量并能进行区分。在编写代码时，开发人员可能由于书写错误，使用单引号或者双引号时少书写一个，或字符串两侧使用的引号不一致而导致语法错误。

**例 21.2 使用 echo 语句输出一个字符串，并得到引号不一致错误。（实例位置：光盘\TM\Instances\21\2）**

```
<?php
echo "Hello PHP";      //引号不一致
?>
```

运行上述实例，将在页面中输出如图 21.3 所示的错误提示。

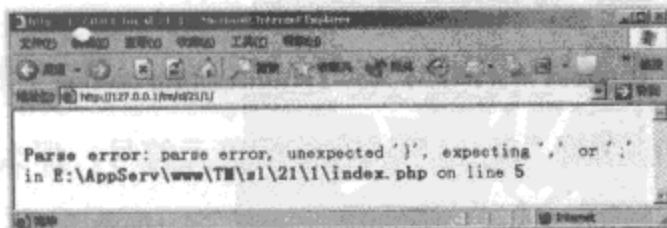


图 21.2 缺少结束符错误

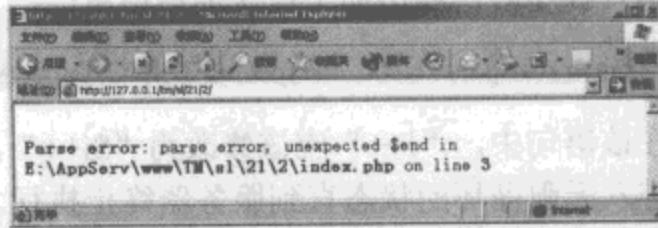


图 21.3 引号不一致错误

从错误提示中可以判断，导致错误的原因是没有结束标志，这说明在程序编译时将字符串后的单引号当作字符串的一部分，而非字符串的结束标志。

### 3. 缺少括号引起的语法错误

与 C/C++ 语法类似，PHP 中诸如 for 循环、while 循环以及包含多条语句的 if 代码块都需要使用大括号。如果代码行数较多，很可能造成大括号遗漏。

**例 21.3 使用双层 for 循环输出两个数的乘积，并得到内层循环缺少结束大括号错误。（实例位置：光盘\TM\Instances\21\3）**

```
<?php
for($i=0; $i<10; $i++)
{
    for($j=0; $j<10; $j++)
    {
        echo $i*$j."<br>"; //遗漏循环体结束大括号
    }
}
?>
```

运行上述实例，将在页面中输出如图 21.4 所示的错误提示，从错误提示中可以得知内层嵌套无结束标志，这就是未写内层 for 循环的结束大括号所导致的。

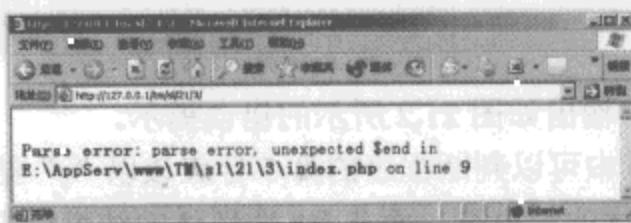


图 21.4 缺少 for 循环结束大括号错误提示

### 4. 缺少变量标识符\$引起的语法错误

在 PHP 中，设置变量时需要使用美元符号 “\$”，如果不添加美元符号就会引起解析错误。

**例 21.4 下面通过实例讲解在使用变量时不添加美元符号会产生什么错误。（实例位置：光盘\TM\Instances\21\4）**

```
<?php
for ( $i=1 ; $i<11; i++) { //缺少一个变量的美元符号
    echo $i."<br>";
}
?>
```

运行程序时，输出如下错误提示信息：

```
Parse error: parse error, unexpected T_INC, expecting ')' in F:\AppServ\www\TM\SL\21\4\index.php on line 9
```

在该语句中，“i++”应该修改为“\$i++”，如果是前面的“\$i<11”没有使用美元符号，那么该程序将进入无限循环的状态直到服务器终止执行。

语法的错误是最基本的错误，只要在编写程序时认真一些，就会减少很多麻烦。这里介绍的只是语

法错误中常见的几种类型，还有很多类似的错误，要避免错误的出现就要靠广大程序设计人员平时编写代码时多加注意，尽量书写准确的代码。建议读者阅读 Zend 官方提供的关于 PHP 编码规范的相关文章。

### 21.2.2 语义错误

语义错误是在语法正确的前提下导致的错误。例如，应用 PHP 连接符实现两个字符串的连接，代码如下。

```
<?php
    $str="PHP 编程词典";
    $url="www.mrbcccd.com";
    echo $str + $url; //错误地使用了字符串连接符号，正确的连接符是“.”
?
>
```

由于 PHP 中的字符串连接符是“.”，而不是“+”。上面的代码错误地使用了“+”作为字符串的连接符。但是由于 PHP 能够隐式转换变量类型，上面的代码并不会导致编译器出错，只是不会输出正确的结果。

### 21.2.3 逻辑错误

逻辑错误对于 PHP 编译器来说并不算错误，但是由于代码中存在的逻辑问题，导致运行结果没有得到期望的结果。逻辑错误在语法上是不存在错误的，但从程序的功能上看是缺陷，它是最难调试和发现的，因为它们不会抛出任何错误信息，唯一能看到的就是程序的功能（或部分功能）没有实现。

例如，某商城实现商品优惠活动，如果用户为普通用户，那么商品不打折；如果是商城的会员，那么商品打八五折，代码如下。

```
<?php
if($user="普通用户"){
    echo $price=485*1; //485 是商品价格，1 是指不打折
}
if($user="会员"){
    echo $price=485*8.5; //485 是商品价格，8.5 是指打八五折
?
>
```

上面的代码对于 PHP 编译器来说没有任何问题。运行程序时，程序没有弹出错误信息。但是当用户为商城的会员时，商品价格乘以一个 8.5，这一点就没有符合要求，属于逻辑错误。应该乘以 0.85 才正确。

 注意：在实现动态的 Web 编程时，通常情况下，数据表中均是以 8.5 进行存储，这时在程序中就应该再除以 10，这样，就相当于原来的商品价格乘以 0.85。正确的代码为：

```
echo $price=485*8.5/10; //485 是商品价格，8.5 是指打八五折
```

对于逻辑错误而言，发现错误是容易的，但要查找出逻辑错误的原因却很困难。因此，在编写程

序的过程中，一定要注意使用语句或者函数的书写完整性，否则将导致程序出错。

#### 21.2.4 注释错误

培养编写注释的习惯，对于一个程序员来说是很有帮助的。它可以增强程序的可读性，便于对程序的修改和后期维护。虽然错误的注释并不影响程序的运行，但是会给维护人员在后期的维护上带来一定的难度。例如：

```
<?php
    $backTime=date("Y-m-d", (time() + 3600 * 24 * 30)); //格式化$backTime 变量为系统当前日期
?>
```

上面获取时间的代码与后面的注释不符，注释应改为“\$backTime 为当前日期+30 天期限”。

#### 21.2.5 运行错误

运行错误的原因不容易确定，它可能是由脚本导致的，也可能是在脚本的交互过程中或其他的事件、条件下产生的，这就需要程序员平时多积累遇错处理的方法，以提高解决问题和分析问题的能力。下面的几种情况是常见的运行错误：

- 调用不存在的文件。在编写程序时，由于调用文件的名称书写错误，导致调用了一个不存在的文件。
- 调用不存在的函数。在编写程序时，如果函数名称输写错误，此时就会产生错误。即使函数名写对了，但使用的参数不对，同样也会产生一个错误。
- 读写文件。访问文件的错误也是经常出现的，如硬盘驱动器出错或写满，人为操作错误导致目录权限改变等。如果没有考虑到文件的权限问题，文件权限设置为只读属性，直接对文件进行操作就会产生错误。由于该文件具有只读的权限，不能进行写入的操作。在执行这项操作的时候，首先要明确该文件的属性是否为可写。如果要坚持执行操作，那就需要修改文件的权限。
- 运算的错误。在进行一些算术运算或者逻辑运算的过程中，如果出现不符合运算法则的运算，例如，在做除法运算时，分母为 0，就会产生错误。

### 21.3 PHP 错误的调试

PHP 最基本的调试策略是使用 PHP 的错误报告机制。之所以说这是最基本的调试策略，是因为对 PHP 文件的设置是在安装时就必须配置好的，否则就无法使用。

#### 21.3.1 PHP 的错误报告

 视频讲解：光盘\TM\Video\21\PHP 的错误报告.exe

通常在一个 PHP 脚本中发生错误时，出错信息会被插入到当前脚本进行输出，如果是致命的错误，

脚本就会终止执行。

PHP 将脚本错误分成提示 (notice)、警告 (warning) 和错误 (error) 3 个等级。

- notice 信息可能是脚本运行错误导致的，也可能只是在正常运行过程中出现的。实际上这也许是代码中的缺陷，因为 PHP 对其解释可能与代码的本义有所不同。
- warning 标识着一个非致命性的错误，在代码运行时产生。它们并不是致命的错误，不会停止脚本的执行。
- error 消息说明出现了致命的错误，会导致脚本停止运行。PHP 运行的任何阶段都可能产生这种错误，包括初始化、解析和执行代码阶段。

### 21.3.2 启动错误报告

#### 视频讲解：光盘\TM\Video\21\启动错误报告.exe

使用 PHP 的错误报告是调试程序的基础。通过对 PHP 配置文件的设置，使程序在执行过程中自动抛出错误的代码行，帮助开发者发现并消除错误。当项目发布后，需要关闭错误报告，用户不能看到错误信息，这样非常不人性化。

启动错误报告，也就是修改 php.ini 文件中的报错设置，具体设置如图 21.5 所示。

display\_errors 变量的目的很明显，用于告诉 PHP 是否显示错误。默认值是 Off。但是，要让开发过程更加轻松，需要把这个值设为 On，以便根据错误提示调试程序。

error\_reporting 变量的默认值是 E\_ALL。这个设置会显示从不良编码实践到无害提示的所有信息。E\_ALL 值的设置对于开发过程来讲过于细化。通常只需要设置错误提示和显示不良编码即可，因此，需要把这个值设置为“E\_ALL & ~E\_NOTICE”。

设置完成后，保存 php.ini 文件，然后重新启动 Apache 服务器即可。

通过这些错误设置，开发者在执行程序的过程中，就可以在浏览器中查看到错误信息，以及出错代码的行号，并分析产生错误的原因。

### 21.3.3 使用 print 语句调试程序

#### 视频讲解：光盘\TM\Video\21\使用 print 语句调试程序.exe

print 语句用于字符串的输出。基于 print 语句灵活的特点，在程序调试过程中经常被广泛使用。

**例 21.5** 通过实例介绍如何应用 print 语句对程序进行调试。在本实例中实现一个获取表单提交的数据功能，程序代码如下：（实例位置：光盘\TM\Instances\21\5）

```
<form name="form1" method="post" action="">
    <input name="txt_key" type="text" id="txt_key">
    <input type="submit" name="Submit" value="搜索">
</form>
<?php
if($_POST[Submit]=="搜索"){


```



图 21.5 设置 php.ini 文件中的错误处理机制

```

    $key=$_POST[txt_key];           //获取文本框的值
}
?>

```

上面的代码实现了获取文本框的值，由于语句中没有输出，因此当提交表单时也看不到是否获取了信息，这时可以应用 print 语句输出文本框的值，以此来判断是否正确获取文本框的值。在加粗的代码下添加如下语句。

```

print $key;           //输出文本框的值

```

print 语句输出了 POST 数组中的数据，即文本框的值，如果数组中的数据未能正确显示，则说明没有正确获取到文本框的值，从而断定代码中有错误，需要查找错误源头，并加以修正。如果 print 语句输出的字符串与输入到文本框中的字符串相符，则说明程序正确。

运行结果如图 21.6 所示。

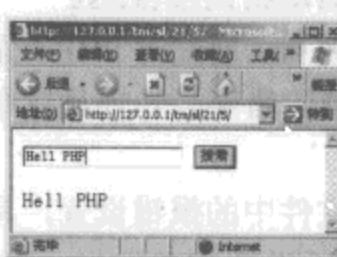


图 21.6 使用 print 语句判断是否正确获取到文本框的值

使用 print 语句可以缩小缺陷可能出现的范围，能够最终确定错误的位置或验证数据信息。在程序中熟练地使用 print 语句，可以迅速找出程序中的错误根源。

#### 21.3.4 应用@前缀字符屏蔽 PHP 脚本错误提示

**视频讲解：**光盘\TM\Video\21\应用@前缀字符屏蔽 PHP 脚本错误提示.exe

前缀字符@不是一个命令，而是 DOS 批处理的一个特殊标记符，是命令行回显屏蔽符。

前缀字符@表示执行时本行在 cmd 中不显示，可以使用 echo off 命令关闭显示，仅用于屏蔽命令行回显。在默认情况下，当 PHP 代码遇到错误时会显示错误信息，包括当前脚本的绝对路径，这样会造成一些不安全的因素，可以在调用函数时加上前缀字符@，以屏蔽错误信息。

**例 21.6** 当试图打开一个不存在的文件时，屏蔽错误信息以避免路径泄露，代码如下：（实例位置：光盘\TM\Instances\21\6）

```

<?php
$file="book.php";           //定义操作文件
@fread($file) or die("文件读取失败！");
fclose($file);
echo "我不能被输出了！程序运行后，前缀字符@后面的命令不被显示！";
?>

```

fread() 函数前应用了前缀字符@，说明程序运行时在浏览器上不会显示前缀字符@后面的命令。  
结果为：文件读取失败！

### 21.3.5 使用错误处理器记录日志

PHP 提供了内建的错误处理器函数 `error_log()`，可以将出错信息记录到管理员所指定的路径。

```
error_log(message,type [, destination, [,extra_headers]]);
```

参数 `message` 指出错信息。参数 `type` 指定出错信息记录的指定位置。如果用 PHP 的日志记录机制保存出错信息，需要将参数 `type` 的值设置为 0。如果将错误追加到 `destination` 文件中，需要将参数的值设置为 3。另外，需要修改 `php.ini` 中的 `error_log` 选项，具体设置如图 21.7 所示。

设置完成后，保存 `php.ini` 文件，然后重新启动 Apache 服务器即可。

例 21.7 实现一个错误处理的例子，将日志记录到日志文件中，并在文件大于 1024B（1KB）时对日志文件进行重命名，代码如下：（实例位置：光盘\TM\Instances\21\7）

```
<?php
function err_log($error,$error_str){ //自定义一个错误处理函数
    $file="php_error.log";
    if(filesize($file)>1024){
        rename($file,$file.(string)time());
        clearstatcache();
    }
    error_log($error_str,0,$file); //将出错信息记录到管理员所指定的路径
}
set_error_handler('err_log'); //执行自定义函数 log_roller()函数
trigger_error(time()."程序报错.\n"); //发出错误信息
restore_error_handler(); //重新编译这个预错处理的函数
?>
```

在上面的代码中，加粗的 PHP 内置错误处理函数 `error_log()` 的 `type` 参数值设置为 0，将出错信息记录在指定的文件 `system logger` 中，每运行一次程序或刷新一次，该文件将即时更新，将每次的出错信息加载到该文件中。

 技巧：如果将参数 `type` 的值设置为 3，则将错误追加到 `destination` 文件中，每运行一次或刷新一次程序，都会生成一个新的错误日志。

通常，内存的地址是以字节（byte）为单位编制的，内存的容量一般以 KB 或 MB 为单位。

一般情况下，在开发一个网站时，为了调试方便，希望错误直接显示在页面上。但当一个网站正式发布到互联网上时，不能把内部的错误信息显示给访问者，最佳的方法是记录到错误日志中。这时，需要在 `php.ini` 文件中做如下设置。

```
display_errors = Off
log_errors=on
error_log=/tmp/errors.log
```



图 21.7 设置 `php.ini` 文件中的错误处理机制

设置完成后，保存 php.ini 文件，然后重新启动 Apache 服务器即可。

## 21.4 SQL 错误的调试

SQL 语句错误也是相当常见的，它们是由 MySQL 数据库报告的。SQL 错误主要体现在两个方面：第一是它和 PHP 的连接方面；第二是对其自身的使用上，也就是在执行 MySQL 语句时的错误。

### 21.4.1 PHP 与 MySQL 连接错误

#### ▶ 视频讲解：光盘\TM\Video\21\PHP 与 MySQL 连接错误.exe

PHP 与 MySQL 数据库的连接是进行程序开发的第一步，如果不能成功地把它们连接起来，工作就无法进行。这就使它们连接中的错误成为首先要解决的问题，也是必须要解决的问题。

在通过 PHP 函数连接 MySQL 数据库时，有可能出现的错误包括：

- 使用的 PHP 函数不正确，不过这种情况不是很常见，因为如果是函数书写错误，那么在具备高亮代码显示功能的开发工具中能够直接看出函数书写是否正确。
- 使用函数没有问题，但输入的参数不正确。输入了错误的数据库用户名、密码或者指定的数据库不存在，同样会导致数据库连接失败。

**例 21.8 通过 PHP 函数连接 db\_database21 数据库，数据库用户名为 root，密码为 root，代码如下：**  
**(实例位置：光盘\TM\Instances\21\8)**

```
<?php
$con=mysql_connect('localhost','root','111') or die("与服务器连接失败!");
echo "与服务器连接成功!<br>";
mysql_select_db('db_database21',$con) or die("没有找到数据库!");
echo "成功!";
?>
```

本实例在编写代码的过程中，使用的密码不是 root，而是 111，所以导致数据库连接失败，其运行结果如图 21.8 所示。

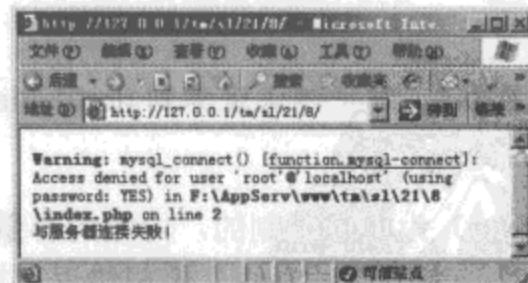


图 21.8 密码不正确导致连接失败

**▶ 注意：**在上述的连接数据库的实例中，应用了一个 die() 语句，它是 PHP 的一种错误处理机制。当它左侧的语句发生错误时，die() 语句会显示其字符串参数的内容，而且程序会退出。

### 21.4.2 SQL 语句错误

#### 视频讲解：光盘\TM\Video\21\SQL 语句错误.exe

编写的 PHP 语句和数据库的连接都没有问题，错误出现在 SQL 语句的执行过程中，出现这种错误的原因有以下 3 个。

- 数据提交页面与数据处理页面之间，form 表单中定义的字段的名称与数据处理页中获取字段值所使用的名称不一致，导致获取不到数据，如图 21.9 所示。

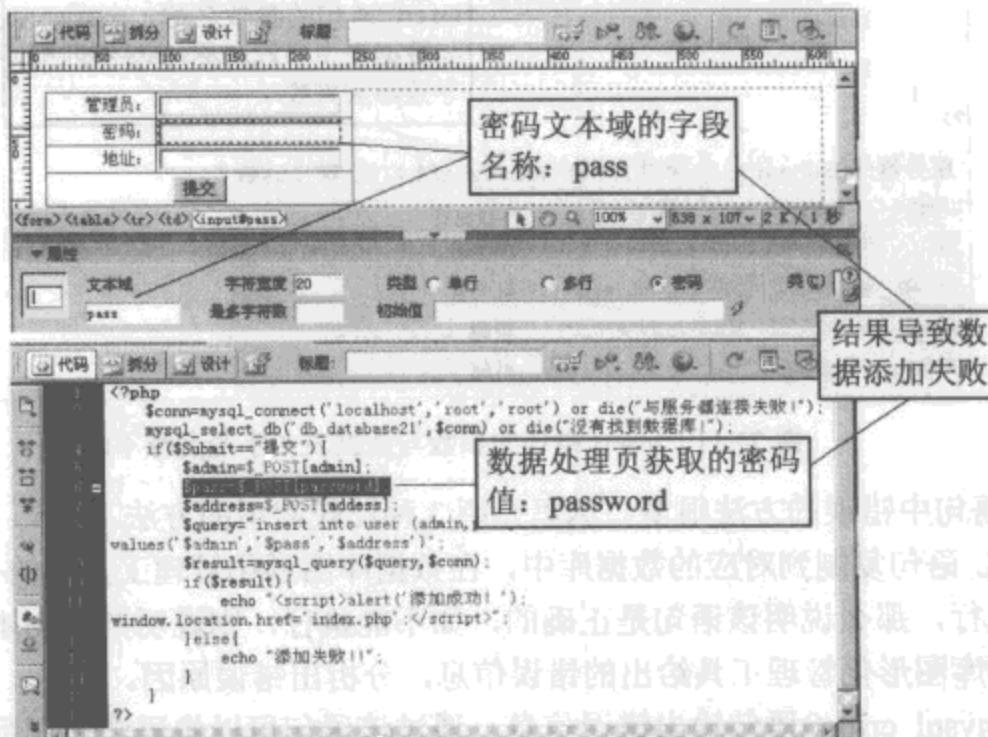


图 21.9 数据提交页与数据处理页使用的字段不统一

- 数据处理页中执行的 insert 语句，参数的数量与参数值的数量不匹配，或者参数与参数值的顺序不匹配，都将导致添加失败，如图 21.10 所示。

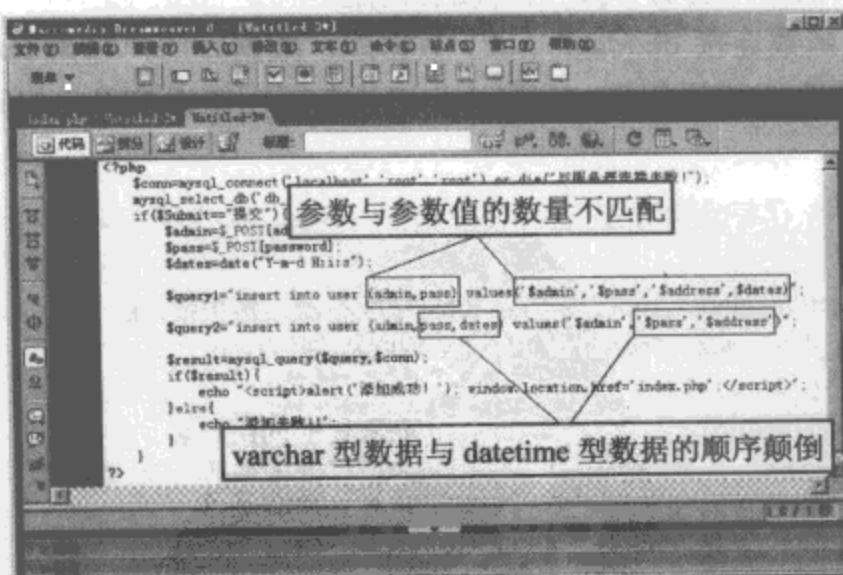


图 21.10 数据处理页中出现的错误

- 数据处理页 insert 语句的参数与数据表中字段不匹配，使得字段名称不一致、字段数量不匹配或者数据类型不一致，都将导致添加失败，如图 21.11 所示。

```

<?php
$conn=mysql_connect('localhost','root','root') or die("与服务器连接失败!");
mysql_select_db('db_database21',$conn) or die("没有找到数据库!");
if($Submit=="提交"){
    $admin=$_POST[admin];
    $pass=$_POST[password];
    $dates=date("Y-n-d H:i:s");

    $query="insert into user (admin,password,dates) values('$admin','$pass','$address')";
    $result=mysql_query($query,$conn);
    if($result){
        echo "<script>alert('添加成功!');</script>";
    }else{
        echo "添加失败!!";
    }
}
?>

```

字段	类型	备注	字段说明
id	int(10)		ID主键
admin	varchar(50)	gb2312_chinese_ci	用户名
pass	varchar(50)	gb2312_chinese_ci	密码
dates	datetime		时间

图 21.11 insert 语句的参数与数据表中字段名称不匹配

解决 SQL 语句中错误的方法很多，这里介绍 3 种比较常用的方法。

- 将 SQL 语句复制到对应的数据库中，在数据库图形化管理工具中的 SQL 语言下执行，如果可以执行，那么说明该语句是正确的，如不能执行，则说明是语句本身存在错误，就可以根据数据库图形化管理工具给出的错误信息，分析出错误原因。
- 使用 mysql\_error()语句输出错误信息，通过该语句可以像语法错误那样返回一个错误信息。该语句的使用被放置于 mysql\_query()函数的后面。如果将 die()语句与 mysql\_error()语句组合应用，当程序结束时就会显示 MySQL 的错误信息。
- 将 SQL 语句定义成一个单独的变量，应用 echo()语句输出这个变量，分析这个变量输出的值是否正确，如果正确则说明 SQL 语句没有问题。

例 21.9 通过 PHP 函数连接 db\_database21 数据库，数据库用户名为 root，密码为 root，向 user 表中添加一条记录，并通过 die()语句与 mysql\_error()语句返回错误信息，代码如下：（实例位置：光盘\TM\Instances\21\9）

```

<!--处理文件代码-->
<?php
$conn=mysql_connect('localhost','root','root') or die("与服务器连接失败!");
mysql_select_db('db_database21',$conn) or die("没有找到数据库!");
if($Submit=="提交"){
    $admin=$_POST[admin]; //获取提交的用户名
    $pass=$_POST[pass]; //获取提交的密码
    $dates=date("Y-m-d H:i:s"); //定义当前时间
    $query="insert into user (admin,password,dates) values('$admin','$pass','$dates')"; //编写添加语句
    $result=mysql_query($query,$conn) or die(mysql_error()); //执行添加语句，并返回错误信息
    if($result){
}
}
?>

```

```

echo "<script>alert('添加成功！'); window.location.href='index.php';</script>";
}else{
    echo "添加失败!!!";
}
}

?>
<!--表单提交页，设置的表单元素-->
<form name="form1" method="post" action="index.php">
    <input name="admin" type="text" id="admin" size="20">
    <input name="pass" type="password" id="pass" size="20">
    <input type="submit" name="Submit" value="提交">
</form>

```

本实例中错误的原因是 insert 语句的参数与数据表中的字段名称不匹配，在数据表中没有找到指定的字段，其运行结果如图 21.12 所示。

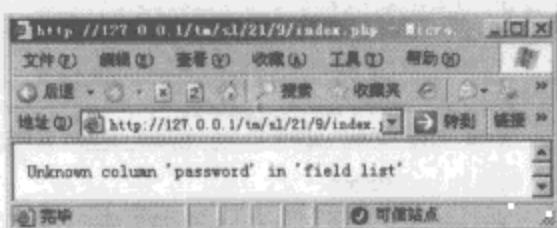


图 21.12 SQL 语句中出现的错误

说明：上面介绍的只是 SQL 语句中的一种错误情况，至于在 select、update 和 delete 等语句的执行过程中的错误需要读者自己去研究与分析。

## 21.5 实 战

### ■ 视频讲解：光盘\TM\Video\21\解决 MySQL 数据库乱码问题.exe

PHP 访问 MySQL 数据库时，有时会出现乱码，这主要是数据库字符集选择不当所造成的。开发中文程序，MySQL 数据库一般选择 UTF-8 编码、GB2312 编码或 GBK 编码，在开发项目时一般采用 UTF-8 编码，因其可识别的字符与 GB2312 和 GBK 相比更多些。

**例 21.10 使用 MySQL 数据库管理工具 phpMyAdmin 创建数据库 db\_test01，并设置该数据库所使用的字符集为 UTF-8 编码，然后创建 tb\_info 数据表并通过 PHP 页面显示 tb\_info 表中的信息。（实例位置：光盘\TM\Instances\21\10）**

本实例的实现步骤如下：

(1) 打开 phpMyAdmin 主界面，在“创建一个新的数据库”选项组中填写新数据库名为 db\_test01，并选择字符集为 utf8\_unicode\_ci，如图 21.13 所示，然后单击“创建”按钮。

(2) 在新页面的“在数据库 db\_test01 中创建一个新表”选项组中输入表名 tb\_info，并设置该表有两个字段，如图 21.14 所示，然后单击“执行”按钮。

(3) 保存 tb\_info 表的主键 id 和普通字段 title 的设置信息后 tb\_info 表即被创建成功，之后向 tb\_info 表中插入一条记录，如图 21.15 所示。

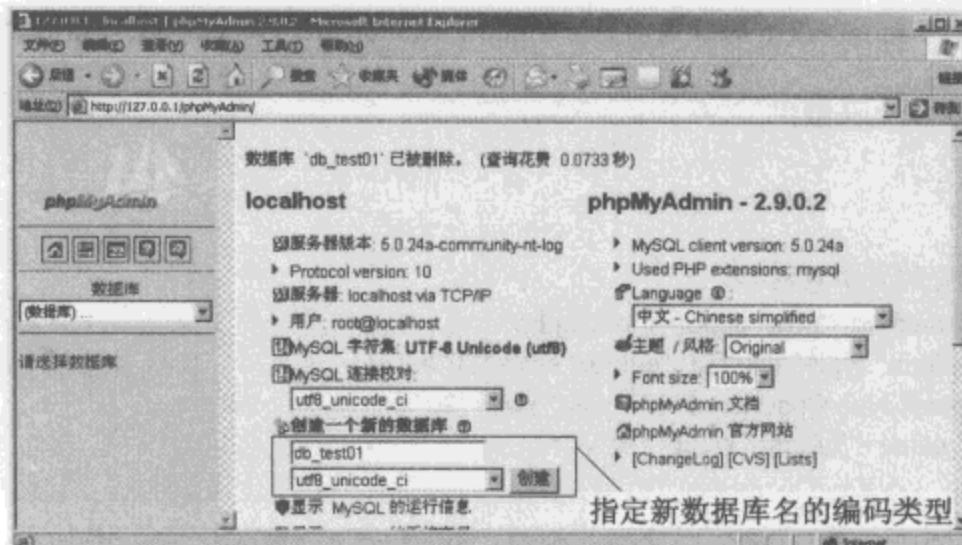


图 21.13 创建新数据库

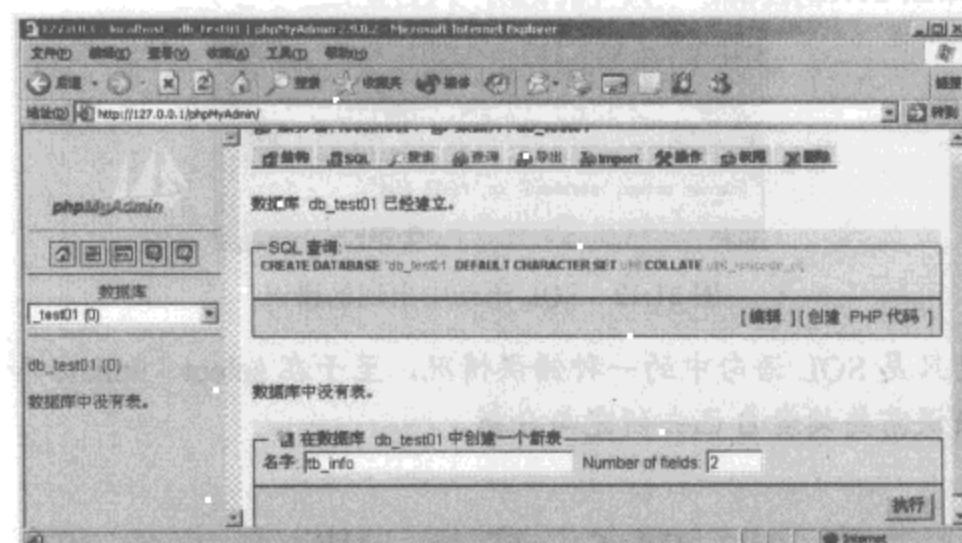


图 21.14 创建数据表

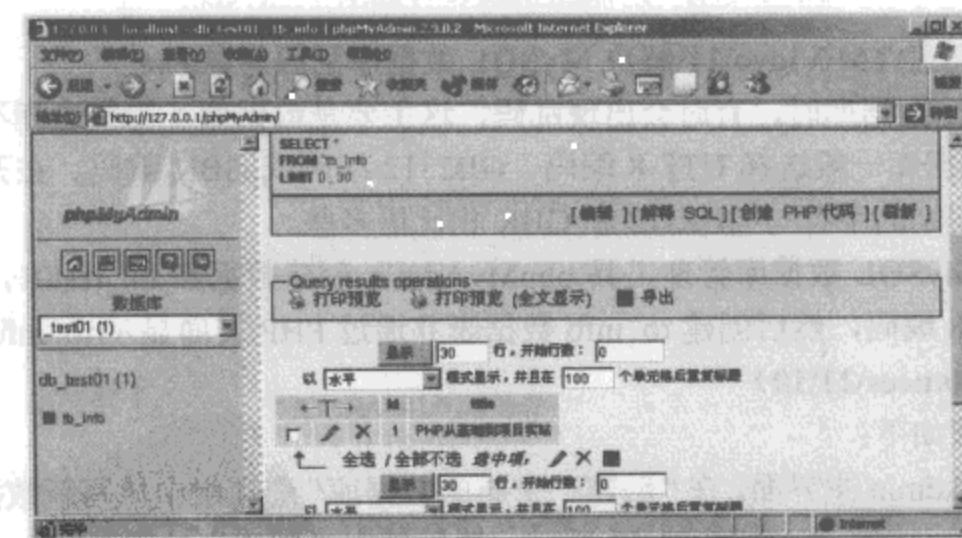


图 21.15 向 tb\_info 表中插入一条记录

(4) 建立 index.php 文件, 该文件中的代码用于建立与 db\_test01 数据库连接并输出 tb\_info 表中的内容, 代码如下。

```

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
<?php
$host = "127.0.0.1";          //MySQL 数据库服务器
$userName = "root";           //用户名
$password = "root";           //密码
$dbName = "db_test01";        //数据库名
$connID = mysql_connect($host, $userName, $password) or die("数据库连接失败！"); //连接数据库
mysql_select_db($dbName, $connID);      //选择数据库
$sql = mysql_query('select * from tb_info'); //执行查询
while($result = mysql_fetch_array($sql))   //获取结果集，并输出 title 字段的内容
{
    echo $result['title'];
}
?>
</body>
</html>

```

运行上述代码，将在页面中输出如图 21.16 所示的内容，该内容为 tb\_info 表的乱码形式，为了解决乱码，在程序中加入如下内容：

```
mysql_query("set names utf8"); //设置字符集
```

刷新页面，效果如图 21.17 所示，在刷新后的页面中正确显示了数据表 tb\_info 中的信息，说明对 MySQL 数据库进行 UTF-8 编码调试成功。

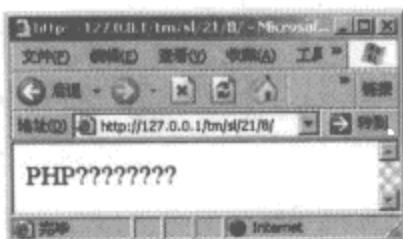


图 21.16 汉字乱码

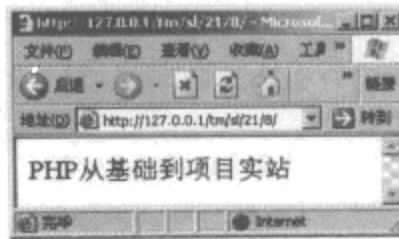


图 21.17 正确显示了 tb\_info 表中的内容

通过本实例的学习不仅可以了解 MySQL 如何实现 UTF-8 编码，而且可以学到如何对 MySQL 数据库信息编码进行调试。如果 MySQL 数据库的编码类型为 GBK 或 GB2312，则可以使用如下语句解决数据库乱码问题：

```
mysql_query('set names gbk'); //设置字符集为 GBK
mysql_query('set names gb2312'); //设置字符集为 GB2312
```

## 21.6 本章小结

本章主要介绍了程序调试及异常处理的一般步骤，并着重讲解了 PHP 中常见的错误及其原因。通

过本章的学习，初学者可以掌握 PHP 中常见错误的处理方式，并能够积累基本的程序调试经验与技巧。希望读者朋友能够以本章介绍的内容为基石，进一步拓展与积累，最终登上编程工程师的舞台。

## 21.7 实战练习

1. 现在很多知名的 PHP 空间提供商还延续使用 PHP 4.x，完全使用 PHP 5.x 开发的程序可能无法在这样的服务器中运行，为了解决上述问题，要求使用 `phpversion()` 函数判断服务器中所安装的 PHP 版本，然后根据该版本号由程序动态选择要使用的 PHP 代码块。（答案位置：光盘\TM\Instances\21\11）
2. 租用服务器发布网站时，一般无法改变 `php.ini` 文件的相关设置，但在某些情况下如果不对该文件的参数进行设置就无法完全发挥网站的功能，如使用 Session 记录用户身份时，有可能在用户还没完成某项操作时，Session 就自动失效。为了解决此问题，要求使用 `ini_set()` 函数在页面中更改 session 的失效时间。（答案位置：光盘\TM\Instances\21\12）

# 第3部分

## 高级应用

- ▶ 第22章 XML技术
- ▶ 第23章 Ajax技术
- ▶ 第24章 Smarty模板技术



# 風魚樂惠

木其 MAX 鮭魚頭 1kg

大骨 XTRA 鮭魚頭 1kg

木其青鱈 Fishfillet 鮭魚片 1kg



# 第22章

基础 JMX + RSS

## XML 技术

( 视频讲解: 79 分钟)

XML 是一种目前趋于流行的半结构化文件格式，以一种类似数据库的格式存储数据，被称为“第二代 Web 语言”，是 Web 2.0 中的一项重要技术。无论是 RSS 订阅、Web Service，还是 Ajax 无刷新，都和 XML 有着直接的联系。

通过 XML 存储数据，不但能够减少与数据库的交互性操作以提高读取效率，而且可以有效利用 XML 的优越性，降低程序的编写难度。PHP 为 XML 提供了一套完整的读取方法，能够很容易地编写基于 XML 的脚本程序。

通过阅读本章，可以学习到：

- XML 基础知识
- 使用 SimpleXML 解析 XML 文档
- 遍历 XML 文档
- 修改、保存 XML 文档
- 动态创建 XML 文档
- 动态读取 XML 文档

PHP 教程  
PDG

## 22.1 XML 概述

XML (eXtensible Markup Language)，扩展性标记语言，是一种类似于 HTML 的标记性语言，主要用于描述数据和存储数据。它允许用户设计自己的标记。XML 是由 W3C (World Wide Web Consortium, 互联网联合组织)于 1998 年 2 月发布的一种标准，它的前身是 SGML (Standard Generalized Markup Language, 标准通用置标语言)。XML 产生的原因是为了解决 HTML 语言的不足，使网络语言更加规范化、多样化。

XML 用于在一个文档中存储数据，但是数据存储并不是其主要目的，其主要目的是通过该格式标准进行数据交换和传递。即将一种格式的数据存储到 XML 文档中，然后对这个 XML 文档进行解析，最后以另一种格式输出数据。

XML 支持 Unicode 编码，可以包含世界各地的任何字符集和二进制数据，并且 XML 不依赖于任何操作系统平台，是真正的跨平台技术。

XML 适用于：

- 结构化数据，如系统配置文件、邮件地址簿。
- 标准数据交换，用于多个平台或应用系统之间的数据传递，如 Web Service、Ajax。
- 应用程序数据的通用，将文字处理程序由原来的二进制格式转换为 XML 格式保存，如微软的 Office 2007 等。
- 创建新的标记语言，用户可以创建新的标记，用于实现更多的功能，如 RSS 和 Atom 等。

## 22.2 XML 语法

XML 语法是 XML 语言的基础，是学好 XML 的前提条件，只有掌握了 XML 语法才能更好地应用 XML 技术。下面就来学习 XML 语法。

### 22.2.1 XML 文档结构

在学习 XML 的语法之前，先通过一个具体的实例来讲解 XML 文档的结构。

**例 22.1** 创建一个 XML 文档，通过这个文档熟悉 XML 的结构。XML 文档的代码如下：（实例位置：光盘\TM\Instances\22\1）

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<?xml-stylesheet type="text/css" href="Book.css"?>
<!-- 下面的标签&lt;计算机图书&gt;就是这个 XML 文档的根目录 --&gt;
&lt;计算机图书&gt;
    &lt;PHP&gt;
        &lt;书名&gt;PHP 开发典型模块大全&lt;/书名&gt;
        &lt;价格 单位="元/本"&gt;79.00&lt;/价格&gt;
        &lt;出版时间&gt;2009-02-01&lt;/出版时间&gt;</pre>

```

```
</PHP>
</计算机图书>
```

例 22.1 包含了一个 XML 文档的最基本的要素，包括 XML 声明、处理指令（PI）、注释、元素等，下面就来一一讲解。

### 22.2.2 XML 声明

XML 声明在文档中只能出现一次，而且必须是在第一行，连空行也不允许存在。XML 声明中包括 XML 版本、编码等信息，例 22.1 中的第一行就是该文档的声明：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
```

XML 声明中的各部分的含义如表 22.1 所示。

表 22.1 XML 声明中的各部分含义

XML 声明部分	含义
<?xml	表示 XML 声明的开始标记。xml 表示该文件是 XML 文件
version="1.0"	XML 的版本说明，是声明中必不可少的属性，而且必须放到第一位
encoding="gb2312"	编码声明。如果声明该属性，那么 XML 默认使用 UTF-8 来解析文档
standalone="yes"	独立声明。如果该属性赋值 yes，那么说明该 XML 文档不依赖于外部文档。如果该属性赋值为 no，则说明该文档有可能依赖于某个外部文档。注意：是“有可能”
?>	XML 声明的结束标记

### 22.2.3 XML 处理指令

顾名思义，就是如何处理 XML 文档的指令。有一些 XML 分析器可能对 XML 文档的应用程序不作处理，这时可以指定应用程序按照这个指令信息来处理，然后再传给下一个应用程序。XML 声明其实就是一个特殊的处理指令。处理指令的格式为：

```
<?处理指令名 处理执行信息?>
```

例 22.1 中的处理指令是：

```
<?xml-stylesheet type = "text/css" href="Book.css"?>
```

具体含义如下。

- xml-stylesheet**: 样式表单处理指令。指明了该 XML 文档所使用的样式表。
- type="text/css"**: 设定文档所使用的样式是 css。
- href="Book.css"**: 设定样式文件的地址。

### 22.2.4 XML 注释

XML 中的注释和 HTML 是相同的，使用“`<!--`”和“`-->`”作为开始和结束界定符。注释的用法十

分简单，这里只介绍在使用注释时需要注意的几个问题。

- 不能出现在 XML 声明之前。
- 不能出现在 XML 元素中间。如“<computer\_book <!-- 这是错误的 -->>”。
- 不能出现在属性列表中。
- 不可嵌套注释。
- 注释内容可以包含“<”、“>”和“&”这些特殊字符，但不允许有“--”。

## 22.2.5 XML 元素

元素是每个 XML 文档不可或缺的部分，也是文档内容的基本单元。每个 XML 文档至少要包含一个元素。一般的元素由 3 部分组成，格式如下：

```
<标签>数据内容</标签>
```

其中，<标签>为元素的开始标签，</标签>是元素的结束标签，中间的数据内容是元素的值。这里要注意如下几点：

### 1. 标签的写法

- <标签>和</标签>都是成对出现的，这是 XML 严格定义的。不允许只有开始标签而没有结束标签，对于空元素，即两个标签之间没有数据，这时可以使用简短形式<标签/>。
- 英文标签名称只能由下划线“\_”或英文字开头，中文标签名称只能使用下划线“\_”或汉字开头。名称中只能有下划线“\_”、连接符“-”、点“.”和冒号“:”几个特殊字符，也可以使用指定字符集下的合法字符。
- <标签>中不能有空格。< 标签>或</ 标签>都是错误的。
- <标签>对英文大小写很敏感，如<name>和<Name>是两个不同的标签。

### 2. 元素的前缀

为指定的元素增加一个前缀，进而保证元素和属性的唯一性，就是所谓的命名空间。其最重要的用途是融汇不同的 XML 文档。命名空间的格式为：

```
<标签名称 xmlns:前缀名称="URL">
```

例 22.2 对元素<PHP 图书>使用命名空间。实例代码如下：（实例位置：光盘\TM\Instances\22\2）

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<PHP 图书 xmlns:frn="http://www.bccd.com/foreign">
    <frn:PHP>
        <frn:书名>PHP 网络编程标准教程</frn:书名>
        <frn:价格 货币种类="RMB" 单位="4 本">59.00</frn:价格>
        <frn:出版时间>2009-02-01</frn:出版时间>
    </frn:PHP>
</PHP 图书>
```

### 3. 特殊元素

在 XML 中对于特殊元素（即特殊字符）“>”、“<”和“&”的输入需要使用实体引用来处理，实体引用就是使用“&…;”的形式来代替那些特殊字符。表 22.2 是 XML 中所用到的实体引用。

表 22.2 XML 中的实体引用

实体参考	字符
&lt;	<
&gt;	>
&apos;	'
&quot;	"
&amp;	&

但如果遇到大量的特殊符号需要输入，使用这种方法就不太实际了。XML 中提供了 CDATA（Character data，字符数据）标记。在 CDATA 标记段的内容都会被当作纯文本数据来看。CDATA 标记的格式如下：

```
<![CDATA[
...
]]>
```

例 22.3 分别使用实体引用和 CDATA 标记来显示特殊符号，实例代码如下：（实例位置：光盘\TM\Instances\22\3）

```
<?xml version="1.0" encoding="GB2312"?>
<exam>
    <实体引用>这里必须使用引用“&lt;”、“&gt;”、“&amp;”</实体引用>
    <CDATA 标记>
    <![CDATA[
        这里可以正常输出“<”、“>”、“&”。
    ]]>
    </CDATA 标记>
</exam>
```

运行结果如图 22.1 所示。

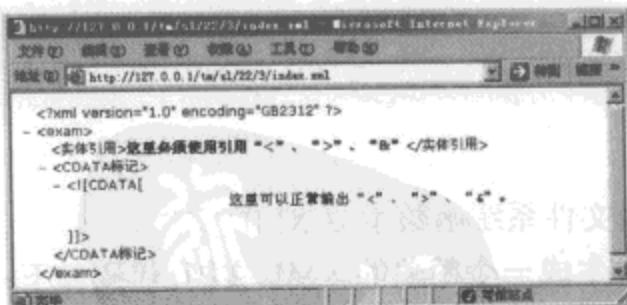


图 22.1 使用实体引用和 CDATA 标记来显示特殊符号

注意：在 CDATA 标记段内不允许出现“]]>”，否则，XML 会认为 CDATA 标记段结束。

## 22.2.6 XML 属性

XML 属性是 XML 元素中的内容，是可选的。XML 属性和 HTML 中的属性在功能上十分相似，但 XML 属性在格式上更加严格，使用上更加灵活。XML 属性的格式为：

```
<标签 属性名="属性值" 属性名=""...>内容</标签>
```

这里要注意：

- 属性名和属性值必须是成对出现的，不像 HTML 中的一些属性，可以不需要值而单独存在。  
对于 XML 来说这是不允许的。如果没有值，写成“属性名=""”也可以。
- 属性值必须用引号括起来。通常使用双引号，除非属性值本身包含了双引号，这时可以用单引号来代替。

## 22.3 创建 XML 文档

### 视频讲解：光盘\TM\Video\22\创建 XML 文档.exe

PHP 可以应用 header() 函数生成 XML 文件，也可以使用文件系统函数来生成，还可以使用 DOM 对象来动态生成 XML 文件。

首先通过实例来讲解如何通过 header() 函数生成 XML 文件。

**例 22.4 通过 header() 函数生成一个简单的 XML 文档，代码如下：(实例位置：光盘\TM\Instances\22\4)**

```
<?php
    header('Content-type:text/xml');
    echo '<?xml version="1.0" encoding="gb2312" ?>';
    echo '<计算机图书>';
    echo '<PHP>';
    echo '<书名>PHP 项目开发全程实录</书名>';
    echo '<价格>85.00RMB</价格>';
    echo '<出版日期>2008-5-5</出版日期>';
    echo '</PHP>';
    echo '</计算机图书>';
?>
```

运行结果如图 22.2 所示。

然后通过实例来看如何利用文件系统函数生成 XML 文件。

**例 22.5 通过文件系统函数生成一个简单的 XML 文档，代码如下：(实例位置：光盘\TM\Instances\22\5)**

```
<a href="index.xml">输出创建的 xml 的文档</a>
<?php
    $fopen = fopen ("index.xml","w"); //使用添加模式打开文件，文件指针指在表尾
```

```

$title='<?xml version="1.0" encoding="gb2312"?>
<rss      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:taxo="http://purl.org/rss/1.0/modules/taxonomy/" version="2.0">
    <channel>
        <title>明日科技</title>
        <link>http://www.mingrisoft.com</link>
        <description>吉林省明日科技有限公司</description>
        <dc:creator>http://www.mingrisoft.com</dc:creator>';
fwrite($fopen,$title);                                //写入头文件
$item="<item>
    <title>编程词典</title>
    <link>http://www.mrbccd.com</link>
    <description>明日科技编程词典重拳出击，隆重上市！</description>
    <pubDate>2008-08-08</pubDate>
    </item>";
if (!fwrite($fopen,$item)){                         //将数据写入到文件中
    echo "写入内容失败！";
    exit ;
}
fwrite($fopen,"</channel></rss>");                  //将 XML 文件的结束符写入到文件中
fclose($fopen);          //关闭文件
?>

```

运行结果如图 22.3 所示。

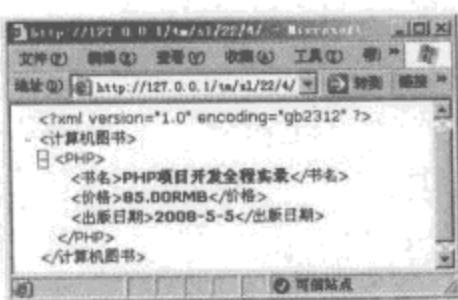


图 22.2 创建 XML 文件

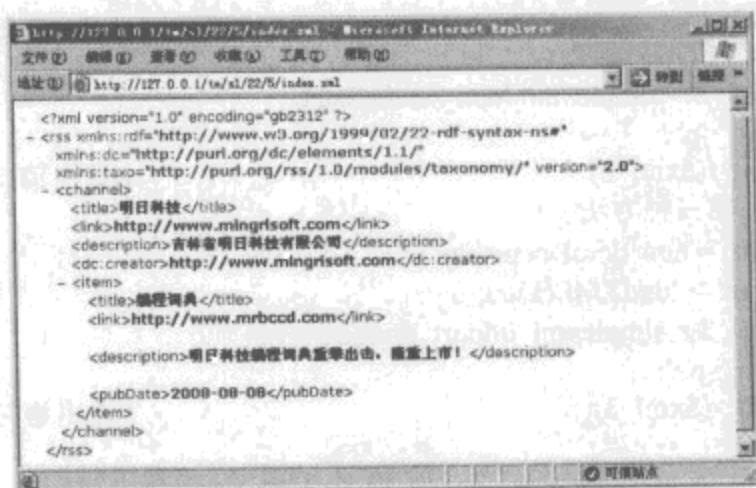


图 22.3 通过文件系统函数创建 XML 文件

有关通过 DOM 对象动态创建 XML 文档，可参考 22.5 节实战中的内容。

## 22.4 SimpleXML

### ■ 视频讲解：光盘\TM\Video\22\SimpleXML 类库.exe

PHP 提供了很多操作 XML 的方法，如 XML 语义解析函数、DOMXML 函数和 SimpleXML 函数等。其中，SimpleXML 函数是 PHP 5 新加入的内容，是 PHP 5 的核心部分，无须安装或引用外部 PECL 扩展，就能够很好地处理简单的、类似记录的数据，要比使用 DOM 简单得多。

## 22.4.1 创建 SimpleXML 对象

使用 SimpleXML 首先要创建对象。共有 3 种方法可以创建对象，分别是：

- ☒ Simplexml\_load\_file() 函数，将指定的文件解析到内存中。
- ☒ Simplexml\_load\_string() 函数，将创建的字符串解析到内存中。
- ☒ Simplexml\_load\_dom() 函数，将一个使用 dom 函数创建的 domDocument 对象导入到内存中。

**例 22.6 使用 3 个函数分别创建 3 个对象，并使用 print\_r 来输出 3 个对象，代码如下：（实例位置：光盘\TM\Instances\22\6）**

```
<?php
header("Content-Type:text/html;charset=utf-8"); //设置编码
/* 第一种方法 */
$xml_1 = simplexml_load_file("5.xml");
print_r($xml_1); //输出第一个对象
/* 第二种方法 */
$str = <<<XMLCONTENT
<?xml version="1.0" encoding="gb2312" ?>
<Object>
    <ComputerBook>
        <title>PHP 从入门到精通</title>
    </ComputerBook>
</Object>
XMLCONTENT;
$xml_2 = simplexml_load_string($str);
echo '<p>';
print_r($xml_2); //输出第二个对象
/* 第三种方法 */
$dom = new domDocument();
$dom ->loadXML($str);
$xml_3 = simplexml_import_dom($dom);
echo '<p>';
print_r($xml_3); //输出第三个对象
?>
```

结果为：

SimpleXMLElement Object ( [ComputerBook] => SimpleXMLElement Object ( [title] => PHP 从入门到精通 ) )

SimpleXMLElement Object ( [ComputerBook] => SimpleXMLElement Object ( [title] => PHP 从入门到精通 ) )

SimpleXMLElement Object ( [ComputerBook] => SimpleXMLElement Object ( [title] => PHP 从入门到精通 ) )

从结果中可以看出，即使 XML 的数据源不同，只要结构相同，那么输出的结果也是相同的。

 注意：程序中的 header() 函数设置了 html 编码。虽然在 XML 文档中设置了编码格式，但那只是针对 XML 文档的，在 html 输出时也要设置编码格式。

### 22.4.2 遍历所有子元素

创建对象后,就可以使用 SimpleXML 的其他函数来读取数据。使用 SimpleXML 对象中的 children()方法和 foreach 语句可以遍历所有子节点元素。

例 22.7 使用 children() 函数遍历所有子节点, 代码如下: (实例位置: 光盘\TM\Instances\22\7)

```
<?php
header('Content-Type:text/html;charset=utf-8');
/* 创建 XML 格式的字符串 */
$str = <<<XML
<?xml version='1.0' encoding='gb2312'?>
<object>
    <book>
        <computerbook>PHP 从入门到精通</computerbook>
    </book>
    <book>
        <computerbook>PHP 项目开发全程实录</computerbook>
    </book>
</object>
XML;
/* **** */
$xml = simplexml_load_string($str);
foreach($xml->children() as $layer_one){
    print_r($layer_one);
    echo '<br>';
    foreach($layer_one->children() as $layer_two){
        print_r($layer_two);
        echo '<br>';
    }
}
?>
```

运行结果如图 22.4 所示。

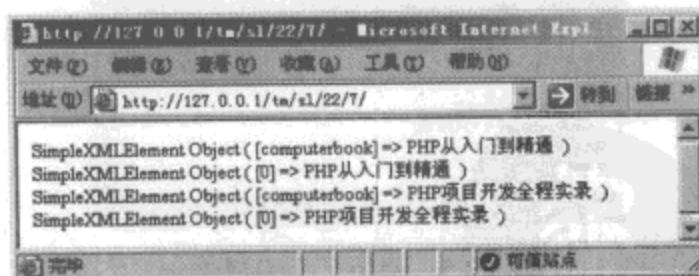


图 22.4 遍历节点

### 22.4.3 遍历所有属性

利用 SimpleXML 不仅可以遍历子元素, 还可以遍历元素的属性, 使用的是 SimpleXML 对象中的

attributes()方法。

**例 22.8 使用 attributes()方法遍历所有的元素属性, 实例代码如下:** (实例位置: 光盘\TM\Instances\22\8)

```
<?php
header("Content-Type:text/html;charset=utf-8"); //设置编码
/* 创建 XML 格式的字符串 */
$str = <<<XML
<?xml version='1.0' encoding='gb2312'?>
<object name='commodity'>
    <book type='computerbook'>
        <bookname name='PHP 从入门到精通' />
    </book>
    <book type='historybook'>
        <bookname name='上下五千年' />
    </book>
</object>
XML;
$xml = simplexml_load_string($str); //创建一个 SimpleXML 对象
foreach($xml->children() as $layer_one){ //循环子节点元素
    foreach($layer_one->attributes() as $name => $vl){ //输出各个节点的属性和值
        echo $name.':::'.$vl;
    }
    echo '<br>';
    foreach($layer_one->children() as $layer_two){ //输出第二层节点元素
        foreach($layer_two->attributes() as $nm => $vl){ //输出各个节点的属性和值
            echo $nm.':::'.$vl;
        }
        echo '<br>';
    }
}
?>
```

运行结果如图 22.5 所示。

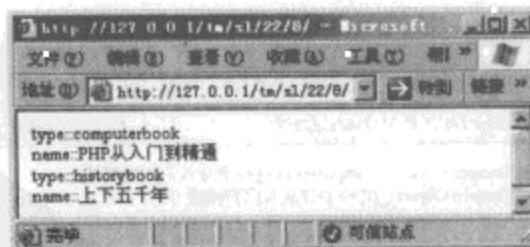


图 22.5 遍历子元素属性

#### 22.4.4 访问特定节点元素和属性

SimpleXML 对象除了可以使用上面两个方法来遍历所有的节点元素和属性, 还可以访问特定的数据元素。SimpleXML 对象可以通过子元素的名称对该子元素赋值, 或使用子元素的名称数组来对该子

元素的属性赋值。

**例 22.9** 使用 SimpleXML 对象直接对 XML 元素和属性进行访问，实例代码如下：（实例位置：光盘\TM\Instances\22\9）

```
<?php
header('Content-Type:text/html;charset=utf-8');
/* 创建 XML 格式的字符串 */
$str = <<<XML
<?xml version='1.0' encoding='gb2312'?>
<object name='商品'>
    <book>
        <computerbook>PHP 从入门到精通</computerbook>
    </book>
    <book>
        <computerbook name='PHP 项目开发全程实录' />
    </book>
</object>
XML;
/* **** */
$xml = simplexml_load_string($str);
echo $xml[name].<br>; //创建 SimpleXML 对象
echo $xml->book[0]->computerbook.<br>; //输出根元素的属性 name
echo $xml->book[1]->computerbook['name'].<br>; //输出子元素中 computerbook 的值
//输出 computerbook 的属性值
?>
```

运行结果如图 22.6 所示。

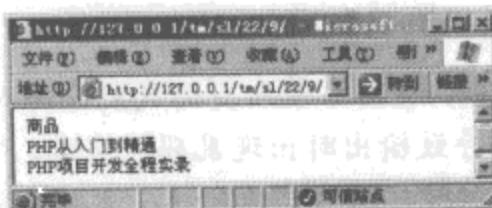


图 22.6 访问特定的节点元素和属性

#### 22.4.5 修改 XML 数据

对 XML 数据的修改，同读取 XML 数据相似。如例 22.9，在访问特定元素或属性时，也可以对其进行修改操作。

**例 22.10** 首先读取 XML 文档，然后输出根元素的属性 name，接着修改子元素 computerbook，最后输出修改后的值。实例代码如下：（实例位置：光盘\TM\Instances\22\10）

```
<?php
/* 设置编码格式 */
header('Content-Type:text/html;charset=utf-8');
/* 创建 XML 格式的字符串 */
$str = <<<XML
<?xml version='1.0' encoding='gb2312'?>
```

```

<object name='商品'>
    <book>
        <computerbook type='PHP 入门应用'>PHP 从入门到精通</computerbook>
    </book>
</object>
XML;
/* **** */
/* 创建 SimpleXML 对象 */
$xml = simplexml_load_string($str);
/* 输出根目录属性 name 的值 */
echo $xml[name].'  
';
/* 修改子元素 computerbook 的属性值 type */
$xml->book->computerbook['type'] = iconv('gb2312','utf-8','PHP 程序员必备工具');
/* 修改子元素 computerbook 的值 */
$xml->book->computerbook = iconv('gb2312','utf-8','PHP 函数参考大全');
/* 输出修改后的属性和元素值 */
echo $xml->book->computerbook['type'].'=>';
echo $xml->book->computerbook;
?>

```

运行结果如图 22.7 所示。

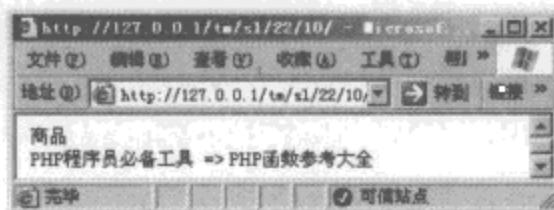


图 22.7 修改元素和属性值

说明：iconv() 函数是转换编码函数。有时希望向页面或文件写入数据，但添加的数据的编码格式和文件原有编码格式不符，导致输出时出现乱码。这时，可使用 iconv() 函数将数据从输入时所使用的编码转换为另一种编码格式后再输出。本例中是将字符串'PHP 程序员必备工具'从 GB2312 的编码格式转换成 UTF-8 编码格式。

#### 22.4.6 保存 XML 文档

数据在 SimpleXML 对象中所做的修改，其实是在系统内存中做的改动，而原文档根本没有变化。当关掉网页或清空内存时，数据又会恢复。要保存一个修改过的 SimpleXML 对象，可以使用 asXML() 方法来实现。该方法可以将 SimpleXML 对象中的数据格式化为 XML 格式，然后再使用 file 函数中的写入函数将数据保存到 XML 文件中。

**例 22.11** 首先从 11.xml 文档中生成 SimpleXML 对象，然后对 SimpleXML 对象中的元素进行修改，最后将修改后的 SimpleXML 对象再保存到 11.xml 文档中，代码如下：(实例位置：光盘\TM\Instances\22\11)

11.xml 文档的内容：

```
<?xml version="1.0" encoding="gb2312"?>
<object name="商品">
    <book>
        <computerbook type="PHP 入门应用">PHP 从入门到精通</computerbook>
    </book>
</object>
```

index.php 文件的代码:

```
<?php
/* 创建 SimpleXML 对象 */
$xml = simplexml_load_file('11.xml');
/* 修改 XML 文档内容 */
$xml->book->computerbook['type'] = iconv('gb2312','utf-8','PHP 程序员必备工具');
$xml->book->computerbook = iconv('gb2312','utf-8','PHP 函数参考大全');
/* 格式化对象$xml */
$modi = $xml->asXML();
/* 将对象保存到 10.xml 文档中 */
file_put_contents('10.xml',$modi);
/* 重新读取 XML 文档 */
$str = file_get_contents('10.xml');
/* 输出修改后的文档内容 */
echo $str;
?>
```

运行结果如图 22.8 所示。

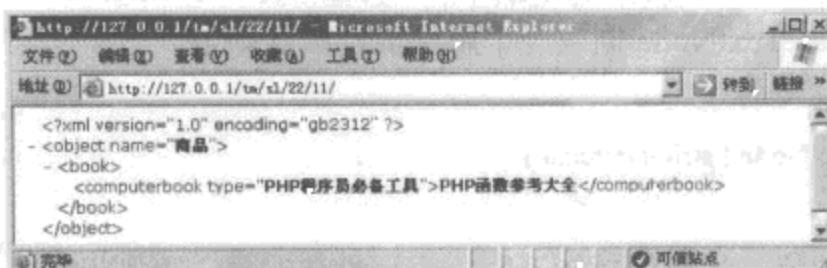


图 22.8 保存 SimpleXML 对象

## 22.5 实战

### 22.5.1 动态创建 XML 文档

视频讲解：光盘\TM\Video\22\动态创建 XML 文档.exe

使用 SimpleXML 对象可以十分方便地读取和修改 XML 文档，但却无法动态建立 XML。这时就需要使用 DOM 来实现。DOM (Document Object Model，文档对象模型)，通过树状结构模式来遍历 XML 文档。使用 DOM 遍历文档的好处是不需要标记就可以显示全部内容。但其缺点同样明显，就是十分消耗内存。

通过 DOM 类库创建 XML 文档应用到的方法如下：

(1) 通过 new DomDocument() 来实例化一个 DomDocument 对象，语法如下。

```
$dom = new DomDocument('1.0','gb2312'); //创建 DOM 对象
```

(2) 通过 \$dom->createElement 创建一个新元素节点，语法如下。

```
class DomDocument {
    DomElement createElement ( string name [, string value] )
}
```

参数 name 是创建的元素的名称，参数 value 是创建元素的值。如果成功则返回一个元素节点对象，否则返回一个错误。

(3) 通过 \$dom->appendChild 实现元素节点的附加，语法如下。

```
class DomNode {
    DomNode appendChild ( DOMNode newnode )
}
```

参数 newnode 是将要附加的节点，会返回更多的节点。

(4) 通过 \$dom->createAttribute 创建一个新的属性，语法如下。

```
class DomDocument {
    DomAttr createAttribute ( string name )
}
```

参数 name 是属性的名称，如果成功则返回一个属性对象，否则返回一个错误。

(5) 通过 \$dom->createTextNode 创建一个属性值，语法如下。

```
class DomDocument {
    DomText createTextNode ( string content )
}
```

参数 content 是属性的值，如果成功则返回一个属性值对象，否则返回一个错误。

(6) 通过 \$dom->saveXML 生成一个 XML 文档，语法如下。

```
class DomDocument {
    string saveXML ( [DOMNode node [, int options]] )
}
```

参数 node 指定生成的节点变量名称，参数 options 是附加的值。如果成功则返回一个 XML 文档，否则返回一个错误。

(7) file\_put\_contents() 函数，将指定文件中的内容读入到一个字符串。如果操作系统支持内存映射技术，还会应用内存映射技术来增强其读入文件的性能，语法如下。

```
string file_get_contents(string filename [, int use_include_path [, resource context]])
```

参数说明如表 22.3 所示。

表 22.3 file\_put\_contents() 函数的参数说明

参 数	说 明
filename	必要参数。用于指定某文件的路径及名称
use_include_path	可选参数。如果将该参数设置为 true, PHP 会尝试按照 include_path 标准包含路径中的每个指向去打开文件
context	可选参数。设置提高文件流性能的一些选项

上述 6 种方法和一个函数是在创建 XML 文档中需要使用的,下面应用上述方法动态创建一个 XML 文档。

**例 22.12 应用上述方法,将从数据库中读取出的数据生成一个标准的 RSS 2.0 格式的 XML 文档。操作步骤如下:** (实例位置: 光盘\TM\Instances\22\12)

(1) 首先创建 conn.php 文件, 实现与数据库的连接, 代码如下。

```
<?php
$id=mysql_connect("localhost","root","root") or die('连接失败:'.mysql_error());
if(mysql_select_db("db_database22",$id))
    echo "";
else
    echo ('数据库选择失败:'.mysql_error());
mysql_query("set names gb2312");
?>
```

(2) 创建 index.php 文件, 通过 include\_once 语句调用 conn.php 文件连接数据库, 调用 rss.php 文件动态生成 XML 文档, 创建查看 XML 文件的超链接, 代码如下。

```
<?php
include_once("conn/conn.php");
include_once("rss.php");
?>
<a href="rss.xml">查看 rss.xml 文件中的内容</a>
```

(3) 创建 rss.php 文件, 从数据库中读取数据, 动态生成标准的 RSS 2.0 格式的 XML 文档, 关键代码如下。

```
<?php
$dom = new DomDocument('1.0','gb2312'); //创建 DOM 对象
$object = $dom->createElement('rss'); //创建根节点 rss
$dom->appendChild($object); //将创建的根节点添加到 dom 对象中
$type1 = $dom->createAttribute('xmlns:rdf'); //创建一个节点属性 xmlns:rdf
$object->appendChild($type1); //将属性追加到 rss 根节点中
$type1_value = $dom->createTextNode('http://www.w3.org/22-rdf-syntax-ns#'); //创建一个属性值
$type1->appendChild($type1_value); //将属性值赋给属性 xmlns:rdf
$channel = $dom->createElement('channel'); //创建节点 channel
$object->appendChild($channel); //将节点 channel 追加到根节点 rss 下
$title = $dom->createElement('title'); //创建节点 title
$channel->appendChild($title); //将节点追加到 channel 节点下
$title_value = $dom->createTextNode(iconv('gb2312','utf-8','明日科技')); //创建元素值
$title->appendChild($title_value); //将值赋给 title 节点
```

```
$modi = $dom->saveXML();
file_put_contents('Rss.xml',$modi);
?>
```

//生成 XML 文档  
//将对象保存到 Rss.xml 文档中

运行结果如图 22.9 所示。

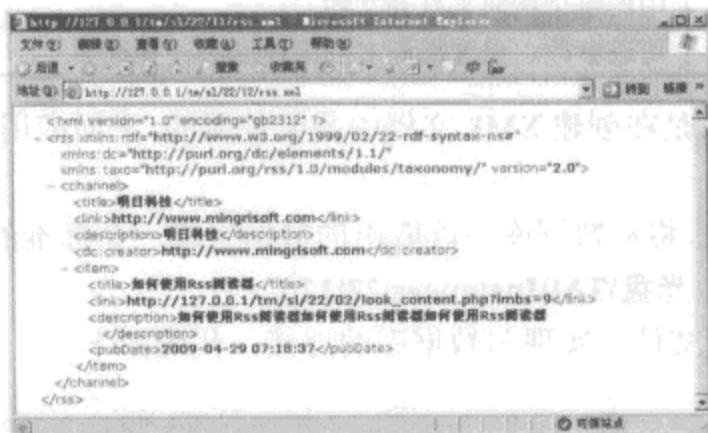


图 22.9 使用 DOM 创建 XML 文档

## 22.5.2 动态读取 XML 文档

### ■ 视频讲解：光盘\TM\Video\22\动态读取 XML 文档.exe

通过 DOM 类库读取 XML 文档主要应用的是 `getElementsByTagName` 方法。

`DOMDocument->getElementsByTagName()` 方法用于获取标签的名称，返回值为一个新的 `DOMNodeList` 对象。语法如下：

```
class DOMDocument {
    DOMNodeList getElementsByTagName ( string name )
}
```

参数 `name` 是 XML 文档中节点的名称。

例 22.13 应用 `DOMDocument->getElementsByTagName()` 方法，读取指定 XML 文档中的内容，并且通过 `foreach` 语句输出 XML 文档中的数据，操作步骤如下：（实例位置：光盘\TM\Instances\22\13）

(1) 创建 `Message_XML` 类，继承 PHP 的预定义类 `DOMDocument`，在该类中创建 `show_message()` 方法用于输出 XML 文件中的内容，代码如下。

```
<?php
class Message_XML extends DOMDocument{ //创建一个类
private $root;
public function __construct(){ //构造函数
    parent::__construct();
    $this->load("Rss.xml"); //读取指定的 XML 文件
}
public function show_message(){ //输出 XML 文件中的内容
    $root=$this->documentElement; //定义根目录
    $node_record=$this->getElementsByTagName("item"); //获取节点 item 下的标签
    $node_record_length=$node_record->length; //获取标签的数量
    //定义表格，输出数据
}
```

```

print"<table width='750' border='1' cellpadding='1' cellspacing='1' bordercolor='#FFFFFF' bgcolor='#99CC00>
<tr><td width='131' align='center' bgcolor='#FFFFFF'><span class='STYLE_5'>标题</span></td><td width='258' align='center' bgcolor='#FFFFFF'><span class='STYLE_5'>链接地址</span></td><td width='343' align='center' bgcolor='#FFFFFF'><span class='STYLE_5'>内容摘要</span></td></tr>";
for($i=0;$i<$node_record->length;$i++){ //根据标签的数量执行 for 循环
    $k=0;
    foreach($node_record->item($i)->childNodes as $articles){ //通过 foreach 语句输出标签中的数据
        $field[$k]=iconv("UTF-8","GB2312",$articles->textContent);
        $k++;
    }
    print"<tr><td align='left' bgcolor='#FFFFFF'>";
    print"$field[0]";
    print"</td><td align='left' bgcolor='#FFFFFF'>";
    print"$field[1]";
    print"</td><td align='left' bgcolor='#FFFFFF'>";
    print"$field[2]";
    print"</td></tr>";
}
print"</table>";
}
}
?>

```

(2) 实例化 Message\_XML 类，并通过实例化的对象\$hawkXML 调用 show\_message()方法，输出存储在 XML 文件中的内容，关键代码如下。

