

PHP Developer's Cookbook

PHP

经典实例

STERLING HUGHES 等著
徐牧 贾东升 付清朝 许斌 等译

SAMS



中国电力出版社
www.infopower.com.cn

PHP

经典实例

STERLING HUGHES 等著
徐牧 贾东升 付清朝 许斌 等译

中国电力出版社

内 容 提 要

本书全面系统地介绍了 PHP 的基础知识和一些相关技术。全书共分五大部分 21 章, 对 PHP 的具体语法和各种技术应用进行了深入而详实的介绍。五个部分分别介绍了 PHP 的具体语法结构、数据库技术、网络应用技术、处理 XML, HTML 以及图像的相关技术、Zend API 技术等内容。书中通过与其他语言进行对比, 使读者对于 PHP 的特点能够有比较全面深入的理解。同时, 所有解决方案均提供有源代码, 从而给读者提供了一个很好的应用参考。

本书针对的读者是 PHP 开发人员。书中提供了最新版本的关于 PHP 语言的完整参考手册, 适合广大的 Web 应用程序开发人员和计算机编程爱好者阅读。

图书在版编目 (CIP) 数据

PHP 经典实例 / (美) 休斯著; 徐牧等译. —北京: 中国电力出版社, 2003

ISBN 7-5083-1290-2

I . P... II . ①休...②徐... III. PHP 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 007938 号

著作权合同登记号 图字: 01-2002-4753 号

Authorized translation from the English language edition, entitled PHP Developer's Cookbook by Sterling Hughes, published by Sams Publishing, Copyright©2002

All rights reserved. NO part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

CHINESE SIMPLIFIED language edition published by China Electric Power Press Copyright©2002

本书中文版由美国培生集团授权出版, 版权所有。

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2003 年 4 月第一版 2003 年 4 月北京第一次印刷

787 毫米×1092 毫米 16 开本 24.25 印张 536 千字

定价 39.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

序

回顾 20 世纪 80 年代末和 90 年代初,当时世界正从使用拨号上 BBS 系统向新兴的 Internet 体系过渡。但这些只局限于工科院校的学生和那些狂热的计算机业余爱好者。许多相当不错的程序(例如 Gopher、Veronica、Archie 和 IRC)的出现,都预示着新事物即将到来。

在 1993 年 4 月, Mosaic 开始进行综合性的开发工作。当 Netscape 1.0 于 1994 年 12 月发布时,网络技术正呈现高速发展的态势。当时的 Web 服务器来自于 CERN 和 NCSA,而且不久发布的 Perl 5 很快就成为开发动态 Web 站点的脚本语言。Internet,作为万维网(World Wide Web)这一美好名词的实质内容,越来越面向大众。

在 1994 年,编写 Web 应用程序还有些冒险。一方面,系统管理员不得不使用专门设计的新型的 Perl 语言来处理大量日志文件;另一方面,用户不得不编写 C 程序处理大量的文本文件以及生成 HTML 输出。由于 Web 编程涉及到分析文本输入以及生成文本输出,对于大多数用户来说, Perl 会比 C 更加适合。

为了说明问题,下面举例对比一下。假设我们要创建一个简单的表单,用来询问用户的姓名和年龄,并且当用户单击“提交”按钮之后,将相应的信息送回并显示。下面是完成这个例子的 C 语言代码:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

#define ishex(x) (((x) >= '0' && (x) <= '9') || ((x) >= 'a' && (x) <= 'f') || \
                ((x) >= 'A' && (x) <= 'F'))

int htoi(char *s) {
    int    value;
    char    c;

    c = s[0];
    if(isupper(c)) c = tolower(c);
    value=(c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10) * 16;

    c = s[1];
    if(isupper(c)) c = tolower(c);
    value += c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10;
    return(value);
}

void main(int argc, char *argv[]) {
    char *params, *data, *dest, *s, *tmp;
```

