

网络编程实战丛书

- 网络编程人员的参考指南
- 资深数据库编程人员理想的参考书

MySQL

网站架设高手

马 冰 洪 海 编著

科学出版社

网络编程实战丛书

MySQL 网站架设高手

马冰 洪海 编著

科学出版社

2001

内 容 简 介

MySQL 和 PHP 是两个风靡全球的开发数据库应用程序的免费软件，越来越多的网站使用它们架设自己的中小型网络数据库服务器。本书通过对四个大型实用例子具体的功能模块的所有源代码的剖析，帮助读者迅速掌握 PHP 与 MySQL 数据库的交互技巧，以及开发一个完整应用程序的过程，是一本数据库开发入门与提高的实用教材。

本书语言简洁，由浅入深，可作为广大网络编程员的参考指南，也可作为资深数据库编程人员理想的参考书。

图书在版编目(CIP)数据

MySQL 网站架设高手/马冰，洪海编著. -北京：科学出版社，2001

(网络编程实战丛书)

ISBN 7-03-009434-4

I.M… II.马… III.①关系数据库—数据库管理系统，MySQL②PHP 语言—程序设计 IV.TP393.09

中国版本图书馆 CIP 数据核字(2001)第 038132 号

科学出版社 出版

北京东黄城根北街 16 号
邮政编码：100717

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2001 年 8 月第 一 版 开本：787×1092 1/16

2001 年 8 月第一次印刷 印张：16 3/4

印数：1—5 000 字数：382 000

定价：21.00 元

(如有印装质量问题，我社负责调换〈环伟〉)

本书编委会

主 任：罗特军

副主任：王 松

委 员： 孙 忠 刘小伟 邓 勇 欧阳劲
胡 杨 刘青松 况 山 孙 韬
邓红梅 孙吉祥 袁 辉 张鸣歧
陈 明 文黎敏 钱晓捷 陈 涛
黄 磊 周 胜 宋 浩 袁 军

前言

随着近几年互联网规模的飞速发展，各类网站提供的服务类型也越来越丰富，例如电子邮件服务、FTP 服务、WWW 服务等，而电子商务在我国正是方兴未艾，充满无限商机之时，如何才能跟上网络时代的浪潮，抢占信息时代的商机呢？

在 Internet 上，任何一个网站在其丰富强大的功能背后，都有一整套稳定高效的服务器应用程序维护着它的运行，完成客户端与服务器端之间的各种交互动作。在自由软件旋风席卷全球的今天，涌现出大批优秀的开放源代码的软件，其中包括 MySQL 的 Linux/Unix 版本。本书正是以 MySQL 为数据库开发工具，以 PHP 为编程语言，向您介绍如何开发各类网络数据库应用程序，并以四个实用性极强的典型实例，向您展示建设一个多功能网站的全过程。

本书特点

- 实用性强，所有源程序均经过运行测试，可直接上机使用。
- 内容丰富，涵盖了网络数据库应用的多种类型。
- 重点突出，以丰富的程序实例为主，使您开发网络数据库的思路更广阔。

适合的读者

本书适合于：

- 希望快速架设一个多功能网站的读者。
- 对网络数据库开发有兴趣的读者。
- 已经掌握了 PHP 或 MySQL 的基础知识，但还没有实际开发经验的读者。
- 希望进一步扩展网站功能的网站建设者。

本书结构

本书共分为 6 章：

第 1 章——介绍网站服务器软件平台的构建，即 MySQL+PHP+Apache 这一黄金组合的安装和配置，以及使用 PHP 访问 MySQL 数据库的基本知识。

第 2 章——介绍 MySQL 与 PHP 的交互。

第 3 章——实例一：在线学生学籍管理系统。

第 4 章——实例二：在线音乐站点及其管理系统。

第 5 章——实例三：股票交易模拟系统。

第 6 章——实例四：在线电脑超市及管理系统。

本书由博嘉科技资讯有限公司主编。由于本书编写时间仓促，作者知识面局限，书中难免有错误和疏漏之处，恳请读者批评指正。如有意见或建议，欢迎与我们联系，我们的通讯地址是：四川大学西区建筑学院成都博嘉科技资讯有限公司，邮编：610065。电话：(028) 5404228，E-mail: bojiakeji@163.net。

