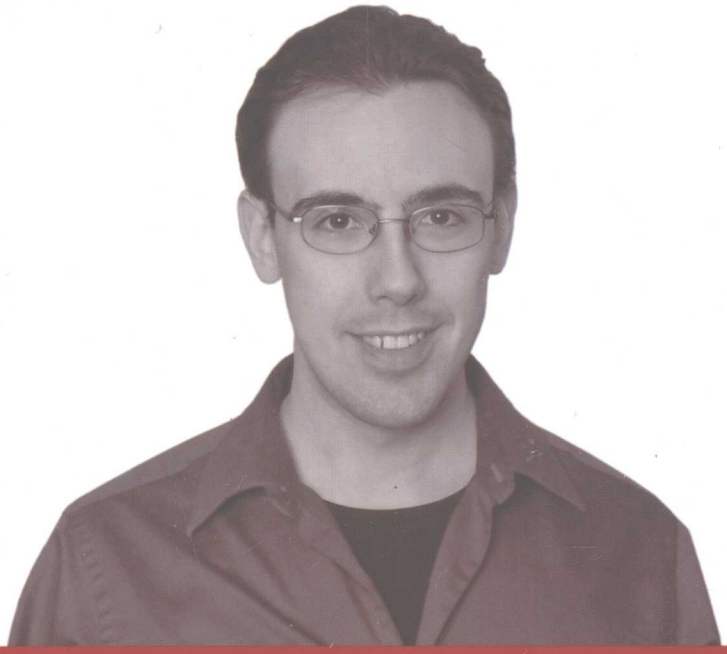
PROGRAMMER TO PROGRAMMER™



PHP and MySQL Create-Modify-Reuse

# PHP & MySQL 范例精解

## ——创建、修改、重用

(美) Tim Boronczyk
Martin E. Psinas 著
熊 伟 译

wrox™
WILEY
清华大学出版社

# PHP & MySQL 范例精解

## —— 创建、修改、重用

(美) Tim Boronczyk
Martin E. Psinas 著

熊 伟 译

清华大学出版社

北 京

Tim Boronczyk, Martin E. Psinas

PHP and MySQL: Create-Modify-Reuse

EISBN: 978-0-470-19242-9



**图书在版编目(CIP)数据**

PHP & MySQL 范例精解—— 创建、修改、重用/(美)波罗斯泽亚克(Boronczyk, T.)，(美)普斯纳斯(Psinas, M.E.)著；熊伟 译. —北京：清华大学出版社，2009.4

书名原文：PHP and MySQL: Create-Modify-Reuse

ISBN 978-7-302-19562-7

I.P…　II.①波…　②普…　③熊…　III.①PHP 语言—程序设计②关系数据库—数据库管理系统，MySQL　IV.TP312　TP311.138

中国版本图书馆 CIP 数据核字(2009)第 019944 号

**责任编辑**：王　军　李楷平
**装帧设计**：孔祥丰
**责任校对**：成凤进
**责任印制**：孟凡玉

**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
**社　总　机**：010-62770175　　**邮　　购**：010-62786544
**投稿与读者服务**：010-62776969，c-service@tup.tsinghua.edu.cn
**质　量　反　馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn

**印 刷 者**：北京市世界知识印刷厂
**装 订 者**：北京市密云县京文制本装订厂
**经　　销**：全国新华书店
**开　　本**：185×260　**印　张**：21.5　**字　数**：523 千字
**版　　次**：2009 年 4 月第 1 版　　**印　次**：2009 年 4 月第1 次印刷
**印　　数**：1～4000
**定　　价**：48.00 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：028903-01

# 译 者 序

随着 Internet 的发展，网络应用成为了目前最流行的应用之一，对人类生活的方方面面都产生着深远的影响。PHP 和 MySQL 作为开发网络应用程序的最重要工具之一，一直受到开发人员的追捧。但网络上存在着各种类型的应用，如何有效地使用 PHP 和 MySQL 进行开发成为程序员面临的一个难点问题。从某种意义上说，本书就是为解决这个问题应运而生的。

