HZ BOOKS
华章科技

Web开发
技术丛书

# PHP与MySQL高性能应用开发

杜江 著

PHP and MySQL
High-Performance Applications Development

作者拥有15年研发经验，资深PHP专家和架构师，曾担任赶集网和今日头条技术总监，好乐买和正和岛的CTO

围绕高性能、可扩展性、可伸缩性、可靠性等与PHP应用性能相关的主题展开，同时还涉及PHP编程思想、底层原理、编程技巧、开发规范等重要内容

机械工业出版社
China Machine Press

# PHP与MySQL高性能应用开发

PHP and MySQL
High-Performance Applications Development

杜江 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

PHP 与 MySQL 高性能应用开发 / 杜江著 . —北京：机械工业出版社，2016.8
（Web 开发技术丛书）

ISBN 978-7-111-54796-9

I. P… II. 杜… III. ① PHP 语言－程序设计 ② 关系数据库系统 IV. ① TP312
② TP311.138

中国版本图书馆 CIP 数据核字（2016）第 214629 号

# PHP 与 MySQL 高性能应用开发

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：李 艺 责任校对：董纪丽
印 刷：北京市荣盛彩色印刷有限公司 版 次：2016 年 9 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：19.25
书 号：ISBN 978-7-111-54796-9 定 价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

# Preface 序

曾经我与你一般，年少时期，对人生只知努力，却不知何往，只得上下求索，东寻西觅。于是求知识、读文字、写代码、做架构，时至而立之年方初识端倪。几年来亲历创业，一路走来有技术的积累，亦有技术外的磨砺。比照更多的同路者，做自己最擅长的才更有力量。

当今社会，如你我这样依靠技术成就理想的开发者，共同特征是吃苦耐劳，也有一些完美主义。我们在互联网上获取大量知识，而上面的信息多数可受其益，但陈旧错漏之文仍有，条理逻辑亦差强人意，难免蒙受其弊。因此，纸质图书阅读对于开发者来说仍有必要。

开发类书籍创作大都不是轻松的工作，但我心中一直存有一份责任，那就是让更多的朋友能够解惑并能目标明确地向前，让“Open & Share”的开源理念得到更多理解，这也是我能够坚持的初心。

每晚在称为“中国硅谷”的中关村软件园区，从窗外看着外面灯火通明的百度大厦，还有很多人在加班工作。也有很多技术类的创业者，他们都在执着地用自己的双手浇灌未来的理想之花。每当此时，耳畔听着西山风声，手中的键盘声响起，眼前屏幕的文字跃动，是另一种喜悦。

创新来源于每天的思考与实践，梦想方能不绝于缕。互联网的新技术每天都在发展，关于 LAMP/LNMP 开发、高性能、高扩展的话题也一直在更新发展中。

本书持续写了两年有余，其中针对 PHP 升级，部分内容也同步做了更新，特别是 PHP7 的发布。书中内容符合 PHP5.6 及以上版本。希望本书能够帮助你避免在开发时遇到坑，或者简单问题复杂化，进而提高编码效率。

人生处处是战场，作为开发者的我们，需要每日积跬步行千里，不断实践让自己更加优秀。既然你已经准备好了，就让我们充满感激和动力，出发！

杜江（别名：洛逸）

# 前　言 Preface

在过去的十几年间，LAMP 开源技术推动着互联网开发，有 4000 万以上的网站在使用 LAMP&LNMP 技术平台驱动。

在互联网和移动互联网平台中，其中 Facebook、开心网、新浪网、Yahoo!、百度、腾讯、搜狐、网易及各个视频网站全部或大部分使用的是 LAMP&PHP 技术。

与其说 Web 的伟大创新，不如说是创新者的智慧，还有 PHP 技术的鲁棒性与相对于其他语言的快速、灵活、敏捷性，给互联网——这个亦庄亦娱的行业带来强大的动力。

近年来，PHP 与互联网正一起创造着流行。2000 年前后，PHP 应用于 Yahoo! 网站，国内门户网站腾讯、新浪、优酷、凤凰及众多在线网络游戏厂商等也都全部或部分使用 PHP 技术。同时，PHP 也为互联网的新兴网站创造了一个又一个神话。