编者

目 录

第 1 章 架站准备	1
1.1 构建软件平台	1
1.1.1 安装软件	2
1.1.2 服务器安全配置	5
第 2 章 MySQL 与 PHP 的交互	7
2.1 PHP 程序设计基础	7
2.1.1 PHP 语言简介	7
2.1.2 PHP 脚本的编写、运行和调试	11
2.2 如何使用 MySQL	13
2.2.1 MySQL 介绍	13
2.2.2 联接和断开服务器	14
2.2.3 输入查询	15
2.2.4 常用查询的例子 (1)	18
2.2.5 常用查询的例子 (2)	22
2.2.6 获得数据库和表的信息	40
2.2.7 MySQL 中的列类型及其选择	41
2.2.8 列的属性描述	44
2.3 MySQL 与 PHP 的交互	46
2.3.1 连接管理函数	47
2.3.2 选择数据库函数	48
2.3.3 构造查询函数	49
2.3.4 结果集处理函数	49
2.3.5 查询和结果处理实例	50
第 3 章 在线学生学籍管理系统	56
3.1 在线学生学籍查询系统	56
3.1.1 查询系统的规划	56
3.1.2 建立整个系统所需的数据库	57
3.1.3 查询系统页面的组成	62
3.1.4 查询系统源程序	63
3.2 在线学生学籍管理系统	71

3.2.1 查询系统的规划.....	71
3.2.2 管理系统页面的组成.....	71
3.2.3 管理系统源程序.....	73
第 4 章 在线音乐站点及其管理系统.....	92
4.1 在线音乐站点.....	92
4.1.1 音乐站点总体规划.....	92
4.1.2 建立整个站点所需的数据库.....	93
4.1.3 音乐站点页面及程序的组成.....	95
4.1.4 站点的源程序.....	98
4.2 音乐站点管理系统.....	125
4.2.1 站点管理系统的规划.....	126
4.2.2 站点管理系统页面的组成.....	126
4.2.3 站点管理系统源程序.....	128
第 5 章 股票交易模拟系统.....	150
5.1 模拟系统自身.....	150
5.1.1 股票模拟系统总体规划.....	150
5.1.2 建立系统所需的数据库.....	151
5.1.3 股票交易系统源程序.....	153
5.2 系统管理部分.....	190
5.2.1 管理系统的功能.....	191
5.2.2 股票管理程序.....	191
第 6 章 在线电脑超市及管理系统.....	210
6.1 在线电脑超市.....	210
6.1.1 超市的总体规划.....	210
6.1.2 建立站点所需的数据库.....	211
6.1.3 站点页面及程序的组成.....	213
6.1.4 在线超市的源程序.....	216
6.2 站点管理系统.....	235
6.2.1 超市管理系统的规划.....	235
6.2.2 站点管理系统页面的组成.....	236
6.2.3 超市管理系统源程序.....	239

第 1 章 架站准备

MySQL 数据库具有优越的性能和广阔的市场发展前景。如今,包括 Siemens 和 Silicon Graphics 这样的国际知名公司也开始把 MySQL 作为其数据库管理系统,越来越多的商业站点把 MySQL 作为后台数据库。

本书主要向广大读者介绍如何将 MySQL 应用于 Web 站点的建设之中。当然,我们仍然需要循序渐进、一步一步地来学习利用 MySQL 建立站点所需的各方面的知识。首先需要做的就是安装和配置服务器,这就是本章的主要内容。

本章要介绍的内容主要分为两大部分:

- 服务器软件平台的安装与配置
- PHP 程序设计基础以及与 MySQL 数据库交互的技巧

通过这一章的学习,读者将能够建立起一个稳定、安全、高效的服务器系统,并掌握学习本书必备的基础知识。由于 Linux 操作系统能够运行在大多数硬件体系中,因此本书对服务器硬件设施的准备不做深入讨论。