很高兴能有机会翻译这本书。虽然，这本书只介绍了一些实例，而对 PHP 和 MySQL 语法没有做过多阐述。但在翻译过程中，笔者发现书中介绍的例子都很有特点，能切中当前最流行的网络应用，而且书中提供的源代码都切实可用，这对那些有一定 PHP 编程经验的开发人员进行快速开发是非常有帮助的。而对于那些刚入门的 PHP 程序员，本书也能让他们了解如何完整地构建一个具体应用。通常，在实践中学习是最有效率的而且对技术的理解也最深刻。

本书由熊伟翻译，由肖国尊负责本书的翻译质量和进度。由于译者的水平有限，加之时间仓促，错误与不到位之处在所难免。敬请广大读者提供反馈意见，读者可以将意见 E-mail 至 wkservice@vip.163.com，我们会仔细查阅读者发来的每一封邮件，以求进一步提高今后译著的质量。

最后，真切希望这本书能对那些希望使用 PHP 和 MySQL 进行项目开发的人有所帮助。

译者

2008 年 11 月

# 关 于 作 者

Timothy Boronczyk 出生于纽约州雪城，他是一个自由开发者、程序员和技术编辑。他从 1998 年开始接触网站设计，并在随后的几年中撰写了多篇关于 PHP 编程的文章和技术教程。Timothy 获得了软件工程学位，目前正开始他的第一个创业计划，开办 Salt City Tech 网站(www.saltcitytech.com)。在业余时间，他喜欢摄影、跟朋友聚会，以及睡觉时把脚从床尾伸出去。还有，在日常生活中，他的注意力很容易被发光物体所干扰。

## 关于贡献者

Martin E. Psinas 是一位公认的计算机安全专家和开源社区的重要成员。他从事过技术编辑、代码审计员等工作，同时也为培生教育出版集团和排名第一的 PHP 杂志 *PHP| Architect* 撰写文章。在空闲时间，他致力于维护他的个人网站和充当 codewalkers.com——一家为 PHP 和 MySQL 开发人员提供各种资源的网站——的义务管理员和投稿人。此外，Martin 还和 PHP 项目的负责人以及 PHP 用户组保持着密切联系。

# 前　言

我很惊讶于 Internet 在过去的十余年间所取得的发展和进步。它已从一个通过少量超链接相互连接的静态文本集发展成为一个能支持功能丰富的分布式程序的平台。通常，在开发这些基于 Web 的应用程序时，许多程序员都会选择使用 PHP 和 MySQL。

在本书中，将介绍 12 个可以按照您的意愿使用和扩展的 PHP 实例。我尽力按照可重用标准来编写这些代码，在一些例子中甚至整个程序都可以被重用。

很高兴能有这个机会来编写并和您共享这些程序，同时我也希望您能在阅读和学习这些代码的过程中得到乐趣。更重要的一点是，我希望您能从这本书中得到有价值的并且实用的代码。

## 本书面向对象

在本书中，我将给出一些基础但功能强大的程序代码。您可以按照自己认为合适的方式实现和扩展这些代码。但前提是您需要懂得一些关于 PHP 和通用 Web 开发技术的基础知识。本书不是一本教科书。然而，您并不需要具备高级 PHP 程序员的资格才能深入理解本书。初学者会发现这本书很实用，因为它能指导他们编写各种类型的程序。本书中的 12 个实例可以激发他们的兴趣来编写更多的属于自己的程序。中级的和更有经验的程序员也能从本书中获益，因为他们可以对书中提供的程序源代码进行一些修改，然后在实际程序中使用。

书中的一些程序是在前面程序的基础上构建的，因此虽然不需要从头至尾通读此书，但是不管读者编程水平如何，我都建议阅读所有相关章节。比如，在第 7 章中，介绍了一个在线相册程序，但是其中的图片上传功能就使用了第 6 章所介绍的 AJAX 文件管理器。书中所有项目都是按照第 1 章确定的代码结构进行设计的。

