



“十二五” 高等教育规划教材

PHP 程序设计 高级教程

PHP CHENGXU SHEJI GAOJI JIAOCHENG

传智播客高教产品研发部 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十二五”高等教育规划教材

PHP 程序设计高级教程

传智播客高教产品研发部 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

PHP 是一种运行于服务器端并完全跨平台的嵌入式脚本编程语言,是目前开发各类 Web 应用的主流语言之一。本书就是面向具备 PHP 编程基础的学习者推出的一本进阶教材,以精心设计的应用案例、阶段案例和项目实战,全面讲解了 PHP 中级项目的开发技术。

本书共 10 章,分类介绍了 PHP 各方面高级应用。本书附有配套视频、源代码、习题、教学课件等资源,而且为了帮助初学者更好地学习本书讲解的内容,还提供了在线答疑,希望得到更多读者的关注。

本书适合作为高等院校计算机相关专业程序设计或者 Web 项目开发的教材,也可作为 PHP 进阶培训的教材,是一本适合广大计算机编程爱好者的优秀读物。

图书在版编目(CIP)数据

PHP 程序设计高级教程 / 传智播客高教产品研发部编
著. —北京: 中国铁道出版社, 2015. 1
“十二五”高等教育规划教材
ISBN 978-7-113-19571-7

I. ①P… II. ①传… III. ①PHP 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 306233 号

书 名: PHP 程序设计高级教程
作 者: 传智播客高教产品研发部 编著

策 划: 翟玉峰 读者热线: 400-668-0820
责任编辑: 翟玉峰 何 佳
封面设计: 徐文海
封面制作: 白 雪
责任校对: 汤淑梅
责任印制: 李 佳

出版发行: 中国铁道出版社(100054, 北京市西城区右安门西街 8 号)
网 址: <http://www.51eds.com>
印 刷: 北京海淀五色花印刷厂
版 次: 2015 年 1 月第 1 版 2015 年 1 月第 1 次印刷
开 本: 787 mm×1 092 mm 1/16 印张: 22.75 字数: 552 千
印 数: 1~2 000 册
书 号: ISBN 978-7-113-19571-7
定 价: 45.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

为什么出这本书

数年前，传智播客团队是一群默默工作在 IT 岗位的程序员，现如今，为了同一个梦想，我们聚集在一起，为“改变中国的 IT 教育”而全力以赴。为什么会有这样的梦想，大家先来听一听 IT 毕业生的心声：

“我们专业的课程太枯燥了，太多理论方面的课程，并且这些理论听起来都是很难理解的，同学们学完了都是一头雾水。”

“终于毕业了，学了四年的计算机课程，我们班上大部分同学都不知道能干什么，感觉好像什么都学了，但却什么都不会干！”

“实习单位根本就不让我们干活，最多只是让我们阅读代码、修改文档，说我们什么都不会，不敢让我们参与实际工作。想想自己辛苦学了几年连实习的要求都达不到，心里很不是滋味。”

“面试的时候，用人单位问能做什么，做过什么项目，我们回答不上来，面试过程中问到的很多问题，我们连听都没有听过。”

... ..

这些心声并不是个别毕业生所表达的，而是中国高校反映出的一种普遍现象。高校的 IT 教育与企业的真实需求存在严重脱节，如果高校的相关课程仍然不进行更新的话，毕业生将继续面临毕业即失业的困境，出现这样的困境说明中国的高等教育仍存在一定缺陷。就目前高校计算机类的相关教育而言，问题主要有两个方面：一是学校所用的教材过于陈旧，教材更新速度非常缓慢，导致学生所学的知识与现在的社会需求严重不符；第二是高校的老师长年从事教育工作，缺乏实际开发经验，从而无法向学生传授企业需要的专业技能。很多高校也都认识到了这些问题，但却无法进行全面改革。

高等教育，肩负着培养高等人才，引领先进生产力，影响社会发展的重大使命。一个民族要想日久昌盛，就必须提高教育质量，加强教育发展。俗话说的好，国家兴亡，匹夫有责。传智播客并不想把自己上升到民族复兴的高度，但传智播客确实将“改变中国的 IT 教育”作为自己的事业并为此拼搏了九年。在这九年期间，传智播客默默耕耘，陆续出版了计算机书籍十几本、教学视频几十套、发表各类技术文章几百篇，直接培养的软件工程师就有 2 万多名，被传智播客影响的在校学生更是多达数百万人。毫不夸张地说，传智播客已经为 IT 学子开辟了一条全新的求知之路。

九年的沉淀让传智播客拥有了目前国内最好的 IT 培训课程。如今，传智播客决定把这套精品课程全部公开，印刷成书并输送至高校，解决眼下高校亟待解决的课程陈旧问题。另外，面向高校计算机专业的老师，传智播客还将提供免费的培训服务，让高校老师可以通过培训提高授课技能，将最专业、最实用的技能传授给学生。为了让广大师生在学习传智播客课程的同时能达到最好的效果，公司投入巨额资金，用于为高校师生提供以下学习配套资源与服务：