1.1 构建软件平台

建设网站除了要准备必要的硬件设备之外,最重要的是首先要考虑软件平台的安装和配置,它往往决定了一个网站的建设成本以及运行时的稳定性和安全性。由于 Linux+MySQL+PHP+Apache 这个组合几乎是完全免费的,并且以其出色的稳定性和安全性征服了无数向往自由的网站建设者,因此国内外很多网站都收集了这一系列软件的不同版本,可以利用网络获得这些软件。

除了从网上下载软件,在很多 Linux 发行版本的配套光盘中,都有 RPM 形式的软件包,它们可以利用 Red Hat 和 Caldera 公司提供的包管理器(RPM)进行安装、删除和检查。常用的命令如下:

```
shell> rpm -i pack_name
shell> rpm -e pack_name
shell> rpm -q pack_name
```

其中-i、-e、-q 三个选项分别表示安装、删除和显示某软件包。下面以安装 Apache 为例,先介绍如何使用 rpm 进行软件管理。

首先要确定系统未装有 Apache。因为如果在安装 Linux 的时候选择了安装 Web 服务器或是完全安装,那么 Apache 可能已经安装好了。要确定系统中是否已安装了 Apache 的

RPM 包，使用如下命令：

```
shell> rpm -q apache
```

若结果显示了一个带有版本号的 **apache**，则说明已经安装了 **Apache**，否则说明还没有安装 **Apache**，可以按照下面的步骤进行安装。

(1) 先装载光驱到 Linux 系统中。

```
# mount /dev/ cdrom / mnt/ cdrom
```

(2) 进入 Red Hat Linux 软件包目录。

```
shell> cd / mnt/ cdrom/ RedHat / RPMS
```

(3) 在此目录下查找是否有 **Apache** 的软件包。

```
shell> ls | grep apache
```

```
apache-1.3.6-7.i386.rpm
```

```
apache-devel-1.3.6-7.i386.rpm
```

(4) 安装 **Apache**。

```
shell> rpm -i apache-1.3.6-7.i386.rpm
```

至此，安装工作就结束了，若以后不再需要 **Apache**，只需执行：

```
shell> rpm -e apache
```

即可卸载 **Apache** 软件包。**MySQL** 和 **PHP** 的 **RPM** 软件包的安装也类似于上述过程。

还有一种配置较为灵活的源代码发布形式，也是本系统使用的软件：

```
mysql-3.22.32.tar.gz
```

```
php-4.0.2.tar.gz
```

```
apache_1.3.12.tar.gz
```

这类软件的编译、安装过程比 **RPM** 形式复杂，将在下面的小节中详细讲述。在此之前，要完成这类软件的安装，系统中还需装有 **gcc** 以及 **make** 等编译工具。

1.1.1 安装软件

下面具体介绍一下源代码形式的 **MySQL+PHP4+Apache** 在 Linux 上的安装过程。

1. 安装 MySQL

首先安装 **MySQL**，它可以作为一个独立的软件安装到服务器上。

(1) 将下载的 **mysql-3.22.32.tar.gz** 文件解压缩还原，若文件所在的目录为 **/root**，则这一步将自动在 **/root** 目录下建立一名为 **mysql-3.22.32** 的子目录。

```
shell> tar zxvf mysql-3.22.32.tar.gz
```

(2) 进入 **mysql-3.22.32** 目录。

```
shell> cd mysql-3.22.32
```

(3) 配置 **MySQL** 安装。**configure** 前的 “./” 是强调执行当前目录下的 **configure** 文件。**--with-charset=gb2312** 选项是使 **MySQL** 支持中文 **gb2312** 字符集，因为本系统中部分用户资料和股票名称需要使用中文。

```
shell> ./configure --with-charset=gb2312
```