```

<?php
$hawkXML=new Message_XML;
$Action=$_GET['Action'];
switch($Action){
    case "show_message":
        $hawkXML->show_message();
        break;
}
?>

```

运行结果如图 22.10 所示。



图 22.10 通过 DOM 读取 XML 文档中的数据

PHP 中的 DOM 函数库十分庞大，这里只是对创建和读取 XML 文档所使用的方法进行了介绍。感兴趣的读者可以参考 XML 和 PHP 的官方手册来了解更多关于 DOM 的知识。

### 22.5.3 通过 XPath 查询 XML 中的数据

#### 视频讲解：光盘\TM\Video\22\通过 XPath 查询 XML 中的数据.exe

通过 DOM 类库查询 XML 数据主要应用的是 XPath，通过 DOMXPath->query()方法来执行查询特定的节点。DOMXPath->query()方法在 XML 中执行查询的操作，查询符合一些模式语法的特定节点。返回值为一个节点列表对象。语法如下：

```
class DOMXPath {
    DOMNodeList query ( string expression [, DOMNode contextnode] )
}
```

参数 expression 是执行的查询语句，该参数的输出写法有多种，进而可分割执行不同的查询操作。例如：

/rss/channel/item[position() = 1]/title	返回第一个 item 元素的所有 title
/rss/channel/item/title[@id = '23']	返回所有含有 id 属性且值为 23 的 title
/rss/channel/&folder&/title	返回所有 articles 元素下面的 title (&folder&代表目录深度)

其返回值为，如果成功则返回一个节点列表对象 DOMNodeList，如果失败则返回一个空的节点列表对象 DOMNodeList。

**例 22.14 应用 DOMXPath->query()方法，实现对 XML 文档中的数据进行查询操作，查询 XML 文档中标题是“明日科技”的数据，将查询结果输出到文本框中，操作步骤如下：（实例位置：光盘\TM\Instances\22\14）**

(1) 编写一个 Message\_XML 类，该类继承于 DomDocument 类，应用该类的 load()方法调用 XML 文件 Rss.xml，在 Message\_XML 类中建立 select\_message()方法，通过该函数实现查询 XML 文件中数据的操作，代码如下。

```
<?php
    class Message_XML extends DomDocument{ //定义类
        private $Root;
        public function __construct(){ //构造函数
            parent::__construct();
            $this->load("Rss.xml");} //加载指定的文件
        public function select_message($Action){ //创建方法
            $Root = $this->documentElement;
            $xpath = new DOMXPath($this); //定义 DOMXPath
            $node_Record = $xpath->query("//item[title='$Action']");
            for($i=0;$i<$node_Record->length;$i++){ //执行查询语句
                $g=0;
                foreach($node_Record->item($i)->childNodes as $node){//通过 foreach 语句读取返回对象中的数据
                    $field[$g]=iconv("UTF-8","GB2312",$node->textContent); //实现编码格式的转换
                    $g++;
                }
            }
        }
    }

```

```

    }
    //输出查询结果
    //省略了输出查询结果的代码
}
}
?

```

(2) 创建一个操作界面，插入一个表单、文本框，设置提交按钮，然后对 Message\_XML 类进行实例化，并应用实例化后的对象 \$hawkXML 调用 select\_message()方法执行查询操作，关键代码如下。

```

<?php
if($Submit==true){
    $hawkXML = new Message_XML;
    $hawkXML->select_message(iconv("GB2312","UTF-8","$_POST[Action]"));
}
?>

```

**技巧：**在执行查询操作过程中，必须要对提交的字符串进行编码格式的转换，因为在 XML 文件中使用的是 UTF-8 的编码格式，而程序中提交的数据使用的是 GB2312 编码，所以要将提交的数据转换成 UTF-8 的编码格式，否则是不能进行匹配查询的。对编码格式的转换使用的是 iconv() 函数。

运行本实例，在文本框中输入关键字“明日科技”，将输入如图 22.11 所示的内容。

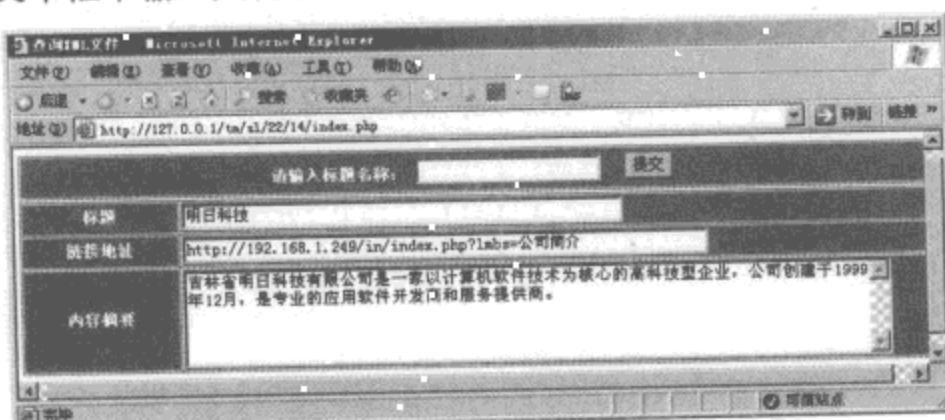


图 22.11 通过 XPath 查询 XML 中的数据

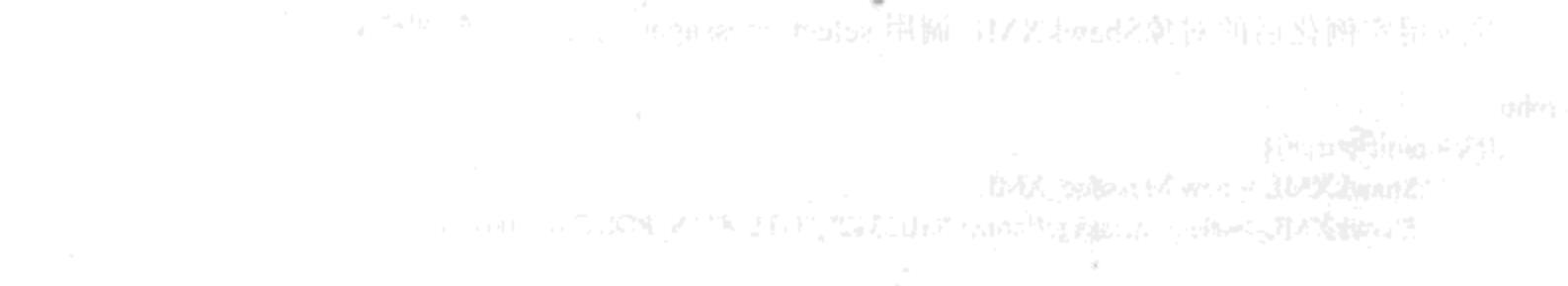
## 22.6 本章小结

本章首先介绍了 XML 的基础语法，使读者对 XML 有了一个初步的印象；然后讲解了如何在 PHP 中创建一个 XML 文档；接着对 PHP 5 最新的 SimpleXML 类库进行了详细的介绍；最后使用 DOM 对象模型动态创建了一个 XML 文档。

希望读者可以通过本章的概念和实例，初步掌握 PHP 对 XML 文档的操作，为学习后面的 Ajax 技术做好准备。

## 22.7 实战练习

1. 通过 DOM 类库向 XML 中添加数据。（答案位置：光盘\TM\Instances\22\15）
2. 通过 replaceChild()方法修改 XML 文档。（答案位置：光盘\TM\Instances\22\16）
3. 通过 removeChild()方法删除 XML 文档。（答案位置：光盘\TM\Instances\22\17）



在本章中，我们学习了如何使用 DOM 类库操作 XML 文档。通过学习，读者应该已经掌握了如何使用 DOM API 来解析 XML 文档、修改 XML 文档以及生成 XML 文档的方法。通过本章的练习，读者可以进一步巩固所学知识。

在下一章中，我们将学习如何使用 PHP 的 SOAP 扩展来实现 Web 服务。通过学习，读者将能够掌握如何使用 SOAP 扩展来发布 Web 服务以及如何调用别人发布的 Web 服务。

在下一章中，我们将学习如何使用 PHP 的 SOAP 扩展来实现 Web 服务。通过学习，读者将能够掌握如何使用 SOAP 扩展来发布 Web 服务以及如何调用别人发布的 Web 服务。

在下一章中，我们将学习如何使用 PHP 的 SOAP 扩展来实现 Web 服务。通过学习，读者将能够掌握如何使用 SOAP 扩展来发布 Web 服务以及如何调用别人发布的 Web 服务。

在下一章中，我们将学习如何使用 PHP 的 SOAP 扩展来实现 Web 服务。通过学习，读者将能够掌握如何使用 SOAP 扩展来发布 Web 服务以及如何调用别人发布的 Web 服务。

在下一章中，我们将学习如何使用 PHP 的 SOAP 扩展来实现 Web 服务。通过学习，读者将能够掌握如何使用 SOAP 扩展来发布 Web 服务以及如何调用别人发布的 Web 服务。

# 第23章

## Ajax 技术

( 视频讲解: 77分钟)

Ajax 是一种创建灵活、交互性强的 Web 应用技术，使用 Ajax 可以实现响应迅速、无刷新的 Web 应用。Ajax 集成了目前浏览器中通过 JavaScript 脚本可以实现的所有功能，并以一种崭新的方式来使用这些技术，使得 B/S 结构的 Web 开发变得更加灵活。本章将深入浅出地剖析 Ajax 的各个方面，并通过具体应用使读者掌握 Ajax 技术的使用方法及 jQuery 框架的基本应用方法。

通过阅读本章，可以学习到：

- Ajax 的概念及开发模式
- Ajax 的组成及各部分的功能
- Ajax 开发中的常见问题
- XMLHttpRequest 对象的应用
- 基于 Ajax 技术的 Web 应用的开发步骤
- jQuery 框架的应用

## 23.1 Ajax 概述

随着 Web 2.0 时代的到来，Ajax 技术产生并逐渐成为主流的 Web 应用。相对于传统的 Web 应用开发，Ajax 运用的是更加先进、更加标准化、更加高效的 Web 开发技术体系。由于 Ajax 是客户端技术，其编译和执行由浏览器完成，所以无论哪种 Web 语言都可以使用 Ajax 技术开发前台应用和页面特效。本章主要介绍 Ajax 技术及如何在 PHP 中应用 Ajax 技术。

### 23.1.1 什么是 Ajax

Ajax 是 JavaScript、XML、CSS、DOM 等多种已有技术的组合，可以实现客户端的异步请求操作，这样可以实现在不需要刷新页面的情况下与服务器进行通信，从而减少了用户的等待时间。Ajax 是由 Jesse James Garrett 创造的，是 Asynchronous JavaScript And XML 的缩写，即异步 JavaScript 和 XML 技术。可以说，Ajax 是“增强的 JavaScript”，是一种可以调用后台服务器获得数据的客户端 JavaScript 技术，支持更新部分页面的内容而不重载整个页面。

### 23.1.2 Ajax 的开发模式

传统的 Web 应用模式中，页面中用户的每一次操作都将触发一次返回 Web 服务器的 HTTP 请求，服务器进行相应的处理后，返回一个 HTML 页面给客户端浏览器。Ajax 中，页面中用户的操作将通过 Ajax 引擎与服务器端进行通信，然后将返回结果提交给客户端页面的 Ajax 引擎，再由 Ajax 引擎来决定将这些数据插入到页面的指定位置。一个典型的 Ajax 应用如图 23.1 所示。

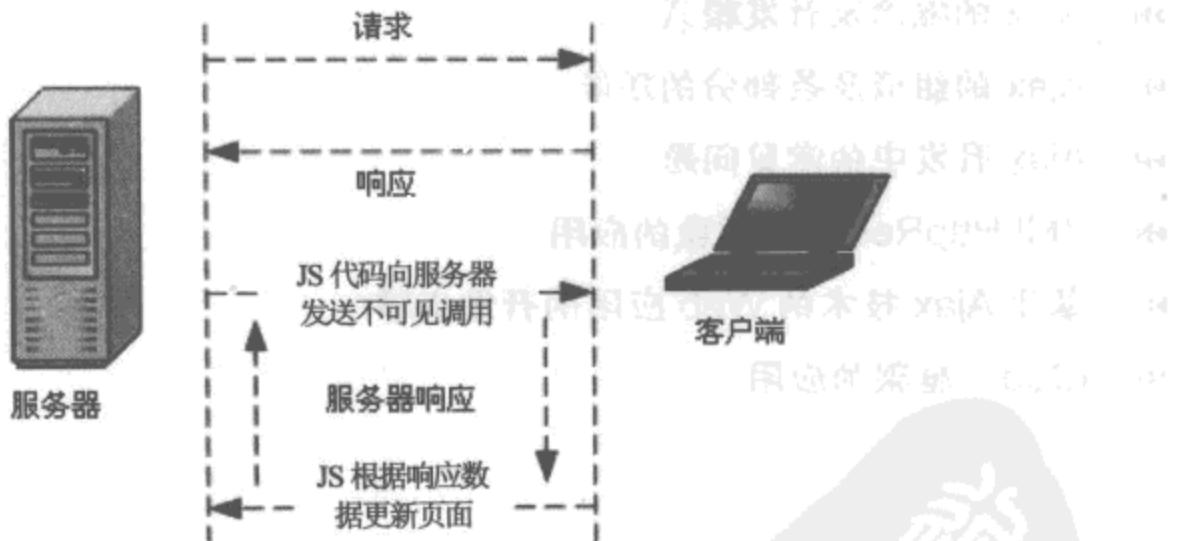


图 23.1 典型的 Ajax 应用

### 23.1.3 Ajax 的优点

Ajax 在用户与服务器之间引入了 Ajax 引擎作为中间媒介，Web 页面不用打断交互流程就可以重新

加载，实现动态更新，从而可以消除网络交互过程中“处理—等待—处理—等待”的缺点。也就是说，不需刷新客户端浏览器就可以实现重新向服务器发出请求。

使用 Ajax 的优点具体表现在以下几个方面：

- 无刷新更新页面。Ajax 使用 XMLHttpRequest 对象向服务器发送请求并得到服务器响应，在不需要重新载入整个页面的情况下，即可根据服务器端返回的请求更改页面内容，使用 Ajax 技术可以有效减少用户等待页面刷新的时间。
- 可以把一部分以前由服务器负担的工作转移到客户端，利用客户端闲置的资源进行处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。
- 减轻服务器的负担。Ajax 的原则是“按需求获取数据”，可以最大程度地减少冗余请求和响应对服务器造成的负担。
- 可以调用 XML 等外部数据，进一步促进 Web 页面显示和数据的分离。
- Ajax 是基于标准化并被广泛支持的技术，不需要下载插件或者小程序。

## 23.2 Ajax 技术的组成

Ajax 技术并不是一种新型的语言，而是多种已有的 Web 技术的整合，主要包括 JavaScript、XML 语言、DOM 和 CSS 等。

### 23.2.1 JavaScript 脚本语言

JavaScript 是一种解释型的、基于对象的脚本语言，其核心已经嵌入到目前主流的 Web 浏览器中。虽然平时应用最多的是通过 JavaScript 实现一些网页特效及表单数据验证等功能，但 JavaScript 可以实现的功能远不止这些。JavaScript 是一种具有丰富的面向对象特性的程序设计语言，利用它能执行许多复杂的任务，例如，Ajax 就是利用 JavaScript 将 DOM、XHTML（或 HTML）、XML 以及 CSS 等技术综合起来，并控制它们的行为。因此，要开发一个复杂高效的 Ajax 应用程序，就必须对 JavaScript 有深入的了解。

JavaScript 不是 Java 语言的精简版，并且只能在某个解释器或“宿主”上运行，如 ASP、PHP、JSP、Internet 浏览器或者 Windows 脚本宿主。

JavaScript 是一种宽松类型的语言，宽松类型意味着不必显式定义变量的数据类型。此外，在大多数情况下，JavaScript 将根据需要自动进行转换。例如，如果将一个数值添加到由文本组成的某项（一个字符串），该数值将被转换为文本。

### 23.2.2 XMLHttpRequest 对象

#### ■ 视频讲解：光盘\TM\Video\23\XMLHttpRequest 对象.exe

XMLHttpRequest 是 Ajax 中最核心的技术，它是一个具有应用程序接口的 JavaScript 对象，能够使用超文本传输协议（HTTP）连接一个服务器，是微软公司为了满足开发者的需要，于 1999 年在 IE 5.0 浏览器中率先推出的。现在许多浏览器都对其提供了支持，不过实现方式与 IE 有所不同。使用

XMLHttpRequest 对象，Ajax 可以像桌面应用程序一样只同服务器进行数据层面的交换，而不用每次都刷新页面，也不用每次都将数据处理的工作交给服务器来做，这样既减轻了服务器负担又加快了响应速度，缩短了用户等待的时间。

使用 XMLHttpRequest 对象发送请求和处理响应之前，首先需要初始化该对象，由于 XMLHttpRequest 不是一个 W3C 标准，所以对于不同的浏览器，初始化的方法也是不同的。

#### IE 浏览器

IE 浏览器把 XMLHttpRequest 实例化为一个 ActiveX 对象。具体方法如下：

```
var http_request = new ActiveXObject("Msxml2.XMLHTTP");
```

或者

```
var http_request = new ActiveXObject("Microsoft.XMLHTTP");
```

上面语法中的 Msxml2.XMLHTTP 和 Microsoft.XMLHTTP 是针对 IE 浏览器的不同版本而进行设置的，目前比较常用的是这两种。

#### Mozilla、Safari 等其他浏览器

Mozilla、Safari 等其他浏览器把它实例化为一个本地 JavaScript 对象。具体方法如下：

```
var http_request = new XMLHttpRequest();
```

为了提高程序的兼容性，可以创建一个跨浏览器的 XMLHttpRequest 对象。方法很简单，只需要判断一下不同浏览器的实现方式，如果浏览器提供了 XMLHttpRequest 类，则直接创建一个实例，否则使用 IE 的 ActiveX 控件。具体代码如下：

```
if (window.XMLHttpRequest) { //Mozilla、Safari 等浏览器
    http_request = new XMLHttpRequest();
}
else if (window.ActiveXObject) { //IE 浏览器
    try {
        http_request = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            http_request = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e) {}
    }
}
```

由于 JavaScript 具有动态类型特性，而且 XMLHttpRequest 对象在不同浏览器上的实例是兼容的，所以可以用同样的方式访问 XMLHttpRequest 实例的属性和方法，不需要考虑创建该实例的方法。下面介绍 XMLHttpRequest 对象的常用方法和属性。

### 1. XMLHttpRequest 对象的常用方法

#### (1) open()方法

open()方法用于设置进行异步请求目标的 URL、请求方法以及其他参数信息，具体语法如下：

```
open("method","URL"[,asyncFlag[,userName[, "password"]]])
```

在上面的语法中，method 用于指定请求的类型，一般为 get 或 post；URL 用于指定请求地址，可以使用绝对地址或者相对地址，并且可以传递查询字符串；asyncFlag 为可选参数，用于指定请求方式，同步请求为 true，异步请求为 false，默认为 true；userName 为可选参数，用于指定用户名，没有时可省略；password 为可选参数，用于指定请求密码，没有时可省略。

### (2) send()方法

send()方法用于向服务器发送请求。如果请求声明为异步，该方法将立即返回，否则将等到接收到响应为止。具体语法格式如下：

```
send(content)
```

在上面的语法中，content 用于指定发送的数据，可以是 DOM 对象的实例、输入流或字符串，如果没有参数需要传递可以设置为 null。

### (3) setRequestHeader()方法

setRequestHeader()方法为请求的 HTTP 头设置值。具体语法格式如下：

```
setRequestHeader("label", "value")
```

在上面的语法中，label 用于指定 HTTP 头，value 用于为指定的 HTTP 头设置值。

 注意：setRequestHeader()方法必须在调用 open()方法之后才能调用。

### (4) abort()方法

abort()方法用于停止当前异步请求。

### (5) getAllResponseHeaders()方法

getAllResponseHeaders()方法用于以字符串形式返回完整的 HTTP 头信息，当存在参数时，表示以字符串形式返回由该参数指定的 HTTP 头信息。

## 2. XMLHttpRequest 对象的常用属性

XMLHttpRequest 对象的常用属性如表 23.1 所示。

表 23.1 XMLHttpRequest 对象的常用属性

属性	说 明
<code>onreadystatechange</code>	每次状态改变都会触发这个事件处理器，通常会调用一个 JavaScript 函数
<code>readyState</code>	请求的状态。有以下 5 个取值： 0=未初始化 1=正在加载 2=已加载 3=交互中 4=完成
<code>responseText</code>	服务器的响应，表示为字符串

续表

属性	说 明
responseXML	服务器的响应，表示为 XML。这个对象可以解析为一个 DOM 对象
status	返回服务器的 HTTP 状态码，如： 200="成功" 202="请求被接受，但尚未成功" 400="错误的请求" 404="文件未找到" 500="内部服务器错误"
statusText	返回 HTTP 状态码对应的文本

### 3. XMLHttpRequest 对象与服务器交互

XMLHttpRequest 对象最大的用途就是不需要刷新页面就可以与服务器进行交互。可以将 Ajax 与服务器的交互分为以下 3 个步骤：

(1) 初始化 XMLHttpRequest 对象，关键代码如下。

```
var xmlhttp = false; // 定义 XMLHttpRequest 对象
try {
    // 如果浏览器支持 XMLHttpRequest 对象，创建 ActiveXObject 对象
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
    try {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e2) {}
}
if (!xmlhttp && typeof XMLHttpRequest != "undefined") {
    try{
        xmlhttp = new XMLHttpRequest();
    }catch(e3){ xmlhttp = false;}
}
```

(2) 设置请求状态和返回处理函数，语法格式如下。

```
xmlobj.onreadystatechange=function_name;
```

其中，xmlobj 是 XMLHttpRequest 的对象；function\_name 是用来处理请求状态和返回码的函数名（即回调函数）。

(3) 发送 HTTP 请求，语法格式如下。

```
xmlobj.open(send_method,url,flag);
```

其中，xmlobj 是 XMLHttpRequest 的对象；send\_method 是发送方法，可以是 GET 或者 POST，对应表单使用的方法；url 是页面要调用的地址；flag 是一个标记，如果为 true 则表示在等待被调用页面响应的时间内可以继续执行页面代码，反之为 false。

**例 23.1 通过 XMLHttpRequest 对象读取 HTML 文件，并输出读取结果。关键代码如下：（实例位置：光盘\TM\Instances\23\1）**

```

<script language="javascript">
var xmlhttp; //定义 XMLHttpRequest 对象
function createXmlHttpRequestObject(){
    if(window.ActiveXObject){
        try{
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }catch(e){
            xmlhttp=false;
        }
    }else{ //如果在 Mozilla 或其他的浏览器下运行
        try{
            xmlhttp=new XMLHttpRequest();
        }catch(e){
            xmlhttp=false;
        }
    }
    if(!xmlhttp)
        alert("返回创建的对象或显示错误信息");
    else
        return xmlhttp;
}
function ReqHtml(){
    createXmlHttpRequestObject();
    xmlhttp.onreadystatechange=StatHandler; //判断 URL 调用的状态值并处理
    xmlhttp.open("GET","text.html",true); //调用 text.html
    xmlhttp.send(null);
}
function StatHandler(){
    if(xmlhttp.readyState==4 && xmlhttp.status==200){
        document.getElementById("webpage").innerHTML=xmlhttp.responseText;
    }
}
</script>
<!--创建超链接-->
<a href="#" onclick="ReqHtml();>通过 XMLHttpRequest 对象请求 HTML 文件</a>
<!--通过 div 标签输出请求内容-->
<div id="webpage"></div>

```

本实例通过 DIV 标签输出请求的 HTML 页面。在 JavaScript 脚本中，应用 `document.getElementById()` 方法来获得页面元素。运行本实例，单击“通过 XMLHttpRequest 对象请求 HTML 文件”超链接，将输出如图 23.2 所示的页面。

通过 XMLHttpRequest 对象不但可以读取 HTML 文件，还可以读取文本文件、XML 文件，其实现交互的方法与读取 HTML 文件类似，这里不再举例。

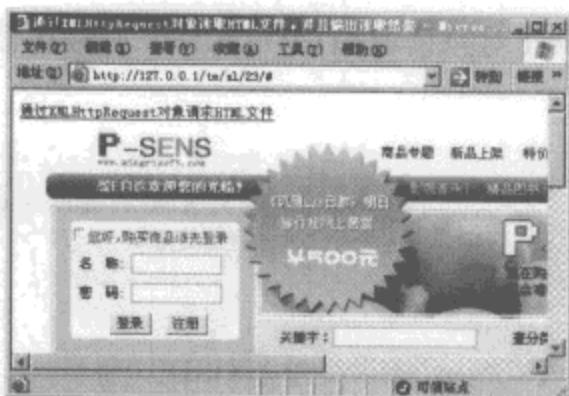


图 23.2 通过 XMLHttpRequest 对象读取 HTML 文件

### 23.2.3 XML 语言

XML 是 eXtensible Markup Language（可扩展的标记语言）的缩写，它提供了用于描述结构化数据的格式。XMLHttpRequest 对象与服务器交换的数据，通常采用 XML 格式，但也可以是基于文本的其他格式。

### 23.2.4 DOM

DOM 是 Document Object Model（文档对象模型）的缩写，它为 XML 文档的解析定义了一组接口。解析器读入整个文档，然后构建一个驻留内存的树结构，最后通过 DOM 可以遍历树以获取来自不同位置的数据，可以添加、修改、删除、查询和重新排列树及其分支。另外，还可以根据不同类型的的数据源来创建 XML 文档。在 Ajax 应用中，通过 JavaScript 操作 DOM，可以达到在不刷新页面的情况下实时修改用户界面的目的。

### 23.2.5 CSS

CSS 是 Cascading Style Sheet（层叠样式表）的缩写，是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。在 Ajax 中，通常使用 CSS 进行页面布局，并通过改变文档对象的 CSS 属性控制页面的外观和行为。CSS 是一种 Ajax 开发人员所需要的重要武器，提供了从内容中分离应用样式和设计的机制。虽然 CSS 在 Ajax 应用中扮演至关重要的角色，但它也是构建创建跨浏览器应用的一大阻碍，因为不同的浏览器厂商支持不同的 CSS 级别。

## 23.3 Ajax 与 PHP 的交互

在 23.2 节介绍 XMLHttpRequest 对象时，讲解了如何实现 Ajax 与 HTML 的交互。这里将介绍 XMLHttpRequest 对象与 PHP 的交互。通过 XMLHttpRequest 对象请求 PHP 页面有两种方式，一种是 GET 方式，另一种是 POST 方式。具体使用哪一种方式与 PHP 进行交互，取决于 PHP 页面中变量值的传递方法，如果使用 POST 方法进行传递变量值，则使用 POST 方式进行交互操作，否则使用 GET

方式进行交互操作。

### 23.3.1 通过 GET 方式与 PHP 进行交互

#### ■ 视频讲解：光盘\TM\Video\23\通过 GET 方式与 PHP 进行交互.exe

通过一个具体的实例来讲解如何通过 GET 方式与 PHP 进行交互。

**例 23.2 应用 GET 方式，通过 XMLHttpRequest 对象与 PHP 进行交互，其操作步骤如下：**（实例位置：光盘\TM\Instances\23\2）

(1) 创建 index.php 文件，编写 JavaScript 脚本，通过 Ajax 请求 searchrst.php 文件，执行查询操作，将查询结果定义到 DIV 标签中；创建 form 表单，提交查询的关键字，通过 DIV 标签输出查询结果，代码如下。

```

<script>
var xmlhttp; // 定义 XMLHttpRequest 对象
function createXmlHttpRequestObject(){
    if(window.ActiveXObject){ // 如果在 Internet Explorer 下运行
        try{
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }catch(e){
            xmlhttp=false;
        }
    }else{ // 如果在 Mozilla 或其他的浏览器下运行
        try{
            xmlhttp=new XMLHttpRequest();
        }catch(e){
            xmlhttp=false;
        }
    }
    if(!xmlhttp) // 返回创建的对象或显示错误信息
        alert("返回创建的对象或显示错误信息");
    else
        return xmlhttp;
}
function showsimple(){
    createXmlHttpRequestObject();
    var cont = document.getElementById("searchtxt").value;
    if(cont==""){
        alert('查询关键字不能为空！');
        return false;
    }
    xmlhttp.onreadystatechange=StatHandler; // 判断 URL 调用的状态值并处理
    xmlhttp.open("GET",'searchrst.php?cont='+cont,false);
    xmlhttp.send(null);
}
function StatHandler(){
    if(xmlhttp.readyState==4 && xmlhttp.status==200){
        document.getElementById("webpage").innerHTML=xmlhttp.responseText;
    }
}

```

```

    }
}
</script>
<form id="searchform" name="searchform" method="get" action="#">

```

(2) 创建 searchrst.php 文件，在该文件中，首先定义页面的编码格式，然后连接数据库，最后根据 Ajax 中传递的值执行查询操作，返回查询结果，代码如下。

```

<?php
    header('Content-type: text/html;charset=GB2312'); //指定发送数据的编码格式
    include_once 'conn/conn.php'; //连接数据库
    $cont = $_GET['cont']; //获取 Ajax 传递的查询关键字
    if(!empty($_GET['cont'])){ //如果关键字不为空
        $sql = "select * from tb_administrator where explains like '%".$cont."%'"; //定义 SQL 语句
        $result=mysql_query($sql,$conn); //执行模糊查询
        if(mysql_num_rows($result)>0){ //获取查询结果
            echo "<table width='500' border='1' cellpadding='1' cellspacing='1' bordercolor=#FFFFCC' bgcolor=#666666>";
            echo "<tr><td height='30' align='center' bgcolor=#FFFFFF>ID</td><td align='center' bgcolor=#FFFFFF>名称</td><td align='center' bgcolor=#FFFFFF>编号</td><td align='center' bgcolor=#FFFFFF>描述</td></tr>";
            while($myrow=mysql_fetch_array($result)){ //循环输出查询结果
                echo "<tr><td height='22' bgcolor=#FFFFFF>".$myrow[id]."</td>";
                echo "<td bgcolor=#FFFFFF>".$myrow[user]."</td>";
                echo "<td bgcolor=#FFFFFF>".$myrow[number]."</td>";
                echo "<td bgcolor=#FFFFFF>".$myrow[explains]."</td>";
                echo "</tr>";
            }
            echo "</table>";
        }else{
            echo "没有符合条件的数据";
        }
    }
?>

```

运行本实例，在查询的文本框中输入关键字“PHP”，单击“查询”按钮，在当前页中将输出如图 23.3 所示的页面。

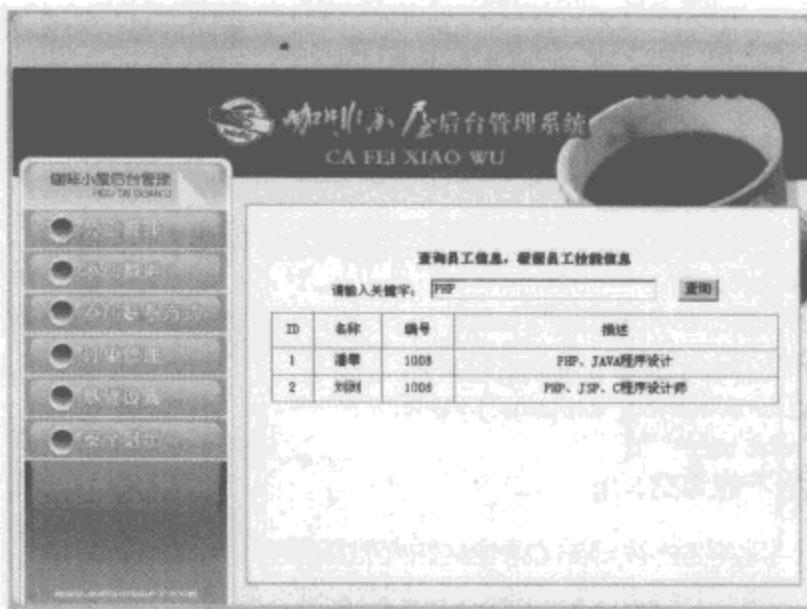


图 23.3 通过 GET 方式与 PHP 进行交互

### 23.3.2 通过 POST 方式与 PHP 进行交互

#### ■ 视频讲解：光盘\TM\Video\23\通过 POST 方式与 PHP 进行交互.exe

通过 POST 方式进行交互主要针对的是 PHP 页面中通过 POST 方法提交的数据。下面的实例是应用 Ajax 技术实现无刷新向数据库中添加数据。

**例 23.3 应用 POST 方式，通过 XMLHttpRequest 对象与 PHP 进行交互。将表单中的数据无刷新添加到指定的数据表中，添加成功后输出数据表中的数据，其操作步骤如下：**（**实例位置：光盘\TM\Instances\23\3**）

(1) 创建 index.php 文件，编写 JavaScript 脚本，并通过 DIV 输出数据库中的数据。在 JavaScript 脚本中，首先定义 Ajax 对象初始化函数 createXmlHttpRequestObject()，然后定义 Ajax 对象处理函数 showSimple()，通过 POST 方式与 searchrst.php 进行交互，最后定义数据处理函数 StatHandler()，数据添加成功后，将数据库中的数据定义到 DIV 标签中，代码如下。

```

<script>
    var xmlhttp; // 定义 XMLHttpRequest 对象
    function createXmlHttpRequestObject(){
        if(window.ActiveXObject){ // 如果在 Internet Explorer 下运行
            try{
                xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
            }catch(e){
                xmlhttp=false;
            }
        }else{ // 如果在 Mozilla 或其他的浏览器下运行
            try{
                xmlhttp=new XMLHttpRequest();
            }catch(e){
                xmlhttp=false;
            }
        }
    }

```

```

if(!xmlHttp) //返回创建的对象或显示错误信息
    alert("返回创建的对象或显示错误信息");
else
    return xmlHttp;
}

function showsimple(){ //创建主控制函数
    createXmlHttpRequestObject();
    var us = document.getElementById("user").value; //获取表单提交的值
    var nu = document.getElementById("number").value;
    var ex = document.getElementById("explains").value;
    if(us=="" && nu=="" && ex=="") //判断表单提交的值（不能为空）
        alert('添加的数据不能为空！');
        return false;
    }

    var post_method="users="+us+"&numbers="+nu+"&explaines="+ex; //构造 URL 参数
    xmlHttp.open("POST","searchrst.php",true); //调用指定的添加文件
    xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded"); //设置请求头信息
    xmlHttp.onreadystatechange=StatHandler; //判断 URL 调用的状态值并处理
    xmlHttp.send(post_method); //将数据发送给服务器
}

function StatHandler(){ //定义处理函数
    if(xmlHttp.readyState==4 && xmlHttp.status==200){ //如果执行成功，则输出下面内容
        if(xmlHttp.responseText!=""){
            alert("数据添加成功！");
            //将服务器返回的数据定义到 DIV 中
            document.getElementById("webpage").innerHTML=xmlHttp.responseText;
        }else{
            alert("添加失败！");
        }
    }
}

</script>
<form id="searchform" name="searchform" method="post" action="#">
    <input name="user" type="text" id="user" size="25" />
    <input type="button" name="Submit" value="提交" onclick="showsimple();"/>
</form>
<td colspan="2" align="center" valign="top"><div id="webpage"></div></td>

```

(2) 创建 searchrst.php 文件，获取 Ajax 中 POST 方法传递的数据，执行 insert 语句，将数据添加到数据表中，添加成功后，查询出数据表中的所有数据，代码如下。

```

<?php
    header('Content-type: text/html;charset=GB2312'); //指定发送数据的编码格式
    include_once 'conn/conn.php'; //连接数据库
    $user = iconv('UTF-8','gb2312',$_POST['users']); //获取 Ajax 传递的值，并实现字符编码转换
    $number = iconv('UTF-8','gb2312',$_POST['numbers']); //获取 Ajax 传递的值，并实现字符编码转换
    $explains = iconv('UTF-8','gb2312',$_POST['explains']); //获取 Ajax 传递的值，并实现字符编码转换
    $sql="insert into tb_administrator(user,number,explains)values('$user','$number','$explains')"; //执行添加语句
    $result=mysql_query($sql,$conn);
    if($result){

```

```
$sqles="select * from tb_administrator ";
$results=mysql_query($sqles,$conn);
echo "<table width='500' border='1' cellpadding='1' cellspacing='1' bordercolor='#FFFFCC' bgcolor=
'#666666>";
echo "<tr><td height='30' align='center' bgcolor='#FFFFFF>ID</td><td align='center' bgcolor='#FFFFFF>名
称</td><td align='center' bgcolor='#FFFFFF>编号</td><td align='center' bgcolor='#FFFFFF>描述</td></tr>";
while($myrow=mysql_fetch_array($results)){ //循环输出查询结果
    echo "<tr><td height='22' bgcolor='#FFFFFF>".$myrow[id]."</td>";
    echo "<td bgcolor='#FFFFFF>".$myrow[user]."</td>";
    echo "<td bgcolor='#FFFFFF>".$myrow[number]."</td>";
    echo "<td bgcolor='#FFFFFF>".$myrow[explains]."</td>";
    echo "</tr>";
}
echo "</table>";
}
```

运行本实例，无刷新添加员工信息后，将在当前页中输出数据表中的所有员工信息，运行结果如图 23.4 所示。

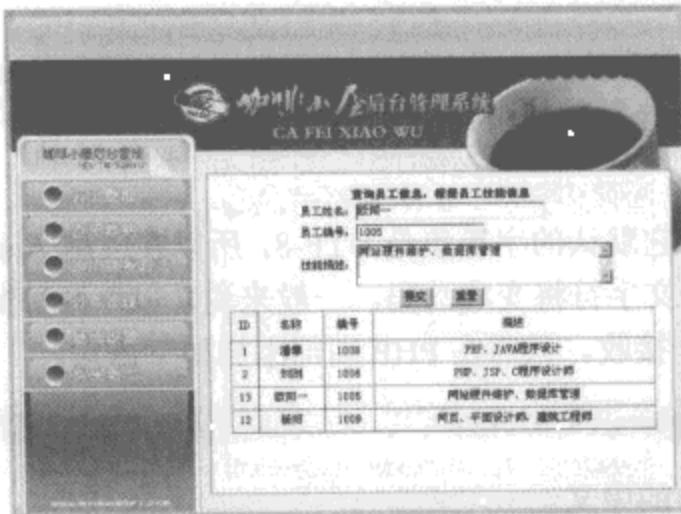


图 23.4 通过 POST 方式与 PHP 进行交互

**注意：**本实例中，在获取 Ajax 中 POST 提交的数据时，需要对数据的编码格式进行转换，才能够将中文字符串添加到数据表中。因为 Ajax 中的数据使用的是 UTF-8 格式的编码，如果要将该数据添加到编码格式为 GB2312 的数据表中，就需要使用 iconv() 函数将 UTF-8 编码转换为 GB2312 编码。

## 23.4 Ajax 开发常见问题

 视频讲解：光盘\TM\Video\23\Ajax 开发需要注意的几个问题.exe

Ajax 在开发过程中需要注意以下几个问题：

### 1. 测量器兼容性问题

Ajax 使用了大量的 JavaScript 和 Ajax 引擎，而这些内容需要浏览器提供足够的支持。目前提供这

些支持的浏览器有 IE 5.0 及以上版本、Mozilla 1.0、NetScape 7 及以上版本。Mozilla 虽然也支持 Ajax，但是提供 XMLHttpRequest 对象的方式不一样。所以，使用 Ajax 的程序必须测试其针对各个浏览器的兼容性。

## 2. XMLHttpRequest 对象封装

Ajax 技术的实现主要依赖于 XMLHttpRequest 对象，但是在调用其进行异步数据传输时，由于 XMLHttpRequest 对象的实例在处理事件完成后就会被销毁，所以如果不对该对象进行封装处理，在下次需要调用它时就得重新构建，而且每次调用都需要写一大段的代码，使用起来很不方便。不过，现在很多开源的 Ajax 框架都提供了对 XMLHttpRequest 对象的封装方案，其详细内容这里不作介绍，读者可参考相关资料。

## 3. 性能问题

由于 Ajax 将大量的计算从服务器移到了客户端，这就意味着浏览器将承受更大的负担，而不再是只负责简单的文档显示。由于 Ajax 的核心语言是 JavaScript，而 JavaScript 并不是以高性能而知名，另外，JavaScript 对象也不是轻量级的，特别是 DOM 元素耗费了大量的内存，因此，如何提高 JavaScript 代码的性能对于 Ajax 开发者来说尤为重要。下面介绍 3 种优化 Ajax 应用执行速度的方法：

- 优化 for 循环。
- 将 DOM 节点附加到文档上。
- 尽量减少点“.”号操作符的使用。

## 4. 中文编码问题

Ajax 不支持多种字符集，它默认的字符集是 UTF-8，所以在应用 Ajax 技术的程序中应及时进行编码转换，否则程序中出现的中文字符将变成乱码。一般来说，以下两种情况会产生中文乱码。

- PHP 发送中文、Ajax 接收。只需在 PHP 顶部添加如下语句。

```
header('Content-type: text/html; charset=GB2312'); //指定发送数据的编码格式
```

XMLHttp 会正确解析其中的中文。

- Ajax 发送中文、PHP 接收。这个比较复杂，Ajax 中先用 encodeURIComponent 对要提交的中文进行编码，在 PHP 页中添加如下代码。

```
$GB2312string=iconv('UTF-8', 'gb2312//IGNORE', $RequestAjaxString);
```

PHP 选择 MySQL 数据库时，应用如下语句设置数据库的编码类型。

```
mysql_query("set names gb2312");
```

## 23.5 实战

Ajax 作为 Web 2.0 时代的关键技术，已经被广泛地应用，并且很多组织已经推出基于 Ajax 的框架，如 jQuery 就是非常快捷、方便的 Ajax 框架，使用 jQuery 技术可以用简单的几行代码完成非常复杂的 Ajax 交互。本节将通过具体实例讲解 Ajax 技术的应用。

### 23.5.1 在 PHP 中应用 Ajax 技术检测用户名

视频讲解：光盘\TM\Video\23\在 PHP 中应用 Ajax 技术检测用户名.exe

用户注册页面中，经常会使用 Ajax 技术对用户录入信息进行校验，这样不仅可以制作出各种动态效果，而且可以降低服务器的负载。

例 23.4 用户注册表单中，使用 Ajax 技术检测用户名是否被占用。（实例位置：光盘\TM\Instances\23\4）

本实例的实现步骤如下：

(1) 建立 fun.js 脚本文件，该文件中的代码用于检测使用 Ajax 技术通过 GET 方法向 chk.php 文件中发送注册表单中在用户名文本框中所录入的用户名，并根据返回值判断该用户名是否被其他用户占用。代码如下。

```
function chkUsername(username){
    if(username==""){
        //判断用户名是否为空
        alert('请输入用户名！');
    }else{
        var xmlObj; //定义 XMLHttpRequest 对象
        if(window.ActiveXObject){ //如果是浏览器支持 ActiveXObject 则创建 ActiveXObject 对象
            xmlObj = new ActiveXObject("Microsoft.XMLHTTP");
        }else if(window.XMLHttpRequest){ //如果浏览器支持 XMLHttpRequest 对象则创建 XMLHttpRequest 对象
            xmlObj = new XMLHttpRequest();
        }
        xmlObj.onreadystatechange = callBackFun; //指定回调函数
        xmlObj.open('GET', 'chk.php?username=' + username, true);
        //使用 GET 方法调用 chk.php 并传递 username 参数的值
        xmlObj.send(null); //不发送任何数据，因为数据已经使用请求 URL 通过 GET 方法发送
        function callBackFun(){ //回调函数
            if(xmlObj.readyState == 4 && xmlObj.status == 200){ //如果服务器已经传回信息并未发生错误
                if(xmlObj.responseText=='y'){ //如果服务器传回的内容为 y，则表示用户名已经被占用
                    alert('该用户名已被他人使用！');
                }else{ //不为 y，则表明用户名未被占用
                    alert('恭喜，该用户名未被使用！');
                }
            }
        }
    }
}
```

上述代码中，首先判断表单中用户名文本框的值是否为空，如果为空则弹出提示对话框要求用户输入用户名，然后判断浏览器所支持的组件类型创建 XMLHttpRequest 对象，最后创建回调函数 callBackFun()，该函数用于获取 chk.php 文件输出的内容，并根据该内容提示用户所录入的用户名是否被占用。

(2) 建立一个基本的用户注册表单，为了说明问题，在表单中只包含用户名录入文本框和用户名

是否为空检测按钮，代码如下。

```
<form name="form_register">
    用户名: <input type="text" id="username" name="username" size="20" />&nbsp;
    <input type="button" value="查看用户名是否被占用"
        onclick="javascript:chkUsername(form_register.username.value)" />
</form>
```

### (3) 建立与 MySQL 数据库的连接，选择数据库并设置字符集。

```
<?php
$host = '127.0.0.1'; //MySQL 数据库服务器地址
$userName = 'root'; //用户名
$password = 'root'; //密码
$connID = mysql_connect($host, $userName, $password); //建立与数据库的连接
mysql_select_db('db_database23', $connID); //选择数据库
mysql_query('set names gbk'); //设置字符集
?>
```

(4) 建立 chk.php 文件，该文件中的代码用于判断客户端通过 GET 方法提交的用户名的值，并判断该值是否存在，如果存在则返回“y”，否则返回“n”。

```
<?php
require_once 'conn.php'; //包含数据库连接文件
$sql = mysql_query("select id, username from tb_user where username='".trim($_GET['username'])."'", $connID);
//执行查询
$result = mysql_fetch_array($sql);
if ($result) { //判断用户名是否存在
    echo 'y';
} else {
    echo 'n';
}
?>
```

运行上述实例，结果如图 23.5 所示，在用户名文本框中随便输入一个用户名，然后单击“查看用户名是否被占用”按钮，在不刷新页面的情况下即可弹出该用户名是否被占用的提示信息。

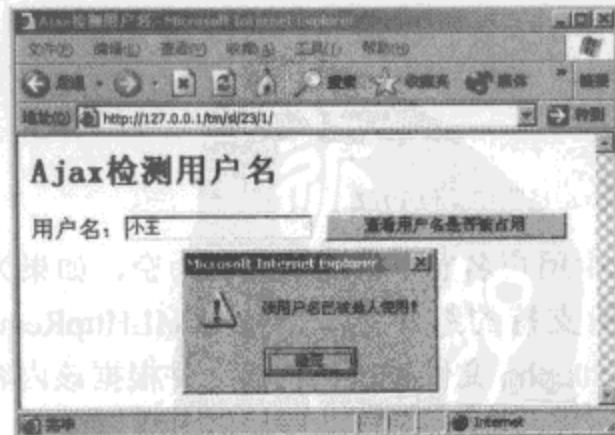


图 23.5 检测用户名是否被占用

### 23.5.2 在 PHP 中应用 Ajax 技术实现博客文章类别添加

#### ■ 视频讲解：光盘\TM\Video\23\在 PHP 中应用 Ajax 技术实现博客文章类别添加.exe

传统方式中，通过 POST 方法提交表单中的数据，页面需要刷新，而使用 Ajax 技术通过 POST 方法提交数据，不需要刷新页面即可实现与服务器的交互，这样可以有效减少刷新页面的等待时间。

**例 23.5 使用 Ajax 技术通过 POST 方法添加文章类别。（实例位置：光盘\TM\Instances\23\5）**

本实例的实现步骤如下：

(1) 建立类别信息录入表单。与传统的表单相比，使用 Ajax 技术通过 POST 方法提交数据时，不需要在客户端表单中使用<form>标记，数据类型通过 XMLHttpRequest 对象的 setRequestHeader()方法设定。类别录入表单结构如表 23.2 所示。

表 23.2 类别录入表单的表单元素

元素类型	元素名称	属性设置	说明
文本域	typename	type="text" name="typename" size="30"	博客类别
文本域	username	type="text" name="username" size="30"	发布人
提交按钮	button	type="button" value="保存" onclick="saveType(typename.value, username.value)"	提交按钮

(2) 建立 fun.js 脚本文件，在该文件中编写用于将客户端录入的数据通过 POST 方法提交到服务器的 Ajax 代码，如下所示。

```
function saveType(typename, username){
    if(typename==""){
        //判断博客类别是否为空
        alert('请输入类别名称！');
    }else if(username==""){
        //判断用户名是否为空
        alert('请输入用户名！');
    }else{
        var xmlObj;
        //定义 XMLHttpRequest 对象
        var urlData = "typename="+typename+"&username="+username;
        //指定要发送的数据
        if(window.ActiveXObject){
            //判断浏览器是否支持 ActiveXObject 组件，如果支持则通过 ActiveXObject 方式创建 XMLHttpRequest 对象
            xmlObj = new ActiveXObject("Microsoft.XMLHTTP");
        }else if(window.XMLHttpRequest){
            //如果浏览器不支持 ActiveXObject 组件，则使用 XMLHttpRequest 组件创建 XMLHttpRequest 对象
            xmlObj = new XMLHttpRequest();
        }
        xmlObj.onreadystatechange = callBackFun;
        //指定回调函数
        xmlObj.open("POST", "saveType.php", true);
        //指定提交方法和页面
        xmlObj.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        //指定发送数据库类型
        xmlObj.send(urlData);
        //指定发送的数据
        function callBackFun(){
            //定义回调函数
            if(xmlObj.readyState == 4 && xmlObj.status == 200){
                //如果服务器端返回内容并无错误发生
            }
        }
    }
}
```

```

        if(xmlObj.responseText == 'y'){           //判断保存是否成功，并给出提示
            alert('类别添加成功！');
        }else{
            alert('类别添加失败！');
        }
    }
}
}

```

上述代码中，首先判断是否已经录入了博客类别信息，并给出相关提示，之后定义并创建 XMLHttpRequest 对象，使用该对象的 onreadystatechange 属性指定回调函数，open 方法指定通过 POST 方法提交以及提交页面，setRequestHeader()方法指定发送数据类型，send()方法指定要发送的数据。在回调函数中根据返回参数判断类别是否保存成功，并通过提示窗口给出相应的提示信息。

(3) 编写 conn.php 文件，该文件中的代码用于实现与 MySQL 数据库的连接，并设置字符集，代码如下。

```

<?php
$host = '127.0.0.1';                                //MySQL 数据库服务器地址
$username = 'root';                                  //用户名
$password = 'root';                                  //密码
$connID = mysql_connect($host, $username, $password); //建立与数据库的连接
mysql_select_db('db_database23', $connID);           //选择数据库
mysql_query('set names gbk');                         //设置字符集
?>

```

为了防止数据库的中文内容发送乱码，在完成数据选择后通过 mysql\_query() 函数执行 SQL 命令，设置数据所使用的字符集为 GBK 编码。

(4) 接收通过 POST 方法提交的内容，并将博客类别等相关内容保存到数据库中。

```

<?php
require_once 'conn.php';                            //包含数据库连接文件
$typename = iconv('utf-8', 'gbk', $_POST['typename']); //将类别名称由 UTF-8 编码转变为 GBK 编码
$username = iconv('utf-8', 'gbk', $_POST['username']); //将用户名由 UTF-8 编码转变为 GBK 编码
if(mysql_query("insert into tb_type(typename, username, addtime) values(\".$typename.\", \".$username.\", \"".date('Y-m-d H:i:s')."\"), $connID)){ //将数据库保存到数据库中
    echo 'y';
} else{
    echo 'n';
}
?>

```

上述代码中，使用 require\_once 语句包含 conn.php 文件从而建立与数据库的连接，由于 Ajax 在通过 POST 方法提交数据时默认采用 UTF-8 编码，所以在接收提交的相关信息后需要使用 iconv() 函数将所接收的内容由 UTF-8 编码转换为 GBK 编码，完成转码后使用 mysql\_query() 函数将类别信息保存到数据库中，如果成功保存则输出 y，否则输出 n。

运行本实例，结果如图 23.6 所示，在“请输入博客类别”和“发布人”文本框中填写相关信息后，

单击“保存”按钮即可将类别信息保存到数据库中，单击“保存”按钮时可以发现没有刷新页面就弹出类别添加成功对话框。这时手工刷新页面可以在图中左侧导航栏中查看到新添加的类别名称。

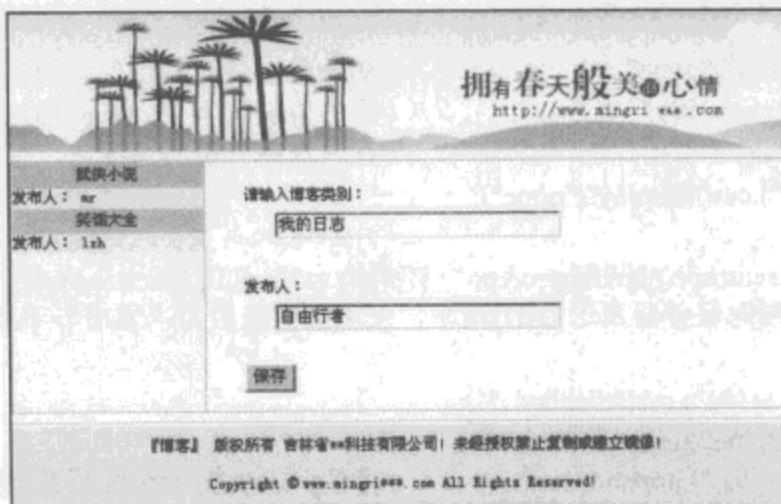


图 23.6 保存博客类别信息

### 23.5.3 使用 jQuery 制作级联下拉列表框

#### 视频讲解：光盘\TM\Video\23\使用 jQuery 制作级联下拉列表框选择地区.exe

直接使用 Ajax 实现与服务器端的请求代码编写相对繁琐，为了简化 Ajax 操作，很多组织都推出了基于 Ajax 的框架，jQuery 就是其中的代表之一。自从 jQuery 面世以来，就以其快速、简洁，能够很轻易地处理 HTML 文档、控制事件、给页面添加动画和 Ajax 效果等特点得到了广泛的应用。

**例 23.6 使用 jQuery 制作级联下拉列表框选择地区。（实例位置：光盘\TM\Instances\23\6）**

本实例的实现步骤如下：

(1) 建立 pc.php 文件，在该文件中分别定义省份数组\$p 和城市数组\$c，其中\$p 为一维数组，\$c 为二维数组，并要求城市数组\$c 的第一维索引下标与对应的省份数组\$p 的下标相同，这样在编写代码时只需根据省份数组的下标就可以获取该省份中的所有城市。省份数组和城市数组的具体定义方式详见本书附带光盘。

(2) 定义“省份”下拉列表框和“城市”下拉列表框，代码如下。

```
<select id="p" name="p">
    <option value="">-请选择地区-</option>
</select>
<select id="c" name="c">
</select>
```

上述代码中，在定义“省份”和“城市”下拉列表框时，并没有指定下拉选项，这是因为在编写程序时可以通过 jQuery 框架动态指定下拉列表框的内容。

(3) 使用 jQuery 框架编写代码实现省份列表的显示，当改变省份列表的省份名称时，在“城市”下拉列表框中显示该省份所有城市的名称。

```
$(document).ready(
    function() {
```

```

$.get("returnpc.php?flag=p", null, function(data){
    //向服务器发送 GET 请求，获取省份的值，并将结果追加到“省份”下拉列表框中
    $("#p").append(data);
});
$("#c").css("display","none");           //初始状态使“城市”下拉列表框不可见
$("#p").change(function(){
    if($("#p").val()==""){
        $("#c").css("display","none"); //在未选择省份的情况下，使“城市”下拉列表框不可见
    }else{
        $.get("returnpc.php?flag=c&p="+$("#p").val(), null, function(data){
            //如果选择了某省份，则向服务器发送 GET 请求，使用回调函数为“城市”下拉列表框赋值，并使“城市”下拉列表框可见
            $("#c").css("display","");
            $("#c").empty();
            $("#c").append(data);      //将数据追加到“城市”下拉列表框
        });
    }
});
);

```

上述代码中，主要使用 Ajax 的 jQuery 框架，没有 jQuery 框架基础的读者可能不容易理解上述代码，下面对该内容进行介绍。

`$(document).ready()` 表示在页面载入时所执行的操作，是 `window.load` 注册事件的替代方法。在 `ready` 方法中使用一个匿名函数作为参数，所有初始操作都在该匿名函数中定义。

`$.get()` 方法用于向服务器发送 GET 请求，该方法的语法格式如下：

```
$.get(url, data, callback);
```

其中，参数 `url` 为待载入页面的 URL 地址，`data` 为要向服务器端发送的数据，`callback` 为载入成功时的回调函数。可以发现，如果直接使用 Ajax 向服务器发送 GET 请求，需要编写多行代码才能实现，而使用 jQuery 框架只需一个方法即可实现，所以 jQuery 确实为程序开发带来了极大的方便。在回调函数中定义如下代码：

```
$("#p").append(data)
```

其中，`$("#p")` 用来获取 HTML 页面中 ID 为 `p` 的元素的引用，相当于 `document.getElementById("p")`，`append()` 方法用于向“省份”下拉列表框中添加列表项。

`$("#p").change()` 方法为“省份”下拉列表框的 `change` 事件，当选择下拉列表框中的选项时触发该事件。在“省份”下拉列表框的 `change` 事件中向服务器发送 GET 请求，并在回调函数中将服务器返回的值追加到“城市”下拉列表框中。

(4) 编写用于返回客户端请求的服务器端代码，根据客户端传递参数 `flag` 的值判断返回省份信息还是城市信息，之后从省份数组或城市数组中提取要返回的内容并返回。

```

<?php
require_once 'pc.php';
$flag = $_GET['flag'];
//包含省份数组和城市数组
//用于区分是返回省份信息还是返回城市信息

```

```

$str="";
if($flag == p){ //如果参数 flag 的值为 p, 则返回省份信息
    for($i=0; $i<count($p); $i++){
        //通过循环整理所有要追加到“省份”下拉列表框的内容
        $str.= "<option value=\"".iconv('gbk','utf-8',$p[$i])."\">>".iconv('gbk','utf-8',$p[$i])."</option>";
    }
    echo $str; //输出要返回的参数
} else{
    $index = array_search($_GET['p'], $p); //提取所选择的省份在省份数组中对应的键值
    for($j=0; $j<count($c[$index]); $j++){ //通过循环整理所有要追加到“城市”下拉列表框的内容
        $str.= "<option value=\"".iconv('gbk','utf-8',$c[$index][$j])."\">>".iconv('gbk','utf-8',$c[$index][$j])."</option>";
    }
    echo $str; //输出要返回的参数
}
?>

```

 注意：为了防止返回的信息乱码，需要使用 iconv() 函数将 GBK 编码转换为 UTF-8 编码。

运行上述实例，结果如图 23.7 所示。默认“城市”下拉列表框不可见，当选择某省份时会显示“城市”下拉列表框，并将该省份所包括的城市显示在“城市”下拉列表框中。

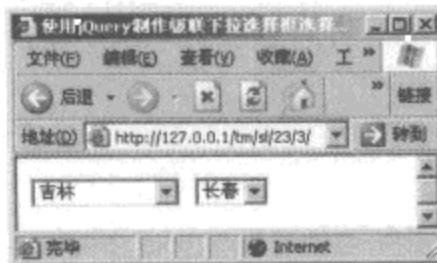


图 23.7 选择地区

## 23.6 本章小结

本章从 Ajax 技术的基本概念开始，采用循序渐进的方式介绍了 Ajax 技术的开发模式、特性及其组成部分，并重点介绍了 Ajax 技术的核心 XMLHttpRequest 对象，以及编写基本的 Ajax 程序的步骤和思路，最后通过具体实例巩固本章所介绍的内容，尤其在本章最后使用一个简单的实例介绍了基于 Ajax 技术的 jQuery 框架的使用方法。通过本章的学习，读者可以理解 Ajax 技术的概念和基本 Ajax 应用的制作方法，能够在此基础上进一步深入学习 Ajax 技术尤其是 jQuery 框架的使用方法。

## 23.7 实战练习

- 向服务器上传文件或从服务器下载文件时，经常会等待很长时间，为了显示上传或下载进度，可以使用进度条来显示。要求用 jQuery 框架制作如图 23.8 所示的进度条。（答案位置：光盘\TM\Instances\23\7）

- 采用图表的形式显示大量的统计数据非常直观，以往在页面中显示的图表都是静态的，如果将

图表制作成动态效果，不仅直观，而且吸引人。要求制作出如图 23.9 所示的动态折线图，按月份显示出完整的折线图。（答案位置：光盘\TM\Instances\23\8）

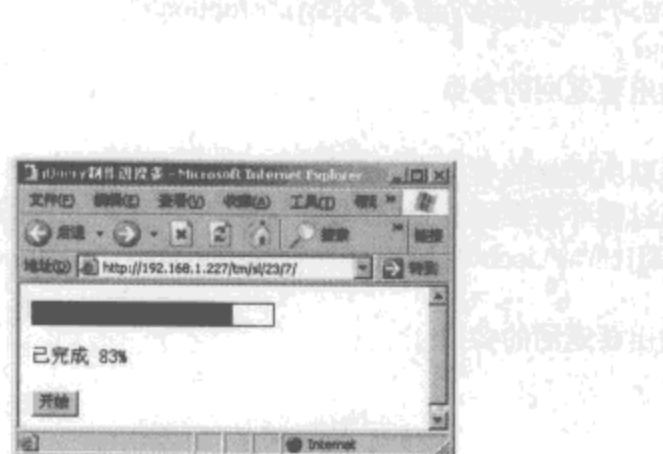


图 23.8 用 jQuery 制作进度条

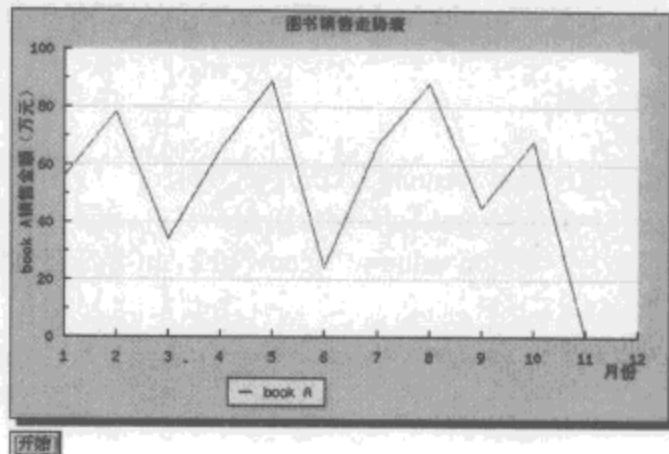


图 23.9 用 jQuery 制作动态折线图

# 第24章

## Smarty 模板技术

( 视频讲解: 102分钟)

目前网络上针对 PHP 的模板数不胜数。作为最早的 MVC 模板之一，Smarty 在功能和速度上处于绝对的领先地位。那么，Smarty 的特点是什么？它是如何完成代码分离的？

通过阅读本章，可以学习到：

- ▶▶ 什么是 Smarty 模板引擎
  - ▶▶ Smarty 的开发模式与运行流程
  - ▶▶ Smarty 模板的安装和配置
  - ▶▶ Smarty 模板设计
  - ▶▶ Smarty 程序设计
  - ▶▶ 如何灵活运行 Smarty

## 24.1 Smarty 简介

Smarty 是 PHP 中的一个模板引擎，是众多 PHP 模板中最优秀、最著名的模板之一。那么，什么是 Smarty？它是基于什么框架的？它有哪些特点？本节将一一回答这些问题。

### 24.1.1 Smarty 模板引擎

当动态网页开始风行时，网站设计师们对 HTML 中嵌入脚本语言又提出了更高的要求，对这种方式很不满意。因为无论是微软的 ASP 还是开放源码的 PHP，都属于服务器端的 HTML 嵌入式脚本语言，作为一个网站设计师必须既懂程序设计，又懂页面设计。但在通常情况下，如果程序设计是高手，那么在页面设计上必然差一些，反之也是如此，正所谓“鱼和熊掌不可兼得”。由此有网站设计师设想，如果能将网页设计与程序开发分离，效果是否会更好呢？因此模板引擎应运而生。

开发模板引擎的目的是实现程序设计与网页设计的逻辑分离，使程序设计者可以专注于程序功能的开放；而网页设计师专注于页面的设计，让网页看起来更加具有专业性。

Smarty 是一个使用 PHP 编写的应用于 PHP 的模板引擎，它将一个应用程序分成两部分：视图和逻辑控制。简单地讲，就是将 UI（用户界面）和 PHP code（PHP 代码）分离。这样，程序员在修改程序时不会影响到页面设计，而美工在重新设计或修改页面时也不会影响程序逻辑。

### 24.1.2 开发模式与运行流程

Smarty 的这种开发模式是基于 MVC 框架的概念。MVC（Model-View-Controller），即模型-视图-控制器。MVC 框架将一个应用程序定义成视图、控制器、模型 3 部分。

- 视图：就是提供给用户的界面。视图只提供信息的收集及显示，不涉及处理。如用户登录，登录界面就是视图，只提供用户登录的用户名和密码输入框（也可以有验证码、安全问题等信息），至于用户名和密码的对与错，这里不去处理，直接传给后面的控制部分。
- 控制器：负责处理视图和模型的对应关系，并将视图收集的信息传递给对应的模型。例如，当用户输入用户名和密码后提交。这时，控制部分接收用户的提交信息，并判断这是一个登录操作，随后将提交信息转发给登录模块部分，也就是模型。
- 模型：对接收过来的信息进行处理，并将处理结果回传给视图。例如，如果用户输入信息正确，那么将给视图一个命令，允许用户进入主页面；反之，则拒绝用户的操作。

Smarty 模板引擎将 PHP 程序直接生成模板文件，最终浏览器中读取的就是 Smarty 模板文件，并且 Smarty 能够对模板文件进行判断，如果是第一次或者模板已经改变，则重新生成模板文件，否则将直接执行原模板文件。Smarty 模板引擎的运行流程如图 24.1 所示。

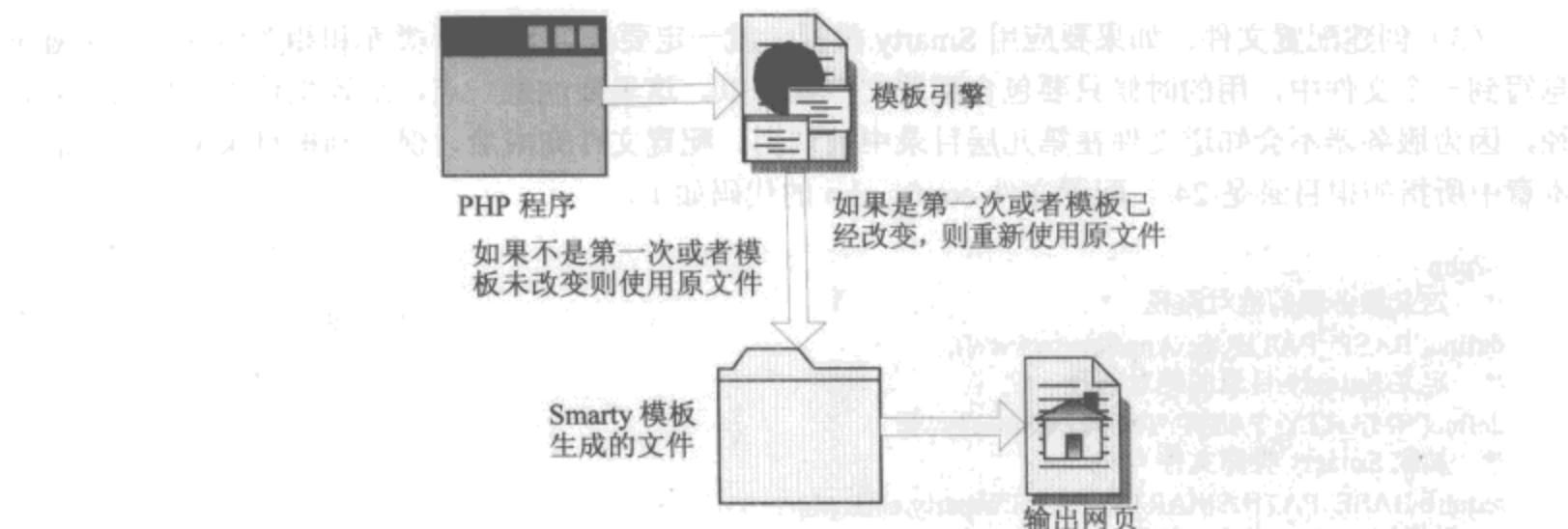


图 24.1 Smarty 模板引擎运行流程图

## 24.2 Smarty 的安装配置

视频讲解：光盘\TM\Video\24\Smarty 的安装配置.exe

### 24.2.1 Smarty 下载和安装

PHP 没有内置 Smarty 模板类，需要单独下载和配置，并且 Smarty 要求服务器上的 PHP 版本最低为 4.0.6。用户可以通过 <http://smarty.php.net/download.php> 网站下载最新的 Smarty 压缩包。本章使用的版本是 Smarty-2.6.19。

将压缩包解压后，有一个 libs 目录，其中包含了 Smarty 类库的核心文件，包括 smarty.class.php、smarty\_Compiler.class.php、config\_File.class.php 和 debug.tpl 4 个文件，还有 internals 和 plug-ins 两个目录。将 libs 目录复制到服务器根目录下，并为其重新命名。一般该目录的名称为 smarty、class 等，这里将 libs 文件夹重新命名为 Smarty。至此，Smarty 模板安装完毕。

注意：凡是在后面的章节中提到 Smarty 类包、Smarty 目录等，都是这个重新命名后的 Smarty 目录，即原 libs 目录。

### 24.2.2 Smarty 配置

出色地完成 Smarty 的配置工作，是应用 Smarty 模板引擎的关键。

(1) 确定 Smarty 目录的位置。因为 Smarty 类库是通用的，每一个项目都可能会使用到它。所以将 Smarty 放到根目录下。因为本章的所有程序都放在 tm/sl/24/文件夹下，所以将 /24/ 作为临时的根目录，Smarty 就放到这个目录下。

(2) 新建 4 个目录 templates、templates\_c、configs 和 cache。因为目录 templates 存放的是项目的模板，所以有人喜欢将 templates 放到 Smarty 目录外。这两种方法没什么区别，只要设置的路径正确即可。

(3) 创建配置文件。如果要应用 Smarty 模板，就一定要包含 Smarty 类库和相关信息。将配置信息写到一个文件中，用的时候只要包含配置文件就可以。这里要注意一点，配置文件中要使用绝对路径，因为服务器不会知道文件在第几层目录中被调用。配置文件完成后，保存到根目录下。不要忘记本章中所指的根目录是 /24/。配置文件 config.php 的代码如下。

