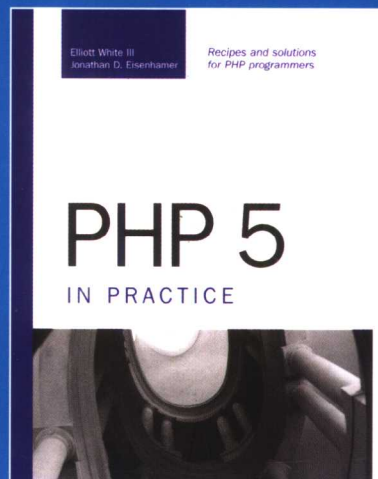


SAMS

From Programmer, For Programmer

- 囊括PHP编程策略与技巧的实战手册
- 为PHP开发人员提供实际解决方案
- 提供大量详尽的代码，可以直接从网络下载

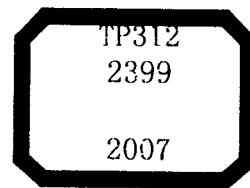


PHP 5 in Practice 中文版

[美] Elliott White III Jonathan D. Eisenhamer 著
王军 译

人民邮电出版社
POSTS & TELECOM PRESS

From Programmer,
For Programmer



PHP 5 in Practice 中文版

[美] Elliott White III Jonathan D. Eisenhamer 著
王军 译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

PHP 5 in Practice 中文版 / (美) 怀特 (White, E.), (美) 艾森汉默 (Eisenhamer, J. D.) 著; 王军译. —北京: 人民邮电出版社, 2007.7

ISBN 978-7-115-16016-4

I. P... II. ①怀...②艾...③王... III. PHP 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 043617 号

版 权 声 明

Elliott White III, Jonathan D. Eisenhamer: PHP 5 in Practice

ISBN: 0672328887

Copyright © 2007 by Sams Publishing.

Authorized translation from the English language edition published by Sams.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

PHP 5 in Practice 中文版

◆ 著 [美] Elliott White III Jonathan D. Eisenhamer
译 王 军
责任编辑 李 际

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
新华书店总店北京发行所经销

◆ 开本: 800×1000 1/16
印张: 25.75
字数: 502 千字 2007 年 7 月第 1 版
印数: 1 - 5 000 册 2007 年 7 月河北第 1 次印刷

著作权合同登记号 图字: 01-2007-0984 号

ISBN 978-7-115-16016-4/TP

定价: 49.00 元

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

内容提要

本书针对开发人员在日常工作中可能遇到的问题，利用最新版本的 PHP 5 提供了解决方案，内容涉及数据库操作、动态创建 Web 页面、数据检验、电子邮件、用户验证、错误处理与调试等多个方面。全书分为三部分。第一部分介绍了 PHP 5 的所有特性，语言简洁明了、通俗易懂。第二部分为应用实战，是对前面知识的具体应用和常见问题的实际解决方案，采用面向问题的写作风格，言之有物，使读者很容易检验前面的知识并解决具体项目中的问题。第三部分为附录，介绍迁移到 PHP 5 时面临的问题和标准库，以及常见错误的应对策略。

本书能够帮助有一定编程经验的 PHP 开发人员解决开发过程中遇到的问题，书中提供了大量详尽的代码，并重点说明了其中的关键之处，是一本不可多得的 PHP 参考书。

作者简介

Elliott White III（喜欢被称为 Eli）是 **digg.com** 的高级 Web 程序员，具有 11 年的 Web 程序开发经验。他出席过全球范围内多个 PHP 会议，并且参与了多本图书的编写工作。他的电子邮件地址是 **php@eliw.com**。

Jonathan Eisenhamer 作为专业程序员已超过 20 年，主要工作于天文学领域。在这段期间里，他获得了计算机科学专业的硕士学位，并且在天文学、天文计算和计算机技术领域与他人共同发表多篇论文。**Eisenhamer** 目前就职于空间望远镜科学研究所(STScI)的公共领域办公室，是 Web 及输出服务组的主管与首席程序员，这个部门的职能是专门为公众提供来自哈勃望远镜的最新成果。