(4) 编译 MySQL 源文件。

```
shell> make
```

(5) 安装编译后的文件。

```
shell> make install
```

(6) 进入 bin 目录。

```
shell> cd /usr/local/bin
```

(7) 执行安装脚本。

```
shell> ./mysql_install_db
```

这样 MySQL 就安装完成了, 下面可以运行试一试。

(8) 首先利用 `safe_mysqld` 程序启动 `mysqld` 服务器进程。其中 “&” 表示将这一进程推向后台。

```
shell> /usr/local/bin/safe_mysqld &
```

另一种方法适用于 System V 风格的 Unix 系统, 即通过调用 `mysql.server` 启动脚本来实现。方法为:

```
shell> /usr/local/share/mysql/mysql.server start
```

但首先要使此脚本可执行, 还需设置 `mysql.server` 文件的执行权限。即执行如下命令:

```
shell> chmod +x mysql.server
```

(9) 初次运行 MySQL。

```
shell> /usr/local/bin/mysql -u root
```

如果在几行欢迎信息和版本号之后看到 “`mysql>`” 提示符就说明已安装成功。如果在配置、编译过程中出现一些错误提示而无法进行, 则可能是您的系统缺少一些库函数或文件包。这时不必沮丧, 先根据显示的错误信息找到缺少的文件名, 然后在您的 Linux 安装光盘上查找相应的文件, 将它们安装到系统中, 即可继续编译。

2. 安装 Apache 和 PHP4

接下来安装 Apache 和 PHP4, 这两个软件需要一起编译安装。

(1) 建立 Apache 的安装目录。

```
shell> mkdir /usr/local/apache
```

(2) 将压缩文件 `apache_1.3.12.tar.gz` 解压缩还原, 自动在 `/root` 目录下建立 `apache_1.3.12` 的子目录。

```
shell> tar zxvf apache_1.3.12.tar.gz
```

(3) 同上, 在 `/root` 目录下建立 `php-4.0.2` 的子目录。

```
shell> tar zxvf php-4.0.2.tar.gz
```

(4) 进入 Apache 原始目录。

```
shell> cd apache_1.3.12
```

(5) 配置安装目录。用 `--prefix` 选项指定了路径后, Apache 将安装在这个目录下。

```
shell> ./configure --prefix=/usr/local/apache
```

(6) 进入 PHP 的原始目录。

```
shell> cd ../php-4.0.2
```

(7) 配置 PHP 的安装。由于本系统需要 MySQL 数据库, 因此要加上--with-mysql 选项和--with-apache 选项, 还要保证 Apache 原始目录的路径要正确。如果您放置 Apache 原始目录的位置与上面所说的不同, 那么在这里就要做相应的变化。--with-apache 选项是将 PHP 编译为 Apache 的模块。

```
shell> ./configure --with-apache = ../apache_1.3.12 --with-mysql \  
-enable-ftp -enable-track-vars
```

(8) 编译 PHP 源文件。

```
shell> make
```

(9) 安装已编译的文件。

```
shell> make install
```

(10) 复制 php.ini 到/usr/local/lib/目录下。

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

(11) 进入 Apache_1.3.12 原始目录。

```
shell> cd ../apache_1.3.12
```

(12) 配置 Apache 安装, 使安装程序加载 PHP 模块。

```
shell> ./configure --prefix=/usr/local/apache \  
-activate-module=src/modules/php4/libphp4.a
```

(13) 编译 Apache 源文件。

```
shell> make
```

(14) 安装已编译的文件。

```
shell> make install
```

至此, 安装工作已经完成, 下面需要修改一些配置文件。

(15) 进入 Apache 的配置文件所在的目录。

```
shell> cd /usr/local/apache/conf
```

(16) 启动 vi 编辑器, 编辑 httpd.conf 文件。

```
shell> vi httpd.conf
```

将 AddType application/x-httpd-php.php 一行前面的“#”去掉, 可以选择再加入一行:

```
AddType application/x-httpd-php.php4
```