1. 本书非常荣幸地落户于在线学习网站【“博学谷”：<http://www.boxuegu.com>】。全宇宙的人都知道：“学习 IT 的人都在博学谷”，难道您真不知道？

2. 在学习本书过程中，遇到任何问题，请找问答精灵，我们承诺工作时间 3 小时内解答您的问题。问答精灵的官方网址为：<http://ask.boxuegu.com>。读者可以通过扫描下面

的二维码，下载问答精灵移动客户端或关注问答精灵微信公众平台。学习 IT 有问题就找问答精灵。



问答精灵移动客户端



问答精灵微信公众平台

3. 我们为本书录制了全程教学视频，通过学习视频您可以更容易理解本书的知识，视频下载网址：<http://dvd.boxuegu.com/php-advanced>。

4. 心中有疑问，想找老师面对面解惑？每周六晚上 8 点到 10 点，“面对面”讲堂，我们与您不见不散，网络公开课网址：<http://openclass.boxuegu.com/php-advanced>。

5. 学习需要氛围，在校园快点组织起您的学习社团吧，只要申请通过，我们将给予社团各种支持，包括：免费提供教材、视频教程、技术参考资料、技术面试资料、技术讲座、实习岗位等支持，优秀社团还可获得现金奖励。

学习社团报名网址：<http://club.boxuegu.com>。

6. 本书配套源代码下载网址：<http://book.boxuegu.com/php-advanced>。

7. 针对高校教学，传智播客用近八年的教育培训经验，精心设计了“教材+教案+授课资源+考试系统+题库+教学辅助案例”一站式 IT 就业培训系列教程，方便老师进行教学，能够有效提高老师日常教学的效率。

如需索要配套教学资源，请关注博学谷高校老师俱乐部，扫描下方二维码关注微信公众平台。



高校老师俱乐部

希望通过我们的努力，在不久的将来，高校能够真正培养出符合企业所需的实用型人才，IT 学子们不再为就业而迷惘！

关于本教材

本教材是面向具备 PHP 编程基础的读者推出的一本进阶教材，围绕 PHP 项目开发的相关技术进行讲解，包括 MVC 思想、Smarty 模板引擎、Ajax、jQuery、ThinkPHP 框架以及 Linux 服务器。本教材对每个知识点都进行了深入分析，并针对每个知识点精心设计了相关案例，然后模拟这些知识点在实际工作中的运用，真正做到了知识的由浅入深、由易到难。

本教材共分为 10 个章节，接下来分别对每个章节进行简单介绍，具体如下：

- 第 1 章主要介绍了 PHP 的数据库操作，即 PHP 的 mysql、mysqli 和 PDO 三个扩展的使用。其中，mysql 扩展在经典项目中比较常用，PDO 扩展适合在新项目中使用，读者应重点掌握这两种扩展的使用。

- 第2章主要讲解 MVC 设计模式。通过对 MVC 的典型实现,可以帮助读者更直观地学习 MVC 的设计思想和开发流程。本章还具有完整的留言板项目开发案例,可以帮助读者积累项目开发经验。
- 第3章主要讲解 Smarty 模板引擎。Smarty 的逻辑显示分离和缓存功能非常实用,通过学习可以使读者体会到 Smarty 的优势。本章还讲解了迷你版 Smarty 的实现和 Smarty 在 MVC 项目中的整合,将使读者对 Smarty 有更深入的理解。
- 第4章、第5章主要讲解了 Ajax 和 jQuery。通过这两章的学习可以使读者深入理解 Ajax 技术,并能运用 jQuery 简化 JavaScript 和 Ajax 的操作。
- 第6章、第7章主要讲解了 ThinkPHP 框架的使用。通过学习 ThinkPHP,读者可以更加高效地开发 PHP 程序。
- 第8章、第9章为项目实战,综合运用前面章节的知识来开发一个电子商务网站。通过这两章的学习,读者可以积累更多的 PHP 项目开发经验。
- 第10章主要讲解了 Linux 操作系统的安装使用、vi 编辑器的使用和 LAMP 环境的搭建与配置,同时讲解了项目部署,使读者具备 LAMP 环境下的开发经验。

在上面所提到的10章中,第1章、第3~7章和第10章主要是讲解新内容,这些章的知识点多而细,大家需要多动手练习,奠定扎实的基础。第2、8、9章是对知识点的综合运用,这些章的内容比较复杂,希望初学者多加思考,认真完成教材中所讲解的每个案例。

在学习本教材时,首先要做到对知识点理解透彻,其次一定要亲自动手去练习教材中所提供的案例,因为在学习软件编程中动手实践是非常重要的。对于一些非常难以理解的知识也可以选择通过案例的练习来学习,如果实在无法理解教材中所讲解的知识,建议读者不要纠结于某一个知识点,可以先往后学习。通常来讲,看了后面一两个小节的内容后再回来学习之前不懂的知识点一般就都能理解了。

致谢

本教材的编写和整理工作由传智播客教育科技有限公司高教产品研发部完成,主要参与人员有徐文海、张绍娟、韩冬、乔治铭、梅杰,研发小组全体成员在这近一年的编写过程中付出了很多辛勤的汗水。另外,传智播客讲师韩顺平、韩忠康、王超平、郭冠召等人也参与了教材的部分编写和修订工作,在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力,但教材中难免会有不妥之处,欢迎各界专家和读者朋友们来信来涵给予宝贵意见,我们将不胜感激。您在阅读本书时,如发现任何问题或有不同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至: itcast-book@vip.sina.com