致 谢

首先，我要感谢我的妻子 Heather，她同意我接受这个任务并且自始至终地支持我。如果没有她的爱与帮助，我不可能在拥有一份全职工作的情况下完成这本书。还要感谢我的儿子 Ramsey，请原谅我为了这本书而不能陪你一起玩耍。在我开始考虑编写这本书时，你甚至还没有出生；在本书开始编写时，你才八个月大；本书完成时，你已经一岁了。我真心地希望能够尽快弥补这些。

我还要感谢 George Schlossnagle，我们在一次会议上相遇时，是你建议我编写这样一本书，如果不是你的建议，这一切根本不可能发生。感谢 Sams Publishing 的编辑 Shelley Johnston，感谢你对我的信任，以及与我共同完成准备工作。当然，我不会忘记感谢本书的另一个作者 Jonathan，没有他的帮助，本书不可能按时完成，其质量也会大打折扣。他的观点和概念使得本书达到了如今的水平。

最后，我要感谢 SCA 的所有朋友，感谢他们带来的欢乐、刺激和友谊，以及经常能让我暂时离开包围我生活的世俗的技术世界。如果没有 SCA 15 年对我的影响，我不会有如今的成就。

——Eli

首先，也是最重要的，我要衷心地感谢我的家庭：我的妻子 Bonnie 和我的儿子 Jonathan，你们的支持始终如此坚定，你们的幽默和体贴是我生命中最大的快乐。还要感谢 Eli，感谢你的友谊，以及让我与你共同完成这本书。你的技术总是能够提高我的能力，而你的智慧总是会让我耳目一新。

我还要感谢两个无意中帮助我选择前进道路的人（这是很久以前的事情了）：John Hill 和 Alan Koski。感谢 John 给了我生命中宝贵的机会，并且向我展示了发明的真正含义。感谢 Alan 耐心地听取一个新程序员的询问，并且让我比在课堂上更深入地理解了计算行为。

——Jonathan

前言

有些书可供一览，有些书可以囫圇吞枣，有不多的几部书应当咀嚼消化。也就是说，有些书只要读读它们的一部分就够了，有些书可以全读，但是不必过于细心地读；还有不多的几部书应当全读、勤读，而且用心地读。

—— 弗兰西斯·培根爵士（1561 ~ 1626）

本书的范围与目标

本书不是针对所有程序员编写的，实际上它并不是一本 PHP 的入门图书，它适合中高级 PHP 程序员阅读。但是，任何水平的程序员都可以从本书中获益，因为只需要一个晚上的时间，任何人都可以通过 <http://php.net> 上的在线 PHP 文档获得相关的基础知识。

从这个意义上来说，在线 PHP 文档具有很高的价值，不应该被忽视。本书在讨论多个主题时都使用它作为更详细内容的参考。而本书所要提供的，正是在线 PHP 文档所缺少的。

本书提供了大量解决问题的方法。PHP 程序员在工作中可能会遇到这些问题，而在这里就能够找到答案（包括完整的代码），从而发现解决实际问题的某种方法。你是否需要计算贷款利息、验证信用卡号码、使用另一种语言对 Web 页面进行本地化、使用数据库甚至是创建自己的 Web 服务器？

本书涉及了上述全部问题，提供了代码清单和极其详细的解释，准确地说明了代码的作用。

本书可以在两个方面发挥作用。它可以作为直接的解决方案。在编写代码的过程中，程序员会发现需要某个函数来处理特定的数据，这时就可以在本书中找到相应的代码，并且直接使用它。另外，它可以作为一个指导，程序员可以查看书中的代码，使用其中的知识来编写类似的代码。

本书不必按顺序从头读到尾，读者可以从书中任何位置开始阅读，并且不会感到任何阻碍。但对于 PHP 的新手来说，可能需要阅读一些特定章节的内容来获取必要的知识。阅读某些章节并且理解其中的内容之后，一定会对掌握相关主题有所帮助。