这是使 Apache 服务器能够识别并执行所有后缀名为 php, php4 的文件, 当然后缀名也可以在此设成其他的名称。比如某些网站将 php 程序的后缀名设为 phtml, 这只是名称上的不同, 本质都是一样的。关于 Apache 的高级配置还有很多复杂的参数, 这里只做了必要的基本设置, 可根据您的实际情况参考其他介绍 Apache 服务器配置的资料。

(17) 接下来试着启动 Apache 服务器, 进入/usr/local/apache/bin 目录。

```
shell> cd ../bin
```

(18) 启动 Apache 服务器。

```
shell> ./apachectl start
```

这时应出现一行“/usr/local/apache/bin/apachectl start: httpd started”, 提示 httpd 进程已经在后台运行。到这里服务器的安装就结束了, 下面要做的是试验安装好的 Web 服务器能否运行 PHP 程序。现在可以在 Apache 的默认文件目录/usr/local/apache/htdocs 下建立一个

简单的 PHP 文件 `test.php`，内容如下：

```
<html>
<head>
<title>My first PHP program</title>
</head>
<body>
<?php
echo "hello, world~";
?>
</body>
</html>
```

由于 PHP 程序是运行在服务器端的，这使得无论 PHP 程序多么复杂庞大，都不会影响客户端的运行速度。同时，在查看 PHP 程序的执行结果时，也需要转到客户端。然后，启动 X window 图形方式，或是在另一台与服务器相连接的计算机上，试着访问一下这个 `test.php`。在客户端浏览器的地址栏中输入“`http://服务器的 IP 地址/test.php`”的形式，例如服务器的 IP 地址为 10.14.1.4，TCP/IP 端口号为默认值，则输入“`http://10.14.1.4/test.php`”，应该可以看到“hello, world~”字样了，这表明 Apache+PHP4+MySQL 已成功地安装到您的系统上。

在调试 PHP 程序的过程中，有时需要关闭服务器，拒绝客户端的请求。这是因为，执行一个没有调试成功的 PHP 程序或者服务器配置不当都有可能造成将程序文本的内容输出到客户端的浏览器页面上，如果程序中包含了连接数据库的密码，也就等于将密码公布于众，这是十分危险的。要停止服务器的运行，可执行 Apache 和 MySQL 提供的控制命令。

Apache 的启停控制命令为 `/usr/local/apache/bin` 目录下的 `apachectl` 命令。通过它，可以启动、停止和重新启动 `httpd` 进程，这三个命令分别为：

```
# /usr/local/apache/bin/apachectl start
# /usr/local/apache/bin/apachectl stop
# /usr/local/apache/bin/apachectl restart
```

MySQL 的停止命令为：

```
# /usr/local/bin/mysqladmin shutdown -u root -p
```

1.1.2 服务器安全配置

对于一个网络站点而言，安全的重要性甚至远远超过了稳定和高效。

1. 为 MySQL 的 root 帐号设置密码

在安装完 MySQL 之后，需要尽快为 MySQL 的 root 帐号设置密码，因为最初的 MySQL 用户都没有密码，任何人无需密码就能以 root 帐号登录到 MySQL，并拥有最高权限。要设置 root 用户的密码，可使用两种较简单的方法：

(1) 使用 `mysqladmin` 程序。例如：

```
shell> /usr/local/bin/mysqladmin -u root password "root's password"
```

(2) 使用 `mysql` 程序。例如:

```
shell> /usr/local/bin/mysql -u root
mysql> update user set Password=PASSWORD("root's password")
-> where User="root";
```

这时, 若要密码设置立即生效, 还需刷新授权表, 即执行:

```
shell> /usr/local/bin/mysqladmin -u root reload
```

则以后以 `root` 身份执行任何 MySQL 程序, 均需要键入密码进行身份验证。

2. 配置文件 `httpd.conf`

在配置 Apache 时, 主要的配置文件是 `httpd.conf`, 它定义了 HTTP 协议的各种参数值, 以及对该服务器的操作说明等。其中几个涉及到服务器安全的参数包括:

(1) `ServerType`

设定系统启动服务器的方式。可选参数有两个: `inetd` 和 `standalone`。

若选择 `inetd`, 则服务器进程的运行将由系统程序 `inetd` 承担。因此要使 Apache 服务器在每次系统启动时自动运行, 需将其启动控制命令放在 `/etc/inetd.conf` 文件里。另一种 `standalone` 的启动方式是以守护进程 (`daemon`) 的方式运行服务器, 这时启动服务器的命令需要放在 `/etc/rc.local` 中。两种方式各有优缺点, 使用 `inetd` 运行服务器时, 对每一个 http 连接请求, 系统都要重新执行一次服务器的拷贝进程, 而当断开该连接后, 这个拷贝进程也随之结束运行。这样一来, 降低了服务器被攻击的可能性, 但同时也降低了服务器的执行效率, 是一种牺牲时间换取安全的选择。一般来说, `standalone` 是 `ServerType` 最普遍的选择, 也是 Apache 安装后的默认选项。这种方式下只需启动一次服务器进程, 就可以在后台监听所有的连线请求, 因此执行效率比 `inetd` 方式高。

(2) `User`

这个参数设置了服务器进程以哪个用户的身份执行。确定不要将 `User` 设为 `root`, 否则以 `root` 身份处理客户端的连接请求, 将使系统容易受到攻击。

3. 设置包含文件目录

为了避免由于脚本语言编写错误或服务器配置不当而暴露任何不能透露的信息, 应当使用包含文件, 将保密信息包含在特殊类型的文件中, 并存放在 Apache 的文件目录之外, 这样当 Apache 服务器配置出现问题时, 包含文件中的内容也不会发送到客户端。设置包含文件目录的过程如下:

(1) 新建包含文件目录 `inc_doc`。

```
shell> mkdir /usr/local/apache/inc_doc
```

(2) 设置 PHP 初始化文件的搜索路径, 使 PHP 程序在 `inc_doc` 目录中搜索包含文件。

```
shell> vi /usr/local/lib/php.ini
```

```
include_path = "/usr/local/apache/inc_doc" 或
```

```
include_path = "other/include_doc:path:/usr/local/apache/inc_doc"
```

包含文件目录创建后, 可将所有要用到的包含文件放在此目录下, 程序运行到调用包含文件的时候, 会自动在 `inc_doc` 目录搜索该文件。

第2章 MySQL 与 PHP 的交互

在上一章中，我们学习了如何安装和配置服务器端的软件。在服务器安装、配置完成后，网站的建设平台就已经搭建好了。但是我们如何来使用这些软件环境进行站点开发呢？

在这一章中，我们就将为大家讲述如何将 MySQL 与 PHP 结合起来进行站点的开发，建立交互式的 Web 网站。这一章从 PHP 的基础知识开始，介绍如何利用 MySQL 数据库服务器与 PHP 进行交互。

本章包括以下两方面的内容：

- PHP 程序设计基础
- 如何使用 MySQL 以及与 PHP 交互的技巧

2.1 PHP 程序设计基础

我们首先来学习进行站点开发所必须的 PHP 语言程序设计的基础知识。PHP 是一种跨平台的服务器端的嵌入式脚本语言，它大量地借用 C、Java 和 Perl 语言的语法，并耦合 PHP 自己的特性，使 Web 开发者能够快速地写出动态页面。它支持目前绝大多数数据库，是开发电子商务应用的利器。

PHP 是一种易于学习和使用的服务器端脚本语言，只需要很少的编程知识，您就能用 PHP 建立一个真正交互的 Web 站点。限于篇幅，本节就只讲述如何使用户能利用 PHP 与 MySQL 的交互来开发动态 Web 站点。

2.1.1 PHP 语言简介

1. PHP 嵌入于 HTML 中

要学好 PHP，首先要熟练掌握 HTML 这一经典的网页编辑语言，才能写出完整有效的 PHP 程序。如果您自信已经熟练掌握了 HTML，并学习了编程的基础知识，那么通过本书丰富的实例，您的 PHP 编程技术将会迅速提高。