```

char *name, *age;

puts("Content-type: text/html\r\n");
puts("<html><header><title>Form Example</title></header>");
puts("<body><h1>My Example Form</h1>");
puts("<form action=\"form.cgi\" method=\"POST\">");
puts("Name: <input type=\"text\" name=\"name\"><p>");
puts("Age: <input type=\"text\" name=\"age\"><p>");
puts("<input type=\"submit\">");
puts("</form>");

data = getenv("QUERY_STRING");
params = data; dest = data;
while(*data) {
    if(*data=='+') *dest=' ';
    else if(*data == '%' && ishex(*(data+1)) && ishex(*(data+2))) {
        *dest = (char) htoi(data + 1);
        data+=2;
    } else *dest = *data;
    data++;
    dest++;
}
*dest = '\0';
s = strtok(params,"&");
do {
    tmp = strchr(s,'=');
    if(tmp) {
        *tmp = '\0';
        if(!strcmp(s,"name")) name = tmp+1;
        else if(!strcmp(s,"age")) age = tmp+1;
    }
} while(s=strtok(NULL,"&"));

printf("Hi %s, you are %s years old\n",name,age);
puts("</body></html>");
}

```

对于编码员来说，这段代码并不糟糕，只是需要将其改成 HTML 时不太好办。要改成 HTML，就不得不重新编译 CGI 程序。Perl 之所以流行，是因为使用 Perl 编程，不仅能够很容易地编写处理表单的代码，而且不需要重新编译 Perl 脚本文件。以下是 Perl 中使用 CGI.pm 创建同样表单的代码：

```

use CGI qw(:standard);
print header;
print start_html('Form Example'),
    h1('My Example Form'),
    start_form,
    "Name: ", textfield('name'),
    p,
    "Age: ", textfield('age'),

```

```

    p,
    submit,
    end_form;
if(param()) {
    print "Hi ",em(param('name')),
        "You are ",em(param('age')),
        " years old";
}
print end_html;

```

这种方法编写和阅读都要容易得多，但还是从一个编码员的角度去看问题的。随着 Web 的流行，由非编码员所创建的 Web 页逐渐增多。在刚开始时，他们只关心静态内容，并且花去大量的时间学习复杂的 HTML。但很快就需要在站点中增加动态内容，这时，使用 C 语言编写 CGI 程序对于大多数非编码员来说是不现实的。所以，一些勇敢者就开始尝试学习 Perl，并掌握了 CGI.pm 样式以及其他 Perl 中有帮助模块。

这正是 PHP 的定位。今天，我多么想说当时我就已经清楚地认识到 PHP 能很好地担任这个角色，但事实并非如此。当时我发现对于每一个请求，运行一个 Perl 解释程序的开销太大了，而且使用 C 语言编写 CGI 程序过于繁杂。但我还是不得不用 C 来编写所有类型的 CGI 程序，一遍又一遍地重复编写相同的代码。我所需要的是一个简单的函数封装，以便将 CGI 脚本中的 HTML 部分从 C 代码中分离开来，这样无需重新编译 C 代码就可以对 HTML 进行修改。这种理念就变成了 PHP，前面的 C 代码和 Perl 代码在 PHP 中就变成如下形式：