Craigslist.org 是在全美第 6 名、全球第 20 名的分类信息网站，每月有 1000 万独立访问量和 30 亿页面浏览量，它使用 LAMP 技术开发，国内类似的网站如赶集网、百姓网也全部使用 PHP 技术。

维基百科（Wikipedia），也称为自由的百科全书。它是由全球不同民族、不同语言共同编撰的一部网络百科全书，由 PHP 开发，并以 Mediawiki 开放源代码。

Yelp 是美国最大的店铺点评网站，相当于中国的大众点评网，2009 年婉拒了 Google 近 6 亿美元的收购要约，目前已成为消费者购买与体验商品的最佳社区，国内有安居客、蚂蚁、小猪短租、好车无忧等类似网站也全部使用了 PHP 技术。

SNS（Social Networking System）巨头 Facebook，是全球最大的 LAMP 网站，目前已有超过 15 亿用户，超过 Google。目前这个全球最火热的社区，已演化为人们生活不可缺少的工具。国内类似的 SNS 网站，如开心网、同学网、腾讯朋友等全部使用 PHP 开发。而 Facebook 的社交开发商（Social Game Developer），如 Zynga 等社交游戏厂商也应

用了 PHP 开发，因为 Facebook 的巨大应用量而赚得盆满钵满。

随着 Twitter 的流行，使国内微博网站愈加火爆，如新浪微博、腾讯微博等网站全部使用了 PHP 开发。而热门、模式创新的网站，非 Foursquar.com 和 Groupon.com 莫属，它们分别是基于位置的地图服务和团购商品的服务，而这些网站的中国版如美团、团宝等网站使用的也是 PHP 技术。

PHP 在电子商务 / 社交化电子商务领域，以及企业软件上同样大展身手，如淘宝前端使用 PHP、Prestashop、ShopEx、Magento、eCart、osCommerc 等。可以预见的是，在未来还会有新的互联网神话出现，而加速这些网站前进的 PHP 将继续担当主力。

还有企业级开发领域，如 Zend、SugarCRM、DotProject 等，也在使用 PHP 来实现云计算等企业级开发领域。而且在当今如火如荼的移动互联网以及网页游戏开发领域，还有 PHP for Android 等框架来帮助开发者实现本地化 App 开发的想法，而且 App 的后面也可使用 PHP 来提供 API 服务接口。

PHP 并非万能，但凭借它实用高效的优势，在 Web 开发领域，PHP 和 MySQL 无疑是“世界上最好的语言”。

现今，国内的各个互联网公司均面临两大问题和挑战：第一，高流量、高负载的商务应用使 Web 系统不堪重负；第二，价格高昂的带宽、硬件、商业软件等成本高居不下，越来越多的互联网公司开始拥抱开源的 LAMP/LNMP 平台。

同时，PHP 也在不断更新。我们需要有众多热爱编程开发，有扎实的基础以及丰富的实际编程经验，有创新、有思想的工程师，加入到 PHP 开发的行列中。

## 为什么要使用本书

如果你已经看过市场上很多初级类书籍，却还在寻找 PHP 编程思想、底层原理、编程技巧、可伸缩性、可靠性、开发规范等内容，那么就请使用本书，相信可以获取更多新鲜与深入的主题。

本书为读者带来的是一系列实用的、进阶的“干货”，相信定会给你的程序生涯和未来发展带来帮助。

书中主要介绍如下主题：

- 解惑：掌握 PHP 编程中的“长尾”细节。
- 深入：PHP 面向对象高级开发。
- 浅出：PHP 开发中的调试与技巧。

- 编程之道：透彻理解面向对象开发思想与设计模式。
- 更快：使用 OpCode 缓存。
- 扩展：memcached 及扩展应用。
- 搜索：Sphinx 全文搜索引擎。