由于本书致力于成为一本参考书，因此每一章开始都有一个“快速提示”，其中列出了适用于本章主题但不包含在本章之中的函数，帮助读者确定相应的参数名称。这个列表并不是特别详尽的，它着重于与主题相关的函数及最常用的函数。列表中也并没有进行详细说明，只提供了基本参数及常见结果。但书中为每个函数都提供了一个 URL，指向完整的文档。“快速提示”可以看做是与主题相关的最常用函数的简明参考。

代码下载

访问 <http://eliw.com/books/>，单击“Download Zip File”链接可以直接下载书中的范例代码，而不必自己将代码输入计算机。

编码标准

本书包含的所有代码都使用了统一的编码标准，这也是作者所认为的 PHP 在当前最好的标准。这主要是基于 PEAR 编码标准 (<http://pear.php.net/>)，但增加了一些内容及修改。

下面的小节将介绍这些标准，读者会了解到为什么以这种方式编写代码，并且很可能在工作中选择使用这些标准。

在使用 E_STRICT 和 E_ALL 的情况下不产生错误

本书的代码都可以在使用 E_STRICT 和 E_ALL 的情况下不产生任何 PHP 错误，这意味着在编写这些代码时花费了很大精力来确保在使用变量之前进行声明，以及其他类似的问题。

引号的使用变化

在本书的代码里，有些字符串是用单引号 (') 标识的，而有些是用双引号 (")，这是特意的。PHP 解析由斜杠 (\) 命令及内嵌变量包围的字符串。这种规范有时是很方便的。但它会占用额外的计算机资源（很少一点）。但如果用单引号标识纯字符串，就可以让 PHP 引擎节省一点时间，因为它提前就知道这是个字符串，而不必再对它进行解析。

更重要的是，明确地使用引号有助于其他程序员阅读代码，当他们看到由单引号标识的字符串时，就知道可以跳过它，而不必详细查看其中内嵌的变量。

注释与空白

对于一个程序员来说，最重要的事情之一是充分地使用空白和编写足够的注释，否则不仅其他程序员在查看这些代码时会产生困难，就是编写代码的本人过一段时间之后看这些代码也会感到吃力。

本书以双斜线 (//) 作为注释的标识符。当然，还可以使用 C 标准的 /* 作为注释开始，以 */ 作为注释结束。本书的作者并不提倡使用这种方式，因为如果代码编辑器不能以不同颜色显示代码与注释，就会让注释与代码不易区分。另外，这样还容易产生错误，无意中会注释掉一大段代码。

虽然 PHP 也支持 Perl 风格的井字符 (#) 作为注释标识，但 PEAR 标准不允许使用它。

缩进与大括号

本书的代码遵守 PEAR 标准，其中对于大括号的使用与 C 标准是一样的。最主要的标准是，{ 应该与需要它的命令位于同一行，而 } 应该与同一个命令具有相同的缩进，而且如果它是最后一个右大括号，就应该单独位于一行。用一个具体的例子更容易说明这个标准：

```
if ($test) {  
    while ($num < 5) {  
        $num++;  
    }  
} elseif ($test2) {  
    echo 'here';  
} else {  
    echo 'there';  
}
```

作者认为，如果要对上面代码里的 `elseif` 语句添加注释，可以把注释放到代码行之下，也可以把注释插入到 `elseif` 与 `}` 之间。

本书中的代码以 4 个空格为单位进行缩进。根据 PEAR 标准，缩进应该是实际的空格，但作者认为在 Tab 间隔设置为 4 个空格时，使用 Tab 键是个不错的替代方法。