```

<html><header><title>Form Example</title></header>
<body><h1>My Example Form</h1>
<form action="form.phtml" method="POST">
Name: <input type="text" name="name">
Age: <input type="text" name="age">
</form>
<?if($name):>
Hi <?echo $name?>, you are <?echo $age?> years old
<?endif?>
</body></html>

```

与 C 和 Perl 版本相比，这很像 HTML。这曾经是，也仍然是 PHP 吸引人的地方。用户使用 PHP，并不想学习如何去编程，只是想使他们编写的 HTML 页更加动态化。

近几年来，PHP 作为一种语言逐渐发展起来，现在它已经成为一种真正意义上的编程语言。但是，它并没有丧失使用户快速地完成简单操作的强大功能。而且，对于非程序员来说，它是十分友好的。在某种程度上，PHP 之所以是一种很好的编程语言，在于它可以使没有任何编程经验的人去完成某项任务。在他们使用 PHP 的过程中，会不知不觉地学习到编程的方法。

PHP 使用起来十分方便，而且通过自学就可以掌握。但有时，我们对其他人如何解决简单或者复杂的问题更感兴趣。而且很多时候，只凭一个人往往不能想到解决问题的巧妙方法，而这些正是本书所能提供和弥补的。Sterling 和 Andrei 编写的这本解决方案手册，向读者展

示了如何解决一些特殊的问题，同时为用户提供了解决问题的基本原理。

Sterling 在 2000 年初加入了 PHP 开发小组，并且对 Curl、SWF 和 Sablotron 等扩展做出了贡献。Andrei 从 1999 年以来一直是开发 PHP 的骨干力量。他的直接贡献主要在于 Perl 类型的正则表达式和基于 XML 的特性（如 WDDX、会话支持和一些通用的处理数组的函数）。

如果你喜欢这本书，并且愿意使用 PHP 的话，就请像 Sterling 和 Andrei 一样加入到开发 PHP 的行列中来吧！这样的工作将会十分有趣，而且对正在流行的开源项目（Open Source Project）也会做出很有价值的贡献。你所做的工作将被成千上万的用户所使用，并且你从同行那里赢得的尊重是无价的。最重要的是，你可以推动自由软件业的持续发展。新人和新思想正是类似 PHP 这样的项目发展的驱动力量。

——Rasmus Lerdorf (rasmus@php.net)

作者简介

Sterling Hughes 是一位自由软件开发人员，具有 7 年的工作经验，主要从事高端的 Web 应用程序的开发。他长期为 Webreference、Devshed、Zend 和 Webtechniques 撰写文章，经常在全球各地举行的各种讨论会上做关于 PHP 的精彩演讲和报告。读者可以通过电子邮件的方式与他联系，他的邮箱地址是 sterling@php.net。

Andrei Zmievski 是一位软件开发工程师，同时又是从事开源研究的研究员。他在 ispi 从事多种电子商务和网上信息发布（Web publishing）项目的研究。他是 PHP 核心开发小组的成员之一，在其他的开源项目中也做出了很大的贡献。生活中，他还喜欢弹钢琴和计算机绘图，他的邮箱地址是 andrei@php.net。

致 谢

献给我的母亲，感谢她所给予我的无条件的爱和支持。

——Sterling

感谢

通常有一种误解，认为一本书只是由作者一人完成的。实际上，一本书的完成是许多人合作的结果。

首先，我要感谢所有 Sams Publishing 的职员，感谢他们的支持和第一时间的指导。感谢我的策划编辑 Elizabeth Finney，在检查文稿以及做技术修改的过程中，都给予我巨大的帮助。感谢协调员 Amy Patton，他组织了许多幕后的工作，从而大大提高了本书的编写质量。感谢 Jeff Schultz 给了我编写此书的机会。感谢负责资料收集工作的编辑 Shelley Johnston 所付出的热情和支持。感谢负责开发工作的编辑 Scott Meyers，他帮助我扩展了本书的内容，并对整个编写过程进行了监督。感谢文本输入编辑 Mike Henry，他为将手稿变成连贯的语言而付出了巨大努力。还要感谢联合发行人 Mark Taber 延长了出版期限，使得作者可以有额外的时间进行技术上的编辑。感谢负责格式编辑的 Katie Robinson，他花费了很大的精力来将我杂乱的文稿转换成 Sam 的标准格式。