传智播客教育科技有限公司 高教产品研发部

2014年11月25日 于北京

第 1 章 PHP 操作数据库..... 1

- 1.1 PHP 访问 MySQL..... 1
 - 1.1.1 PHP 访问 MySQL 的基本步骤 1
 - 1.1.2 PHP 相关扩展介绍 2
- 1.2 mysql 扩展的使用 4
 - 1.2.1 连接 MySQL 服务器 4
 - 1.2.2 选择数据库 5
 - 1.2.3 执行 SQL 语句 6
 - 1.2.4 处理结果集 7
 - 1.2.5 释放资源与关闭连接 13
- 1.3 mysqli 扩展的使用 15
 - 1.3.1 mysqli 连接并选择数据库 15
 - 1.3.2 mysqli 操作数据库 16
- 1.4 PDO 扩展的使用 18
 - 1.4.1 什么是 PDO 18
 - 1.4.2 PDO 连接数据库 18
 - 1.4.3 PDO 执行 SQL 语句 19
 - 1.4.4 PDO 处理结果集 24
 - 1.4.5 PDO 错误处理机制 30
- 本章小结 32

第 2 章 MVC 设计模式 33

- 2.1 什么是 MVC 33
- 2.2 MVC 典型实现 34
 - 2.2.1 模型 34
 - 2.2.2 控制器 40
 - 2.2.3 框架 45
- 2.3 阶段案例——留言板 50
 - 2.3.1 案例分析 50
 - 2.3.2 前台模块实现 53
 - 2.3.3 数据安全处理 67
 - 2.3.4 后台模块实现——用户登录 69
 - 2.3.5 后台模型实现——留言管理 74
- 本章小结 81

第 3 章 Smarty 模板引擎 82

- 3.1 Smarty 入门 82
 - 3.1.1 什么是模板引擎 82
 - 3.1.2 Smarty 的下载与配置 83

- 3.1.3 案例——Smarty 模板简单应用 85
- 3.2 Smarty 实现原理 86
 - 3.2.1 深入分析 Smarty 实现原理 86
 - 3.2.2 案例——动手实现迷你版 Smarty 87
- 3.3 Smarty 详解 89
 - 3.3.1 Smarty 的基础语法 89
 - 3.3.2 变量修饰器 94
 - 3.3.3 内置函数 104
 - 3.3.4 自定义函数 108
 - 3.3.5 缓存 112
- 3.4 阶段案例——优化留言板 117
- 本章小结 121

第 4 章 Ajax 技术 122

- 4.1 什么是 Ajax 122
- 4.2 Ajax 具体使用 123
 - 4.2.1 Ajax 对象创建 123
 - 4.2.2 常用方法和属性 126
- 4.3 JSON 数据格式 134
 - 4.3.1 JSON 的介绍与使用 134
 - 4.3.2 案例——获取天气预报信息 137
- 4.4 Ajax 应用案例 140
 - 4.4.1 案例——Ajax 实现无刷新分页 140
 - 4.4.2 案例——实现进度条文件上传 144
- 本章小结 146

第 5 章 jQuery 框架 147

- 5.1 jQuery 入门 147
 - 5.1.1 什么是 jQuery 147
 - 5.1.2 jQuery 的下载与使用 147
- 5.2 jQuery 选择器 149
 - 5.2.1 基本选择器 149
 - 5.2.2 层次选择器 150
 - 5.2.3 过滤选择器 152
 - 5.2.4 表单选择器 160
- 5.3 DOM 文档操作 162

5.3.1	元素遍历	162
5.3.2	元素属性操作	163
5.3.3	元素内容操作	167
5.3.4	元素样式操作	169
5.3.5	文档结点操作	170
5.4	事件和动画效果	177
5.4.1	常用事件	177
5.4.2	页面加载事件	178
5.4.3	事件绑定与切换	180
5.4.4	动画效果	182
5.5	jQuery 的 Ajax 操作	183
5.6	常用 jQuery 插件	187
5.6.1	日历插件	187
5.6.2	在线编辑器	189
	本章小结	190

第 6 章 ThinkPHP 框架 191

6.1	ThinkPHP 入门	191
6.1.1	什么是 ThinkPHP	191
6.1.2	案例——实现用户登录	193
6.2	ThinkPHP 目录结构	196
6.3	ThinkPHP 配置详解	197
6.3.1	入口文件的配置	197
6.3.2	配置文件的配置	199
6.4	ThinkPHP 实现 MVC	203
6.4.1	控制器 (Controller)	203
6.4.2	模型 (Model)	207
6.4.3	视图 (View)	220
	本章小结	229