PHP 程序可以嵌入到 HTML 中，为了与 HTML 区分开，PHP 可使用四种标记进行两种语言的切换：

(1) 标准的 PHP 标记

```
<?php echo "hello,world!"; ?>
```

(2) 简化的 PHP 标记

```
<? echo "hello,world!"; ?>
```

(3) ASP 风格标记

```
<% echo "hello,world!"; %>
```

(4) Script 标记

```
<SCRIPT LANGUAGE="php"> echo "hello,world!"; </SCRIPT>
```

也就是说,当 PHP 解释器遇到上面四种标记,就按照 PHP 的程序流程执行脚本,如果遇到 HTML 的标记,就转向执行 HTML 的标记,来显示 Web 页面。

注意:ASP 风格的标记和简化的 PHP 标记需要在 PHP 的初始化文件 `php.ini` 中特别声明,否则无效。

例如要使用 ASP 风格的标记,则编辑 `/usr/local/lib/php.ini`,进行如下设置:

```
asp_tag = On;
```

类似地,要使用简化的 PHP 标记,设置:

```
short_open_tag = On;
```

PHP 也可以像 C 语言那样引用包含文件。除了前面提到安全方面的考虑,包含文件还可以经常使用。将实现某种功能的模块化语句写成若干个函数,放在一个包含文件中,当程序需要调用这些功能时,引用此包含文件,之后即可调用其中的函数了。包含文件的好处无论在编写还是在调试 PHP 程序时都是显而易见的,它将那些琐碎但又必不可少的实现过程封装在一个文件中,随时可以调用,使得一个功能模块的添加/删除操作简化到一两个语句的添加/删除,在简化程序结构的同时,提高了程序的可读性和可扩展性。

在程序中引用包含文件的方法有两种:使用 `require` 语句或 `include` 语句。二者的使用形式是一样的,即:

```
require("php_filename");
```

```
include("php_filename");
```

其中“`php_filename`”即包含文件名,在这个语句后面就可以调用 `php_filename` 中包含的任何函数。甚至为了区分于一般的 PHP 程序,包含文件的后缀可以取任意的后缀名。在后面的实例程序中,可以看到很多使用包含文件的例子。

在利用 PHP 编写 MySQL 的应用程序时,经常需要获取和处理客户端提交的 HTML 表单信息,PHP 正是利用这种方法,来完成服务器/客户端的交互,例如邮件列表、在线点播等技术的实现。

让我们先来简单回顾一下 HTML 表单的组成,以下面的 `example.html` 文件为例:

```
<HTML>
<HEAD>
<TITLE>Example</TITLE>
</HEAD>
<BODY>
<CENTER>
<H4>键入查询内容</H4>
<FORM ACTION="search.php" METHOD="POST">
<INPUT TYPE= "TEXT" NAME= "user_id" SIZE=10>
<INPUT TYPE= "SUBMIT" VALUE= "搜索">
```

```
</FORM>
<H4>键入用户密码</H4>
<FORM ACTION="search.php" METHOD="POST">
<INPUT TYPE="PASSWORD" NAME="passwd" SIZE=10>
<INPUT TYPE="SUBMIT" VALUE="提交">
</FORM>
<H4>选择一个颜色</H4>
<FORM ACTION="color.php" METHOD="POST">
<INPUT TYPE="RADIO" NAME="color" VALUE="red" default checked>红色
<INPUT TYPE="RADIO" NAME="color" VALUE="green">绿色
<INPUT TYPE="RADIO" NAME="color" VALUE="yellow">黄色
<INPUT TYPE="SUBMIT" VALUE="提交">
</FORM>
<H4>选择若干个爱好</H4>
<FORM ACTION="hobby.php" METHOD="POST">
<INPUT TYPE="CHECKBOX" NAME="read" VALUE="read">读书
<INPUT TYPE="CHECKBOX" NAME="swim" VALUE="swim">游泳
<INPUT TYPE="CHECKBOX" NAME="music" VALUE="music">听音乐
<INPUT TYPE="SUBMIT" VALUE="提交">
</FORM>
<H4>用户个人信息</H4>
<FORM ACTION="search.php" METHOD="POST">
<INPUT TYPE="HIDDEN" NAME="account" VALUE="root">
<INPUT TYPE="HIDDEN" NAME="passwd" VALUE="1234567">
</FORM>
</CENTER>
</BODY>
</HTML>
```