本书共有 6 位技术编辑，他们的工作避免了技术上的错误。他们都很辛苦，对书中的每一段内容都要仔细核对，以确保本书的编写质量。Patrick Lambert 审阅了整篇文稿并提出了许多很有价值的意见。Chris Newman、Kenneth J.Kloeppel、Mark Maslakowski、Brian Schaffner 和 Ken Jenks 分别是本书不同部分的技术编辑。其中 Chris Newman 和 Kenneth J.Kloeppel 还做了许多本职以外的工作。

除了技术编辑之外，本书还有 5 位技术审阅者，他们通读整篇文稿以确保文稿的正确性和风格。Zak Greant、Florian Lanthaler 和 Sebastian Bergmann 都审阅了整篇文稿，并提出了宝贵意见。Jannis Hermanns 和 Landon Bradshaw 审阅了本书的不同部分。这里要特别感谢 Zak Greant 和 Florian Lanthaler，他们的审阅非常细致，远远超过了对技术审阅所做的要求。

特别感谢 Andrei Zmievski，没有他宝贵的工作，就没有本书的出版。他对 PHP 的了解是无人可比的。

特别感谢 Rasmus Lerdorf，首先感谢他开发了 PHP，其次感谢他在百忙之中还为本书作了序。

感谢所有为 PHP 的发展投入宝贵时间的人们。他们包括 PHP 研究小组、PHP 开发小组、文献工作组、质量保证小组以及其他各小组的成员，还有所有为 PHP 提供支持的人。他们或者出书，或者撰写技术性文章，或者致力于 IRC 的研究。没有这些人的辛勤工作，就不会有本书的出版。

还要感谢我的老师和朋友，感谢他们在我编写此书遇到疑难时所给予我的关心和理解。

我的家人是我的精神支柱。没有他们一贯的支持，我不可能完成此书的编写工作。我非常地爱你们。特别感谢我的祖父和祖母，他们总是强调教育的重要性，并教导我去珍惜生活。

我要感谢的最后一个人是我的母亲，我无法用语言来表达我的感激之情。妈妈，您不仅给予了我生命，还让我在生活的道路上继续前进。我爱您。

——Sterling

我要感谢所有参与 PHP 工作的人，包括 PHP 内核开发小组、文献工作组、质量保证小组以及所有其他的开发人员，他们投入了大量的时间以使 PHP 尽善尽美。特别感谢 Rasmus Lerdorf，是他让我在这片领域中驰骋。

另外，要感谢 Monte Ohrt、Eric Elliott、Luc Suryo 和 David Hahn，他们使 ispi 成为一个工作的好地方。同时他们的意见和支持都是无价的。

我还要感谢所有 Sams Publishing 的职员，是他们使本书得以呈现在读者面前，特别是 Sheley Johnston 和 Amy Patton。

最后，感谢我的父母亲。

——Andrei

意见反馈

作为本书的读者，你们是我们最值得信赖的批评家和评论家。我们将重视你们的看法和观点，同时也想知道我们哪些地方做得好，哪些地方需要改进，需要补充什么内容。我们会接受你们所给予的所有金玉良言。

大家可以写信或发电子邮件来和我直接联系，以使我知道本书中让人满意和不满意的地方，以及还有什么地方需要进一步完善。

请大家注意的是，我不能够帮助你解决本书中所涉及到的技术问题，并且由于收到的邮件太多，不可能一一答复。

请大家在写信时，注明书的作者和名称以及您的名字、电话以及传真。我将认真考虑您的建议并将它反馈给书的作者。

Email: webdev@sampublishing.com

Mail: Mark Taber

Associate Publisher

Sams

201 West 103rd Street

Indianapolis, IN 46290 USA

译者序

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，形成了独特的风格。它使 Web 开发人员能够快速编写出动态生成页面的脚本。

PHP 是一种开源的脚本语言，主要用于开发 Web 应用程序。同 ColdFusion 和 ASP 一样，它也具有内嵌和独立的脚本功能，而且也像 C++ 和 Java 一样具有面向对象的结构体系。同时，PHP 简单易学并且是一种完全免费的自由软件。PHP 简洁直观的结构和大家熟知的语法让人学习起来得心应手。而且 PHP 可以在其官方站点上 (<http://www.php.net>) 自由下载。PHP 在大多数 UNIX 系统平台、GUN/Linux 和 Windows 平台上均可运行。