小括号

作者认为，在语句中使用了多少小括号都不过分。虽然运算优先级的规则是很直观的，但只依靠这些规则不仅会增加产生错误的概率，而且会增加代码的混乱度，使代码

的作用更难以理解，比如下面的代码：

```
if (isset($bob) && $bob != 9) { echo 'Yes'; }
```

虽然上面的代码能正常工作，但是不如下面这样写更清楚：

```
if (isset($bob) && ($bob != 9)) { echo 'Yes'; }
```

为了更好地说明这个问题，可以看一看下面这行代码：

```
if ($x + $y / $z < 9 or $a && $n ? $m : $o == $b || $c and $d) { echo 'Y'; }
```

谁能准确地说出这行代码的含义？操作符优先规则很可能会导致有些操作顺序并不像人们预期的那样，从而使程序产生错误。然而通过有效地使用小括号，就可以清楚地展示程序员的意图（也更便于判断对错）。为此，上面的语句可以写成如下所示，它实际的效果与上面的语句也有所区别：

```
((($x + $y) / $z < 9) or (($a && (($n ? $m : $o) == $b)) || $c)) and $d)
```

虽然这行代码被很多小括号划分为多个部分，但通过仔细分析就能准确地理解它在进行什么比较。否则在它出现问题时，调试过程将会是个噩梦。

PHP 短标记

PHP 有一个配置选项，允许使用“短标记”。当这个选项被打开时（默认），程序员就可以使用两个速记技术来开始一段 PHP 代码。比如可以用简短的 `<? ?>` 来代替 `<?php ?>`，也可以用 `<?= ?>` 来代替这样的 PHP 代码 `<?php echo 'something'; ?>`。

目前一般认为使用 `<?>` 标记是不好的，因为 XML 文档或其他语言也使用这种标记，所以我们要避免使用这种风格的编码方式。

不幸的是，本书的读者不能关闭短标记，并且仍然要使用 `<?=>` 的结构。作者认为 `<?=>` 是个很有用的快捷方式，并且经常使用它，因为在向巨大的 HTML 块里插入一些 PHP 值时这是非常方便的。本书中的代码需要打开短标签选项才能正常运行。

如果需要在不支持这种结构的服务器上运行本书的代码，只需要把代码中的 `<?=>` 都替换为 `<?php echo`，这样程序就能正常运行了。

进入与离开 PHP 模式

PHP 最大的优点之一就是可以在 HTML 页面里内嵌 PHP 代码，并且能够随意地进入与离开 PHP 模式。但是这需要仔细考虑代码对外观所带来的影响。如果只是插入一些文本，就不应该频繁地打开和关闭 PHP 代码块。在这种情况下，使用 `echo` 语句在屏幕上显示代码能够使得到的代码更简洁。在有特殊需要的情况下，甚至可以利用多行字符

串或使用 `heredoc` 语法。其他更多的方法请见 <http://php.net/string>。

一般来说应该根据常识来决定使用什么方法，尽量让代码简洁易懂。

在字符串里使用 {}

PHP 允许在字符串里利用大括号 ({}) 标识变量的引用，从而确保它们能够被正确地解析。因此，代码中可能出现这样的字符串：

```
$sstr = "{$name} said that {$pronoun}{$saction['extra']}";
```

在使用关联数组元素时，或是在代码引用变量不明确时，就会使用这种语法。

本书会始终使用 {} 符号，因为这是个很好的习惯，它不仅能够帮助程序员识别字符串中包含的变量，而且有助于避免字符串被错误地解析，而这正是调试过程中的难点。本书的两个作者都曾经因为字符串传递问题而浪费大量的时间。但如果养成用 {} 标识变量的习惯，就可以避免这样的问题。

语言结构 (echo、include、return)

PHP 中的很多“函数”实际上并不是函数，而是语言结构，如 `echo`、`include` 和 `return`。因此虽然它们允许程序员像使用函数一样把参数放在一对小括号里，但这不是必须的，而且实际上并不鼓励这样做，因为这样会导致 PHP 增加额外的工作量。

include、require 与 once 选项

有 4 种不同的语言结构可以把 PHP 文件导入另一个文件，这 4 种方法分别有不同的用途，应当根据需要有意识地使用它们。如果 Web 页面里必须包含某个 PHP 文件才能正常工作，就应该使用 `require`，因为这样在文件找不到或不能打开时，页面也不能正常打开。相应地，如果 PHP 文件对于页面只是一个可选的组成部分，就应该使用 `include`，这样当文件找不到时，页面仍然能够打开。