## 本书涵盖内容

本书的代码基于 MySQL 5.0 Community Server 和 PHP 5.2.5，也支持更高版本的服务器。如果想要在低版本的服务器中运行本书代码，可能需要根据实际情况做一些修改。

## 本书组织结构

书中所有章节都是按照特定顺序组织的，以便后面章节中的程序利用前面章节所做的

工作。下面是本书的纲要介绍。

**第 1 章：用户注册系统**
创建基本的用户注册系统
可重用组件：configuration/include 文件，401.php，User 类

**第 2 章：社区论坛**
扩展用户注册系统，创建一个具有用户权限和按话题发帖功能的社区论坛
可重用组件：JpegThumbnail 类，BBCode 类

**第 3 章：邮件列表**
创建一个具有控制地址和邮件摘要功能的邮件列表
可重用组件：POP3Client 类

**第 4 章：搜索引擎**
为个人网站定制搜索引擎
可重用组件：整个程序

**第 5 章：个人日历**
编写一个个人日历工具以使生活变得更有计划
可重用组件：整个程序

**第 6 章：AJAX 文件管理器**
创建一个基于 AJAX 的文件上传和目录查看器
可重用组件：整个程序(这个项目介绍了后面章节中将用到的 AJAX 技术)

**第 7 章：在线相册**
创建一个基于文件的图库，能自动生成 JPEG 和 QuickTime 格式文件的缩略图
可重用组件：MovThumbnail 类

**第 8 章：购物车**
编写一个分类购物车
可重用组件：ShoppingCart 类

**第 9 章：网站统计**
记录网站流量和登录用户的信息以帮助做出更好的商业决策
可重用组件：PieChart 类，BarChart 类

**第 10 章：新闻/博客系统**
开发一个支持评论和 RSS 反馈的新闻或博客系统
可重用组件：整个程序(项目中也介绍了一些第三方的可重用组件，如 YUI 日程表和 TinyMCE 富文本控件)

**第 11 章：shell 脚本**

编写和运行管理控制脚本程序

可重用组件：CommandLine 类，recurs_copy()函数

**第 12 章：安全和日志**

介绍 SQL 注入、路径模式发掘攻击、弱认证和跨站脚本攻击以及如何修复这些安全漏洞

可重用组件：write_log()函数，view_log.php，痕迹删除脚本

## 本书学习要求

在学习本书的过程中需要编写 PHP 程序，因此需要一个编辑器来输入代码。可根据个人喜好选择适合自己的编辑器。此外，还需要一个能支持 PHP 和 MySQL 的服务器来运行示例程序。当然，访问这些程序的浏览器也必不可少。至于使用何种服务器和浏览器则由个人决定。为了方便读者阅读本书，在必要的时候，书中将分别给出在 UNIX 和 Windows 平台下运行本书演示程序的说明，比如第 3 章的邮件列表，它将作为 Windows 系统的一个计划任务运行。

就个人而言，我喜欢使用 Vi 编写代码，在 Slackware Linux 服务器上运行程序，并通过 Windows XP 系统上的 Firefox 浏览器访问这些程序。

虽然尽力避免，但本书中的一些程序还是调用了 PHP 中的扩展函数。例如，第 4 章的搜索引擎程序使用了 pspell 扩展。对于那些只在扩展中才有的功能，本书没有采用第三方的非标准扩展，因此读者在安装扩展时只需要参考 www.php.net 中的官方文档。这些扩展将在相关章节中进行详细介绍。

## 源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 http://www.wrox.com/ 或 www.tupwk.com.cn
/downpage 上下载。登录到站点 http://www.wrox.com/，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

**注释：**

*由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是* 978-0-470-19242-9。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 http://www.wrox.com/dynamic/books/download.aspx 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

## 勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们就会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 http://www.wrox.com，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

## p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 http://p2p.wrox.com 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

(1) 进入 p2p.wrox.com，单击 Register 链接。

(2) 阅读使用协议，并单击 Agree 按扭。

(3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按扭。

(4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

**注释：**

*不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。*

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

# 目　录

# 第 1 章

# 用户注册系统

在网站中，账号注册和用户登录是让用户体验个性化服务和查看感兴趣内容的重要方式。身份认证功能在许多社区和电子商务网站中都发挥着重要作用。因此，本书介绍的第一个应用程序就是用户注册系统。