1995 年，PHP 创始人 Rasmus Lerdorf 对外发布了第一个 PHP 版本。之后，PHP 迅速流行起来。1997 年，开始进行 PHP3 的开发。PHP3 几乎支持所有主流和非主流的数据库，再加上高速的执行效率，使得 PHP 在 1999 年中的使用网站数量超过 15 万。目前，最新版本为 PHP 4。整个脚本程序的核心大幅改动，使得 Zend 核心引擎性能更加优越，程序的执行速度更快。PHP 最佳化之后的效率，较之传统的 CGI 或者 ASP 等程序来讲，有了很大的提高。

PHP 能够诞生并且发展到今天的规模，是全世界各位免费代码开发者共同努力的结果。也是 Internet 的代表产物，是代码免费自由交流的胜利成果。具有更强的新功能、更丰富的函数库的 PHP 必将在 Web 领域里掀起翻天覆地的革命。

随着 PHP 开发人员和使用者的增多，人们越来越需要更多有关 PHP 的书籍问世。本书的翻译出版相信会为广大 PHP 使用者提供一个很好的选择。本书既是一本参考手册，又是一本指导性书籍，它全面系统地介绍了 PHP 基本知识和一些相关技术。本书作者 Sterling 和 Andrei 都是 PHP 方面的专家。他们收集整理了解决各类简单或者复杂问题的方法和技巧，同时向读者提供了解决问题的基本原理。

全书介绍了 PHP 的具体语法结构、数据库技术、网络应用技术、处理 XML 及 HTML 以及图像的相关技术、Zend API 技术等内容。本书列出了所有解决方案的源代码，给读者提供了一个很好的应用参考。通过此书，读者可以更好地掌握 PHP 的相关技术。

本书的翻译工作由虎贲工作室组织，主要部分由徐牧、贾东升、付清朝、许斌、张磊、赵锋、林小青等人完成，另外还有王英、林敏、雷易平、张静、程信林、王江书、刘培英、罗毅、王勇、薛宏伟、陈亮亮、金江、任芳和李玲等人分别参加了部分翻译、校对、录入和排版等工作，全书最后由邓少鹄统校。译者尊重原文的表述内容，翻译力求简明易懂。但由于时间仓促，且经验和水平有限，在翻译过程中难免有不足之处，恳请读者批评指正。

译者

前言

“经典的东西每个人都希望自己已经读过，但每个人都不愿意去拜读。”

——Mark Twain

目的

我有大量的编程实践方面的书籍——它们有着质朴的内容，被我放在书架上。这些书籍影响着我编程的方法和风格。它们是“宝典”，程序员通过它们形成自己整个的思维理念。

但是，我还有一批书。这些书放在我的办公桌上、床上和卧室的沙发椅上。他们已经残缺，皱得变形并且滴满了咖啡污渍。但这恰恰是我给予这些书的最大荣誉，因为这意味着它们很实用，它们是我解决每天遇到的难题的参考。

本书可以看成是这样的一本书：它精心编写的目的是为了帮助程序员解决日常编程中所遇到的难题。能够给予本书的最大荣誉就是去使用它——从中发现对你有帮助的地方，即使只是用来擦拭不小心滴上的咖啡。

如何使用本书

本书的特别之处在于它既是一本参考手册，又是一本教科书。尽管它并不是纯粹的参考手册，但它确实包含一些以前没有被记载的资料，而且是很多函数文档的补充。尽管它不是纯粹的教科书，但是通过对本书的阅读，会使你成为一个更好的程序员，并且为你解决难题提供更有效的方法。

本书不应该看成是 PHP 惟一的参考手册，而只是一个补充性说明。如果读者想要找一本印刷成册的参考手册的话，这里推荐 Leon Atkinson 的《Core PHP》一书，由 Prentice Hall 出版发行。此外还有 Rasmus Lerdorf 的《PHP: Pocket Reference》一书，由 O'Reilly 出版。

另外，本书也不能视作学习 PHP 的入门书籍。阅读本书需要对 PHP 语言有基本的理解。如果读者想要找一本入门书籍的话，Julie Meloni 的《PHP Fast & Easy Web Development》是不错的选择。

本书在逻辑上没有什么承接关系，所以读者可以选择从任一章节开始阅读，这和从头开始阅读没什么区别。本书作为一本参考手册，当读者需要了解某一方面的内容时，只需查阅本书适当的章节（查询目录或索引），然后翻到相应的页码阅读就可以了。

每章概述

本书的每一章都涵盖程序开发人员日常所遇到的普遍问题。

第 1 章“字符串的使用”，是有关字符串操作的内容，包括：每次逐个字符地去处理字符串，以及将字符串转换到不同的字符集等。

第 2 章“数值、日期和时间”，都是关于数字方面的内容，包括：任意精度数字，生成有偏估计的随机数，以及 PHP 内置的处理和获取日期和时间的函数等。

第 3 章“数组”，讨论如何合并数组以及将数组自然排序的方法等内容。

第 4 章“PHP 的内置数组和常量”，解释了如 `_FILE_` 常量和 `$HTTP_POST_VARS` 数组这些概念。

第 5 章“数据和正则表达式的匹配”，讲述数据匹配，验证数据的合法化，数据解析等。讨论从匹配一个 E-mail 地址到在 PHP 中使用与 Perl 兼容的正则表达式这些内容。本章还用了很长的篇幅来介绍基本正则表达式的语法结构。