类似地，当相应的文件应该只加载一次时，就要使用 `require_once` 和 `include_once`。比如大多数库文件和支持文件都是这样的。使用 `once` 选项可以确保文件不会被重复加载，从而避免产生相应的问题。

echo 与 print

这两个语言结构基本上是完成相同的工作，但略有不同。`print` 会返回数值 1，而且

具有优先权，因此可以用于较长的表达式里，但实际上这种需要是很少的，而且这也让它比 `echo` 的执行速度稍稍慢了一点。

更重要的是，`echo` 可以把用逗号分隔的多个不同值显示在屏幕上。出于上述原因，本书在绝在大多数情况下都使用 `echo`。

函数、方法与类的命名

一些命名方案具有很不错的实用效果，本书所遵循的原则包括以下内容。

- 函数名：小写字母，以下划线分隔，如 `my_function_name()`。
- 方法名：StudlyCaps 格式，第一个字母小写。也就是说名称里的单词以词首大写字母分隔，但名称的第一个字母是小写的。如 `myMethodName()`。
- 类名：StudlyCaps 格式，第一个字母大写如 `MyClassName`。
- 类里私有属性和方法：以单个下划线开始，表示它们是内部的，如 `_myPrivateMethod()`。
- 变量和属性名：应该都为小写字母，可以根据需要使用下划线。
- 常数名：应该都为大写字母。

目 录

第一部分 PHP 基础知识

第 1 章 字符串	3
1.1 匹配模板（正则表达式）	6
1.2 清除空白	8
1.3 扩展和压缩制表符	8
1.4 在 Macintosh、UNIX 和 PC 格式之间转化文本	10
1.5 解析逗号分隔的数值（CSV）	11
1.6 针对指定空间来截短文本	13
1.7 填充数据数组，以列的形式显示	14
1.8 检查单词拼写	15
1.9 匹配近似的字符串	17
1.10 实现标题的正确大写	18
1.11 生成惟一标识符	20
1.12 统计某个单词出现的次数	21
第 2 章 数值	23
2.1 从字符串里提取数值	25
2.2 输出正确的复数文本	26
2.3 把数值转化为罗马数字	27
2.4 计算利息	28
2.5 模拟骰子	33

2.6	计算经纬度	34
2.7	公制/英制转化	39
2.8	温度转化	42
2.9	创建统计程序包	43
第3章 日期与时间		46
3.1	计算两个日期之间的差	48
3.2	确定指定月份的最后一天	50
3.3	闰年的计算	51
3.4	处理时区	51
3.5	处理数据库或文件中的时标	53
3.6	确定工作日的数量	54
3.7	生成指定月份的日历	56
第4章 变量		60
4.1	判断两个变量是否相等	61
4.2	从函数内部访问外部变量（全局变量）	63
4.3	在函数内部保存持久值（静态变量）	64
4.4	让一个变量指向另一个变量（引用）	65
4.5	使用变量保存另一个变量的名称	65
4.6	声明常数	67
4.7	在多个数值里判断第一个非假数值	67
第5章 数组		69
5.1	超全局数组及其应用	73
5.2	实现堆栈	74
5.3	实现队列	77
5.4	根据用户定义的比较函数进行排序	79
5.5	使用其他算法进行排序	81
5.6	递归处理多维数组	85
5.7	对数组执行集合操作	87
5.8	使用数组实现矩阵的数学运算	88