为了提供更好的实用性，本书除了详解 PHP 中的深度开发外，还提供了相应的代码实例。读者可登录 21CTO（www.21cto.com）本书相关页面下载。

## 本书写给谁

本书适合 PHP 中级开发及以上资质的读者，需要读者充分了解 PHP 技术，可结合其他书籍进行同步阅读。

本书读者对象可为 PHP 研发工程师、软件架构师、系统架构师。本书也可作为 IT 运维人员、DBA、计算机专业本科以上学生的参考用书。

## 本书特点

书中讲解了 PHP 5.6 以上及 PHP7.02 版本的新特性，涵盖了目前大中型网站使用的研发技术，包括扩展、伸缩、负载、优化等，以及实际研发中的解决方案。本书不只停留在代码应用层，还包括架构方面的方法与思路，相信会帮助读者更好掌握 PHP。

## 致谢

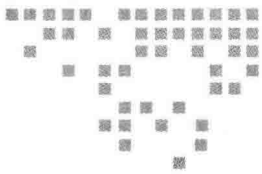
感谢机械工业出版社杨福川、高靖雅和李艺，以及曾经并肩战斗的朋友，是你们的鼓励才能使本书得以展现给各位。PHP 由 PHP 开发小组和众多的 PHPer 共建。同样，本书也得到了很多同仁的支持，在此一并致谢！

## 社区支持

如果你从本书中发现错误或漏洞，或者发现一些有价值和感兴趣的内容，可登录本书的技术支持平台：21CTO（www.21cto.com）与笔者进行交流。

同时，欢迎大家提出宝贵意见，以便在本书再版时为读者带来更好的体验。

# Contents 目　录



# 第 1 章 Chapter 1

# PHP 解惑

和其他语言相比，PHP 给人的印象是入门简单的语言。当你的技术能力达到一定阶段时，会发现情况并非如此。PHP 采用“极简主义”，就是以入门容易为准则设计的，在十几年的持续发展历程中，它早已成为一个开源领域的语言且具备现代语言特性的平台之一，在 Web 开发领域，我们相信 PHP 就是“世界上最好的语言”。

人无完人，语言也一样。天下事物都需要花大量精力去研究实践，深入下去不是易事，了解越多越敬畏。况且 Web 开发又是个严谨创意，如不能通透理解隐藏在后面的深层机制，就有可能损害应用的性能，导致低级错误的发生。

互联网产品的特性是小步快跑，快速迭代。这就经常需要我们直接开发，为快速实现功能而忽略一些性能、降低代码质量，但上线后一定要对代码进行整理、优化与修正。事实上，有的开发者从事开发若干年，却未必会对一些技术原理深究，加上网上大量的开源代码，借 Google、Github 等发扬拿来主义，复制粘贴未经推敲的代码，似乎没花太大力气就完成了任务。由于不同的架构设计，没有经过严谨的代码审核，这样的代码怎么能保证产品正常运行？

古人有这样一句话——“勿以浮沙筑高台”，即不要在浮沙上面建筑高台。基础不扎实，台子搭得再高也会倒掉，没有坚实的基础，是无法做好开发的。为保证开发的网站平台健壮，使平台能够承载更高的流量，需要理解、领悟更多的技术点，才能写出高质量、高扩展、高性能的代码。

## 1.1 省略结束标签的便利性

一个优秀的程序员会在编码前习惯把 PHP 标签成对写完，再写功能逻辑——我也不例外，不过有一次忘记了写结束标签，却发现也能正常运行，当时感觉很奇怪，还以为是神奇的 PHP 高度容错的结果。

其实对于 PHP 编译器来说，脚本的结束标签“ ?>”是可选的，在写程序时你可以忽略它。你或许碰见过：在使用 include()、require() 或输入输出缓冲函数时，页面顶部有时多空行或者出现“ header had send”之类的错误信息，这类问题与结束标签有关。省略结束标签适合纯 PHP 文件，如果是 PHP 与 HTML 混合开发，则不可省略。