第 6 章“文件处理”，包括打开文件，将文件上锁，创建临时文件，处理文件内容等知识。

第 7 章“目录文件”，是有关目录访问的知识。详细解释了如何移动整个目录树，搜索单个目录树，本中的结束部分将给出一个完整的搜索引擎和分度器 (indexer)。

第 8 章“函数”，讨论如何为函数建立默认值，以及基于用户输入的函数动态声明这样的问题。

第 9 章“类”，研究 PHP 的面向对象特性，包括在 PHP 中声明某一类以及声明析构函数等内容。

第 10 章“使用 PHP 维护会话”，包括 PHP 4 的会话管理特性以及 WDDX 等内容。本章中也特别举了一个购物车系统的实际应用。

第 11 章“与 Web 页和服务器结合”，讨论 Web 自动化的有关内容。包括自动提交 Web 表单和动态生成 JavaScript。

第 12 章“用 PHP 创建独立于数据库的 API”，讨论在程序中创建与 PHP 3 兼容，与数据库独立的 API 的过程。

第 13 章“与其他程序和语言的接口”，本章讨论如何使用 `exec()` 函数和 `system()` 函数的基本概念，然后深入讨论了更多高级特性，例如处理 COM、.NET 和 Java 对象。

第 14 章“与套接字通信”，内容包括如何打开和使用 SSL 连接，以及使用 PHP 的套接字扩展来创建服务器。

第 15 章“处理电子邮件”，详细讨论了如何在 PHP 中处理电子邮件。介绍基本的发送纯文本形式的电子邮件，然后讨论电子邮件的高级特性，例如发送 HTML 形式的电子邮件和带附件的电子邮件，以及处理 IMAP 邮件信箱的过程。

第 16 章“处理 SNMP 对象”，第 17 章“LDAP”，讨论 LDAP 协议和 SNMP 协议。我们将解决所有与 LDAP 和 SNMP 扩展有关的常见问题。

第 18 章“创建并管理图像”，讨论了使用 PHP 和 GD 库来创建和处理图像的过程。

第 19 章“HTML”，介绍了如何使用 PHP 处理 HTML 文件。

第 20 章“XML”，介绍 Expat 库和 DOM XML 库，讨论所有分析 XML 过程中遇到的常见问题。

第 21 章“Zend API”，介绍了诸如参数获取之类的基本知识，以及向已存在的对象中增加方法之类的高级操作。

DB.php 及其在本书中的作用

在需要对数据库进行访问的章节中，我们使用 DB.php 访问数据库。DB.php 仅在 PHP 4 中使用，而且在编写此书时，其技术还不完全成熟，所以没有在第 12 章中全面地讲述。有关 DB.php 的更多信息，参见 <http://pear.php.net/> 上 PEAR 提供的资源。

PHP 的历史

PHP 和其他重大的事物一样，产生于需求。Rasmus Lerdorf 在某天的下午编写了 PHP 的最初版本，当时他只是需要一个快速的工具来跟踪在线履历的访问者。但是，这个用 Perl /CGI 封装的最初版本，在 Web 服务器上使用时却遇到了越来越多的问题。因此，Rasmus 决定用 C 语言重写此 Perl 程序，以消除履历每次被访问时用于派生 Perl 所需的开销。

不久，在相同 Web 服务器上的人们注意到了 Rasmus 的程序，并向他询问是否可以使用。PHP 就这样发展起来了。随着越来越多的人使用此程序，他们要求增加更多的特性，这样一来，越来越多的人加入到使用者的行列。

最后，Rasmus 决定发布一个完整的版本，它包括一个程序文档（document），一个邮件列表（mailing list）和一个 FAQ。第一个版本命名为 Personal Home Page Tools，之后不久又演变成 Personal Home Page Construction Kit。

当 Rasmus 开发 PHP 的同时，他还在开发另一种能够将 SQL 查询方便地嵌入到 Web 页中的工具。从根本上说，它是另一个 CGI 封装，可以用来分析 SQL 查询，并使基于查询创建表格、表单的工作变得很简单。这个工具就被命名为 FI（Form Interpreter）。

PHP/FI 2.0 是将 PHP 和 FI 两者重新编写到一个程序中而形成的。PHP/FI 是一种简单的编程语言而不是一种简单的 CGI 包装。它允许程序员在 Web 页中直接嵌入简单脚本。

PHP/FI 在程序开发者中很快就流行起来，它吸引了 Andi Gutmans 和 Zeev Suraski 的注意。他们虽然很欣赏 PHP/ FI 的基本思想，但是他们对于 PHP/ FI 的执行效果不甚满意，于是 Andi Gutmans 和 Zeev Suraski 重写了 PHP/FI，PHP 3 诞生了。