```
<?php
/* 定义服务器的绝对路径 */
define('BASE_PATH','F:\AppServ\www\\');
/* 定义 Smarty 目录的绝对路径 */
define('SMARTY_PATH','TM\sl24\Smarty\\');
/* 加载 Smarty 类库文件 */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/* 实例化一个 Smarty 对象 */
$smarty = new Smarty;
/* 定义各个目录的路径 */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';
/* 定义定界符 */
//$smarty->left_delimiter = '{';
//$smarty->right_delimiter = '}';
?>
```

上述配置文件的参数说明如下。

- BASE\_PATH:** 指定服务器的绝对路径。

 **说明：**指定服务器绝对路径的目的是找到 Smarty 文件夹在服务器中的存储位置。这里有两种方法可以使用：第一种，直接指定绝对路径，如 F:\AppServ\www\\，使用这种方法来指定服务器的绝对路径，一旦服务器的绝对路径发生更改，就必须要修改配置文件，否则程序就会运行出错；第二种，通过全局变量\$\_SERVER['DOCUMENT\_ROOT']来获取服务器的绝对路径，使用该方法不会因为服务器路径的更改而影响到程序的执行。推荐使用第二种方法定义服务器的绝对路径。

- SMARTY\_PATH:** 指定 Smarty 目录的绝对路径。
- require():** 加载 Smarty 类库文件 Smarty.class.php。
- \$smarty:** 实例化 Smarty 对象。
- \$smarty->template\_dir:** 定义模板目录存储位置。
- \$smarty->compile\_dir:** 定义编译目录存储位置。
- \$smarty->config\_dir:** 定义 configs 文件存储位置。
- \$smarty->cache\_dir:** 定义模板缓存目录。
- \$smarty->left\_delimiter:** 定义 Smarty 使用的开始定界符。
- \$smarty->right\_delimiter:** 定义 Smarty 使用的结束定界符。

有关定界符的使用，开发者可以指定任意的格式。也可以不指定定界符，使用 Smarty 默认的定界符 { 和 }。

符“{”和“}”。

到此，Smarty 的配置讲解完毕。

### 24.2.3 第一个 Smarty 程序

#### 视频讲解：光盘\TM\Video\24\第一个 Smarty 程序.exe

介绍 Smarty 的下载、安装和配置方法后，接下来就创建一个 Smarty 程序，通过具体的操作来了解 Smarty 的应用。

**例 24.1 创建第一个 Smarty 应用实例，初步了解 Smarty 的使用过程。(实例位置：光盘\TM\Instances\24\1)**

(1) 在服务器根目录下的/tm/sl/24/文件夹下，新建一个文件夹，命名为 1。

(2) 复制 Smarty 到目录 1 下。在 Smarty 目录下新建 4 个目录，分别是 templates、templates\_c、configs 和 cache。此时，例 24.1 的目录结构如图 24.2 所示。

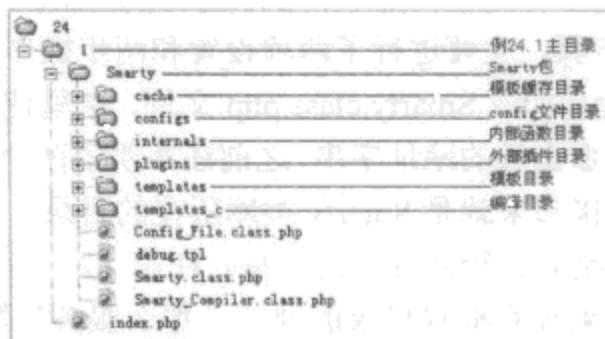


图 24.2 Smarty 包的目录结构

(3) 新建一个.html 静态页，输入数据。输入完毕后将文件保存到刚刚新建的 templates 目录下，并命名为 index.tpl。实例代码如下。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
</head>
<body>
    {$content}
</body>
</html>
```

**说明：**代码中加粗的部分就是 Smarty 标签，大括号“{}”为标签的定界符，\$title 和\$content 为变量。

(4) 回到上级目录，在目录/1/下新建一个.php 文件，使用 Smarty 变量和方法对文件进行操作，输入完毕后保存为 index.php，代码如下。

```
<?php
/* 定义服务器的绝对路径 */
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);
```

```

/* 定义 Smarty 目录的绝对路径 */
define('SMARTY_PATH','\tm\sl\24\1\Smarty\\');
/* 加载 Smarty 类库文件 */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/* 实例化一个 Smarty 对象 */
$smarty = new Smarty;
/* 定义各个目录的路径 */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs/';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';
/* 使用 Smarty 赋值方法将指定数据发送到模板中 */
$smarty->assign('title','第一个 Smarty 程序');
$smarty->assign('content','Hello,Welcome to study \'Smarty\'!');
/* 显示模板 */
$smarty->display('index.tpl');
?>

```

这是 Smarty 运行最关键的步骤，主要进行了两项设置和两步操作。

- 加载 Smarty 类库。也就是加载 Smarty.class.php 文件，这里使用的是绝对地址。为了稍后在配置其他路径时不用输入那么长的地址字串，之前还声明了两个常量：服务器地址常量和 Smarty 路径常量。两个常量连接起来就是 Smarty 类库所在的目录。
- 保存新建的 4 个目录的绝对路径到各自的变量。在第（2）步时，曾创建了 4 个目录。这 4 个目录各有各的用途，如果没有配置目录的地址，那么服务器默认的路径就是当前执行文件所在的路径。除了上面两项必须设置的变量外，还可以改变很多 Smarty 参数值，如开启/关闭缓存、改变 Smarty 的默认定界符等，这些变量将在 24.4.2 节中介绍。
- 给模板赋值。设置成功后，需要给指定的模板赋值。assign()就是赋值方法。
- 显示模板。一切操作结束后，调用 display()方法来显示页面。实际上，用户真正看到的页面是 templates 模板目录下的 index.tpl 模板文件；作为首页的 index.php，只是用来传递结果和显示模板；而在浏览器中运行的是 templates\_c 文件夹下的文件。

打开 IE 浏览器，运行 index.php 文件。运行结果如图 24.3 所示。

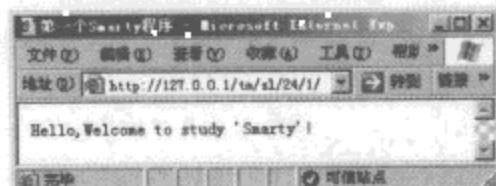


图 24.3 第一个 Smarty 程序

### 24.3 Smarty 模板设计

#### 视频讲解：光盘\TM\Video\24\Smarty 模板设计.exe

Smarty 的特点是将用户界面和过程实现分离，让美工和程序员各司其职，互不干扰。这样，Smarty

类库也自然被分成两部分来使用，即 Smarty 模板设计和 Smarty 程序设计。两部分内容既相互独立，同时也有一部分共通。本节首先来学习 Smarty 模板设计。

### 24.3.1 Smarty 模板文件

Smarty 模板文件是由一个页面中所有的静态元素，加上定界符“{…}”组成的。模板文件统一存放的位置是 templates 目录下（模板文件的存储位置可以在配置文件中指定，可以根据个人习惯而定）。模板中不允许出现 PHP 代码段。Smarty 模板中的所有注释、变量、函数等都要包含在定界符内。

### 24.3.2 注释

Smarty 中的注释和 PHP 注释类似，都不会显示在源代码中。注释包含在两个星号“\*”中间，格式如下：

```
/* 这是注释 */
```

### 24.3.3 变量

Smarty 中的变量来自以下 3 个部分：

- 来自 PHP 页面中的变量，也就是 assign() 方法传过来的变量。

使用方法和在 PHP 中是相同的，也需要使用“\$”符号。略有不同的是对数组的读取。Smarty 中读取数组有两种方法。一种是通过索引获取，与 PHP 中相似，可以是一维，也可以是多维；另一种是通过键值获取数组元素，这种方法的格式与以前接触过的不同，使用符号“.”作为连接符，数组 \$arr = array {'object' => 'book', 'type' => 'comptuer', 'unit' => '本'}，如果想得到 type 的值，表达式的格式为 \$arr.type。这个格式同样适用于二维数组。

例 24.2 使用上述两种方法读取数组值，代码如下：（实例位置：光盘\TM\Instances\24\2）

```
Smarty/templates/2/index.tpl 文件
<html>
<head>
/* 页面的标题变量$title */
<title>{$title}</title>
</head>
<body>
购书信息：<p>
/* 使用索引取得数组的第一个元素值 */
图书类别：{$arr[0]}<br />
/* 使用键值取得第二个数组元素值 */
图书名称：{$arr.name}<br />
/* 使用键值取得二维数组的元素值 */
图书单价：{$arr.unit_price.price}/{ $arr.unit_price.unit }
</body>
```

```
</html>
index.php 文件
<?php
    include_once("../config.php"); //调用 Smarty 类库文件
    $arr = array('computerbook','name' => 'PHP 从入门到精通','unit_price' => array('price' => '¥65.00','unit' => '本'));
    $smarty->assign('title','使用 Smarty 读取数组');
    $smarty->assign('arr',$arr);
    $smarty->display('2/index.tpl');
?>
```

运行结果如图 24.4 所示。

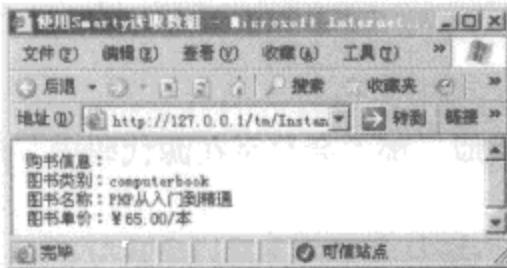


图 24.4 使用 Smarty 读取数组

保留变量，相当于 PHP 中的预定义变量。

在 Smarty 模板中使用保留变量时，无须使用 assign()方法传值，直接调用变量名即可。Smarty 中常用的保留变量如表 24.1 所示。

表 24.1 Smarty 中常用的保留变量

保留变量名	说 明
get、post、server、session、cookie、request	等价于 PHP 中的\$_GET、\$_POST、\$_SEVER、\$_COOKIE、\$_REQUEST
now	当前的时间戳等价于 PHP 中的 time()
const	用 const 包含修饰的为常量
config	配置文件内容变量。参见例 24.4

例 24.3 在模板文件中输出一些保留变量的值，代码如下：（实例位置：光盘\TM\Instances\24\3）

```
templates/3/index.tpl 文件
{* 设置标题名称 *}
<title>{ $title }</title>
<body>
{* 使用 get 变量获取 url 中的变量值(ex: http://localhost/tm/s1/26/3/index.php?type=computer) *}
变量 type 的值是: { $smarty.get.type }<br />
当前路径为: { $smarty.server.PHP_SELF }<br />
当前时间为: { $smarty.now }
</body>
index.php 文件
<?php
    include '../config.php'; //载入配置文件
    $smarty->assign('title','Smarty 保留变量'); //向模板中赋值
    $smarty->display('3/index.tpl'); //显示指定模板
?>
```

运行结果如图 24.5 所示。

从配置文件中读取数据。Smarty 模板也可以通过配置文件来赋值。对于 PHP 开发人员来说，对配置文件的使用从安装服务器就开始了，对文件的格式也有了一个初步的了解。调用配置文件中变量的格式有以下两种：

- 使用“#”号。将变量名置于两个“#”号中间，即可像普通变量一样调用配置文件内容。
- 使用保留变量中的\$smarty\_config 来调用配置文件。

**例 24.4 通过上面两种方法来调用配置文件 4.conf 的内容，代码如下：（实例位置：光盘\TM\Instances\24\4）**

```
configs/4/4.conf 文件
title = "调用配置文件"
bgcolor = "#f0f0f0"
border = "5"
type = "计算机类"
name = "PHP 从入门到精通"
templates/04/index.tpl 文件
{ config_load file="4/4.conf" }
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title#}</title>
</head>
<body bgcolor="#{$bgcolor#}">
<table border="#{$border#}">
<tr>
    <td>{$smarty.config.type}</td>
    <td>{$smarty.config.name}</td>
</tr>
</table>
</body>
</html>
index.php 文件
<?php
    include_once './config.php';
    $smarty->display('4/index.tpl');
?>
```

运行结果如图 24.6 所示。

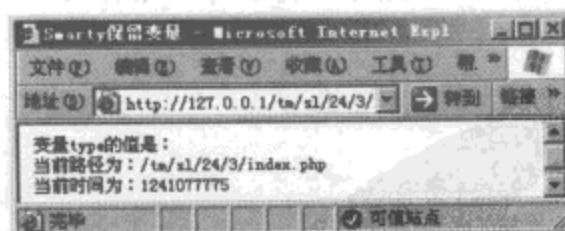


图 24.5 Smarty 保留变量

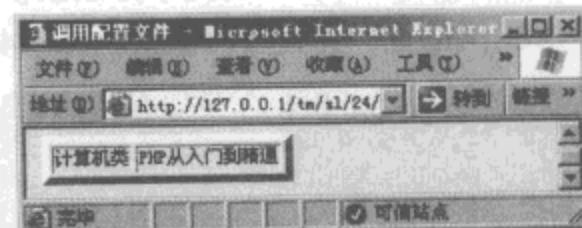


图 24.6 调用配置文件

### 24.3.4 修饰变量

24.3.3 节中介绍了如何在 Smarty 模板中调用变量。但有的时候，不仅要取得变量值，还要对变量进行处理。变量修饰的一般格式如下：

```
{variable_name|modifier_name: parameter1:...}
```

- variable\_name 为变量名称。
- modifier\_name 为修饰变量的方法名。变量和方法之间使用符号“|”分隔。
- parameter1 是参数值。如果有多个参数，则使用“:”分开。

Smarty 提供了修饰变量的方法。常用的方法如表 24.2 所示。

表 24.2 修饰变量的常用方法和说明

方法名	说 明
capitalize	首字母大写
count_characters:true/false	变量中的字符串个数。如果后面有参数 true，则空格也被计算，否则忽略空格
cat: "characters"	将 cat 中的字符串添加到指定字符串的后面
date_format: "%Y-%M-%D"	格式化日期和时间。等同于 PHP 中的 strftime() 函数
default: "characters"	设置默认值。当变量为空时，将使用 default 后面的默认值
escape: "value"	用于字符串转码。value 值可以为 html、htmlall、url、quotes、hex、hexentity 和 javascript，默认为 html
lower	将变量字符串改为小写
nl2br	所有的换行符将被替换成  ，功能同 PHP 中的 nl2br() 函数一样
regex_replace:"parameter1":"value2"	正则替换。用 value2 替换所有符合 parameter1 标准的字串
replace: "value1":"value2"	替换。使用 value2 替换掉所有 value1
string_format: "value"	使用 value 来格式化字符串。如果 value 为 %d，则字符串被格式化为十进制数
strip_tags	去掉所有的 html 标签
upper	将变量改为大写

在对变量进行修饰时，不仅可以单独使用上面的方法，还可以同时使用多个。在每个方法之间使用“|”分隔即可。

**例 24.5** 使用表 24.2 中的几个方法来修饰字符串。其他方法的使用，读者可以自行练习。实例代码如下：（实例位置：光盘\TM\Instances\24\5）

#### Templates/5/index.tpl 文件

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
<link rel="stylesheet" href=".//css/style.css" />
</head>
```

```

<body>
原文: {$str}
<p>
变量中的字符数(包括空格): {$str|count_characters:true}
<br />
使用变量修饰方法后: {$str|nl2br|upper}
</body>
</html>
index.php 文件
<?php
    include_once "../config.php";
    $str1 = '这是一个实例。';
    $str2 = "\n 图书->计算机类->php\n 书名: 《php 从入门到精通》";
    $str3 = "\n 价格: ￥59/本。";
    $smarty->assign('title','使用变量修饰方法');
    $smarty->assign('str',$str1.$str2.$str3.$str4);
    $smarty->display('5/index.tpl');
?>

```

运行结果如图 24.7 所示。

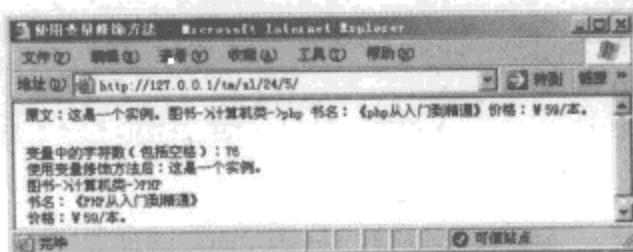


图 24.7 使用变量修饰方法

### 24.3.5 流程控制

Smarty 模板中的流程控制语句包括 if…elseif…else 条件控制语句和 foreach、section 循环控制语句。

#### 1. if…elseif…else 语句

if 条件控制语句的使用和 PHP 中的 if 大同小异。需要注意的是, if 必须以 /if 为结束标记。if 语句的格式如下:

```

{if 条件语句 1}
    语句 1
{elseif 条件语句 2}
    语句 2
{else}
    语句 3
{/if}

```

上述条件语句中, 除了可以使用 PHP 中的 <、>、=、!= 等常见运算符外, 还可以使用 eq、ne、neq、gt、lt、lte、le、gte、ge、is even、is odd、is not even、is not odd、not、mod、div by、even by、odd by 等修饰词修饰。具体含义留给读者自己动手来理解。

**例 24.6 使用条件判断语句选择不同的返回信息，代码如下：**（实例位置：光盘\TM\Instances\24\6）

**templates/6/index.tpl 文件**

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
<link rel="stylesheet" href="../css/style.css" />
</head>
<body>
<p>
{if $smarty.get.type == 'tm'}
欢迎光临，{$smarty.get.type}
{else}
对不起，您不是本站 VIP，无权访问此栏目。
{/if}
</body>
</html>
```

**index.php 文件**

```
<?php
include_once "../config.php";
$smarty->assign("title","if 条件判断语句");
$smarty->display("6/index.tpl");
?>
```

运行结果如图 24.8 所示。

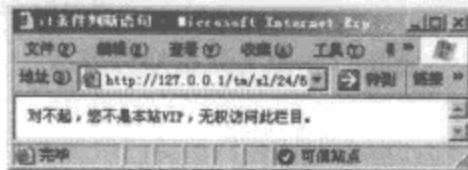


图 24.8 if 条件判断语句

## 2. foreach 循环控制

Smarty 模板中的 foreach 语句可以循环输出数组，与另一个循环控制语句 section 相比，在使用格式上要简单得多，一般用于简单数组的处理。foreach 语句的格式如下：

```
{foreach name=foreach_name key=key item=item from=arr_name}
...
{/foreach}
```

其中，name 为该循环的名称；key 为当前元素的键值；item 为当前元素的变量名；from 为该循环的数组。其中，item 和 from 是必要参数，不可省略。

**例 24.7 使用 foreach 语句，循环输出数组 infobook 的全部内容，代码如下：**（实例位置：光盘\TM\Instances\24\7）

**templates/7/index.tpl 文件**

```
<html>
<head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
</head>
<body>
使用 foreach 语句循环输出数组。<p>
{foreach key=key item=item from=$infobook}
{$key} => {$item}<br />
{/foreach}
</body>
</html>
index.php 文件
<?php
    include_once '../config.php';
    $infobook = array('object'=>'book','type'=>'computer','name'=>'PHP 从入门到精通','publishing'=>'清华大学出版社');
    $smarty->assign('title','使用 foreach 循环输出数组内容');
    $smarty->assign('infobook',$infobook);
    $smarty->display('7/index.tpl');
?>

```

运行结果如图 24.9 所示。

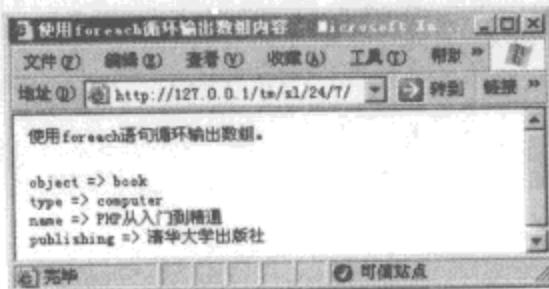


图 24.9 使用 foreach 语句循环输出数组内容

### 3. section 循环

Smarty 模板中的另一个循环语句是 section，该语句可用于比较复杂的数组。section 语句的语法如下：

```
{section name="sec_name" loop=$arr_name start=num step=num}
```

其中，name 为该循环的名称；loop 为循环的数组；start 表示循环的初始位置，如 start=2，则说明循环是从 loop 数组的第二个元素开始的；step 表示步长，如 step=2，则循环一次后数组的指针将向下移动两位，依此类推。

**例 24.8 使用 section 语句循环输出一个二维数组，代码如下：(实例位置：光盘\TM\Instances\24\8)**

```

templates/8/index.tpl 文件
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>{$title}</title>
<link rel="stylesheet" href="../css/style.css" />
</head>
<body>

```

```

<table width="100" border="0" align="left" cellpadding="0" cellspacing="0">
{section name=sec1 loop=$obj}
<tr>
    <td colspan="2"{$obj[sec1].bigclass}</td>
</tr>
{section name=sec2 loop=$obj[sec1].smallclass}
<tr>
    <td width="25">&ampnbsp</td>
    <td width="75"{$obj[sec1].smallclass[sec2].s_type}</td>
</tr>
{/section}
{/section}
</table>
</body>
</html>
index.php 文件
<?php
    require "../config.php";
    $obj = array(array("id" => 1, "bigclass" => "计算机图书", "smallclass" => array(array("s_id" => 1, "s_type" => "PHP")), array("id" => 2, "bigclass" => "历史传记", "smallclass" => array(array("s_id" => 2, "s_type" => "中国历史"), array("s_id" => 3, "s_type" => "世界历史"))), array("id" => 3, "bigclass" => "畅销小说", "smallclass" => array(array("s_id" => 4, "s_type" => "网络小说"), array("s_id" => 5, "s_type" => "科幻小说"))));
    $smarty->assign('title', 'section 循环控制');
    $smarty->assign("obj", $obj);
    $smarty->display("8/index.tpl");
?>

```

运行结果如图 24.10 所示。

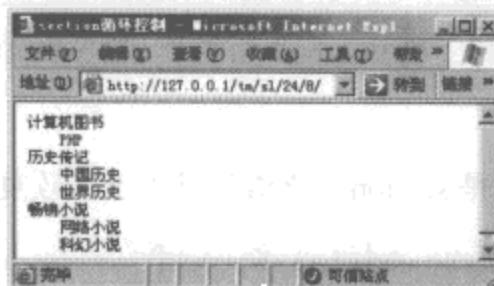


图 24.10 使用 section 循环控制输出数组

### 24.3.6 内建函数

Smarty 自带一些内建函数，内建函数是模板语言的一部分，用户不能创建名称与内建函数相同的自定义函数，也不能修改内建函数。下面介绍几个比较常用的内建函数。

#### 1. config\_load 函数

从配置文件中加载变量。配置文件有可能包含多个部分，此时可以使用附加属性 section 指定从哪一部分中取得变量：

```
{config_load file="file_name" section="add_attribute"}
```

其中, `file_name` 指定包含的配置文件的名称; `add_attribute` 为附加属性, 当配置文件中包含多个部分时应用, 指定具体从哪一部分中取得变量。

有关该函数的应用可以参考例 24.4。

## 2. include 标签

`include` 标签用于在当前模板中包含其他模板, 当前模板中的变量在被包含的模板中可用。必须指定 `file` 属性, 该属性指明模板资源的位置。标签语法如下:

```
{include file="file_name" assign="" var=" "}
```

其中, `file` 指定包含模板文件的名称; `assign` 指定一个变量保存包含模板的输出; `var` 传递给待包含模板的本地参数, 只在待包含模板中有效。

## 3. php 标签

`php` 标签允许在模板中直接嵌入 PHP 脚本。是否处理这些语句取决于 `$php_handling` 的设置。该语句通常不需要使用, 当然如果非常了解此特性或认为有必要, 也可以使用。

`php` 标签的应用演示如下:

```
{php}
    // including a php script directly
    // from the template.
    include("/conn.php");
{/php}
```

上述代码为直接在模板中应用 `include` 语句调用包含文件。

**例 24.9 应用 `include` 标签在模板页中包含其他模板, 并通过 `php` 标签在模板页中包含其他 PHP 脚本文件。实例代码如下: (实例位置: 光盘\TM\Instances\24\9)**

```
templates/9/index.tpl 文件
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>{$title}</title>
<link rel="stylesheet" href=".//css/style.css" />
</head>
<body>
<!--应用 php 标签通过 include 语句调用包含文件调用-->
{php}
    include("top.php");
{/php}
<table width="99" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td height="20"><span class="STYLE1">{$arr1.object}</span></td>
    </tr>
    <tr><td height="27">市场价: {$arr1.price} ￥</td></tr>
    <tr><td height="22">热卖价: {$arr1.t_price} ￥</td></tr>
</table>
<!--通过 include 标签包含 tpl 文件-->
```

```

{include file="9/bottom.tpl"} //调用头文件
index.php 文件
<?php
    include '../config.php';
    $arr1 = array("object"=>'液晶显示器',"price"=>'3500',"t_price"=>'2888');
    $arr2 = array("object"=>'羽绒服',"price"=>'150',"t_price"=>'88');
    $arr3 = array("object"=>'电视',"price"=>'1000',"t_price"=>'888');
    $smarty->assign('title','应用内建函数');
    $smarty->assign("arr1",$arr1);
    $smarty->assign('arr2',$arr2);
    $smarty->assign('arr3',$arr3);
    $smarty->display('9/index.tpl');
?>

```

在本实例中，通过 `php` 标签在模板页中直接使用 PHP 的 `include` 包含语句调用头文件，通过 Smarty 的 `include` 标签包含 `tpl` 文件输出页面的尾文件，运行结果如图 24.11 所示。



图 24.11 `php` 标签的应用

## 24.4 Smarty 程序设计

### ■ 视频讲解：光盘\TM\Video\24\Smarty 程序设计.exe

通过前面的学习已经知道，在 Smarty 模板中，是不推荐有 PHP 代码段的，所有的 PHP 程序都要另写成文件。Smarty 程序的功能主要分两种：一种功能是和 Smarty 模板之间的交互，如方法 `assign`、`display`；另一种功能就是配置 Smarty，如变量 `template_dir`、`$config_dir` 等。本节将学习 Smarty 程序设计的其他一些方法和配置参数。

### 24.4.1 Smarty 中的常用方法

#### 1. `assign`

`assign` 用于在模板被执行时为模板变量赋值。语法如下：

```
{assign var="" value=""}
```

参数 var 是被赋值的变量名，参数 value 是赋给变量的值。

## 2. display

display 方法用于显示模板，需要指定一个合法的模板资源的类型和路径。还可以通过第二个可选参数指定一个缓存号，相关的信息可以查看缓存。

```
void display (string template [, string cache_id [, string compile_id]])
```

参数 template 指定一个合法的模板资源的类型和路径；参数 cache\_id 为可选参数，指定一个缓存号；参数 compile\_id 为可选参数，指定编译号。编译号可以将一个模板编译成不同版本使用，如针对不同的语言编译模板。编译号的另外一个作用是，如果存在多个\$template\_dir 模板目录，但只有一个\$compile\_dir 编译后存档目录，就可以为每一个\$template\_dir 模板目录指定一个编译号，以避免相同的模板文件在编译后互相覆盖。相对于在每一次调用 display() 时都指定编译号，也可以通过设置\$compile\_id 编译号属性来一次性设定。

Smarty 中除了使用 assign 和模板交互外，还有一些比较常用的方法。方法名称和功能说明如表 24.3 所示。

表 24.3 Smarty 程序设计常用方法和说明

方法格式	说明
void append (string varname, mixed var [, boolean merge])	该方法向数组中追加元素
void clear_all_assign ()	清除所有模板中的赋值
void clear_assign (string var)	清除一个指定的赋值
void config_load (string file [, string section])	加载配置文件，如果有参数 section，说明只加载配置文件中相对应的一段数据
string fetch (string template)	返回模板的输出内容，但不直接显示出来
array get_config_vars ([string varname])	获取指定配置变量的值，如果没有参数，则返回一个所有配置变量的数组
array get_template_vars ([string varname])	获取指定模板变量的值，如果没有参数，则返回一个所有模板变量的数组
bool template_exists (string template )	检测指定的模板是否存在

### 24.4.2 Smarty 的配置变量

Smarty 中只有一个常量 SMARTY\_DIR，用来保存 Smarty 类库的完整路径，其他的所有配置信息都保存到相应的变量中。这里将介绍包括前面章节中接触过的 template\_dir 等变量的作用及设置。

- \$template\_dir：模板目录。模板目录用来存放 Smarty 模板，在前面的实例中，所有的.tpl 文件都是 Smarty 模板。模板的后缀没有要求，一般都定义为.tpl、.html 等。
- \$compile\_dir：编译目录。顾名思义，就是编译后的模板和 PHP 程序所生成的文件，默认路径为当前执行文件所在的目录下的 templates\_c 目录。进入到编译目录，可以发现许多“%…%

“%index.tpl.php”格式的文件。随便打开一个这样的文件可以发现，实际上 Smarty 将模板和 PHP 程序又重新组合成一个混编页面。

- \$cache\_dir：缓存目录。用来存放缓存文件。同样，在 cache 目录下可以看到生成的.html 文件。如果 caching 变量开启，那么 Smarty 将直接从这里读取文件。
- \$config\_dir：配置目录。该目录用来存放配置文件。例 24.4 中所用到的配置文件，就保存到这里。
- \$debugging：调试变量。该变量可以打开调试控制台。只要在配置文件（config.php）中将 \$smarty->debugging 设为 true 即可使用。
- \$caching：缓存变量。该变量可以开启缓存。只要当前模板文件和配置文件未被改动，Smarty 就直接从缓存目录中读取缓存文件而不重新编译模板。

## 24.5 实 战

### 24.5.1 通过 Smarty 实现网页的动静分离

#### ■ 视频讲解：光盘\TM\Video\24\通过 Smarty 实现网页的动静分离.exe

Smarty 的功能是实现网页的动静分离，本章中已经对安装、配置和使用 Smarty 的方法进行了详细讲解，下面将应用 Smarty 模板技术真正实现一次网页的动静分离。

**例 24.10 应用 Smarty 模板技术，将从数据库中读取的数据输出到网页中。操作步骤如下：**（实例位置：光盘\TM\Instances\24\10）

(1) 安装和配置 Smarty。将 Smarty 模板的压缩包解压，复制 libs 目录到服务器 tm\sl\24\10\system 目录下，并重新命名为 smarty，在 smarty 目录下新建 4 个目录 templates、templates\_c、configs 和 cache，用于存储不同的文件。

编写配置文件，分别指定 4 个目录的存储位置。配置文件 system.smarty.inc.php 的代码如下。

```
<?php
require("smarty/smarty/libs/Smarty.class.php");
class SmartyProject extends Smarty{
    function SmartyProject(){
        $this->template_dir = "/"; //指定模板文件存储在根目录下
        $this->compile_dir = "./system/smarty/templates_c/";
        $this->config_dir = "./system/smarty/configs/";
        $this->cache_dir = "./system/smarty/cache/";
    }
}
?>
```

(2) 创建数据库操作类。本实例中通过 ADODB 来操作数据库，首先将 ADODB 包放置在 tm\sl\24\10\system 目录下。然后创建数据库连接类，将其存储在 system.class.inc.php 文件中。数据库连接类的代码如下。

```
class ConnDB{ //数据库连接类
    var $dbtype;
```

```

var $host;
var $user;
var $pwd;
var $dbname;
var $debug;
var $conn;
function ConnDB($dbtype,$host,$user,$pwd,$dbname,$debug=false){           //构造方法
    $this->dbtype=$dbtype;
    $this->host=$host;
    $this->user=$user;
    $this->pwd=$pwd;
    $this->dbname=$dbname;
    $this->debug=$debug;
}
function GetConnId(){           //实现数据库的连接并返回连接对象
    require("adodb/adodb.inc.php");
    if($this->dbtype=="mysql" || $this->dbtype=="mssql"){
        if($this->dbtype=="mysql")
            $this->conn=NewADOConnection("mysql");
        else
            $this->conn=NewADOConnection("mssql");
        $this->conn->Connect($this->host,$this->user,$this->pwd,$this->dbname);
    }elseif($this->dbtype=="access"){
        $this->conn=NewADOConnection("access");
        $this->conn->Connect("Driver={Microsoft Access Driver (*.mdb)};Dbq=".$this->dbname.";Uid=
".$this->user.";Pwd=".$this->pwd.");"
    }
    $this->conn->Execute("set names gb2312");
    if($this->dbtype=="mysql")
        $this->conn->debug=$this->debug;
    return $this->conn;
}
function CloseConnId(){           //关闭与数据库的连接
    $this->conn->Disconnect();
}
}

```

(3) 创建 system.inc.php 文件，实例化数据库连接类，代码如下。

```

<?php
require("system.smarty.inc.php");
require("system.class.inc.php");
//数据库连接类实例化
$connobj=new ConnDB("mysql","localhost","root","root","db_shop",false);
$conn=$connobj->GetConnId();
//调用 Smarty 模板
$smarty=new SmartyProject();
?>

```

(4) 创建 index.php 页面，定义主页面的布局，通过 include 语句调用指定的包含文件，关键代码

如下。

```
<tr>
<td width="286" height="514" rowspan="2" align="left" valign="top">&nbsp;</td>
<td width="648" align="center" valign="top">
    <?php      include('nominate.php');      ?>
</td>
<td width="70" rowspan="2" align="center" valign="top">&nbsp;</td>
</tr>
<tr>
    <td align="center" valign="top">
        <?php      include('newhot.php'); ?>
    </td>
</tr>
```

(5) 创建 PHP 脚本文件 nominate.php 和 newhot.php, 从数据库中读取数据, 并将数据存储到模板变量中, 最后指定模板页面。这里只给出 nominate.php 文件中的代码。

```
<?php
require("system/system.inc.php"); //调用指定的文件
$sql = "select id,name,pics,m_price,v_price from tb_commo where isnom = 1 order by id desc"; //定义查询语句
$num = 4; //定义输出记录数量
$rst = $conn->SelectLimit($sql,$num); //执行查询语句
$nomarr = $rst->GetArray(); //返回查询结果
$smarty->assign('nomarr',$nomarr); //将查询结果赋给一个数组
$smarty->display('nominate.tpl'); //指定模板页面
?>
```

(6) 创建 tpl 静态页面 nominate.tpl 和 newhot.tpl, 通过 foreach 循环读取出模板变量中的数据, 由于篇幅所限, 这里没有给出 tpl 文件的代码, 可参考光盘中的内容。

运行结果如图 24.12 所示。



图 24.12 通过 Smarty 输出数据库中的数据

## 24.5.2 Smarty 分页类

### ■ 视频讲解：光盘\TM\Video\24\Smarty 分页类.exe

上述实例讲解了如何通过 Smarty 实现网页的动静分离，在此基础上讲解如何在 Smarty 中实现分页功能。

**例 24.11 应用 Smarty 模板技术，通过分页类将从数据库中读取的数据分页输出。操作步骤如下：**  
**(实例位置：光盘\TM\Instances\24\11)**

(1) 首先进行 Smarty 的安装和配置。这里不再赘述，具体操作方法可参考 24.2 节中的内容。

(2) 创建数据库连接和管理类。这里不但编写了一个数据库连接类，而且又编写了一个数据库的管理类。通过数据库管理类可以实现对数据库进行查询、修改、删除和更新的操作，代码如下。

```
class AdminDB{                                //数据库管理类
    function ExecSQL($sqlstr,$conn){          //执行 SQL 语句
        $sqltype=strtolower(substr(trim($sqlstr),0,6));    //截取 SQL 语句中的前 6 个字符串
        $rs=$conn->Execute($sqlstr);
        if($sqltype=="select"){                //判断如果执行 select 查询
            $array=$rs->GetRows();           //返回查询结果
            if(count($array)==0 || $rs==false) //如果不执行查询
                return false;                //返回 false
            else
                return $array;               //否则返回一个数组
        }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){ //如果执行操作语句
            if($rs)
                return true;
            else
                return false;
        }
    }
}
```

数据库连接和管理类的完整代码存储在 system.class.inc.php 文件中。

(3) 创建分页类，同样将其存储在 system.class.inc.php 文件中，分页类的代码如下。

```
class SepPage{
    var $rs;
    var $pagesize;
    var $nowpage;
    var $array;
    var $conn;
    var $sqlstr;
    function ShowDate($sqlstr,$conn,$pagesize,$nowpage){ //构造方法
        if(!isset($nowpage) || $nowpage=="")
            $this->nowpage=10;                      //判断变量值是否为空
                                                //定义每页输出的记录数
        else
            $this->nowpage=$nowpage;
    }
}
```

```

    $this->pagesize=$pagesize;           //定义每页输出的记录数
    $this->conn=$conn;                  //连接数据库返回的标识
    $this->sqlstr=$sqlstr;              //执行的查询语句
    $this->rs=$this->conn->PageExecute($this->sqlstr,$this->pagesize,$this->nowpage);
    @$this->array=$this->rs->GetRows(); //获取记录数
    if(count($this->array)==0 || $this->rs==false)
        return false;
    else
        return $this->array;
    }

    function ShowPage($contentname,$utits,$anothersearchstr,$class){
        $allrs=$this->conn->Execute($this->sqlstr); //执行查询语句
        $record=count($allrs->GetRows());           //统计记录总数
        $pagecount=ceil($record/$this->pagesize);    //计算总页数
        $str.="共有".$contentname." ".$record." ".$utits." 每页显示 ".$this->pagesize." 第 ".$this->rs->AbsolutePage()." 页 / 共 ".$pagecount." 页";
        $str.="";
        if(!$this->rs->AtFirstPage())
            $str.= "<a href='".$._SERVER['PHP_SELF']."'?page=1".$anothersearchstr." class='".$class."'>首页</a>";
        else
            $str.= "<font color='#555555'>首页</font>";
        $str.="";
        if(!$this->rs->AtFirstPage())
            $str.= "<a href='".$._SERVER['PHP_SELF']."'?page=".($this->rs->AbsolutePage()-1).$anothersearchstr." class='".$class."'>上一页</a>";
        else
            $str.= "<font color='#555555'>上一页</font>";
        $str.="";
        if(!$this->rs->AtLastPage())
            $str.= "<a href='".$._SERVER['PHP_SELF']."'?page=".($this->rs->AbsolutePage()+1).$anothersearchstr." class='".$class."'>下一页</a>";
        else
            $str.= "<font color='#555555'>下一页</font>";
        $str.="";
        if(!$this->rs->AtLastPage())
            $str.= "<a href='".$._SERVER['PHP_SELF']."'?page=".$pagecount.$anothersearchstr." class='".$class."'>尾页</a>";
        else
            $str.= "<font color='#555555'>尾页</font>";
        if(count($this->array)==0 || $this->rs==false)
            return "";
        else
            return $str;
    }
}

```

(4) 创建 system.inc.php 文件，对数据库连接、操作类以及分页类进行实例化，代码如下。

```
<?php
    require("system.smarty.inc.php");
    require("system.class.inc.php");
    //数据库连接类实例化
    $connobj=new ConnDB("mysql","localhost","root","root","db_shop",false);
    $conn=$connobj->GetConnId();
    //数据库操作类实例化
    $admindb=new AdminDB();
    //分页类实例化
    $seppage=new SepPage();
    //调用 Smarty 模板
    $smarty=new SmartyProject();
?>
```

(5) 创建 index.php 文件，对网页进行布局。

(6) 创建 allnew.php 文件，编写 PHP 代码页，从数据库中读取数据，并将数据赋给指定的模板变量，指定模板页，代码如下。

```
<?php
require_once("system/system.inc.php"); //调用指定的文件
$arraybbstell=$admindb->ExecSQL("select * from tb_commo where isnew = 1 order by isnew,id desc",$conn);
//执行 select 查询语句
if(!$arraybbstell){
    $smarty->assign("isbbstell","F"); //如果执行失败，则输出模板变量 isbbstell 的值为 F
} else {
    $smarty->assign("isbbstell","T"); //如果执行成功，则输出模板变量 isbbstell 的值为 T
    $smarty->assign("arraybbstell",$arraybbstell); //定义模板变量 arraybbstell，输出数据库中的数据
}
$arraybbs=$seppage->ShowDate("select * from tb_commo where isnew = 1 ",$conn,3,$_GET["page"]);
//调用分页类，实现分页功能
if(!$arraybbs){
    $smarty->assign("isbbs","F");
} else {
    $smarty->assign("isbbs","T");
    $smarty->assign("showpage",$seppage->ShowPage("帖子","条","","al")); //定义输出分页数据的模板变量 showpage
    $smarty->assign("arraybbs",$arraybbs);
}
$smarty->assign('title','最新商品');
$smarty->display('allnew.tpl');
```

(7) 创建 allnew.tpl 模板页，输出模板变量中传递的数据，实现数据的分页输出，完整代码可参考光盘中的内容。

至此，Smarty 分页类的讲解完毕，其运行结果如图 24.13 所示。

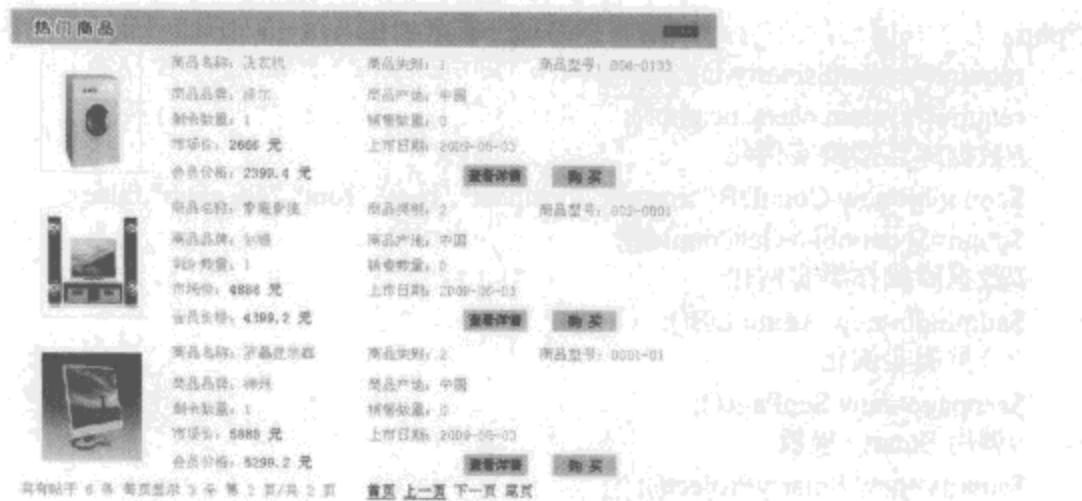


图 24.13 在 Smarty 中通过分页类输出数据

### 24.5.3 网站公告

#### 视频讲解：光盘\TM\Video\24\网站公告.exe

**例 24.12 应用 Smarty 模板技术，实现输出网站公告信息的功能。操作步骤如下：**（实例位置：**光盘\TM\Instances\24\12**）

（1）创建 config.php 文件，配置 Smarty 模板。指定服务器的根目录，Smarty 模板在实例中的存储位置，以及各个文件夹的存储位置，代码如下。

```
<?php
/* 定义服务器的绝对路径 */
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);
/* 定义 Smarty 目录的绝对路径 */
define('SMARTY_PATH','tm\sl\24\12\Smarty\\');
/* 加载 Smarty 类库文件 */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/* 实例化一个 Smarty 对象 */
$smarty = new Smarty;
/* 定义各个目录的路径 */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs/';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';
?>
```

（2）创建 public.php 文件，从数据库中读取出公告信息，将公告信息存储到指定的模板变量中，并指定模板页。

```
<?php
include_once "conn/conn.php";           //连接数据库
include_once "config.php";              //调用配置文件
$sql = "select id,title from tb_public order by id"; //编写查询语句
$num = 4;
$rst = $conn->SelectLimit($sql,$num);    //执行查询操作
```

```

$arr = $rst->GetAssoc();
$smarty->assign('arr',$arr);
$smarty->display('public.tpl');
?>
    
```

(3) 在 smarty\templates 文件夹下创建 public.tpl 模板文件，输出模板变量中的公告标题信息，关键代码如下。

```

<td width="193" align="left" valign="top" class="exam">
    {foreach key=key item=item from=$arr}
        <a href="#" class="lk" onclick="return showme({$key},'showpub.php');" >{$item}</a><br />
    {/foreach}
</td>
    
```

(4) 创建 links.js 文件，定义 showme() 函数，通过 JavaScript 脚本调用 showpub.php 文件，查看公告的详细内容。showme() 函数的语法如下：

```

// JavaScript Document
function showme(key,wurl){
    var purl = wurl + "?id=" + key;
    open(purl,'_blank','width=450 height=200',false);
    return false;
}
    
```

该函数的参数是指定公告信息的 ID 和调用的文件。

(5) 创建 showpub.php 文件，根据传递的 ID 值，从数据库中查询出数据的详细信息，并将数据添加到指定的模板变量中，最后指定模板页。

```

<?php
    include_once 'conn/conn.php';
    include_once 'config.php';
    $id = $_GET['id'];
    $sql = "select * from tb_public where id = ". $_GET['id'];
    $rst = $conn->execute($sql);
    $arr = $rst->getAssoc();
    $smarty->assign('title','查看公告');
    $smarty->assign('arr',$arr[$id]);
    $smarty->display('showpub.tpl');
?>
    
```

(6) 创建模板页 showpub.tpl，读取模板变量中的数据，输出公告的详细内容，代码如下。

```

<table width="400" align="center" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td width="70%" height="25" align="center" valign="middle" class="left">标题: {$arr.title}</td>
        <td width="30%" height="25" align="center" valign="middle" class="right">&nbsp;{$arr.addtime}</td>
    </tr>
    <tr>
        <td height="100" colspan="2" align="left" valign="top" class="all" style="text-indent: 10px;"><br>&nbsp;
    
```

```

{$arr.content}</td>
</tr>
</table>

```

运行本实例，将直接输出网站公告的标题信息，单击某个具体的标题，将弹出一个新的窗口，输出具体公告的内容，如图 24.14 所示。

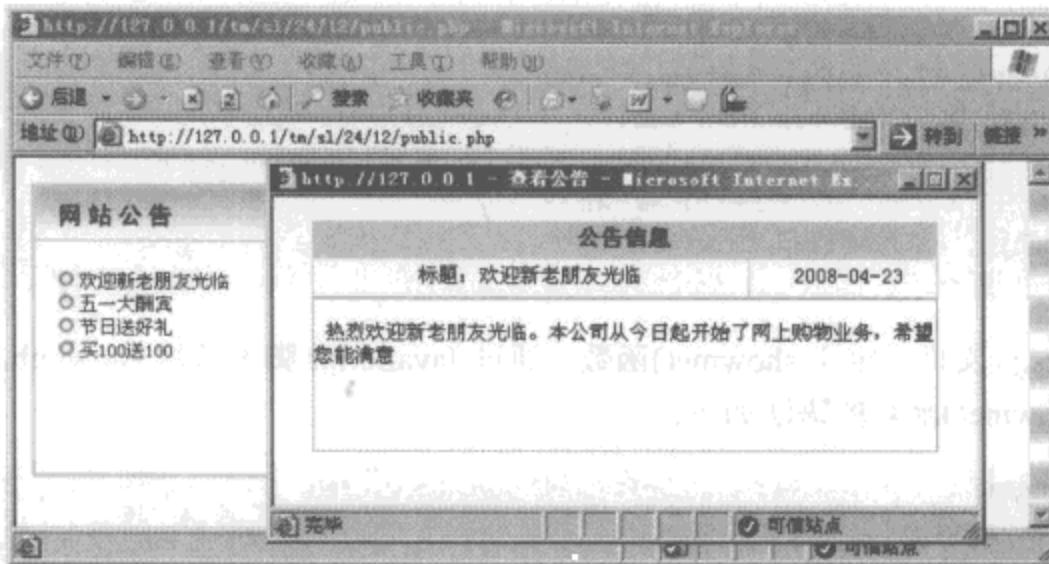


图 24.14 输出公告信息

## 24.6 本章小结

本章主要介绍了 Smarty 模板的安装、配置及使用，并且对 Smarty 的模板设计和程序设计进行了讲解，它们是两个相辅相成的内容，是应用 Smarty 模板的基础。在实战中讲解了 Smarty 的实际应用。希望通过本章的学习，读者能够掌握 Smarty 模板技术，并且能够将其灵活地运用到实际的网站开发中。

## 24.7 实战练习

1. 自定义配置一个 Smarty 环境，并将 Smarty 定界符换成其他符号。（答案位置：光盘\TM\Instances\24\13）
2. 使用 Smarty 模板创建一个购物车。（答案位置：光盘\TM\Instances\24\14）

# 第4部分

## 项目实战

- » 第25章 博客管理系统
- » 第26章 应用Smarty模板开发电子商务平台
- » 第27章 办公自动化系统
- » 第28章 供求信息网
- » 第29章 图书借阅管理系统





# 第 25 章

## 博客管理系统

( 视频讲解: 169 分钟)

Blog 中文译为“博客”，全名 Weblog，后来缩写为 Blog。Blogger 就是写博客的人，是指习惯于在网上书写日记、发布个人照片、展示个性自我的用户群体。博客的人性化和平民视角使得它提供的消息更贴近人们的生活，每个人都可以随时把自己的想法、灵感和评论在博客上发表，游客可以浏览所有的博客文章及相关评论。在网络高速发展的今天，个人电脑逐渐普及，博客作为 Web 2.0 时代的核心概念之一，已经被很多网民所使用，所以 Web 开发人员掌握博客的开发流程是非常必要的。

通过阅读本章，可以学习到：

- 博客的概念和功能
- 开发博客的一般流程
- 巩固 PHP 对 MySQL 数据库的增、删、改、查操作
- 常用页面特效的制作方法
- 文件上传的方法
- 项目开发过程中常见问题的解决方法
- 将网站发布到 Internet 的方法

## 25.1 项目描述

博客作为 Web 2.0 时代的重要应用之一，已经得到广泛地应用，所以 Web 开发人员应该掌握博客的基本开发流程以及博客中各个模块的具体功能和实现方式。本章的博客主要具有发表、查询、删除和分页浏览博客文章，发表、删除文章评论，查询、上传、删除分页浏览图片，添加、查询、删除、分页浏览朋友圈，以及查询、浏览和删除用户信息等功能。

注册为正式用户后，即可发表博客文章、评论博客文章、上传图片、设置自己的朋友圈。普通用户具有删除自己所写的文章和上传的图片的权限，网站管理员具有删除普通用户及普通用户所发表的文章和上传的照片的权限。博客管理系统的首页运行效果如图 25.1 所示。



图 25.1 博客管理系统首页运行效果

## 25.2 系统分析

无论是简单的桌面应用还是复杂的电子商务网站，在进行项目开发前都需要花费大量的时间对系统各个方面进行分析，“磨刀不误砍柴工”，好的开头才是整个项目顺利、合格竣工的有效保障。

### 25.2.1 需求分析

在网络高速发展的今天，博客管理系统给人们提供了抒发个人情感、人与人之间进行良好沟通的平台。博客拥有真实的内容，人们可以通过博客记录工作、学习、生活和娱乐的点滴，以及发表文章和对事物的评论，从而在网上建立一个完全属于自己的天地，博客已成为当今网络最为个性化和平民化的个人展示的空间和平台。对于网民，只要拥有博客，就可以超越现实生活，拥有不同以往的全新网上生活；对于博客服务商，则必须从功能提供转型到全方位社会服务的提供，建立虚拟社会，并负责其维护，保证博客日常运行；对于社会而言，有利于构建和谐的互联网空间，维护和谐的社会环境。从这个角度来说，构建新生活方式，将是互联网发展的一个里程碑。很多门户网站都为注册用户提供

了免费的个人博客空间，也因此吸引了很多用户注册，从而很大程度上提高了网站的访问量和知名度，所以在开发社区性、娱乐性等网站中，提供博客模块是很有必要的。

### 25.2.2 系统目标

根据网站性质和需求，网站所包含的功能也是不同的，所以在项目开发前一定要认真做好需求分析，掌握用户所需和项目目标。本博客系统主要实现如下目标：

- 系统采用 B/S 架构，实现人机交互。
- 系统运行稳定，安全可靠。
- 系统界面设计美观友好，操作简便。
- 突出重点内容，显示最新文章和图片等信息。
- 非登录用户可以浏览文章、浏览图片、发表评论，登录用户可以管理自己的文章和上传的图片等信息。
- 全面的搜索查询功能，包括精确查询和模糊查询。
- 完善的文章管理功能，包括文章的发表、删除，及对文章的评论与回复。
- 支持图片上传功能。
- 支持好友功能。
- 支持公告栏功能。

### 25.2.3 系统功能结构

本系统的项目描述中已经详细介绍了该博客所具有的主要功能，为了使读者更加直观和理性地了解该博客系统所具有的功能，下面以结构图的形式进行展示，如图 25.2 所示。

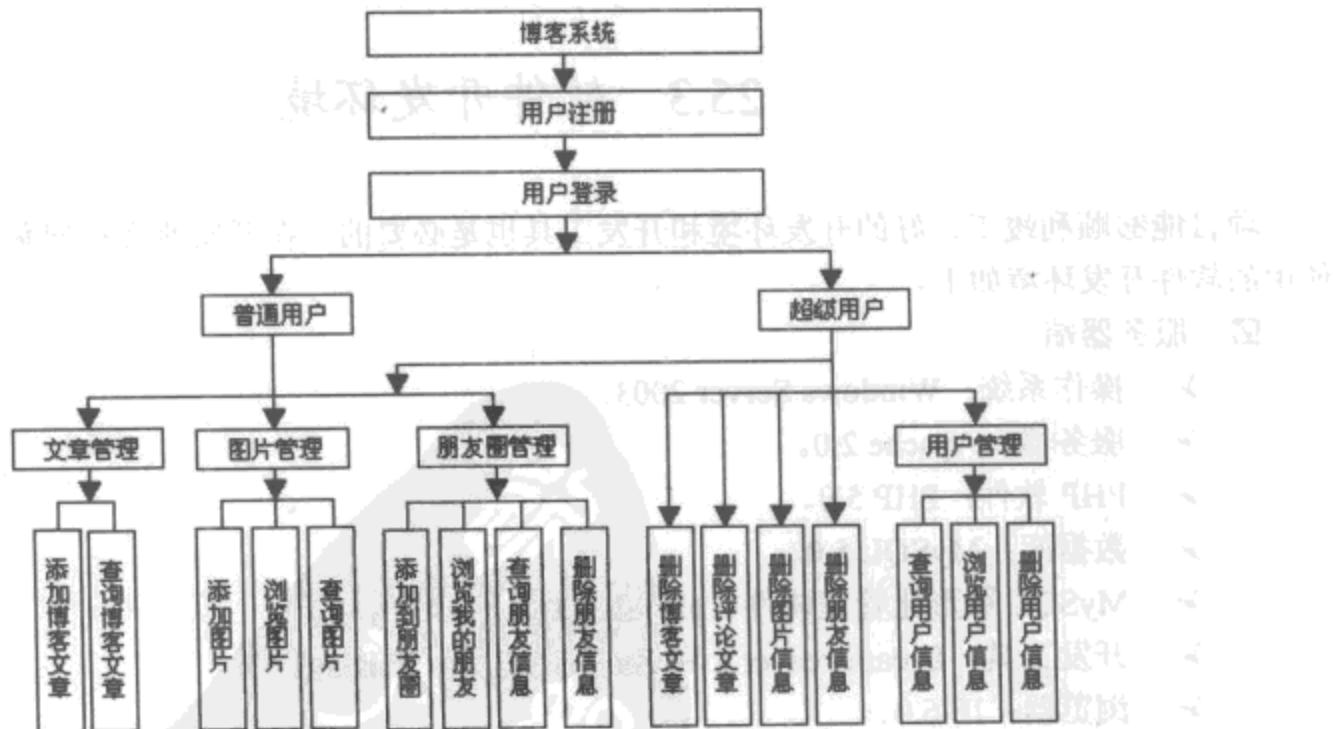


图 25.2 博客管理系统的功能结构图

## 25.2.4 系统流程图

完成系统的基本需求和功能分析后，即可规划出博客所具备的功能和操作流程。本博客系统的流程图如图 25.3 所示。

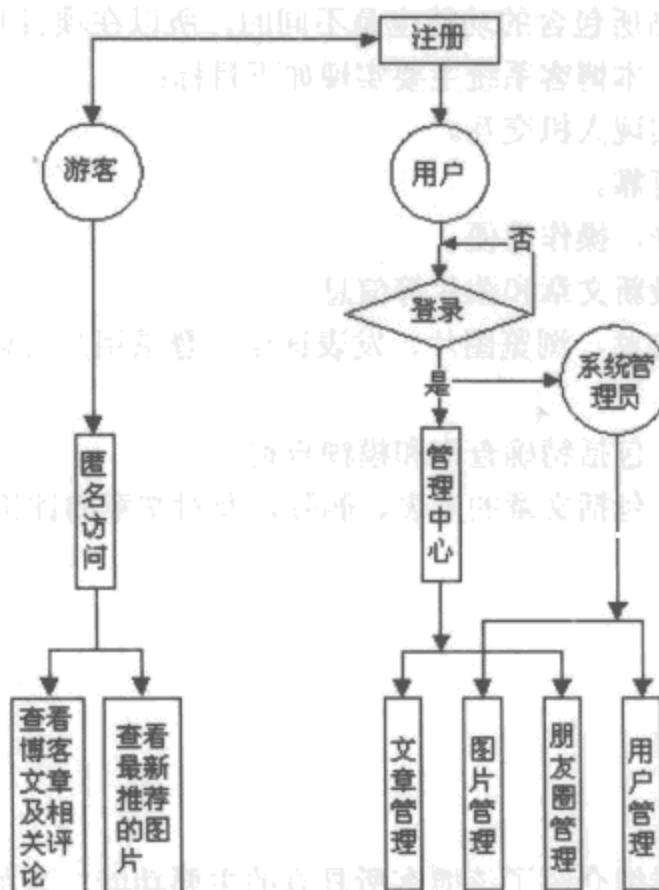


图 25.3 博客管理系统流程图

## 25.3 软件开发环境

项目能够顺利竣工，好的开发环境和开发工具也是必要的。在开发博客管理系统平台时，该项目使用的软件开发环境如下。

### 服务器端

- 操作系统：Windows Server 2003。
- 服务器：Apache 2.0。
- PHP 软件：PHP 5.0。
- 数据库：MySQL 5.0。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- 开发工具：Dreamweaver 8 和 Zend Studio for Eclipse。
- 浏览器：IE 6.0。
- 分辨率：最佳效果 1024×768 像素。

### ☒ 客户端

- 浏览器：推荐使用 IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

## 25.4 数据库与数据表的设计

目前，数据库是程序开发中大量数据有效存储的最佳载体，任何计算机语言都有与之结合性最好的数据库系统。PHP 一般与 MySQL 数据库组合开发，所以本博客系统也将采用 MySQL 作为后台数据库。下面具体讲解本系统中数据库及数据表的设计过程。

### 25.4.1 数据库分析

PHP 可以支持多种数据库，如 Oracle、Sybase、SQL Server、mSQL、MySQL 等。在这些数据库中，由于 MySQL 具有免费、跨平台、使用方便、访问效率高等特点，所以得到广泛地应用，是业界公认的 PHP 的“黄金搭档”，Linux 系统、Apache 服务器、MySQL 数据库和 PHP 语言构成了现在主流的 LAMP 网站架构模式。综上原因，本博客系统也将使用 MySQL 作为博客系统中数据存储的载体。

### 25.4.2 数据库 E-R 图设计

根据需求分析和功能设计，本系统拟订了用户实体、上传图片实体、朋友圈实体、文章实体以及留言实体。为了使读者更清晰地了解每个实体所具备的功能，下面给出系统中主要实体的 E-R 模型。

用户是博客活动的主要参与者，必须对用户身份进行收集，所以将用户个人信息抽象为用户实体表示。用户实体包括注册用户的详细信息，如果想在本博客系统中进行发表文章、上传图片等操作，则必须要先进行注册。用户信息实体 E-R 图如图 25.4 所示。

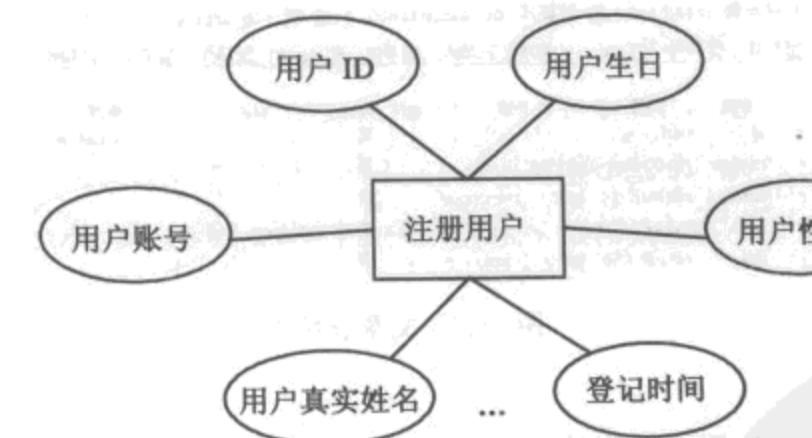


图 25.4 用户实体 E-R 图

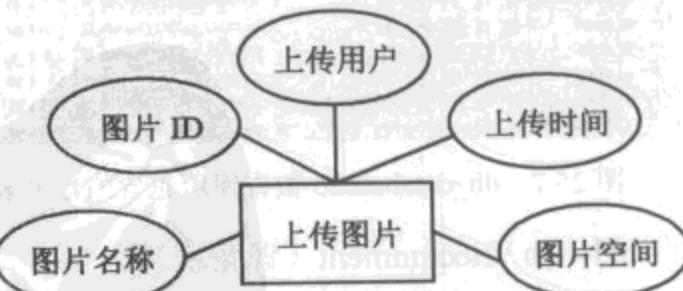


图 25.5 图片实体 E-R 图

### 25.4.3 创建数据库和数据表

完成对博客实体分析后，下面根据分析结果创建数据库和数据表。创建和维护 MySQL 数据库及数据表，既可以在 Windows 系统的命令提示符窗口或 Linux/UNIX 系统的终端命令窗口中实现，也可以使用 MySQL 的图形化界面管理工具实现。MySQL 图形界面管理工具有多种，其中，phpMyAdmin 是由 PHP 编写的通过客户端浏览器对 MySQL 数据库进行管理的工具，其完全免费、安装方便，所以本系统将使用 phpMyAdmin 作为 MySQL 数据库创建和维护的工具。

在浏览器中打开 phpMyAdmin，如图 25.6 所示，在“创建一个新的数据库”选项组中的文本框中输入“db\_database25”作为博客数据库的名称，然后单击“创建”按钮即可完成数据库的创建。

完成数据库的创建后，即可创建博客所要使用的数据表。根据实体 E-R 图和本系统的实际情况，需要创建 6 张数据表，其中 db\_database25 数据库中所包含的表如图 25.7 所示。

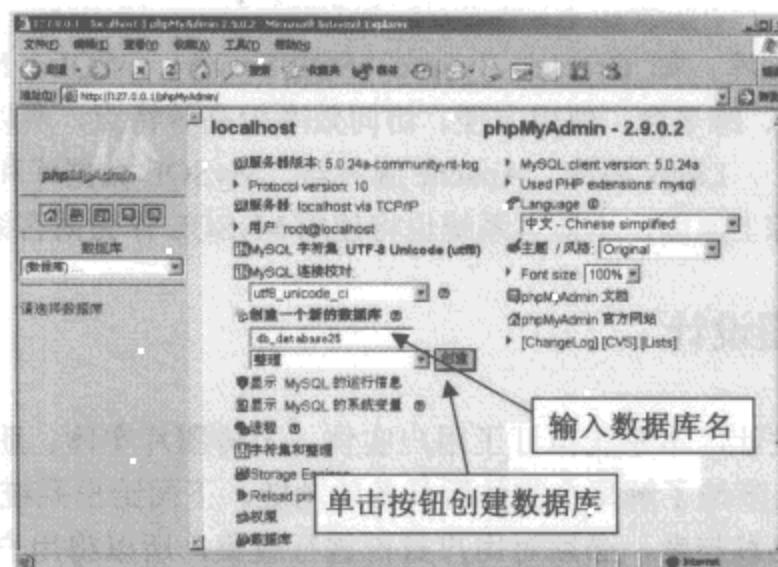


图 25.6 创建数据库

下面具体介绍这 6 张数据表的结构设计和说明。

#### tb\_article (文章表)

文章表用于存储用户发表过的文章信息。tb\_article 表的结构如图 25.8 所示。

所有数据表						
表	类型	键	长度	大小	多值	
tb_article	MyISAM	gb2312_chinese_ci	3.5 KB	-		
tb_comment	MyISAM	gb2312_chinese_ci	2.1 KB	-		
tb_friend	MyISAM	gb2312_chinese_ci	2.4 KB	-		
tb_public	MyISAM	gb2312_chinese_ci	2.1 KB	-		
tb_space	MyISAM	gb2312_chinese_ci	160.4 KB	-		
tb_user	MyISAM	gb2312_chinese_ci	2.2 KB	-		
总计	MyISAM	gb2312_chinese_ci	112.6 KB	8 字节		

图 25.7 db\_database25 数据库所包含的数据表

字段	类型	长度	属性	默认	额外
id	int(10)				auto_increment
content	mediumtext	gb2312_chinese_ci			
author	varchar(20)	gb2312_chinese_ci			
now	datetime			0000-00-00 00:00:00	
title	varchar(200)	gb2312_chinese_ci			

图 25.8 文章表结构

#### tb\_filecomment (评论表)

评论表存储的是用户对文章的评论，注册用户和游客都可以发表评论。tb\_filecomment 表的结构如图 25.9 所示。

**tb\_friend** (好友信息表)

好友信息表主要记录好友的姓名、性别、生日等好友的个人信息。tb\_friend 表的结构如图 25.10 所示。

字段	类型	是否	默认	额外
id	int(4)	否	auto_increment	
name	varchar(50)	否	gb2312_chinese_ci	
sex	varchar(10)	否	gb2312_chinese_ci	
bir	date	否	0000-00-00	
city	varchar(50)	否	gb2312_chinese_ci	
address	varchar(100)	否	gb2312_chinese_ci	
postcode	varchar(6)	否	gb2312_chinese_ci	
email	varchar(50)	否	gb2312_chinese_ci	
tel	varchar(20)	否	gb2312_chinese_ci	
bandset	varchar(20)	否	gb2312_chinese_ci	
qq	varchar(20)	否	gb2312_chinese_ci	
username	varchar(20)	否	gb2312_chinese_ci	

图 25.9 评论表结构

字段	类型	是否	默认	额外
id	int(4)	否		auto_increment
name	varchar(50)	否	gb2312_chinese_ci	
sex	varchar(10)	否	gb2312_chinese_ci	
bir	date	否	0000-00-00	
city	varchar(50)	否	gb2312_chinese_ci	
address	varchar(100)	否	gb2312_chinese_ci	
postcode	varchar(6)	否	gb2312_chinese_ci	
email	varchar(50)	否	gb2312_chinese_ci	
tel	varchar(20)	否	gb2312_chinese_ci	
bandset	varchar(20)	否	gb2312_chinese_ci	
qq	varchar(20)	否	gb2312_chinese_ci	
username	varchar(20)	否	gb2312_chinese_ci	

图 25.10 好友信息表结构

**tb\_public** (公告栏信息表)

公告栏信息表主要记录了网站情况、博客系统的版本情况或网站活动等内容。公告栏信息表的结构如图 25.11 所示。

**tb\_tpsc** (图片信息表)

图片信息表存储的是上传图片的信息，如图片名称、上传用户、上传时间等。tb\_tpsc 表的结构如图 25.12 所示。

字段	类型	是否	默认	额外
id	int(4)	否	auto_increment	
title	varchar(50)	否	gb2312_chinese_ci	
content	varchar(200)	否	gb2312_chinese_ci	
pub_time	date	否		

图 25.11 公告栏信息表结构

字段	类型	是否	默认	额外
id	int(10)	否		auto_increment
tpmc	varchar(30)	否	gb2312_chinese_ci	
file	blob	是	BINARY	否
author	varchar(20)	否	gb2312_chinese_ci	
scqj	date	否	0000-00-00	

图 25.12 图片信息表结构

**tb\_user** (用户信息表)

用户信息表主要存储用户的个人信息。tb\_user 表的结构如图 25.13 所示。

字段	类型	是否	默认	额外
id	int(20)	否		auto_increment
regname	varchar(20)	否	gb2312_chinese_ci	
regrealname	varchar(20)	否	gb2312_chinese_ci	
regpasswd	varchar(40)	否	gb2312_chinese_ci	
regbirthday	date	否	0000-00-00	
regemail	varchar(100)	否	gb2312_chinese_ci	
regcity	varchar(100)	否	gb2312_chinese_ci	
regico	varchar(50)	否	gb2312_chinese_ci	
regsex	varchar(4)	否	gb2312_chinese_ci	
regqq	varchar(40)	否	gb2312_chinese_ci	
reghomepage	varchar(100)	否	gb2312_chinese_ci	
regsign	varchar(200)	否	gb2312_chinese_ci	
regintroduce	mediumtext	否	gb2312_chinese_ci	
ip	varchar(20)	否	gb2312_chinese_ci	
flg	int(1)	否	0	

图 25.13 用户信息表结构

## 25.5 首页设计

首页是整个博客的入口地址，合理的首页布局和模块分配直接影响到用户对博客的第一印象。为了吸引更多的用户浏览和使用博客，在开发项目时需要精心地对首页进行设计。

### 25.5.1 首页概述

本博客系统在页面设计上采用传统的排版方式，主要由首部导航栏、左侧显示区和主显示区 3 部分组成，如图 25.14 所示。

- 首部导航栏：包括导航链接、用户登录等模块。
- 左侧显示区：包括最新文章、最新图片和系统时间等模块。
- 主显示区：为系统公告栏，显示系统及网站的最新咨询等信息。

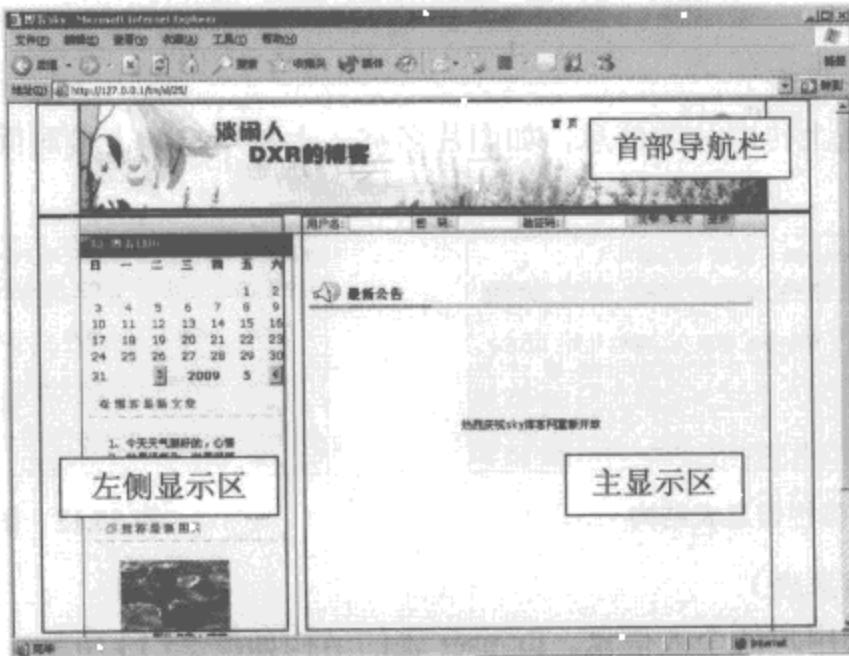


图 25.14 博客系统首页结构

### 25.5.2 首页技术分析

本博客首页的主显示区是一个可以自下而上移动的公告栏。公告栏主要用于公布系统版本的更新或升级信息、网站的最新活动安排等内容，也可以链接一些用户的精彩文章。本系统的公告栏可以上下移动，主要是应用 HTML 语言的<marquee>标记实现。使用<marquee>标签可以实现文字或图片的滚动效果，增强了实用性的同时，也增强了观赏性。下面将介绍这个标签的常用属性和方法。

#### 1. <marquee>标签的文字移动属性

<marquee>标签的特点就是可以使文字或图片上下移动。早期，这是一个了不起的技术，随着 W3C 标准的逐渐完善，<marquee>标签已经少有用武之地了。但有些技术是永远不变的，如对于<marquee>

标签中的属性名称和属性值，了解 CSS 样式表和 JavaScript 脚本语言的人会感觉到非常熟悉，因为这些属性名称和属性值都经常被提及和使用。<marquee>标签常用的文字移动属性及属性值如表 25.1 所示。

表 25.1 &lt;marquee&gt;标签常用的文字移动属性及说明

属性名称	属性值	属性说明	应用举例
marquee	无	标签基本语法，除文字以外，还可以是图片	<marquee>你好，PHP</marquee>
direction	left,right,up,down	文字移动属性，分别表示：从右往左、从左往右、从下到上、从上到下	<marquee direction="up">从下到上移动</marquee>
behavior	scroll,slide,alternate	文字移动方式，分别表示：沿同一方向不停地滚动、只滚动一次、在两个边界内来回滚动	<marquee behavior="scroll">不停地循环播放</marquee>
loop	数值 1,2,3…	循环次数，不指定则表示为无限循环	<marquee loop=5 behavior=slide>只循环滚动 3 次</marquee>
scrollmount	数值 10,20,30…	滚动速度，数值越大，速度越快	<marquee scrollmount=25>我漂起来了！！</marquee>
Scrolldelay	数值 100,200,300…	延时，每动一次，停止的时间	<marquee scrolldelay="500" scrollamount=100>停停走走</marquee>