从功能上来看，用户使用注册系统可以创建新账号。在注册过程中，用户必须提供一个电子邮箱来确认注册信息。在以后的使用过程中，用户还需要更改密码和电子邮箱地址，以及重新设置忘记的用户密码。这些都是目前注册系统必须提供的标准功能，也是用户认为一个完善的注册系统应该拥有的基本功能。

从程序结构方面来看，必须合理地设计程序代码的目录结构。例如，支持文件和引用文件(support and include files)不能保存在公众可以访问的目录中。而用户信息则应该存储在数据库中。这是因为目前有大量工具支持查看和操作关系数据库(如 MySQL)中的数据，这为数据访问提供了透明性和灵活性。

## 1.1 目录结构设计

在开发过程中，第一步是设计程序的目录结构。建议创建三个主文件夹：第一个是 public_files，用来保存所有可以公共访问的页面；第二个是 lib，用来保存可以被其他文件调用的引用文件；最后一个是 templates，用于保存页面显示文件。虽然 PHP 可以调用程序目录下的所有文件，但是 Web 服务器应该只允许外界访问 public_files 目录中的文件。把支持文件保存在外界可以访问的文件夹之外可以增强系统的安全性。

在 public_files 中，创建 css 子目录来保存样式表，js 子目录保存 JavaScript 文件以及 img 子目录保存图片文件。可能还需要创建其他文件夹来保持代码结构的清晰性。例如，可以创建用于保存 MySQL 文件的 sql 目录、保存文档和开发笔记的 doc 目录，以及包含冒烟测试和单元测试代码的 tests 目录。

## 1.2 数据库设计

除了代码目录层次结构，还要考虑怎样设计数据库结构。程序需要从用户那里收集的信息与网站所提供的服务种类相关。而这些信息反过来又会影响数据库表结构。在注册系统中，保存在数据库中的信息至少包括一个独有的用户 ID、用户名、密码哈希表和电子邮箱地址。此外，还必须有用于区分和定位已认证账户和待认证账户的机制。

```
CREATE TABLE WROX_USER (
    USER_ID     INTEGER UNSIGNED   NOT NULL AUTO_INCREMENT,
    USERNAME    VARCHAR(20)        NOT NULL,
    PASSWORD    CHAR(40)           NOT NULL,
    EMAIL_ADDR  VARCHAR(100)       NOT NULL,
    IS_ACTIVE   TINYINT(1)         DEFAULT 0,

    PRIMARY KEY (USER_ID)
)
ENGINE=MyISAM DEFAULT CHARACTER SET latin1
    COLLATE latin1_general_cs AUTO_INCREMENT=0;

CREATE TABLE WROX_PENDING (
    USER_ID        INTEGER UNSIGNED  NOT NULL,
    TOKEN          CHAR(10)          NOT NULL,
    CREATED_DATE   TIMESTAMP         DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (USER_ID)
       REFERENCES WROX_USER(USER_ID)
)
ENGINE=MyISAM DEFAULT CHARACTER SET latin1
    COLLATE latin1_general_cs;
```

因为程序中将使用其返回值为由 40 个十六进制字符所组成的数组的 sha1()函数，因此 WROX_USER 表中密码哈希值表的 PASSWORD 字段大小被设置为 40。另外，在编写程序时要记住永远不要直接把原始密码保存到数据库中——从安全方面看，这是一个很好的预防措施。在本章程序中采用的方法是：当用户第一次提交密码时，生成密码所对应的哈希值并将其保存到数据库。当用户下一次登录时，用同一个哈希函数根据密码进行哈希索引并与数据库中的保存值进行比较。

程序中的电子邮箱地址长度的最大值被设为 100 个字符。虽然技术标准允许电子邮箱地址的最大长度为 320 个字符(其中 64 个字符用于用户名，1 个字符用于@，剩下的 255 个字符则用于保存主机名)。但在实际生活中我还没见过有谁使用这么长的邮箱名，事实上很多使用长度限制为 100 个字符的邮箱名的数据库都工作得很好。