第 7 章 ThinkPHP 框架进阶 230

7.1	ThinkPHP 路由	230
7.1.1	什么是路由	230
7.1.2	路由规则定义	233
7.1.3	案例——实现规则路由	236
7.2	数据过滤	238
7.2.1	输入过滤	238
7.2.2	数据验证	242
7.3	ThinkPHP 扩展功能	248
7.3.1	案例——上传文件	248
7.3.2	案例——制作缩略图	250
7.3.3	案例——实现分页	251
7.3.4	案例——生成验证码	254
	本章小结	257

第 8 章 电子商务网站项目实战 (上) 258

8.1	项目分析	258
-----	------------	-----

8.1.1	需求分析	258
8.1.2	系统分析	258
8.1.3	数据库设计	260
8.2	开发前准备	261
8.3	后台管理员模块开发	263
8.4	后台商品模块开发	267
8.4.1	商品分类	267
8.4.2	商品属性	273
8.4.3	商品添加	279
8.4.4	商品列表	286
8.4.5	商品回收站	294
8.5	会员管理模块开发	299
	本章小结	301

第 9 章 电子商务网站项目实战 (下) 302

9.1	前台首页模块开发	302
9.1.1	前台首页概述	302
9.1.2	分类导航实现	303
9.1.3	商品推荐实现	305
9.2	前台会员模块开发	306
9.2.1	会员注册功能	306
9.2.2	会员登录功能	308
9.2.3	会员中心功能	311
9.3	前台商品列表模块开发	317
9.4	前台商品信息模块开发	322
9.5	购物车模块开发	326
	本章小结	328

第 10 章 LAMP 环境 329

10.1	Linux 入门	329
10.1.1	什么是 Linux	329
10.1.2	Linux 的安装	330
10.1.3	Linux 目录结构	333
10.1.4	Linux 常用命令	334
10.1.5	vi 编辑器	342
10.1.6	网络配置	345
10.2	LAMP 环境搭建	346
10.2.1	环境搭建前的 准备工作	346
10.2.2	Apache 服务器的安装	347
10.2.3	PHP 的安装	349
10.2.4	MySQL 的安装	350
10.2.5	LAMP 后续配置	351
10.3	项目部署	353
10.3.1	phpMyAdmin 的安装	353
10.3.2	项目部署	353
	本章小结	354

第1章

PHP 操作数据库

学习目标

- 掌握 PHP 访问数据库的基本步骤，可对访问 MySQL 数据库进行描述
- 掌握 mysql 扩展，会使用 PHP 对 MySQL 数据库进行增、删、改、查操作
- 熟悉 mysqli 扩展，能够使用面向对象语法的方式操作 MySQL 数据库
- 掌握 PDO 扩展，学会使用 PDO 的统一接口对不同数据库进行操作

任何一种编程语言都需要对数据进行处理，PHP 语言也不例外。PHP 所支持的数据库类型较多，在这些数据库中，由于 MySQL 的跨平台性、可靠性、访问效率较高以及免费开源等特点，备受 PHP 开发者的青睐，一直以来被认为是 PHP 的最佳搭档。本章将针对 PHP 如何操作 MySQL 数据库进行详细讲解。

1.1 PHP 访问 MySQL

1.1.1 PHP 访问 MySQL 的基本步骤

PHP 提供了大量的 MySQL 数据库操作函数，可以方便地实现访问 MySQL 数据库的各种需要，从而轻松完成 Web 应用程序开发。本节主要讲解 PHP 访问 MySQL 数据库的基本步骤，具体如图 1-1 所示。

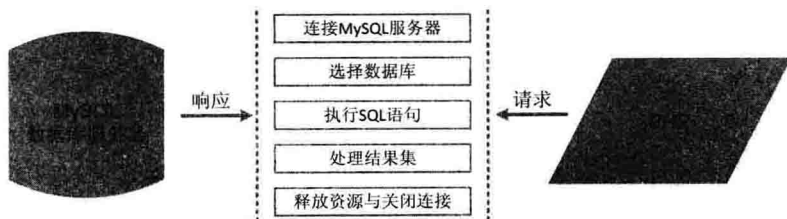


图 1-1 PHP 访问 MySQL 数据库的基本步骤

为了让读者更好地理解，接下来对图 1-1 中的步骤分别进行讲解。

1. 连接 MySQL 服务器

使用 `mysql_connect()` 函数建立与 MySQL 服务器的连接。有关 `mysql_connect()` 函数的使用会在本章 1.2.1 节中介绍。

2. 选择数据库

使用 `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。有关 `mysql_select_db()` 函数的使用会在本章 1.2.2 节中介绍。

3. 执行 SQL 语句

在所选择的数据库中使用 `mysql_query()` 函数执行 SQL 语句。有关 `mysql_query()` 函数的使用会在本章 1.2.3 节中介绍。

4. 处理结果集

执行完 SQL 语句后，使用 `mysql_fetch_array()`、`mysql_fetch_row()`、`mysql_fetch_object()` 等函数对结果集进行相关操作，有关处理结果集的相关函数会在本章 1.2.4 节中介绍。

5. 释放资源与关闭连接

处理完结果集后，需要使用 `mysql_free_result()` 函数关闭结果集，以释放系统资源。为了避免多用户连接造成系统性能下降甚至死机，在完成数据库的操作后，应使用 `mysql_close()` 函数断开与 MySQL 服务器的连接，有关 `mysql_free_result()` 和 `mysql_close()` 函数的使用会在本章 1.2.5 节中介绍。