### 2. <marquee>标签的外观设置

<marquee>标签不仅可以设置文字移动效果，还能设置标签的外观。外观包括对齐方式、标签底色、面积等，<marquee>标签常用的外观属性及说明如表 25.2 所示。

表 25.2 &lt;marquee&gt;标签常用的外观属性及说明

属性名称	属性值	属性说明	应用举例
align	top,middle,bottom	指定文本的对齐方式	<marquee align="middle">我在中间</marquee>
bgcolor	颜色值 "#000000"	滚动栏颜色	<marquee bgcolor="#DEEBEF">看到了吗？</marquee>
height	数值 10,20,30…	滚动栏高度	<marquee height="15">不要，太矮了</marquee>
width	数值 10,20,30…	滚动栏宽度	<marquee width="100">哇！好大的房子啊</marquee>

### 3. <marquee>标签的其他属性

除了以上的常规设置，<marquee>标签还有两个特殊的属性：start() 和 stop()。JavaScript 脚本语言与<marquee>标签一起使用，可以制作出非常别致的页面效果，例如：

```
onMouseOut="this.start()"; //当鼠标移出该区域时，开始滚动
onMouseOver="this.stop()"; //当鼠标进入该区域时，停止滚动
```

博客系统首部导航区域的用户登录模块中使用了汉字验证码，这样可以有效地防止恶意程序采用试探的方式登录本系统，从而能够提高系统的安全性。验证码的种类和实现方式有多种，如数字验证码、图片验证码和汉字验证码等，在本博客的用户登录模块中使用汉字图片验证码，实现思路是：首先定义一个一维数组，该数组的内容即为所要随机显示的汉字图片所对应的汉字，然后通过 for 循环随机输出 4 幅汉字图片，同时定义一个变量将这 4 幅汉字图片所对应的汉字连接成一个字符串，这样就可以记录汉字验证码的值，然后即可使用该值与用户在登录表单中所输入的验证码的值进行比较，进

而判断用户所输入的验证码是否正确，代码如下。

```
<?php
$str=array("大","更","创","天","科","客","博","技","立","新"); //汉字数组
$word=strlen($str);
for($i=0;$i<4;$i++){
    $num=rand(0,$word*2-1); //随机获取汉字数组的键值
    $img=$img."<img src='images/checkcode/".$num.".gif' width='16' height='16'>"; //显示随机图片
    $pic=$pic.$str[$num]; //将图片转换成数组中的文字
}
?>
```

### 25.5.3 首页的实现过程

#### ■ 视频讲解：光盘\TM\Video\25\首页的实现过程.exe

(1) 使用 session\_start() 函数启用 Session 支持，并使用 include 语句包含数据库连接文件，从而在首页中导入与数据库的连接代码。

```
<?php
session_start(); //开启 Session 功能
include "Conn/conn.php"; //包含数据库连接文件
?>
```

(2) 建立用户登录表单，用于录入用户的登录信息，表单及表单结构如表 25.3 所示。

表 25.3 表单及表单元素

元素类型	元素名称	属性设置	说明
表单	form	name="form" method="post" action="checkuser.php"	表单
文本域	txt_user	type="text" name="txt_user" size="10"	用户名
密码域	txt_pwd	name="txt_pwd" type="password" style="FONT-SIZE: 9pt; WIDTH: 65px" size="6"	密码
文本域	txt_yan	name="txt_yan" style="FONT-SIZE: 9pt; WIDTH: 65px" size="8"	输入验证码
隐藏域	txt_hyan	type="hidden" name="txt_hyan" id="txt_hyan" value=" php echo \$pic;?&gt;"</td <td>保存随机生成的验证码内容</td>	保存随机生成的验证码内容
提交按钮	sub_dl	style="FONT-SIZE: 9pt" type="submit" value="登录" name="sub_dl" onClick="return f_check(form)"	提交按钮

(3) 在左侧显示区内显示最新的 5 篇文章的主题，代码如下。

```
<?php
$sql=mysql_query("select id,title from tb_article order by now desc limit 5"); //查询最新 5 篇文章
$i=1;
while($info=mysql_fetch_array($sql)) //循环显示文章主题
{
?>
<tr>
    <td width="201" align="left" valign="top">&ampnbsp&ampnbsp&ampnbsp<a href="article.php?file_id=<?php echo
```

```
$info[id];?>" target="_blank"><?php echo $i."、".substr($info[title],0,20);?></a>
        </td>
    </tr>
<?php
    $i=$i+1;
}
?>
```

(4) 在左侧显示区内显示用户最新上传的两幅图片或照片, 代码如下。

```
<?php
    $sql=mysql_query("select id,tpmc,file from tb_tpsc order by scsj desc limit 2"); //查询最新两幅图片
    while($info=mysql_fetch_array($sql)) //循环显示图片
    {
?
<tr>
    <td width="9" rowspan="2" align="center">&ampnbsp</td>
    <td width="147" align="center">
        <a href="image.php?recid=<?php echo $info[id]; ?>" target="_blank">
            
        </a>
    </td>
    <td width="10" rowspan="2" align="center">&ampnbsp</td>
</tr>
<tr>
    <td align="center">图片名称: <?php echo $info[tpmc];?></td>
</tr>
<?php
    }
?
?>
```

## 25.6 用户管理模块设计

### 25.6.1 用户管理模块概述

博客活动的主体是用户, 博客中的文章、图片等内容都具有自己的拥有者, 为了定位这些内容的所属用户和更好地管理博客中的内容, 一个合理的用户管理模块是必不可少的。本博客的用户管理模块主要包括用户注册、用户登录和用户信息管理等部分。

### 25.6.2 用户注册

#### ■**视频讲解:** 光盘\TM\Video\25\用户注册.exe

通过用户注册的方式可以将用户的信息保存起来、跟踪不同站内活动的参与者。同样, 本博客只为普通浏览者提供了最基本的浏览权限, 如果用户想使用该系统发表自己的文章或上传照片, 首先需

要通过本系统的注册模块进行注册。具体注册流程如下：

(1) 单击博客首部导航栏的“博客注册”超链接，即可进入注册协议许可页面。用户首先需要阅读该页面中的协议内容，之后单击“同意以上条款”按钮即可进行下一步操作。

(2) 用户单击注册许可协议页面的“同意以上条款”按钮后，将进入用户信息录入页面，如图 25.15 所示。

The screenshot shows a registration form with the following fields:

- 必填内容:
  - 用户名:
  - [检测用户名]
  - 真实姓名:
  - 密码:
  - 确认密码:
  - 出生日期:  (日期格式为: yyyy-mm-dd)
  - Email:
  - 所在城市:
  - 选择头像:
  - 
  - 头像:
- 选填内容:
  - 性别:
  - QQ:

图 25.15 录入用户注册信息

用户按要求填写个人的注册信息后，单击“注册”按钮即可完成注册，这时系统自动为用户提供了个人空间，用户可以发表文章、上传图片。

为了区分用户，要求博客中所有注册用户的用户名不能相同，这就需要在用户填写注册信息的同时检测用户名是否被他人占用，本博客中使用弹出窗口的方式检测用户名是否已经被占用。首先在注册表单中添加用于弹出提示窗口的代码，如下所示。

```
<a href="#" onClick="javascript:openwin(myform.txt_regname.value)">[检测用户名]</a>
```

上述代码定义 JavaScript 脚本语言的 openwin()方法，并将“用户名”文本框中的内容作为函数的参数值，该函数的定义方法如下：

```
function openwin(x){
    if(x==""){
        //判断是否已经输入了用户名
        alert("请输入用户名!"); myform.txt_regname.focus();return false;
    }else{
        window.open("submit_checkuser.php?x="+x,"newframe","width=300,height=150");
        //使用 window 的 open 方法弹出对话框
    }
}
```

用户注册模块应用 JavaScript 脚本的 window 对象弹出新窗口以显示用户名是否合法。window 对象主要用于在 HTML 中打开窗口，应用极为普遍。用户浏览器决定了窗口的外观，设计者无法更改其窗口的大小及样式，但 JavaScript 给了程序这种控制权，在 JavaScript 中，可以使用 window 对象来实现对窗口的控制。window 对象的常用方法如表 25.4 所示。

表 25.4 window 对象的常用方法

方 法	说 明
alert()	弹出一个警告对话框
confirm()	弹出一个确认对话框，并在确认对话框中显示指定的字符串
prompt()	弹出一个提示对话框
close()	关闭被引用的窗口
focus()	将被引用的窗口放在所有打开窗口的前面
open()	打开新浏览器窗口并显示由 URL 或名字引用的文档，并设置创建窗口的属性
resized(x,y)	设置窗口的大小
resizeBy(offsetx,offsety)	按照指定的位移量设置窗口的大小

下面对 window 对象的 open() 方法进行讲解。

使用 JavaScript 的 open() 方法打开新浏览器窗口，可以基于特定的条件创建带有被装入特定文档的新窗口，也可以指定如新窗口的大小以及窗口中可用的选项，并且可以为引用它的窗口指派名字。

使用 window 对象打开窗口的语法格式如下：

```
windowVar=window.open(url,windowname[,location]);
```

参数说明如下。

- ☒ windowVar：当前打开窗口的句柄。如果 open() 方法成功，则 windowVar 的值为一个 window 对象的句柄，否则 windowVar 的值是一个空值。
- ☒ url：目标窗口的 URL。如果 URL 是一个空字符串，则浏览器将打开一个空白窗口，允许用 write() 方法创建动态 HTML。
- ☒ windowname：window 对象的名称。
- ☒ location：可选参数，对于窗口属性进行设置，其参数如表 25.5 所示。

表 25.5 对窗口属性进行设置的可选参数

属 性	说 明
width	窗口的宽度
height	窗口的高度
scrollbars	是否显示滚动条
resizable	设定窗口大小是否固定
toolbar	浏览器工具条，包括后退及前进按钮等
menubar	菜单条，一般包括文件、编辑以及其他一些条目
location	定位区，也叫地址栏，是可以输入 URL 的浏览器文本区
direction	更新信息的按钮

单击“提交”按钮后，用户所录入的注册信息将被提交到 register\_deal.php 文件中，该文件用于实现将用户的注册信息保存到数据库中。代码如下：

```
<?php
session_start();
include "Conn/conn.php";
$UserName=$_POST[txt_regname];
//开启 Session 支持
//包含数据库连接文件
//接收用户名，并判断用户名是否已经被注册
```

```

$Sql=mysql_query("select * from tb_user where regname = '$UserName'");
$result=mysql_fetch_array($Sql);
if ($result!=false){ //如果用户名已经被注册则给出提示
    echo ("<script>alert('用户名已被注册！');history.go(-1);</script>");
    exit();
}
$_SESSION[username]=$_POST[txt_regname]; //将用户名保存到 Session 中
$regname=$_POST[txt_regname]; //接收表单内容
$regrealname=$_POST[txt_Regrealname];
$regpwd=md5($_POST[txt_Regpwd]);
$regbirthday=$_POST[txt_birthday];
$regemail=$_POST[txt_Regemail];
$regcity=$_POST[txt_Province].$_POST[txt_City];
$regico=$_POST[txt_ico];
$regsex=$_POST[txt_Regsex];
$regqq=$_POST[txt_Regqq];
$reghomepage=$_POST[txt_Reghomepage];
$regsign=$_POST[txt_Regsign];
$regintroduce=$_POST[txt_Regintroduce];
$ip=getenv(REMOTE_ADDR); //获取用户 IP
$INS=mysql_query("Insert Into tb_user (regname, regrealname, regpwd, regbirthday, regemail, regcity, regico, regsex, regqq, reghomepage, regsign, regintroduce, ip, fig) Values ('$regname', '$regrealname', '$regpwd', '$regbirthday', '$regemail', '$regcity', '$regico', '$regsex', '$regqq', '$reghomepage', '$regsign', '$regintroduce', '$ip', 0)");
echo "<script> alert('用户注册成功！');</script>"; //提示用户注册成功
echo "<script> window.location='file.php';</script>"; //重新定位到 file.php 页面
?>

```

### 25.6.3 用户登录

#### 视频讲解：光盘\TM\Video\25\用户登录.exe

如果用户已经注册为本博客的会员，那么在发表文章或上传图片时需要进行登录。本博客的登录页面如图 25.16 所示。

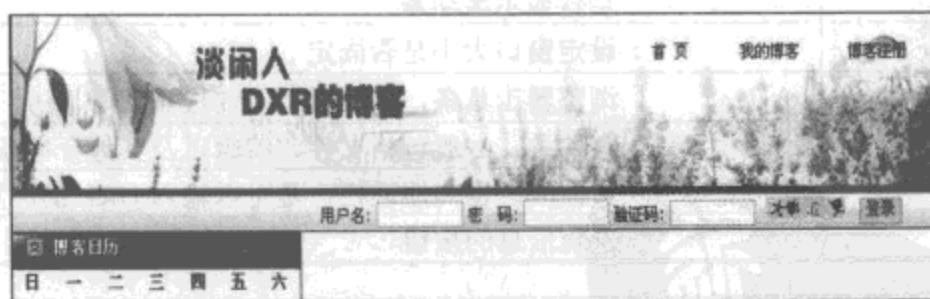


图 25.16 用户登录页面

在介绍首页的实现过程时，已经详细介绍了用户登录表单及表单元素，这里不再赘述。当用户填写完个人信息后，单击“登录”按钮即可将用户录入的表单信息提交到 chkuser.php 页面，该页面用于对用户的登录信息进行验证，代码如下。

```

<?php
session_start();
include "Conn/conn.php";
$name=$_POST[txt_user];
$pwd=md5($_POST[txt_pwd]);
$sql=mysql_query("select * from tb_user where regname='".$name."' and regpwd='".$pwd."'");
$result=mysql_fetch_array($sql);
if($result!=""){
    $_SESSION[fig]=$result[fig]; //启用 Session 支持
    $_SESSION[username]=$name; //包含数据库连接文件
    $_SESSION[password]=$pwd; //获取表单提交的用户名
    $_SESSION[hash]=$pwd; //获取表单提交的密码，并用 MD5 加密
    $sql=mysql_query("select * from tb_user where regname='".$name."' and regpwd='".$pwd."'");
    $result=mysql_fetch_array($sql);
    if($result!=""){
        $_SESSION[fig]=$result[fig]; //将用户名保存到 Session 中
        $_SESSION[username]=$name;
    }
}
?>
<script language="javascript">
    alert("登录成功");window.location.href="file.php"; //重新定向到 file.php
</script>
<?php
}else{
?>
<script language="javascript">
    alert("对不起，您输入的用户名或密码不正确，请重新输入!");window.location.href="index.php";
    //如果密码错误则给出提示，并重新定向到首页
</script>
<?php
    }
?>

```

上述代码中首先用 `$_POST` 全局数组接收用户提交的登录信息，并使用函数 `md5()` 加密用户的登录密码，之后查询用户表中是否存在与用户提交的内容相同的用户名和密码，如果存在则说明用户提交的登录信息正确，并将用户名保存到 Session 中，反之，弹出提示对话框提示密码输入错误，并重新定位到博客首页。

#### 25.6.4 查询用户信息

##### 视频讲解：光盘\TM\Video\25\查询用户信息.exe

随着时间的推移，博客中注册用户也会随之不断的增长，那么在后台用户管理模块中，管理员如何才能够迅速定位到指定的用户呢？建立一个查询模块即是迅速定位到指定用户的最为有效的方式之一。本博客系统中实现用户查询的步骤如下：

(1) 建立用户查询表单，该表单及表单元素详细说明如表 25.6 所示。

表 25.6 用户查询页面中的表单及表单元素

元素类型	元素名称	属性设置	含义
<input type="form"/>	myform	name="myform" method="post" action=""	用户查询表单
<input type="select"/>	sel_tj	name="sel_tj" id="sel_tj"	查询关键字类别
<input type="text"/>	sel_key	name="sel_key" type="text" id="sel_key" size="30"	查询关键字
<input type="submit"/>	Submit	name="Submit" value="检索" onClick="return check();"	提交按钮

(2) 用户单击“提交”按钮后，首先应该对查询表单中数据进行合法性验证，本模块中使用

JavaScript 在前台对用户所输入的内容进行验证，代码如下。

```
<script language="javascript">
function check(form){ //定义 check()函数
    if (document.myform.sel_key.value==""){ //判断关键字是否为空
        alert("请输入查询条件!");myform.sel_key.focus();return false;
    }
}
</script>
```

(3) 用户正确填写查询表单中的内容并单击“提交”按钮后，查询信息将被提交到当前页进行处理，并将查询结果显示在当前页，代码如下。

```
<?php
if (isset($_POST[sel_key])){ //判断是否单击“提交”按钮
    $tj=$_POST[sel_tj];
    $key=$_POST[sel_key];
    $sql=mysql_query("select * from tb_user where $tj='$key'"); //查询用户信息
    $result=mysql_fetch_array($sql); //获取查询结果集
    if($result==false){
        echo ("[<font color=red>Sorry!该用户不存在!</font>]");
    } else{
    ?>
<!--省略了部分代码-->
<?php
    }
    ?>
} //查询模块结束
?>
```

进入用户信息查询模块，首先选择查询条件并在“关键字”文本框中输入要查询的关键字，然后单击“检索”按钮将显示出查询结果。例如，查询条件选择“用户名”，关键字输入“Tsoft”，然后单击“检索”按钮，将在页面中显示名为 Tsoft 的用户信息，如图 25.17 所示。



图 25.17 查询用户信息

## 25.6.5 浏览用户信息

### ■ 视频讲解：光盘\TM\Video\25\浏览用户信息.exe

本博客不仅可以通过查询方式快速定位到指定的用户，而且可以通过分页的方式浏览所有用户信息，这样管理人员就可以通过分页链接浏览所有用户信息，便于对用户信息进行管理。本博客中用于浏览用户信息的代码如下：

```

<?php
    if ($page=="") {$page=1;}                                //默认显示第一页
?>
<table width="600" height="291" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td height="291" align="center" valign="top">
            <table width="600" border="1" align="center" cellpadding="3" cellspacing="1" bordercolor="#E1E2DD"
bgcolor="#FFFFFF">
                <tr align="left" colspan="2" >
                    <td width="390" height="25" colspan="3" valign="top" bgcolor="#E9EAES" > <span class=
"tableBorder_LTR"> 浏览用户信息</span> </td>
                </tr>
<?php
    if ($page){
        $page_size=2;                                         //每页显示两条记录
        $query="select count(*) as total from tb_user order by id desc";
        $result=mysql_query($query);                         //查询总的记录条数
        $message_count=mysql_result($result,0,"total");      //为变量赋值
        $page_count=ceil($message_count/$page_size);          //根据记录总数除以每页显示的记录数求出所分的页数
        $offset=($page-1)*$page_size;                         //计算下一页从第几条数据开始循环
        for ($i=1; $i<2; $i++) {
            if ($i==1) {
                $sql=mysql_query("select * from tb_user order by id desc limit $offset, $page_size");
                $result=mysql_fetch_array($sql);
            }
            do{
?>
        <tr>
            <td height="184" align="center" valign="middle" >
                <!--省略了部分代码-->
            </td>
        </tr>
<?php
    }while($result=mysql_fetch_array($sql));
?>
</table>
<table width="600" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr bgcolor="#EFF7DE">
        <td bgcolor="#E9EAES" >&ampnbsp&ampnbsp页次: <?php echo $page;?>/<?php echo $page_count;?>
        </td>
    </tr>
    <tr>
        <td>页记录: <?php echo $message_count;?> 条&ampnbsp</td>
    </tr>
</table>

```

```

<td align="right" bgcolor="#E9EAES" class="hongse01">
<?php
    if($page!=1){
        echo "<a href=browseuser.php?page=1>首页</a>&nbsp;";
        echo "<a href=browseuser.php?page=".($page-1).">上一页</a>&nbsp;";
    }
    if($page<$page_count)
    {
        echo "<a href=browseuser.php?page=".($page+1).">下一页</a>&nbsp;";
        echo "<a href=browseuser.php?page=".$page_count.">尾页</a>";
    }
}
?>

```

进入用户信息浏览模块，如图 25.18 所示，用户可以通过单击分页链接实现用户信息的分页显示。如果是管理员登录，还可以删除指定的用户信息。



图 25.18 浏览用户信息

## 25.7 文章管理模块设计

文章管理模块是博客系统中较为重要的模块之一，因为博客的一个核心功能就是为用户提供发表自己的文章的功能。所以，对博客的文章管理模块进行精心设计是必要的。

### 25.7.1 文章管理模块概述

本系统的文章管理模块包括“发表文章”、“查看文章”、“其他用户文章”、“发表评论”、“删除文章”和“删除评论”6项功能。其中，普通用户只能删除自己的文章及对文章进行评论，只有管理员才有权删除任何一篇文章及回复。文章管理模块的框架如图 25.19 所示。

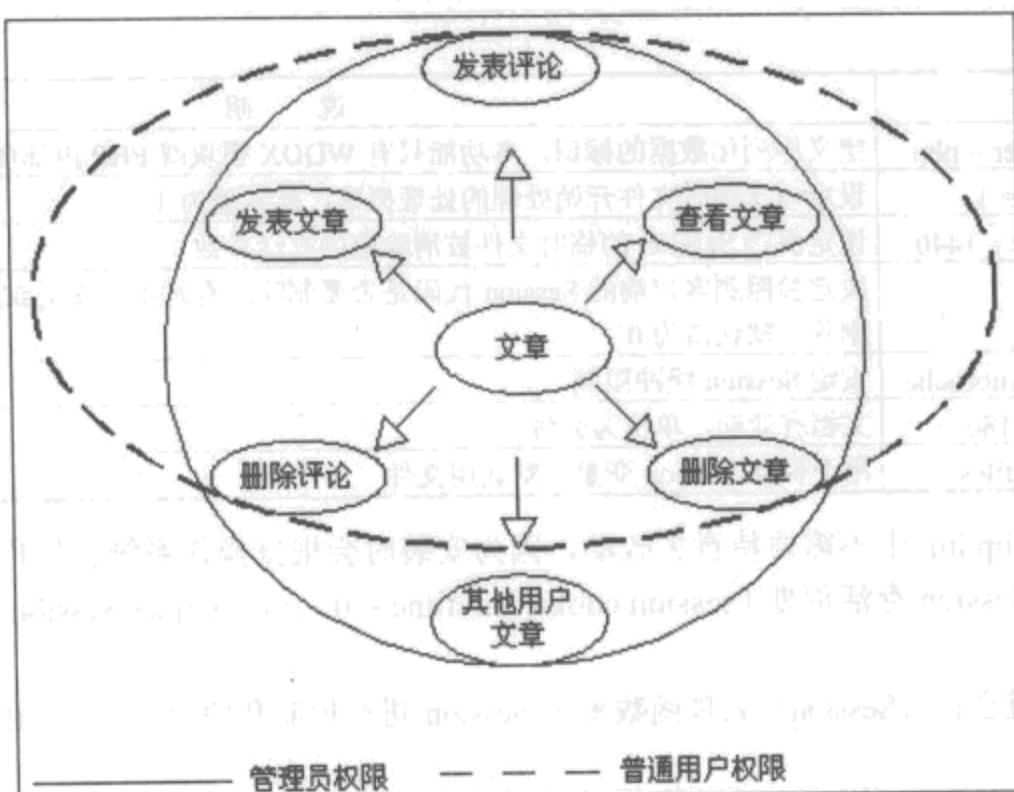


图 25.19 文章管理模块框架图

### 25.7.2 文章管理模块技术分析

#### ■ 视频讲解：光盘\TM\Video\25\文章管理模块分析.exe

大多数情况下，只有用户成功登录后才可以在论坛或博客中发表帖子或文章，这样用户可以有效地管理自己所发布的内容，而且可以定位不同文章的作者。25.7.1 节中已经介绍了不同用户具有不同的权限，那么如何为用户分配权限呢？有 Web 编程基础的读者都知道通过 Session 可以跟踪和定位访问 Web 服务器的不同用户，同样，PHP 也可以采用 Session 来保存用户的身份信息，在进行站内活动时，根据 Session 所保存的用户身份信息来判断用户是否具备某项操作的权限。

Session 的中文含义是“会话”，是指用户从进入网站开始，直到关闭网站这段时间内，所有网页共同使用的公共变量的存储机制。Session 比 Cookie 更有优势：Session 是存储在服务器端的，不易被伪造；Session 的存储没有长度限制；Session 的控制更容易、更安全。

程序开发人员要在程序编写过程中灵活运用 Session，首先应该掌握 Session 的相关配置，PHP 中对 Session 进行配置是在 php.ini 文件中进行的。表 25.7 为 PHP 中 Session 的相关配置。

表 25.7 Session 的常用配置选项

配置选项	说 明
session.save_path = c:/temp	保存 Session 变量的目录，在 Linux/UNIX 下为/tmp
session.uses_cookies = 1	是否使用 Cookie
session.name = PHPSESSID	表示会话 ID
session.auto_start = 0	是否自动启用 Session，当为 1 时，在每页中不必调用 Session_start() 函数
session.cookie_lifetime = 0	设定 Cookie 送到浏览器后的保存时间，单位为秒。默认值为 0，表示直到浏览器关闭
session.cookie_path = /	Cookie 有效路径
session.cookie_domain =	有效域名

续表

配置选项	说 明
session.serialize_handler = php	定义序列化数据的标识，本功能只有 WDDX 模块或 PHP 内部使用，默认值为 PHP
session.gc_probability = 1	设定每次临时文件开始处理的处理概率，默认值为 1
session.gc_maxlifetime = 1440	设定保存 Session 的临时文件被清除前的存活秒数
session.referer_check =	决定参照到客户端的 Session 代码是否要删除。有时出于安全或其他考虑，会设定不删除，默认值为 0
session.cache_limiter = nocache	设定 Session 缓冲限制
session.cache_expire = 180	文档有效期，单位为分钟
session.save_handler = files	用于保存 Session 变量，默认用文件

一般情况下，php.ini 中不需要特意去改动，因为安装时会根据操作系统自行作出适当的调整。只有少数的几项，如 Session 存活周期（session.cookie\_lifetime = 0）、自动开启 Session（session.auto\_start）等稍加改动即可。

PHP 主要是通过会话（Session）处理函数来对 Session 进行控制和使用的，常用的处理函数如表 25.8 所示。

表 25.8 PHP 常用的会话处理函数

函 数	函 数 说 明
session_start();	开启 Session 或返回已经存在的 Session
\$SESSION['name'] = value;	注册一个 Session 变量
session_id()	设定或取得当前的 session_id 值
isset(\$_SESSION['name'])	检测指定的 Session 值是否存在。isset 不只可以检测 Session，还可以检测其他类型，如 isset(\$_POST['name'])、isset(\$_GET['name']) 等
session_regenerate_id()	更改 session_id 的值
session_name()	返回或改变当前 Session 的名称
unset(\$SESSION['name'])	删除名为 name 的 Session
session_destroy()	结束当前会话，删除所有 Session

程序开发过程中，使用 Session 时要注意以下常见问题：

在使用 Session 之前一定要在页面开始使用 session\_start() 函数启用 Session 支持。

大多数读者在使用 Session 之前都能先调用 session\_start() 函数，但对于 session\_destroy() 函数却经常忽略。session\_destroy() 虽然用于结束当前会话并删除所有 Session，但在删除之前，也要先开启 Session 支持才可以，不然会产生这样的错误代码：

```
session_destroy() [function.session-destroy]: Trying to destroy uninitialized session in...
```

所以，凡是在使用 Session 及其函数的页面中，都要加上 session\_start() 这句话。

尽可能将 session\_start() 函数放到第一行。

因为在一些高版本的 PHP 中，如果不将 session\_start() 函数放到第一行，就很有可能产生如下错误提示：

```
Warning: session_start() [function.session-start]: Cannot send session cache limiter - headers already sent...
```

其原因就是在使用 session\_start() 函数之前，已经有页面输出。所以，为了避免这类错误的发生，

尽可能的将 session\_start() 函数放到第一行。

### 25.7.3 添加文章的实现过程

#### ■ 视频讲解：光盘\TM\Video\25\添加文章实现过程.exe

用户成功登录博客后，会自动跳转到添加文章页面，当然也可以通过单击“文章管理” / “添加博客文章”超链接返回到添加文章页面。文章添加页面的运行结果如图 25.20 所示。

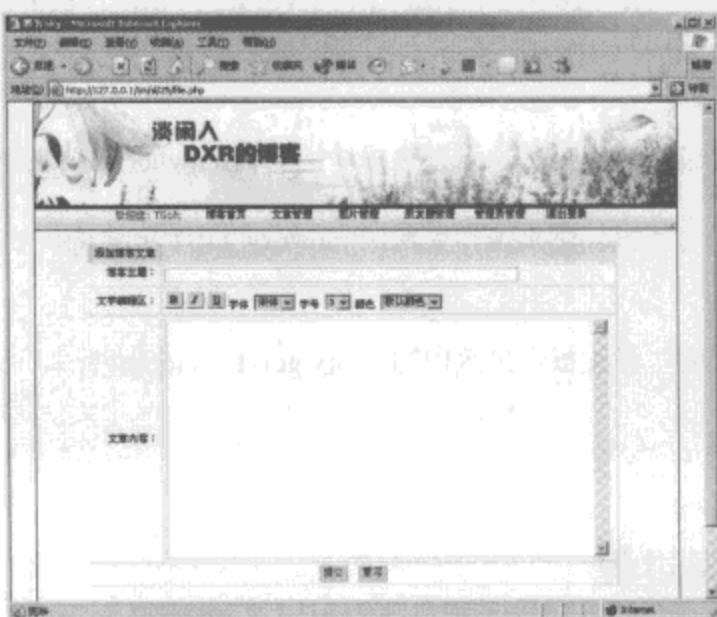


图 25.20 添加文章页面的运行结果

添加文章页面所涉及的表单及表单元素如表 25.9 所示。

表 25.9 添加文章页面中的表单及表单元素

元素类型	元素名称	属性设置	含义
<input type="form"/>	myform	method="post" action="check_file.php"	添加文章表单
<input type="text"/>	txt_title	id="txt_title" size="68"	文章标题
<input type="select"/>	font	name="font" class="wenbenkuang" id="font" onChange="showfont(this.options[this.selectedIndex].value)"	文章字体
<input type="select"/>	size	class="wenbenkuang" onChange="showsize(this.options[this.selectedIndex].value)"	字体大小
<input type="select"/>	color	onChange="showcolor(this.options[this.selectedIndex].value)" name="color" size="1" class="wenbenkuang" id="select"	字体颜色
<input type="textarea"/>	file	cols="75" rows="20" id="file" style="border:0px; width:520px;"	文章内容
<input type="submit"/>	btn_tj	id="btn_tj" value="提交" onClick="return check();"	提交按钮

用户填写完博客主题和文章内容后，单击“提交”按钮，系统将跳转到处理页（check\_file.php）进行处理。在处理页中，将传过来的文章标题、文章作者和文章内容等参数组成 insert 语句，并最终保存到数据表中。如果添加信息成功，系统返回到本页，可继续执行添加操作；如果添加失败，则返回到上一步，程序的关键代码如下：

```

<?php
session_start(); //开启 Session 支持
include "Conn/conn.php"; //包含数据库连接文件
if($btn_tj<>""){
    $title=$_POST[txt_title]; //判断传值页面
    $author=$_SESSION[username]; //取得文章标题
    $content=$_POST[file]; //取得作者
    $now=date("Y-m-d H:i:s"); //取得文章内容
    //使用 date 函数生成发布时间
    /* 生成 insert 语句 */
    $sql="Insert Into tb_article (title,content,author,now) Values ('$title','$content','$author','$now')";
    $result=mysql_query($sql); //执行 insert 语句
    /* 根据$result, 返回结果 */
    if($result){
        echo "<script>alert('恭喜您，您的文章发表成功!!!');window.location.href='file.php';</script>";
    }else{
        echo "<script>alert('对不起，添加操作失败!!!');history.go(-1);</script>";
    }
}else{
    echo "<script>alert('对不起，添加操作失败!!!');history.go(-1);</script>";
}
?>

```

#### 25.7.4 文章列表的实现过程

视频讲解：光盘\TM\Video\25\文章列表实现过程.exe

单击“文章管理”/“我的文章”超链接，将显示用户发表过的文章列表，文章列表页面（myfiles.php）的运行结果如图 25.21 所示。

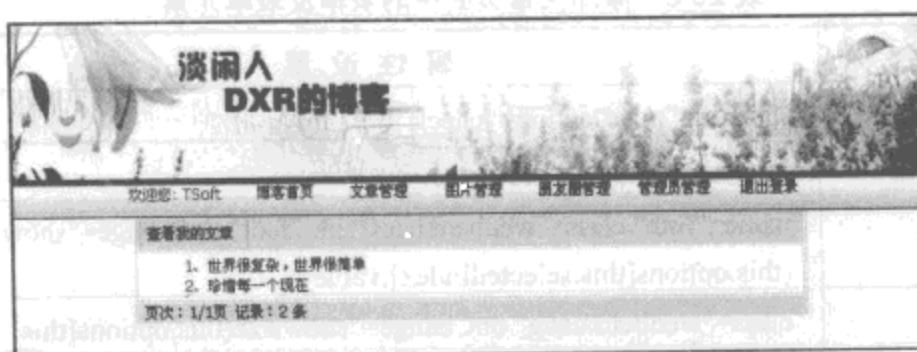


图 25.21 文章列表页面的运行结果

文章列表页面使用了分页技术和 do…while 循环语句来输出文章标题。程序关键代码如下：

```

<?php
session_start(); //开启 Session 支持
include "Conn/conn.php"; //包含数据库连接文件
include "check_login.php"; //包含权限检查文件
?>
<!-- 分页 -->
<?php
/* $page 为当前页，如果$page 为空，则初始化为 1 */

```

```

if ($page==""){
    $page=1;
}
if (is_numeric($page)){
    $page_size=20;                                //每页显示 20 条记录
    $query="select count(*) as total from tb_article where author = '".$_SESSION[username]."' order by id desc";
    $result=mysql_query($query);                  //查询符合条件的记录总数
    $message_count=mysql_result($result,0,"total"); //要显示的总记录数
    $page_count=ceil($message_count/$page_size);   //根据记录总数除以每页显示的记录数求出所分的页数
    $offset=($page-1)*$page_size;                  //计算下一页从第几条数据开始循环
    $sql=mysql_query("select id,title from tb_article where author = '".$_SESSION[username]."' order by id desc limit $offset, $page_size");
    $info=mysql_fetch_array($sql);
?>
...
<?php
/* 输出结果集 */
if($info){
    $i=1;                                         //文章序号
    do{                                           //do...while 循环开始
?>
<tr>
<!-- 显示文章序号和文章标题 -->
<td width="498" align="left" valign="top"> &ampnbsp&ampnbsp&ampnbsp<a href="showmy.php?file_id=<?php echo $info[id];?>"><?php echo $i."、".$info[title];?></a> </td>
</tr>
<?php
    $i=$i+1;                                     //循环结束
?>
</table></td>
</tr>
</table></td>
</tr>
<?php } ?>
...
<!-- 翻页条 -->
<td width="33%">&ampnbsp&ampnbsp页次: <?php echo $page;?>/<?php echo $page_count;?>页&ampnbsp记录: <?php echo $message_count;?> 条&ampnbsp </td>
<td width="67%" align="right" class="hongse01">
<?php
/* 如果当前页不是首页 */
if($page!=1){
/* 显示“首页”超链接 */
echo "<a href=myfiles.php?page=1>首页</a>&ampnbsp";
/* 显示“上一页”超链接 */
echo "<a href=myfiles.php?page=".($page-1).">上一页</a>&ampnbsp";
}
/* 如果当前页不是尾页 */
if($page<$page_count)

```

```

{
/* 显示“下一页”超链接 */
echo "<a href=myfiles.php?page=".($page+1).">下一页</a>&nbsp;";
/* 显示“尾页”超链接 */
echo "<a href=myfiles.php?page=$page_count.>尾页</a>";
}
?>

```

上述代码中使用 `is_numeric()` 函数判断变量 `$page` 是否为数字，如果是数字则返回 `true`，否则返回 `false`。如果一组字符全部由数字组成，那么 `is_numeric()` 函数也将返回 `true`。

在编写上述代码时，要注意 `do...while()` 循环和 `while()` 循环的区别。`do...while()` 循环是先执行 “{}” 中的代码段，然后判断 `while` 中的条件表达式是否成立，如果返回 `true`，则重复输出 “{}” 中的内容，否则结束循环，执行 `while` 下面的语句。`while` 循环是先判断 `while` 中的表达式，当返回 `true` 时，再执行 “{}” 中的代码。所以在上述代码中，如果是使用 `while` 循环，那么第一条记录将不被输出。

### 25.7.5 查看、评论文章的实现过程

#### ■ 视频讲解：光盘\TM\Video\25\查看、评论文章的实现过程.exe

单击列表中任意一个文章标题，都会看到对应的文章内容和文章评论，查看文章页面（`showmy.php`）的运行结果如图 25.22 所示。

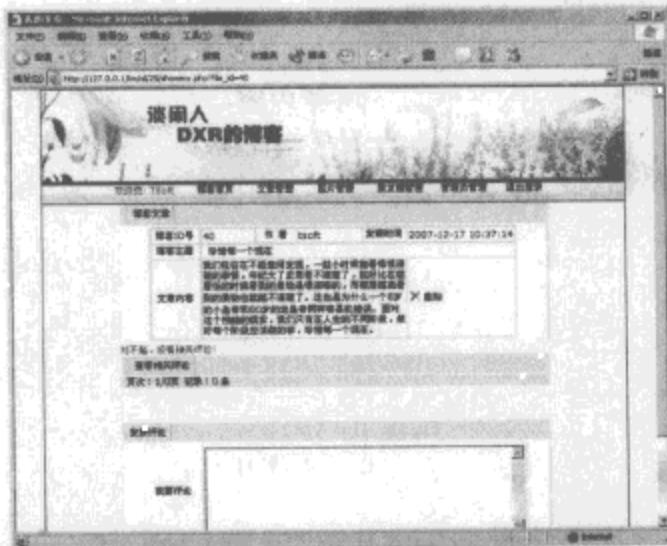


图 25.22 查看文章页面的运行结果

程序根据前页面传过来的文章 ID 值从数据表 `tb_article` 中返回对应的文章信息（包括文章 ID、文章作者、文章标题、文章内容和发表时间）、输出文章信息后，开始查找表 `tb_filecomment` 中 `file_id` 字段值等于文章 ID 的所有评论集，并通过分页显示出来。显示文章页面（`showmy.php`）的关键代码如下：

```

<?php
session_start();
include "Conn/conn.php";
include "check_login.php";
$file_id1=$_GET[file_id];
$bool = false;
//开启 Session 支持
//包含数据库连接文件
//包含权限检查文件
//取得文章 ID 号
//判断用户删除权限，默认的 false 是没有权限

```

```

?>
    <!-- 首部导航栏和文章显示区 -->
<?php
    $sql=mysql_query("select * from tb_article where id = '$file_id1'); //根据文章 ID 号查找文章
    $result=mysql_fetch_array($sql); //返回结果集
?>
<!-- ----- 显示文章 ----- -->





```

```

cellspacing="1" bordercolor="#D6E7A5" bgcolor="#FFFFFF" class="i_table">
<tr bgcolor="#FFFFFF">
    <td width="14%" align="center">评论 ID 号</td>
    <!-- -----评论 id----- -->
    <td width="15%"><?php echo $result[id]; ?></td>
    <td width="11%" align="center">评论人</td>
    <!-- -----评论人----- -->
    <td width="18%"><?php echo $result[username]; ?></td>
    <td width="12%" align="center">评论时间</td>
    <!-- -----评论时间----- -->
    <td width="30%"><?php echo $result[datetime]; ?></td>
</tr>
<tr bgcolor="#FFFFFF">
    <td align="center">评论内容</td>
    <!-- -----评论内容----- -->
    <td colspan="4"><?php echo $result[content]; ?></td>
    <td>
<?php
    /* 如果$bool 为真，输出删除超链接，否则跳过 */
if ($bool){
?>
    <a href="del_comment.php?comment_id=<?php echo $result[id]; ?>"></a>
<?php }?>
</td></tr> </table></td> </tr>
<?php
    }while($result=mysql_fetch_array($sql));
}
?>
</table>
<!-- 翻页功能代码部分略 -->

```

### 25.7.6 删除文章或评论的实现过程

#### 视频讲解：光盘\TM\Video\25\删除文章或评论的实现过程.exe

在查看文章评论页面，当系统判定当前用户为管理员或是文章作者时，在每篇文章和评论的后面都将显示相应的“删除”按钮。单击任一个“删除”按钮，系统会提示是否删除，如果确认，将跳转到处理页（`del_file.php` 和 `del_comment.php`），完成删除操作。

在删除文章的处理页中，删除文章的同时也删除了该篇文章的相关评论。处理页首先在文章列表（`tb_article`）中删除 ID 等于`$file_id` 的记录，如果没有可删除记录，则提示失败，并返回上一步。如果删除成功，则转到评论列表（`tb_filecomment`）中，删除所有该篇文章的评论。删除文章页（`del_file.php`）的关键代码如下：

```

<?php
    session_start();
    include "check_login.php";

```

//开启 Session 支持  
//包含权限检查文件

```

include "Conn/conn.php"; //包含数据库连接文件
$sql="delete from tb_article where id=".$file_id; //删除文章的 delete 语句
$result=mysql_query($sql,$link); //执行删除操作
if($result){
    $sql1 = "delete from tb_filecomment where fileid = ".$file_id; //删除相对应的文章评论
    $rst1 = mysql_query($sql1,$link); //执行删除评论的操作
    if($rst1)
        echo "<script>alert('博客文章已被删除!');location='myfiles.php'];</script>";
    else
        echo "<script>alert('删除失败!');history.go(-1);</script>";
}
else{
    echo "<script>alert('博客文章删除操作失败!');history.go(-1);</script>";
}
?>

```

因为删除文章的过程包含了删除评论的过程，所以这里不再给出删除评论的代码，相关内容可参见本书附带光盘。

## 25.8 图片上传模块设计

用户不仅可以在博客中发表自己的文章或日志，而且可以将个人具有纪念意义的照片发布到博客中。所以，在博客中，一个高效、人性化的图片上传模块是必不可少的。

### 25.8.1 图片上传模块概述

本系统的图片上传模块主要实现对图片的添加、浏览、查询和删除操作。对图片的删除则只有管理员才有权限。图片上传模块框架图如图 25.23 所示。

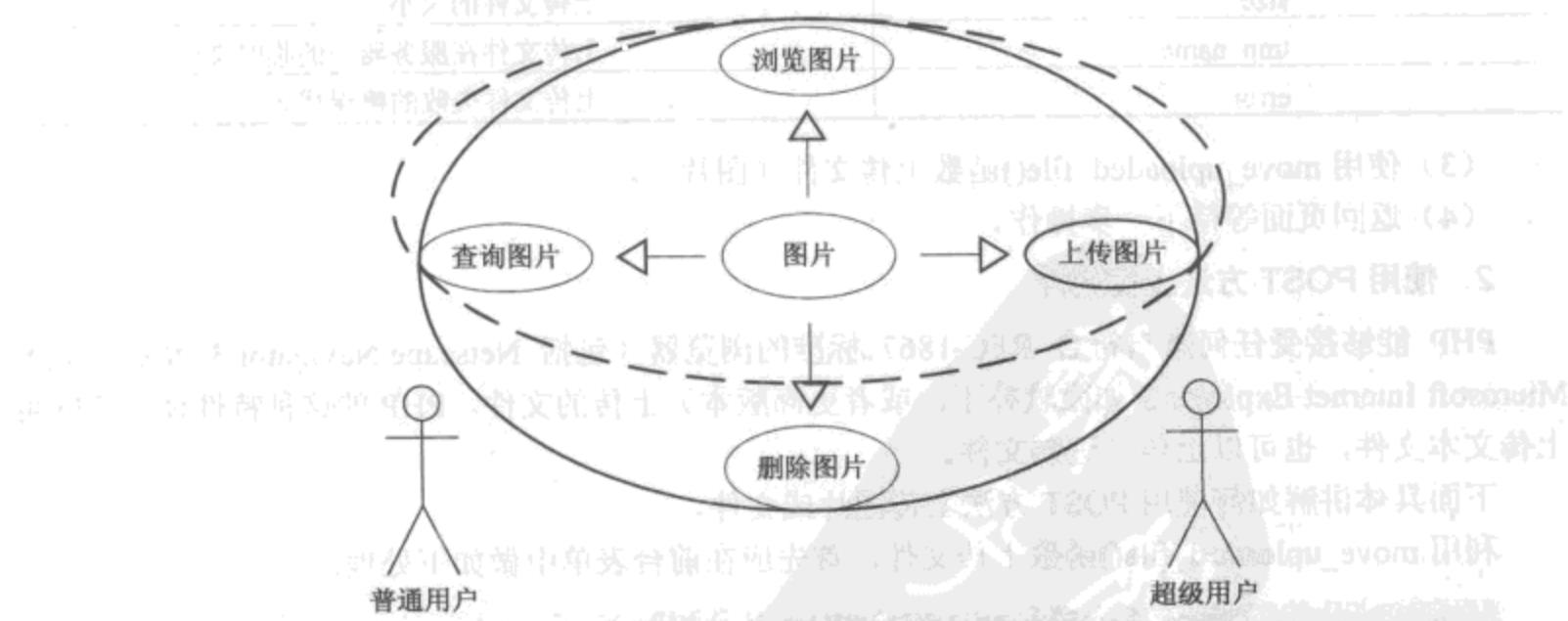


图 25.23 图片上传模块框架图

## 25.8.2 图片上传模块技术分析

### 视频讲解：光盘\TM\Video\25\图片上传模块技术分析.exe

博客中，用户除了可以发表文章，还可以将自己的照片或喜爱的图片发布到网上，保存到自己的博客中。文字信息可以通过简单的表单保存到数据库中，那么图片该如何从本地保存到服务器中呢？与保存文件类似，图片可以通过上传的方式保存到服务器某目录中，也可以转换为二进制数据保存到数据库中。

### 1. 上传图片的基本流程

上传图片到服务器既可以通过 FTP 实现，也可以通过 POST 方法实现。本系统中使用 POST 方法实现，采用这种方法更加方便，具体实现步骤如下：

(1) 在<form>表单中定义文件选择域，用来选择要上传的图片。这里要注意，通过 POST 方法上传文件或图片时，一定要在 form 表单中加上属性 enctype="multipart/form-data"，否则上传不了文件（图片）。

(2) 在处理文件上传的页面中，使用\$\_FILES 全局数组来判断上传文件类型和上传文件（图片）大小是否符合要求。

\$\_FILES 数组为系统预定义的全局数组，保存的是上传文件（图片）的相关属性。使用格式为：

```
$_FILES[name][property];
```

其中，第一维的键值 name 要与上传表单中的文件选择域名相同，第二维的键值 property 的取值说明如表 25.10 所示。

表 25.10 \$\_FILES 的相关属性

属性 值	说 明
type	上传文件的类型
size	上传文件的大小
tmp_name	上传文件在服务器中的临时文件名
error	上传文件失败的错误代码

(3) 使用 move\_uploaded\_file()函数上传文件（图片）。

(4) 返回页面等待下一步操作。

### 2. 使用 POST 方法上传图片

PHP 能够接受任何来自符合 RFC-1867 标准的浏览器（包括 Netscape Navigator 3 及更高版本；Microsoft Internet Explorer 3 加微软补丁，或者更高版本）上传的文件。PHP 的这种特性使用户既可以上传文本文件，也可以上传二进制文件。

下面具体讲解如何使用 POST 方法上传图片或文件。

利用 move\_uploaded\_file()函数上传文件，首先应在前台表单中做如下处理：

```
<form enctype="multipart/form-data" action="URL" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="30000">
```

```
上传文件路径: <input name="file" type="file">
<input type="submit" value="上传">
</form>
```

上述代码中的 URL 应该替换成指向一个 PHP 文件的真实提交地址，在该文件中将使用 move\_uploaded\_file() 函数进行文件上传。

取值为 MAX\_FILE\_SIZE 的隐藏域（单位为字节）必须先于文件选择框，其值为可以接收的文件的最大尺寸。同时，要保证文件上传表单中有 enctype="multipart/form-data" 的属性，否则文件上传将不能实现。

MAX\_FILE\_SIZE 的值只是对浏览器的一个建议，实际上可以不进行设置，php.ini 文件中已经设置了允许上传文件的最大值。但最好还是在表单中加上 MAX\_FILE\_SIZE，因为它可以避免用户在长时间等待大文件上传之后才发现该文件过大的麻烦。

为上传文件定义的变量会根据 PHP 的版本及设置的不同而存在差异。自动全局变量 \$\_FILES 从 PHP 4.1.0 版本开始被支持。在这之前，从 4.0.0 版本开始，PHP 支持 \$HTTP\_POST\_FILES 数组。这些数组将包含所有关于用户上传的文件的信息，这里推荐使用 \$\_FILES。如果 PHP 的设置选项 register\_globals 为 On，则相关的变量名也将会存在。从 PHP 4.2.0 版本开始，register\_globals 的默认值被设为 Off。

`$_FILES` 数组的内容如下所示（假设文件上传字段的名称为 file）。

```
$_FILES["file"]["name"]
```

表示客户端机器文件的原名称。

```
$_FILES["file"]["type"]
```

表示文件的 MIME 类型，需要浏览器提供该信息的支持，如“image/gif”。

```
$_FILES["file"]["size"]
```

表示已上传文件的大小，单位为字节。

```
$_FILES["file"]["tmp_name"]
```

表示文件被上传后在服务器端储存的临时文件名。

下面根据一段代码介绍如何使用函数 move\_uploaded\_file() 进行文件上传，如下所示。

```
function getname($sexname) {
    $dir = "upimages/"; // 定义上传文件的存储路径
    $i=1;
    if(!is_dir($dir)){
        mkdir($dir,0777); // 判断定义的路径是否已经存在，如不存在则创建它
    }
    while(true){
        if(!is_file($dir.$i.".".$sexname)){
            $name=$i.".".$sexname;
            break; // 用数字加扩展名作为上传文件的存储名，如果该文件已经存在，则使数字加1后再存储
        }
        $i++;
    }
}
```

```

    }
    return $dir.$name;
}
$extname=strtolower(substr($_FILES['upfile']['name'],(strpos($_FILES['upfile']['name'],'.')+1)));
//获取上传文件的扩展名
$uploadfile = getname($extname);
move_uploaded_file($_FILES['upfile']['tmp_name'], $uploadfile);

```

上述代码中主要应用到函数 `is_dir()`、`mkdir()`、`is_file()`、`strtolower()`、`strpos()`、`move_uploaded_file()`，下面将逐一介绍这些函数。

#### `is_dir()` 函数

```
bool is_dir(string filename)
```

`is_dir()` 函数的作用是判断该函数的参数是否是一个目录，如果该参数存在而且是一个目录就返回 `true`，否则返回 `false`。如应用 `is_dir()` 函数判断 “c:/mr” 是否为目录的代码如下：

```

<?php
if(is_dir("c:/mr")){
    //判断 c:/mr 是否为目录
    echo "是目录!";
} else{
    echo "不是目录!";
}
?>

```

#### `mkdir()` 函数

```
bool mkdir ( string pathname [, int mode])
```

`mkdir()` 函数的作用是创建由 `pathname` 指定的目录。`mode` 在 Windows 下被忽略，自 PHP 4.2.0 起为可选项。默认的 `mode` 为 0777，表示最大可能的访问权。如应用 `mkdir()` 函数在当前目录下建立目录 `mr`，代码如下：

```

<?php
if(mkdir("mr")){
    //创建目录 mr
    echo "目录建立成功!";
} else {
    echo "目录建立失败!";
}
?>

```

#### `is_file()` 函数

```
bool is_file(string filename)
```

`is_file()` 函数的作用是判断 `filename` 指定的文件是否为一个常规文件，如果该文件存在而且是常规文件就返回 `true`，否则返回 `false`。如应用 `is_file()` 函数判断 “c:/mr/mr.txt” 是否为合法文件，代码如下：

```

<?php
if(is_file("c:/mr/mr.txt")){
    //判断 c:/mr/mr.txt 是否为一个合法文件
}
?>

```

```

    echo "是合法文件!";
} else{
    echo "不是合法文件!";
}
?>

```

**strtolower()函数**

**string strtolower(string str)**

strtolower()函数的作用是将 str 中的字母全部转换成小写字母。如应用 strtolower()函数将字符串中的大写字母转换成小写字母。其代码如下：

```

<?php
echo strtolower("PHP 函数参考手册");
echo strtolower("JAVA 案例精选");
?>

```

本示例的运行结果如下：

php 函数参考手册

java 案例精选

**strrpos()函数**

**int strpos(string str1,char str2)**

strpos()函数的作用是在 str1 中查找 str2 首字符最后一次出现的位置。如应用 strpos()函数查找指定字符在字符串中最后出现的位置，代码如下：

```

<?php
echo strpos("Hello world!","WO"); //区分大小写，搜索失败
echo strpos("I am fine !","f",5); //从第 5 个字符开始搜索，输出 5
echo strpos("明日图书","图书"); //输出 4
?>

```

本示例的运行结果为： 54

**move\_uploaded\_file()函数**

**bool move\_uploaded\_file ( string filename, string destination)**

move\_uploaded\_file()函数的作用是将 filename 指定的文件上传到 destination 指定的位置。如应用 move\_uploaded\_file()函数将“c:/mr/mr.rar”上传到服务器当前目录下的 mrsoft 目录中，代码如下：

```

<?php
if(!is_dir("mrsoft")){
    mkdir("mrsoft");
}
if(move_uploaded_file("c:/mr/mr.rar","mrsoft/mr.rar")){
    echo "文件上传成功!";
} else{
    echo "文件上传失败!";
}
?>

```

```

    }
?>

```

### 3. 使用二进制形式保存图片

上传图片的另一种保存方式是以二进制的形式存储在数据库中。在计算机看来，再美的图片，再感人的文章，也不过是一堆形如“0111010011…”的代码段而已，和其他的数据没有什么区别，存到数据表中都是一样的。存储图片不再需要使用 `move_uploaded_file()` 函数，在处理页中直接使用二进制的形式读取出文件图片，并存到数据表中。本系统采用的就是以二进制的形式来保存图片，具体操作将在实现过程中给出。

使用二进制形式来保存图片的好处是安全，特别是涉及个人隐私，不易被窃取；存储方便，和表中其他的数据同等操作就可以；节省磁盘空间，由于文件系统类型的限制，放到数据表中的数据一定比直接放到磁盘中要节省空间。

当然二进制形式的图片也有很大弊端，在传出的过程中，如果某个二进制位出现丢失或损坏，那么整张图片将无法显示，而是变为一堆乱码。而且如果存储的图片过多，造成数据库过于庞大，就会影响其他数据的正常读取，增大服务器的负担。所以，对于太大的文件，就不适宜保存到数据库中而是应存放到文件夹。

### 25.8.3 图片上传的实现过程

#### ■ 视频讲解：光盘\TM\Video\25\图片上传的实现过程.exe

用户成功登录后，单击导航栏中的“图片管理” / “添加图片”超链接，即可进入添加图片页面，在“图片名称”文本框中添加上传的图片名称，在“上传路径”文本框中输入或者单击“浏览”按钮选择自己喜欢的图片，单击“上传”按钮，即可以二进制形式将图片上传到数据库中。图片上传页面的运行结果如图 25.24 所示。

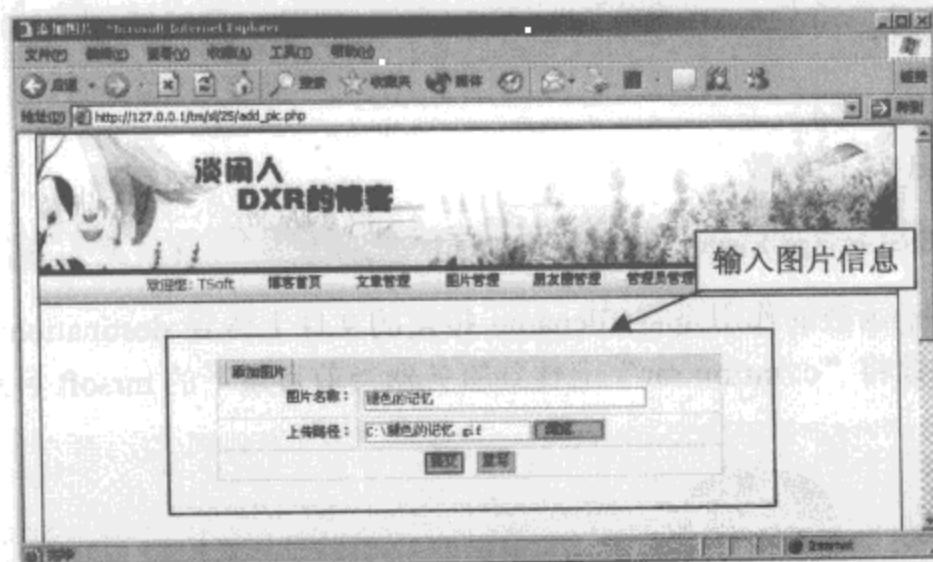


图 25.24 图片上传页面的运行结果

添加图片表单及表单说明如表 25.11 所示。

表 25.11 图片上传页面中的表单元素

元素类型	元素名称	属性设置	含义
form	myform	method="post" action="tptj_ok.php" enctype="multipart/form-data"	图片上传表单
text	tpmc	type="text" id="tpmc" size="40"	图片名称
file	file	type="file" size="23" maxlength="60"	上传文件域
submit	btn_tj	type="submit" id="btn_tj" value="提交" onClick="return pic_chk();"	提交按钮

当用户输入图片名称，并选择图片路径后，单击“提交”按钮，系统将进入到上传处理页(tptj\_ok.php)中进行处理。在处理页中，首先对图片名称进行处理，去掉特殊字符、空行和空格，然后对上传的文件进行类型检查、文件大小检查。最后以二进制的形式与图片的其他信息(如上传用户、上传时间等)一起存进数据表中。关键代码如下：

```

<?php
session_start();
include "check_login.php";
include "Conn/conn.php";
if($_POST["btn_tj"]=="提交"){
    $tpmc=htmlspecialchars($tpmc);
    $tpmc=str_replace("\n","<br>", $tpmc);
    $tpmc=str_replace(" "," ", $tpmc);
    $author=$_SESSION[username];
    $scsj=date("Y-m-d");
    $profix = array(".jpg", ".gif", ".jpeg");
    $f_name = $_FILES[file][name];
    $pro_name=substr($f_name,strrpos($f_name,"."));
    /* 判断上传文件的类型是否为允许类型 */
    if(!in_array(strtolower($pro_name), $profix)){
        echo "<script>alert('文件格式不对');history.go(-1);</script>";
        exit();
    }
    /* 判断上传文件的大小，如果文件过大，提示错误 */
    if($_FILES[file][size] > 500000){
        echo "<script>alert('文件上传错误，请重新上传');history.go(-1)</script>";
        exit();
    }else{
        $fp=fopen($file,"r");
        $file=addslashes(fread($fp,filesize($file)));
        $query="insert into tb_tpsc (tpmc,file,author,scsj) values ('$tpmc','$file','$author','$scsj')";
        $result=mysql_query($query);
        echo "<meta http-equiv=\"refresh\" content=\"1;url=browse_pic.php\">图片上传成功，请稍等...";
    }
}
?>

```

上述代码中，主要涉及 htmlspecialchars() 函数、 str\_replace() 函数的使用方法，这两个函数在处理字符串时经常会用到，下面具体介绍这两个函数的使用方法。

**htmlspecialchars()函数**

```
string htmlspecialchars(string string)
```

该函数将一些特殊字符转换成 HTML 格式，而不会将所有字符都转换成 HTML 格式。该函数转换的特殊字符如表 25.12 所示。

表 25.12 htmlspecialchars()函数转换的特殊字符

参 数	说 明
& (和)	转成&amp;
“ ” (双引号)	转成&quot;
> (大于号)	转成&gt;
< (小于号)	转成&lt;

例如，应用 htmlspecialchars() 函数将特殊字符转换成 HTML 格式，代码如下：

```
$str='<table width="371" border="0">特殊字符的转换</table>';
echo htmlspecialchars($str);
```

本示例的运行结果如下：

```
<table width="371" border="0">特殊字符的转换</table>
```

**str\_replace()函数**

```
mixed str_replace ( mixed search, mixed replace, mixed subject , int &count )
```

str\_replace() 函数将所有在参数 subject 中出现的 search 以参数 replace 替换，参数 &count 表示替换字符串执行的次数。

str\_replace() 函数的参数说明如表 25.13 所示。

表 25.13 str\_replace() 函数的参数说明

参 数	说 明
search	指定将要被替换的字符
replace	指定替换所使用的字符
subject	指定被操作的字符串
&count	替换字符串执行的次数

例如，应用 str\_replace() 函数实现对字符串的替换，代码如下：

```
<?php
$str = "吉林省明日科技";
$str1 = array("明日软件", "明日图书", "明日论坛");
$str2 = array("明日");
$strs = str_replace($str2, $str, $str1);
print_r($str1); //输出替换前的结果
print_r($strs); //输出替换后的结果
$str = str_replace("mrsoft", "吉林省明日科技有限公司", "Welcome to mrsoft!", $count);
echo "Welcome to mrsoft!";
```

```

echo $str;           //输出替换后的结果
echo $count;         //输出替换字符的次数
?>

```

#### 25.8.4 图片浏览的实现过程

##### 视频讲解：光盘\TM\Video\25\图片浏览的实现过程.exe

本页的实现代码与查看文章页面略有不同，查看文章页面中，每条数据占了一行，而查看图片则采用的是分栏显示，以每行两张图片的格式输出，每页显示 4 张图片。通过单击小图片，可以查看图片原效果图。图片浏览页面运行效果如图 25.25 所示。

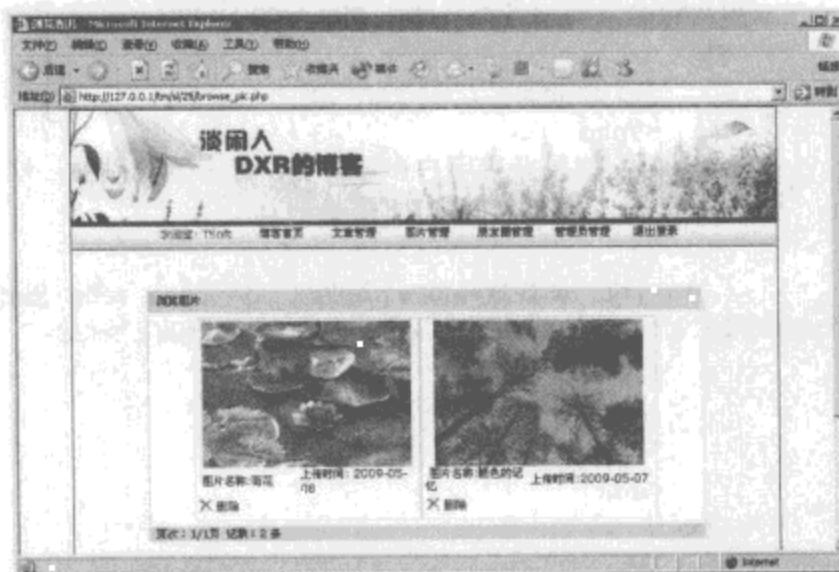


图 25.25 浏览图片页面的运行结果

实现图片浏览的关键代码如下：

```

<?php
session_start();           //开启 Session 支持
include "Conn/conn.php";   //包含数据库连接文件
include "check_login.php"; //包含权限检查文件
?>
<!-- 首部导航栏 --&gt;
&lt;tr&gt;
    &lt;td height="192" align="center" valign="top" &gt;
        &lt;?php
        //分页显示
        $query="select * from tb_tpsc where scsj order by id desc limit $offset, $page_size";
        /* 返回结果集 */
        $result=mysql_query($query);
        ?&gt;
        &lt;table width="496" border="1" align="center" cellpadding="3" cellspacing="1" bordercolor="#D6D7D6"&gt;
            &lt;tr&gt;
                &lt;?php
                /* 设置一个变量，判断当前数据是奇数输出还是偶数输出 */
                $i=1;
</pre>

```

```

while($info=mysql_fetch_array($result))
{
/* 如果当前为偶数输出，记录输出完毕后结束<tr>标签 */
if($i%2==0){
?>
<td width="500"><table width="245" border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="2"><div align="center">
<!--显示图片页 image.php -->
<a href="image.php?recid=<?php echo $info[id]; ?>" target="_blank"><img src=
"image.php?recid=<?php echo $info[id];?>" width="150" height="100"></a></div></td></tr>
<tr>
<td width="109" height="25" align="left">&nbsp;图片名称:<?php echo $info[tpmc];?></td>
<td width="128">上传时间:<?php echo $info[scsj];?></td></tr>
<tr>
<td colspan="2" height="25">
<?php
/* 如果登录用户为管理员，显示删除操作 */
if ($_SESSION[fig]==1){
?>
<a href="remove.php?pic_id=<?php echo $info[id];?>"></a>
<?php
}
?>
</td>
</tr>
</table></td>
</tr>
<?php
} else{
/* 否则，开始新的一行 */
?>
<td width="500" ><table width="236" border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="2">
<div align="center"><a href="image.php?recid=<?php echo $info[id]; ?>" target="_blank"></a>
</div>
</td>
</tr>
<tr>
<td width="110" height="25">&nbsp;图片名称:<?php echo $info[tpmc];?></td>
<td width="126">上传时间:&nbsp;<?php echo $info[scsj];?></td>
</tr>
<tr>
<td colspan="2" height="25">
<?php
if ($_SESSION[fig]==1){
?>
<a href="remove.php?pic_id=<?php echo $info[id];?>">></a>
        <?php
        }
    ?>
</td>
</tr>
</table></td>
<?php
}
/* 变量 $i 自加 1, while 循环结束 */
$i++;
}
?>
</tr>

```

通过上述代码可以发现，预览图片和显示图片实际调用的都是同一页面 image.php。image.php 页用来显示图片的实际效果，实现代码很简单，如下所示。

```

<?php
    include "Conn/conn.php";                                //包含数据库连接文件
    $query="select id,tpmc,file,author,scsj from tb_tpsc where id='".$_GET['recid']; //根据 ID 生成查询语句
    $result=mysql_query($query);                            //执行查询语句
    if(!$result) die("error: mysql query");               //判断是否有返回结果
    $num=mysql_num_rows($result);                          //计算结果集个数
    if($num<1) die("error: no this recorder");           //返回图片数据
    $data = mysql_result($result,0,"file");                //输出图片
    echo $data;
?>

```

## 25.8.5 删除图片的实现过程

### 视频讲解：光盘\TM\Video\25\删除图片的实现过程.exe

在博客中，不能保证每个用户所发表的文章或照片都是合法的，为了保证博客能够正常运营、保持良好的形象，管理员需要及时将一些非法内容进行删除。本博客中用于删除图片的关键代码如下：

```

<?php
    session_start();                                     //开启 Session 支持
    include "check_login.php";                           //包含权限检查文件
    include "Conn/conn.php";                            //包含数据库连接文件
    if($_SESSION[fig] != "1"){                         //判断是否为管理员
        echo "<script>alert('您没有删除权限');history.go(-1);</script>";
    }
    $sql="delete from tb_tpsc where id='".$_GET['pic_id']; //生成删除语句
    $result=mysql_query($sql);                          //执行删除语句
    if($result){                                       //根据$result 返回结果
        echo "<script>alert('图片删除成功!');location='browse_pic.php';</script>";
    }
    else{

```

```

echo "<script>alert('图片删除操作失败!');history.go(-1);</script>";
}
?>

```

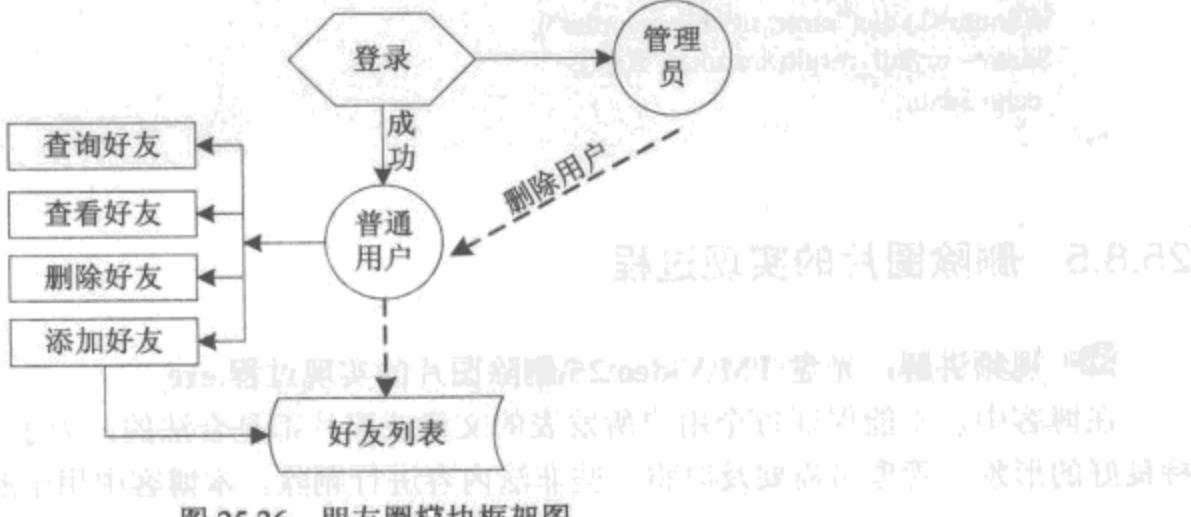
上述代码中，首先使用`$_GET[]`全局数组接收通过 GET 方法传递的图片的 ID 值，然后使用`mysql_query()`函数执行查询，根据该 ID 值删除指定的图片。

## 25.9 朋友圈模块设计

在博客中除了可以发表文章和上传图片或照片外，还可以结交一些网络朋友，本博客的朋友圈模块即可为用户提供相互交流的平台。

### 25.9.1 朋友圈模块概述

通过本博客的朋友圈模块，用户可以对好友信息进行添加、查询和删除等操作。为了保证用户隐私不会泄露，用户所添加的好友信息只有本人可以看到，其他用户即使管理员也不能随意查看。如果博客中出现恶意用户，并且被管理员删除，那么该用户的朋友圈也一并被删除。朋友圈模块框架图如图 25.26 所示。



### 25.9.2 朋友圈模块技术分析

#### 视频讲解：光盘\TM\Video\25\朋友圈模块技术分析.exe

为了有效查询朋友圈中的好友信息，该模块中使用模糊查询的方式对好友信息进行检索。SQL 语言中使用关键字 like 实现数据库中信息的模糊查询，一般采用如下两种方式对数据库中的信息进行模糊查询。

##### 1. 使用通配符“%”的 where 子句

通配符“%”表示 0 个或多个、任意长度和类型的字符，包括汉字。

示例 1：表示查找所有内容中包含“好”字的文章。

```
select * from tb_file where content like '%好%';
```

示例 2：查找所有包含“好”字或“高”字的文章，这时可以配合 or 运算符来使用，代码如下。

```
select * from tb_file where content like '%好%' or content like '%高%';
```

## 2. 使用通配符“\_”的 where 子句

通配符“\_”表示匹配任意的单个字符。

示例 1：查找用户名只包含 5 个字符，其中后 4 个字符为 soft 的用户，代码如下。

```
select * from tb_user where regname like '_soft';
```

示例 2：查找所有以 t 开头，并且以 t 结尾的、中间包含 3 个字符的用户，代码如下。

```
select * from tb_user where regname like 't___t';
```