在数据库中，还需要保存其他信息，如名字、姓氏、家庭地址、所在城市、州/省份、邮政编码和电话号码等。

WROX_PENDING 表中包含一个能自动初始化的时间戳字段，程序利用它回溯数据库并删除那些已经很长时间没有被激活的登录账户。这个表本来可以和 WROX_USER 合并，

但由于 WROX_PENDING 表中待确认标志只会使用一次，因此本书中选择将它们分开存储。在使用过程中，数据库中论坛用户的个人信息相对比较稳定，这样 WROX_USER 表就不会由于存在临时数据而出现冗余。

## 1.3　编写共享代码

系统把被多个文件共享的代码保存在一个专门文件夹中，并通过 include 或者 require 的方式调用这些共享代码从而避免产生冗余，这样也使程序的维护变得更简单。只要有可能，在今后的程序中都应该以函数或者类的形式实现能够被重用的代码。在编写程序时采用代码重用的思想是一个很好的习惯。程序中的 common.php 文件包含了被其他脚本文件调用的共享代码，这些代码用于实现一个完整、健全的基础运行环境。因为用户不需要直接访问这个文件，所以将它保存在 lib 目录中。

```
<?php
// set true if production environment else false for development
define ('IS_ENV_PRODUCTION', true);

// configure error reporting options
error_reporting(E_ALL | E_STRICT);
ini_set('display_errors', !IS_ENV_PRODUCTION);
ini_set('error_log', 'log/phperror.txt');

// set time zone to use date/time functions without warnings
date_default_timezone_set('America/New_York');

// compensate for magic quotes if necessary
if (get_magic_quotes_gpc())
{
    function _stripslashes_rcurs($variable, $top = true)
    {
        $clean_data = array();
        foreach ($variable as $key => $value)
        {
            $key = ($top) ? $key : stripslashes($key);
            $clean_data[$key] = (is_array($value)) ?
                stripslashes_rcurs($value, false) : stripslashes($value);
        }
        return $clean_data;
    }
    $_GET = _stripslashes_rcurs($_GET);
    $_POST = _stripslashes_rcurs($_POST);
    // $_REQUEST = _stripslashes_rcurs($_REQUEST);
    // $_COOKIE = _stripslashes_rcurs($_COOKIE);
}
?>
```

在程序运行时，并不是任何时候都能控制着服务器的运行，所以设置一些常用管理指

令可以使系统更加灵活。例如，设置错误报告选项，可以在开发阶段直接显示错误，而在系统实际运行后则把错误信息重定向到一个特定位置以避免显示给用户。

魔术引号(Magic quotes)是一个 PHP 配置选项，它让 PHP 自动对输入流中的单引号、双引号和反斜杠进行转义操作。虽然这个功能看起来很有用，但在实际中，简单地将该选项设置为开或者关常常会带来许多问题。所以最好的方法是先规范化数据然后用 addslashes() 或者 mysql_real_escape_string()(当需要把输入数据保存到数据库中时使用后者比较好)进行处理。魔术引号修正能确保数据按用户自己确定的方式和时间被正确转义而不用考虑 PHP 的配置方式，这样开发将会变得更简单而且错误更少。

建立程序与 MySQL 数据库之间的连接是一个很常用的功能，因此可以用单独的文件实现。db.php 包含了配置常量和建立连接的函数代码。同样地，由于需要被其他文件引用且不能被用户直接访问，因此 db.php 被保存在 lib 目录中。

```
<?php
// database connection and schema constants
define('DB_HOST', 'localhost');
define('DB_USER', 'username');
define('DB_PASSWORD', 'password');
define('DB_SCHEMA', 'WROX_DATABASE');
define('DB_TBL_PREFIX', 'WROX_');

// establish a connection to the database server
if (!$GLOBALS['DB'] = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD))
{
   die('Error: Unable to connect to database server.');
}
if (!mysql_select_db(DB_SCHEMA, $GLOBALS['DB']))
{
   mysql_close($GLOBALS['DB']);
   die('Error: Unable to select database schema.');
}
?>
```