忽略结束标签不仅能少写两个字符，还让我们的开发更顺利，何乐而不为。

## 1.2 emtpy、isset、is_null 的区别

变量在所有计算机语言中均有提供，它用来保存数值、文本、对象等内容。我们可以把变量看作一个有名称的桶，里面放着一个值，这个值可以是数字、字符串或对象，以及包含你想到的任何合法的内容。

PHP 提供了 3 个用于测试变量值的函数，分别是：isset()、empty() 和 is_null()。这几个函数均返回布尔值，有时使用不当会造成意想不到的结果，需要详细说明。

比如，用 isset() 和 empty() 返回的结果是相反的，但有时却并非一直如此，下面我们一起来了解这几个函数的具体区别。

isset() 用来检测一个变量是否已声明且值不为 NULL。换句话说，只能在变量值不是 NULL 时返回真值。

empty() 用来检测一个变量是否为空，也就是说有如下情况时返回真值：变量是一个空字符串，false，空数组 [array()], NULL，0，''，以及被 unset 删除后的变量。


*在 PHP5.5 之后，empty() 函数可以接受任意类型的表达式。*

正确地检查一个变量是否为空，可使用如下格式：

```
if(empty($approve)){
    //etc
}
```

这种形式可适用在 PHP 的任意版本中。如果你用的是 PHP5.5 以上版本，可以使用

如下格式：

```
if(empty(0)){
    //etc
}
if(empty(CreateNew())){
        //etc
}
```

以上格式在 PHP5.5 以上版本中均可以使用，如果小于该版本会返回解析错误。

is_null() 函数用来判断变量内容是否是 NULL 值，即返回真值的条件仅为变量是 NULL 时。值得一提的是，is_null() 是 isset() 函数的反函数，区别是 isset() 函数可以应用到未知变量，但 is_null() 只能针对已声明变量。

我们用一张表格来汇总这些函数返回值的不同之处（表 1-1），表中空白表示函数返回布尔值假（false）。

**表 1-1　测试函数返回值的区别**

| 对比项 | | | |
|---|---|---|---|
| 变量值（$var） | isset($var) | empty($var) | is_null($var) |
| " "（一个空字符串） | bool(true) | bool(true) | |
| " "（空格） | bool(true) | | |
| FALSE | bool(true) | bool(true) | |
| TRUE | bool(true) | | |
| array()（一个空数组） | bool(true) | bool(true) | |
| NULL | | bool(true) | bool(true) |
| "0"（0 是一个字符串） | bool(true) | bool(true) | |
| 0（0 是一个整型值） | bool(true) | bool(true) | |
| 0.0（0 是一个浮点值） | bool(true) | bool(true) | |
| var $var;（一个变量声明，但是没赋值） | | bool(true) | bool(true) |
| NULL byte（"\0"） | bool(true) | | |

## 1.3　布尔值的正确打开方式

关于布尔值，在 PHP 中可以这么来写：

```
<?php $flag = True; ?>
<?php $flag = TRUE; ?>
```

```
<?php $flag = true; ?>
```

有点儿像孔乙己的“茴香豆”写法，这 3 段代码都可以正常运行。但是，哪个最好？哪个是正确的？在 PHP 中，常量规定为大写，第二行代码显然是正确的。

下面我们再来看一下比较语句。比较常用于两个变量之间，但是，也会有这样的代码：

```
<?php
if($price = $cart->price){
    echo 'function return TRUE';
}else{
    echo 'function return FALSE';
}
?>
```

可以看到，这段代码也没有错，但不怎么容易理解。仔细看，这个分支里面的表达式是一个变量跟一个对象方法的赋值，并不是一个布尔值运算，很容易把人引入不正确的思路。

这种方法尽量不要用。正确的写法可以是这个样子的：

```
$user_id == $user->getUserId()
```

## 1.4 变量作用域实践

我们知道，在 PHP 中定义一个变量后，在脚本任意位置都可以存取访问，这被称为“全局变量”，而定义在函数或类的方法中的变量只可以在函数内部访问，这叫作“局部变量”。