2. 表单的定义

HTML 使用一对<FORM>和</FORM>括号注释一个表单域,也就是说,当解释器执行到表单定义的时候,就会将<FORM>和</FORM>之间的内容提交给服务器应用程序进行处理,而表单定义之外的信息无法提交,只能反映在客户端的浏览器页面上。表单的定义还需说明表单的提交方式和表单的处理程序,分别用 METHOD 和 ACTION 属性定义。表单的提交方式 METHOD 有两种选择:

(1) POST

将表单数据本身传递给服务器。对 PHP 来说,此类表单的数据存放在 \$HTTP_POST_VARS 数组中。例如要提交上例对用户的搜索,只要在处理表单信息的 search.php 中使用 \$HTTP_POST_VARS[user_id] 即可得到客户端输入的表单信息。

(2) GET

将表单数据以 URL 的形式传递给服务器。对于上面的例子,也可以在地址栏中输入:

http://.../example.php?user_id=root, 则 PHP 程序将从 URL 行获取的表单信息存储在 \$HTTP_GET_VARS 数组中, 同样可利用 \$HTTP_GET_VARS[user_id] 进行数据处理。

对于上例中的第一个表单, 其处理程序 ACTION 定义为 search.php, 即当用户按下“搜索”按钮提交表单后, user_id 域的值就传递给 search.php 进行处理。

3. 单组件的定义

在表单定义之间, 可以插入若干个表单组件。表单组件的定义使用 <INPUT> 标签进行注释。其中包括下列属性:

(1) TYPE

定义组件的类型, 包括: TEXT (文本输入框)、PASSWORD (密码输入框)、RADIO (单选框)、CHECKBOX (复选框)、RESET (复位按钮)、SUBMIT (提交按钮)、HIDDEN (隐藏域) 等。它们各自的表现形式如图 2-1 所示。

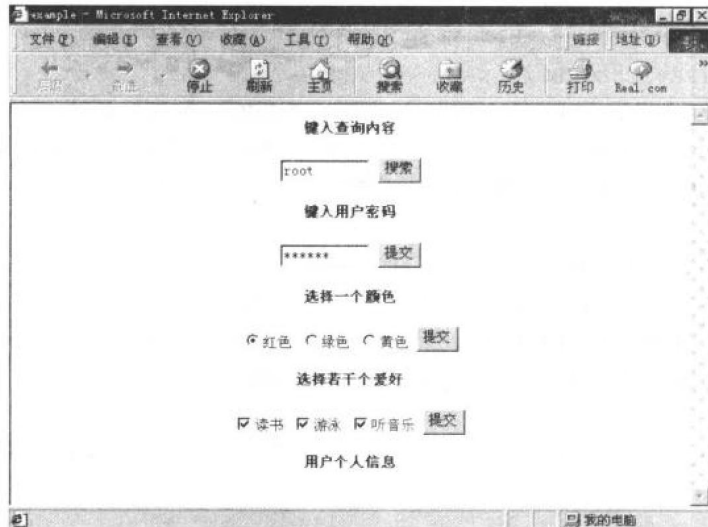


图 2-1 example.html 显示的各类表单

(2) NAME

定义组件的名称, 对大多数组件, PHP 程序可使用其名称做变量名来表示一个组件的值。例如, 要获得 example.html 中用户输入的第一项数据——要查询的用户名, 只需在 search.php 中使用 \$user_id 或 \$HTTP_POST_VARS[user_id] 即可表示客户端在该输入框中输入的数据。

(3) VALUE

定义组件的值, 当需要某输入框显示缺省值时, 使用该属性定义。

(4) SIZE

定义组件的大小。

(5) MAXLENGTH

定义组件所能显示的最大字符数。