常量 DB_HOST、DB_USER、DB_PASSWORD 和 DB_SCHEMA 包含了创建一个成功的数据库连接所需的参数。如果在运行过程中，数据库服务器与 PHP 和 Web 服务器不在同一台主机上，就需要设置一个有效的 DB_PORT 值并相应地调整对 mysql_connect()函数的调用方式。

在系统中，数据库连接句柄被保存到超全局数组$GLOBALS，因此在所有包含 db.php 的文件的任何作用域中(或者是引用了 db.php 的文件中的所有对象)都能访问到它。

与此同时，给数据库表名加前缀也能防止与其他程序保存到同一个数据库模式中的表相冲突，并且将前缀设置为常量可以使代码在将来需要时更容易进行升级，因为这个值的定义只在一个地方出现。

通用函数也可以保存到专门文件中。例如，本书在程序中计划使用 random_text()函数来生成 CAPTCHA 验证码和确认标志，因此这个函数被保存到一个名为 functions.php 的文件中。

```
<?php
// return a string of random text of a desired length
function random_text($count, $rm_similar = false)
{
    // create list of characters
    $chars = array_flip(array_merge(range(0, 9), range('A', 'Z')));

    // remove similar looking characters that might cause confusion
    if ($rm_similar)
    {
        unset($chars[0], $chars[1], $chars[2], $chars[5], $chars[8],
            $chars['B'], $chars['I'], $chars['O'], $chars['Q'],
            $chars['S'], $chars['U'], $chars['V'], $chars['Z']);
    }

    // generate the string of random text
    for ($i = 0, $text = ''; $i <$count; $i++)
    {
        $text .= array_rand($chars);
    }

    return $text;
}
?>
```

在编程时，一条重要的法则是不管使用的是何种语言，永远都不要信任用户的输入。人们能(而且会)提供各种疯狂和不可预料的输入。有些时候是无意的，但有些时候则是不怀好意的。虽然 PHP 中的 filter_input()和 filter_var()函数能用于验证用户的输入数据，但一些开发人员还是偏向于自己实现该功能，因为 PHP 5.2.0 以前的版本不支持该功能。如果您也是他们中的一员，也可以把这些函数保存到 functions.php 中。

## 1.4　User 类

可以将维护用户账户信息的主要代码封装到一个数据结构中，以利于在以后的应用中进行重用或者扩展。这些代码包括使信息存储和读取更简捷的数据库交互逻辑。下面是 User.php 文件的代码：

```
<?php
class User
{
    private $uid;       // user id
    private $fields;    // other record fields

    // initialize a User object
    public function __construct()
    {
        $this-> uid = null;
```

```
        $this-> fields = array('username' => '',
                               'password' => '',
                               'emailAddr' => '',
                               'isActive' => false);
    }

    // override magic method to retrieve properties
    public function __get($field)
    {
        if ($field == 'userId')
        {
            return $this-> uid;
        }
        else
        {
            return $this-> fields[$field];
        }
    }

    // override magic method to set properties
    public function __set($field, $value)
    {
        if (array_key_exists($field, $this-> fields))
        {
            $this-> fields[$field] = $value;
        }
    }

    // return if username is valid format
    public static function validateUsername($username)
    {
        return preg_match('/^[A-Z0-9]{2,20}$/i', $username);
    }

    // return if email address is valid format
    public static function validateEmailAddr($email)
    {
        return filter_var($email, FILTER_VALIDATE_EMAIL);
    }

    // return an object populated based on the record's user id
    public static function getById($user_id)
    {
        $user = new User();
        $query = sprintf('SELECT USERNAME,PASSWORD, EMAIL_ADDR, IS_ACTIVE ' .
            'FROM %sUSER WHERE USER_ID = %d', DB_TBL_PREFIX, $user_id);
        $result = mysql_query($query, $GLOBALS['DB']);
        if (mysql_num_rows($result))
        {
            $row = mysql_fetch_assoc($result);
```