1.1.2 PHP 相关扩展介绍

PHP 中提供了操作不同数据库的扩展，这里只介绍其中常用的 `mysql` 扩展、`mysqli` 扩展以及抽象层的 `PDO` 扩展。

1. `mysql` 扩展

在默认情况下，`mysql` 扩展已经安装好了，但没有开启。要想开启 `mysql` 扩展，需要打开 `php.ini` 文件，将文件中“`;extension=php_mysql.dll`”语句的分号删除，同时保存修改后的文件，重新启动 Apache 服务器即可启动 `mysql` 扩展。

在成功启动 `mysql` 扩展后，可以通过 `phpinfo()` 函数获取 `mysql` 扩展的相关信息，验证 `mysql` 扩展是否开启成功。在 Apache 服务器的根目录下创建一个 `phpinfo.php` 文件，并在文件中编写下列代码：

```
<?php
phpinfo(); // 输出 PHP 配置信息
```

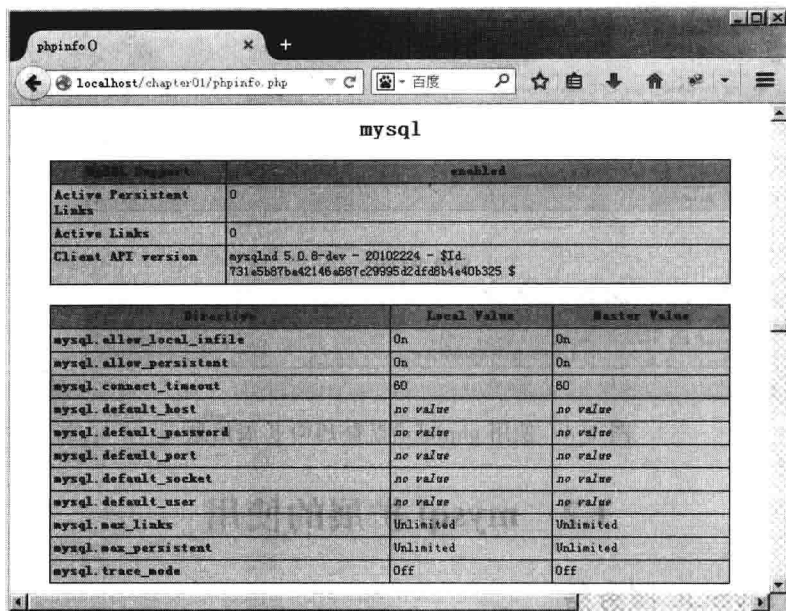
使用火狐浏览器访问地址“`http://localhost/chapter01/phpinfo.php`”，浏览器会输出 PHP 相关信息，其中包含 `mysql` 扩展的详细信息，具体如图 1-2 所示。

2. `mysqli` 扩展

`mysqli` 扩展是 `mysql` 的增强版扩展，它是 MySQL 4.1 及以上版本提供的功能。`mysqli` 扩展在默认情况下已经安装好了，需要开启时，在 `php.ini` 配置文件中找到“`;extension=php_mysqli.dll`”去掉分号注释即可。修改后重新启动 Apache，然后通过 `phpinfo()` 函数查看 `mysqli` 扩展是否开启成功，具体如图 1-3 所示。

3. `PDO` 扩展

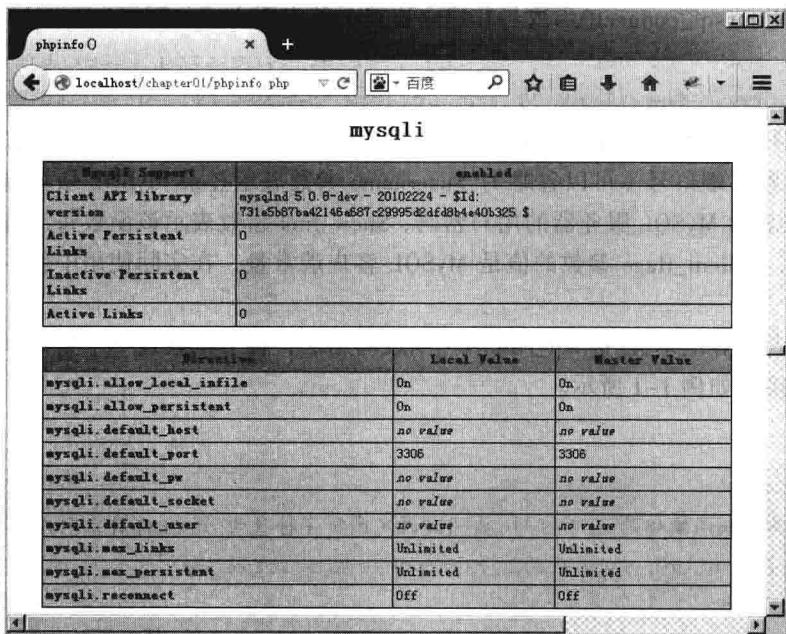
在早期的 PHP 版本中，由于不同数据库扩展的应用程序接口互不兼容，导致用 PHP 所开发的程序维护困难、可移植性差。为了解决这个问题，PHP 开发人员编写了一种轻型、便利的 API 来统一操作各种数据库，即数据库抽象层——`PDO` 扩展。