查找的结果为：tsoft。

 注意：使用 MySQL 进行模糊查询要注意编码问题。如果编码不统一，那么查询时就容易查不到数据，或返回的数据不匹配。所以在安装 MySQL 时，要保持和系统编码的统一。常用的编码格式有 GB2312、ISO-8859-1、UTF-8 和 GBK 等。

### 25.9.3 查询好友的实现过程

#### 视频讲解：光盘\TM\Video\25\查询好友的实现过程.exe

单击“朋友圈管理” / “查询朋友信息”超链接即可进入查找好友圈模块，如图 25.27 所示。查找好友时首先在“查询条件”下拉列表框中选择按什么条件进行查询，然后在“关键字”文本框中输入要查询的关键字，最后单击“检索”按钮即可查找出相关的好友信息。

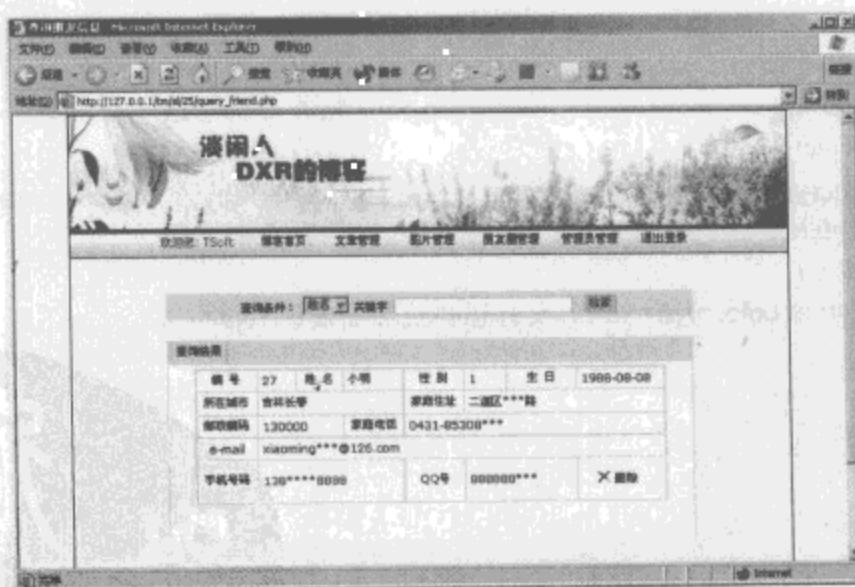


图 25.27 查询好友页面的运行结果

查询页面所涉及的表单及表单元素说明如表 25.14 所示。

表 25.14 查询好友页面的表单及表单元素

元素类型	元素名称	属性设置	含义
<input type="form"/>	myform	method="post" action="query_friend.php" onSubmit="return check();"	查询好友表单
<input type="select"/>	sel_tj	<option value="name" selected>姓名</option> <option value="id">编号</option>	查询条件选择
<input type="text"/>	sel_key	id="sel_key" size="30"	查询关键字
<input type="submit"/>	submit	type="submit" name="submit" value="检索"	提交按钮

查询处理页通过 POST 方法接收到查询条件及查询关键字后，根据提交关键字模糊查询语句，执行 SQL 语句并返回查询结果。如果没有输入关键字，则弹出提示框；如果没有查找到任何结果，则输出“Sorry! 没有您要找的朋友！”。查询处理页的关键代码如下：

```

<?php
session_start();
include "Conn/conn.php"; //开启 Session 支持
include "check_login.php"; //包含数据库连接文件
//包含权限检查文件
?>
<script language="javascript">
/* JavaScript 脚本，验证输入信息，如果为空，则弹出提示框 */
function check(form){
    if (document.myform.sel_key.value==""){
        alert("请输入查询条件!");
        myform.sel_key.focus();
        return false;
    }
}
</script>
...
<?php
if ($_POST[sel_key]!=""){
    $tj=$_POST[sel_tj]; //接收查询字段
    $key=$_POST[sel_key]; //接收查询关键字
    $sql=mysql_query("select * from tb_friend where $tj like '%$key%' and username='$_SESSION[username]'");
    $result=mysql_fetch_array($sql); //执行查询语句
    if($result==false){
        echo ("[<font color=red>Sorry! 没有您要找的朋友!</font>]");
    }else{
?>
<!-- 显示记录表单 -->
<?php
    }
}
?>

```

## 25.10 项目发布

将网站发布到 Internet 的方法有多种，最常用的方式有使用独立服务器发布站点和租用虚拟主机空间发布站点。

### 25.10.1 通过个人服务器发布项目

#### ■ 视频讲解：光盘\TM\Video\25\通过个人服务器发布项目.exe

完成网站的开发后，还需要将网站发布到 Internet 中，如果使用独立 IP 并且拥有自己的服务器，那么发布网站相对简便，只需安装 PHP 运行环境并填写独立 IP 地址，并将申请的域名解析到服务器 IP 即可。如果希望在一台服务器上运行多个站点，那么可以配置 Apache 服务器的 httpd.conf 文件中 Virtual Hosts 模块。

例如，在独立 IP 为 221.8.65.\*的服务器中发布网站 www.mingri\*\*.com 和 www.mr\*\*.com。

- (1) 正确填写服务器 IP 地址、域名地址等信息。
- (2) 将 www.mingri\*\*.com 和 www.mr\*\*.com 都解析到 221.8.65.\*这个地址。
- (3) 配置 httpd.conf 文件（假设 www.mingri\*\*.com 域名所对应的站点文件存放在服务器 c 盘的 mingri\*\* 目录下，www.mr\*\*.com 存放在 c 盘的 mr\*\* 下）。

```
<VirtualHost *:80>
    DocumentRoot c:/mingri**
    ServerName www.mingri**.com
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot c:/mr**
    ServerName www.mr**.com
</VirtualHost>
```

- (4) 将项目文件分别复制到指定的目录中。
- (5) 重新启动 Apache 服务器生效。

### 25.10.2 通过租用空间发布项目

#### ■ 视频讲解：光盘\TM\Video\25\通过租用空间发布项目.exe

要使用租用虚拟主机空间的方式发布站点，需要经过注册域名、申请空间、解析域名和上传网站 4 个步骤。首先介绍一下注册域名。

#### 1. 域名申请

域名就是用来代替 IP 地址，以方便记忆及访问网站的名称。例如，www.mingribook.com 就是明日图书网的域名，www.yahoo.com.cn 就是中文雅虎的域名。域名需要到指定的网站中注册购买，如

www.net.com（万网）和 www.xinnet.com（新网）等。

购买注册域名的步骤如下：

- (1) 登录域名服务商网站。
- (2) 注册会员。如果不是会员则无法购买域名。
- (3) 进入域名查询页面，查询要注册的域名是否已经被注册。
- (4) 如果用户欲注册的域名未被注册，则进入域名注册页面，在该页面中填写相关的个人资料。
- (5) 填写成功后，单击“购买”按钮，注册成功。
- (6) 付款后，等待域名开启。

## 2. 申请空间

域名注册完毕后，下一步是申请空间。空间可以使用虚拟主机或租借服务器。目前，许多企业建立网站都采用虚拟主机，这样既节省了购买机器和租用专线的费用，同时也不必聘用专门的管理人员来维护服务器。申请空间的步骤如下：

- (1) 登录虚拟空间服务商网站。
- (2) 注册会员（如果已有会员账号，则直接登录即可）。
- (3) 选择虚拟空间类型（空间支持的语言、数据库、空间大小、流量限制等）。
- (4) 确定机型后，直接购买。
- (5) 进入到缴费页面，选择缴费方式。
- (6) 付费后，空间在 24 小时内开通，随后即可使用此空间。

## 3. 将域名解析到服务器

域名和空间购买成功后，接下来需要将域名地址指向虚拟服务器的 IP。进入域名管理页面，添加主机记录。一般先要输入主机名，注意不包括域名。例如，解析 www.bccd\*\*\*.com，如果要添加该主机，只需输入 www 即可，后面的 bccd\*\*\*.com 不需要填写。接下来填写 IP 地址。最后单击“确定”按钮，剩下的工作交给域名服务器即可。如果想添加多个主机名，则重复上面的操作即可。

## 4. 上传网站

最后就是上传网站程序。上传网站需要使用 FTP 软件。如果使用 Dreamweaver，则可以直接在 Dreamweaver 中上传。这里以 CuteFTP 为例讲解上传步骤。

- (1) 打开 FTP 软件。
- (2) 选择 File/Site-Manager 命令，弹出站点面板。
- (3) 单击 New 按钮，新建一个站点。
- (4) 在 Label for site 文本框中输入站点名。
- (5) 在 FTP Host Address 文本框中输入域名。
- (6) 在 FTP site User Name 文本框中输入用户名。
- (7) 在 FTP site Password 文本框中输入密码。
- (8) 单击 Edit 按钮，弹出编辑窗口。
- (9) 选中 Use PASV mode 和 Use firewall setting 复选框。
- (10) 单击“确定”按钮。
- (11) 单击 Connet 按钮连接到服务器。

(12) 连接服务器后，在左侧的本地页面中右击需要上传的文件，单击“上传”按钮即可。

(13) 如果上传过程中出现错误，选择“继续上传”命令即可。

(14) 上传成功后，关闭 FTP 软件。

## 25.11 开发常见问题与解决

### ■ 视频讲解：光盘\TM\Video\25\开发常见问题与解决.exe

在项目开发过程中，免不了会遇到一些问题，其实问题并不可怕，关键是需要开发人员能够认真细心地去面对、去解决。下面将介绍如何处理和解决博客开发过程中常见的问题。

#### 1. 数据库乱码问题

首先查看本博客的数据库连接文件，代码如下：

```
<?php
$link=mysql_connect("localhost","root","root");           //建立与 MySQL 数据库的连接
mysql_select_db("db_blog",$link);                         //选择数据库
mysql_query("set names gb2312");                          //设置字符集
?>
```

最初在开发本博客时，并没有加上“mysql\_query("set names gb2312")”这句代码，可是在调试程序时，发现从数据库提取的所有中文字符都为乱码，如图 25.28 所示。

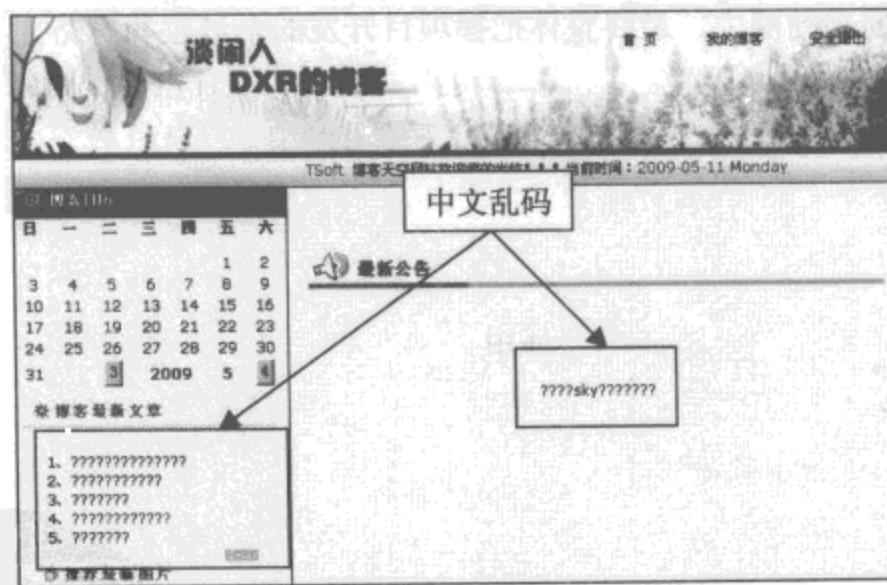


图 25.28 中文乱码问题

这是没有正确设置数据库中的字符集所导致的乱码错误。“mysql\_query("set names gb2312")”这句代码的作用就是设置 MySQL 数据库的字符集为 GB2312，当然也可以使用这句代码设置 MySQL 数据库的字符集为其他类型，如 UTF-8 等。

#### 2. 接收 GET 方法或 POST 方法提交数据问题

在接收 GET 方法或 POST 方法提交的数据时，一般使用\$\_GET 或\$\_POST 全局数组进行接收。

在实际开发过程中，有时不用这两个全局数组进行接收，而直接用与传递的参数同名或与提交表

单元素名同名的变量名也可以获取提交的内容。这是因为如果 php.ini 文件中的 register\_globals 的值为 On，那么使用 GET 方法或 POST 方法提交的参数将被保存到与参数名同名的全局变量中。

但是在开发过程中，不建议使用 GET 方法或 POST 方法提交的参数，而是使用\$\_GET 或\$\_POST 全局数组进行接收。这样既可以防止全局变量与页面中所定义的局部变量出现命名冲突的情况，也可以提高程序的可移植性。

php.ini 文件中 register\_globals 选项的修改方法如图 25.29 所示。

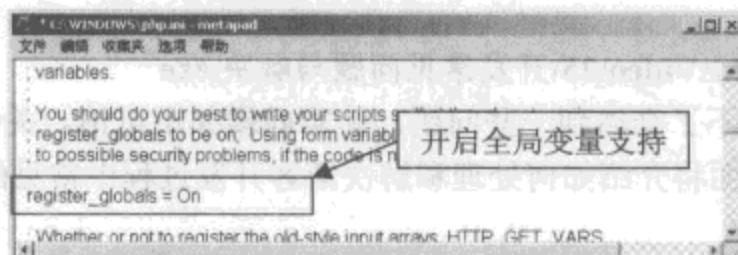
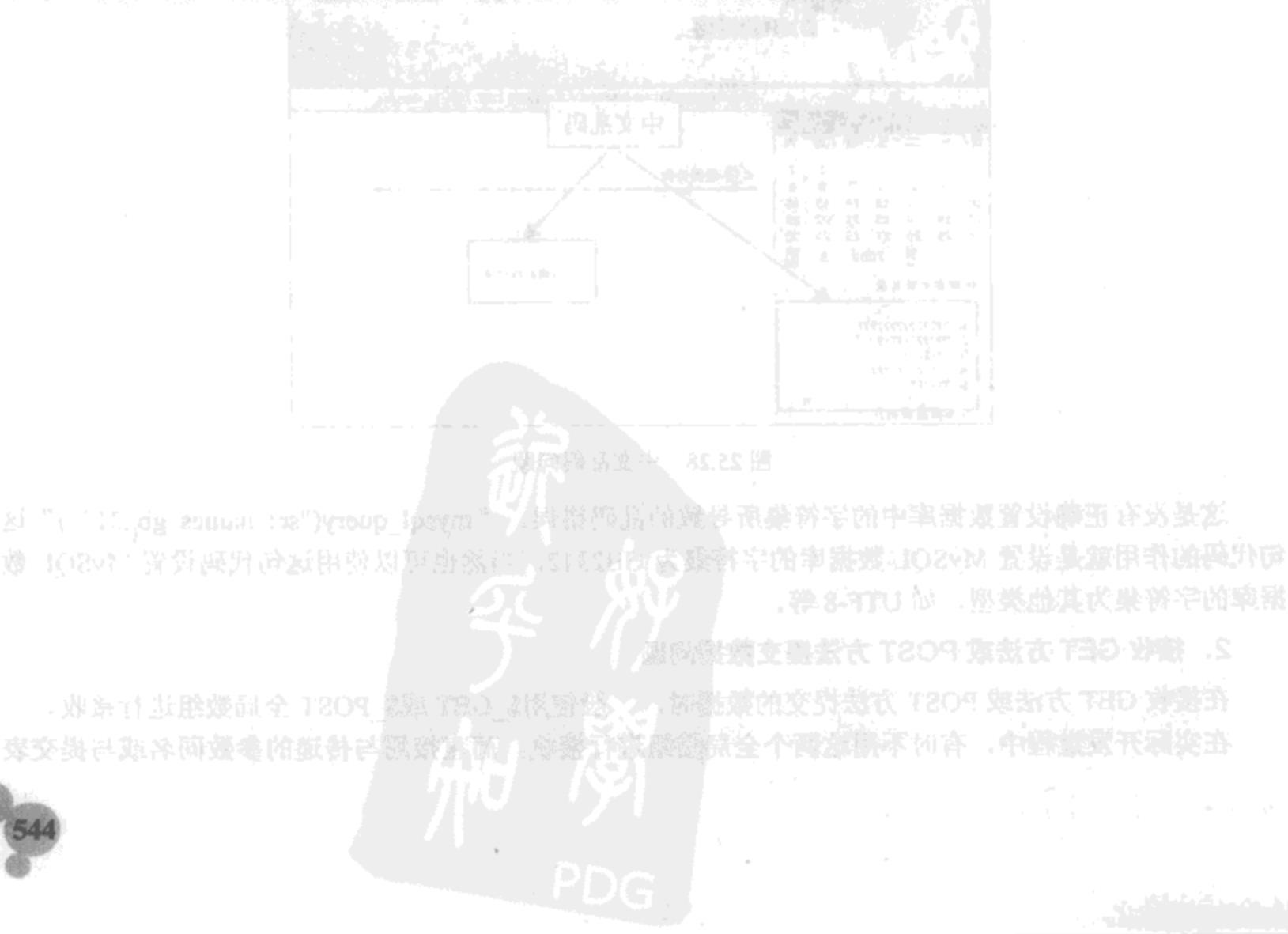


图 25.29 修改 php.ini 文件中 register\_globals 参数

## 25.12 本章小结

本章从博客管理系统的项目描述开始，详细介绍了博客系统的系统分析过程、数据库设计过程，并对博客系统相对重要的模块进行了详细介绍。通过本章的学习，读者可以对前面的基础知识进行全面、系统的复习，进一步了解和深化前面所学基础知识的含义和实际应用价值，并能够培养自身独立思考、分析问题、解决问题的能力，最终整体把握项目开发的全过程和思路。



# 第26章

## 应用 Smarty 模板开发电子商务平台

( 视频讲解: 281分钟)

电子商务 (Electronic Commerce), 简称 EC, 是指整个事务活动和贸易活动的电子化, 它通过先进的信息网络, 将事务活动和贸易活动中发生关系的各方有机地联系起来。电子商务网站实际上就是企业为消费者提供的网上虚拟购物商城, 在该网站中, 用户可以购买任何商品, 而管理员可以对商品和订单等信息进行管理。通过本章的学习, 读者不仅可以丰富自己的基础知识、掌握大量网站制作技巧及 PHP 常用函数的使用方法, 更能掌握网络编程的设计思路、方法和过程, 快速提高 PHP 开发能力和设计水平。

通过阅读本章, 可以学习到:

- ■ **如何进行系统分析**
- ■ **如何编写项目计划书**
- ■ **数据库设计流程**
- ■ **搭建系统架构**
- ■ **简单的树状菜单**
- ■ **如何灵活运用 JS 和 Ajax 技术**
- ■ **如何灵活运用购物车**
- ■ **如何灵活运用 Smarty 模板开发程序**
- ■ **ADODB 在实际中的应用**

## 26.1 项目描述

开发电子商务平台实质上就是建立一个虚拟的购物超市，当在超市选购商品时，首先应该将商品放到购物车中，待挑选好所有商品之后就可以到收银台去结算，根据结算单据付款。开发电子商务平台的原理与在超市购物的原理是相同的：客户通过网页浏览商品，选购自己喜欢的商品，将商品放入购物车中，当然也可以改变购买商品的数量或者清空购物车中的商品，在选购好商品后到收银台去结算，生成订单，填写并提交收货人信息，最终按照订单中提交的付款方式进行付款，等待送货上门，期间还可以登录平台查看自己的订单。

企业就是这个虚拟超市的管理者，可以对登录的用户、商品信息、商品类别、订单和公告等内容进行管理。

本系统建立一个电子商务平台，实现网上购物的功能，包括从客户最初的商品浏览到最后的购物结算、订单查看，以及企业的商品信息、商品类别、客户信息和订单的管理。

系统开发应用 PHP 语言+MySQL 数据库，通过 Smarty 模板实现网页动静分离，通过 ADODB 操作数据库，其中还应用到 Ajax、Fram 框架、JavaScript、CSS 和 DIV 等技术。

系统中典型的功能模块是：找回密码、Ajax 无刷新注册验证、GD2 图像验证码、购物车、收银台、树状菜单和商品信息管理。

## 26.2 系统分析

### 26.2.1 需求分析

中国的市场经济日趋成熟，中国企业面对的竞争压力也越来越大，企业正在寻求一种高效、便于客户购物和支付的购物形式，而电子商务这种新的商业运营模式完全符合商家们的要求，同时也可以满足消费者的需求。消费者能够在网上实现互动的交流及足不出户地购买产品，向企业发表自己的意见、服务需求及有关投诉，并且可以通过网站的交互式操作向企业进行产品的咨询，以此来获取相应的回馈信息及技术支持。

电子商务被越来越多的商家和消费者认同，必将成为 21 世纪网络发展的主流，而网上购物将成为一种购物时尚。目前国内企业正纷纷加入到阿里巴巴等一系列大型电子商务网站中，通过网络进行 B2B（企业对企业）的网上交易之后，也充分认识到网络交易将成为未来商品交易的重要组成部分，在通过网络宣传企业的同时，也逐步扩大企业自身的网络销售渠道，建立起自己的电子商务平台，完成了从 B2B 到 B2C（企业对个人）进行网络交易的过渡，大大提高了企业的经济效益。因此，企业开发属于自己的电子商务平台已经势在必行。

### 26.2.2 编写项目计划书

项目计划书是项目负责人写给投资者的一份详细的开发报告，包括开发背景、开发原因、可行性

分析、成本预算、开发周期等。给投资者的计划书无论是详细还是简短，要以引起投资者对项目的兴趣为目的，在真实的基础上，清楚、明确地表达该项目对投资人的必要性和迫切性。本节就来看一份典型的电子商务项目开发计划书。

## 电子商务项目开发计划书

**版本：v1.0**

**拟制人：邹甜甜**

**审核人：高盈月**

**批准人：赛凯臣**

**批准日期：二零零七年八月二日**

项目名称：电子商务项目

项目编号：0727

项目负责人：邹甜甜

项目审核人：高盈月

项目批准人：赛凯臣

项目完成日期：2007 年 10 月 15 日

项目地点：中国北京

项目预算：人民币 100 万元

项目规模：中型项目

项目状态：启动

项目描述：电子商务项目

项目目标：实现电子商务功能

项目需求：客户管理、商品管理、订单管理、支付管理、物流管理、统计分析

项目风险：市场竞争激烈、技术更新快、资金不足

项目优势：市场需求大、政策支持、人才充足

项目周期：3 个月

项目团队：项目经理 1 人，开发人员 5 人，测试人员 2 人，市场人员 1 人，客服人员 1 人

项目预算：人民币 100 万元

项目规模：中型项目

项目状态：启动

项目描述：电子商务项目

项目目标：实现电子商务功能

项目需求：客户管理、商品管理、订单管理、支付管理、物流管理、统计分析

项目风险：市场竞争激烈、技术更新快、资金不足

项目优势：市场需求大、政策支持、人才充足

项目周期：3 个月

项目团队：项目经理 1 人，开发人员 5 人，测试人员 2 人，市场人员 1 人，客服人员 1 人

项目预算：人民币 100 万元

项目规模：中型项目

项目状态：启动

项目描述：电子商务项目

项目目标：实现电子商务功能

项目需求：客户管理、商品管理、订单管理、支付管理、物流管理、统计分析

项目风险：市场竞争激烈、技术更新快、资金不足

项目优势：市场需求大、政策支持、人才充足

## 电子商务平台项目计划书

**一、项目开发背景**

自 20 世纪 90 年代，互联网的蓬勃发展，为企业提供了一个全新的机遇。企业网站、电子商务成为热门话题。其中，电子商务更是关系到经济结构、产业升级和国家整体经济竞争力。为此，我国已经将发展电子商务列为信息化建设的重要内容，并努力创造条件，积极地推进电子商务的发展。

据美国在线（AOL）和 Henley Centre 联合进行的一项调查显示：国外有 80% 的受调查者会选择网上购物或寻求帮助，10% 的受调查者会选择熟悉的品牌或厂商来购买。而在国内，自 1997 年拉开了电子商务的序幕，短短的 10 年时间里，全国已有 4 万家商业网站，几乎每天都有新的网站诞生，厂商所在地也从上海、广州、深圳等沿海发达地区扩展到全国各大中城市。

**二、电子商务的概念及特点**

电子商务（Electronic Commerce），简称 EC，是目前发展较快的一种商务模式。不同领域的人群对 EC 的理解各有不同，综合来说，电子商务是通过互联网进行的商务活动，它已经渗透了各个领域，如服务、金融、销售等。

目前来说，电子商务主要有两种模式。一种是纯粹的电子商务网站，如阿里巴巴，它为交易双方提供了一个丰富的信息平台；另一种是传统产业的电子商务模式，它借助互联网进行特定商品的销售及采购。

**三、电子商务的优势**

电子商务的优势是显而易见的，主要有以下几个方面：

- 大大节约了企业的成本预算，如人工费、场地费等。
- 降低了产品的零售价格。因为是和消费者直接交流沟通，取消了代理、批发、商场等费用，在同行业的竞争中占据优势。
- 扩展了业务范围，只要能够上网的人，就有机会成为客户，一定程度上消除了地域的影响。
- 加速了资本流通。
- 实现 24 小时连续服务。

**四、可行性分析**

- 市场前景。数据表明：2007 年中国 B2B 电子商务交易规模为 12500 亿元，比 2006 年增长 25.5%。预计未来两年我国 B2B 电子商务交易规模将继续高速增长，2008 年将达到约 16200 亿元，2009 年交易规模有望达 21300 亿元。B2C 网站总收入为 52.2 亿元，2009 年有望达到 98.6 亿元。而且，随着网络购物环境的好转，未来两年 B2C 电子商务交易模式将更受欢迎，用户数和年平均消费金额均会提高。这里所蕴含的商机毋庸置疑。
- 技术可行性。电子商务相关的技术（如认证加密技术、数据备份技术、网上交易技术）已经十分完善，能够保障系统正常、无错地运行。
- 经济可行性。通过网络进行商务活动，可以足不出户地完成交易。既可节省交通费，又减少了务工人员费用，同时，前期可以使用小型服务器或虚拟空间以减少成本预算。

**五、项目实施的技术方案**

- 为了节约成本，使用 PHP 做开发，数据库使用 MySQL。
- 采用 Smarty 模板，可以最大限度地加快网页访问速度。
- 考虑到以后可能升级数据库，使用 ADODB 类库连接。
- 使用 Ajax 技术来提高交互性。
- 提供多种付款方式，包括网银、邮购、汇款等。
- 提供在线咨询。

## 电子商务平台项目计划书

- 操作简单容易，只需要简单指点便可使用，无须专业人员。

**六、总体投入**

报价

顶级域名	.com 或.net 结尾（英文域名）	¥100/年
虚拟主机	智尊 U300 (250MB/赠送 125MB 邮局 /50MB mysql/php/perl)	¥750/年
电子商务平台	电子商务网上交易系统	¥20000/套（包括一年的免费维护，如需二次开发费用另算）
留言板	用户提交留言，不需审核立即显示，管理员可随时回复删除相关主题，是一套人性化的沟通工具，有助于搞好客主关系	¥500/个
友情链接	管理员可随时添加相关链接	¥500/个
公告版		¥500/个
LOGO		¥200/个
搜索引擎优化	使网站能够被搜索引擎抓取，并排名靠前，利于网站的推广	赠送

共计：域名+虚拟主机+电子商务系统+留言板+友情链接+公告版=¥22550，以后每年只需交 ¥850（域名+主机）即可。

**七、人员安排**

人员安排

项目负责人	1 名（邹甜甜）	负责项目的总体设计以及与企业沟通
系统策划	1 名（刘明松）	负责网站的策划和制作
网站开发人员	4 名（潘明远、孙海、刘忻南、梁焱）	负责系统后台开发
美工	1 名（董朋）	负责页面和 LOGO 设计
测试人员	1 名（刘杏）	负责系统的后期调试

**八、后期维护**

在系统正式使用的一年时间内，将对系统进行免费维护（添加系统功能或修改框架除外），及时修改程序缺陷，还为客户提供技术培训。同时，保证运行速度和系统安全。

**九、系统功能模块**

本系统分前台和后台两部分。

- 前台包括

- 登录及注册模块：包括会员的注册、登录和找回密码功能。
- 商品模块：包括商品的分类浏览和查看。
- 商品搜索模块：包括精确查询和模糊查询两种模式。实现对商品任意信息的搜索。
- 购物车模块：帮助会员完成购物功能。
- 在线支付模块：支持工行、招行等各个银行的信用卡、支付卡。
- 公告栏模块：最新的咨询、活动、声明等。
- 友情链接模块：行业内网站之间的互换链接栏目的要求，实现网站的免费推广与宣传。
- 留言模块：包括留言板、在线 QQ 等客户和企业的交流平台。

- 后台包括

- 商品管理：包括对商品的增删改查等操作和商品的类别管理。
- 会员管理模块：包括对会员的信息和留言管理。
- 订单管理模块：管理员可以对客户提交的订单进行编辑、查看和处理。

## 电子商务平台项目计划书

■ 公告管理模块：对公告版的信息进行添加、删除、修改等操作。

### 十、项目安排及开发时间表

项目共分 3 部分进行。

第一部分：页面验收。从签订合同之日起，美工开始设计效果图，然后交给客户验收。

如果客户不满意，则美工重新设计或修改。只有客户满意后才继续进行。

第二部分：客户验收合格后，网站开发人员开始实现程序的功能模块。如果在开发过程中发现漏洞或是其他一些技术原因，将尽快和用户协调商定，以免造成严重影响。

第三部分：模块测试。系统开发完成后，由测试人员进行调试，将错误降到最低。

项目开始日期：2007 年 8 月 3 日

项目完成日期：2007 年 8 月 24 日

开发时间表

任务安排	开始时间	完成时间	工作日	负责人
验收页面设计	2007-08-03	2007-08-09	7	董明
系统策划分析	2007-08-03	2007-08-05	3	刘明松
商品模块设计	2007-08-10	2007-08-12	3	
商品搜索	2007-08-13	2007-08-15	3	潘明远
商品管理模块	2007-08-16	2007-08-18	3	
登录及注册模块设计	2007-08-10	2007-08-12	3	
留言模块	2007-08-13	2007-08-15	3	孙海
会员管理模块	2007-08-17	2007-08-18	3	
在线支付模块	2007-08-10	2007-08-12	3	
购物车模块	2007-08-13	2007-08-15	3	刘忻南
订单管理模块	2007-08-16	2007-08-18	3	
数据库设计	2007-08-06	2007-08-08	3	
公告栏模块	2007-08-10	2007-08-11	2	
友情链接模块	2007-08-12	2007-08-13	2	梁焱
公告管理模块	2007-08-14	2007-08-16	3	
系统测试	2007-08-19	2007-08-24	5	刘杏

### 十一、责任人签字

所有参与开发的人员，在开发期间，要严格遵守保密协议，凡是涉及客户企业的商业机密、内部资料等，均不得外泄，否则将追究法律责任。

签字：刘忻南 梁焱 刘明松 董明 潘明远

孙海 刘杏

## 26.3 系统设计

### 26.3.1 系统目标

根据电子商务平台的操作流程，结合客户提供的需求，以及对实际情况的考察与分析，将电子商务平台的特点总结如下：

- 页面设计简洁、明快，突出重点，以吸引用户目光为目的。
- 前台操作流程清晰，确立从商品浏览、购物车、提交订单、查看订单的基本流程。
- 后台管理目标明确，包括对用户、商品信息、订单和公告等信息的管理。
- 可操作性高，避免复杂的、有异议的链接。
- 浏览速度快，尽量避免长时间打不开页面的情况发生。
- 商品浏览以实物图片为参照，图像清晰，文字醒目。
- 商品查询功能，可以通过商品的各个属性来搜索。
- 商品分类功能，为用户购物提供导向。
- 网站架构合理，数据库稳定。
- 易维护，并提供二次开发支持。

### 26.3.2 系统功能结构

电子商务平台分前台系统和后台系统。下面分别给出前、后台的系统功能结构图。电子商务前台系统功能结构图如图 26.1 所示。

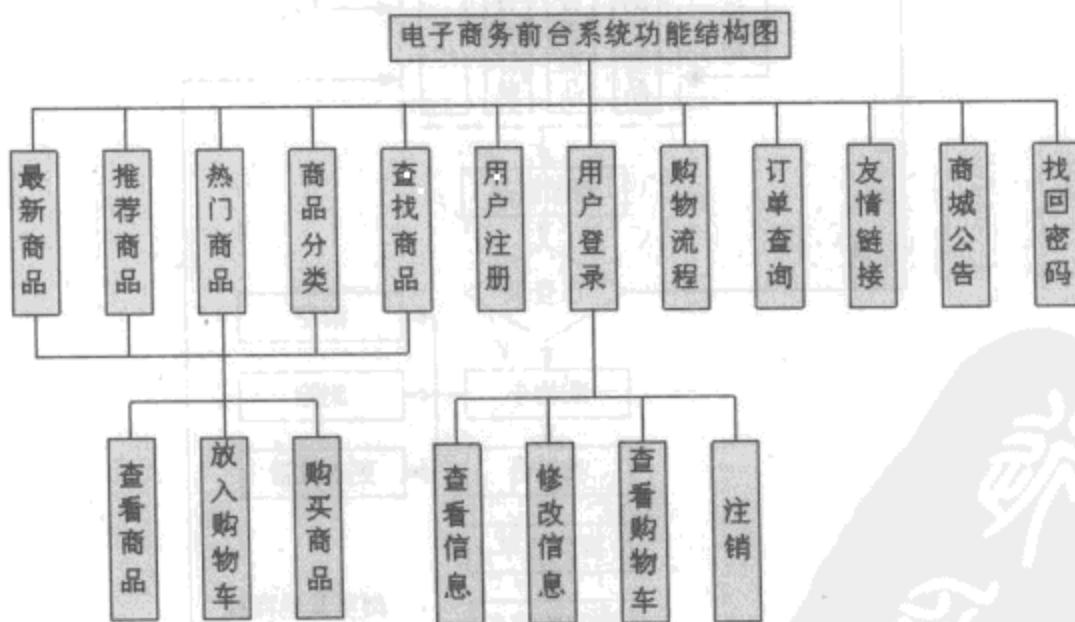


图 26.1 电子商务前台系统功能结构图

电子商务后台系统功能结构图如图 26.2 所示。

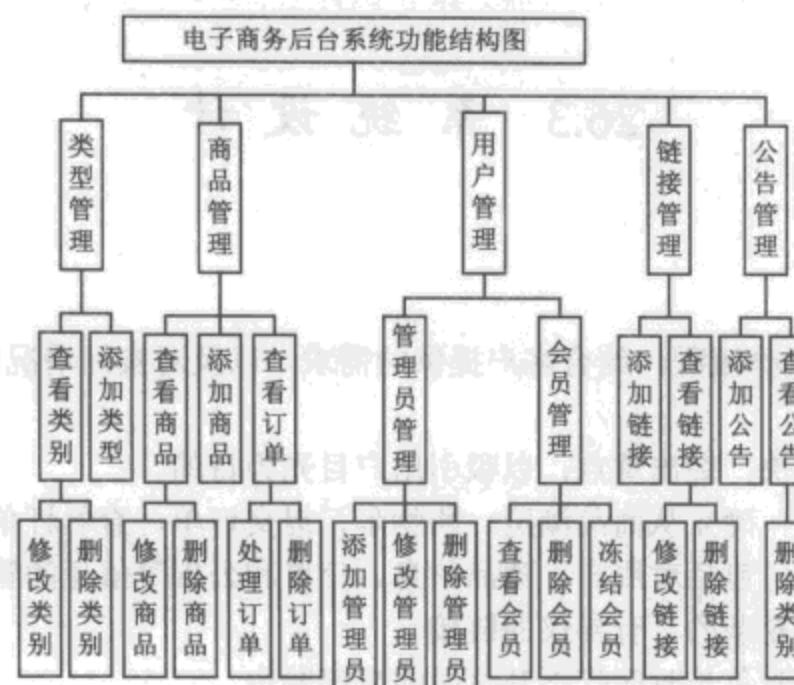


图 26.2 电子商务后台系统功能结构图

### 26.3.3 系统流程图

为了便于开发人员了解系统各功能模块之间的联系及完整的购物流程，下面给出系统的购物流程图，如图 26.3 所示。

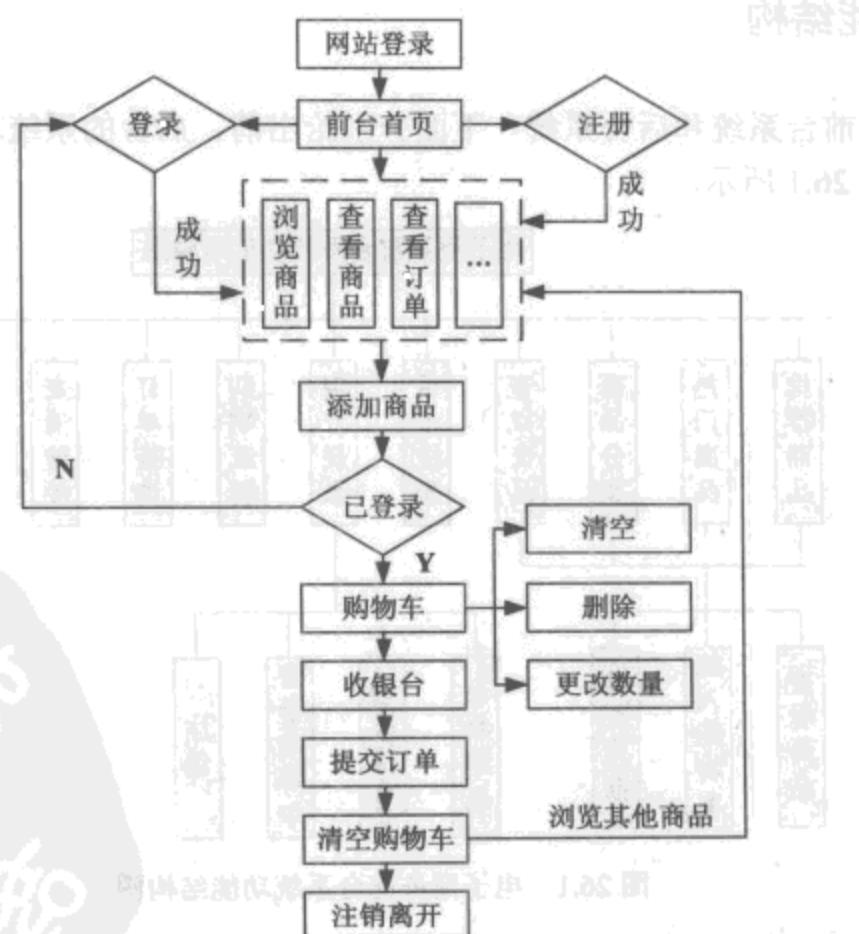


图 26.3 系统流程图

## 26.4 系统开发环境

在开发电子商务平台时，该项目应用到的软件如下。

服务器端

- 操作系统：Windows 2003 Server/Linux（推荐）。
- 服务器：Apache 2.2.3。
- PHP 软件：PHP 5.1.6。
- 数据库：MySQL 5.0.24。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- ADODB：ADODB 5.0。
- Smarty 模板：Smarty-2.6.19。
- 开发工具：Dreamweaver 8。
- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

客户端

- 浏览器：推荐 IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

## 26.5 数据库与数据表的设计

 视频讲解：光盘\TM\Video\26\数据库与数据表的设计.exe

无论是什么系统软件，其最根本的功能就是对数据的操作与使用。所以，一定要先做好数据的分析、设计与实现，才能更好地完成对应的功能模块。

### 26.5.1 数据库分析

根据需求分析和系统的功能结构图，找出需要保存的信息数据（也可以理解为现实世界中的实体），并将其转化为原始数据（属性类型）形式。这种描述现实世界的概念模型，可以使用 E-R 图来表示，也就是实体-联系图。最后将 E-R 图转换为关系数据库。这里重点介绍几个 E-R 图。

#### 1. 会员信息实体

会员信息实体包括编号、用户名、密码、E-mail、身份证号、联系电话、QQ 号、密码提示、密码答案、邮编、注册时间、真实姓名等属性。会员信息实体 E-R 图如图 26.4 所示。

#### 2. 商品信息实体

商品信息实体包括编号、名称、上市时间、添加时间、型号、图片、库存、销售、商品类型、会员价、市场价、是否打折等属性。商品信息实体 E-R 图如图 26.5 所示。

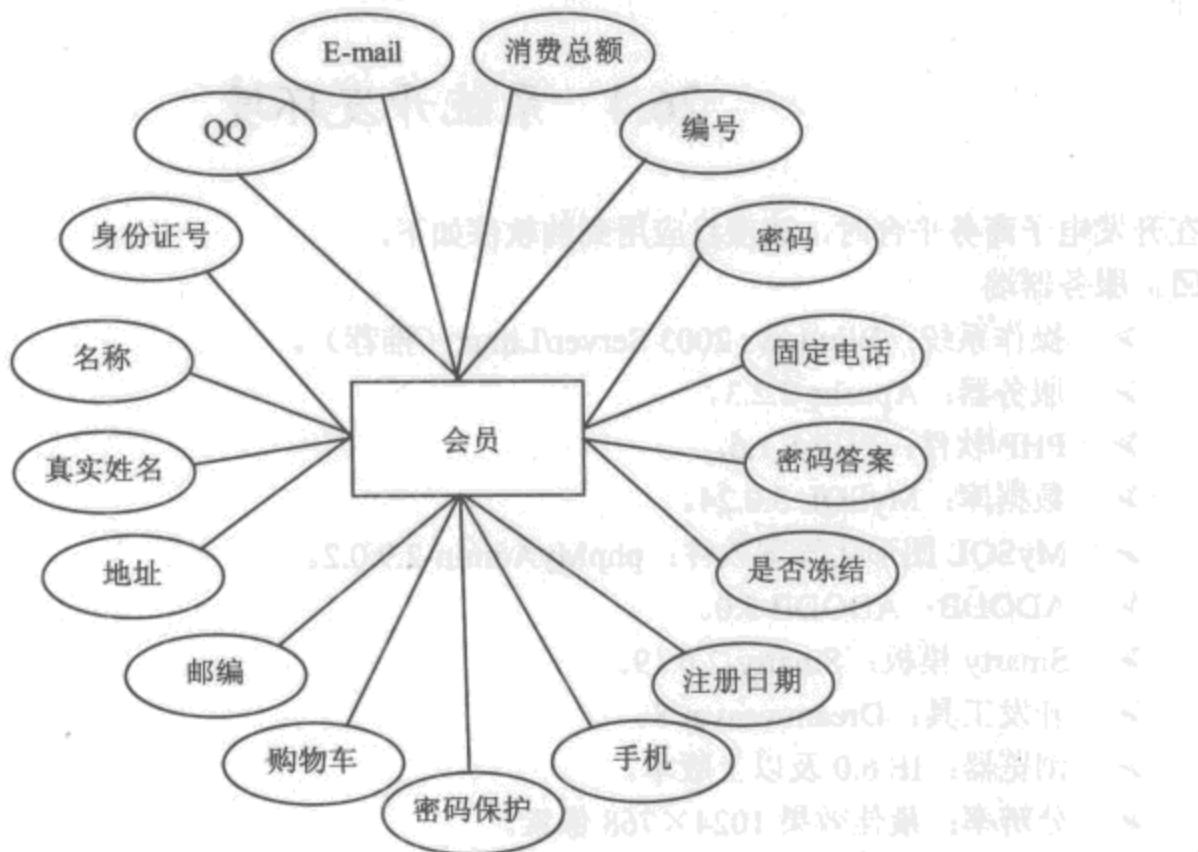


图 26.4 会员信息 E-R 图



图 26.5 商品信息 E-R 图

### 3. 商品订单实体

商品订单实体包括编号、订单号、商品 ID、数量、单价、打折率、收货人姓名、收货地址、邮编、联系电话、E-mail、收货方式、支付方式、订单时间、发货人姓名、状态、价格总计等属性。商品订单

实体的 E-R 图如图 26.6 所示。

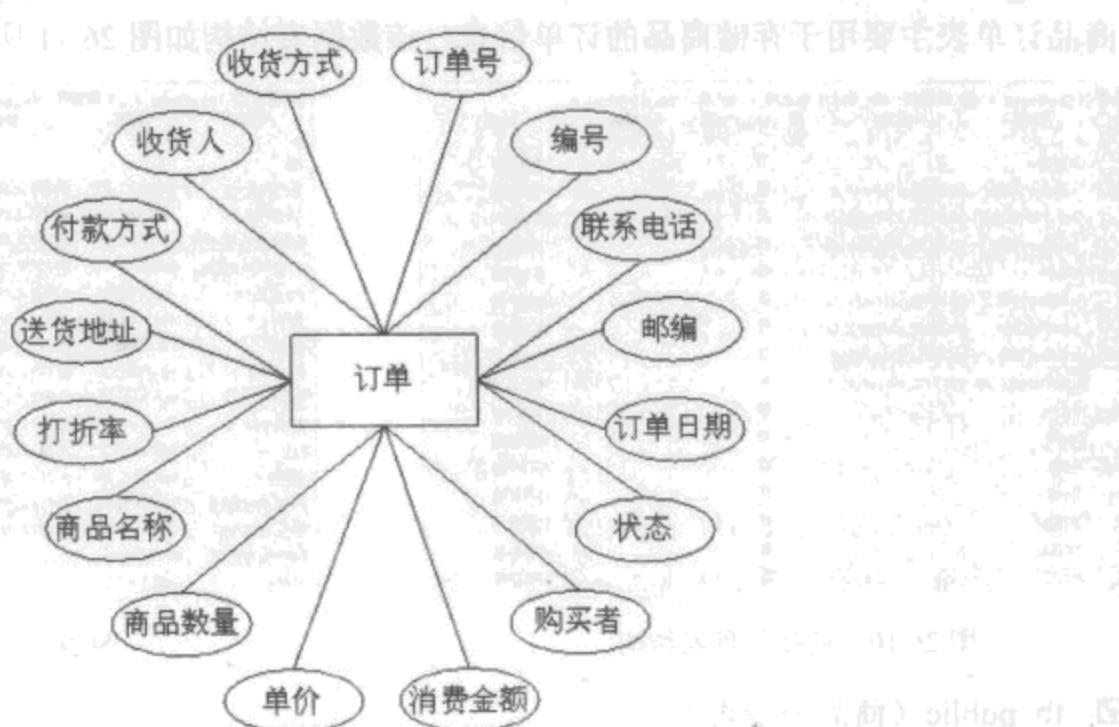


图 26.6 商品订单 E-R 图

除了上面介绍的 3 个 E-R 图, 还有公告实体、管理员实体、类型实体、友情链接实体, 限于篇幅, 这里不再赘述。

## 26.5.2 创建数据库和数据表

系统 E-R 图设计完成后，接下来根据 E-R 图来创建数据库和数据表。关于数据库和数据表的创建可参考第 17 章 MySQL 数据库基本操作，这里不再赘述。首先看一下电子商务平台都使用了哪些数据表，如图 26.7 所示。

下面来看各个数据表的结构和字段说明。

### tb\_admin (管理员信息表)

管理员信息表主要用于存储管理员的信息。该数据表结构如图 26.8 所示。

tb\_class (商品类型表)

商品类型列表主要用于添加商品的类别，可以设定多个子类别，目前最多只能到二级子类别。该数据表结构如图 26.9 所示。

服务器: localhost	数据库: db_database26		
表	类型	整理	说明
tb_admin	MyISAM	gb2312_chinese_ci	管理员信息表
tb_class	MyISAM	gb2312_chinese_ci	商品分类表
tb_commo	MyISAM	gb2312_chinese_ci	商品信息表
tb_form	MyISAM	gb2312_chinese_ci	商品订单表
tb_links	MyISAM	gb2312_chinese_ci	友情链接表
tb_opinion	MyISAM	gb2312_chinese_ci	商品评论表
tb_public	MyISAM	gb2312_chinese_ci	公告信息表
tb_user	MyISAM	gb2312_chinese_ci	会员信息表

图 26.7 电子商务数据表

数据库: db_database26 > 表: tb_admin						
字段	类型	长度	属性	Null	默认	说明
id	int(4)		否		auto_increment	自动编号
name	varchar(50)	gb2312_chinese_ci	否			管理员账号
pwd	varchar(20)	gb2312_chinese_ci	否			管理员密码

图 26.8 管理员信息表结构

图 服务器: localhost > 命名数据库: db_database26 > 表: tb_class							
字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自增编号
name	varchar(20)	gb2312_chinese_ci		否			类型名称
level	int(1)			否	1		等级

图 26.9 商品类型表结构

tb\_commo (商品信息表)

商品信息表主要用于存储关于商品的相关信息。该数据表结构如图 26.10 所示。

**tb\_form** (商品订单表)

商品订单表主要用于存储商品的订单信息。该数据表结构如图 26.11 所示。

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否	auto_increment		自动编号
name	varchar(50)	gb2312_chinese_ci		否			商品名称
pics	varchar(200)	gb2312_chinese_ci		否			商品图片
info	mediumtext	gb2312_chinese_ci		否			商品介绍
addtime	date			否			添加时间
area	varchar(50)	gb2312_chinese_ci		否			商品产地
model	varchar(50)	gb2312_chinese_ci		否			商品型号
class	varchar(50)	gb2312_chinese_ci		否			商品类别
brand	varchar(50)	gb2312_chinese_ci		否			商品品牌
stocks	int(4)			否	1		商品库存
sell	int(4)			否	0		商品销量
m_price	float			否			市场价格
v_price	float			否			会员价格
fold	int(2)			否	9		打折率
isnew	int(1)			否	1		是否新品
isnom	int(1)			否	0		是否推荐

图 26.10 商品信息表结构

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否	auto_increment		自动编号
formid	varchar(25)	gb2312_chinese_ci		否			订单号
commo_id	varchar(100)	gb2312_chinese_ci		否			商品id
commo_name	varchar(50)	gb2312_chinese_ci		否			商品名称
commo_num	varchar(100)	gb2312_chinese_ci		否			商品数量
agoprice	varchar(50)	gb2312_chinese_ci		否			商品价格
fold	varchar(50)	gb2312_chinese_ci		否			商品折率
total	varchar(50)	gb2312_chinese_ci		否			总金额
vendee	varchar(50)	gb2312_chinese_ci		否			订单用户
talker	varchar(50)	gb2312_chinese_ci		否			收货人
address	varchar(200)	gb2312_chinese_ci		否			收货地址
tel	varchar(20)	gb2312_chinese_ci		否			移动电话
code	varchar(10)	gb2312_chinese_ci		否			邮编
pay_method	varchar(20)	gb2312_chinese_ci		否			付款方式
del_method	varchar(20)	gb2312_chinese_ci		否			送货方式
formtime	timestamp			否			订单时间
state	int(1)			否			订单状态

图 26.11 商品订单表结构

**tb\_public** (商品公告表)

商品公告表主要用于展示网站的最新活动和最新消息，包括公告时间、公告标题和公告内容。该数据表结构如图 26.12 所示。

**tb\_user** (用户信息表)

用户信息表主要用于存储用户的基础信息。该数据表结构如图 26.13 所示。

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否	auto_increment		自动编号
title	varchar(50)	gb2312_chinese_ci		否			公告标题
content	mediumtext	gb2312_chinese_ci		否			公告内容
is_time	date			否			发布时间

图 26.12 商品公告表结构

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否	auto_increment		自动编号
name	varchar(50)	gb2312_chinese_ci		否			会员名称
password	varchar(50)	gb2312_chinese_ci		否			密码
question	varchar(50)	gb2312_chinese_ci		否			密码保护
answer	varchar(50)	gb2312_chinese_ci		否			问题答案
consume	float			否	0		消费总额
realname	varchar(50)	gb2312_chinese_ci		否			真实姓名
card	varchar(20)	gb2312_chinese_ci		否			身份证号
tel	varchar(20)	gb2312_chinese_ci		否			移动电话
phone	varchar(20)	gb2312_chinese_ci		否			固定电话
Email	varchar(25)	gb2312_chinese_ci		否			E-mail
QQ	varchar(10)	gb2312_chinese_ci		否			QQ
code	varchar(10)	gb2312_chinese_ci		否			邮编
address	varchar(200)	gb2312_chinese_ci		否			地址
addtime	date			否			注册时间
isfreeze	int(1)			否	0		是否冻结
shopping	varchar(200)	gb2312_chinese_ci		否			购物车信息

图 26.13 用户信息表结构

此外还有友情链接表和商品评论表，限于篇幅，这里不再介绍，读者可参考本书配套光盘中的数据库文件。

## 26.6 搭建系统框架

编写代码之前，可以把系统中可能用到的文件夹先创建出来（如创建一个名为 images 的文件夹，

用于保存程序中所使用的图片），这样不但可以方便以后的开发工作，也可以规范系统的整体架构。因为本项目使用的是 Smarty+ADODB 技术，所以目录较多。下面来看本系统的目录结构（到三级目录），如图 26.14 所示。

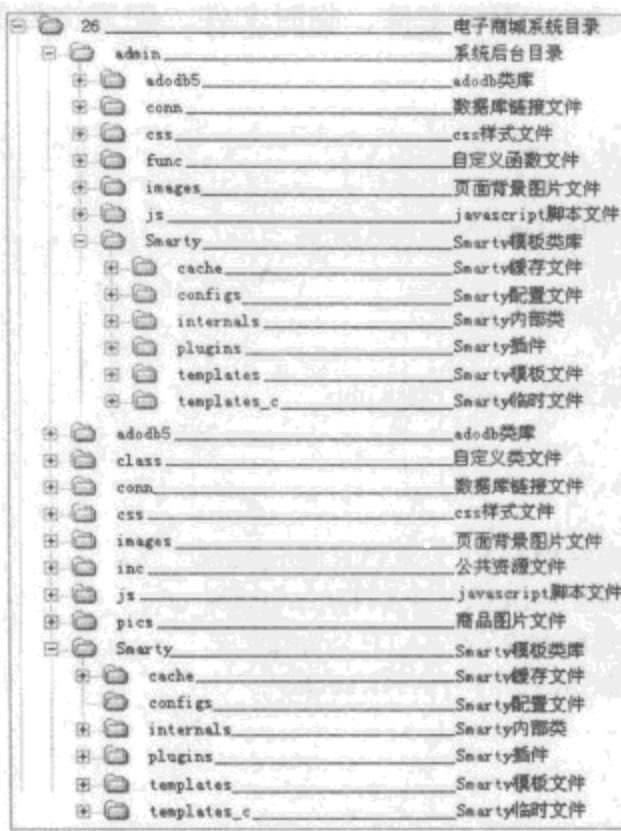


图 26.14 系统目录结构图

## 26.7 公共文件设计

### 视频讲解：光盘\TM\Video\26\公共文件设计.exe

公共文件就是将多个页面都可能使用到的代码写成单独的文件，在使用时只要用 include 或 require 将文件包含进来即可。如本系统中的数据库连接文件、Smarty 配置文件、CSS 样式文件、JS 脚本文件等。以前台系统为例，下面给出主要的公共文件。后台的公共文件与前台大同小异。

### 26.7.1 数据库连接文件

数据库连接文件，就是在读取或存储数据时连接数据库所使用到的函数。conn.php 文件代码如下。

```
<?php
include_once ("adodb5/adodb.inc.php");
$conn = ADONewConnection('mysql');
$conn->PConnect('localhost','root','root','db_database26') or die('connection error');
$conn->Execute('set names gb2312');
$ADODB_FETCH_MODE = ADODB_FETCH_BOTH;
?>
```

//载入 ADODB 类库  
//连接 MySQL  
//db\_database26 库  
//设置编码  
//设置结果集存储方式

## 26.7.2 Smarty 配置文件

Smarty 配置文件主要配置 Smarty 模板文件、临时文件、配置文件等文件路径。config.php 文件代码如下。

```
<?php
/* 定义服务器的绝对路径 */
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);
/* 定义 Smarty 目录的绝对路径 */
define('SMARTY_PATH','tm\sl\26\Smarty\\');
/* 载入 Smarty 类库文件 */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/* 实例化一个 Smarty 对象 */
$smarty = new Smarty;
/* 定义各个目录的路径 */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache';
/* 调试控制台 */
//$smarty->debugging = true;
/* Smarty 缓存 */
//$smarty->caching = true;
?>
```

## 26.7.3 XMLHttpRequest 对象文件

本系统应用了 Ajax 技术，所以预先创建了 XMLHttpRequest 对象，creat xmlhttp.js 文件代码如下。

```
var xmlhttp = false;
if(window.ActiveXObject){
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); //声明 xmlhttp 变量
} else if(window.XMLHttpRequest){
    xmlhttp = new XMLHttpRequest(); //判断 IE 版本
}
```

//根据不同的版本来创建对象

## 26.7.4 表单样式文件

本系统使用了大量的表格、表单，所以预先定义了表单与表格的样式。table.css 样式文件的代码如下。

```
<!-- 超链接样式 -->
.lk:link {
    text-decoration: none;
    color: #350407;
```

```
}

.lk:visited {
    text-decoration: none;
    color: #350407;
}

.lk:hover {
    text-decoration: none;
}

.lk:active {
    text-decoration: none;
    color: #CCCCFF;
}

<!-- 表格首行样式 -->
.first {
    background-color: #FFC4C4;
    border-bottom-width: 1px;
    border-bottom-style: solid;
    border-bottom-color: #FFC4C4;
    color: #000000;
    font-size: 14px;
    font-weight: bold;
}

<!-- 左侧表格样式 -->
.left {
    border-left-width: 1px;
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-left-style: solid;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-color: #FFC4C4;
    border-right-color: #FFC4C4;
    border-bottom-color: #FFC4C4;
}

<!-- 中间表格样式 -->
.center {
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-right-style: solid;
    border-bottom-style: solid;
    border-right-color: #FFC4C4;
    border-bottom-color: #FFC4C4;
}

<!-- 右侧表格样式 -->
.right {
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-right-style: solid;
    border-bottom-style: solid;
    border-right-color: #FFC4C4;
```

```

        border-bottom-color: #FFC4C4;
    }
<!-- 文本框样式 -->
.txt {
    margin: 0px;
    padding: 0px;
    height: 18px;
    width: 100px;
    border: 1px solid #979797;
}
<!-- 小文本样式 -->
.shorttxt {
    margin: 0px;
    padding: 0px;
    height: 18px;
    width: 50px;
    border: 1px solid #979797;
}
<!-- 长文本样式 -->
.langtxt {
    margin: 0px;
    padding: 0px;
    height: 18px;
    width: 200px;
    border: 1px solid #979797;
}
<!-- 文本域样式 -->
.areatxt {
    margin: 0px;
    padding: 0px;
    border: 1px solid #979797;
}
<!-- 按钮样式 -->
.btn {
    font-size: 12px;
    color: #350407;
    background-color: #FFFFFF;
    margin: 0px;
    padding: 0px;
    height: 15px;
    width: 50px;
    border: 1px solid #000000;
}
<!-- 整个表格样式 -->
.all{
    border: 1px solid #979797;
}

```

本系统的公共文件介绍完毕，下面讲解本系统中重点模块的实现方法。

## 26.8 前台首页设计

### ■ 视频讲解：光盘\TM\Video\26\前台首页设计.exe

前台首页一般没有多少实质性的技术，主要是加载一些功能模块，如登录模块、导航栏模块、公告栏模块等，使浏览者能够了解网站内容和特点。首页的重要之处是合理地对页面进行布局，既要尽可能地将重点模块显示出来，同时又不能因为页面凌乱无序，而让浏览者无所适从、产生反感。本系统前台首页 index.php 的运行结果如图 26.15 所示。



图 26.15 前台首页运行结果

本页面使用的是表格布局，除了商品模块和登录模块外，其他模块都是固定不变的。登录模块在会员成功登录后，将显示会员信息模块；而商品模块的位置，则根据不同的链接而显示不同的页面。index.php 页面的核心代码如下。

```
<?php
    session_start(); /* 开启 Session */
?>
<table width="850" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <!-- 载入导航栏模块 -->
        <td colspan="2"><?php include_once 'top.php'; ?></td>
    </tr>
    <tr><td width="216" align="left" valign="top">
```

```

<?php
    /* 载入登录模块、公告栏模块和友情链接模块 */
    include_once 'login.php';
    include_once 'public.php';
    include_once 'links.php';
?>
</td>
<td width="634" height="848" align="center" valign="top">
<?php
    /* 载入查询模块 */
    include_once 'search.php';
?>
<?php
/* 根据不同的 page 值，载入不同的模块 */
switch($_GET["page"]){
    case "":
        include_once 'nominate.php';                                //载入商品模块
        include_once 'newhot.php'; break;
    case "hyzx":include_once "member.php"; break;                //载入会员中心模块
    case 'allpub': include_once 'allpub.php'; break;             //显示公告栏模块
    case 'nom': include_once 'allnom.php'; break;                //显示推荐商品模块
    case 'new': include_once 'allnew.php'; break;                 //显示最新商品模块
    case 'hot': include_once 'allhot.php'; break;                 //显示热门商品模块
    case 'shopcar': include_once 'myshopcar.php' break;;          //显示购物车模块
    case 'settle': include_once 'settle.php'; break;              //收银台模块
    default:
        include_once 'nominate.php';                                //默认显示商品模块
        include_once 'newhot.php';
    }
?></td></tr>
</table>
<table width="850" border="0" cellspacing="0" cellpadding="0">
    <tr><td><?php include_once 'button.html'; ?></td></tr>      //底部导航栏模块
</table>

```

下面详细讲解其中几个重点模块的实现过程。只要将这几个模块所使用的技术和方法掌握好，其他模块的实现也就迎刃而解了。

 **说明：**本系统的功能较多，结构比较复杂，对于初学者来说学起来可能会比较困难。所以，本书将系统中的各个功能模块所涉及的文件（PHP、TPL、CSS、JS等）尽可能都单独实现。读者在学习其中某个模块时，可以将相关的文件统一放到一个目录下单独测试。

## 26.9 登录模块设计

用户登录模块是会员进入的窗口。匿名用户虽然也可以访问本网站，但只能进行浏览、查询等简单操作，而会员则可以购买商品，并且能享受商品折扣。登录模块包括用户注册、用户登录和找回密

码 3 部分。

### 26.9.1 用户注册

#### 视频讲解：光盘\TM\Video\26\用户注册.exe

用户注册通过一个 form 表单，将新用户填写的完整、合理、正确的信息保存到数据库中。如果用户填写的信息不正确或者不符合要求，则直接显示错误原因，以便用户修改。本系统的登录模块使用目前比较流行的 Ajax 技术来验证用户输入的数据，应用 GD2 函数库生成验证码。本页面的运行结果如图 26.16 所示。

图 26.16 注册模块页面

由于篇幅的限制，这里只给出有代表性的程序块，即用户名验证和验证码验证这两部分，而其他信息的验证方法在这两个程序块中都使用到了。

#### 1. 创建模板

首先创建模板文件。该模板主要包括注册表单和 div 标签。其中，div 标签是用来提示录入的回馈信息。模板文件 register.tpl 的核心代码如下。

```
<!-- 载入 xmlhttp.js 文件和 check.js 文件 -->
<script language="javascript" src="js/creat xmlhttp.js"></script>
<script language="javascript" src="js/check.js"></script>
<!-- 注册表单部分 -->
<form id="register" name="register" action="reg_chk.php" method="post" onSubmit="return chkinput(this)">
    <!-- 用户名验证部分 -->
    <tr>
        <td width="81" height="25"><div align="right">用户名: </div></td>
        <td height="25" colspan="3">&nbsp;
            <!-- 文本框，将会触发 onBlur 事件，调用 chkname() 方法 -->
            <input id="name" name="name" type="text" onBlur="javascript:chkname(register)" />
            <!-- 隐藏域，默认值为“not”即不允许注册 -->
            <input id="c_name" name="c_name" type="hidden" value="not" style="color: red;" />
        </td>
    </tr>
```

```

<!-- div 卷标用于显示对用户名的检测结果 -->
<td height="25"><div id="name1"><font color="#999999">请输入用户名</font></div></td>
</tr>
<tr>
    <td height="25" colspan="2">&nbsp;
        <input type="submit" value="提交"/>
        &nbsp;&nbsp;
        <input type="reset" value="重写" /></td>
    <td height="25" colspan="3"><div style="color:#FF0000">带“*”号的为必填项</div></td>
</tr>
</form>
</table>

```

这里包含了两个 js 文件：creat xmlhttp.js 和 check.js。其中，对 creat xmlhttp.js 文件在 26.7 节中已作了详细的讲解，而 check.js 文件用于实现对用户名和验证码的验证过程。

## 2. 实现用户名验证

用户名验证包括两个部分。第一部分是当用户输入名称后，焦点离开当前文本框时所触发的，触发的函数为 chkname(form)。代码如下。

```

/* form 为传入的表单名称，本段代码为 register 表单 */
function chkname(form){
    /* 如果 name 文本域的信息为空，名为 name1 的 div 标签显示如下信息 */
    if(form.name.value==""){
        name1.innerHTML=<font color="#FF0000">请输入用户名！</font>;
    }else{
        /* 否则 获取文本域的值 */
        var user = form.name.value;
        /* 生成 url 链接，将 user 的值传到 chkname.php 页进行判断 */
        var url = "chkname.php?user="+user;
        /* 使用 XMLHttpRequest 技术运行页面 */
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
            if(xmlhttp.readyState == 4){
                /* 根据不同的返回值，在 div 标签中输出不同信息 */
                var msg = xmlhttp.responseText;
                if(msg == '2'){
                    name1.innerHTML=<font color="#FF0000">用户名被占用！</font>;
                    return false;
                }else if(msg == '1'){
                    name1.innerHTML=<font color=green>恭喜您，可以注册！</font>;
                /* 如果用户名正确，则将隐藏域的值改为 “yes” */
                    form.c_name.value = "yes";
                }else{
                    name1.innerHTML=<font color=green>未知错误</font>;
                }
            }
        }
        xmlhttp.send(null);
    }
}

```

```

    }
}

```

这个函数调用了 chkname.php 页，该页在会员登录时也会被调用，所以这里分两种情况：有密码和无密码。无密码为注册验证，当没有返回结果时，说明该用户名可用；有密码为登录验证，和无密码相反，只有查询记录存在时，才允许登录，并将用户名和用户 ID 存储到 Session 中。该页面代码如下：

```

<?php
session_start()                                //开启 Session 支持
include_once "conn/conn.php";                   //加载数据库连接文件
$reback = '0';                                  //声明返回变数，初值为 '0'
$sql = "select * from tb_user where name=\"".$_GET['user']."\""; //生成 SQL 语句
/* 下面的 SQL 语句加上了对密码的比较，这是用来验证登录的，因为登录验证使用的也是本页面 */
$password = $_GET['password'];
if(!empty($password)){
    $sql .= " and password = \"".md5($password).\"";
}                                                 //如果密码不为空，说明是登录验证
/* 执行 SQL 语句 */
$rst = $conn->Execute($sql) or die('execute error');
if($rst->RecordCount() == 1){                  //根据 name 字段查询数据库
    /* 登录所用 */
    if($rst->fields['isfreeze'] != 0){
        $reback = '3';
    }else{
        $_SESSION['member'] = $rst->fields['name'];
        $_SESSION['id'] = $rst->fields['id'];
        $reback = '2';
    }
}else{
    $reback = '1';
}
echo $reback;
?>

```

上面的过程是第一次对用户名进行判断。当用户无视警告信息单击“提交”按钮时，就会触发<form>表单中的 onsubmit 事件，该事件触发 chkinput() 函数，此函数通过隐藏域来判断该注册名称是否合法。程序代码如下：

```

function chkinput(form){
    if(form.name.value==""){
        //判断用户名文本域是否为空
        form.name.focus();
        return false;
    }
    if(form.c_name.value == "not"){
        //判断隐藏域 c_name 的值是否为 not
        form.name.select();
        return false;
    }
    ...
}

```

运行结果如图 26.17 所示。

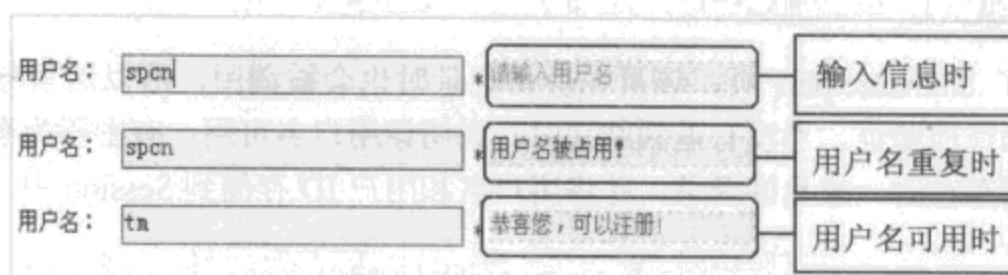


图 26.17 用户名验证过程

### 3. 验证码验证

验证码验证，先来看 register.tpl 模板是如何显示验证码的，代码如下：

```
<!-- 验证码验证部分 -->
<tr>
    <td height="25"><div align="right">验证码: </div></td>
    <td width="65" height="25">&nbsp;
        <!-- 验证码输入框，会触发 onBlur 事件 -->
        <input id="yzm" type="text" name="yzm" size="8" onBlur="javascript:chkyzm(register)" />
        <!-- 隐藏域，保存验证码 -->
        <input name="yzm2" type="hidden" value="" /></td>
        <!-- 验证码显示区 -->
        <td width="65" align="center" valign="middle"><script>yzm(register)</script></td>
        <!-- 在看不清验证码的时候更换验证图片 -->
        <td width="51"><a href="javascript:code()">看不清</a></td>
        <!-- div 标签，用于显示验证信息 -->
        <td height="25"><div id="yzm1"><font color="#999999">输入验证码</font></div></td>
    </tr>
    ...