使用局部变量可以使源代码易于管理，试想如果所有的变量都是全局的，任何位置都可访问、修改它的内容，如果变量重名就可能发生“污染”。通过声明局部变量来限制一个变量的存取范围，可以让代码模块化，易调试，让应用运行更健壮。

下面我们就来看看如何使用全局变量和局部变量，如代码清单 1-1 所示：

**代码清单1-1 使用全局变量与局部变量**

```
<?php
$globalName = "老杜";
function getvar() {
    $localName = "Raymond";
    echo"Hello, $localName!<br>";
```

```
}
getvar();
echo "The value of \$globalName is: '$globalName'<br />";
echo "The value of \$localName is: '$localName'<br />";
?>
```

该脚本运行后将显示如下内容：

```
Hello, 老杜!
The value of $globalName is: 'Raymond'
The value of $localName is: ''
```

在上面的代码中，我们一共创建了两个变量：其中 $globalName 是全局变量，它没有在任何函数体里；另一个是名为 $localName 的局部变量，是在 sayHello() 函数里内部定义的。

程序运行时先是调用 sayHello() 函数，显示的是“hello,Raymond！”，接下来用 echo 显示两个变量，分别是 $globalName 和 $localName。由于 $globalName 是定义在函数之外的全局变量，在脚本任何位置都可以访问，因此显示为“ Raymond”。而 $localName 定义在 sayHello() 函数内部，只能在函数内访问。脚本中使用 echo 来访问这个局部变量，而 PHP 不允许外部访问此局部变量。因此运行时，PHP 认为程序要创建一个新的全局变量 $localName，并将默认值初始化为空，所以显示的时候是空白的。

PHP 允许函数内部可访问外部全局变量，只需在函数中使用 global 关键字即可。我们来看代码清单 1-2：

**代码清单1-2　使用全局变量与局部变量**

```
<?php
$globalName = "老杜";
function sayHello() {
    $localName = "Harry";
    echo "Hello, $localName!<br />";
    global $globalName;
    echo "Hello, $globalName!<br />";
}
sayHello();
?>
```

该段脚本会输出下面的内容：

```
Hello, Harry!
Hello, 老杜!
```

由于在 sayHello() 函数里使用了 global 来声明 $globalname 为全局性质，因此它的内容被打印了出来。

## 1.5 多维数组排序

使用 PHP 开发应用，几乎就是一直跟数组打交道。PHP 数组的强大和灵活性能够解决大部分应用的问题。在数组编程中，常用的有 sort()、ksort() 等相关函数，使用它们就可以很方便地处理一维数组，比如按键值降序和升序排列。

这些函数不能用于多维数组，但是在开发中常常是对多维数组排序处理。下面我们定义一个二维数组，如代码清单 1-3 所示：

**代码清单1-3 定义一个标准二维数组**

```
<?php
$a = array(
        array("sky", "blue"),
        array("apple", "red"),
        array("tree", "green")
    );
?>
```

这是一个简单的二维数组，数组的元素也是数组。我们可能需要对 userid 这个键排序，或者按汉字或英文字符排序。

为了给多维数组进行排序，我们需要自定义排序函数，然后再调用 sort()、usort()、ksort() 这些函数，让这些函数使用自定义函数。

uasort 函数接受两个参数，并且返回一个值表示哪个参数应该排在前面。负数或 FALSE 意味着第一个参数应该排在第二个参数之前。正数或者 TRUE 表示第二个参数应该排在前面，如果值为 0，则表示两个参数相等。

下面，我们对前面的数组第一个键进行排序，代码清单 1-4 是一个自定义函数。

**代码清单1-4 将数组按键值排序的自定义函数**

```
function my_compare($a, $b) {
if ($a[1] < $b[1]) {
        return -1;
    }else if ($a[1] == $b[1]){
        return 0;
    }else{
        return 1;
```