The screenshot shows the output of the phpinfo() function in a web browser. The title is 'mysql'. The first table lists general information: 'Enabled' is 'enabled', 'Active Persistent Links' is '0', 'Active Links' is '0', and 'Client API version' is 'mysqlnd 5.0.8-dev - 20102224 - \$Id: 731a5b87ba42146a687c29995a2df8b4e40b325 \$'. The second table lists configuration options with columns 'Directive', 'Local Value', and 'Master Value'.

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	no value	no value
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

图 1-2 使用 phpinfo()查看 mysql 扩展信息



The screenshot shows the output of the phpinfo() function in a web browser. The title is 'mysqli'. The first table lists general information: 'Enabled' is 'enabled', 'Client API Library version' is 'mysqlnd 5.0.8-dev - 20102224 - \$Id: 731a5b87ba42146a687c29995a2df8b4e40b325 \$', 'Active Persistent Links' is '0', 'Inactive Persistent Links' is '0', and 'Active Links' is '0'. The second table lists configuration options with columns 'Directive', 'Local Value', and 'Master Value'.

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	no value	no value
mysqli.default_port	3306	3306
mysqli.default_pw	no value	no value
mysqli.default_socket	no value	no value
mysqli.default_user	no value	no value
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off

图 1-3 使用 phpinfo()查看 mysqli 扩展信息

PHP 从 5.1 版本开始，在安装文件中含有 PDO，在 PHP 5.2 中默认为开启状态，但是若要启动对 MySQL 数据库驱动程序的支持，仍需要进行相应的配置操作。需要开启时，在 php.ini 配置文件中找到“;extension=php_pdo_mysql.dll”去掉分号注释即可。修改完成后重新启动 Apache，可通过 phpinfo()函数查看 PDO 扩展是否开启成功，具体如图 1-4 所示。

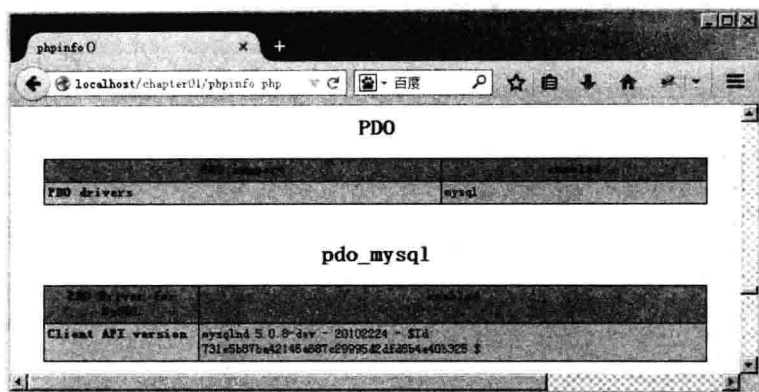


图 1-4 使用 phpinfo()查看 PDO 扩展信息

1.2 mysql 扩展的使用

1.2.1 连接 MySQL 服务器

在操作 MySQL 数据库之前,需要先与 MySQL 数据库服务器建立连接。在 PHP 的 mysql 扩展中通常使用 `mysql_connect()` 函数与其建立连接,其声明方式如下:

```
resource mysql_connect ([ string $server [, string $username [, string $password [, bool $new_link [, int $client_flags ]]]]] )
```

在上述声明中,参数 `$server` 的默认值是“localhost:3306”,其中 localhost 表示本地服务器,3306 表示默认端口号(可以省略)。`$username` 参数表示登录 MySQL 服务器的用户名,`$password` 参数表示 MySQL 服务器的用户密码,`$new_link` 参数表示该函数每次被调用时总是打开新的连接,`$client_flags` 参数的值是 MySQL 客户端常量,在实际使用中较少,具体参考 PHP 手册。

为了让读者更好地掌握 `mysql_connect()` 函数的用法,接下来通过一个案例来演示如何进行数据库的连接,如例 1-1 所示。

【例 1-1】

```
1 <?php
2 //当文件的默认编码是 utf-8 时,要同时设定网页字符集为 utf-8,防止中文乱码
3 header("Content-Type:text/html;charset=utf-8");
4 //连接数据库
5 $link = mysql_connect('localhost:3306','root','123456');
6 //判断数据库是否连接正确
7 if($link){
8     echo '数据库连接正确';
9 }else{
10     echo '数据库连接失败';
11 }
```

运行结果如图 1-5 所示。

在例 1-1 中，第 5 行代码用于连接数据库，其中 MySQL 服务器的主机名是 localhost，端口号是 3306，用户名是 root，用户密码是 123456。如果连接正确，则返回值类型是资源类型，否则返回值类型是布尔类型 FALSE。