```

通过模板页代码可以看到，模板页通过调用 yzm() 函数来显示验证码。yzm() 函数的代码如下：

```
function yzm(form){
    var num1=Math.round(Math.random()*10000000);           //产生随机数
    var num=num1.toString().substr(0,4);                     //获取前 4 位数
    document.write("<img name=codeimg src='yzm.php?num="+num+"'">"); //输出图片形式的验证码
    form.yzm2.value=num;                                     //将验证码保存到隐藏域中
}
```

在 yzm() 函数内，使用 Math.round() 函数生成随机码，并截取前 4 位，然后将验证码提交给 yzm.php 页进行转换，并显示由 yzm.php 生成的 png 图片。最后将验证码存储到名字是 yzm2 的隐藏域中，用作验证码检查。yzm.php 页面使用的是 GD2 函数库。生成验证码的代码如下：

```
<?php
 srand((double)microtime()*1000000);
 $im=imagecreate(42,16);
 $black=imagecolorallocate($im,0,0,0);
 $white=imagecolorallocate($im,255,255,255);
 $gray=imagecolorallocate($im,200,200,200);
 //生成随机数
 //创建画布
 //定义背景
 //定义背景
 //定义背景
```

```

imagefill($im,0,0,$gray); //填充颜色
for($i=0;$i<4;$i++){
    $str=mt_rand(1,3); //定义 4 位随机数
    $size=mt_rand(3,6); //定义随机字符所在位置的 Y 坐标
    $authnum=substr($_GET[num],$i,1); //定义随机字符的字体
    $authnum=imagecolorallocate($im,rand(0,130),rand(0,130),rand(0,130)); //获取超链接中传递的验证码
    imagestring($im,$size,(2+$i*10),$str,$authnum,imagecolorallocate($im,rand(0,130),rand(0,130),rand(0,130))); //水平输出字符串
}
for($i=0;$i<200;$i++){ //执行 for 循环，为验证码添加模糊背景
    $randcolor=imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255)); //创建背景
    imagesetpixel($im,rand()%70,rand()%30,$randcolor); //绘制单一元素
}
imagepng($im); //生成 png 图像
imagedestroy($im); //销毁图像
?>

```

当用户输入验证码后，通过 onBlur 事件的 chkzym() 函数进行检查。chkzym 函数的代码如下：

```

function chkzym(form){
    if(form.yzm.value==""){
        <!-- 当输入框信息为空时 --&gt;
        yzm1.innerHTML=&lt;font color="#FF0000&gt;请输入效验码！&lt;/font&gt;";
    }else if(form.yzm.value!=form.yzm2.value){
        <!-- 当用户输入与验证码不符时 --&gt;
        yzm1.innerHTML=&lt;font color="#FF0000&gt;效验码输入错误!&lt;/font&gt;";
    }else{
        yzm1.innerHTML=&lt;font color=green&gt;输入正确&lt;/font&gt;";
    }
}
</pre>

```

除了验证输入，还可以使用 code() 函数重新更换验证码。code() 函数的内容几乎和 yzm() 函数完全相同，只是显示验证码的方法不一样。code() 函数的代码如下：

```

function code(){
    var num1=Math.round(Math.random()*10000000);
    var num=num1.toString().substr(0,4);
    document.codeimg.src="yzm.php?num="+num;
    form.yzm2.value=num;
}

```

运行结果与用户名验证大同小异，这里不再给出。

#### 4. 保存注册

如果用户输入的信息准确无误，系统就会跳转到处理页面（reg\_chk.php）来连接数据库，并保存用户数据。保存完毕后，自动使用新用户登录系统。reg\_chk.php 页面的代码如下：

```

<?php
session_start(); //开启 Session 支持
include_once 'conn/conn.php'; //加载数据库连接文件
/* 获取表单信息 */
$name = $_POST['name'];

```

```

$password = md5($_POST['pwd1']);           //使用 MD5 给密码加密
$question = $_POST['question'];
$answer = $_POST['answer'];
$realname = $_POST['realname'];
$card = $_POST['card'];
$tel = $_POST['tel'];
$phone = $_POST['phone'];
$Email = $_POST['email'];
$QQ = $_POST['qq'];
$code = $_POST['code'];
$address = $_POST['address'];
$addtime = $conn->DBDate(time());          //使用 DBDate() 函数生成当前时间
/****************************************/
/* 生成 SQL 语句 */
$sql = "insert into tb_user ( name, password, question, answer, realname, card, tel, phone, Email, QQ, code, address, addtime, isfreeze, shopping) ";
$sql .= " values ('$name', '$password', '$question', '$answer', '$realname', '$card', '$tel', '$phone', '$Email', '$QQ', '$code', '$address', '$addtime', '0', '')";
/* 执行 SQL 语句 */
$rst = $conn->execute($sql);
/* 判断添加是否成功 */
if($rst == false){
    echo '<script>alert("添加失败");history.back;</script>';
} else {
    /* 如果成功，则将用户 ID 和用户名存到 Session 中 */
    $_SESSION['member'] = $name;
    $_SESSION['id'] = $conn->Insert_ID();          //Insert_ID() 函数获取新增数据的 ID 号
    echo "<script>top.opener.location.reload();alert('注册成功');window.close();</script>";
}
?>

```

## 5. 加载模板

因为本系统使用了 Smarty，所以还需要创建一个 PHP 页来加载模板文件。register.php 页面的内容非常简单，代码如下：

```

<?php
    include_once 'config.php';                  //载入 config.php 文件
    $smarty->assign('title','新用户注册');      //设置 title
    $smarty->display('register.tpl');          //显示 register.tpl
?>

```

## 6. 创建链接

以上就是用户注册模块的核心代码。当测试没有问题后，最后创建一个“注册”超链接。当用户单击前台的 **注册** 按钮时，系统会调用 js 的 onclick 事件，弹出注册窗口。代码如下：

```
<a href="#" id="login" onclick="reg0">用户注册</a>
```

这里使用到的 js 文件为 js/login.js。调用的函数为 reg0，该函数代码如下：

```
function reg(){
    window.open("register.php", "_blank", "width=500,height=450", false); //弹出新窗口
}
```

## 26.9.2 用户登录

### 视频讲解：光盘\TM\Video\26\用户登录.exe

除了在注册时可以直接登录，其他情况下用户可以通过登录区进行登录操作。用户登录模块的运行结果如图 26.18 所示。

#### 1. 信息验证

用户登录同样使用的是 Ajax 验证方法。首先创建页面模板文件 login.tpl，该文件主要包含一个登录表单。登录表单各元素的属性值及超链接说明如表 26.1 所示。



图 26.18 用户登录页面

表 26.1 登录表单各元素属性说明

元素名称	元素属性	元素说明
<form>	id = "login" name = "login" method = "post" action = "#" onsubmit = "return lg( this )"	登录表单可以触发 onsubmit 事件来调用 lg() 函数，表单名称为 login
text	id="name" name="name" type="text"	用户名文本框
text	id="password" name="password" type="password"	密码文本框
submit	id="enter" name="enter" type="submit"	登录按钮
<a>看不清</a>	onclick="javascript:code(login)" style="cursor:hand"	更换验证码图片
hidden	name="check2" type="hidden" value=""	保存验证码的隐藏域
<a>用户注册</a>	id="login" href="#" onclick="reg()"	注册页面超链接
<a>找回密码</a>	id="login" href="#" onclick="found()"	找回密码超链接

通过表 26.1 可知，当单击 submit 按钮时，系统将调用 lg() 函数。lg() 函数包含在 js/login.js 脚本文件内，该函数代码如下：

```
function lg(form){
    /* 检验用户名 */
    if(form.name.value==""){
        alert('请输入用户名');
        form.name.focus();
        return false;
    }
    /* 检验密码 */
    if(form.password.value == "" || form.password.value.length < 6){
        alert('请输入正确密码');
        form.password.focus();
        return false;
    }
    /* 检验验证码 */
    if(form.check.value == ""){
        alert('请输入验证码');
        form.check.focus();
        return false;
    }
    /* 提交表单 */
    document.login.submit();
}
```

```

        alert('请输入验证码');
        form.check.focus();
        return false;
    }
    /* 比较验证码 */
    if(form.check.value != form.check2.value){
        form.check.select();
        code(form);
        return false;
    }
    /* 根据输入的用户名和密码生成 url */
    var user = form.name.value;
    var password = form.password.value;
    var url = "chkname.php?user="+user+"&password="+password;
    /* 使用 xmlhttp 对象来返回验证结果 */
    xmlhttp.open("GET",url,true);
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
            var msg = xmlhttp.responseText;
            /* 如果返回值不等于 2，说明用户名或密码错误 */
            if(msg == '1'){
                alert('用户名或密码错误!!');
                form.password.select();
                form.check.value = "";
                code(form);
                return false;
            }if(msg == "3"){
                alert("该用户被冻结，请联系管理员");
                return false;
            } else{ /* 如果成功，重新刷新本页 */
                alert('欢迎光临');
                location.reload();
            }
        }
    }
    xmlhttp.send(null);
    return false;
}

```

用户名和密码是在 chkname.php 页面中被验证的。chkname.php 的相关内容在 26.9.1 节中已经介绍过，这里不再赘述。

## 2. 用户信息

用户登录成功后，在原登录框位置将显示用户信息模块。用户可以通过“会员中心”超链接对自己的信息进行修改；也可以单击“查看购物车”超链接查看购物车商品；用户还可以通过单击“安全离开”超链接离开。关于会员信息模块，将在 26.10 节中进行讲解。用户信息模块的主要代码如下：

```

<!-- 显示当前登录用户名 -->
欢迎您: {$member}

```

```
<!-- 会员中心超链接 -->
<a href="?page=hyzx" id="info" class="lk">会员中心</a>
<!-- 查看购物车 -->
<a href="?page=shopcar" class="lk">查看购物车</a>
<!-- 安全离开 -->
<a onclick="javascript:logout()" style="cursor:hand" id="info">安全离开</a>
```

用户信息页面的运行结果如图 26.19 所示。

### 3. 加载模板

如果希望看到这个页面，首先必须要载入 login.php 页。该页面的作用是根据 Session 值自动选择显示的模块，包括 26.9.1 节介绍的登录模块和本节介绍的用户信息模块。login.php 页面的代码如下：

```
<?php
    session_start(); //开启 Session 支持
    include_once 'config.php'; //加载 Smarty 配置文件
    if($_SESSION['member'] == ""){ //判断 Session 值是否为空
        $smarty->display('login.tpl'); //如果为空，显示登录模块
    }else{
        $smarty->assign('member',$_SESSION['member']); //否则显示用户信息模块
    }
?>
```

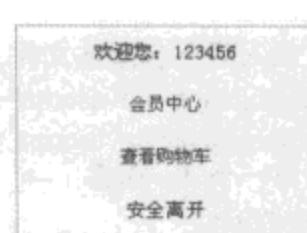


图 26.19 用户信息页面的运行结果

### 26.9.3 找回密码

#### 视频讲解：光盘\TM\Video\26\找回密码.exe

登录模块的最后一个部分就是找回密码。找回密码是根据用户在填写资料时所填写的密保问题和密保答案来实现的。当用户单击“找回密码”超链接时，系统首先提示用户输入要找回密码的会员名称，然后用户根据密保问题填写密保答案，最后重新输入密码。找回密码的流程如图 26.20 所示。

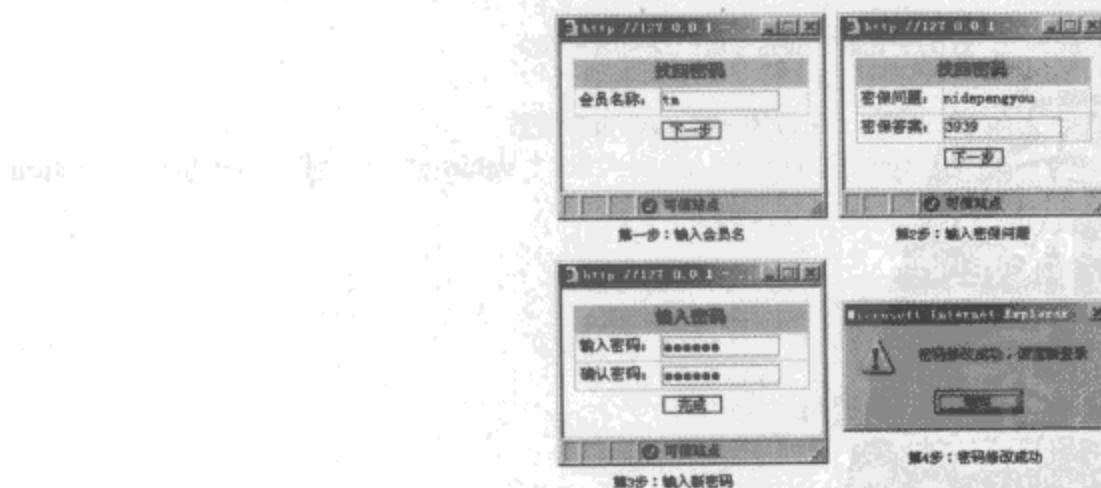


图 26.20 找回密码流程图

## 1. 创建模板文件

虽然找回密码需要 4 个步骤，但实际上每个步骤使用的都是相同的模板文件和 js 文件，只是被调用的表单和 js 函数略有差别。这里根据不同的文件分别进行介绍。

首先来看模板文件。该模板文件一共包含了 3 个表单，分别代表了 3 个步骤。下面是模板文件的核心代码。

```
<!-- 载入两个 js 脚本文件 -->
<script language="javascript" src="js/creatxmlhttp.js"></script>
<script language="javascript" src="js/found.js"></script>
<!-- 第一个 div 标签 -->
<div id="first">
<table width="200" border="0" cellspacing="0" cellpadding="0">
<form id="foundname" name="found" method="post" action="#">
    <tr><td> 找回密码</td></tr>
    <tr><td>会员名称: </td>
        <!-- text 文本域，用于输入要找回密码的会员名称 -->
        <td><input id="user" name="user" type="text" class="txt"></td>
    </tr>
    <tr><td>
        <!-- 单击“下一步”按钮，能触发 onclick 事件来调用 chkname() 函数 -->
        <input id = "next1" name = "next1" type = "button" class = "btn" value = "下一步" onClick = "return chkname( foundname )"/></td></tr>
    </form>
</table>
</div>
<!-- 第二个 div 标签，样式为隐藏 -->
<div id="second" style="display:none;">
<table>
<form id="foundanswer" name="found" method="post" action="#">
    <tr><td> 找回密码</td></tr>
    <tr><td>密保问题: </td>
        <!-- 用于显示密保问题的 div 标签 -->
        <td <div id="question"></div></td></tr>
    <tr><td>密保答案: </td>
        <!-- 文本域，用于填写密保答案 -->
        <td><input id="answer" name="answer" type="text" class="txt" /></td></tr>
    <br>
        <!-- 单击“下一步”按钮，用来触发 onclick 事件，并调用 chkanswer() 函数 -->
        <td><input id = "next2" name = "next2" type = "button" class = "btn" value = "下一步" onClick = "return chkanswer( foundanswer )"/></td>
    </tr>
</form>
</table>
</div>
<!-- 第三个 div 标签，样式也为隐藏，作用是修改密码 -->
<div id="third" style="display:none;">
<table>
<form id="modifypwd" name="found" method="post" action="#">
```

```

<tr><td> 输入密码</td></tr>
<tr><td>输入密码: </td>
    <td><input id="pwd1" name="pwd1" type="password" class="txt"></td></tr>
<tr><td>确认密码: </td>
    <td><input id="pwd2" name="pwd2" type="password" class="txt" /></td>
</tr>
<tr>
<!-- 单击“完成”按钮，调用 chkpwd() 函数 --&gt;
    &lt;td&gt;&lt;input id = " mod " name = " mod " type = " button " class = " btn " value = " 完成 " onClick = " return chkpwd(modifypwd) "&gt;&lt;/td&gt;
&lt;/tr&gt;
&lt;/form&gt;
&lt;/table&gt;
&lt;/div&gt;
</pre>

```

可以发现，3个表单中，只有一个表单默认情况下是显示的，其他则为隐藏。只有通过调用不同的js函数，才可以对其他表单进行操作。

## 2. 创建 js 脚本文件

found.js 脚本文件包含 3 个函数：chkname()、chkanswer() 和 chkpwd()。其中 chkname() 函数的作用是检查用户输入的会员名称，如果存在，则使用 xmlhttp 对象去调用生成的 url 进行判断。如果该用户存在，则隐藏当前表单，并显示下一个表单，最后输出密保问题。chkname() 函数的代码如下：

```

function chkname(form){
    /* 获取文本框信息 */
    var user = form.user.value;
    /* 如果为空，则输出提示 */
    if(user == ""){
        alert('请输入用户名');
        form.user.focus();
        return false;
    }else{
        /* 否则，生成 url，并调用 xmlhttp 对象 */
        var url = "foundpwd.php?user="+user;
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
            if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                /* 如果没有结果，则提示 */
                if(msg == '0'){
                    alert('没有该用户，请重新查找!');
                    form.user.select();
                    return false;
                }else{
                    /* 否则，隐藏当前表单，显示下一个表单，并输出密保问题 */
                    document.getElementById('first').style.display = 'none';
                    document.getElementById('second').style.display = '';
                    document.getElementById('question').innerHTML = msg;
                }
            }
        }
    }
}

```

```
        }
    }
    xmlhttp.send(null);
}
}
```

其他两个函数也使用了 XMLHttpRequest 对象，实现方法相差无几，不同之处在于对返回值的处理。chkanswer() 函数隐藏当前表单，显示下一个表单。chkanswer() 函数的代码如下：

```
function chkanswer(form) {
    /* 获取已隐藏表单数据 */
    var user = document.getElementById('user').value;
    /* 获取密保答案 */
    var answer = form.answer.value;
    /* 表单验证 */
    if(answer == ""){
        ...
    }else{
        /* 生成 url */
        var url = "foundpwd.php?user="+user+"&answer="+answer;
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
            if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                if(msg == '0'){
                    alert('问题回答错误');
                    form.answer.select();
                    return false;
                }else{
                    document.getElementById('second').style.display = 'none';
                    document.getElementById('third').style.display = '';
                }
            }
        }
        xmlhttp.send(null);
    }
}
```

而 chkpwd() 函数则弹出提示信息，如果成功，则关闭当前页。chkpwd() 函数的代码如下：

```
function chckpwd(form){  
    /*  获取名称和密码 */  
    var user = document.getElementById('user').value;  
    var pwd1 = form.pwd1.value;  
    var pwd2 = form.pwd2.value;  
    /* 省略部分为表单验证 */  
    ...  
    var url = "foundpwd.php?user="+user+"&password="+pwd1;  
    xmlhttp.open("GET",url,true);  
    xmlhttp.onreadystatechange = function(){
```

```

if(xmlhttp.readyState == 4){
    var msg = xmlhttp.responseText;
    if(msg == '1'){
        alert('密码修改成功，请重新登录');
        window.close();
    }
}
xmlhttp.send(null);
}

```

### 3. 创建数据处理文件

foundpwd.php 文件的功能是根据用户输入信息来检测数据表中的数据，并根据不同的输入项返回不同的结果，代码如下。

```

<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    $name= $_GET['user']; //获取表单值
    $answer = $_GET['answer'];
    $password = $_GET['password'];
    $reback = '0';
    /* 如果变量$answer 和$password 为空，说明用户只输入了会员名称 */
    if(empty($answer) && empty($password)){
        /* 生成查询语句 */
        $namesql = "select * from tb_user where name = '".$name."'";
        /* 获取查询结果 */
        $namerst = $conn->execute($namesql);
        /* 如果查询记录为 1，说明输入正确 */
        if($namerst->recordCount() == 1){
            /* 获取密保问题，并赋给变量$reback */
            $question = $namerst->fields['question'];
            $reback = $question;
        }
        /* 如果变量$answer 不为空，说明用户输入了密保答案 */
    }else if(!empty($answer)){
        /* 根据密保答案和会员名称，生成查询语句 */
        $answersql = "select * from tb_user where name = '".$name."' and answer = '".$answer."'";
        /* 返回查询结果，并判断是否有返回记录，如果有，说明输入正确 */
        $answerrst = $conn->execute($answersql);
        if($answerrst->recordCount() == 1){
            $reback = '1';
        }
        /* 如果变量$password 不为空，说明用户输入了新密码 */
    }else if(!empty($password)){
        /* 根据用户名和新密码更新记录 */
        $sql = "select * from tb_user where name = '".$name."'";
        $arr = array();
        $rst = $conn->execute($sql);
        if($rst->RecordCount() == 1){

```

```

        $arr['password'] = md5($password);
        $updateSQL = $conn->GetUpdateSQL($rst,$arr,true);
        $conn->execute($updateSQL);
        $reback = '1';
    }
}
/* 输出返回值 */
echo $reback;
?>

```

#### 4. 加载模板页

因为所有登录模块的模板都是不需要、或者只需要传递一两个变量，所以 PHP 加载页的内容比较简单。找回密码页面的 PHP 加载文件是 found.php，代码可参考配套光盘中的内容。

## 26.10 会员信息模块设计

### 视频讲解：光盘\TM\Video\26\会员信息模块设计.exe

用户登录后，即可看到会员信息模块。在会员信息模块中，可以执行查看、修改个人信息及密码，查看购物车和安全退出等操作。本节只对会员信息模块中的“会员中心”和“安全退出”超链接进行讲解，关于“查看购物车”超链接将在商品模块中进行介绍。

### 26.10.1 会员中心

单击“会员中心”超链接，会回传给当前页一个 page 值，当前页根据这个 page 值来载入 member.php 文件。member.php 页面的运行结果如图 26.21 所示。

会员编号:	14
会员名称:	tm
密保问题:	nidejengyou
密保答案:	3939
注册时间:	2008-04-23
消费总额:	666634
<hr/>	
真实姓名:	spcn
身份证号:	220204198*****
移动电话:	13514402359
固定电话:	043112345678
Email:	tm@tm.com
QQ号:	12345678
邮编:	130000
地址:	吉林省吉林市

图 26.21 “会员中心”页面的运行结果

#### 1. 创建 PHP 页面

与登录模块设计不同，本节首先来创建 PHP 页面，因为该模块中的模板需要使用数据库中的数据

及一些动态信息，这些都需要在 PHP 页中先行获取及处理，然后再传给模板页。“会员中心”页面的代码如下。

```
<?php
    session_start();                                //开启 Session 支持
    include_once 'config.php';                      //载入 Smarty 配置文件
    include_once 'conn/conn.php';                    //载入数据库连接文件
    /*  查找用户资料 */
    $sql = 'select * from tb_user where id = '.$_SESSION['id'];
    $rst = $conn->execute($sql);
    $arr = $rst->GetArray();
    if($_GET['action'] == 'modify'){
        $smarty->display('modifypwd.tpl');
    }else{
        $smarty->assign('arr',$arr[0]);
        $smarty->display('membershow.tpl');
    }
?>
```

## 2. 创建模板页

该模块包括查看信息模板及修改密码模板。首先介绍查看信息模板，代码如下：

```
<!-- 显示超链接 -->
<p>{$smarty.session.member}<a href = '?page=hzyx'>查看信息</a><a href = '?page = hzyx&action=modify'
id="mem">修改密码</a></p>
<table>
<form id = "member" name = "member" method = "post" action = "modify_info_chk.php" onsubmit = "return
mem( member )">
<!-- 显示当前会员信息，此处为不可更改信息 -->
<tr><td colspan="2">{$arr.name}信息（不可更改信息）</td></tr>
<!-- 因为格式相同，所以只显示一条，其他省略 -->
<tr><td>会员编号：</td><td>&nbsp;{$arr.id}</td></tr>
...
<tr><td>{$arr.name}信息（可更改信息）</td></tr>
<!-- 可以修改的信息，放到表单元素中 -->
<tr><td>真实姓名：</td>
<td><input id="realname" name="realname" type="text" value="{$arr.realname}" /></td></tr>
<!-- 省略了部分代码..... -->
<tr><td>
    <input name="enter" type="submit" id="enter" value="修改" />
    <input name="reset" type="reset" id="reset" value="重置" />
</td></tr>
</form>
</table>
```

修改密码模板比信息模板要简单一些，因为不需要额外的传值。模板代码如下：

```
<!-- 载入脚本文件 -->
<script language="javascript" src="js/member.js"></script>
```

```

<p>{$smarty.session.member}<a href="?page=hyzx">查看信息</a><a href="?page=hyzx&action=modify" >修改密码</a></p>
<table>
<!-- 修改密码表单 -->
<form id = " member " name = " member " method = " post " action = " modify_pwd_chk.php " onsubmit = " return
pwd(member) ">
    <tr><td>修改密码</td></tr>
    <tr><td>原密码: </td><td><input id="old" name="old" type="password" /></td></tr>
    <tr><td>新密码: </td><td><input id="new1" name="new1" type="password" /></td></tr>
    <tr><td>确认密码: </td><td><input id="new2" name="new2" type="password" /></td></tr>
    <tr><td><input id="enter" name="enter" type="submit" value="修改" /></td></tr>
</form>
</table>

```

### 3. 创建脚本文件

该模块的脚本文件和用户注册模块类似，都是对信息的合法性进行验证，如信息是否为空、是否符合规范等。这里不给出代码，读者可以参考 26.9.1 节的内容。

### 4. 创建处理页

当信息验证通过后，系统将跳到处理页，进行信息处理。本模块的处理页分为信息修改和密码修改两个页面。首先介绍信息修改页，代码如下：

```

<?php
    session_start();                                //开启 Session 支持
    include_once 'conn/conn.php';                    //载入数据库连接文件
    $sql = 'select * from tb_user where id = '.$_SESSION['id']; //根据 ID 生成 SQL 查询语句
    $rst = $conn->execute($sql);                   //执行 SQL 语句，返回查询结果集
    $mod = array();                                  //声明数组
    /* 将修改信息添加到数组中 */
    $mod['realname'] = $_POST['realname'];
    $mod['card'] = $_POST['card'];
    $mod['tel'] = $_POST['tel'];
    $mod['phone'] = $_POST['phone'];
    $mod['Email'] = $_POST['email'];
    $mod['QQ'] = $_POST['qq'];
    $mod['code'] = $_POST['code'];
    $mod['address'] = $_POST['address'];
    *****
    $updateSQL = $conn->GetUpdateSQL($rst,$mod);      //生成更新语句
    if($conn->execute($updateSQL))
        echo "<script>alert('修改成功');location='index.php';</script>"; //修改成功，跳到首页
    else
        echo "<script>alert('修改失败');history.go(-1);</script>";           //否则返回上一页
?>

```

密码修改页的操作流程与信息修改页类似，只是更新的数组少，只有一个字段。修改密码页的代码如下：

```

<?php
    session_start(); //开启 Session 支持
    include_once 'conn/conn.php'; //载入数据库连接文件
    $oldpwd = md5($_POST['old']); //为旧密码进行 MD5 加密
    /* 根据 ID 和 MD5 加密后的密码查询用户 */
    $sql = 'select * from tb_user where id = "'.$_SESSION['id'].'" and password = "'.$oldpwd.'"; //返回查询结果
    $rst = $conn->execute($sql); //创建更新数组
    $mod = array();
    $mod['password'] = md5($_POST['new1']); //为新密码进行 MD5 加密
    $updateSQL = $conn->GetUpdateSQL($rst,$mod); //生成更新语句
    if($conn->execute($updateSQL)) //修改成功，返回首页
        echo "<script>alert('修改成功');location='index.php';</script>";
    else //否则返回上一页
        echo "<script>alert('修改失败');history.go(-1);</script>";
?>

```

## 26.10.2 安全退出

当用户需要离开网站时，可以单击“安全退出”超链接来调用 logout() 函数，当确认用户退出后，跳转到 logout 页面，销毁 Session，并回到首页。安全退出所涉及的页面及代码如下：

```

function logout(){
    if(confirm("确定要退出登录吗？")){
        window.open('logout.php','_parent',"false"); //输出选择框，用户可以单击“确认”或“取消”按钮
    }else
        return false;
}
<?php
    session_start(); //开启 Session 支持
    session_destroy(); //销毁 Session
    echo '<script>alert("用户已安全退出!");location="index.php";</script>'; //回到首页
?>

```

## 26.11 商品展示模块

### 视频讲解：光盘\TM\Video\26\商品展示模块.exe

本系统根据商品的类别，为用户提供了多种不同的商品展示方式，包括推荐商品、最新商品和热门商品等，能够使消费者有目的地选购一些商品。商品信息对商品的一些属性进行了描述，还提供了商品的实物图片，为用户提供信息，让用户对商品有更多的了解。商品展示模块对电子商务平台来说尤为重要。本系统中商品展示模块的运行结果如图 26.22 所示。

因为推荐商品、最新商品和热门商品的实现方法和过程基本相同，所以本节只讲解推荐商品模块，其他的程序可参见本书配套光盘。



图 26.22 商品展示模块页面

### 26.11.1 创建 PHP 页

既然是商品展示模块，首先就是要取得商品记录。在数据库设计一节中已经介绍，商品字段 `isnom` 表示商品是否被推荐，如果该字段为 1，即为推荐，否则为不推荐。所以在 PHP 页面中，首先要获取数据表中 `isnom` 字段属性为 1 的记录，因为受页面限制，所以只返回 4 条记录。首页中的推荐商品页面的代码如下：

```
<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    include_once 'config.php'; //载入模板配置文件
    /* 根据 isnom 字段，生成 SQL 语句 */
    $sql = "select id,name,pics,m_price,v_price from tb_commo where isnom = 1";
    $num = 4; //设置提取记录条数
    $rst = $conn->SelectLimit($sql,$num); //使用 SelectLimit 方法获取记录集
    $nomarr = $rst->GetArray(); //将记录集保存为数组

    $smarty->assign('nomarr',$nomarr); //将数组传递到模板页
    $smarty->display('nominate.tpl'); //显示模板
?>
```

### 26.11.2 创建模板页

记录集数组被传递到模板页后，模板页使用 `foreach` 标签输出商品信息，并添加相应的操作按钮和超链接。模板页中一共有显示更多商品、查看商品和放入购物车 3 个事件。

- 当单击“更多商品”超链接时，将重新加载本页面，并传递一个 `page` 变量。`switch` 语句会根据 `page` 值来显示。
- 单击“查看商品”按钮时，将触发 `onclick` 事件，并调用 `openshowcommo()` 函数，同时，商品 ID 会作为函数的唯一参数被传递。
- 当单击“放入购物车”按钮时，同样会触发 `onclick` 事件，并调用 `buycommo()` 函数，唯一的

参数也是商品的 ID。商品模板页面的代码如下：

```
<!-- 载入 js 脚本文件 -->
<script language="javascript" src="js/creat xmlhttp.js"></script>
<script language="javascript" src="js/showcommo.js"></script>
<table border="0" cellpadding="0" cellspacing="0">
    <!-- “更多商品”超链接 -->
    <tr><td><a href="?page=new">&gt;&gt;more</a></td><td rowspan="3">&ampnbsp</td></tr>
    <tr><td>&ampnbsp</td>
    <!-- 使用 Smarty 的 foreach 标签，循环输出数组 -->
    {foreach key=key item=item from=$nomarr}
    <td>
        <table border="0" cellpadding="0" cellspacing="0" >
            <tr><td>
                <!-- 商品图片显示 -->
                </td></tr>
            <tr><td>&ampnbsp{$item.name}</td></tr>
            <tr><td>市场价： {$item.m_price}&ampnbsp元</td></tr>
            <tr><td>会员价： {$item.v_price}&ampnbsp元</td></tr>
            <tr><td>
                <!-- “查看商品”按钮 -->
                <input id = "showinfo" name = "showinfo" type = "button" value = "" onclick = "openshowcommo({$item.id})"/>
                <!-- “放入购物车”按钮 -->
                <input id = "buy" name = "buy" type = "button" value = "" onclick = "return buycommo({$item.id})" />
            </td></tr></table></td>
    <!-- 循环结束 -->
    {/foreach}
    <td>&ampnbsp</td></tr>
    <tr><td colspan="6">&ampnbsp</td></tr>
</table>
```

### 26.11.3 js 脚本页面

在模板页中的 3 个链接，其中两个都用到了 js 函数。通过 js 函数，当单击“查看商品”按钮时，系统会弹出一个新的页面，并显示商品的详细信息；当单击“购买”按钮时，该商品将会被放到当前用户的购物车中，如果没有登录用户或商品已添加，则弹出提示信息。js 脚本文件的代码如下：

```
/* 查看商品信息函数，将打开一个新页面 */
function openshowcommo(key){
    open('showcommo.php?id=' + key, '_blank', 'width=560 height=300', false);
}

/* 将购买商品添加到购物车中，将在 26.12 节中进行讲解 */
function buycommo(key){
    ...
}
```

显示更多商品列表和商品详细信息页的实现过程和 26.11.1 节、26.11.2 节所述类似，只是显示更多字段信息而已，这里不再赘述。26.12 节将重点讲解购物车的实现。

## 26.12 购物车模块设计

### 视频讲解：光盘\TM\Video\26\购物车模块设计.exe

购物车在电子商务平台中是非常关键的一个功能模块。购物车保留用户选择的商品信息，用户可以在购物车内设置选购商品的数量，显示选购商品的总金额，清除选择的全部商品，以及重新选择商品。购物车页面运行结果如图 26.23 所示。

我的购物车					
商品名称	购买数量	市场价格	会员价格	折扣率	合计
<input type="checkbox"/> 自行车	3	368	349.2	9	1047.6
<input type="checkbox"/> 数码相机	5	1888	1699.2	9	8496
<input checked="" type="checkbox"/> 洗衣机	1	2666	2399.4	9	2399.4
<input type="checkbox"/> 家庭影院	1	4888	4399.2	9	4399.2
全选	取消选择	继续购物	去收银台	共计：16342.2 元	

图 26.23 购物车页面运行结果

购物车模块主要实现以下功能：添加商品、删除商品、更改商品数量。

### 26.12.1 添加商品

在商品展示模块中，单击指定商品中的“购买”按钮，即可将物品放到购物车中，并进入到“购物车”页面。从 26.11 节中已经了解到“购买”动作调用的是 buycommo() 函数，购买商品的 ID 是该函数的唯一参数。下面就来看 buycommo() 函数的功能结构。此函数代码如下：

```
/*添加商品，同时检查用户是否登录、商品是否重复等 */
function buycommo(key){
    /* 根据商品 ID，生成 url */
    var url = "chklogin.php?key="+key;
    /* 使用 xmlhttp 对象调用 chklogin.php 页 */
    xmlhttp.open("GET",url,true);
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
            var msg = xmlhttp.responseText;
            /* 用户没有登录 */
            if(msg == '2'){
                alert('请您先登录');
                return false;
            }else if(msg == '3'){
                /* 商品已添加 */
                alert('该商品已添加');
                return false;
            }else{
                /* 显示购物车 */
            }
        }
    }
}
```

```

        location='index.php?page=shopcar';
    }
}
xmlhttp.send(null);

```

可以看到, buycommo() 函数通过 xmlhttp 对象调用 chklogin.php 页, 并根据回传值作出相应的处理, chklogin.php 页才是将商品添加到购物车中的操作页。chklogin.php 页的代码如下:

```

<?php
session_start(); //开启 Session 支持
/** 
 * 1 表示添加成功
 * 2 表示用户没有登录
 * 3 表示商品已添加过
 * 4 表示添加时出现错误
 * 5 表示没有商品添加
*/
include_once 'conn/conn.php'; //载入数据库连接文件
$reback = '0'; //声明返回值
if(empty($_SESSION['member'])){ //如果 Session 值为空, 说明没有登录
    $reback = '2'; //参见头注释
} else{ //获取商品 ID
    $key = $_GET['key'];
    if($key == ''){ //获取登录会员 ID 号
        $reback = '5';
    } else{ //设置一个布尔变量, 初始为 false
        $id = (int)$_SESSION['id'];
        $boo = false;
        $addshop = array(); //数据库更新数组
        $sql = "select id,shopping from tb_user where id = ".$id; //生成当前用户的查询语句
        $rst = $conn->execute($sql); //返回当前用户记录集
        $shopcont = $rst->fields['shopping']; //找到 shopping 字段
        if(!empty($shopcont)){ //如果字段不为空, 说明已有商品
            $arr = explode('@',$shopcont); //将字段用 "@" 拆分并另存数组
            foreach($arr as $value){ //循环输出数组内容
                if($key == $value[0]){ //判断需要添加的商品是否存在
                    $reback = '3'; //参见头注释
                    $boo = true; //同时布尔变量变为 true
                    break; //停止循环
                }
            }
        }
        if(!$boo){ //如果布尔值没有改变, 再添加商品
            /* 将商品数量默认设为 1, 连同商品 ID 保存到字串中 */
            $shopcont .= '@'.$key.',1';
            /* 将商品字串保存到更新数组中 */
            $addshop['shopping'] = $shopcont;
            /* 生成更新语句 */
            $updateSQL = $conn->GetUpdateSQL($rst,$addshop);
            /* 更新数据表 */
        }
    }
}
xmlhttp.send(null);

```

```

        if(false == $conn->execute($updateSQL)){
            $reback = '4';
        }else{
            $reback = '1';
        }
    }
}else{
    /* 如果 shopping 字段为空, 则直接更新 */
    $tmparr = $key.",1";
    $addshop['shopping'] = $tmparr;
    $updateSQL = $conn->GetUpdateSQL($rst,$addshop);
    if(false == $conn->execute($updateSQL)){
        $reback = '4';
    }else{
        $reback = '1';
    }
}
echo $reback;
?>

```

通过分析上述代码可以知道, shopping 字段保存的是购物车中的商品信息, 一条商品信息包括两部分: 商品 ID 和商品数量, 其中商品数量默认为 1。两部分之间使用逗号 “,” 分隔。如果添加多个商品, 则每个商品之间使用 “@” 来分隔。

## 26.12.2 显示购物车

商品添加后, 系统直接跳到购物车页面, 显示添加的商品信息。购物车页面同样分 PHP 代码页和 Smarty 模板页。在 PHP 代码页中, 先读取 tb\_user 数据表中 shopping 字段的内容, 如果字段为空, 则输出“暂无商品”字样; 如果数据库中有数据, 则使用循环, 将商品信息保存到数组中, 再传给模板页。购物车页面的代码如下:

```

<?php
/* 载入数据库连接文件和 Smarty 配置文件 */
include_once 'conn/conn.php';
include_once 'config.php';
/* 根据登录会员的 ID 生成查询语句 */
$sql1 = "select id,shopping from tb_user where id =".$_SESSION['id'];
/* 执行查询语句返回结果集 */
$rst = $conn->execute($sql1);
/* 如果字段为空, 直接输出无商品信息, 并结束运行 */
if($rst->fields['shopping'] == ""){
    echo "<p>";
    echo '购物车中暂时没有商品!';
    exit();
}

```

```

/* 如果有结果集，则将结果集另存为数组 */
$tmparr = $rst->GetAssoc();
$commarr = array();
/* 输出 shopping 字段值 */
foreach($tmparr as $value){
    /* 使用“@”进行分隔，得到数组*/
    $tmpnum = explode('@', $value);
    /* 得到购买商品种类 */
    $shopnum = count($tmpnum);
    /* 商品总价值 */
    $sum = 0;
    /* 数组循环 */
    foreach($tmpnum as $key => $vl){
        /* 使用逗号“,”拆分单一的商品信息 */
        $s_commo = explode(',', $vl);
        /* 根据保存的商品信息查找商品 */
        $sql2 = "select id, name, m_price, fold, v_price from tb_commo";
        $commsql = $sql2." where id = ".$s_commo[0];
        /* 返回结果集 */
        $commrst = $conn->execute($commsql);
        /* 将结果集另存数组 */
        $arr = $commrst->GetArray();
        /* 将需要显示的信息处理后保存到数组中 */
        $arr[0]['num'] = $s_commo[1];
        $arr[0]['total'] = $s_commo[1]*$arr[0]['v_price'];
        $sum += $arr[0]['total'];
        $commarr[$key] = $arr[0];
    }
}
/* 将得到的两个数组和总消费金额传递给模板 */
$smarty->assign('shoparr', $shopnum);
$smarty->assign('commarr', $commarr);
$smarty->assign('sum', $sum);
$smarty->assign('title', '我的购物车');
/* 显示购物车模板 */
$smarty->display('myshopcar.tpl');
?>

```

商品的模板页不仅要负责将传入的商品信息数组显示出来，而且还要提供可以对商品进行修改、删除等操作的接口。模板页的代码如下：

```

<!-- 载入使用到的js脚本文件 -->
<script language="javascript" src="js/creatxmlhttp.js"></script>
<script language="javascript" src="js/shopcar.js"></script>
<table>
<!-- 购物车表单 -->
<form id="myshopcar" name="myshopcar" method="post" action="#">
    <tr><td>我的购物车</td></tr>
    <tr><td>&nbsp;</td><td>商品名称</td><td>购买数量</td><td>市场价格</td>
        <td>会员价格</td><td>折扣率</td><td>合计</td></tr>

```

```

<!-- 使用 foreach 表单循环输出商品种类 -->
{foreach key=key item=item from=$commarr}
<!-- 删除复选框，名称用数组表示，每个复选框的值为商品的 ID 号 -->
<tr><td><input id="chk" name="chk[]" type="checkbox" value="{{$item.id}}"></td>
<!-- 显示该商品的名称 -->
<td><div id="c_name{$key}"> &ampnbsp{{$item.name}}</div></td>
<td>
<!-- 商品数量文本框 每个复选框的名称为“cnum”+key 值 -->
<input id="cnum{$key}" name="cnum{$key}" type="text" value="{{$item.num}}" onkeyup="
  cvp({$key}, {$item.v_price}, {$shoparr})"></td>
<!-- 商品的其他信息 -->
<td><div id="m_price{$key}">&ampnbsp{{$item.m_price}}</div></td>
<td><div id="v_price{$key}">&ampnbsp{{$item.v_price}}</div></td>
<td><div id="fold{$key}">&ampnbsp{{$item.fold}}</div></td>
<td><div id="total{$key}">&ampnbsp{{$item.total}}</div></td>
</tr>
{/foreach}
<tr>
<td colspan="3">
<!-- 全选、反选和删除操作 -->
<a href="#" onclick="return alldel(myshopcar)">全选</a>
<a href="#" onclick="return overdel(myshopcar);">反选</a>
<input type="button" value="删除选择" onClick='return del(myshopcar);'></td>
<td>
<!-- 继续购物 -->
<input type="button" value="继续购物" onclick="return conshop(myshopcar)" /></td>
<td>
<!-- 将当前用户名保存到隐藏域中 -->
<input id="uid" name="uid" type="hidden" value="{$smarty.session.member}" >
<!-- 结算，去收银台 -->
<input type="button" class="btn" value="去收银台" onclick="return formset(form)" /></td>
<td colspan="2"><div id='sum'>共计: {$sum}&ampnbsp元</div></td>
</tr>
</form>
</table>

```

### 26.12.3 更改商品数量

对于新添加的商品，默认的购买数量为 1，在购物车页面可以对商品的数量进行修改。当商品数量发生变化时，商品的“合计”金额和商品总金额会自动发生改变。该功能是通过触发 text 文本域的 onkeyup 事件调用 cvp() 函数实现的。cvp() 函数有 3 个参数：商品 ID、商品单价和商品类别。

首先通过商品 ID 得到要修改商品的相关表单和标签属性，然后通过商品单价和输入的商品数量计算该商品的合计金额，接着使用 for 循环得到其他商品的合计金额，最后将所有的合计金额累加，并输出到购物车页面。cvp() 函数的代码如下：

```

/* 自动刷新总金额
 * key: 商品 ID

```

```

/* vpr: 商品单价
 * shoparr: 商品种类数
 */
function cvp(key,vpr,shoparr){
    var n_pre = 'total';
    var num = 'cnum'+key.toString();
    var total = n_pre+key.toString();
    var t_number = document.getElementById(num).value;
    var ttl = t_number * vpr;
    document.getElementById(total).innerHTML = ttl;
    var sm = 0;
    for(var i = 0; i < shoparr; i++){
        var aaa = document.getElementById(n_pre+i.toString()).innerText;
        sm += parseInt(aaa);
    }
    document.getElementById('sum').innerHTML = '共计: '+sm+' 元';
}

```

这里所更改的商品数量并没有被保存到数据库中，如果希望保存，则单击“继续购物”按钮，即可将商品数量更新到数据库中。该功能将在 26.12.5 节中实现。

#### 26.12.4 删除商品

当对添加的商品不满意时，可以对商品进行删除操作。操作流程为：首先选中要删除的商品前面的复选框，如果全部删除，则可以单击“全选”按钮，或“反选”按钮；然后单击“删除选择”按钮；在弹出的警告框中单击“确定”按钮，商品将被全部删除。删除商品的操作流程如图 26.24 所示。



图 26.24 删除商品流程

所有的删除操作都是通过 js 脚本文件 shopcar.js 来实现的，相关的函数包括 alldel() 函数、overdel() 函数和 del() 函数。下面分别介绍这 3 个函数的实现过程。

alldel() 函数和 overdel() 函数实现的原理比较简单，通过触发 onclick 事件来改变复选框的选中状态。实例代码如下：

```

/* 全部选择 */
function alldel(form){
    var leng = form.chk.length;
    if(leng==undefined){
        if(!form.chk.checked)
            form.chk[0].checked = true;
        else
            form.chk[0].checked = false;
    }
    else{
        for(var i=0;i<leng;i++)
            form.chk[i].checked = !form.chk[i].checked;
    }
}

```

//获取复选框数量  
//如果等于 undefined，说明只有一个复选框

```

        form.chk.checked=true;           //将复选框置于选中状态
    }else{
        for( var i = 0; i < leng; i++){
            if(!form.chk[i].checked)
                form.chk[i].checked = true;
        }
    }
    return false;
}

/* 反选，就是将选中的复选框取消选中，而未被选中的复选框则被选中 */
function overdel(form){
    var leng = form.chk.length;          //获取复选框个数
    if(leng==undefined){               //如果 leng 为 undefined，说明只有一个复选框
        if(!form.chk.checked)
            form.chk.checked=true;
        else
            form.chk.checked=false;
    }else{                            //否则，说明有多个复选框
        for( var i = 0; i < leng; i++){
            if(!form.chk[i].checked)      //根据复选框 checked 属性的情况，进行反向选择
                form.chk[i].checked = true;
            else
                form.chk[i].checked = false;
        }
    }
    return false;
}

```

使用 alldel 或 overdel 选中复选框后，即可调用 alldel() 函数来实现删除功能。alldel() 函数首先使用 for 循环，将被选中的复选框的 value 值取出并存成数组；然后根据数组生成 url，并使用 xmlhttp 对象调用这个 url，当处理完毕后，根据返回值，或弹出提示，或刷新本页。该函数代码如下：

```

/* 删除记录 */
function del(form){
    if(!window.confirm('是否要删除数据??')){
    }else{
        var leng = form.chk.length;          //返回复选框的长度
        if(leng==undefined){
            if(!form.chk.checked){
                alert('请选择要删除数据!');
            }else{
                rd = form.chk.value;           //如果只有一个复选框，且处于选中状态
                var url = 'delshop.php?rd=' + rd; //将复选框的 value 值直接赋给变量 rd
                xmlhttp.open("GET",url,true);   //根据 rd 生成 url
                xmlhttp.onreadystatechange = delnow; //调用 xmlhttp 对象
                xmlhttp.send(null);           //调用 delnow 函数
            }
        }else{                                //如果复选框为多个
            var rd=new Array();                //声明数组

```

```

var j = 0;
for( var i = 0; i < leng; i++){
    if(form.chk[i].checked){           //循环检查复选框状态
        rd[j++] = form.chk[i].value;   //将被选中的复选框的 value 值存到 rd 内
    }
}
if(rd == ""){
    alert('请选择要删除数据!');
} else{
    var url = "delshop.php?rd=" + rd;      //解释同上
    xmlhttp.open("GET", url, true);
    xmlhttp.onreadystatechange = delnow;
    xmlhttp.send(null);
}
}
return false;
}
/* 显示状态 */
function delnow(){
    if(xmlhttp.readyState == 4){
        if(xmlhttp.status == 200){
            var msg = xmlhttp.responseText;          //获取 xmlhttp 对象返回的文本值
            if(msg != '1'){
                alert('删除成功');                  //如果为 1，说明删除成功
            } else{
                alert('删除成功');                  //否则，说明删除成功，刷新购物车
                location = ('?page=shopcar');
            }
        }
    }
}
}

```

## 26.12.5 保存购物车

当用户希望保存商品数量时，可以单击“继续购物”按钮。这时，系统将触发 onclick 事件调用 conshop() 函数来保存数据。该函数有一个参数，就是当前表单的名称。在 conshop() 函数内，根据复选框和商品数量文本域，生成两个数组 fst 和 snd，分别保存商品 ID 和商品数量。这里要注意：两个数组的值是要相互对应的，如商品 1 的 ID 保存到 fst[1] 中，那么商品 1 的数量就要保存到 snd[1] 中。然后根据这两个数组生成一个 url，使用 XMLHttpRequest 对象调用 url，最后根据回传信息作出相应的判断。conshop() 函数的代码如下：

```

/* 更改商品数量 */
function conshop(form){
    var n_pre = 'cnum';                      //商品数量文本域前缀
    var lang = form.chk.length;               //获取复选框的数量
    /* 如果只有一个复选框，那么将商品 ID 和商品数量直接保存到变量中 */

```

```

if(lang == undefined){
    var fst = form.chk.value;
    var snd = form.cnum0.value;
/* 否则，将商品 ID 和对应的商品数量保存到两个相应的数组中 */
} else{
    var fst= new Array(); //商品 ID 数组
    var snd = new Array(); //商品数量数组
/* 循环获取复选框的 value 值和商品数量文本框的 value 值 */
    for(var i = 0; i < lang; i++){
        var nm = n_pre+i.toString();
        var stmp = document.getElementById(nm).value;
/* 对商品数量文本框进行判断，不允许为空，而且不允许为非数字输入 */
        if(stmp == "" || isNaN(stmp)){
            alert('不允许为空、必须为数字');
            document.getElementById(nm).select();
            return false;
        }
        snd[i] = stmp;
        var ftmp = form.chk[i].value;
        fst[i] = ftmp;
    }
}
var url = 'changecar.php?fst='+fst+'&snd='+snd; //生成 url
xmlhttp.open("GET",url,true); //调用 xmlhttp 对象
xmlhttp.onreadystatechange = updatecar; //调用 updatecar 函数
xmlhttp.send(null);
}
/* 对 xmlhttp 对象的返回值进行处理 */
function updatecar(){
    if(xmlhttp.readyState == 4){
        var msg = xmlhttp.responseText;
        if(msg == '1'){
            location='index.php'; //如果操作成功，返回首页
        }else{
            alert('操作失败'); //否则提示错误
        }
    }
}
}

```

函数中调用的 changecar.php 页为数据处理页，该页将商品 ID 和商品数量进行重新排列，并保存到 shopping 字段内。该页面代码如下：

```

<?php
session_start(); //开启 Session 支持
include_once 'conn/conn.php'; //载入数据库连接文件
$sql = 'select id,shopping from tb_user';
$sql .= ' where id = '.(int)$_SESSION['id'];
$fst = $_GET['fst']; //获取商品 ID
$snd = $_GET['snd']; //获取商品数量
$reback = '0'; //设置返回值，默认为 0

```

```

$rst = $conn->execute($sql); //执行查询语句，返回查询结果集
$changecar = array(); //创建更新数组
if($fst != " and $snd != ""){
    /* 使用逗号 “,” 将商品 ID 和商品数量分隔，并分别保存到两个临时数组中 */
    $farr = explode(',',$fst);
    $sarr = explode(',',$snd);
    /* 临时数组 */
    $upcar = array();
    /* 使用 for 循环，将商品 ID 和商品数量一一对应，重新组合 */
    for($i = 0; $i < count($farr); $i++){
        $upcar[$i] = $farr[$i].','.$sarr[$i];
    }
    /* 如果数组大于 1，说明有多个商品，使用 “@” 将不同商品分开 */
    if(count($farr) > 1){
        $changecar['shopping'] = implode('@',$upcar);
    }else{
        /* 如果只有 1 个商品，则直接保存到更新数组中 */
        $changecar['shopping'] = $upcar[0];
    }
    /* 生成更新语句 */
    $UpdateSql = $conn->GetUpdateSQL($rst,$changecar,true);
    /* 更新数据库 */
    if(false == $conn->execute($UpdateSql)){
        $reback = '2';
    }else{
        $reback = '1';
    }
}
echo $reback;
?>

```

## 26.13 收银台模块设计

### ■ 视频讲解：光盘\TM\Video\26\收银台模块设计.exe

当用户停止浏览商品准备结账时，可以通过单击购物车页面中的“去收银台”按钮来实现。该按钮将触发 onclick 事件调用 formset() 函数来显示订单，当用户提交订单后，系统将订单保存到数据表 tb\_form 中，同时清空购物车，并显示订单信息，提醒用户记录订单号。当货款发出后，还可以对订单进行查询。收银台页面的运行结果如图 26.25 所示。

本节所涉及的页面有显示订单（formset() 函数）、填写订单（settle.php、settle.tpl）、提交订单（settle\_chk.php）、反馈订单（forminfo.php、forminfo.tpl）和查询订单 5 部分。首先来介绍 formset() 函数。

The screenshot shows a Microsoft Internet Explorer browser window with the URL http://127.0.0.1. The title bar says '收银台 - Microsoft Internet Explorer'. The main content area has a form titled '订单信息' (Order Information). It contains fields for '收货人' (Recipient), '联系电话' (Phone), '邮编' (Postcode), '地址' (Address), '送货方式' (Delivery Method), and '付款方式' (Payment Method). The '送货方式' dropdown is set to '快递' (Express). The '付款方式' dropdown is set to '银行汇款' (Bank Transfer). At the bottom right of the form is a '提交订单' (Submit Order) button.

图 26.25 收银台页面运行结果

### 26.13.1 显示订单

formset()函数的作用是将商品信息整理后，通过 open 方法打开 settle.php 页来显示订单，并将整理后的商品信息一并传过去。formset()函数的代码如下：

```

function formset(form){
    var uid = form.uid.value;                                //获取订单提交人的名称
    var n_pre = 'cnum';                                     //商品数量文本前缀
    var lang = form.chk.length;                             //复选框个数
    /* 如果复选框只有 1 个 */
    if(lang == undefined){
        var fst = form.chk.value;                           //商品 ID
        var snd = form.cnum0.value;                         //商品数量
    /* 当有多个复选框时 */
    }else{
        var fst= new Array();                            //商品 ID
        var snd = new Array();                           //商品数量
    /* 使用 for 循环，获取商品数量和商品 ID */
        for(var i = 0; i < lang; i++){
            var nm = n_pre+i.toString();
            var stmp = document.getElementById(nm).value;
            if(stmp == "" || isNaN(stmp)){
                alert('不允许为空、必须为数字');
                document.getElementById(nm).select();
                return false;
            }
            snd[i] = stmp;
            var ftmp = form.chk[i].value;
            fst[i] = ftmp;
        }
    }
    /* 使用 open 方法，打开 settle.php 页，并将当前会员名称、商品 ID 和商品数量的值传过去 */
    open('settle.php?uid='+uid+'&fst='+fst+'&snd='+snd,'_blank','width=420 height=220',false);
}

```

说明：因为 open 方法使用了 \_blank 参数来打开一个新的页面，Session 值传不过去。所以这里使用隐藏域来传递会员名称。

### 26.13.2 填写订单

settle.php 直接将接收的值传给 settle.tpl 模板，并载入 settle.tpl 模板。settle.php 页面的代码如下：

```

<?php
    include_once 'config.php';
    $fst = $_GET['fst'];
    //载入 Smarty 配置文件
    //获取 get 属性值

```

```

$snd = $_GET['snd'];
$uid = $_GET['uid'];
$smarty->assign('title','结账');           //将标题、数组等数据传递给模板
$smarty->assign('fst',$fst);
$smarty->assign('snd',$snd);
$smarty->assign('uid',$uid);
$smarty->display('settle.tpl');           //显示 settle.tpl 模板文件
?>

```

settle.tpl 模板显示一个表单，这个表单的内容需要用户来填写，包括收货人、联系电话等信息，而从 PHP 页传过来的几个变量则被保存到隐藏域以传递到处理页。settle.tpl 模板的表单元素及属性值如表 26.2 所示。

表 26.2 settle.tpl 模板的表单元素及属性值

元素名称	元素属性	元素说明
<form>	id = "buyform" name = "buyform" method = "post" action = "settle_chk.php"	表单名为 buyform，提交页为 settle_chk.php
text	id="taker" name="taker" type="text"	收货人文本域
text	id="code" name="code" type="text"	邮编文本域
text	id="tel" name="tel" type="text"	电话文本域
text	id="address" name="address" type="text"	地址文本域
<select>	<select id="del" name="del"> <option value="平邮">平邮</option> <option value="快递">快递</option> <option value="送货上门">送货上门</option> </select>	送货方式
<select>	<select id="pay" name="pay"> <option value="银行转账">银行转账</option> <option value="邮局汇款">邮局汇款</option> <option value="支付宝">支付宝</option> </select>	付款方式
<hidden>	id="fst" name="fst" type="hidden" value="{{\$fst}}"	保存商品 ID
<hidden>	id="snd" name="snd" type="hidden" value="{{\$snd}}"	保存商品数量
<hidden>	id="uid" name="uid" type="hidden" value="{{\$uid}}"	保存当前用户
<submit>	id="enter" name="enter" type="submit" value="提交订单"	提交订单按钮

### 26.13.3 处理订单

用户添加表单信息后，单击“提交”按钮，将会调用处理页 settle\_chk.php 来处理表单数据。处理页接收到数据后，根据用户提交的商品信息，重新查找数据表 tb\_commo，并从数据表中提取商品的一些信息，保存到数组中，然后将数组作为一条记录添加到表 tb\_form 内。数据添加成功的同时，处理页会根据 UID 找到该用户，将 shopping 字段清空，最后调用 forminfo.php 页来显示新添加的订单信息。

settle\_chk.php 页的代码如下：

```

<?php
/* 获取表单值，并存储到添加数组中 */
include_once 'conn/conn.php';
$sql = "select * from tb_form where id = -1";
$rst = $conn->execute($sql);
$addform = array();
$addform['vendee'] = $_POST['uid']; //提交订单用户
$addform['commo_id'] = $_POST['fst']; //购买商品 ID
$addform['commo_num'] = $_POST['snd']; //购买商品数量
/*
根据商品 ID 和商品数量，获取所需数据 */
$addform['formid'] = time() //订单号
$tmpid = explode(',',$addform['commo_id']); //将商品 ID 存为数组
$tmpnm = explode(',',$addform['commo_num']); //将商品数量存为数组
$number = count($tmpid); //获取商品数组长度
/*
当购买商品种类大于 1 时 */
if($number > 1){
    $tmpna = array(); //商品名称数组
    $tmpvp = array(); //商品单价数组
    $tmpfd = array(); //商品折扣数组
    $tmptt = 0; //商品总额数组
    /*
    使用 for 循环获取各种商品的信息 */
    for($i = 0; $i < $number; $i++){
        /* 生成查询语句，执行 SQL，最后返回查询结果集 */
        $tmpsql = "select name,v_price,fold from tb_commo where id = ".$tmpid[$i];
        $tmprst = $conn->execute($tmpsql);
        /* 将每种商品的信息，放到对应的数组中 */
        $tmpna[$i] = $tmprst->fields['name'];
        $tmpvp[$i] = $tmprst->fields['v_price'];
        $tmpfd[$i] = $tmprst->fields['fold'];
        $tmptt += $tmprst->fields['v_price'] * $tmpnm[$i];
    }
    /*
    将获取的信息数组分别生成长字符串，数组值之间使用逗号“,” 分隔 */
    $addform['commo_name'] = implode(',',$tmpna); //商品名称
    $addform['agoprice'] = implode(',',$tmpvp); //商品价格
    $addform['fold'] = implode(',',$tmpfd); //商品折扣
    $addform['total'] = $tmptt; //商品总额
    /*
    当购买一种商品时 */
} else if($number == 1){
    $tmpsql = "select name,v_price,fold from tb_commo where id = ".$tmpid[0];
    $tmprst = $conn->execute($tmpsql);
    $addform['commo_name'] = $tmprst->fields['name'];
    $addform['agoprice'] = $tmprst->fields['v_price'];
    $addform['fold'] = $tmprst->fields['fold'];
    $addform['total'] = $tmprst->fields['v_price'] * $tmpnm[0];
    /*
    没有商品时 */
} else{
    echo 'error';
}

```

```

    exit();
}

/* 将表单的其他信息存入添加数组 */
$addform['taker'] = $_POST['taker']; //收货人
$addform['code'] = $_POST['code']; //邮编
$addform['tel'] = $_POST['tel']; //联系电话
$addform['address'] = $_POST['address']; //地址
$addform['del_method'] = $_POST['del']; //送货方式
$addform['pay_method'] = $_POST['pay']; //付款方式
$addform['state'] = '0'; //订单状态, 0 为未处理
$InsertSQL = $conn->GetInsertSQL($rst,$addform); //生成 SQL 添加语句
if(false == $conn->execute($InsertSQL)){
    /* 如果购买失败, 返回上层 */
    echo "<script>alert('购买失败');history.back;</script>";
} else{
    /* 如果成功, 删除购物车信息, 并显示订单信息 */
    $updsql = "select * from tb_user where name = '".$_POST['uid']."'"; //";
    $updrst = $conn->execute($updsql);
    $arr = array();
    $arr['consume'] = $addform['total'];
    $arr['shopping'] = "";
    $UpdateSQL = $conn->GetUpdateSQL($updrst,$arr);
    $conn->execute($UpdateSQL);
    echo "<script>top.opener.location.reload();</script>";
    echo "<script>open('forminfo.php?fid=$fid','_blank','width=600 height=450',false);</script>";
    echo "<script>window.close();</script>";
}
?>

```

#### 26.13.4 反馈订单

订单提交成功后, forminfo.php 页会将新添加的订单反馈给用户, 并提醒用户保存订单号, 以便根据订单号来付款、查询及处理问题。反馈订单页面的运行结果如图 26.26 所示。

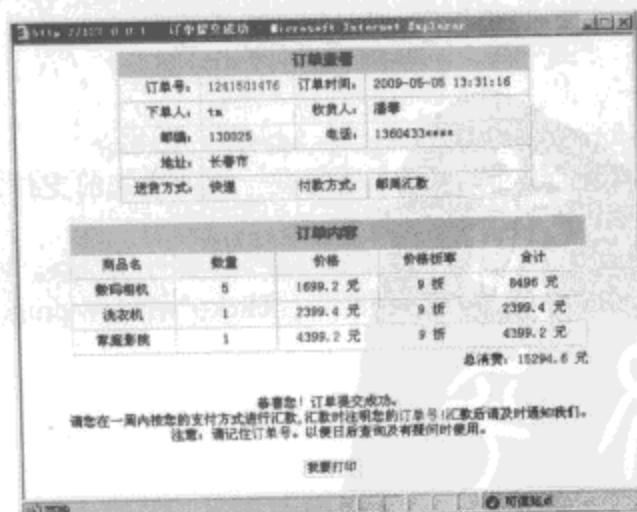


图 26.26 反馈订单页面的运行结果

forminfo.php 页的代码如下：

```

<?php
    include_once 'conn/conn.php';
    include_once 'config.php';
    $id = $_GET['fid'];
    $sql = "select * from tb_form where id = ".$id;
    $rst = $conn->execute($sql);
    $formarr=$rst->GetArray();
    $commname = explode(',',$formarr[0]['commo_name']);
    $commnumber = explode(',',$formarr[0]['commo_num']);
    $commagoprice = explode(',',$formarr[0]['agoprice']);
    $commfold = explode(',',$formarr[0]['fold']);
    /* 将各个数组传给模板，最后显示模板 */
    $smarty->assign('formarr',$formarr[0]);
    $smarty->assign('commname',$commname);
    $smarty->assign('commnumber',$commnumber);
    $smarty->assign('commagoprice',$commagoprice);
    $smarty->assign('commfold',$commfold);
    $smarty->assign('title','订单提交成功');
    $smarty->display('forminfo.tpl');
?>

```

模板页 forminfo.tpl 包含了两个表格，第一个表格显示提交订单用户所填信息；第二个表格为购买的商品信息。最后还有一行注意文字，提醒用户不要忘记订单号。如果有条件，还可以将订单打印出来。模板页商品信息代码如下：

```

<!-- 商品信息 -->





```

### 26.13.5 查询订单

订单提交后，可以通过导航栏中的“订单查询”超链接来查看订单情况。查询关键字可以是会员

名称或订单号，只要输入一项就可以。单击“查询”按钮，将会在页面下方显示查询结果。查询订单页面运行结果如图 26.27 所示。

查询结果						
订单号	订货会员	收货人	订单金额	付款方式	收款方式	订单状态
1209117958	ta	sp	6666	银行转账	平邮	已付款
1209119797	ta	spcn	73326	银行转账	平邮	未作处理
1241501476	ta	潘琴	15294.6	邮局汇款	快递	未作处理

图 26.27 查询订单页面运行结果

本模块采用 Ajax 技术，对查询结果在本页进行直接输出。共使用 4 个文件：queryform.php、queryform.tpl、queryform.js 和 query.php。  
queryform.php 页用于载入 Smarty 配置文件，并通过 display 方法指定 queryform.tpl 模板页。  
queryform.tpl 包含了一个查询表单和一个 div 标签，其元素属性和名称如表 26.3 所示。

表 26.3 查询表单元素和 div 标签属性及说明

元素名称	元素属性	元素说明
<form>	id='queryform' name='queryform' method='post' action='#'	查询表单，名称为 queryform
text	input id='name' name='name' type='text' class='txt'	会员查询文本框
text	id='formid' name='formid' type='text' class='txt'	订单号查询文本框
button	id='enter' name='enter' type='button' value='查询' class='btn' onClick='return queryrst(queryform)'	查询按钮，调用 queryrst 函数
<div>	id=exam	用于显示查询结果

通过表 26.3 可以看到，当单击“查询”按钮时，将触发 onclick 事件，调用 queryrst() 函数。queryrst() 函数在脚本文件 js/queryform.js 内。该文件代码如下：

```

function queryrst(form) {
    /* 获取会员名称 */
    var name = form.name.value;
    /* 获取订单号 */
    var formid = form.formid.value;
    /* 如果两者都为空，则弹出警告框 */
    if(name == "" && formid == ""){
        alert('用户或订单号至少有一个不能为空');
        form.name.focus();
        return false;
    }
    /* 将获取的变量传给 query.php 页 */
    var url = "query.php?vendee='"+name+"&formid='"+formid;
    xmlhttp.open("GET",url,true);
    /* 调用 showfm() 函数 */
    xmlhttp.onreadystatechange = showfm;
    xmlhttp.send(null);
}
function showfm(){

```

```

if(xmlhttp.readyState == 4){
    var msg = xmlhttp.responseText;
    if(msg == '0'){
        /* 如果没有查询结果，则清空 exam */
        exam.innerHTML = "";
        alert('没有查询结果');
    }else{
        /* 将结果输出到页面 */
        exam.innerHTML = msg;
    }
}
}

```

query.php 页根据传递的 vendee 和 formid 参数，从 tb\_form 表中查询记录，如果没有记录，则返回 0；如果有记录，则输出完整的表单字串。query.php 页的代码如下：

```

<?php
/* 载入数据库连接文件 */
include_once 'conn/conn.php';
/* 获取 url 参数 */
$vendee = $_GET['vendee'];
$formid = $_GET['formid'];
$reback = '0';
/* 创建一个表格，并将表单存储到临时变量中 */
$tmp = "<table width='600' border='0' align='center' cellpadding='0' cellspacing='0'><tr><td height='25' colspan='7' align='center' valign='middle' class='first'>查询结果 </td></tr><tr><td width='100' height='25' align='center' valign='middle' class='left'>订单号 </td><td width='75' height='25' align='center' valign='middle' class='center'>订货会员 </td><td width='75' height='25' align='center' valign='middle' class='center'>收货人 </td><td width='100' height='25' align='center' valign='middle' class='center'>订单金额 </td><td width='75' height='25' align='center' valign='middle' class='center'>付款方式 </td><td width='75' height='25' align='center' valign='middle' class='center'>收款方式 </td><td width='100' height='25' align='center' valign='middle' class='right'>订单状态 </td></tr>";
/* 根据参数值，生成查询语句 */
$sql = "select id,form:id,vendee,taker,total,pay_method,del_method,state from tb_form where vendee = '".$vendee."' or formid = '".$formid."'";
/* 执行 SQL 语句，返回查询结果集 */
$rst = $conn->execute($sql);
if($rst->RecordCount() == 0){
} else {
    /* 如果有查询记录，则将结果循环输入到表格内 */
    $arr = $rst->getArray();
    foreach($arr as $value){
        $tmp .= "<tr><td height=25 align=center valign=middle class=left>".$value['formid']."</td><td align=center valign=middle class=center>".$value['vendee']."</td><td align=center valign=middle class=center>".$value['taker']."</td><td align=center valign=middle class=center>".$value['total']."</td><td align=center valign=middle class=center>".$value['pay_method']."</td><td align=center valign=middle class=center>".$value['del_method']."</td><td align=center valign=middle class=right>";
    }
    /* 对订单状态进行处理 */
    switch (state){
        case 0:$tmp .= '未作处理';break;
        case 1:$tmp .= '已付款';break;
    }
}

```

```

        case 2:$tmp .= '已发货';break;
        case 3:$tmp .= '已收货';break;
    }
    $tmp .= "</td></tr>";
}
$tmp .= "</table>";
/* 将临时变量赋给 $reback 并输出 */
$reback = $tmp;
}
echo $reback;
?>

```

有关电子商务平台前台功能模块的讲解到此结束，接下来讲解后台管理系统中首页、类别管理和订单管理模块的设计。

## 26.14 后台首页设计

### ■ 视频讲解：光盘\TM\Video\26\后台首页设计.exe

后台管理系统对网站的商品、商品类别、会员及公告等信息进行统一管理。本系统的后台主要包括如下功能模块。

- 类别管理模块：**主要包括对商品类别的添加、修改及删除操作。
- 商品管理模块：**主要包括对商品的添加、修改、删除及订单处理。
- 用户管理模块：**主要包括管理员管理和会员管理。其中，管理员管理是实现对管理员的添加、删除和修改功能；会员管理则包括删除和冻结功能。
- 公告管理模块：**主要包括公告的添加及删除操作。
- 链接管理模块：**主要包括添加、修改和删除友情链接。

后台首页的运行结果如图 26.28 所示。

The screenshot shows the admin homepage for '明日购物商城' (www.mingri.com). On the left, there's a sidebar with links like '商品管理', '添加商品', '查看商品', '查看订单', '用户管理', '公告管理', '添加公告', '查看公告', and '友情链接'. The main content area displays a table of orders with columns: 编号 (Order ID), 订单号 (Order No.), 收货人 (Recipient), 付费方式 (Payment Method), 银行转帐 (Bank Transfer), 平邮 (Plain Mail), 已收款 (Received), 收货 (Delivery), and 备注 (Remarks). There are 12 rows of data, each with a checkbox in the first column. At the bottom of the table, there are buttons for '全选' (Select All), '反选' (Invert Selection), and '刷新' (Refresh).