PHP 3 是对 PHP/ FI 的重大改善。其特色是具有更好的执行效果，更清晰的语法结构，支持面向对象的方法，支持 for 循环的使用，还有其他方面的改善。PHP 3 取得了令人难以置信的成功。根据 Netcraft 的最新调查结果，PHP 3 是到目前为止最流行的 Apache 服务器端模块，大约有 150 万 Web 服务器上装有 PHP 3。

进入 PHP 4

尽管 PHP 3 引擎极为流行且执行起来高效快速，但随着越来越多的人使用它来进行复杂的应用，它也出现了许多不尽人意的地方。因此，PHP 3 引擎的创始者 Andi Gutmans 和 Zeev Suraski 决定彻底重写核心引擎。

重写的结果是 Zend 引擎的诞生（以 Zeev 和 Andi 命名）。Zend 引擎是 PHP 4 的核心，它在性能方面改善了许多，并提供了许多新的特性。它扩展了 PHP 的功能，使之成为复杂的 Web 应用程序以及建设中小型 Web 站点的优秀工具。

性能改善

PHP 4 最强大的特性之一是较之 PHP 3 在性能方面的显著增强。来自 Zend 的测试表明相同的脚本在 PHP 4 中运行与在 PHP 3 中运行相比，性能会有 50%~1 000% 的提升。也可以使用带有免费的 Zend 优化器（可以从 Zend 技术支持的站点上获得，<http://www.zend.com/>）的 PHP 4，它能使 PHP 4 的速度提高 40%~100%，速度提高的程度由脚本的结构所决定。

支持面向对象技术

尽管 PHP 4 仍然不是一种功能强大的面向对象语言（如 Java 或 C++），但 PHP 4 还是大大改善了对面向对象程序设计的支持。包括了对间接继承、直接访问类的方法和每对象变量的支持。

支持数组处理

PHP 4 提供了丰富的用于访问和处理数组的新函数。这些函数包括：array_splice()、array_push()、array_pop()、array_diff()等，但不仅仅限于这些。

完整的会话支持

PHP 3 最大的问题之一就是它缺乏对会话管理系统的完全支持。用户不得不使用自己编写的会话程序，或者依赖外部的会话管理系统，如 PHPLIB 的会话处理函数。但随着 PHP 4 的出现，PHP 工作组定义了一个新的会话扩展来支持会话。

Java 集成

PHP 4 可将 PHP 与 Java 库和 Java servlet 集成在一起。只需创建一个新的 Java 对象，然后调用该方法并访问它的属性即可。

输出缓存支持

PHP 4 提供了一个输出缓存子系统，允许用户访问脚本的输出、处理这些输出并对输出的结果进行操作。此特性的一种可能应用是用来创建一个网页高速缓存系统（page-caching system）。

新的扩展

PHP 4 新增了许多 PHP 3 没有的新扩展。包括 swf、curl、exif、cybercash、sockets 以及 ingres_ii 扩展，但不仅仅限于这些。

PEAR

PHP 扩展和应用仓库（PHP Extension and Application Repository, PEAR）。PEAR 的概念与 Perl 的 CPAN 很相似——它是由 PHP 类和支持的扩展所组成的库，有利于实际的编程。例如 File_Find 类就是通过 PEAR 发布的。此类允许用户映射(map)和搜索不同的目录树。有关 PEAR 的更多信息，可以访问 <http://pear.php.net>。

目 录

序
译者序
前 言

第一部分 语法结构和相关技术

第 1 章 字符串的使用.....	3
1.0 引言.....	3
1.1 串的分析.....	3
1.2 三元运算符.....	6
1.3 变量交换.....	7
1.4 ASCII 码转换为字符.....	8
1.5 拆分字符串.....	8
1.6 字符串中的反转.....	10
1.7 字符串大小写的转换.....	11
1.8 删除字符串中的空白.....	12
1.9 避开特殊字符.....	13
1.10 读取一个带有逗号分隔符的文本文件.....	14
1.11 分析 URL.....	15
1.12 模糊匹配.....	16
1.13 创建惟一标识符 (ID).....	17
1.14 字符串加密.....	17
1.15 Cyrillic 字符集的转换.....	19
第 2 章 数值、日期和时间.....	21
2.0 引言.....	21
2.1 检查变量是否为一个有效数字.....	21
2.2 一批数字的操作.....	22
2.3 使用非浮点型或非整型的数字.....	23
2.4 任意精度数的取舍.....	25
2.5 数在不同进制之间的转换.....	26