图 1-5 例 1-1 运行结果

注意：在连接数据库发生错误时，会出现错误信息，但在上线项目中建议对错误信息进行屏蔽，并可以自定义错误提示，通常有如下两种方式：

- ① 在 mysql_connect() 函数前面添加符号 “@”，可以用于屏蔽这个函数出错信息的显示。
- ② 当需要自定义错误提示时，可以写成如下形式：

```
mysql_connect('localhost:3306','root','123456') or die('数据库服务器连接失败!');
```

在上述代码中，如果调用函数出错，将执行 or 后面的语句，其中 die() 函数用于停止脚本执行并向用户输出错误信息。建议在程序开发阶段不要屏蔽错误信息，避免出错后难以找到问题。

1.2.2 选择数据库

连接 MySQL 数据库成功之后，接下来使用 mysql_select_db() 函数选择数据库，其声明方式如下：

```
bool mysql_select_db ( string $database_name [, resource $link_identifier ] )
```

在上述声明中，参数 \$database_name 表示要选择的数据库名称，可选参数 \$link_identifier 表示 MySQL 连接，默认使用最近打开的连接；如果没有找到该连接，则尝试不带参数调用 mysql_connect() 来创建；如果没有找到并无法建立该连接，则会生成 E_WARNING 级别的错误。

为了让读者更好地掌握 mysql_select_db() 函数的用法，接下来通过一个案例来演示如何使用此函数选择数据库，如例 1-2 所示。

【例 1-2】

```
1 <?php
2 //设定字符集
3 header('Content-Type:text/html;charset=utf-8');
4 //连接数据库服务器
5 $link = mysql_connect('localhost:3306','root','123456');
6 $db_selected = mysql_select_db('itcast',$link);
7 //判断选择数据库是否成功
8 if($db_selected){
9     echo '数据库选择成功!';
10 }
```

运行结果如图 1-6 所示。



图 1-6 例 1-2 运行结果

在例 1-2 中,第 6 行代码表示使用名字为 \$link 的 MySQL 数据库连接,选择名字为“itcast”的数据库,并对返回的布尔类型结果进行判断,如果选择数据库成功,则显示提示信息“数据库选择成功”,否则没有任何输出。

1.2.3 执行 SQL 语句

完成数据库的选择后,就是对 SQL 语句的执行操作了,在 PHP 中,通常使用 `mysql_query()` 函数执行 SQL 语句,其声明方式如下:

```
resource mysql_query ( string $query [, resource $link_identifier = NULL] )
```

在上述声明中 \$query 表示 SQL 查询语句,但是查询字符串不应以“;”结束。\$link_identifier 是可选项,表示 MySQL 连接标识,若省略,则使用最近打开的连接。

需要注意的是,该函数仅对 SELECT、SHOW、EXPLAIN 或 DESCRIBE 语句执行成功时返回一个资源标识符,如果查询执行失败则返回 FALSE。如果 SQL 语句是 INSERT、DELETE、UPDATE 等操作指令,成功则返回 TRUE,否则返回 FALSE。

为了让读者更好地掌握 `mysql_query()` 函数的用法,接下来通过一个案例来演示如何使用此函数执行 SQL 语句,如例 1-3 所示。

【例 1-3】

```
1 <?php
2 //设定字符集
3 header('Content-Type:text/html;charset=utf-8');
4 //连接数据库服务器
5 $link = mysql_connect('localhost:3306','root','123456');
6 //选择数据库
7 $db_selected = mysql_query ('use itcast',$link);
8 //执行 SQL 语句
9 if(mysql_query('set names utf8')){// 告知 MySQL 服务器使用 utf-8 编码进行通信
10     $sql = 'show tables';
11     $result = mysql_query($sql);
12     var_dump($result);
13 }else{
14     echo '设定字符集失败';
15 }
```

运行结果如图 1-7 所示。



图 1-7 例 1-3 运行结果

在例 1-3 中,第 9 行代码用于判断设定字符集是否成功,若结果为 FALSE,则输出提示语句“设定字符集失败”,若结果为 TRUE,则继续执行 SQL 语句,从图 1-7 中可知结果为 resource 资源类型,说明执行 SHOW 操作成功。

注意:选择数据库的操作也可使用 `mysql_query('use 数据库名称')` 函数。

1.2.4 处理结果集

执行完 SQL 语句后,需要使用函数从结果集中获取信息,在 PHP 中常用的处理结果集的函数有 `mysql_fetch_row()` 函数、`mysql_fetch_assoc()` 函数、`mysql_fetch_array()` 函数以及 `mysql_fetch_object()` 函数。下面分别介绍这几个函数的具体使用方法。

1. `mysql_fetch_row()` 函数

首先了解一下 `mysql_fetch_row()` 函数,其声明方式如下:

```
array mysql_fetch_row ( resource $result )
```

从上述声明中可知,该函数的返回值是数组类型,参数 `$result` 表示资源型结果集。每执行一次该函数都将从结果集资源中取出一条记录放入到一维数组中,下标从 0 开始,并且内部数据指针自动指向下一条数据,直到没有更多行时返回 FALSE。

为了让读者更好地掌握 `mysql_fetch_row()` 函数的用法,接下来通过从 itcast 数据库中获取图书列表来演示如何使用该函数获取结果集,如例 1-4 所示。