	订单号	收货人	付费方式	银行转帐	平邮	已收款	收货	备注	
<input checked="" type="checkbox"/>	1209117968	ta	spm	6666	银行转帐	平邮	已收款	直邮	速递员送
<input checked="" type="checkbox"/>	1209118797	ta	spm	73326	银行转帐	平邮	未处理	收货	速递员送
<input checked="" type="checkbox"/>	1209119925	ta	spm	73326	银行转帐	平邮	未处理	寄回	速递员送
<input checked="" type="checkbox"/>	1209120302	ta	spm	73326	银行转帐	平邮	未处理	收货	速递员送
<input checked="" type="checkbox"/>	1209120844	ta	sp	21.9	银行转帐	平邮	未处理	寄回	速递员送
<input checked="" type="checkbox"/>	1209120610	ta	spm	6666	银行转帐	送货上门	未处理	收货	速递员送
<input checked="" type="checkbox"/>	1209120377	ta	spm	2080	银行转帐	平邮	未处理	收货	速递员送
<input checked="" type="checkbox"/>	1241501476	ta	spm	15294.6	银联IC卡	快递	未处理	收货	速递员送

图 26.28 后台首页运行结果

### 26.14.1 后台首页布局

后台首页和前台首页不同，使用的是框架布局。main.php 页的框架代码如下：

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<link rel="stylesheet" href="css/style.css" />
<title>明日购物商城后台管理系统</title>
</head>
<!-- 使用 frame 框架布局 -->
<frameset rows="113,* ,100" cols="1004" frameborder="no" border="1" framespacing="0">
    <!-- 头部框架，命名为 topFrame，载入文件为 top.php -->
    <frame src="top.php" name="topFrame" scrolling="No" noresize="noresize" id="topFrame" title="topFrame" />
    <frameset rows="*" cols="310,* " framespacing="0" frameborder="no" border="1">
        <!-- 左侧框架，名字为 leftFrame，载入文件为 left.php -->
        <frame src="left.php" name="leftFrame" frameborder="0" scrolling="auto" noresize="noresize" id="leftFrame" title="leftFrame" />
        <!-- 中部框架，名字为 mainFrame，载入文件是 default.php -->
        <frame src="default.php" name="mainFrame" id="mainFrame" title="mainFrame" />
    </frameset>
    <!-- 底部框架，名字为 bottomFrame，载入文件是 bottom.php -->
    <frame src="bottom.php" name="bottomFrame" scrolling="No" noresize="noresize" id="bottomFrame" title="bottomFrame" />
</frameset>
<noframes><body></body></noframes>
</html>

```

在框架页中包含了 4 个文件：top.php 为后台页面的标题；left.php 为导航栏；default.php 为后台的功能输出页面；bottom.php 为网站的尾文件。

## 26.14.2 DIV+JavaScript+CSS 实现树状菜单

在 left.php 页面中设计了一个树状菜单，是应用 DIV+JavaScript+CSS 来实现的。首先来看 div 标签。

```

<!-- 载入 css 样式和 javascript 脚本 -->
<link href="css/left.css" rel="stylesheet" type="text/css" />
<script language="javascript" src="js/left.js"></script>
<!-- 类别管理菜单，注意加粗的地方 -->
<div id="type" align="center" onclick="javascript:change(one,type);">类别管理</div>
<!-- 子菜单 -->
<div id="one" style="display: ">
    <div id="addtype" align="center"><a href="addtype.php" target="mainFrame" id="menu">添加类别</a></div>
    <div id="showtype" align="center"><a href="showtype.php" target="mainFrame" id="menu">查看类别</a></div>
</div>
<div id="hidediv" align="center"></div>
<!-- 商品管理菜单 -->
<div id="commo" align="center" onclick="javascript:change(two,type);">类别管理</div>
<div id="two" style="display:none">
    <!-- 商品管理子菜单 -->
    ...
</div>

```

说明：除了加粗的 ID 名称和 JS 事件不同外，其他菜单的结构完全相同，只需修改超链接即可。

注意：除第一个类别菜单的子菜单 display 样式为空外，其他几个子菜单的 div 样式都为 display=none;。

这时，该页面在 Dreamweaver 中的样式如图 26.29 所示。



图 26.29 div 树状菜单

因为其他子菜单的样式为“display=none”，所以只有“类别管理”子菜单是可见的。下面为它添加 JavaScript 事件。left.js 脚本文件代码如下：

```
/* 函数功能：改变子菜单样式，更换当前菜单的背景图片
 * nu：为子菜单 div 标签的 ID 号
 * lx：为调用 change()函数所在的标签 ID
 */
function change(nu, lx){
    /* 如果当前子菜单的样式为 none，则显示子菜单 */
    if(nu.style.display == "none"){
        nu.style.display = "";
        lx.style.background="url(images/main_openroot.gif)";
    }else{ /* 否则，隐藏当前菜单的子菜单 */
        nu.style.display = "none";
        lx.style.background="url(images/main_closeroot.gif)";
    }
}
```

最后，在 left.css 中设置 div 的长、宽等一些默认参数，一个简单而实用的树状菜单就完成了。在 26.15 节中将介绍后台管理系统中的类别管理模块。

## 26.15 类别管理模块设计

视频讲解：光盘\TM\Video\26\类别管理模块设计.exe

类别管理模块的功能是方便管理员添加、修改及删除商品类别。本系统只提供两级类别的管理和维护。类别管理模块的实现难点在于：如何动态地存取类别记录，使一级类别和二级类别自动关联。首先来看添加类别的实现。

### 26.15.1 添加类别

添加类别的操作流程是：当用户选择“添加类别”子菜单时，将在右侧主框架区显示添加商品类别表单，用户可以在“类别名称”文本框中输入类别名，还可以在“类别等级”下拉列表框中选择添

加商品的类别。当选择“一级类别”时，下面的“父类名称”则不可选。添加类别页面的运行结果如图 26.30 所示。

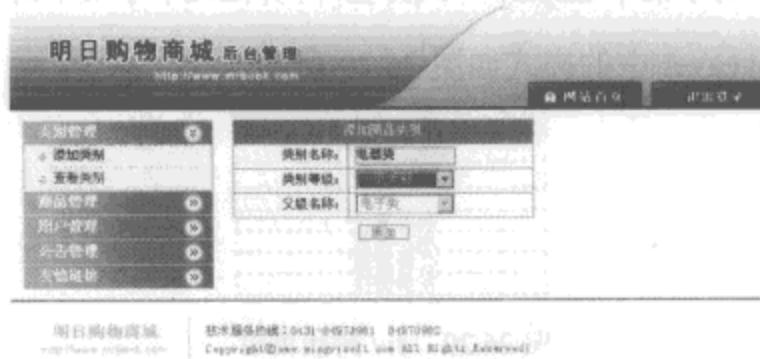


图 26.30 添加类别页面的运行结果

当单击“添加类别”超链接时，将调用 addtype.php 页。该页从数据库中读取所有 supid=0（也就是一级类别）的记录，并将结果集直接输出为下拉菜单，传递给模板页。addtype.php 页的代码如下：

```
<?php
    /* 载入数据库连接文件和 Smarty 模板配置文件 */
    include_once 'config.php';
    include_once 'conn/conn.php';
    /* 生成查询语句，查询所有 supid=0 的一级类别 */
    $sql = "select name,id from tb_class where supid = 0";
    /* 执行 SQL 语句，返回结果集 */
    $rst = $conn->execute($sql);
    /* 将结果集直接输出为下拉菜单，并传递给模板 */
    $smarty->assign('op',$rst->GetMenu2("supid","","$blank = false","","class='txt'"));
    $smarty->display('addtype.tpl');
?>
```

模板页 addtype.tpl 接受 op 下拉菜单后，将其显示在“添加商品类别”表单中。addtype.tpl 模板代码如下：

```
<!-- 载入 js 脚本文件 -->
<script language="javascript" src="js/creat xmlhttp.js"></script>
<script language="javascript" src="js/chktype.js"></script>
<table width="300" border="0" align="center" cellpadding="0" cellspacing="0">
<form id="addtype" name="addtype" method="post" action="#">
    <tr><td colspan="2">添加商品类别</td></tr>
    <tr><td>类别名称: </td><td><input name="names" type="text" id="names" class="txt"></td></tr>
    <tr><td>类别等级: </td><td>
        <!-- 类别等级下拉菜单，会触发 onchange 事件，调用 changetype() 函数 -->
        <select name="grade" OnChange="changetype(addtype)">
            <option value="1">一级类别</option>
            <option value="2" selected>二级类别</option>
        </select>
    </td></tr>
    <!-- 显示 op 变量 -->
    <tr><td>父级名称: </td><td>{$op}</td></tr>
    <tr><td colspan="2"><input type="button" value="添加" class="btn" onClick="chktype(addtype)"></td>
```

```

</tr>
</form>
</table>

```

该模板通过“类别等级”下拉菜单的 `changetype()` 函数来控制 `op` 菜单的状态。如果添加的是一级类别，则 `op` 菜单不可用；如果添加的是二级菜单，则 `op` 菜单可用。当用户输入添加类别后，单击“添加”按钮时，会调用 `chktype()` 函数。包含这两个函数的 `chktype.js` 脚本文件的代码如下：

```

/* 控制 op 菜单的状态 */
function changetype(form){
    if(form.grade.value=="2"){
        form.supid.disabled=false;
    }else{
        form.supid.disabled=true;
    }
}
/* 处理“类别名称”文本框 */
function chktype(form){
    var name = form.names.value;
    var supid = form.supid.value;
    if(form.names.value == ""){
        alert('请填写类别名称');
        form.names.focus();
        return false;
    }
    /* 根据不同的类别等级，生成不同的 url */
    if(form.grade.value == "1"){
        var url = "chktype.php?name="+name;
    }else{
        var url = "chktype.php?name="+name+"&supid="+supid;
    }
    /* 使用 xmlhttp 对象打开该 url，并调用 check 函数处理返回值 */
    xmlhttp.open("GET",url,true);
    xmlhttp.onreadystatechange = check;
    xmlhttp.send(null);
}
/* 根据返回值作出相应的处理 */
function check(){
    if(xmlhttp.readyState == 4){
        if(xmlhttp.status == 200){
            var msg = xmlhttp.responseText;
            if(msg == "1"){
                alert('类名重复');
            }else if(msg == "2"){
                alert('添加失败!');
            }else{
                alert('添加成功');
                location='showtype.php';
            }
        }
    }
}

```

在 `chktype.php` 处理页实现对类别的添加，并返回不同的参数以便 `chktype()` 函数进行处理。

chktype.php 页的代码如下：

```

<?php
    include_once 'conn/conn.php'; //载入数据库连接文件
    $name = $_GET['name'];
    ($_GET['supid'] == "")?($supid = 0):($supid = $_GET['supid']); //获取类别名称
    $reback = ""; //获取类别等级
    $sql = "select * from tb_class where name = '$name'"; //设置返回变量
    $rst = $conn->execute($sql); //声明 SQL 语句
    if($rst->RecordCount() == 1){ //返回查询结果集
        $reback = '1'; //如果有查询结果，说明类别名被占用
    }else{ //否则执行添加操作
        $ine = array();
        $ine["name"] = $name;
        $ine["supid"] = $supid;
        $intsql = $conn->GetInsertSQL($rst,$ine); //生成添加语句
        if($conn->execute($intsql) == false){
            $reback = '2';
        }else{
            $reback = '3'; //如果成功，返回 3
        }
    }
    echo $reback;
?>

```

## 26.15.2 查看类别

类别添加成功后，可以通过“查看类别”超链接来修改、删除类别。在查看商品页中，类别按照父类 1：子类 1、子类 2……；父类 2：子类 1、子类 2……的格式显示。修改类别名称时，直接在对应的类别文本框中输入新类名，然后单击对应的“修改”按钮即可。删除时则直接单击对应的“删除”按钮。当要删除的父类包含子类时，则不允许删除父类。查看类别页面的运行结果如图 26.31 所示。

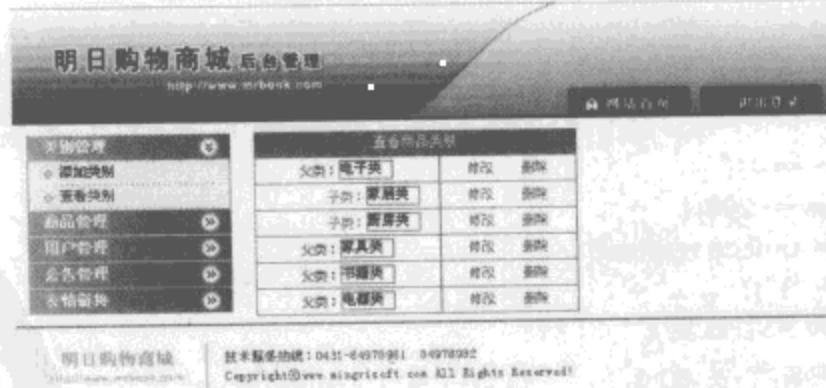


图 26.31 查看类别页面运行结果

首先来介绍 showtype.php 页。该页执行两次查询，分别输出一级类别和二级类别的数组，传递给模板页。showtype.php 页的代码如下：

```

<?php
    include_once 'config.php'; //载入 Smarty 配置文件

```

```

include_once 'conn/conn.php';
$bigsq1 = 'select id,name from tb_class where supid = 0';
$smallsq1 = 'select * from tb_class where supid != 0';
$bigclass = $conn->execute($bigsq1);
$smallclass = $conn->execute($smallsq1);
$bigarray = $bigclass->GetAssoc();
$smallarray = $smallclass->GetAssoc();
$smarty->assign('bigarray',$bigarray);
$smarty->assign('smallarray',$smallarray);
$smarty->display('showtype.tpl');

?>

```

showtype.tpl 模板页通过嵌套 foreach 循环，将一级类别和二级类别一一对应起来，并输出到页面。showtype.tpl 页的代码如下：

```

<!-- 载入 js 脚本文件-->
<script language="javascript" src="js/creatxmlhttp.js"></script>
<script language="javascript" src="js/changetype.js"></script>
<table>
<!-- 省略 html 无用代码 -->
...
<!-- 外层循环，输出一级类别 -->
{foreach name=f1 type key=fkey item=fitem from=$bigarray}
<tr><td>父类：
<input id="modiftype{$fkey}" name="modiftype{$fkey}" type="text" class="shorttxt" value="{$fitem}" style="border-color:#996633;" /></td>
<td>
<input id="modify" name="modify" type="button" value="修改" onclick="modifytype({$fkey});" />
<input id="delete" name="delete" type="button" value="删除" onclick="delbigtype({$fkey});" />
</td></tr>
<!-- 内层循环，输出对应的二级类别 -->
{foreach name=s1 type key=skey item=sitem from=$smallarray}
{if $sitem[1] == $fkey}
<tr><td>子类：
<input id="modiftype{$skey}" name="modiftype{$skey}" type="text" value="{$sitem[0]}"/></td>
<td><input id="modif" name="modif" type="button" value="修改" onclick="modiftype({$skey});" />
<input id="del" name="del" type="button" value="删除" onclick="delsmalltype({$skey});" />
</td></tr>
{/if}
{/foreach}
{/foreach}
</form>
</table>

```

showtype.tpl 模板中包含的 changetype.js 脚本文件主要有 3 个函数：modifytype()、delbigtype() 和 delsmalltype()，分别控制类别的修改和删除。该脚本文件的代码如下：

```

/* 修改类别名称函数 */
function modifytype(key){
    var nm = 'modiftype'+key;

```

```
var names = document.getElementById(nm).value; //获取类别名称
if(names == ""){
    alert('请填写类别名称');
    document.getElementById(nm).focus();
    return false;
}
/* 将类名、ID 传给 changetype.php 页 */
var url = "changetype.php?action=m&names="+names+"&key="+key;
xmlhttp.open("GET",url,true);
/* 调用 check 函数，返回处理结果*/
xmlhttp.onreadystatechange = check;
xmlhttp.send(null);
}
/* 删除大类，解释同上 */
function delbigtype(key){
    if(confirm("您要删除的是一级类，您确定要删除吗？")){
        var url = "changetype.php?action=bd&key="+key;
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = check;
        xmlhttp.send(null);
    }else{
        return false;
    }
}
/* 删除小类，解释同上 */
function delsmalltype(key){
    if(confirm("确定要删除选中的项目吗？一旦删除将不能恢复！")){
        var url = "changetype.php?action=sd&key="+key;
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = check;
        xmlhttp.send(null);
    }else{
        return false;
    }
}
/* 根据不同的返回结果进行不同的操作 */
function check(){
    if(xmlhttp.readyState == 4){
        if(xmlhttp.status == 200){
            var msg = xmlhttp.responseText;
            if(msg == "1"){
                alert('类名重复');
            }else if(msg == "2"){
                alert('操作失败！');
            }else if(msg == '3'){
                alert('操作成功');
                location='showtype.php';
            }else if(msg == '4'){
                alert('该大类有子类，不能删除');
            }else if(msg == '0'){
                alert('未知错误！'+msg);
            }
        }
    }
}
```

无论是执行修改还是删除操作，都是调用 changetype.php 页进行处理的，当执行修改操作时，传递的 action 参数为 m；当执行删除大类时，action 为 bd；当执行删除小类时，action=sd。changetype.php 页的代码如下：

```
<?php
/*
*$reback 说明：
0、未知错误 1、类名重复 2、操作失败 3、操作成功 4、有二级分类
*/
include_once 'conn/conn.php';
$action = $_GET['action']; //获取动作类型
$reback = '';
/* 修改操作 */
if($action == 'm'){
    /* 获取类别名称和 ID */
    $names = $_GET['names'];
    $key = $_GET['key'];
    /* 检测添加名称是否存在 */
    $sql = "select * from tb_class where name = '$names'";
    $rst = $conn->execute($sql);
    if($rst->RecordCount() == 1){
        $reback = '1';
    }else{
        /* 如果不存在，创建更新语句，执行更新操作 */
        $updatesql = "select * from tb_class where id = ".$key;
        $updaterst = $conn->execute($updatesql);
        $upd = array();
        $upd["id"] = $key;
        $upd["name"] = $names;
        $update = $conn->GetUpdateSQL($updaterst,$upd); //创建更新语句
        if($conn->execute($update) == false){ //执行更新操作
            $reback = '2';
        }else{
            $reback = '3';
        }
    }
}
/* 删除小类，根据类别 ID，直接删除 */
}else if($action == 'sd'){
    $key = $_GET['key']; //类别 ID
    $delsql = "delete from tb_class where id = ".$key; //删除语句
    if($conn->execute($delsql) == false){ //执行删除操作
        $reback = '2';
    }else{
        $reback = '3';
    }
}
/* 删除大类 */
}else if($action == 'bd'){
    $key = $_GET['key'];
    $sql = "select * from tb_class where supid = ".$key;
```

```

$rst = $conn->execute($sql);
/* 如果要删除的大类有子类存在，则不进行删除处理 */
if($rst->RecordCount() >= 1){
    $reback = '4';
} else {
    /* 否则，删除该小类 */
    $delsql = "delete from tb_class where id = ".$key;
    if($conn->execute($delsql) == false){
        $reback = '2';
    } else {
        $reback = '3';
    }
}
else{
    $reback = '0';
}
echo $reback;
?>

```

## 26.16 订单管理模块设计

### 视频讲解：光盘\TM\Video\26\订单管理模块设计.exe

订单管理模块的功能包括查看、删除和处理订单等操作。其中，查看订单功能和 26.13.4 节的反馈订单实现方法与步骤类似，而删除操作和 26.12.4 节的删除商品模块类似。所以，这里只给出处理订单的功能实现。

处理订单是指将用户提交的订单进行处理。默认的订单状态是“未处理”，可以将订单的状态修改为“已付款”、“已发货”和“已收货”3 种。处理订单页面的运行结果如图 26.32 所示。

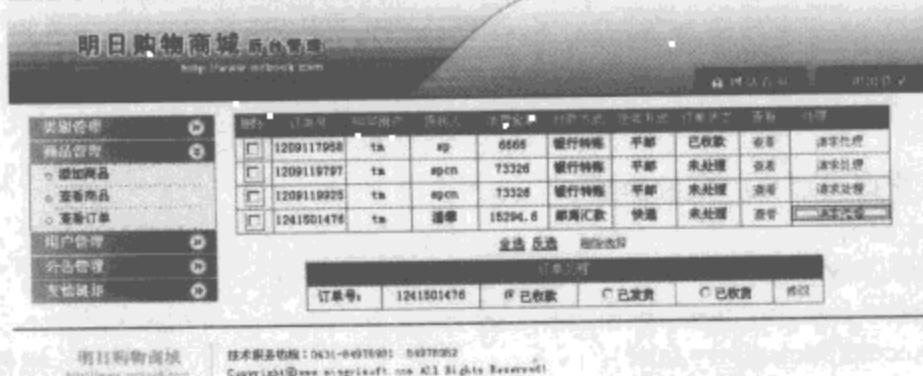


图 26.32 处理订单页面运行结果

限于篇幅，这里只给出与订单相关的程序代码。首先介绍模板页 shwform.tpl，该页显示所有订单的记录。在每条记录的后面，可以对订单进行查看及处理。当单击“请求处理”按钮时，该页面下方的隐藏表单将变为显示状态，并将订单号显示在该表单中。该隐藏表单包含一组单选按钮，可以对表单状态进行修改。shwform.tpl 页的代码如下：

```

<!-- 载入 js 脚本文件 -->
<script language="javascript" src="js/creatxmlhttp.js"></script>
<script language="javascript" src="js/shwform.js"></script>
<table border="0" cellspacing="0" cellpadding="0" align="center">
    <form id="shwfrm" name="shwfrm" method="post" action="#">
        <tr>
<!-- 省略部分为要显示的订单信息的字段名 -->
        ...
        <td>订单状态</td>
        <td>处理</td>
    </tr>
    {foreach key=key item=item from=$formarr}
    <tr>
<!-- 省略部分为显示的订单信息 -->
        <td>
<!-- 根据不同的状态值，显示不同的结果 -->
{if $item.state == 0}未处理{elseif $item.state == 1}已收款{elseif $item.state == 2}已发货{elseif $item.state == 3}已收货{/if}
        </td>
<!-- 单击“请求处理”按钮，调用 showme() 函数 -->
        <td><input id="deal" name="deal" type="button" value="请求处理" onclick="showme({$item.formid})" />
    </td>
    </tr>
    {/foreach}
    </form>
</table>
<!-- 隐藏的表单，ID 为 chdl -->
<div id="chdl" style="display:none;">
    <table width="525" border="0" align="center" cellpadding="0" cellspacing="0">
        <form id="changedeal" name="changedeal" method="post">
            <tr><td colspan="6" style="text-align:center;">订单处理</td></tr>
            <tr><td>订单号：</td></tr>
<!-- 当隐藏表单显示时，该<div>标签将显示被激活订单的 formid -->
            <td><div id="formid">&ampnbsp</div></td>
<!-- 下面为一组单选按钮，它们有着相同的名称和不同的 ID * /
            <td><input id="acceptmon" name="acceptsend" type="radio" value="1" checked="checked" />已收款</td>
            <td><input id="sendwa" name="acceptsend" type="radio" value="2" />已发货</td>
            <td><input id="acceptwa" name="acceptsend" type="radio" value="3" />已收货</td>
<!-- 单击“修改”按钮，会调用 changeme() 函数来更新订单状态 -->
            <td><input id="chg" name="chg" type="button" value="修改" onclick="return changeme(changedeal)" /></td>
        </tr>
    </form>
</table>
</div>

```

在该页中所调用的函数都包含在 shwform.js 脚本文件中。showme() 函数的作用是改变 id=chdl 的 `<div>` 标签的状态，唯一的参数是该条订单的订单号（formid），并在 id=formid 的 `<div>` 标签中显示出来；另一个函数 changeme() 的作用是将 formid 值和被选中的单选按钮值取出并传给 changestate.php 进行处理，根据处理返回值刷新本页面。shwform.js 页面的主要代码如下：

```

/* 显示/隐藏表单 */
function showme(formid){
    /* 判断 “id=chdl” 的<div>标签的样式，如果是显示，则改为隐藏 */
    if(document.getElementById('chdl').style.display == ""){
        document.getElementById('chdl').style.display = 'none';
    /* 否则，将该标签显示出来，并将 formid 显示在 “id=formid” 的<div>标签中 */
    }else{
        document.getElementById('chdl').style.display = "";
        document.getElementById('formid').innerHTML = formid;
    }
}

/* 改变状态 */
function changeme(form){
    /* 获得 formid */
    var fid = document.getElementById('formid').innerText;
    var lang = form.acceptsend.length;
    var state;
    /* 循环查看单选按钮的状态，并取得被选中按钮的 value 值 */
    for(var i=0; i < lang; i++){
        if(form.acceptsend[i].checked == true){
            state = form.acceptsend[i].value;
        }
    }
    if(state == undefined){
        alert('请选择处理项');
        return false;
    }
    /* 将 formid 和 state 值传给 changestate.php 页 */
    var url = "changestate.php?formid="+fid+"&state="+state;
    /* 使用 XMLHttpRequest 对象调用该 url */
    xmlhttp.open("GET",url,true);
    /* 对返回进行处理响应 */
    xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
            var msg = xmlhttp.responseText;
            if(msg == '1'){
                location.reload();
            }else{
                alert('修改失败'+msg);
            }
        }
    }
    xmlhttp.send(null);
}

```

最后的 changestate.php 负责将改变的记录更新到数据表中。changestate.php 页的代码如下：

```

<?php
include_once 'conn/conn.php'; //调用数据库连接文件
$formid = $_GET['formid'];
$state = $_GET['state'];
$reback = '0'; //声明返回值，初始化为 0

```

```

$arr = array();
$sql = "select * from tb_form where formid = '".$formid."'";
$rst = $conn->execute($sql);
if($rst->RecordCount() != 1){
    $reback = '2';
} else{
    $arr['state'] = $state;
    $UpdateSQL = $conn->GetUpdateSQL($rst, $arr);
    if(false == $conn->execute($UpdateSQL)){
        $reback = '3';
    } else{
        $reback = '1';
    }
}
echo $reback;
?>

```

## 26.17 开发的常见问题

### ■ 视频讲解：光盘\TM\Video\26\开发的常见问题.exe

在本系统开发和后期测试的过程中，开发人员遇到了各种各样的疑难问题。这里精选一些常见的、容易被忽略的问题加以讲解，希望可以为初学者提供一些帮助，在开发程序时少走一些弯路。

### 26.17.1 解决 Ajax 的乱码问题

**问题描述：**当使用 Ajax 传递数据时，要么在数据处理页中数据不能被正确处理，要么输出返回值时显示无法识别的乱码。

**解决方法：**这是因为 PHP 在传递数据时使用的编码默认为 UTF-8，造成了非英文字符不能正确传递的情况。解决方法如下：

在所有的 PHP 页中都输入代码：

```
Header("Content-Type:text/html;charset=gb2312");
```

这样，所有的页面即可正确显示。

### 26.17.2 使用 JS 脚本获取、输出标签内容

**问题描述：**获取、更改表单元素值和特定标签内容。

**解决方法：**使用 JS 脚本获取页面内容的方式主要有两种。第一种是通过表单获取表单元素的 value 值。格式为：

```
表单名称.元素名.value
```

该方式只能获取表单中的元素值，对于其他标签元素则无能为力。而第二种方式可以通过 ID 名来

获取页面中任意标签的内容：

格式 1

```
document.getElementById('id').value;
```

格式 2

```
document.getElementById('id').innerText;
```

使用第二种方式时要注意，标签的 ID 名必须存在而且唯一，否则就会出现错误。向标签内容赋值时，则使用如下的格式：

```
id.innerHTML = '要显示的内容';
```

### 26.17.3 使用浮动框架做关联菜单

问题描述：在添加商品时，商品类别选项是一个关联菜单。在不刷新本页的前提下实现该功能十分复杂（当然，如果使用了 ADODB 和 Ajax，那就另当别论了）。

解决方法：这里使用了一个比较灵活的方法，就是使用浮动框架技术，将一个二级管理菜单放到单独的一个网页中，然后在框架中显示。当选择不同类别时，菜单页虽然也会刷新，但整个的大表单页则不会跟着刷新。其关键代码如下：

```
<tr>
<td>商品类别： </td>
<td>
<iframe id="menu1" name="menu1" src="menu.php" width="300" height="22" frameborder="0" scrolling="no" style=" margin-top:0px; margin-left: 0px; top:0px;"></iframe>
<input id="stype" name="stype" type="hidden" value="" />
</td>
</tr>
```

加粗的代码部分为隐藏表单，保存当前被选择的类别 ID。添加商品处理页就是从这个隐藏表单中获取商品类别的。

### 26.17.4 禁用页面缓存

问题描述：使用 Ajax 技术可以防止页面刷新，但有时也会产生新的问题。例如，在“会员管理”页面中，如果连续“冻结”和“解冻”会员，超过 3 次后，该功能就会失效。因为在一定时间内，如果做相同的操作，那么 XMLHttpRequest 对象会执行缓存里有的页面，从而造成操作失败。

解决办法：使用 header() 函数将缓存关闭。使用如下语句：

```
header("CACHE-CONTROL:NO-CACHE");
```

将这句代码添加到 XMLHttpRequest 对象所调用的处理页的顶部即可。

### 26.17.5 在新窗口中使用Session

问题描述：使用js的open方法打开新窗口时，原浏览器中的Session值不会被传递到新窗口中，这会造成数据查询失败。

解决方法：将Session值另存到隐藏域或随着url一起传递到新窗口。代码如下：

```
<!-- 在模板页中，将 Session 值赋给隐藏域 -->
<input id="uid" name="uid" type="hidden" value="{$smarty.session.id}">
...
/* 在 js 脚本中，获取到隐藏域 value 值 */
function getInput(){
    var uid = document.getElementById('uid').value;
    /* 将获取的 value 值通过 url 传给新页面 */
    open("operator.php?uid="+uid, '_blank', '', false);
    ...
}
```

### 26.17.6 防止站外链接

问题描述：用户可以在没有登录的情况下访问某些网页，或者通过站外访问、盗链等。

解决方法：通过预定义变量来防止站外链接。`$_SERVER['HTTP_REFERER']`变量可以获取前页的url。如果该变量为空，说明用户是直接从地址栏中输入地址的；如果该变量的服务器地址和系统服务器地址不符，说明是非法用户站外链接。该功能的完整代码如下：

```
<?php
session_start(); //开启 Session 支持
if(!isset($_SESSION['id']) or !isset($_SESSION['member'])){
    echo "<script>alert('您没有登录或超时');history.back;</script>";
    exit();
}
$ref = $_SERVER['HTTP_REFERER'];
if($ref == ""){
    echo '对不起，不允许从地址栏访问!';
} else{
    $url = parse_url($ref);
    if($url[host] != '127.0.0.1' && $url[host] != 'localhost'){
        echo '不允许盗链';
        exit();
    }
}
?>
```

### 26.17.7 判断上传文件格式

问题描述：添加商品时，可以上传商品的图片，但有时可能会误传非图片格式的文件。这里通过

自定义函数判断上传文件的后缀。

解决方法：创建自定义函数。f\_postfix()函数的代码如下：

```
/*
 * 判断文件后缀
 * $f_type: 允许文件的后缀类型（数组）
 * $f_upfiles: 上传文件名
 */
function f_postfix($f_type,$f_upfiles){
    $is_pass = false;
    $tmp_upfiles = split(".", $f_upfiles); // 使用 split() 函数分割文件
    $tmp_num = count($tmp_upfiles); // 查找文件后缀
    if(in_array(strtolower($tmp_upfiles[$tmp_num - 1]),$f_type)) // 判断后缀是否在允许列表内
        $is_pass = $tmp_upfiles[$tmp_num - 1]; // 如果是，则将后缀名赋给变量
    return $is_pass; // 返回变量
}
```

## 26.18 网站运行错误分析

### ■ 视频讲解：光盘\TM\Video\26\网站运行错误分析.exe

电子商务平台开发完成，测试成功后即可投入使用，但是将程序上传到服务器、或者复制到其他机器中运行时，却又出现了一些错误。下面将分析在这种情况下出现错误的原因以及解决方法。

#### 26.18.1 数据库导致的错误

在运行本系统时，由于没有附加数据库或者数据库的用户名和密码不正确，会出现如图 26.33 和图 26.34 所示的错误信息。

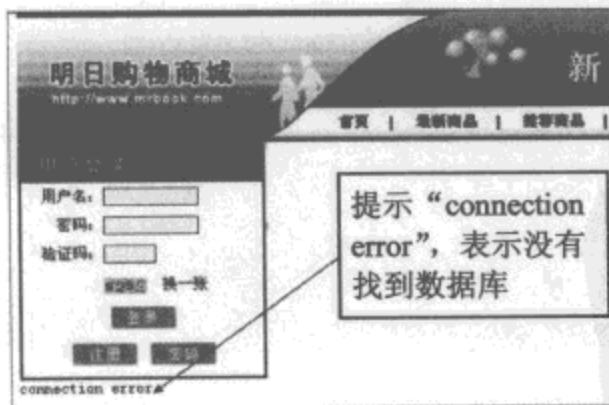


图 26.33 没有附加 MySQL 数据库导致的错误



图 26.34 MySQL 数据库用户名和密码不正确导致的错误

解决方法如下：

(1) 从本系统中的 26/data 文件夹下复制本系统的数据库，附加到本机 MySQL 数据库的 data 文件夹下。

(2) 如果 MySQL 数据库的用户名和密码与本系统所使用的用户名和密码不一致，有如下两种解决方案。

第一种，修改本系统的数据库连接文件，找到 conn 文件夹下的 conn.php 文件，修改其中的用户名和密码，使其与本机 MySQL 数据库的用户名和密码一致。conn.php 文件的代码如下：

```
<?php
include_once ('inc/char.php');
include_once ("adodb5/adodb.inc.php");
$conn = ADONewConnection('mysql');
$conn->PConnect('localhost','root','root','db_database26') or die('connection error');
$conn->Execute('set names gb2312');
$ADODB_FETCH_MODE = ADODB_FETCH_BOTH;
?>
```

本系统中 MySQL 数据库的用户名为 root，密码为 root，数据库名称为 db\_database26。

 注意：如果要修改本系统连接 MySQL 数据库的用户名和密码，需要修改两个 conn.php 文件，一个在 26/conn 文件夹下，另一个在 26/admin/conn 文件夹下。

第二种，修改本机 MySQL 数据库的用户名和密码，该操作可以通过 phpMyAdmin 图形化管理工具来完成，具体操作方法可以参考本书第 18 章的实战。

## 26.18.2 Smarty 模板的配置错误

由于本系统是应用 Smarty 模板进行开发的，所以在运行本系统时需要对 Smarty 模板进行配置，如果没有正确指定 Smarty 类库的位置或者 Smarty 类库不存在，将出现如图 26.35 所示的错误。

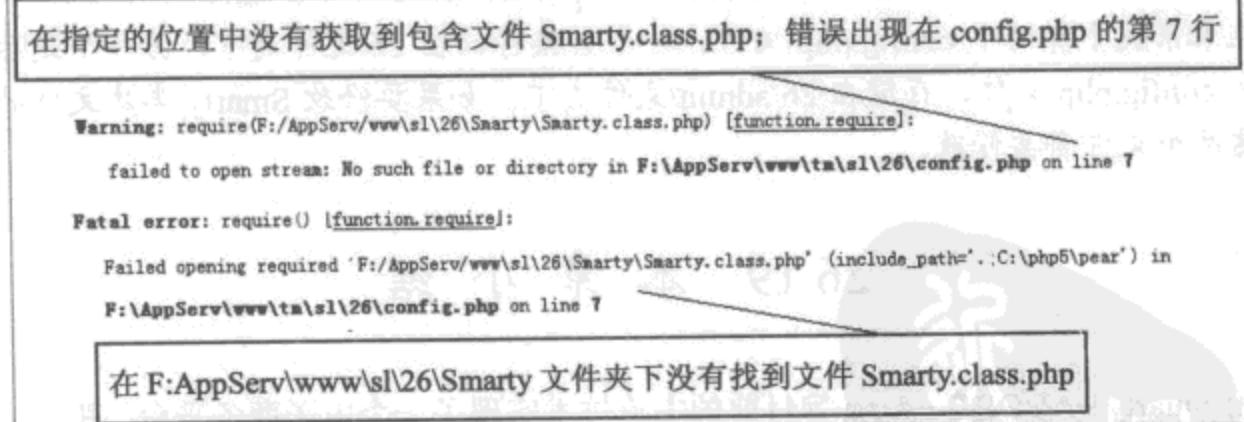


图 26.35 配置 Smarty 的路径不正确

上述错误出现在本系统的 config.php 文件的第 7 行，即在指定的路径下没有找到 Smarty.class.php 文件。

修改这个错误之前，首先要明确几个目录在用户机器上的位置：

- ☑ 确定用户计算机上 Apache 服务器的根目录（即 PHP 程序运行文件夹）。
- ☑ 确定本系统在 Apache 服务器根目录下的位置。
- ☑ 确定 Smarty.class.php 文件在本系统中的位置。

在明确了这 3 个目录之后，再来修改这个错误信息：

- ☑ 确定 Apache 服务器根目录 “F:\AppServ\www”。
- ☑ 确定本系统在服务器下的位置 “\tm\sl\”。
- ☑ 确定 Smarty.class.php 文件在本系统中的位置 “26\Smarty\Smarty.class.php”。

确定了上述 3 个目录以后，该错误即修改完成。在“F:\AppServ\www\tm\sl\26”文件夹下打开 config.php 文件，锁定第 7 行的内容，其中常量 BASE\_PATH 指定的是服务器的根目录，没有任何问题；错误出现在常量 SMARTY\_PATH 指定的 Smarty 类库文件的绝对路径上，图 26.35 所示指定 Smarty 类库文件的绝对路径是 “\sl\26\Smarty\\”，而实际的绝对路径应该是 “\tm\sl\26\Smarty\\”。config.php 的正确代码如下：