第6章 函数	96
6.1 设置可选参数	97
6.2 创建递归函数	98
6.3 使用变量中保存的名称调用函数	99
6.4 动态创建函数（ λ 样式）	101
6.5 使用函数数组	102
6.6 按引用传递或返回值	103
6.7 使用任意数量的参数	105
6.8 要求参数为特定类型	106
第7章 类和对象	108
7.1 自动加载类的源文件	110
7.2 保护对象数据（公共/私有/受保护）	113
7.3 在对象创建或销毁时自动执行代码	113
7.4 不进行实例化就访问类的成员	114
7.5 扩展类定义	116
7.6 创建抽象类	118
7.7 使用对象接口	120
7.8 动态变量名和重载变量名	122
7.9 方法重载	126
7.10 实现链表	129
7.11 实现二叉树	132
第8章 文件和目录	138
8.1 生成完整目录列表	141
8.2 文件大小的常规显示	142
8.3 重新命名一个目录里的全部文件	145
8.4 在目录树里搜索文件	147
8.5 处理相对和绝对路径	148
8.6 通过 HTTP 或 FTP 读取文件	150
8.7 实时查看文件内容（模拟 UNIX 的 <code>tail -f</code> ）	152
8.8 生成两个文件的差别报告	154

8.9 锁定文件独占使用	158
8.10 在本地缓存远程文件	162
8.11 文件压缩与解压	165
8.12 自动从父目录包含特定文件	167

第二部分 应 用

第9章 生成 Web 页面/XHTML/CSS	171
-------------------------------	-----

9.1 创建多级下拉菜单	172
9.2 突出显示用户所在的站点区域	178
9.3 显示动态进程栏	179
9.4 利用 XHTML/CSS 模拟图表	182
9.5 Web 页面的分页	185
9.6 为服务器优化而缓存 Web 页面输出	187
9.7 用不同语言实现 Web 页面本地化	189
9.8 联合使用 Ajax 和 PHP 创建交互式 Web 页面	192

第10章 处理 Web 表单	196
----------------------	-----

10.1 轻松获取表单数据	197
10.2 获取表单数据的多维数组	198
10.3 共同使用 GET 和 POST 表单数据	200
10.4 接受上传文件	201
10.5 生成选择语句	203
10.6 表单的必填字段	206
10.7 显示在公告板上的文本	208
10.8 为链接添加警示信息	210
10.9 避免表单重复提交	211

第11章 数据检验与标准化	214
---------------------	-----

11.1 电话号码	215
11.2 邮政编码	217
11.3 社会保障号 (SSN)	218

11.4 数字.....	220
11.5 信用卡号码.....	221
11.6 日期.....	223
11.7 电子邮件账户.....	224
11.8 URL.....	225
第 12 章 会话与用户追踪	227
12.1 利用 cookie 记录数据.....	228
12.2 利用会话保存用户数据.....	231
12.3 针对用户实现自定义显示设置.....	235
12.4 创建函数库用于追踪站点上的用户.....	238
12.5 实现简单的购物车.....	240
12.6 在服务器之间传递会话数据.....	244
12.7 从日志文件解析特定浏览器信息.....	246
第 13 章 Web 服务和其他协议	251
13.1 提交具有 cURL 的 POST HTTP 请求.....	252
13.2 与 LDAP 服务器通信.....	254
13.3 通过 SOAP 使用 Web 服务.....	255
13.4 连接到 FTP 服务器.....	258
13.5 创建基于 PHP 的 FTP 客户端.....	259
13.6 使用套接字连接到互联网服务器.....	264
13.7 创建基本的 Web 服务程序.....	265
第 14 章 关系型数据库	270
14.1 与 MySQL 的通信.....	271
14.2 与 Oracle 的通信.....	272
14.3 与 PostgreSQL 的通信.....	274
14.4 与 Sybase 的通信.....	275
14.5 与 Microsoft SQL 服务器的通信.....	277
14.6 与 SQLite 的通信.....	279
14.7 通过 ODBC 与数据库进行通信.....	280
14.8 使用抽象层与数据库进行通信 (PDO).....	281