【例 1-4】

```
1 <?php
2 //设定字符集
3 header('Content-Type:text/html;charset=utf-8');
4 //连接数据库服务器
5 $link = mysql_connect('localhost:3306','root','123456') or die('连接
   数据库失败!'.mysql_error());
6 //选择 itcast 数据库
7 mysql_select_db('itcast',$link);
8 //设置数据库编码格式为 utf8
9 mysql_query('set names utf8');
10 //执行 SQL 语句
11 $sql = "select * from book_list";
12 $result = mysql_query($sql,$link);
13 //显示图书的标题行
14 echo "<table border='1' cellpadding='1' align='CENTER' width='720'>";
```

```

15 echo "<tr height='50' bgcolor='#ccc'>";
16     echo "<td width='50' align='center'>ID</td>";
17     echo "<td width='210' align='center'>图书名称</td>";
18     echo "<td width='210' align='center'>出版日期</td>";
19     echo "<td width='50' align='center'>价格</td>";
20     echo "<td width='200' align='center'>作者</td>";
21 echo "</tr>";
22 //使用 mysql_fetch_row() 函数处理结果集
23 while($row = mysql_fetch_row($result)){
24     //取出结果集中相关内容
25     echo "<tr height='40'>";
26     echo "<td width='50' align='center'>".$row[0]."</td>";
27     echo "<td width='250' align='center'>".$row[1]."</td>";
28     echo "<td width='150' align='center'>".date('Y-m',$row[2])."</td>";
29     echo "<td width='50' align='center'>".$row[3]."</td>";
30     echo "<td width='200' align='center'>".$row[4]."</td>";
31     echo "</tr>";
32 }
33 echo "</table>";

```

运行结果如图 1-8 所示。



ID	图书名称	出版日期	价格	作者
1	《PHP程序设计基础教程》	2014-08	40	传智播客高教产品研发部
2	《网页设计与制作》	2014-08	45	传智播客高教产品研发部
3	《C#程序设计基础入门教程》	2014-09	45	传智播客高教产品研发部
4	《C语言开发入门教程》	2014-09	39.8	传智播客高教产品研发部
5	《Java基础入门》	2014-05	44.5	传智播客高教产品研发部

图 1-8 例 1-4 运行结果

在例 1-4 中，第 5 行代码中 `mysql_error()` 函数用于返回 MySQL 连接的错误信息，如果没有出错则返回空字符串。第 13~21 行代码定义了图书列表的标题行，由于 `mysql_fetch_row()` 函数每次只能取出一条数据，所以要和 `while` 循环配合使用。从第 26~30 行代码可以看出，`mysql_fetch_row()` 函数只能通过数字索引下标的形式输出数据。

2. `mysql_fetch_assoc()` 函数

`mysql_fetch_assoc()` 函数也可以获取结果集，与 `mysql_fetch_row()` 函数的唯一区别是通过字段名称来获取数据，其声明方式如下：

```
array mysql_fetch_assoc ( resource $result )
```

在上述声明中，`array` 表示该函数的返回值类型是数组，参数 `$result` 表示资源型结果集。

每执行一次该函数都将从结果集资源中取出一条记录放入到一维数组中，并且内部数据指针自动指向下一条数据，直到没有更多行时返回 FALSE。

为了让读者更好地掌握 `mysql_fetch_assoc()` 函数的用法，接下来使用此函数来获取结果集实现例 1-4 的功能，具体如例 1-5 所示。

【例 1-5】

```

1 <?php
2 //设定字符集
3 header('Content-Type:text/html;charset=utf-8');
4 //连接数据库服务器
5 $link = mysql_connect('localhost:3306','root','123456') or die('连接数
   据库失败!'.mysql_error());
6 //选择 itcast 数据库
7 mysql_select_db('itcast',$link);
8 //设置数据库编码格式 utf8
9 mysql_query('set names utf8');
10 //执行 SQL 语句
11 $sql = "select * from book_list";
12 $result = mysql_query($sql,$link);
13 //显示图书的标题行
14 echo "<table border='1' cellpadding='1' align='CENTER' width='720'>";
15 echo "<tr height='50' bgcolor='#ccc'>";
16     echo "<td width='50' align='center'>ID</td>";
17     echo "<td width='210' align='center'>图书名称</td>";
18     echo "<td width='210' align='center'>出版日期</td>";
19     echo "<td width='50' align='center'>价格</td>";
20     echo "<td width='200' align='center'>作者</td>";
21 echo "</tr>";
22 //使用 mysql_fetch_assoc() 函数处理结果集
23 while($row = mysql_fetch_assoc($result)){
24     //取出结果集中相关内容
25     echo "<tr height='40'>";
26     echo "<td width='50' align='center'>".$row['id']."</td>";
27     echo "<td width='250' align='center'>".$row['bname']."</td>";
28     echo "<td width='150' align='center'>".date('Y-m',$row['pub_
       time'])."</td>";
29     echo "<td width='50' align='center'>".$row['price']."</td>";
30     echo "<td width='200' align='center'>".$row['author']."</td>";
31     echo "</tr>";

```