```
<?php
/* 定义服务器的绝对路径 */
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']); //获取服务器的根目录 D:/AppServ/www
/* 定义 Smarty 目录的绝对路径 */
define('SMARTY_PATH','\tm\sl\26\Smarty\\'); //指定 Smarty.class.php 文件在这个服务器中的实际位置
/* 加载 Smarty 类库文件 */
require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
/* 实例化一个 Smarty 对象 */
$smarty = new Smarty;
/* 定义各个目录的路径 */
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs/';
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';
?>
```

 注意：在本系统中有两个 config.php 文件，一个直接存储在 26/文件夹下，另一个是后台管理系统的 config.php 文件，存储在 26/admin/文件夹下。如果要修改 Smarty 类库文件的绝对路径，这两个文件都要修改。

## 26.19 本章小结

本章使用 Smarty、ADODB、Ajax 等目前的主流技术实现了一个电子商务平台。其中，应用 Smarty 模板实现网站的动静分离，通过 ADODB 操作 MySQL 数据库，通过 Ajax 实现无刷新验证，通过 DIV、JavaScript 和 CSS 实现树状导航菜单，并且对在开发过程中出现的疑难问题进行了总结。希望读者能通过这个项目实例，将本书前面章节中讲解的内容融会贯通，掌握网络开发的整体操作流程。

# 第27章

## 办公自动化系统

办公自动化英文原称 Office Automation，缩写为 OA，办公自动化系统即能实现办公室内事务性业务的自动化。

办公自动化没有明确的定义，最普遍的说法是：凡是在传统的办公室中采用各种新技术、新机器、新设备从事办公业务，都属于办公自动化的领域。

办公自动化系统与办公室自动化在概念上存在一定的差别。办公自动化通常指办公室中配备具有自动化功能的设备，这些设备能实现某些办公活动自动化或某个单位业务的自动化处理；而办公自动化系统则是在办公室自动化功能的基础上发展起来，以办公自动化技术为主体，同人、组织、制度、环境等相结合的完整的系统。

通过阅读本章，可以学习到：

- 办公自动化管理系统的开发流程
- 进一步学习如何做项目需求分析与系统设计
- 页面布局中的框架布局
- 使用递归函数做多级下拉菜单
- 系统日志的实现



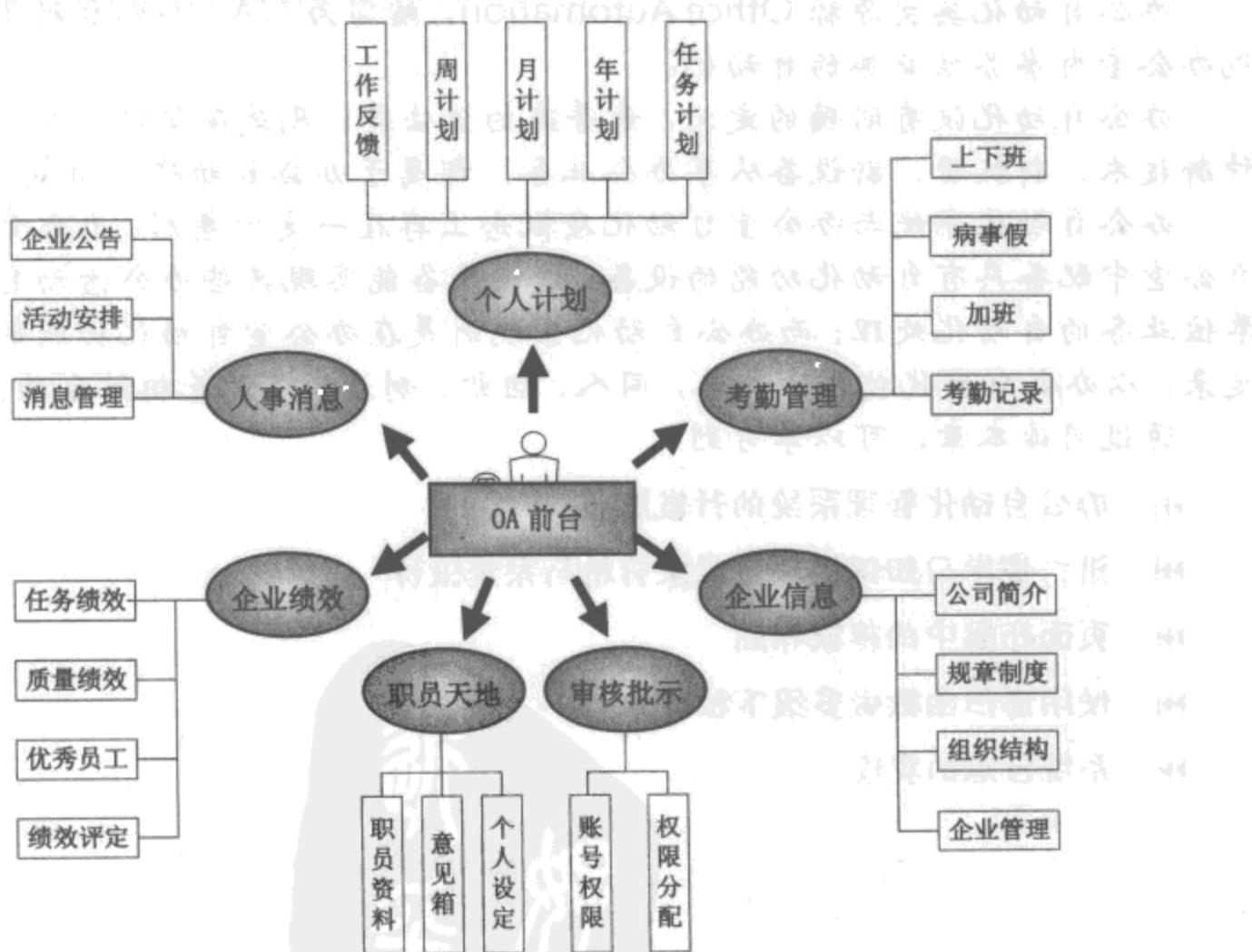
## 27.1 项目描述

本系统是针对中小型企业内部自动化办公管理的要求进行设计的，将达到以下目标：

- 键盘操作，快速响应。
- 实现了文件类信息的强大的管理能力。
- 实现对员工基础信息（人事消息）的管理功能等。
- 实现了个人办公的信息自动化管理功能。
- 发布会议信息，并对会议信息进行管理。
- 可以对系统用户进行管理。
- 为了加强数据保密性，为每个用户组设置权限级别。
- 最大限度地实现易安装性、易维护性和易操作性。
- 运行稳定、安全可靠。

根据系统分析，规划出本系统前后台的功能结构。

办公自动化系统前台功能结构图如图 27.1 所示。



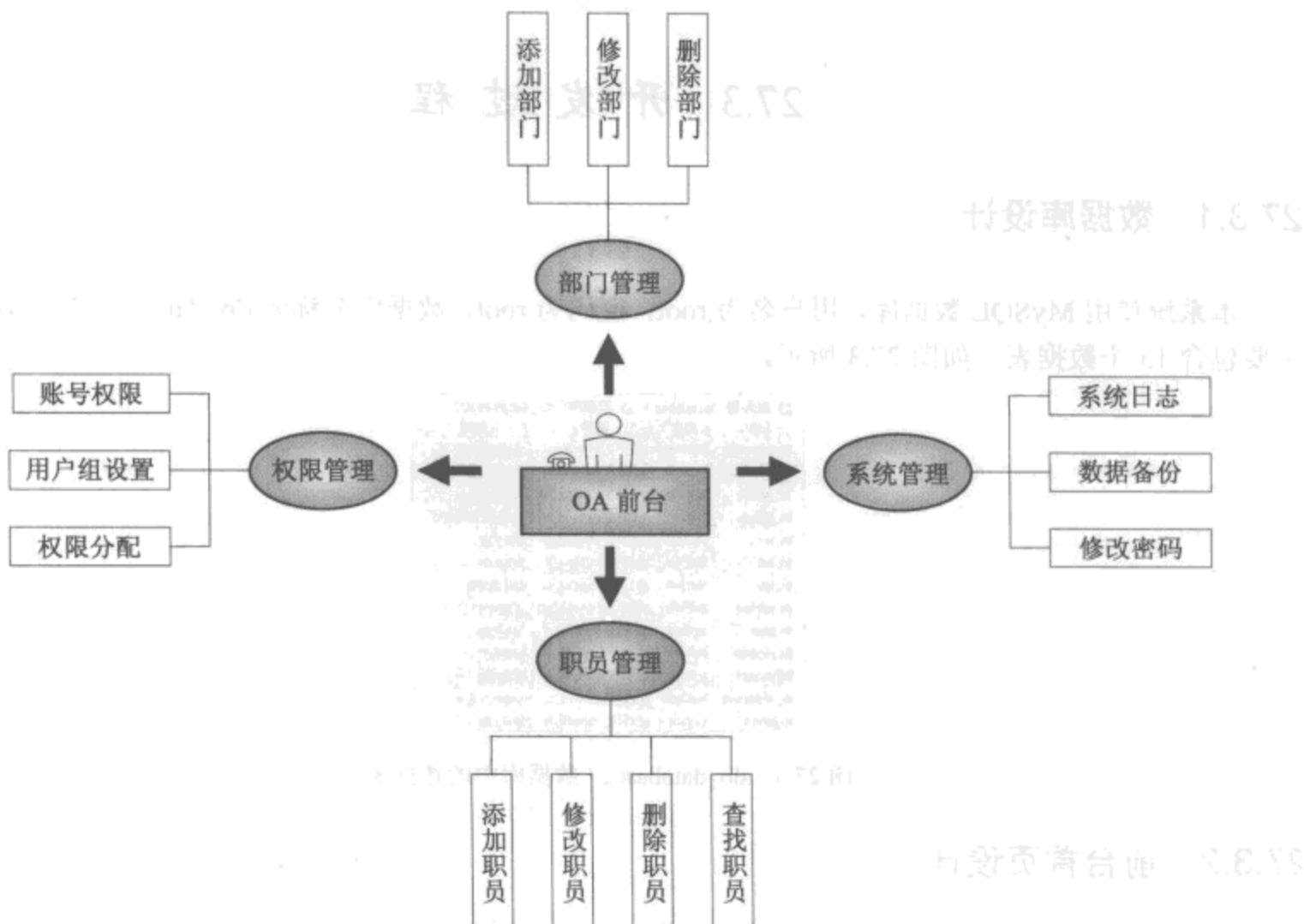


图 27.2 办公自动化后台管理系统

## 27.2 技术准备

在开发办公自动化管理系统平台时，该项目所使用的软件及开发环境如下。

服务器端

- 操作系统：Windows 2003 Server/Linux（推荐）。
- 服务器：Apache 2.2。
- PHP 软件：PHP 5.1.6。
- 数据库：MySQL 5.0.24。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- 开发工具：Dreamweaver 8。
- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

客户端

- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

## 27.3 开发过程

### 27.3.1 数据库设计

本系统使用 MySQL 数据库，用户名为 root，密码为 root，数据库名称为 db\_database27。本数据库主要包含 13 个数据表，如图 27.3 所示。

表	类型	索引	说明
tb_company	MyISAM	gb2312_chinese_ci	企业信息列表
tb_controller	MyISAM	gb2312_chinese_ci	管理员列表
tb_depart	MyISAM	gb2312_chinese_ci	部门管理列表
tb_group	MyISAM	gb2312_chinese_ci	用户组管理列表
tb_iss	MyISAM	gb2312_chinese_ci	审核列表
tb_it	MyISAM	gb2312_chinese_ci	功能列表
tb_gb	MyISAM	gb2312_chinese_ci	意见和列表
tb_person	MyISAM	gb2312_chinese_ci	人事列表
tb_plan	MyISAM	gb2312_chinese_ci	计划列表
tb_register	MyISAM	gb2312_chinese_ci	登记列表
tb_setup	MyISAM	gb2312_chinese_ci	时间列表
tb_supervisor	MyISAM	gb2312_chinese_ci	优秀员工列表
tb_users	MyISAM	gb2312_chinese_ci	用户列表

图 27.3 db\_database27 数据库中的数据表

### 27.3.2 前台首页设计

前台主要是员工操作页面，其功能的使用权限由后台管理员进行分配。前台首页页面如图 27.4 所示，主要显示企业公告信息、活动安排信息、个人计划信息及审核批示信息，此处的“个人计划”和“审核批示”只显示当前员工的计划和批示情况，在“任务绩效”、“质量绩效”及“批示审核”页面中可对所有员工信息进行查询。

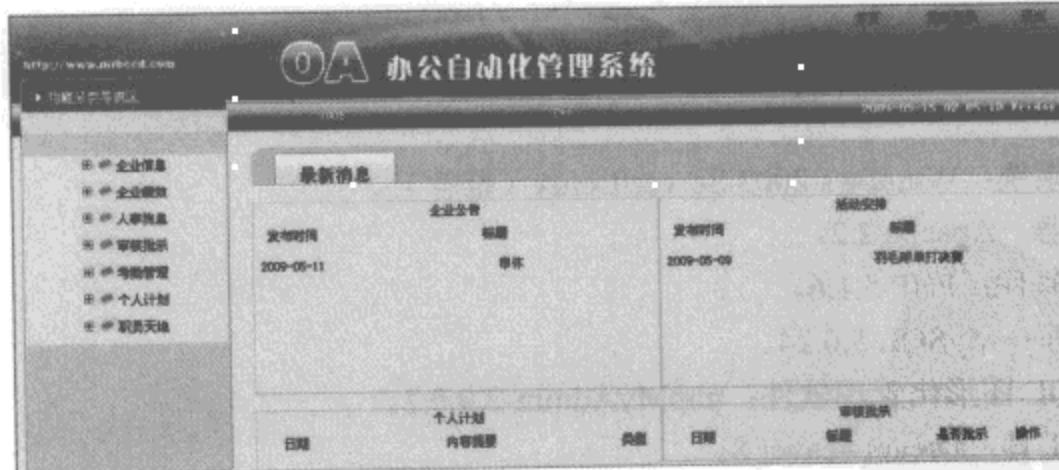


图 27.4 前台首页页面

#### (1) 企业信息

通过“企业管理”页面可对“公司简介”进行修改，进行“规章制度”的添加、修改及删除。

#### (2) 个人计划

通过“个人计划”中的各页面可以添加工作反馈、周计划、月计划、年计划及任务计划。

### (3) 企业绩效

通过“任务绩效”页面可查看及删除所有员工的年计划和任务计划，通过“质量绩效”页面可查看周计划、月计划及工作反馈信息。

### (4) 人事消息

通过“消息管理”页面可对公告信息及活动安排信息进行发布、修改及删除。

### (5) 审核批示

在“发布审核”页面中可以发布、修改及删除审核信息，在“批示审核”页面中可审批及删除审核信息。

### (6) 考勤管理

通过“上下班登记”、“病事假登记”、“加班登录”页面可添加签到时间，通过“考勤记录”页面可设置考勤时间、查看及删除员工的考勤记录。

### (7) 职员天地

在“职员浏览”页面中可以查看员工信息；在“个人设定”页面中可修改个人资料；“意见箱”只有部门为“人事部”的员工可以进行回复及删除，其余用户只可以发表意见。

## 27.3.3 人事消息模块设计

人事消息模块主要是对文件的收发进行管理。模块的设计和实现并不是十分复杂，但在整个系统中的位置却很重要。因为公告栏和意见箱面向的是全体用户，企业的最新动向和职员的意见首先都是在这里体现出来的。

人事消息模块主要包含两部分内容，一部分是面向全体用户的，包括查看公告、活动、意见箱等；一部分是仅对人事部开放，其他用户不允许、也不会看到的页面，如公告管理、意见管理等。人事消息用例图如图 27.5 所示。

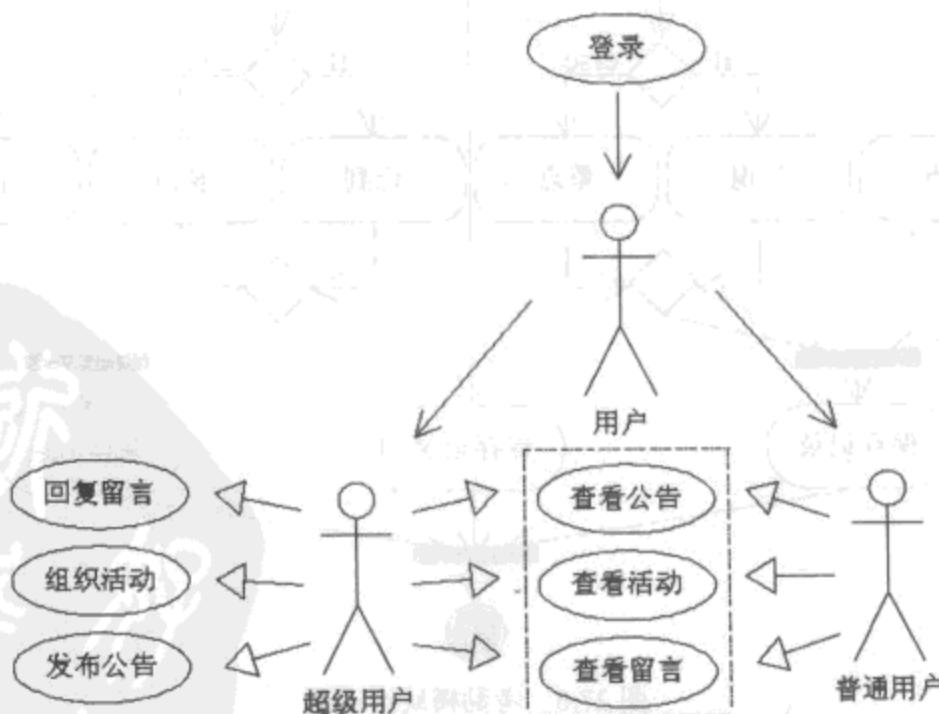


图 27.5 人事消息用例图

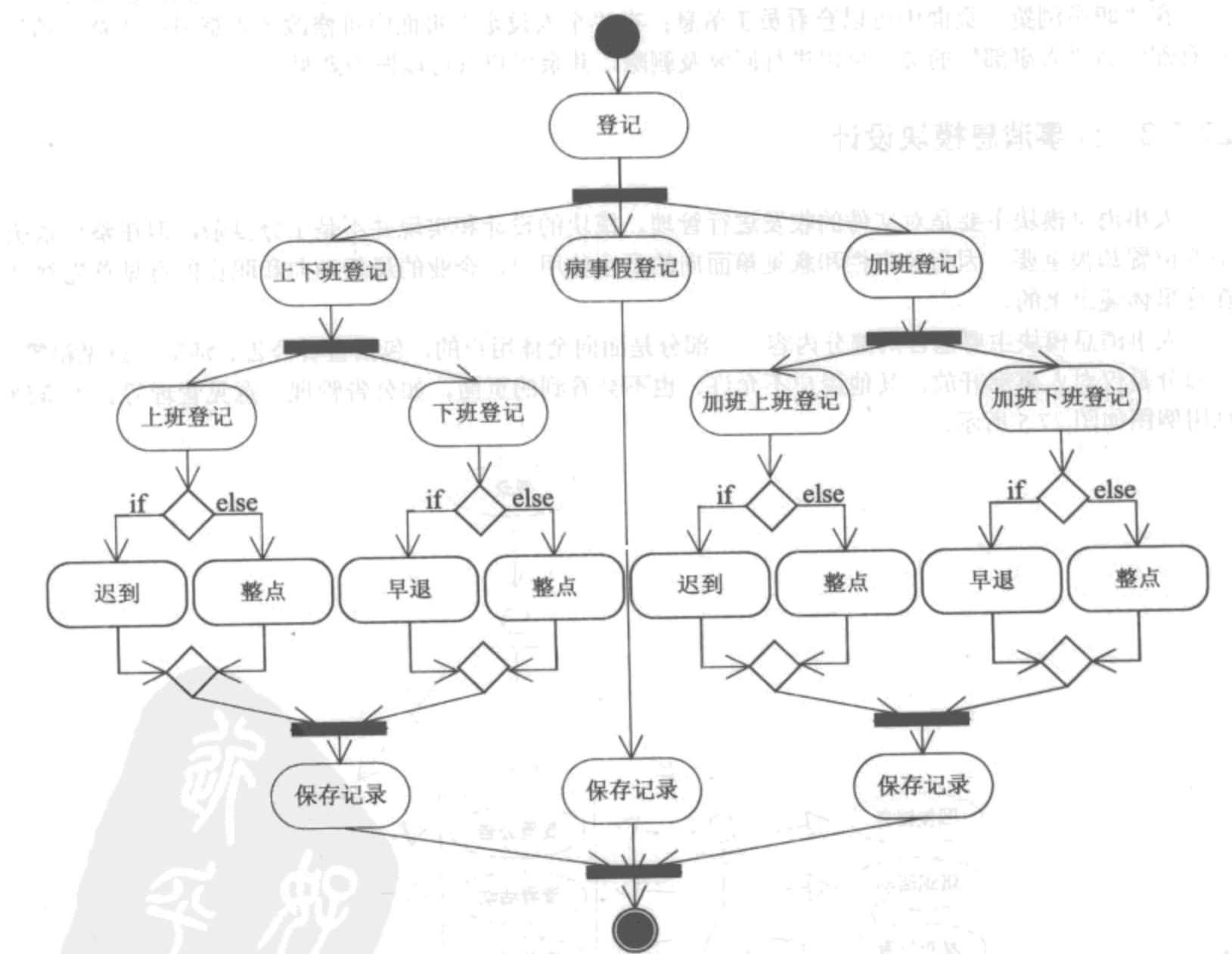
### 27.3.4 考勤管理模块设计

考勤功能是办公自动化管理系统中每天都要使用到的功能模块，也是比较重要的模块之一；除了包括正常的上下班登记功能外，还要包含病事假登记和加班登记等特殊登记功能。管理员通过考勤记录实现对上下班及加班标准时间的设置。

考勤模块主要的功能如下。

- 上下班登记：包括上班登记和下班登记。
- 病事假登记：包括病假登记和事假登记。
- 加班登记：包括加班上班登记和加班下班登记。
- 考勤记录：包括上下班标准时间设置和加班标准时间设置。

考勤模块的活动图如图 27.6 所示。



### 27.3.5 后台首页设计

在 IE 地址栏中输入 “<http://127.0.0.1/tm/sl/27/admin/>” , 即可访问办公自动化系统的后台登录页面。后台操作页面如图 27.7 所示。后台主要对部门信息、职员信息、用户权限及数据库等信息进行管理。

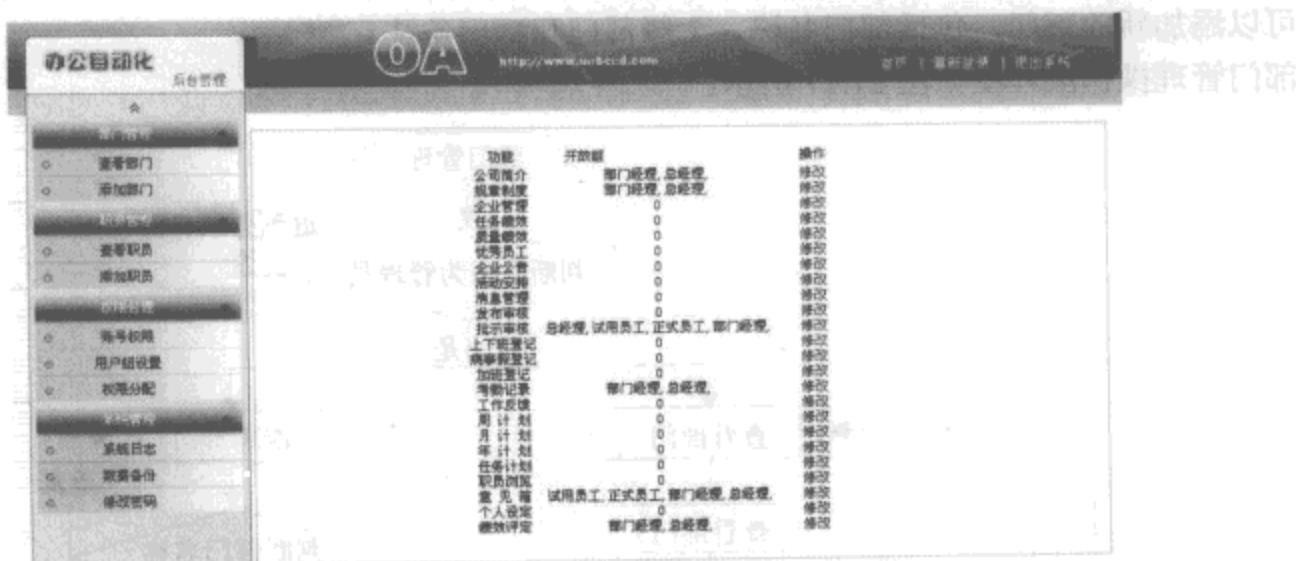


图 27.7 后台操作页面

- 部门管理：实现部门信息的查看、修改和删除。
- 员员管理：实现职员信息的查看、修改和删除。
- 权限管理：设置账号的权限；控制用户属于哪个用户组；控制权限的分配。

**◆ 注意：**当功能没有被指定操作权限组时（就是“开放组”状态为 0 时，如图 27.8 所示），则该功能默认可被所有权限组操作，如图 27.9 所示。但如果该功能被指定操作权限组，则其余权限组将无权操作。而且当新添加权限组时，该功能默认为拒绝操作状态，如图 27.10 所示，如果启用则需要重新添加组。

功能	开放组	操作
公司简介	部门经理, 总经理	修改
规章制度	0	修改
企业管理	0	修改
任务绩效	0	修改
质量绩效	0	修改

图 27.8 权限分配

拒绝访问组	更改所在组	允许访问组
试用员工 正式员工 总经理	添加组>> <<删除组	试用员工 正式员工 部门经理

图 27.9 权限分配

拒绝访问组	更改所在组	允许访问组
试用员工 正式员工 总经理	添加组>> <<删除组	部门经理

图 27.10 权限分配

- 系统管理：管理系统日志，备份数据和修改管理员密码。

### 27.3.6 部门管理模块设计

一个成熟、稳定的企业，一定要有一个合理的部门结构，包括每个部门需要做什么工作、部门之间的协调关系以及部门之间的上下级关系等都要明确下来，这样才能使各部门各尽其职，不会出现资源的浪费。

源浪费，提高企业整体的效率。本系统中部门管理模块包括两大部分：

查看部门

查看各个部门名称和部门之间的上下级关系，还可以对部门的信息和级别进行修改，如果是最低层的部门，则可以对其进行删除操作。

添加部门

可以添加新的部门，包括部门名称、上级部门和部门备注。

部门管理操作流程图如图 27-11 所示。

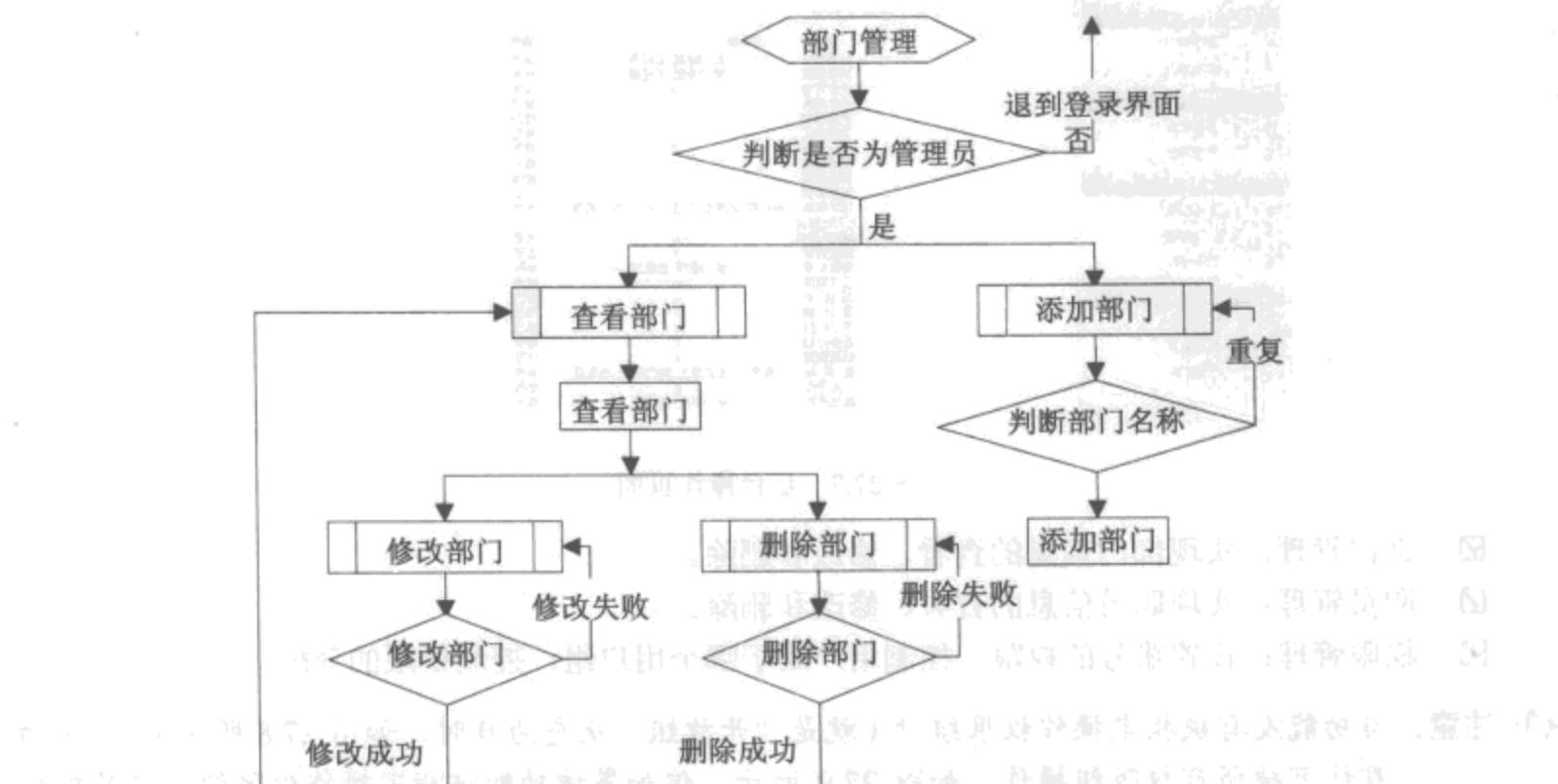


图 27.11 部门管理操作流程图

27.4 本章小结

本章通过一个完整的办公自动化管理系统开发过程，向读者讲述了系统开发的几个重要环节，还包括了几个主要模块的功能实现，最后着重讲解了 MySQL 备份的相关知识。相信通过本章的学习，读者对如何开发一个真实的项目会有进一步的了解，通过自己动手实践，完成一个功能相似的办公自动化管理系统是完全不成问题的。

说明：由于篇幅所限，本章只对办公自动化系统的模块进行了简单介绍，关于办公自动化系统的详细代码可参见本书附带光盘。

# 第28章

总页数 182

## 供求信息网

在全球知识经济和信息化高速发展的今天，信息化是决定企业成败的关键因素，企业需要在网站上发布供求信息，才能在同领域的竞争中得到突飞猛进的发展。

一个广泛、快速、自由的信息交流平台，为用户带来方便的同时，也会给企业带来无限的商机。于是，以因特网为基础的信息交流平台的易查供求信息网出现了。易查供求信息网致力于优化信息交流，实现信息的快速交流。

在本系统中应用到的关键技术和方法如下：

- 使当前窗口承载框架页中的超链接页面
- 自动计算以系统日期为基数的相对日期
- do…while 循环语句
- 查询关键字描红
- 框架技术在 Web 网站中的应用
- 表单数据的两种提交方式
- 通过 MySQL 函数连接、操作数据库
- 通过 SQL 语句实现数据的查询、更新、修改和删除操作



## 28.1 项目描述

XXX 信息科技有限公司是一家以整合渠道资源为主的高科技公司。为了扩大企业规模，增强企业的竞争力，该企业决定向多元化发展，借助 Internet 在国内的快速发展，聚集部分资金投入网站建设，为企业和用户提供综合信息服务，以向企业提供有偿信息服务为盈利方式，打造一个全新的供求信息网。

供求信息网的设计从两个角度出发：

- 致力于客户，包括企业和个人。为企业提供广告、发布供求信息、招聘信息等；为个人提供大量有价值的信息，包括求职、求购、出租、出兑、公寓和培训等。
- 从经济利益的角度出发。以为个人提供大量的、免费的、有价值的信息为基础，吸引更多人来关注网站；以为企业提供广告和各种有偿信息服务为盈利方式。

根据网站设计的角度和企业的要求，确立供求信息网的目标和提供的服务如下。

- 目标：能够提供信息搜索、信息定位描红、发布免费信息、发布付费信息、发布企业广告、对各类发布的信息进行审核、删除、检索等功能。
- 提供的服务：招聘信息、求职信息、培训信息、公寓信息、家教信息、车辆信息、物品求购、物品出售、求租出兑、寻求合作、企业广告等服务。

供求信息网的前台运行效果如图 28.1 所示。



图 28.1 供求信息网前台页面

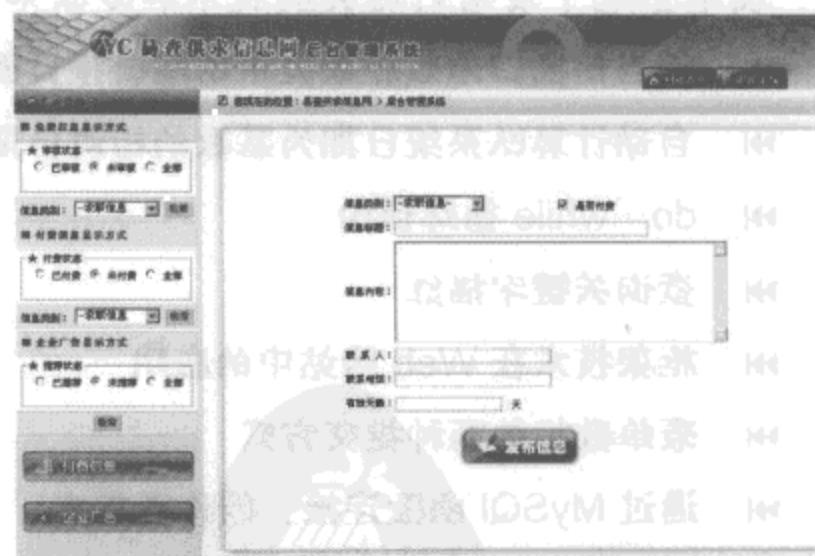


图 28.2 供求信息网后台页面

## 28.2 技术准备

在开发供求信息网时，该项目所使用的软件及开发环境如下。

服务器端

- 操作系统: Windows 2003 Server/Linux (推荐)。
- 服务器: Apache 2.2.
- PHP 软件: PHP 5.1.6.
- 数据库: MySQL 5.0.24.
- MySQL 图形化管理软件: phpMyAdmin-2.9.0.2.
- 开发工具: Dreamweaver 8.
- 浏览器: IE 6.0 及以上版本。
- 分辨率: 最佳效果 1024×768 像素。

 客户端

- 浏览器: IE 6.0 及以上版本。
- 分辨率: 最佳效果 1024×768 像素。

### 28.3 开发过程

#### 28.3.1 数据库设计

本系统是一个中小型的供求信息平台，但是由于平台会涉及海量数据，因此需要充分考虑到成本问题及用于需求（如跨平台）等问题，而 MySQL 是世界上最流行的开放源码的数据库，是完全网络化的跨平台的关系型数据库系统，这正好满足了中小型企业的需求，所以本系统采用 MySQL 数据库。

结合实际情况及对用户需求的分析，供求信息网中应用的 db\_database28 数据库主要包含 4 个数据表，如图 28.3 所示。

#### 28.3.2 前台功能设计

网站前台关乎着网站的建设及形象宣传，它对网站生存和发展起着非常重要的作用。网站前台应该是一个信息含量较高、内容较丰富的宣传平台。供求信息网前台主要包含以下内容：

- 网站菜单导航（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等）。
- 发布免费的供求信息（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等）。
- 推荐供求信息显示（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等），其中，付费信息按时间顺序降序排列，免费信息按时间顺序分页显示。
- 显示推荐的企业广告信息。

服务器: localhost > 数据库: db_database28			
表	类型	备注	说明
tb_admin	MyISAM	gb2312_chinese_ci	管理员信息表
tb_advertising	MyISAM	gb2312_chinese_ci	企业广告信息表
tb_info	MyISAM	gb2312_chinese_ci	免费供求信息表
tb_inforinfo	MyISAM	gb2312_chinese_ci	付费供求信息表

图 28.3 易查供求信息网数据表

- 供求信息快速检索，支持模糊查询和查询关键字描红功能。
  - 后台登录入口，为管理员进入后台提供一个入口。
- 前台功能结构如图 28.4 所示。

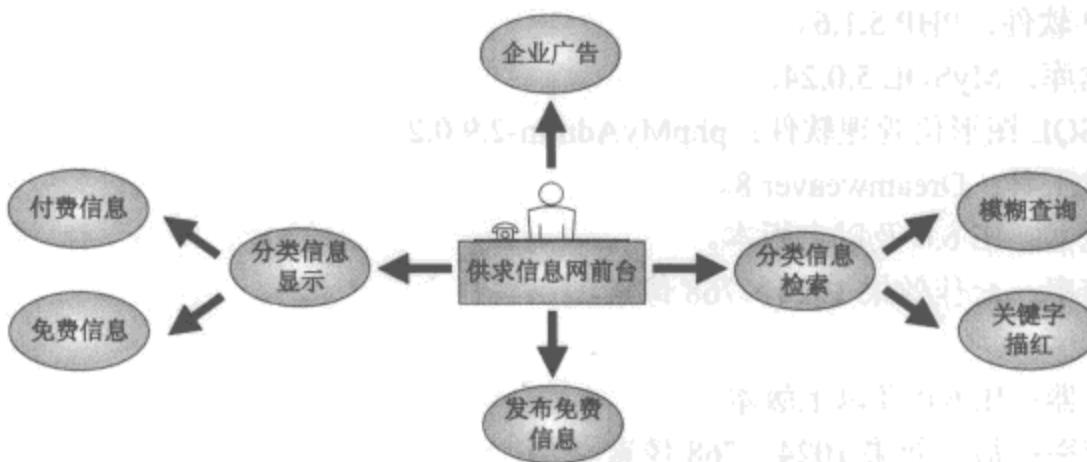


图 28.4 供求信息网前台功能结构图

### 28.3.3 免费供求信息发布模块设计

免费供求信息发布提供对象为供求信息用户，是供求信息网站非常重要的功能，也是供求信息网站的核心功能。

免费供求信息发布模块可以完成 11 种不同类别信息的发布。用户可以根据自身需要将供求信息发布到相应的信息类别中（公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示）。信息发布成功后，需要管理员进行审核，只有审核成功的信息才能显示在前台相应的信息类别网页中。免费供求信息发布的流程如图 28.5 所示。

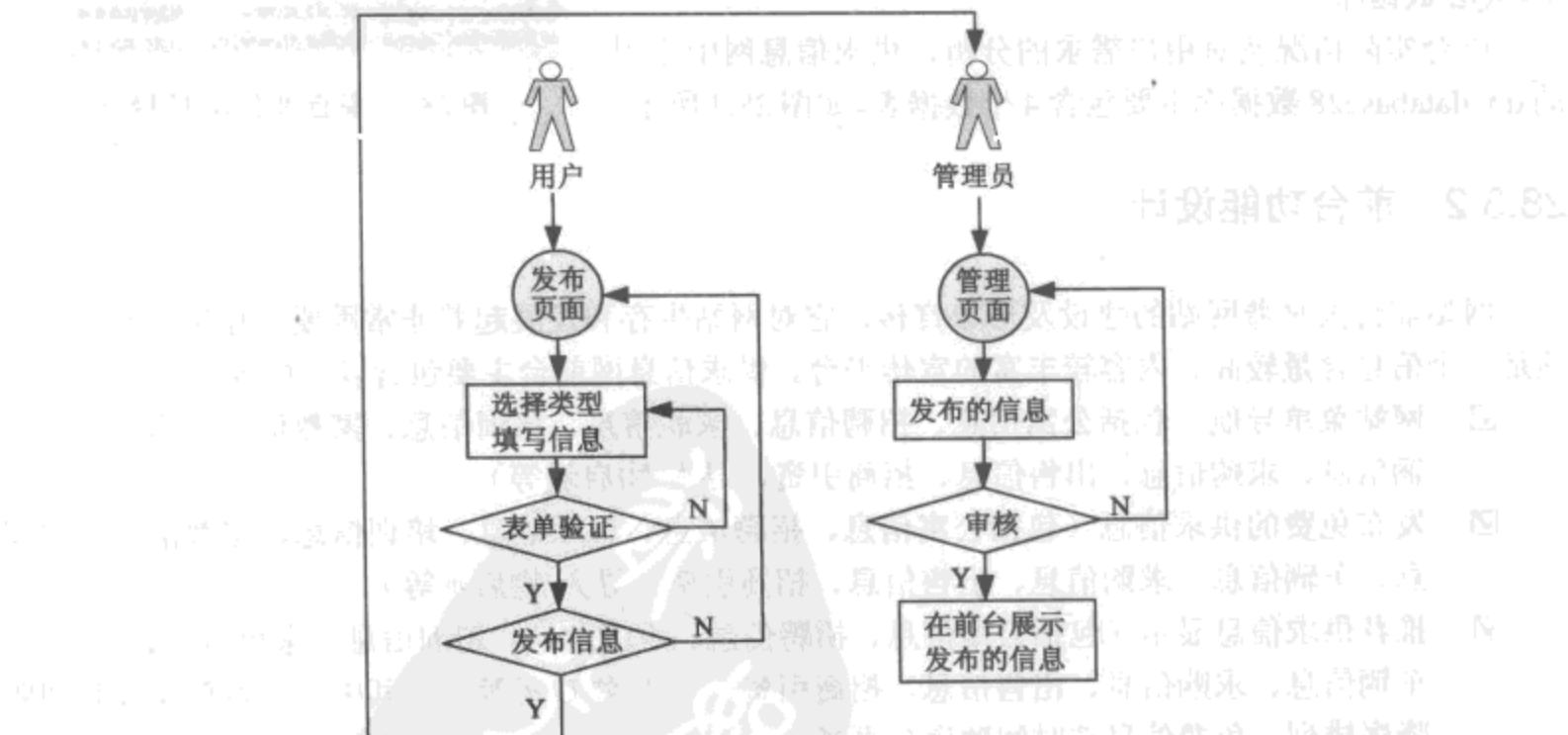


图 28.5 免费供求信息发布流程图

功能创建成功后，用户通过单击前台页面导航栏的“我要发布”超链接，进入信息发布页面，如图 28.6 所示。程序会先来验证用户是否输入信息，若验证失败，则返回信息发布页面，进行相应提示。若验证成功，则向数据库中插入记录，完成发布操作。

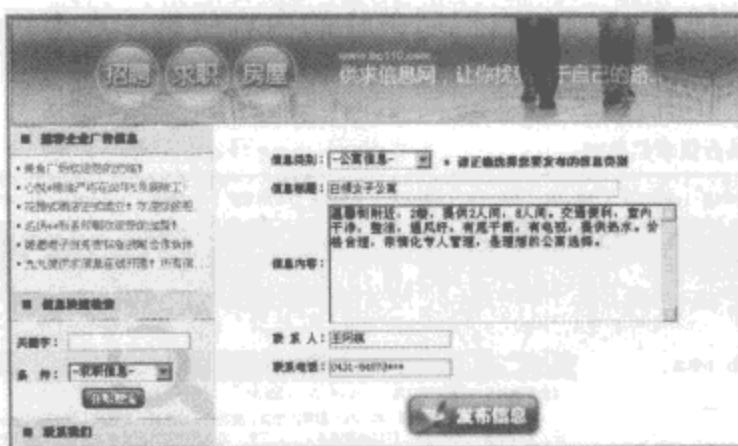


图 28.6 免费供求信息发布网页

#### 28.3.4 信息检索模块设计

信息检索是对已存在于数据库中的数据按条件进行筛选浏览，是查看历史信息和确认数据操作最为快速、有效的办法。信息检索模块主要通过选择信息类型和输出查询关键字模糊查询供求信息资源，并输出查询结果。考虑到供求信息的信息量较大，本模块对与查询关键字相匹配的查询结果进行描红，从而方便用户的浏览。信息检索模块示意图如图 28.7 所示。

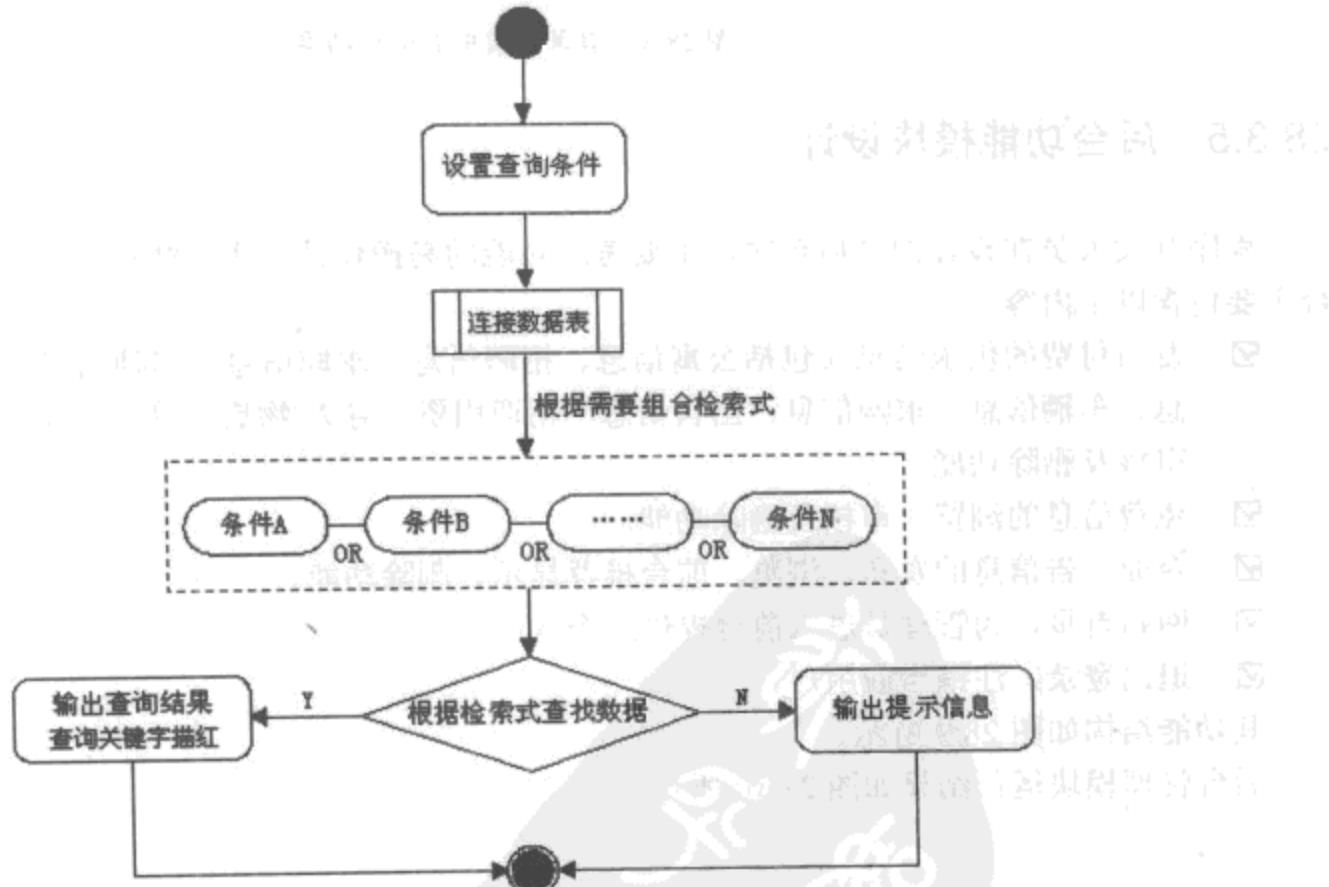


图 28.7 信息检索模块示意图

在开发信息检索模块时，由于该网站含有大量的数据信息，为了方便用户浏览网站信息，需要添加复合条件查询实现搜索功能。在信息检索区的“关键字”文本框中输入欲查询的关键字，在下拉列表框中选择要搜索的信息类型，然后单击“开始搜索”按钮，对指定条件的记录进行检索并输出结果集到浏览器，同时，为了方便浏览者查找自己所关注的内容信息，本模块对查询关键字进行描红。运行结果如图 28.8 所示。

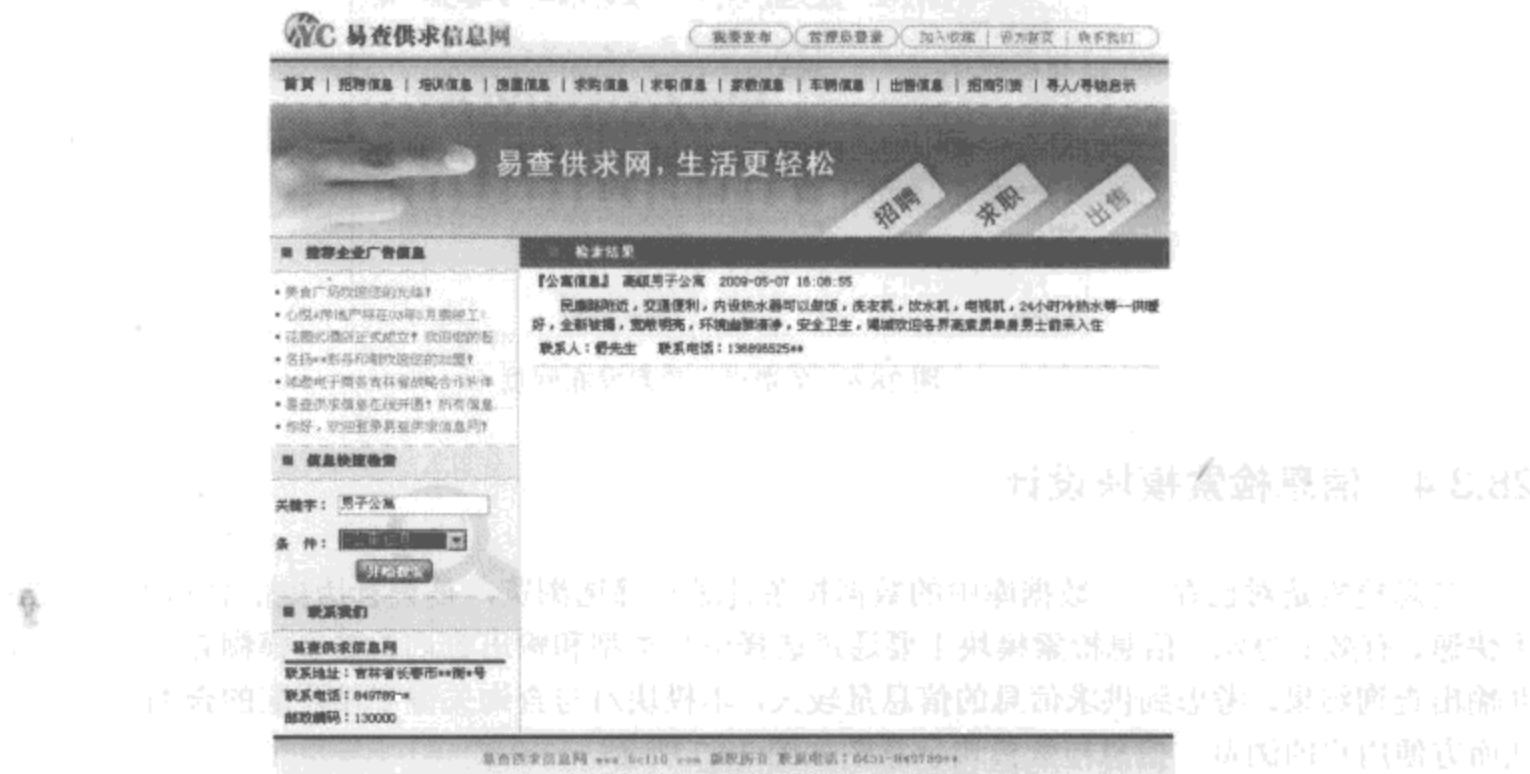


图 28.8 信息检索页面运行结果

### 28.3.5 后台功能模块设计

程序开发人员在设计网站后台时，主要考虑功能的易操作性、实用性和易维护性。供求信息网后台主要包含以下内容：

- 发布付费的供求信息（包括公寓信息、招聘信息、求职信息、培训信息、家教信息、房屋信息、车辆信息、求购信息、出售信息、招商引资、寻人/物启示等），以及付费信息的浏览、审核及删除功能。
- 免费信息的浏览、审核及删除功能。
- 企业广告信息的发布、浏览、前台推荐显示、删除功能。
- 网站首页，为管理员进入前台提供一个入口。
- 退出登录，注销当前用户。

其功能结构如图 28.9 所示。

后台管理模块运行结果如图 28.10 所示。

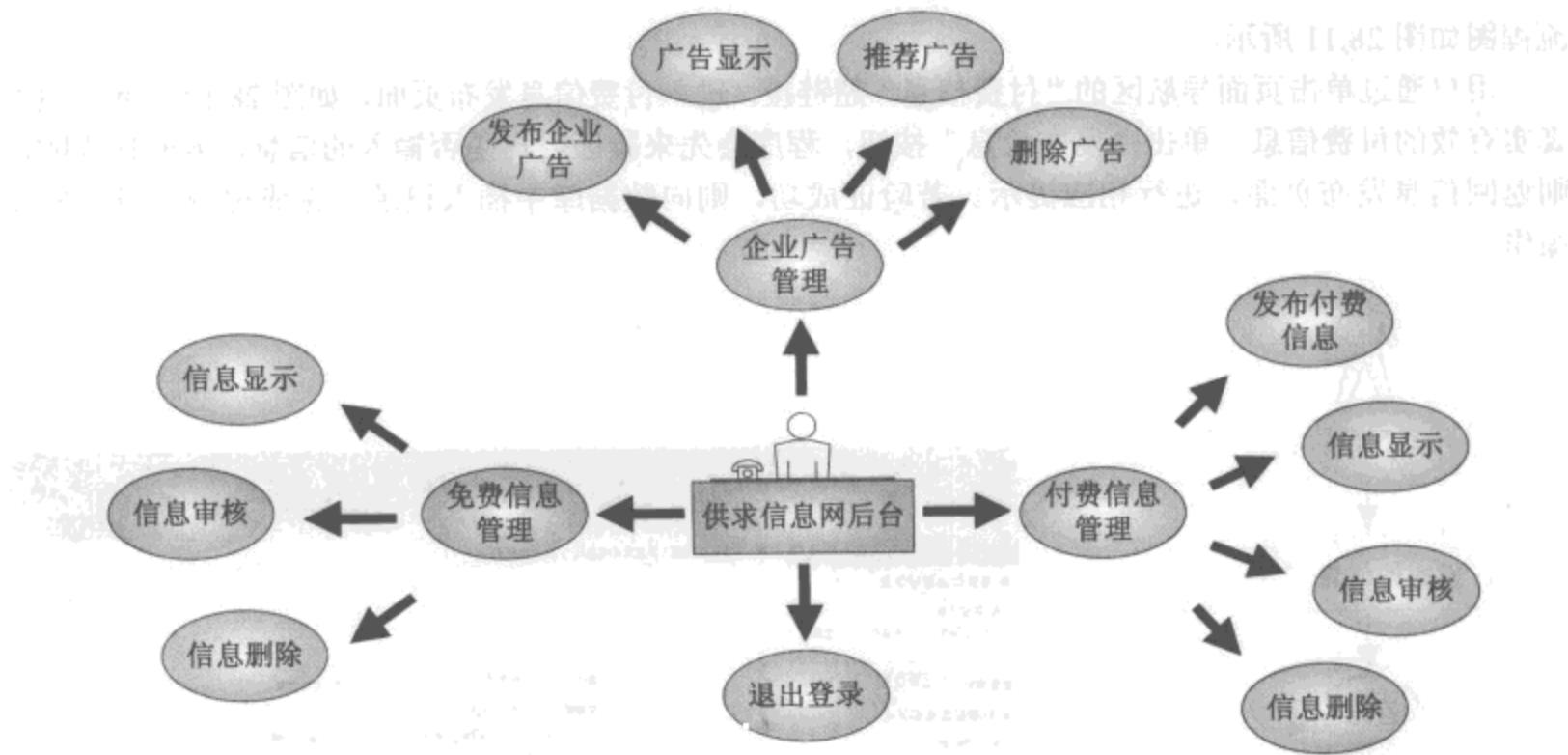


图 28.9 易查供求信息网后台功能结构图

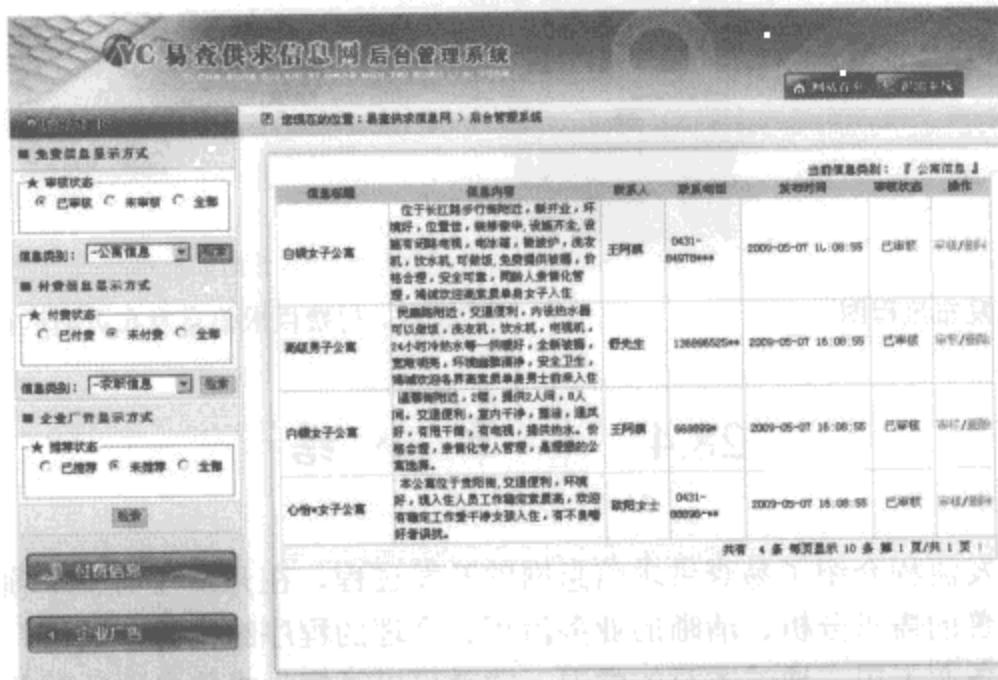


图 28.10 后台管理模块运行结果

### 28.3.6 付费供求信息发布模块设计

付费供求信息发布的提供对象为供求信息用户，是供求信息网站非常重要的功能，也是供求信息网站的盈利点。企业或用户可以根据自身需要对供求信息先进行付费，付费后由管理员在后台将供求信息发布到相应的信息类别中（招聘信息、求职信息、培训信息、公寓信息、家教信息、车辆信息、物品求购、物品出售、求兑出兑、寻求合作、企业广告）。供求信息成功发布后，管理员需要在后台对发布的供求进行审核，如果审核通过，则显示在前台相应的信息类别网页中。付费供求信息发布的

流程图如图 28.11 所示。

用户通过单击页面导航区的“付费信息”超链接，进入付费信息发布页面，如图 28.12 所示。填写真实有效的付费信息，单击“发布信息”按钮，程序会先来验证用户是否输入的信息，若验证失败，则返回信息发布页面，进行相应提示。若验证成功，则向数据库中插入记录，完成付费信息的发布操作。

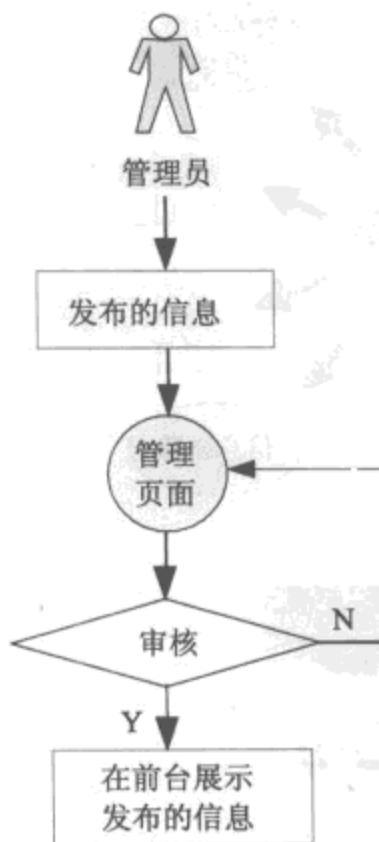


图 28.11 付费供求信息发布流程图

该截图显示了“YC 易查供求信息网后台管理系统”的付费信息发布界面。界面上有多个设置框，包括审核状态、信息类别、付费状态、信息内容等。下方有一个“发布信息”按钮。

图 28.12 付费供求信息发布页面运行结果

## 28.4 本章小结

本章依据软件开发流程介绍了易查供求信息网的开发过程。在开发任何一个项目前，首先要充分做好前期准备，如完善的需求分析、清晰的业务流程、合理的程序结构等，这样，在后期的程序开发中才会得心应手、有备而无患。通过本章的学习，读者可以了解供求信息网站的开发流程，熟悉框架技术在 Web 应用程序中的应用。由于篇幅所限，这里没有对该网站的开发进行详细讲解，其具体的功能可参考本书配套光盘中的源代码。

片忠对脚本文字的水印部分：0.8.85

# 第29章

## 图书借阅管理系统

在图书馆中，读者可能看到过这样一幅画面：很多人徘徊于图书馆书架之间，找寻着自己需要的图书；图书管理员则翻阅着图书借阅的记录，查看着哪些图书被借出，哪些需要归还，进行着繁琐的统计工作。而当计算机进入到我们的生活以后，可以通过它来管理图书馆中的图书，对图书进行登记、编号、分类；记录图书的借阅信息；查询图书馆中的图书，一切工作都变得那么简单、流畅。不但提高了工作的效率，而且使图书的查询变得方便、快捷，这就是图书借阅管理系统的魅力所在。本章将应用 Apache+PHP+MySQL 开发一个图书借阅管理系统，其中应用到的关键技术和方法如下：

- 通过 MySQL 函数连接数据库的方法
- 通过 PHP 函数操作数据库的方法
- 面向对象的编程方法
- 自动计算图书归还日期
- 管理员权限的设置方法
- MySQL 中添加、修改、删除和更新语句的应用
- MySQL 中多表查询方法
- MySQL 中内联接和外联接语句的应用

## 29.1 项目描述

知源图书馆是吉林 XX 师范学校的大型图书馆，随着学校图书馆规模的不断壮大，经营图书品种、数量也逐渐增多。同时，此图书馆常年采用的传统的人工方式管理暴露了一些问题。例如，查找读者借阅的某一本图书的具体摆放位置，需要靠人工记忆在书海中苦苦查找，由于图书储存量大，很难准确定位图书的具体位置，因此每天都要浪费大量的人力资源。

学校图书馆为提高工作效率，同时摆脱图书管理人员在工作中出现的种种弊端，现决定开发一个学校图书馆管理系统，通过计算机对图书进行管理。

图书馆管理系统为图书馆注入了新的生机，不但在运营过程中节省了大量的人力、物力、财力和时间，提高了图书馆的效率，而且还为图书馆在读者群中树立了一个全新的形象，为图书馆日后的发展奠定了一个良好的基础。

通过对一些大型图书馆的实际考察、分析，并结合图书馆的要求以及实际的市场调查，要求本系统具有以下功能：

- 系统页面设计美观大方、个性化，功能全面，操作简单。
- 实现基础信息的管理平台。
- 统计、管理读者信息。
- 统计图书借阅排行，了解当前的畅销书。
- 商品分类详尽，可按不同类别查看图书信息。
- 提供快速的图书信息、图书借阅检索功能，保证数据查询的灵活性。
- 实现图书借阅、图书续借、图书归还的功能。
- 实现综合条件查询，如按用户指定条件查询、按日期时间段查询、综合条件查询等。
- 实现对图书借阅、续借和归还过程的全程数据信息跟踪。
- 提供借阅到期提醒功能，使管理者可以及时了解到已经到达归还日期的图书借阅信息。
- 提供灵活、方便的权限设置功能，使整个系统的管理分工明确。
- 具备稳定的数据库系统。
- 安全性高，易于维护。

图书借阅管理系统主页的运行效果如图 29.1 所示。



图 29.1 图书借阅管理系统的主页

图书档案管理页面如图 29.2 所示。

图书借阅操作页面如图 29.3 所示。

图 29.2 图书档案管理页面

图 29.3 图书借阅操作页面

## 29.2 技术准备

在开发图书借阅管理系统时，该项目所使用的软件及开发环境如下。

### 服务器端

- 操作系统：Windows 2003 Server/Linux（推荐）。
- 服务器：Apache 2.2。
- PHP 软件：PHP 5.1.6。
- 数据库：MySQL 5.0.24。
- MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
- 开发工具：Dreamweaver 8。
- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

### 客户端

- 浏览器：IE 6.0 及以上版本。
- 分辨率：最佳效果 1024×768 像素。

## 29.3 开发过程

### 29.3.1 数据库设计

图书借阅管理系统的开发，数据库是必不可少的内容，需要通过数据库来存储图书档案数据、读者档案数据、图书借阅记录数据、图书归还记录数据和管理员数据。从开发的成本和用户需求考虑，

使用 MySQL 数据库是最为合适的选择。

结合实际情况及对用户需求的分析，图书借阅管理系统 db\_database29 数据库主要包含 10 个数据表，如图 29.4 所示。

表	类型	索引	说明
tb_bookcase	MyISAM	gb2312_chinese_ci	图书书架信息表
tb_holdinfo	MyISAM	gb2312_chinese_ci	图书信息表
tb_borrow	MyISAM	gb2312_chinese_ci	图书借阅信息表
tb_library	MyISAM	gb2312_chinese_ci	图书馆信息表
tb_manager	MyISAM	gb23f2_chinese_ci	管理员信息表
tb_parameter	MyISAM	gb2312_chinese_ci	参数设置信息表
tb_publishing	MyISAM	gb2312_chinese_ci	出版社信息表
tb_purview	MyISAM	gb2312_chinese_ci	权限信息表
tb_reader	MyISAM	gb2312_chinese_ci	读者信息表
tb_readertype	MyISAM	gb2312_chinese_ci	读者类型信息表

图 29.4 图书借阅管理系统数据表

### 29.3.2 功能模块概述

图书借阅管理系统主要分为 6 个部分，每个部分中包括多个具体的功能模块。图书借阅管理系统的整体功能结构如图 29.5 所示。

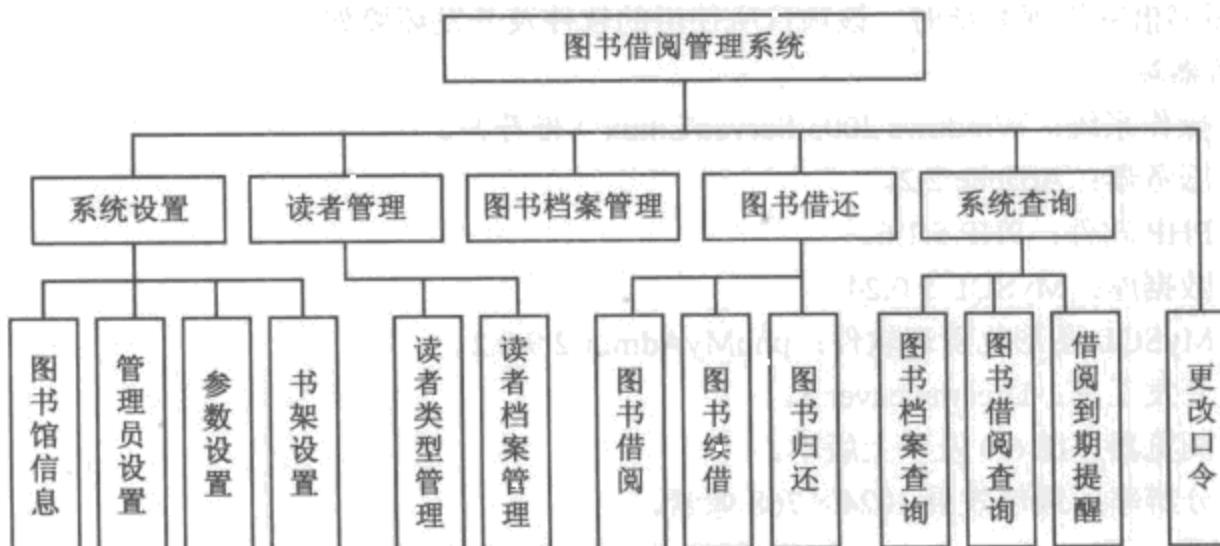


图 29.5 图书借阅管理系统整体功能结构图

### 29.3.3 管理员模块

管理员模块的功能包括管理员登录、查看管理员列表、添加管理员信息、管理员权限设置、管理员删除和更改口令等。管理员模块的框架图如图 29.6 所示。

首先是管理员登录，用于验证管理员的身份。运行本系统，首先进入的是管理员登录页面，在该页面中，系统管理员可以通过输入正确的管理员名称和密码登录到系统首页，当用户没有输入管理员名称或密码时，系统会通过 JavaScript 进行判断，并给予提示信息。管理员登录页面的运行结果如图 29.7 所示。

管理员登录后，选择“系统设置”/“管理员设置”命令，进入到查看管理员列表页面，页面中将以表格的形式显示全部管理员及其权限信息，并提供添加管理员信息、删除管理员信息和设置管理员权限的超链接。管理员列表页面的运行结果如图 29.8 所示。

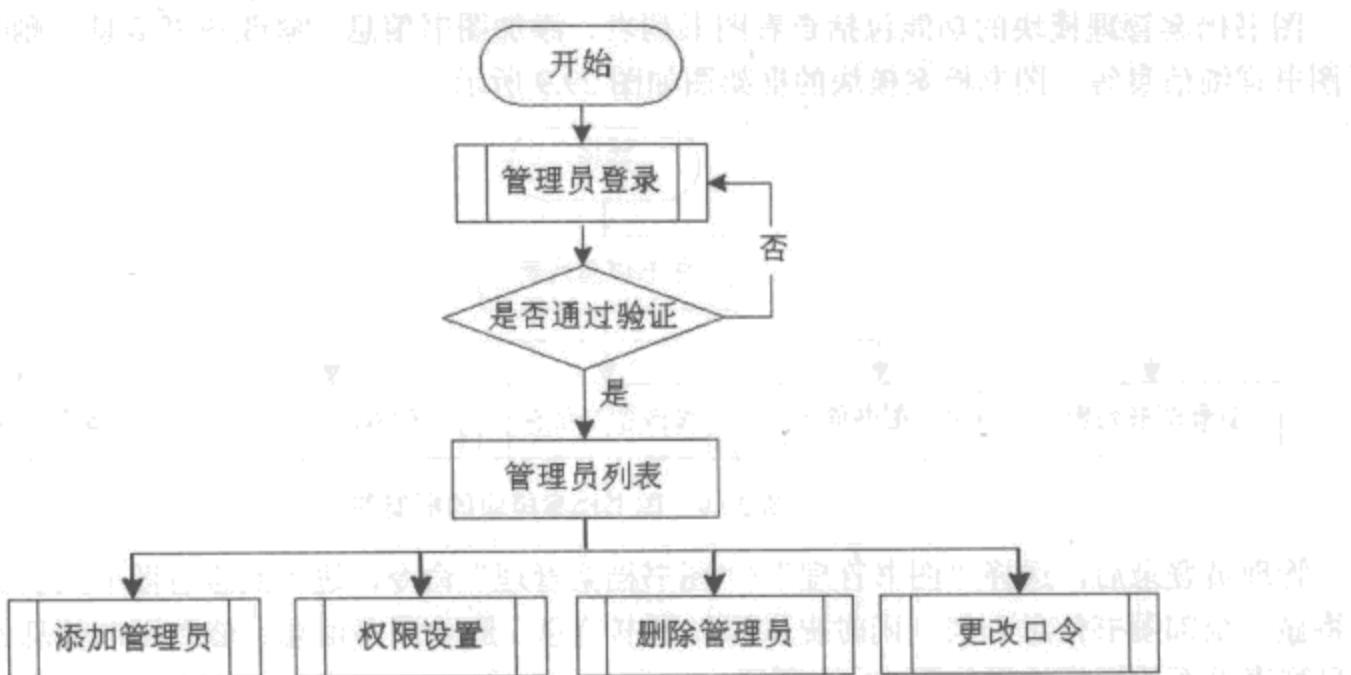


图 29.6 管理员模块的框架图



图 29.7 管理员登录页面的运行结果

该截图显示了管理员列表页面。上方有面包屑导航“当前位置：系统设置 > 管理员设置”。中间是一个管理员列表表单，包含以下数据：

管理员名称	系统设置	读者管理	图书档案管理	图书借还	系统查询	权限设置	操作
Tssoft	<input checked="" type="checkbox"/>	<a href="#">添加管理员信息</a>					
lx	<input checked="" type="checkbox"/>	<a href="#">权限设置</a>					
sl	<input checked="" type="checkbox"/>	<a href="#">删除</a>					

底部有版权信息“CopyRight © 2007 www.wrbccd.com 吉林\*\*\*师范大学图书馆”和提示“本站请使用IE 6.0 或以上版本 1024\*768 为最佳显示效果”。

图 29.8 管理员列表页面的运行结果

本书由人民邮电出版社授权华章公司统一出版，未经出版者书面许可，不得以任何形式对本书的全部或部分内容进行复制、传播、改编、翻译、再版等。

### 29.3.4 图书档案管理模块

图书档案管理模块的功能包括查看图书列表、添加图书信息、修改图书信息、删除图书信息和查看图书详细信息等。图书档案模块的框架图如图 29.9 所示。

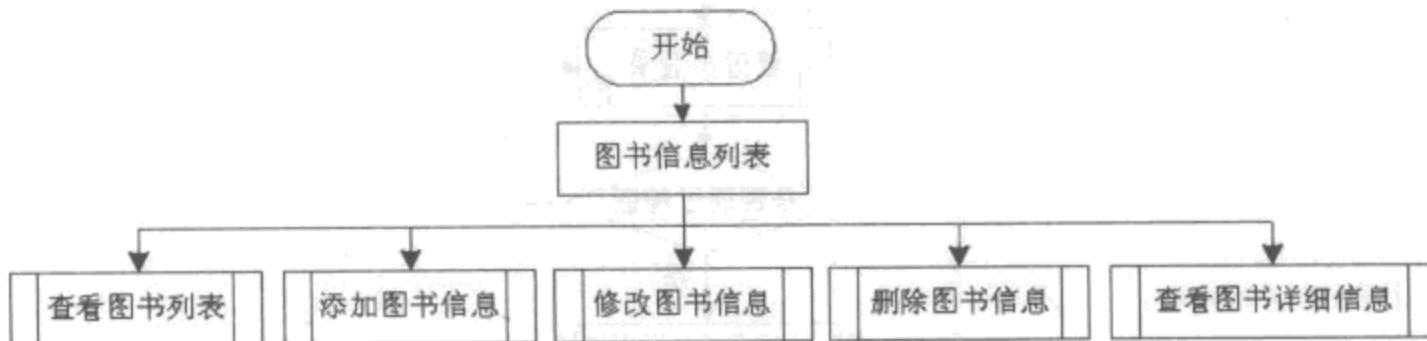


图 29.9 图书档案模块的框架图

管理员登录后，选择“图书管理” / “图书档案管理”命令，进入到查看图书列表页面，在该页面中将显示全部图书信息列表，同时提供添加图书信息、删除图书信息、修改图书信息的超链接。图书信息列表页面的运行结果如图 29.10 所示。

条形码	图书名称	图书类型	出版社	书架	操作	
					修改	删除
123456789	PHP 和 MySQL 开发完全手册	数据库技术	人民邮电出版社	PHP书架	修改	删除
123454321	PHP 程序开发范例宝典	宝典系列	人民邮电出版社	PHP书架	修改	删除
987654321	PHP 函数大全	信息科学技术	人民邮电出版社	PHP书架	修改	删除
9787115154101	Visual Basic 控件参考大全	计算机程序设计	人民邮电出版社	VB书架	修改	删除

图 29.10 图书信息列表的运行结果

### 29.3.5 图书借还模块

图书借还模块主要包括图书借阅、图书续借、图书归还、图书档案查询、图书借阅查询、借阅到期提醒等功能。在图书借阅模块中的用户只有一种身份，即操作员，通过该身份可以进行图书借还等相关操作。图书借还模块的用例图如图 29.11 所示。

管理员登录后，选择“图书借还” / “图书借阅”命令，进入到图书借阅页面，在该页面中的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的图书借阅信息。如果检索到对应的读者信息，将其显示在页面中，此时输入图书的条形码或图书名称后，单击“确定”按钮，即可借阅指定的图书。图书借阅页面的运行结果如图 29.12 所示。

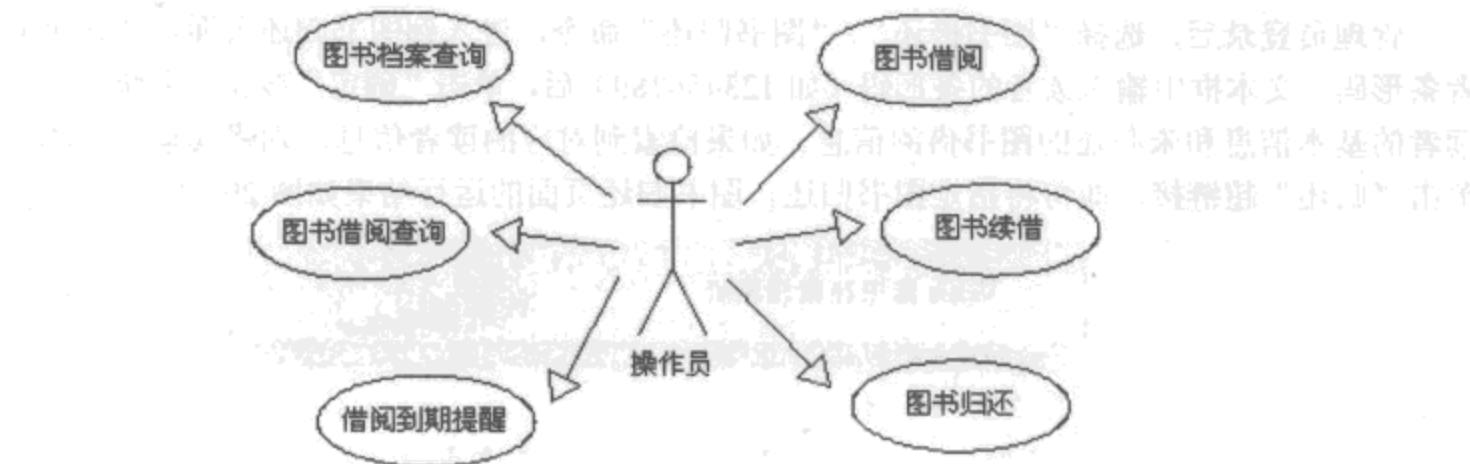


图 29.11 图书借还模块的用例图



图 29.12 图书借阅页面的运行结果

管理员登录后，选择“图书借还” / “图书续借”命令，进入到图书续借页面，在该页面中的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的图书借阅信息。如果检索到对应的读者信息，则将其显示在页面中，此时单击“续借”超链接，即可续借指定图书（将该图书的归还时间加上该书的可借天数得出）。图书续借页面的运行结果如图 29.13 所示。

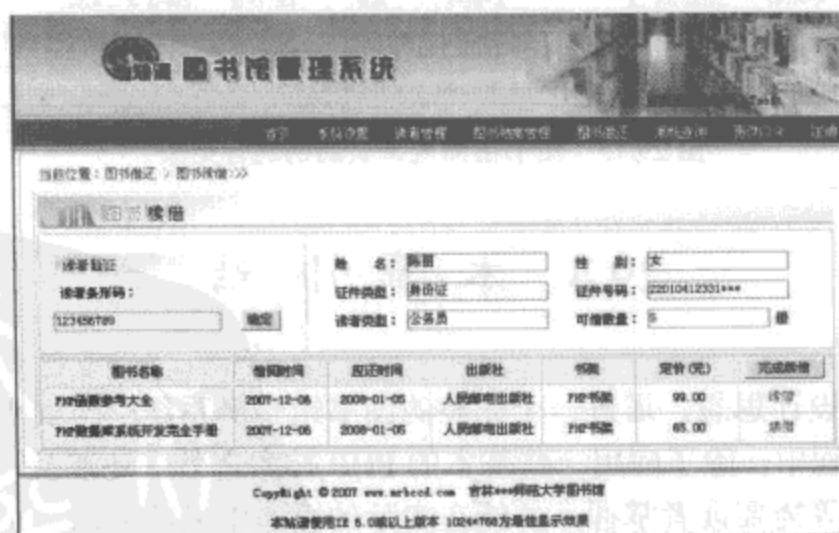


图 29.13 图书续借页面的运行结果

管理员登录后，选择“图书借还” / “图书归还”命令，进入到图书归还页面，在该页面中的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的图书借阅信息。如果检索到对应的读者信息，则将其输出到浏览器，此时单击“归还”超链接，即可将指定图书归还。图书归还页面的运行结果如图 29.14 所示。

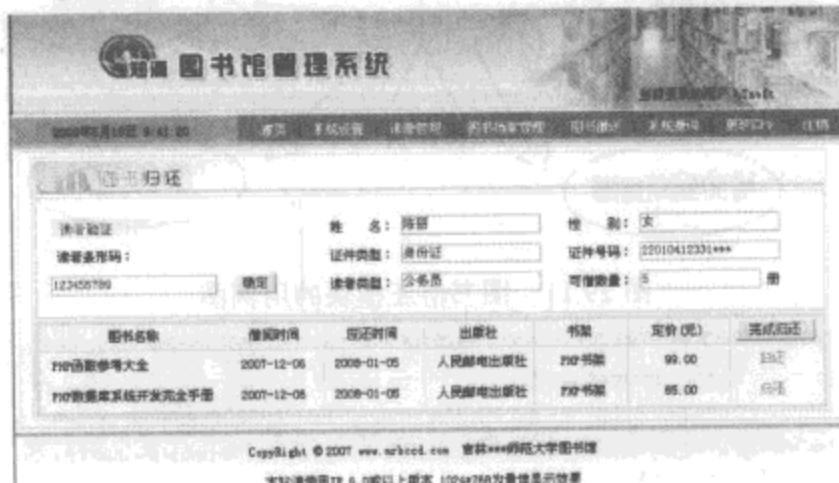


图 29.14 图书归还页面的运行结果

管理员登录后，选择“系统查询” / “图书借阅查询”命令，进入到图书借阅查询页面。图书借阅查询页面的运行结果如图 29.15 所示。在该页面中可以按指定的字段或某一时间段进行查询，同时还可实现按指定字段及时间段进行综合条件查询。

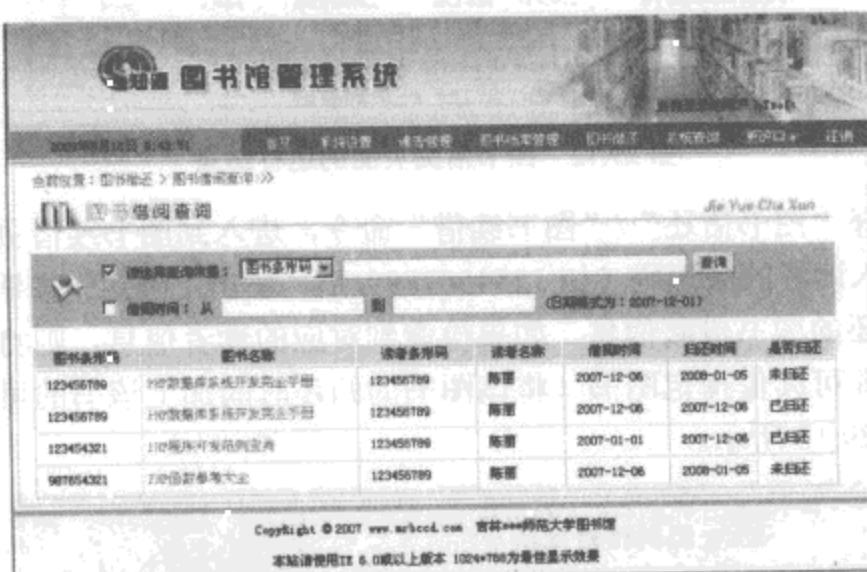


图 29.15 图书借阅查询页面的运行结果

## 29.4 本章小结

本章运用软件工程的设计思想，通过一个完整的图书馆管理系统引导读者深入了解系统的开发流程。在这个系统的实现过程中，除了应用一些基本的 PHP 技术之外，还涉及一些独特的技术细节，如权限设置、多表查询技术等需要读者掌握，并能在实际的操作中灵活应用，举一反三。本系统的详细代码可参考本书配套光盘中的内容。

# 附录 A PHP 编码规则

PHP 编码规则不是一个定理、一个公式，不需要 PHP 程序员去刻意遵循，但是如果 PHP 程序员遵循 PHP 编码规则，就可以提高代码的质量、程序的可维护性，及开发速度和效率。如果是团队开发项目，还可以让参与项目的其他开发人员更容易了解项目中的代码，弄清程序的状况，更快地投入到项目开发的工作中来。

如果是闭门造车，形成一套自己的编码风格，就会导致编写的东西只有自己才能看懂，其他人在阅读时会浪费过多的时间和精力，不易于对程序的维护、更新。如果是团队开发项目，就很难投入到项目的开发工作中，因为其他人不了解这个编码规则，也就不能快速地阅读相应的程序代码。

## A.1 PHP File 文件格式

### A.1.1 常规

对于只包含 PHP 代码的文件，结束标志（">"）是不允许存在的，否则会导致文件末尾被意外地注入空白并显示输出。

由 `_HALT_COMPILER()` 允许的任意的二进制代码的内容被 Zend Framework PHP 文件或由它们产生的文件禁止。这个功能只对特殊的安装脚本开放。

### A.1.2 缩进

使用 4 个空格的缩进，而不使用制表符 TAB。

### A.1.3 行的最大长度

一行在 80 个字符以内比较合适，长些也可以，但最多为 120 个字符。

### A.1.4 行结束标志

行结束标志只能是标准的 UNIX 文本文件的换行，换行符在文件中表示为 10，或十六进制的 0x0A。

不要使用 Macintosh 的回车，如（0x0D）。

不要使用 Windows 的回车换行组合，如（0x0D,0x0A）。

## A.1.5 引号

PHP 中单引号和双引号具有不同的含义，主要区别如下：

- 单引号中，任何变量（\$var）、特殊转义字符（如 “\t\r\n” 等）不会被解析，因此 PHP 的解析速度更快，转义字符仅支持 “\’” 和 “\\” 这样对单引号和反斜杠本身的转义。
- 双引号中，变量（\$var）值会代入字符串中，特殊转义字符也会被解析成特定的单个字符，还有一些专门针对上述两项特性的特殊转义，如 “\\$” 和 "{\$array[‘key’]}”。这样虽然程序编写更加方便，但 PHP 的解析也很慢。
- 数组中，如果下标不是整型，而是字符串类型，必须使用单引号将下标括起，正确的写法为 \$array['key']，而不是\$array[key]。不正确的写法将导致 PHP 解析器认为 key 是一个常量，进而先判断常量是否存在，不存在时才以 key 作为下标传入表达式中，同时触发错误事件，产生一条 Notice 级错误。

因此，在绝大多数可以使用单引号的场合，都禁止使用双引号。根据上述分析，应该或必须使用单引号的情况如下：

- 字符串为固定值，不包含 “\t” 等特殊转义字符。
- 数组的固定下标，如\$array['key']。
- 表达式中不需要传入变量，如\$string = 'string'，而非\$string = 'string\$var'。
- 还有一些特例，在正则表达式（preg\_系列函数和 ereg 系列函数）中全部使用双引号，这是为了人工分析和编写的方便，并保持正则表达式的统一，减少不必要的分析混淆。
- 数据库 SQL 语句中，所有数据都不得加单引号，但是在进行 SQL 查询之前必须经过 intval 函数处理；所有字符串都必须加单引号，以避免出现 SQL 注入漏洞。在 SQL 查询语句中的正确写法如下：

```
$user_id = intval($user_id);
select * from tb_user where user_name='$username' and user_id=$user_id;
```

- 缓存文件中，一般对缓存数据的值采用 addslashes(\$string, '\\') 进行转义。

## A.1.6 运算符、小括号、空格和关键词

每个运算符与参与运算的值或表达式之间要有一个空格，唯一的特例是字符连接运算符两边不加空格。

左括号 “(” 应和函数关键词紧贴在一起，除此以外，应当使用空格将 “(” 同前面内容分开。

右括号 “)” 除后面是 “)” 或者 “.” 以外，其他情况下一律用空格分隔。

除非字符串中有特殊情况，否则，在程序以及 HTML 中不允许出现两个连续的空格。

任何情况下，PHP 程序中不能出现空白的带有 TAB 或空格的行，即：这类空白行应当不包含任何 TAB 或空格。同时，任何程序行尾也不能出现多余的 TAB 或空格。多数编辑器具有自动去除行尾空格的功能，如果未养成良好的编程习惯，可临时使用它，以避免产生多余空格。

每段较大的程序体，上、下应当加入空白行，两个程序块之间只使用一个空行，禁止使用多行。

程序块划分尽量合理，过大或者过小的分割都会影响他人对代码的阅读和理解，一般以较大函数定义、逻辑结构、功能结构来进行划分。少于 15 行的程序块，可不加上下空白行。

说明或显示部分中，内容如中文、数字、英文单词混杂，应当在数字或者英文单词的前后加入空格。符合上述编码规范的示例代码如下：

```
$result = (( $a + 1 ) * 3 / 2 + $num) . 'Test';
$condition ? func1($var) : func2($var);
$condition ? $long_statement
: $another_long_statement;
if ($flag)
{
}

//Statements
//More than 15 lines
}
Showmessage( '请使用 restore.php 工具恢复数据。' );
```

## A.2 命名约定

### A.2.1 类

类名只允许有字母和数字字符，但不鼓励使用数字。下划线只允许作为路径分隔符，如 Zend/Db/Table.php 文件中对应的类名称是 Zend\_Db\_Table。

如果类名包含多个单词，每个单词的第一个字母必须大写，连续的大写是不允许的，例如，Zend\_PDF 是不允许的，而 Zend\_Pdf 是可接受的。

由 Zend 或其参与 Zend Framework 项目的伙伴公司发行的类必须以 Zend\_开头并且必须按等级顺序放在 Zend/目录下。

### A.2.2 接口

接口类也必须遵循同样的约定（如上所述），但必须以 Interface 结尾，例如：

```
Zend_Log_Adapter_Interface
Zend_Controller_Dispatcher_Interface
```

### A.2.3 文件名

对于其他文件，只有字母和数字字符、下划线和短横线“-”可用，空格是不允许的。

包含任何 PHP 代码的任何文件必须以.php 扩展名结尾。

普通程序和能够被 URL 直接调用的程序，如 list.php、index.php，可以直接使用“程序名+.php”

的方式命名。

函数库和类库程序，应以小写.func.php 和.class.php 作为扩展名。函数库和类库程序只能被其他程序引用，而不能独立运行。其中，不能包含任何流程性的、不属于任何函数或类的程序代码。

流程性程序，以小写.inc.php 作为扩展名。只能被其他程序引用，而不能独立运行。其中，不能包含任何函数或类代码的程序代码。

模板源文件，以小写.html 作为扩展名。模板源文件按照 PHP 模板编码规则进行编写，不是可执行的程序，而只能被 PHP 模板编译器所解析，放置于./templates/default 或./templates 目录下。

模板目标文件，模板文件被编译后自动生成的目标程序，以小写.php 作为扩展名，存放于./data/templates 目录下。

语言包文件，以小写.lang.php 作为扩展名，只能存放模板或程序使用的语言包信息。

缓存文件，此类文件为系统自动生成，以 cache\_xxx.php、usergroup\_xxx.php、style\_xxx.php 等类似形式命名，存放于./data/cache 目录下。

文件名必须遵循上述的对应类名的规则。

#### A.2.4 函数和方法

函数名只能包含字母和数字字符，但不鼓励使用数字，不允许使用下划线。

函数名总是以小写字母开头，当函数名包含多个单词时，每个单词的首字母必须大写，这就是所谓的“驼峰”格式。

鼓励使用冗长的名字，这样容易理解代码。下面是符合约定的命名：

```
filterInput()
getElementById()
widgetFactory()
```

对于面向对象编程，对象的访问器总是以 get 或 set 为前缀。当使用设计模式如单态模式(singleton)或工厂模式(factory)时，方法的名字应当包含模式的名字，这样容易从名字识别设计模式。

在对象中的方法，声明为 private 或 protected 的，名称的首字符必须是一个单独的下划线，这是唯一的下划线在方法名字中的用法。声明为 public 的方法不以下划线开头。

允许但不鼓励全局函数(floating functions)，建议把这类函数封装到静态类中。

以标准计算机英文为蓝本，尽量避免使用拼音或拼音英文混合的命名方式。

函数的名字和变量的命名规范一致。

函数定义中的左小括号与函数名之间无空格。

开始的左大括号另起一行。

具有默认值的参数应该位于参数列表的后面。

函数定义与调用的参数之间加入一个空格。

尽量避免函数起始缩进位置与结束缩进位置不同。

符合上述书写规则的函数定义方法如下：

```
function authcode($string, $operation, $key = "")  
{
```

```

if($flag)
{
//Statement
}
//函数体
}

```

## A.2.5 变量

变量只包含数字和字母字符，不鼓励使用数字，不允许为下划线。

声明为 `private` 或 `protected` 的类成员变量名必须以一个单独的下划线开头，这是唯一的下划线在变量名中的用法。声明为 `public` 的变量从不以下划线开头。

与函数名一样，变量名总以小写字母开头并遵循“驼峰式”命名约定。

鼓励使用冗长的名字，这样容易理解代码。除在小循环中以外，不鼓励使用简洁的名字（如`$i` 和 `$n`）。如果一个循环超过 20 行代码，索引的变量名必须是具有描述意义的名字。

变量命名只能使用项目中有据可查的英文缩写方式，例如，可以使用`$data` 而不可使用`$data1`、`$data2` 这样容易产生混淆的形式，应当使用`$username`、`$password` 这样能使人一目了然容易理解的形式。

可以合理地对过长的命名进行缩写，如`$bio($biography)`、`$tpp($threadsPerPage)`，前提是英文中存在这样既有的缩写形式，或字母符合英文缩写规范。

必须清楚所使用英文单词的词性，在权限相关的范围内，大多使用`$enable***`、`$is***` 的形式，前者后面接动词，后者后面接形容词。

下面是一些基本的命名规则：

- 名称的开头不能使用数字及特殊符号。
- “.” 及类型声明等专用语不能作为名称。
- 变量或程序名的长度必须在 255 个字符以内。
- 和保留字相同的名称不能使用。

## A.2.6 常量

常量包含数字、字母和下划线，数字允许作为常量名。

常量名的所有字母必须大写。

为加强可读性，常量中的单词必须以下划线分隔，例如，可以使用 `EMBED_SUPPRESS_EMBED_EXCEPTION`，但是不允许使用 `EMBED_SUPPRESSEMBEDEXCEPTION`。

常量必须通过 `const` 定义为类的成员，不鼓励使用 `define` 定义的全局常量。

## A.2.7 数据库表和字段

表和字段的命名以命名原则中定义的规范为依据。

所有数据表名称，只要其名称是可数名词，应尽量以复数方式命名；存储多项内容的字段，或代

表数量的字段，也应当以复数方式命名，如 hits（查看次数）、items（内容数量）。

当几个表间的字段有关联时，要注意表与表之间关联字段命名的统一，如 forum\_articles 表中的 articleid 与 forum\_restores 表中的 articleid。

代表 ID 自增量的字段，通常用以下几种形式：

- 一般情况下，使用全称的形式，如 userid、articleid。
- 没有功能性作用，只为管理和维护方便而设的 ID，可以使用全称的形式，也可只将其命名为 ID。

## A.3 编码风格

### A.3.1 PHP 代码划分 (Demarcation)

PHP 代码总是用完整的标准的 PHP 标签定界：

```
<?php  
?  
>
```

短标签（<? ?>）是不允许的，只包含 PHP 代码的文件不要结束标签。

### A.3.2 字符串

#### 1. 字符串文字

如果字符串是文字（不包含变量），则用单引号（apostrophe）引起：

```
$a = 'Example String';
```

#### 2. 包含单引号（'）的字符串文字

如果文字字符串包含单引号（apostrophe），就用双引号引起，特别是在 SQL 语句中：

```
$sql = "SELECT 'id', 'name' from 'people' WHERE 'name'='Fred' OR 'name'='Susan"';
```

在转义单引号时，上述语法是首选。

#### 3. 变量替换

变量替换有下面两种形式：

```
$greeting = "Hello $name, welcome back!";  
$greeting = "Hello {$name}, welcome back!";
```

为保持一致，下面的形式是不允许的：

```
$greeting = "Hello ${name}, welcome back!";
```

#### 4. 字符串连接

字符串用“.”操作符连接，在它的前后加上空格以提高可读性：

```
$project = 'PHP' . 'Zend';
```

当用“.”操作符连接字符串时，代码可以分成多个行，目的也是为了提高可读性。在这些例子中，每个连续的行应当由 whitespace 来填补，如“.”和“=”对齐：

```
$sql = "SELECT 'id', 'name' FROM 'tb_admin' "
    . "WHERE 'name' = 'Susan' "
    . "ORDER BY 'name' ASC ";
```

### A.3.3 数组

#### 1. 数字索引数组

索引不能为负数。

建议数组索引从 0 开始。

用 array 声明有索引的数组，在每个逗号的后面间隔空格以提高可读性：

```
$exampleArray = array(a, b, c, 'PHP', 'Studio');
```

也可以用 array 声明多行有索引的数组，在每个连续行的开头要用空格填补对齐：

```
$exampleArray = array(a, b, c, 'PHP', 'Studio',
    $a, $b, $c,
    56.44, $d, 500);
```

#### 2. 关联数组

用 array 声明关联数组，鼓励把代码分成多行，在每个连续行的开头用空格填补以便对齐键和值：

```
$exampleArray = array('firstKey' => 'firstValue',
    'secondKey' => 'secondValue');
```

### A.3.4 类

#### 1. 类的声明

用下面的约定来命名类：

- 花括号总是从类名下一行开始。
- 每个类必须有一个符合 PHP 编码标准的文档块。
- 4 个空格的缩进。
- 每个 PHP 文件中只有一个类。

在类中允许放置另外的代码，但不鼓励这样做。在这些文件中，通过两行空格来分隔类和其他代码。

码。示例如下：

```
/** * Documentation Block Here */
class ExampleClass
{
    // entire content of class
    // must be indented four spaces
}
```

## 2. 类成员变量

必须用下面的变量名约定来命名类成员变量：

- 变量的声明必须在类的顶部，要先于方法的声明。
- 不允许使用 var，应使用 private、protected 或 public。

### A.3.5 函数和方法

#### 1. 函数和方法声明

必须用下面的变量名约定来命名函数：

- 在类中的函数必须用 private、protected 或 public 声明其可见性。
- 和类一样，花括号从函数名的下一行开始，函数名与包括参数的圆括号中间没有空格。
- 不能使用全局函数。

示例如下：

```
/** * Documentation Block Here */
class Euspice
{
    /**
     * Documentation Block Here
     */
    public function cattle()
    {
        // entire content of function
        // must be indented four spaces
    }
}
```

#### 2. 函数和方法的用法

函数的参数用逗号和空格将紧接着的参数分隔。下面例子中的函数带有 3 个参数：

`threeArguments(1, 2, 3);`

传值方式在调用时是禁止的。

带有数组参数的函数，函数的调用可包括 array 并分成多行来提高可读性，同时，书写数组的标准仍然适用：

```
threeArguments(array(1, 2, 3), 2, 3);
threeArguments(array(1, 2, 3, 'Zend', 'Studio',
    $a, $b, $c,
    56.44, $d, 500), 2, 3);
```

### A.3.6 控制语句

#### 1. if/else/elseif

使用 if/else/elseif 的控制语句时，在条件语句的圆括号前后都必须有一个空格。

在圆括号中的条件语句，操作符必须用空格分开，鼓励使用多重圆括号以便在复杂的条件下划分逻辑组合。

前花括号必须和条件语句在同一行，后花括号单独在最后一行，其中的内容用 4 个空格缩进。

```
if ($a != 2) {
    $a = 2;
}
```

下面的例子应用 if、else 和 elseif，符合上述格式约定：

```
if ($a != 2) {
    $a = 2;
} else {
    $a = 7;
}
if ($a != 2) {
    $a = 2;
} elseif ($a == 3) {
    $a = 4;
} else {
    $a = 7;
}
```

在有些情况下，PHP 允许这些语句不用花括号，但在 ZF 代码标准里，if、elseif 或 else 语句必须使用花括号。

elseif 是允许的但强烈不鼓励使用，最好使用 else if 组合。

#### 2. switch

在 switch 结构的控制语句中，条件语句的圆括号前后都必须有一个单独的空格。

switch 中的代码必须有 4 个空格缩进，在 case 中的代码再缩进 4 个空格。

```
switch ($numPeople) {
    case 1:
```

```

        break;

    case 2:
        break;

    default:
        break;
}

```

switch 语句中必须有 default。

### A.3.7 数据库

#### 1. 字段结构

允许 NULL 值的字段，数据库在进行比较操作时，会先判断其是否为 NULL，非 NULL 时才进行值的比较。因此基于效率的考虑，所有字段均不能为空。

#### 2. SQL 语句

所有 SQL 语句中，除了表名、字段名称以外，全部语句和函数均需大写，应当杜绝小写形式或大小写混杂的写法，如“select \* from tb\_user;”是不符合规范的写法。

长 SQL 语句应当有适当的断行，依据 JOIN、FROM、ORDER BY 等关键字进行界定。

通常情况下，在对多表进行操作时，要根据不同表的名称，对每个表指定一个 1~2 个字母的缩写，以使语句简洁并提高可读性。下面的 SQL 语句是符合编码规范的：

```

$result = $db->query("Select m.*, i.*
FROM ".TABLE_MEMBER." m, ".TABLE_MEMBERINFO." i
Where m.userid=i.userid AND m.userid='$_userid'");

```

#### 3. 定长表与变长表

包含任何 varchar、text 等变长字段的数据表，即为变长表，反之则为定长表。

对于变长表，由于记录大小不同，在其上进行许多删除和更改将会使表中的碎片更多，需要定期运行 OPTIMIZE TABLE 以保持性能，而定长表就没有这个问题。

如果表中有可变长的字段，将它们转换为定长字段能够改进性能，因为定长记录易于处理。但在这样做之前，应该考虑下列问题：

- 使用定长列速度更快，但占用的空间更多。char(n)类型列的每个值总要占用 n 个字节（即使空串也是如此），因为在表中存储时，值的长度不够将在右边补空格。
- varchar(n)类型的列所占空间较少，因为只给它们分配存储每个值所需要的空间，每个值再加一个字节用于记录其长度。因此，如果在 char 和 varchar 类型之间进行选择，需要对时间与空间作出折衷。

变长表到定长表的转换，不能只转换一个可变长字段，必须对它们全部进行转换，而且必须使用一个 Alter TABLE 语句同时全部转换，否则转换将不起作用。

有时不能使用定长类型，如对于比 255 字符更长的字串就没有定长类型。

在设计表结构时如果能够使用定长数据类型尽量用定长的，因为定长表的查询、检索、更新速度都很快。必要时可以把部分关键的、承担频繁访问的表进行拆分。

进行表结构设计时，应当做到恰到好处、反复推敲，从而实现最优的数据存储体系。

#### 4. 结构优化与索引优化

索引能加快查询速度，而索引优化和查询优化是相辅相成的，既可以依据查询对索引进行优化，也可以依据现有索引对查询进行优化，这取决于修改查询或索引哪个对现有架构和效率的影响最小。

索引优化与查询优化的基本准则如下：

- ☒ 根据产品的实际运行和被访问情况，找出哪些 SQL 语句是最常被执行的。最常被执行和最常出现在程序中是完全不同的概念。最常被执行的 SQL 语句，又可被划分为对大表（数据条目多的）和对小表（数据条目少的）的操作。无论大表或小表，又可分为读（Select）多、写（Update/Insert）多或者读写都多的操作。

对常被执行的 SQL 语句而言，对大表操作需要注意：

- 写操作多的，通常可使用写入缓存的方法，先将需要写或需要更新的数据缓存至文件或其他表，定期对大表进行批量写操作。同时，应尽量使得常被读写的大表为定长类型，即使原本的结构中大表并非定长。大表定长化，可以通过改变数据存储结构和数据读取方式，将一个大表拆成一个读写多的定长表和一个读多写少的变长表来实现。
- 读操作多的，需要依据 SQL 查询频率设置专门针对高频 SQL 语句的索引和联合索引。

而小表就相对简单，加入符合查询要求的特定索引，通常效果比较明显。同时，定长化小表也有益于效率和负载能力的提高。字段比较少的小定长表，甚至可以不需要索引。

- ☒ 看 SQL 语句的条件和排序字段是否动态性很高（即根据不同功能开关或属性，SQL 查询条件和排序字段的变化很大的情况），动态性过高的 SQL 语句是无法通过索引进行优化的。唯一的办法只有将数据缓存起来，定期更新，这适用于结果对实效性要求不高的操作。

事实上，索引是将条件查询、排序的读操作的资源消耗分布到了写操作中，索引越多，耗费磁盘空间越大，写操作越慢。因此，决不能盲目添加索引。对字段索引与否，最根本的依据是 SQL 语句执行的概率、表的大小和写操作的频繁程度。

#### A.3.8 注释文档

程序注释中要用简单、明了、精确的语言表达出程序的处理说明。应避免每行程序都使用注释，可以在一段程序的前面加一段注释，要具有明确的处理逻辑。注释虽然必不可少，但也不应过多，不要被动地为写注释而写注释。

##### 1. 文件

每个包含 PHP 代码的文件必须至少在文件顶部包含这些 phpDocumentor 标签：

```
/**
 * 文件的简短描述
 *
 * 文件的详细描述（如果有的话）...
 *
```

```
* LICENSE: 一些 license 信息
*
* @copyright 2005 Zend Technologies
* @license http://www.zend.com/license/3_0.txt PHP License 3.0
* @version $Id:$
* @link http://dev.zend.com/package/PackageName
* @since File available since Release 1.2.0
*/

```

## 2. 类

每个类必须至少包含这些 phpDocumentor 标签:

```
/**
 * 类的简述
 *
 * 类的详细描述 (如果有的话) .....
 *
 * @copyright 2005 Zend Technologies
 * @license http://www.zend.com/license/3_0.txt PHP License 3.0
 * @version Release: @package_version@
 * @link http://dev.zend.com/package/PackageName
 * @since Class available since Release 1.2.0
 * @deprecated Class deprecated in Release 2.0.0
*/

```

## 3. 函数

每个函数，包括对象方法，必须有最少包含下列内容的文档块 (docblock) :

- 函数的描述。
- 所有参数。
- 所有可能的返回值。

因为访问级已经通过 public、private 或 protected 声明，所以不需要使用@access。

如果函数/方法抛出一个异常，使用@throws:

```
@throws exceptionclass [description]
```

## 4. 块注释

块注释通常用于提供对文件、方法、数据结构和算法的描述。块注释被置于每个文件的开始处以及每个方法之前，也可以被用于其他地方，如方法内部。在功能和方法内部的块注释应该和它们所描述的代码具有一样的缩进格式。

块注释起始处应该有一个空行，用于把块注释和代码分隔开，例如:

```
/*
 * 这里是块注释
*/

```

## 5. 单行注释

短注释可以显示在一行内，并与其后的代码具有一样的缩进层级。如果一个注释不能在一行内写完，就应采用块注释。单行注释之前应该有一个空行。以下是一个代码中单行注释的例子：

```
If (condition) {  
    /* 以下代码运行的条件 */  
    ...  
}
```

## 6. 尾端注释

极短的注释可以与其所要描述的代码位于同一行，但是应该有足够的空白来分开代码和注释。若多个短注释出现于大段代码中，它们应该具有相同的缩进。

以下是一个代码中尾端注释的例子：

```
If ($a == 2) {  
    return TRUE; /* 对单一条件的说明 */  
} Else {  
    return isPrime($a); /* 其余的条件 */  
}
```

## 7. 行末注释

注释界定符“//”，可用于注释整行或者一行中的一部分。

```
If ($Euspice > 1) {  
    ... //注释用法  
}
```